



Pusat Penelitian dan Pengabdian Masyarakat **STMIK Nusa Mandiri**

SURAT TUGAS **165/B.01/PPPM-NM/XI/2020**

Tentang

**Pelindungan Ciptaan di Bidang Ilmu Pengetahuan, Seni dan Sastra Berdasarkan Undang-Undang
Nomor 28 Tahun 2014 tentang Hak Cipta
Nomor dan Tanggal Permohonan: EC00202057324, 8 Desember 2020
Nomor Pencatatan: 000224466**

**PADA SURAT PENCATATAN CIPTAAN
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA REPUBLIK INDONESIA**

Program Komputer

Judul Ciptaan :

Aplikasi LEMES (Learning Management System)

MEMUTUSKAN

- Pertama : Kepada saudara
Hafis Nurdin S.Kom, M.Kom
Sebagai Pencipta yang mempublikasikan karyanya.
- Kedua : Mempunyai tugas sbb:
Melaksanakan Tugas yang diberikan dengan penuh rasa tanggung jawab.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila
dikemudian hari terdapat kekeliruan akan diubah dan diperbaiki sebagaimana
mestinya.

Jakarta, 9 November 2020

Ketua PPPM

Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri

Norma Yurnita, M.Kom

Tembusan

- Ketua Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
- Arsip
- Ybs





REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202057324, 8 Desember 2020

Pencipta

Nama : **Sumarna, Mohammad Roffi Suhendry dkk**
Alamat : Jl. Dewi Sartika No.77 Cawang, Jakarta Timur, DKI JAKARTA, 13630
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Sumarna, Mohammad Roffi Suhendry dkk**
Alamat : Jl. Dewi Sartika No.77 Cawang, Jakarta Timur, DKI JAKARTA, 13630
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Aplikasi LEMES (Learning Management System)**

Tanggal dan tempat diumumkan untuk pertama kali di : 8 Desember 2020, di DKI Jakarta
wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000224466

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan pemohon.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Sumarna	Jl. Dewi Sartika No.77 Cawang
2	Mohammad Roffi Suhendry	Jl. Dewi Sartika No.77 Cawang
3	Hafis Nurdin	Jl. Dewi Sartika No.77 Cawang
4	Taufik Rahman	Jl. Dewi Sartika No.77 Cawang
5	Herman Kuswanto	Jl. Dewi Sartika No.77 Cawang
6	Aji Sudibyo	Jl. Dewi Sartika No.77 Cawang
7	Anton	Jl. Dewi Sartika No.77 Cawang

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Sumarna	Jl. Dewi Sartika No.77 Cawang
2	Mohammad Roffi Suhendry	Jl. Dewi Sartika No.77 Cawang
3	Hafis Nurdin	Jl. Dewi Sartika No.77 Cawang
4	Taufik Rahman	Jl. Dewi Sartika No.77 Cawang
5	Herman Kuswanto	Dewi Sartika No.77 Cawang
6	Aji Sudibyo	Jl. Dewi Sartika No.77 Cawang
7	Anton	Jl. Dewi Sartika No.77 Cawang



Aplikasi LEMES

(Learning Management System)



Login sebagai admin

admin : admin



Sign In to E-Learning Rekind

Happy to see you again!

[Forgot Password ?](#)

[Sign In](#)

Managers Management di gunakan untuk menentukan privilege seorang user sebagai manajer dari sebuah divisi

E-Learning

Access by: Administrator

+ Add Manager

Data Users

Show 10 entries Search:

No	Organization	Manager
1	Information System Development Department	Budi
2	Budget & Control Department	Herman

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2020 Mohammad Roffi S. All Rights Reserved. Crafted with ❤ by Lavalite

User Management digunakan untuk menambah user berikut dengan privilege nya, pada menu ini terdapat 3 buah privilege (user, administrator dan instruktur)

User : Orang yang berhak mengikuti course (melihat materi, mengikuti quiz ataupun tanya jawab dengan instruktornya.

Instructor : Orang yang mengelola course (membuat materi, membuat quiz, manage user serta melakukan tanya jawab dengan user)

Administrator : Orang yang mengelola keseluruhan isi dari elearning

E-Learning

Administrator

List Course

My Course

My Profile

Forum

Instructor

Category

Create Course

User Enroll

Send Email

Administrator

Users Management

Managers Management

Data Users

Show 10 entries

Search:

No	NPK	Name	Email	Organization	Role	
1	admin	Administrator	admin@gmail.com	Knowledge & Talent Management Division	Administrator	
2	budi	Budi	budi@gmail.com	Information System Development Department	User	
3	herman	Herman	herman@gmail.com	Budget & Control Department	User	
4	instruktur	Instruktur	instruktur@gmail.com	Knowledge & Talent Management Division	Instructor	
5	1	User 1	user1@gmail.com	Building Management & Office Facility Department	User	
6	2	User 2	user2@gmail.com	Organization Development Department	User	

Showing 1 to 6 of 6 entries

Previous 1 Next

Login instruktur

Instruktur : 123



Sign In to E-Learning Rekind

Happy to see you again!

[Forgot Password ?](#)

[Sign In](#)

Seorang intruktur wajib membuat course sebagai sarana belajar mengajar dengan usernya.

 **E-Learning**

User

List Course

My Course

My Profile

Forum

Instructor

Category

Create Course

User Enroll

Send Email

Create Course

Access by: Instructor

Create Course

Title

Title

Instructor

Instruktur

Category

Training A

Description

 **B** U  Nunito Sans **A**  

Course enroll merupakan sebuah fitur yang dapat digunakan oleh seorang instruktur untuk menambahkan atau mengurangi seorang user kedalam sebuah course.

The screenshot shows the 'Course Enroll' page in an 'E-Learning' system. The interface includes a dark sidebar with navigation links for 'User' (List Course, My Course, My Profile, Forum) and 'Instructor' (Category, Create Course, User Enroll, Send Email). The main content area is titled 'Course Enroll' with the subtitle 'Access by: Instruktur'. It features a 'Data Course' section with a table listing available courses. The table has columns for 'No', 'Title', 'Instructor', 'Date', and 'Category'. A single course is listed: 'Mobile Game' by 'Instruktur' on '14 October 2020' under the 'Computer & Science' category. An 'Enroll' button is present next to this entry. Below the table, it says 'Showing 1 to 1 of 1 entries' and includes 'Previous', '1', and 'Next' pagination controls. A search bar is located at the top right of the data section.

E-Learning

User

- List Course
- My Course
- My Profile
- Forum

Instructor

- Category
- Create Course
- User Enroll
- Send Email

Course Enroll
Access by: Instruktur

Course Enroll

Data Course

Show 10 entries Search:

No	Title	Instructor	Date	Category	
1	Mobile Game	Instruktur	14 October 2020	Computer & Science	Enroll

Showing 1 to 1 of 1 entries

Previous 1 Next

Menambahkan user kedalam course mobile game (contoh)

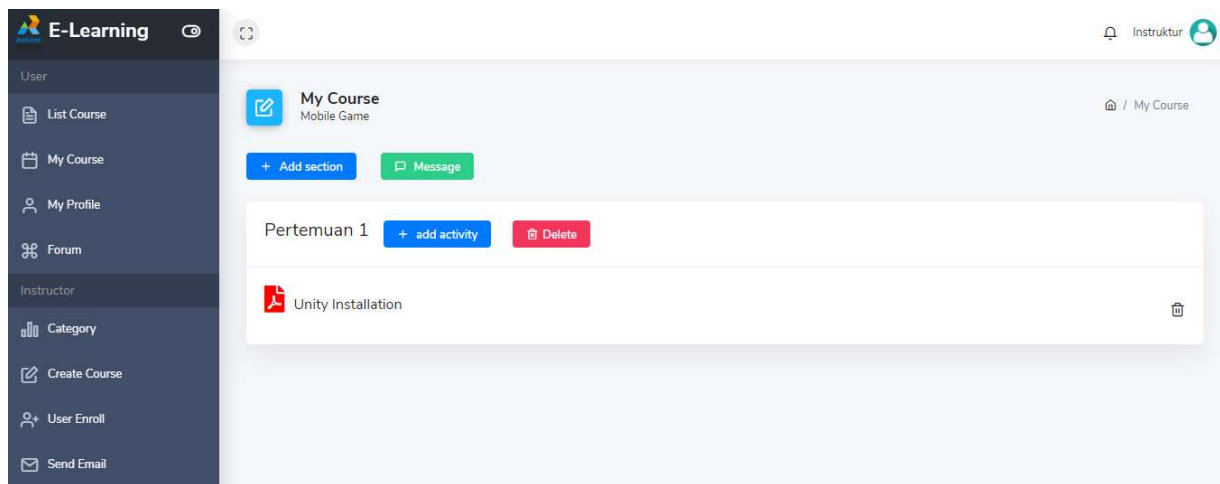
The screenshot displays the 'E-Learning' system interface. On the left is a dark sidebar with navigation options for 'User' (List Course, My Course, My Profile, Forum) and 'Instructor' (Category, Create Course, User Enroll, Send Email). The main content area shows the date '14 October 2020' and a blue 'Enroll' button. Below this is the 'Users Enrolled' section, which includes a search bar and a table of enrolled users.

No	NPK	Name	Email	Approval Status	
1	1	User 1	user1@gmail.com	Approved	×
2	2	User 2	user2@gmail.com	Approved	×

Showing 1 to 2 of 2 entries

Navigation: Previous | 1 | Next

Seorang instruktur wajib memberikan materi kepada user.



Selain menambahkan materi, seorang instruktur juga dapat menambahkan quiz, soal pada quiz akan tersimpan kedalam bank soal dan akan diberikan kepada user secara random sesuai dengan jumlah yang di inginkan oleh instruktur nya.

The screenshot shows the 'Add Quiz' page in an E-Learning system. On the left is a dark sidebar with a menu. The top of the sidebar has the 'E-Learning' logo and a user icon. Below this, the menu is divided into two sections: 'User' and 'Instructor'. The 'User' section includes 'List Course', 'My Course', 'My Profile', and 'Forum'. The 'Instructor' section includes 'Category', 'Create Course', 'User Enroll', and 'Send Email'. The main content area is titled 'Add Quiz' with a sub-header 'Access by: Instructor'. A breadcrumb trail at the top right reads 'My Course / Add Quiz'. The central part of the page is a 'Form add activity' with several input fields: 'Quiz Name *', 'Start date *', 'End date *', 'Limit time *', and 'Grade to pass *'. The 'Limit time *' field is split into a 'Limit' text input and a 'Minute' dropdown menu.

E-Learning

User

- List Course
- My Course
- My Profile
- Forum

Instructor

- Category
- Create Course
- User Enroll
- Send Email

Add Quiz
Access by: Instructor

My Course / Add Quiz

Form add activity

Quiz Name *

Start date *

End date *

Limit time *

Grade to pass *

Limit Minute

Memasukan soal kedalam bank soal, hal ini bertujuan agar antara user yang satu dengan user yang lain kecil kemungkinan nya mendapatkan soal yang sama.

The screenshot displays the E-Learning system interface. On the left is a dark sidebar with navigation options for Users and Instructors. The main content area shows the 'My Course' configuration for a user. A green 'Report' button is visible. The quiz configuration section, titled 'Quiz Pertemuan 1', specifies the following details:

- This quiz will be open on **14 October 2020 12:01 AM**
- This quiz will be close on **21 October 2020 11:59 PM**
- Grade to pass: **60**
- Limit: **60 Minutes**
- Total Question: **5**

At the bottom of the configuration section are two buttons: 'Attempts now' and 'Questions Bank'.

Untuk mengisi soal, bisa manual (satu per satu) atau import menggunakan ms.excel

The screenshot displays the E-Learning system interface. On the left is a dark sidebar with a menu for 'User' (List Course, My Course, My Profile, Forum) and 'Instructor' (Category, Create Course, User Enroll, Send Email). The main content area is titled 'Question' with the subtitle 'Access by: Instructor'. It includes two buttons: '+ Add Question' and '+ Import From Excel'. Below these is a 'Data Question' section with a 'Show 10 entries' dropdown and a search bar. A table header is visible with columns: No, Question, Option A, Option B, Option C, Option D, Option E, and Correct Answer. The table body contains the message 'No data available in table'. At the bottom, it shows 'Showing 0 to 0 of 0 entries' and 'Previous'/'Next' navigation buttons. The top right corner shows a user profile icon labeled 'Instruktur'.

E-Learning

User

- List Course
- My Course
- My Profile
- Forum

Instructor

- Category
- Create Course
- User Enroll
- Send Email

Question
Access by: Instructor

/ Course / Quiz / Question

+ Add Question + Import From Excel

Data Question

Show 10 entries Search:

No	Question	Option A	Option B	Option C	Option D	Option E	Correct Answer
No data available in table							

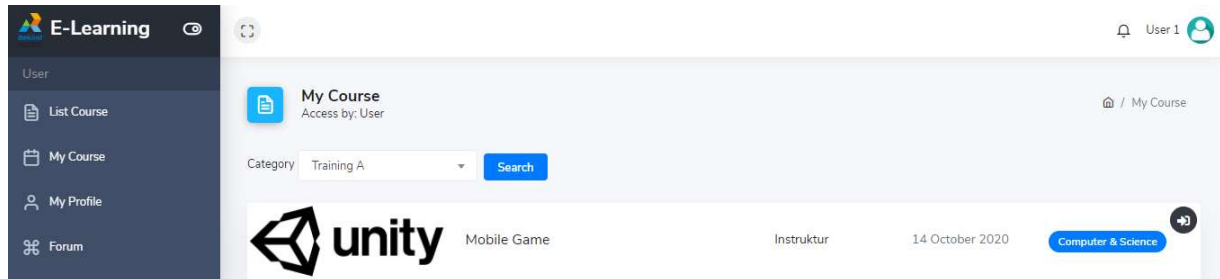
Showing 0 to 0 of 0 entries Previous Next

Menambahkan soal secara manual

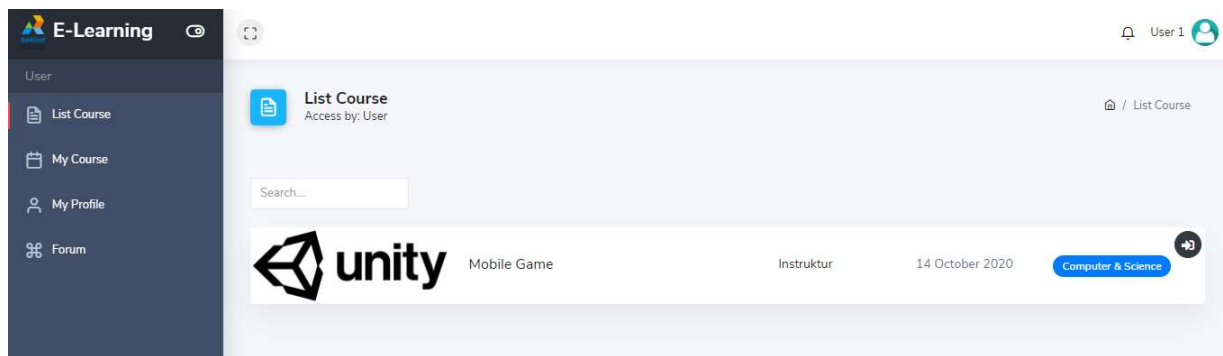
The screenshot shows the 'My Course' interface in an E-Learning system. On the left is a dark sidebar with navigation links for 'User' (List Course, My Course, My Profile, Forum) and 'Instructor' (Category, Create Course, User Enroll, Send Email). The main content area is titled 'My Course' with the subtitle 'Access by: Instructor'. A breadcrumb trail shows 'My Course / Quiz'. The 'Add Question' form contains two text input fields: 'Question *' and 'Option A *', both with red asterisks indicating they are required. The 'Question' field has a small icon in its bottom right corner, and the 'Option A' field has a similar icon.

Login user

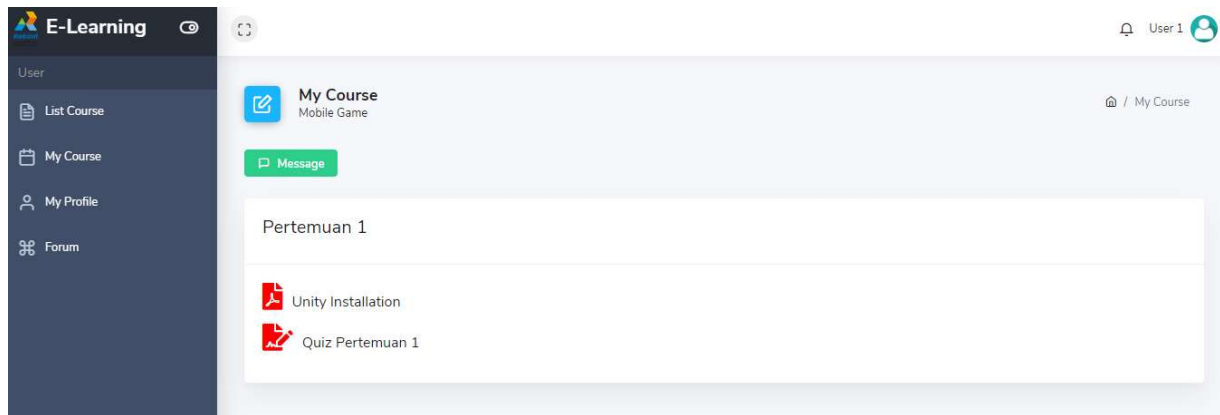
1 : 123



List course merupakan course yang di ikuti oleh seorang user.



Menu My Course berisi aktifitas belajar mengajar yang sedang diikuti, pada menu ini seorang user dapat mendownload materi, mengerjakan quiz dan dapat melakukan chat yang terhubung dengan whatsapp.



Menu quiz, dimana pada menu ini user dapat mengerjakan (menjawab) soal yang diberikan oleh instruktur.

LMSRekind

User 1

Selesai Ujian1

Ujian

Lembar Ujian

Home > Lembar Ujian

Question #1

Time 00:59:44

Tempat mendownload kebutuhan game dalam unity disebut..?

a

Asset Store

b

app store

c

Store Market

d

Marketplace

e

Supermarket

Next

Question Navigation

1. -2. -

Source Code :

index.php

```
1. <?php
2. /**
3.  * CodeIgniter
4.  *
5.  * An open source application development framework for PHP
6.  *
7.  * This content is released under the MIT License (MIT)
8.  *
9.  * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10. *
11. * Permission is hereby granted, free of charge, to any person obtaining a copy
12. * of this software and associated documentation files (the "Software"), to deal
13. * in the Software without restriction, including without limitation the rights
14. * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15. * copies of the Software, and to permit persons to whom the Software is
16. * furnished to do so, subject to the following conditions:
17. *
18. * The above copyright notice and this permission notice shall be included in
19. * all copies or substantial portions of the Software.
20. *
21. * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22. * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23. * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24. * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25. * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26. * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27. * THE SOFTWARE.
28. *
29. * @package CodeIgniter
30. * @author EllisLab Dev Team
31. * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32. * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33. * @license https://opensource.org/licenses/MIT MIT License
34. * @link https://codeigniter.com
35. * @since Version 1.0.0
36. * @filesource
37. */
38.
39. /*
40. *-----
41. * APPLICATION ENVIRONMENT
42. *-----
43. *
44. * You can load different configurations depending on your
45. * current environment. Setting the environment also influences
46. * things like logging and error reporting.
47. *
48. * This can be set to anything, but default usage is:
49. *
50. *     development
51. *     testing
52. *     production
53. *
54. * NOTE: If you change these, also change the error_reporting() code below
55. */
```

```

56.     define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ? $_SERVER['CI_ENV'] : 'development
57. ');
58. /*
59. *-----
60. * ERROR REPORTING
61. *-----
62. *
63. * Different environments will require different levels of error reporting.
64. * By default development will show errors but testing and live will hide them.
65. */
66. switch (ENVIRONMENT)
67. {
68.     case 'development':
69.         error_reporting(-1);
70.         ini_set('display_errors', 1);
71.         break;
72.
73.     case 'testing':
74.     case 'production':
75.         ini_set('display_errors', 0);
76.         if (version_compare(PHP_VERSION, '5.3', '>='))
77.         {
78.             error_reporting(E_ALL & ~E_NOTICE & ~E_DEPRECATED & ~E_STRICT & ~E_USER_NOTI
79. ICE & ~E_USER_DEPRECATED);
80.         }
81.         else
82.         {
83.             error_reporting(E_ALL & ~E_NOTICE & ~E_STRICT & ~E_USER_NOTICE);
84.         }
85.         break;
86.     default:
87.         header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
88.         echo 'The application environment is not set correctly.';
89.         exit(1); // EXIT_ERROR
90. }
91.
92. /*
93. *-----
94. * SYSTEM DIRECTORY NAME
95. *-----
96. *
97. * This variable must contain the name of your "system" directory.
98. * Set the path if it is not in the same directory as this file.
99. */
100.     $system_path = 'system';
101.
102.     /*
103.     *-----
104.     * APPLICATION DIRECTORY NAME
105.     *-----
106.     *
107.     * If you want this front controller to use a different "application"
108.     * directory than the default one you can set its name here. The directory
109.     * can also be renamed or relocated anywhere on your server. If you do,
110.     * use an absolute (full) server path.
111.     * For more info please see the user guide:
112.     *
113.     * https://codeigniter.com/user\_guide/general/managing\_apps.html
114.     *

```

```

115.      * NO TRAILING SLASH!
116.      */
117.      $application_folder = 'application';
118.
119.      /*
120.      * -----
121.      * VIEW DIRECTORY NAME
122.      * -----
123.      *
124.      * If you want to move the view directory out of the application
125.      * directory, set the path to it here. The directory can be renamed
126.      * and relocated anywhere on your server. If blank, it will default
127.      * to the standard location inside your application directory.
128.      * If you do move this, use an absolute (full) server path.
129.      *
130.      * NO TRAILING SLASH!
131.      */
132.      $view_folder = '';
133.
134.
135.      /*
136.      * -----
137.      * DEFAULT CONTROLLER
138.      * -----
139.      *
140.      * Normally you will set your default controller in the routes.php file.
141.      * You can, however, force a custom routing by hard-coding a
142.      * specific controller class/function here. For most applications, you
143.      * WILL NOT set your routing here, but it's an option for those
144.      * special instances where you might want to override the standard
145.      * routing in a specific front controller that shares a common CI installation.
146.      *
147.      * IMPORTANT: If you set the routing here, NO OTHER controller will be
148.      * callable. In essence, this preference limits your application to ONE
149.      * specific controller. Leave the function name blank if you need
150.      * to call functions dynamically via the URI.
151.      *
152.      * Un-comment the $routing array below to use this feature
153.      */
154.      // The directory name, relative to the "controllers" directory.  Leave blank
155.      // if your controller is not in a sub-
156.      // directory within the "controllers" one
157.      // $routing['directory'] = '';
158.
159.      // The controller class file name.  Example:  mycontroller
160.      // $routing['controller'] = '';
161.
162.      // The controller function you wish to be called.
163.      // $routing['function'] = '';
164.
165.      /*
166.      * -----
167.      * CUSTOM CONFIG VALUES
168.      * -----
169.      *
170.      * The $assign_to_config array below will be passed dynamically to the
171.      * config class when initialized. This allows you to set custom config
172.      * items or override any default config values found in the config.php file.
173.      * This can be handy as it permits you to share one application between

```

```

174.      * multiple front controller files, with each file containing different
175.      * config values.
176.      *
177.      * Un-comment the $assign_to_config array below to use this feature
178.      */
179.      // $assign_to_config['name_of_config_item'] = 'value of config item';
180.
181.
182.
183.      // -----
184.      // END OF USER CONFIGURABLE SETTINGS.  DO NOT EDIT BELOW THIS LINE
185.      // -----
186.
187.      /*
188.      * -----
189.      *   Resolve the system path for increased reliability
190.      * -----
191.      */
192.
193.      // Set the current directory correctly for CLI requests
194.      if (defined('STDIN'))
195.      {
196.          chdir(dirname(__FILE__));
197.      }
198.
199.      if (($temp = realpath($system_path)) !== FALSE)
200.      {
201.          $system_path = $temp.DIRECTORY_SEPARATOR;
202.      }
203.      else
204.      {
205.          // Ensure there's a trailing slash
206.          $system_path = strtr(
207.              rtrim($system_path, '\\'),
208.              '\\',
209.              DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
210.          ).DIRECTORY_SEPARATOR;
211.      }
212.
213.      // Is the system path correct?
214.      if ( ! is_dir($system_path))
215.      {
216.          header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
217.          echo 'Your system folder path does not appear to be set correctly. Pleas
e open the following file and correct this: '.pathinfo(__FILE__, PATHINFO_BASENAME);
218.          exit(3); // EXIT_CONFIG
219.      }
220.
221.      /*
222.      * -----
223.      *   Now that we know the path, set the main path constants
224.      * -----
225.      */
226.      // The name of THIS file
227.      define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
228.
229.      // Path to the system directory
230.      define('BASEPATH', $system_path);
231.
232.      // Path to the front controller (this file) directory
233.      define('FCPATH', dirname(__FILE__).DIRECTORY_SEPARATOR);

```

```

234.
235.     // Name of the "system" directory
236.     define('SYSDIR', basename(BASEPATH));
237.
238.     // The path to the "application" directory
239.     if (is_dir($application_folder))
240.     {
241.         if (($temp = realpath($application_folder)) !== FALSE)
242.         {
243.             $application_folder = $temp;
244.         }
245.         else
246.         {
247.             $application_folder = strtr(
248.                 rtrim($application_folder, '/\\'),
249.                 '/\\',
250.                 DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
251.             );
252.         }
253.     }
254.     elseif (is_dir(BASEPATH.$application_folder.DIRECTORY_SEPARATOR))
255.     {
256.         $application_folder = BASEPATH.strtr(
257.             trim($application_folder, '/\\'),
258.             '/\\',
259.             DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
260.         );
261.     }
262.     else
263.     {
264.         header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
265.         echo 'Your application folder path does not appear to be set correctly.  

Please open the following file and correct this: '.SELF;
266.         exit(3); // EXIT_CONFIG
267.     }
268.
269.     define('APPPATH', $application_folder.DIRECTORY_SEPARATOR);
270.
271.     // The path to the "views" directory
272.     if ( ! isset($view_folder[0]) && is_dir(APPPATH.'views'.DIRECTORY_SEPARATOR)
273.     )
274.     {
275.         $view_folder = APPPATH.'views';
276.     }
277.     elseif (is_dir($view_folder))
278.     {
279.         if (($temp = realpath($view_folder)) !== FALSE)
280.         {
281.             $view_folder = $temp;
282.         }
283.         else
284.         {
285.             $view_folder = strtr(
286.                 rtrim($view_folder, '/\\'),
287.                 '/\\',
288.                 DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
289.             );
290.         }
291.     }
292.     elseif (is_dir(APPPATH.$view_folder.DIRECTORY_SEPARATOR))

```

```

293.         $view_folder = APPPATH.strtr(
294.             trim($view_folder, '\\'),
295.             '\\',
296.             DIRECTORY_SEPARATOR.DIRECTORY_SEPARATOR
297.         );
298.     }
299.     else
300.     {
301.         header('HTTP/1.1 503 Service Unavailable.', TRUE, 503);
302.         echo 'Your view folder path does not appear to be set correctly. Please
open the following file and correct this: '.SELF;
303.         exit(3); // EXIT_CONFIG
304.     }
305.
306.     define('VIEWPATH', $view_folder.DIRECTORY_SEPARATOR);
307.
308.     /*
309.     * -----
310.     * LOAD THE BOOTSTRAP FILE
311.     * -----
312.     *
313.     * And away we go...
314.     */
315.     require_once BASEPATH.'core/CodeIgniter.php';

```

.htaccess

```

1. RewriteEngine On
2. RewriteCond %{REQUEST_FILENAME} !-f
3. RewriteCond %{REQUEST_FILENAME} !-d
4. RewriteRule ^(.*)$ index.php/$1 [L]

```

Application\index.html

```

1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.     <title>403 Forbidden</title>
6. </head>
7.
8. <body>
9.
10.    <p>Directory access is forbidden.</p>
11.
12. </body>
13.
14. </html>

```

Application\.htaccess

```

1. <IfModule authz_core_module>
2.     Require all denied
3. </IfModule>
4. <IfModule !authz_core_module>
5.     Deny from all

```

```
6. </IfModule>
```

application\cache\index.html

```
1. <!DOCTYPE html>
2. <html>
3.
4. <head>
5.     <title>403 Forbidden</title>
6. </head>
7.
8. <body>
9.
10.     <p>Directory access is forbidden.</p>
11.
12. </body>
13.
14. </html>
```

application\config\autoload.php

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | AUTO-LOADER
7. | -----
8. | This file specifies which systems should be loaded by default.
9. |
10. | In order to keep the framework as light-weight as possible only the
11. | absolute minimal resources are loaded by default. For example,
12. | the database is not connected to automatically since no assumption
13. | is made regarding whether you intend to use it. This file lets
14. | you globally define which systems you would like loaded with every
15. | request.
16. |
17. | -----
18. | Instructions
19. | -----
20. |
21. | These are the things you can load automatically:
22. |
23. | 1. Packages
24. | 2. Libraries
25. | 3. Drivers
26. | 4. Helper files
27. | 5. Custom config files
28. | 6. Language files
29. | 7. Models
30. |
31. */
32.
33. /*
34. | -----
35. | Auto-load Packages
36. | -----
37. | Prototype:
38. |
```

```

39. | $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
40. |
41. */
42. $autoload['packages'] = array();
43.
44. /*
45. | -----
46. | Auto-load Libraries
47. | -----
48. | These are the classes located in system/libraries/ or your
49. | application/libraries/ directory, with the addition of the
50. | 'database' library, which is somewhat of a special case.
51. |
52. | Prototype:
53. |
54. |     $autoload['libraries'] = array('database', 'email', 'session');
55. |
56. | You can also supply an alternative library name to be assigned
57. | in the controller:
58. |
59. |     $autoload['libraries'] = array('user_agent' => 'ua');
60. */
61. $autoload['libraries'] = array('database', 'email', 'session', 'upload', 'encryption');
62.
63. /*
64. | -----
65. | Auto-load Drivers
66. | -----
67. | These classes are located in system/libraries/ or in your
68. | application/libraries/ directory, but are also placed inside their
69. | own subdirectory and they extend the CI_Driver_Library class. They
70. | offer multiple interchangeable driver options.
71. |
72. | Prototype:
73. |
74. |     $autoload['drivers'] = array('cache');
75. |
76. | You can also supply an alternative property name to be assigned in
77. | the controller:
78. |
79. |     $autoload['drivers'] = array('cache' => 'cch');
80. |
81. */
82. $autoload['drivers'] = array();
83.
84. /*
85. | -----
86. | Auto-load Helper Files
87. | -----
88. | Prototype:
89. |
90. |     $autoload['helper'] = array('url', 'file');
91. */
92. $autoload['helper'] = array('url', 'learning', 'form', 'file');
93.
94. /*
95. | -----
96. | Auto-load Config files
97. | -----
98. | Prototype:

```

```

99. |
100. |     $autoload['config'] = array('config1', 'config2');
101. |
102. |     NOTE: This item is intended for use ONLY if you have created custom
103. |     config files. Otherwise, leave it blank.
104. |
105. |     */
106. |     $autoload['config'] = array();
107. |
108. |     /*
109. |     -----
110. |     Auto-load Language files
111. |     -----
112. |     Prototype:
113. |
114. |         $autoload['language'] = array('lang1', 'lang2');
115. |
116. |     NOTE: Do not include the "_lang" part of your file. For example
117. |     "codeigniter_lang.php" would be referenced as array('codeigniter');
118. |
119. |     */
120. |     $autoload['language'] = array();
121. |
122. |     /*
123. |     -----
124. |     Auto-load Models
125. |     -----
126. |     Prototype:
127. |
128. |         $autoload['model'] = array('first_model', 'second_model');
129. |
130. |     You can also supply an alternative model name to be assigned
131. |     in the controller:
132. |
133. |         $autoload['model'] = array('first_model' => 'first');
134. |     */
135. |     $autoload['model'] = array();

```

application\config\config.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Base Site URL
7. | -----
8. |
9. | URL to your CodeIgniter root. Typically this will be your base URL,
10. | WITH a trailing slash:
11. |
12. |     http://example.com/
13. |
14. | WARNING: You MUST set this value!
15. |
16. | If it is not set, then CodeIgniter will try guess the protocol and path
17. | your installation, but due to security concerns the hostname will be set
18. | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
19. | The auto-detection mechanism exists only for convenience during
20. | development and MUST NOT be used in production!

```

```

21. |
22. | If you need to allow multiple domains, remember that this file is still
23. | a PHP script and you can easily do that on your own.
24. |
25. */
26. $config['base_url'] = 'http://localhost/elearning/';
27.
28. /*
29. | -----
30. | Index File
31. | -----
32. |
33. | Typically this will be your index.php file, unless you've renamed it to
34. | something else. If you are using mod_rewrite to remove the page set this
35. | variable so that it is blank.
36. |
37. */
38. $config['index_page'] = '';
39.
40. /*
41. | -----
42. | URI PROTOCOL
43. | -----
44. |
45. | This item determines which server global should be used to retrieve the
46. | URI string. The default setting of 'REQUEST_URI' works for most servers.
47. | If your links do not seem to work, try one of the other delicious flavors:
48. |
49. | 'REQUEST_URI'    Uses $_SERVER['REQUEST_URI']
50. | 'QUERY_STRING'   Uses $_SERVER['QUERY_STRING']
51. | 'PATH_INFO'      Uses $_SERVER['PATH_INFO']
52. |
53. | WARNING: If you set this to 'PATH_INFO', URIs will always be URL-decoded!
54. */
55. $config['uri_protocol'] = 'REQUEST_URI';
56.
57. /*
58. | -----
59. | URL suffix
60. | -----
61. |
62. | This option allows you to add a suffix to all URLs generated by CodeIgniter.
63. | For more information please see the user guide:
64. |
65. | https://codeigniter.com/user\_guide/general/urls.html
66. */
67. $config['url_suffix'] = '';
68.
69. /*
70. | -----
71. | Default Language
72. | -----
73. |
74. | This determines which set of language files should be used. Make sure
75. | there is an available translation if you intend to use something other
76. | than english.
77. |
78. */
79. $config['language'] = 'english';
80.
81. /*

```

```

82. |-----
83. | Default Character Set
84. |-----
85. |
86. | This determines which character set is used by default in various methods
87. | that require a character set to be provided.
88. |
89. | See http://php.net/htmlspecialchars for a list of supported charsets.
90. |
91. |*/
92. $config['charset'] = 'UTF-8';
93.
94. /*
95. |-----
96. | Enable/Disable System Hooks
97. |-----
98. |
99. | If you would like to use the 'hooks' feature you must enable it by
100. | | setting this variable to TRUE (boolean). See the user guide for details.
101. | |
102. | */
103. $config['enable_hooks'] = FALSE;
104.
105. /*
106. |-----
107. | Class Extension Prefix
108. |-----
109. |
110. | This item allows you to set the filename/classname prefix when extending
111. | native libraries. For more information please see the user guide:
112. |
113. | https://codeigniter.com/user\_guide/general/core\_classes.html
114. | https://codeigniter.com/user\_guide/general/creating\_libraries.html
115. |
116. | */
117. $config['subclass_prefix'] = 'MY_';
118.
119. /*
120. |-----
121. | Composer auto-loading
122. |-----
123. |
124. | Enabling this setting will tell CodeIgniter to look for a Composer
125. | package auto-loader script in application/vendor/autoload.php.
126. |
127. | $config['composer_autoload'] = TRUE;
128. |
129. | Or if you have your vendor/ directory located somewhere else, you
130. | can opt to set a specific path as well:
131. |
132. | $config['composer_autoload'] = '/path/to/vendor/autoload.php';
133. |
134. | For more information about Composer, please visit http://getcomposer.org/
135. |
136. | Note: This will NOT disable or override the CodeIgniter-specific
137. | autoloading (application/config/autoload.php)
138. | */
139. $config['composer_autoload'] = FALSE;
140.
141. /*
142. |-----

```

```

143. | Allowed URL Characters
144. | -----
145. |
146. | This lets you specify which characters are permitted within your URLs.
147. | When someone tries to submit a URL with disallowed characters they will
148. | get a warning message.
149. |
150. | As a security measure you are STRONGLY encouraged to restrict URLs to
151. | as few characters as possible. By default only these are allowed: a-z 0-
    9~%.:_-
152. |
153. | Leave blank to allow all characters -- but only if you are insane.
154. |
155. | The configured value is actually a regular expression character group
156. | and it will be executed as: ! preg_match('/^[<permitted_uri_chars>]+$'/i
157. |
158. | DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
159. |
160. */
161. $config['permitted_uri_chars'] = 'a-z 0-9~%.:_-';
162.
163. /*
164. | -----
165. | Enable Query Strings
166. | -----
167. |
168. | By default CodeIgniter uses search-engine friendly segment based URLs:
169. | example.com/who/what/where/
170. |
171. | You can optionally enable standard query string based URLs:
172. | example.com?who=me&what=something&where=here
173. |
174. | Options are: TRUE or FALSE (boolean)
175. |
176. | The other items let you set the query string 'words' that will
177. | invoke your controllers and its functions:
178. | example.com/index.php?c=controller&m=function
179. |
180. | Please note that some of the helpers won't work as expected when
181. | this feature is enabled, since CodeIgniter is designed primarily to
182. | use segment based URLs.
183. |
184. */
185. $config['enable_query_strings'] = FALSE;
186. $config['controller_trigger'] = 'c';
187. $config['function_trigger'] = 'm';
188. $config['directory_trigger'] = 'd';
189.
190. /*
191. | -----
192. | Allow $_GET array
193. | -----
194. |
195. | By default CodeIgniter enables access to the $_GET array. If for some
196. | reason you would like to disable it, set 'allow_get_array' to FALSE.
197. |
198. | WARNING: This feature is DEPRECATED and currently available only
199. | for backwards compatibility purposes!
200. |
201. */
202. $config['allow_get_array'] = TRUE;

```

```

203.
204.      /*
205.      |-----
206.      | Error Logging Threshold
207.      |-----
208.
209.      | You can enable error logging by setting a threshold over zero. The
210.      | threshold determines what gets logged. Threshold options are:
211.
212.      |     0 = Disables logging, Error logging TURNED OFF
213.      |     1 = Error Messages (including PHP errors)
214.      |     2 = Debug Messages
215.      |     3 = Informational Messages
216.      |     4 = All Messages
217.
218.      | You can also pass an array with threshold levels to show individual error type
219.      |
220.      |     array(2) = Debug Messages, without Error Messages
221.
222.      | For a live site you'll usually only enable Errors (1) to be logged otherwise
223.      | your log files will fill up very fast.
224.
225.      */
226.      $config['log_threshold'] = 0;
227.
228.      /*
229.      |-----
230.      | Error Logging Directory Path
231.      |-----
232.
233.      | Leave this BLANK unless you would like to set something other than the default
234.      | application/logs/ directory. Use a full server path with trailing slash.
235.
236.      */
237.      $config['log_path'] = '';
238.
239.      /*
240.      |-----
241.      | Log File Extension
242.      |-----
243.
244.      | The default filename extension for log files. The default 'php' allows for
245.      | protecting the log files via basic scripting, when they are to be stored
246.      | under a publicly accessible directory.
247.
248.      | Note: Leaving it blank will default to 'php'.
249.
250.      */
251.      $config['log_file_extension'] = '';
252.
253.      /*
254.      |-----
255.      | Log File Permissions
256.      |-----
257.
258.      | The file system permissions to be applied on newly created log files.
259.
260.      | IMPORTANT: This MUST be an integer (no quotes) and you MUST use octal
261.      | integer notation (i.e. 0700, 0644, etc.)

```

```

262.    */
263.    $config['log_file_permissions'] = 0644;
264.
265.    /*
266.    |-----
267.    | Date Format for Logs
268.    |-----
269.    |
270.    | Each item that is logged has an associated date. You can use PHP date
271.    | codes to set your own date formatting
272.    |
273.    */
274.    $config['log_date_format'] = 'Y-m-d H:i:s';
275.
276.    /*
277.    |-----
278.    | Error Views Directory Path
279.    |-----
280.    |
281.    | Leave this BLANK unless you would like to set something other than the default
282.    | application/views/errors/ directory. Use a full server path with trailing slash.
283.    |
284.    */
285.    $config['error_views_path'] = '';
286.
287.    /*
288.    |-----
289.    | Cache Directory Path
290.    |-----
291.    |
292.    | Leave this BLANK unless you would like to set something other than the default
293.    | application/cache/ directory. Use a full server path with trailing slash.
294.    |
295.    */
296.    $config['cache_path'] = '';
297.
298.    /*
299.    |-----
300.    | Cache Include Query String
301.    |-----
302.    |
303.    | Whether to take the URL query string into consideration when generating
304.    | output cache files. Valid options are:
305.    |
306.    | FALSE      = Disabled
307.    | TRUE       = Enabled, take all query parameters into account.
308.    |              Please be aware that this may result in numerous cache
309.    |              files generated for the same page over and over again.
310.    | array('q') = Enabled, but only take into account the specified list
311.    |              of query parameters.
312.    |
313.    */
314.    $config['cache_query_string'] = FALSE;
315.
316.    /*
317.    |-----
318.    | Encryption Key
319.    |-----

```

```

320.
321. | If you use the Encryption class, you must set an encryption key.
322. | See the user guide for more info.
323.
324. | https://codeigniter.com/user\_guide/libraries/encryption.html
325.
326. */
327. $config['encryption_key'] = '42c1e1c7dd4e90308db75e7e430ba7d1';
328.
329. /*
330. | -----
331. | Session Variables
332. | -----
333.
334. | 'sess_driver'
335. |
336. | The storage driver to use: files, database, redis, memcached
337.
338. | 'sess_cookie_name'
339. |
340. | The session cookie name, must contain only [0-9a-z_-] characters
341.
342. | 'sess_expiration'
343. |
344. | The number of SECONDS you want the session to last.
345. | Setting to 0 (zero) means expire when the browser is closed.
346.
347. | 'sess_save_path'
348. |
349. | The location to save sessions to, driver dependent.
350.
351. | For the 'files' driver, it's a path to a writable directory.
352. | WARNING: Only absolute paths are supported!
353.
354. | For the 'database' driver, it's a table name.
355. | Please read up the manual for the format with other session drivers.
356.
357. | IMPORTANT: You are REQUIRED to set a valid save path!
358.
359. | 'sess_match_ip'
360. |
361. | Whether to match the user's IP address when reading the session data.
362.
363. | WARNING: If you're using the database driver, don't forget to update
364. | your session table's PRIMARY KEY when changing this setting.
365.
366. | 'sess_time_to_update'
367. |
368. | How many seconds between CI regenerating the session ID.
369.
370. | 'sess_regenerate_destroy'
371. |
372. | Whether to destroy session data associated with the old session ID
373. | when auto-regenerating the session ID. When set to FALSE, the data
374. | will be later deleted by the garbage collector.
375.
376. | Other session cookie settings are shared with the rest of the application,
377. | except for 'cookie_prefix' and 'cookie_httponly', which are ignored here.
378.
379. */
380. $config['sess_driver'] = 'files';

```

```

381.     $config['sess_cookie_name'] = 'ci_session';
382.     $config['sess_expiration'] = 7200;
383.     $config['sess_save_path'] = NULL;
384.     $config['sess_match_ip'] = FALSE;
385.     $config['sess_time_to_update'] = 300;
386.     $config['sess_regenerate_destroy'] = FALSE;
387.
388.     /*
389.     |-----
390.     | Cookie Related Variables
391.     |-----
392.     |
393.     | 'cookie_prefix'   = Set a cookie name prefix if you need to avoid collisions
394.     | 'cookie_domain'   = Set to .your-domain.com for site-wide cookies
395.     | 'cookie_path'     = Typically will be a forward slash
396.     | 'cookie_secure'   = Cookie will only be set if a secure HTTPS connection exist
397.     | 'cookie_httponly' = Cookie will only be accessible via HTTP(S) (no javascript)
398.     |
399.     | Note: These settings (with the exception of 'cookie_prefix' and
400.     |       'cookie_httponly') will also affect sessions.
401.     |
402.     */
403.     $config['cookie_prefix']   = '';
404.     $config['cookie_domain']   = '';
405.     $config['cookie_path']     = '/';
406.     $config['cookie_secure']   = FALSE;
407.     $config['cookie_httponly'] = FALSE;
408.
409.     /*
410.     |-----
411.     | Standardize newlines
412.     |-----
413.     |
414.     | Determines whether to standardize newline characters in input data,
415.     | meaning to replace \r\n, \r, \n occurrences with the PHP_EOL value.
416.     |
417.     | WARNING: This feature is DEPRECATED and currently available only
418.     |          for backwards compatibility purposes!
419.     |
420.     */
421.     $config['standardize_newlines'] = FALSE;
422.
423.     /*
424.     |-----
425.     | Global XSS Filtering
426.     |-----
427.     |
428.     | Determines whether the XSS filter is always active when GET, POST or
429.     | COOKIE data is encountered
430.     |
431.     | WARNING: This feature is DEPRECATED and currently available only
432.     |          for backwards compatibility purposes!
433.     |
434.     */
435.     $config['global_xss_filtering'] = FALSE;
436.
437.     /*
438.     |-----
439.     | Cross Site Request Forgery

```

```

440. | -----
441. | Enables a CSRF cookie token to be set. When set to TRUE, token will be
442. | checked on a submitted form. If you are accepting user data, it is strongly
443. | recommended CSRF protection be enabled.
444. |
445. | 'csrf_token_name' = The token name
446. | 'csrf_cookie_name' = The cookie name
447. | 'csrf_expire' = The number in seconds the token should expire.
448. | 'csrf_regenerate' = Regenerate token on every submission
449. | 'csrf_exclude_uris' = Array of URIs which ignore CSRF checks
450. */
451. $config['csrf_protection'] = FALSE;
452. $config['csrf_token_name'] = 'csrf_test_name';
453. $config['csrf_cookie_name'] = 'csrf_cookie_name';
454. $config['csrf_expire'] = 7200;
455. $config['csrf_regenerate'] = TRUE;
456. $config['csrf_exclude_uris'] = array();
457.
458. /*
459. | -----
460. | Output Compression
461. | -----
462. |
463. | Enables Gzip output compression for faster page loads. When enabled,
464. | the output class will test whether your server supports Gzip.
465. | Even if it does, however, not all browsers support compression
466. | so enable only if you are reasonably sure your visitors can handle it.
467. |
468. | Only used if zlib.output_compression is turned off in your php.ini.
469. | Please do not use it together with httpd-level output compression.
470. |
471. | VERY IMPORTANT: If you are getting a blank page when compression is enabled i
472. | t
473. | means you are prematurely outputting something to your browser. It could
474. | even be a line of whitespace at the end of one of your scripts. For
475. | compression to work, nothing can be sent before the output buffer is called
476. | by the output class. Do not 'echo' any values with compression enabled.
477. */
478. $config['compress_output'] = FALSE;
479.
480. /*
481. | -----
482. | Master Time Reference
483. | -----
484. |
485. | Options are 'local' or any PHP supported timezone. This preference tells
486. | the system whether to use your server's local time as the master 'now'
487. | reference, or convert it to the configured one timezone. See the 'date
488. | helper' page of the user guide for information regarding date handling.
489. |
490. */
491. $config['time_reference'] = 'local';
492.
493. /*
494. | -----
495. | Rewrite PHP Short Tags
496. | -----
497. |
498. | If your PHP installation does not have short tag support enabled CI
499. | can rewrite the tags on-the-fly, enabling you to utilize that syntax

```

```

500. | in your view files. Options are TRUE or FALSE (boolean)
501. |
502. | Note: You need to have eval() enabled for this to work.
503. |
504. */
505. $config['rewrite_short_tags'] = FALSE;
506.
507. /*
508. |-----
509. | Reverse Proxy IPs
510. |-----
511. |
512. | If your server is behind a reverse proxy, you must whitelist the proxy
513. | IP addresses from which CodeIgniter should trust headers such as
514. | HTTP_X_FORWARDED_FOR and HTTP_CLIENT_IP in order to properly identify
515. | the visitor's IP address.
516. |
517. | You can use both an array or a comma-separated list of proxy addresses,
518. | as well as specifying whole subnets. Here are a few examples:
519. |
520. | Comma-separated: '10.0.1.200,192.168.5.0/24'
521. | Array:          array('10.0.1.200', '192.168.5.0/24')
522. */
523. $config['proxy_ips'] = '';
524.
525. $config['default_timezone'] = date_default_timezone_set('Asia/Jakarta');

```

application\config\constants.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. |-----
6. | Display Debug traceback
7. |-----
8. |
9. | If set to TRUE, a traceback will be displayed along with php errors. If
10. | error_reporting is disabled, the traceback will not display, regardless
11. | of this setting
12. |
13. */
14. defined('SHOW_DEBUG_BACKTRACE') OR define('SHOW_DEBUG_BACKTRACE', TRUE);
15.
16. /*
17. |-----
18. | File and Directory Modes
19. |-----
20. |
21. | These prefs are used when checking and setting modes when working
22. | with the file system. The defaults are fine on servers with proper
23. | security, but you may wish (or even need) to change the values in
24. | certain environments (Apache running a separate process for each
25. | user, PHP under CGI with Apache suEXEC, etc.). Octal values should
26. | always be used to set the mode correctly.
27. |
28. */
29. defined('FILE_READ_MODE') OR define('FILE_READ_MODE', 0644);
30. defined('FILE_WRITE_MODE') OR define('FILE_WRITE_MODE', 0666);
31. defined('DIR_READ_MODE') OR define('DIR_READ_MODE', 0755);

```

```

32. defined('DIR_WRITE_MODE') OR define('DIR_WRITE_MODE', 0755);
33.
34. /*
35. |-----
36. | File Stream Modes
37. |-----
38. |
39. | These modes are used when working with fopen()/popen()
40. |
41. */
42. defined('FOPEN_READ') OR define('FOPEN_READ', 'rb');
43. defined('FOPEN_READ_WRITE') OR define('FOPEN_READ_WRITE', 'r+b');
44. defined('FOPEN_WRITE_CREATE_DESTRUCTIVE') OR define('FOPEN_WRITE_CREATE_DESTRUCTIVE', 'wb'); // truncates existing file data, use with care
45. defined('FOPEN_READ_WRITE_CREATE_DESTRUCTIVE') OR define('FOPEN_READ_WRITE_CREATE_DESTRUCTIVE', 'w+b'); // truncates existing file data, use with care
46. defined('FOPEN_WRITE_CREATE') OR define('FOPEN_WRITE_CREATE', 'ab');
47. defined('FOPEN_READ_WRITE_CREATE') OR define('FOPEN_READ_WRITE_CREATE', 'a+b');
48. defined('FOPEN_WRITE_CREATE_STRICT') OR define('FOPEN_WRITE_CREATE_STRICT', 'xb');
49. defined('FOPEN_READ_WRITE_CREATE_STRICT') OR define('FOPEN_READ_WRITE_CREATE_STRICT', 'x+b');
50.
51. /*
52. |-----
53. | Exit Status Codes
54. |-----
55. |
56. | Used to indicate the conditions under which the script is exit()ing.
57. | While there is no universal standard for error codes, there are some
58. | broad conventions. Three such conventions are mentioned below, for
59. | those who wish to make use of them. The CodeIgniter defaults were
60. | chosen for the least overlap with these conventions, while still
61. | leaving room for others to be defined in future versions and user
62. | applications.
63. |
64. | The three main conventions used for determining exit status codes
65. | are as follows:
66. |
67. | Standard C/C++ Library (stdlibc):
68. | http://www.gnu.org/software/libc/manual/html\_node/Exit-Status.html
69. | (This link also contains other GNU-specific conventions)
70. | BSD sysexits.h:
71. | http://www.gsp.com/cgi-bin/man.cgi?section=3&topic=sysexits
72. | Bash scripting:
73. | http://tldp.org/LDP/abs/html/exitcodes.html
74. |
75. */
76. defined('EXIT_SUCCESS') OR define('EXIT_SUCCESS', 0); // no errors
77. defined('EXIT_ERROR') OR define('EXIT_ERROR', 1); // generic error
78. defined('EXIT_CONFIG') OR define('EXIT_CONFIG', 3); // configuration error
79. defined('EXIT_UNKNOWN_FILE') OR define('EXIT_UNKNOWN_FILE', 4); // file not found
80. defined('EXIT_UNKNOWN_CLASS') OR define('EXIT_UNKNOWN_CLASS', 5); // unknown class
81. defined('EXIT_UNKNOWN_METHOD') OR define('EXIT_UNKNOWN_METHOD', 6); // unknown class member
82. defined('EXIT_USER_INPUT') OR define('EXIT_USER_INPUT', 7); // invalid user input
83. defined('EXIT_DATABASE') OR define('EXIT_DATABASE', 8); // database error
84. defined('EXIT__AUTO_MIN') OR define('EXIT__AUTO_MIN', 9); // lowest automatically-assigned error code

```

```
85. defined('EXIT__AUTO_MAX')          OR define('EXIT__AUTO_MAX', 125); // highest automatical
    ly-assigned error code
```

application\config\database.php

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | DATABASE CONNECTIVITY SETTINGS
7. | -----
8. | This file will contain the settings needed to access your database.
9. |
10. | For complete instructions please consult the 'Database Connection'
11. | page of the User Guide.
12. |
13. | -----
14. | EXPLANATION OF VARIABLES
15. | -----
16. |
17. | ['dsn']          The full DSN string describe a connection to the database.
18. | ['hostname']     The hostname of your database server.
19. | ['username']     The username used to connect to the database
20. | ['password']     The password used to connect to the database
21. | ['database']     The name of the database you want to connect to
22. | ['dbdriver']     The database driver. e.g.: mysqli.
23. |         Currently supported:
24. |             cubrid, ibase, mssql, mysql, mysqli, oci8,
25. |             odbc, pdo, postgre, sqlite, sqlite3, sqlsrv
26. | ['dbprefix']     You can add an optional prefix, which will be added
27. |                 to the table name when using the Query Builder class
28. | ['pconnect']     TRUE/FALSE - Whether to use a persistent connection
29. | ['db_debug']     TRUE/FALSE - Whether database errors should be displayed.
30. | ['cache_on']     TRUE/FALSE - Enables/disables query caching
31. | ['cachedir']     The path to the folder where cache files should be stored
32. | ['char_set']     The character set used in communicating with the database
33. | ['dbcollat']     The character collation used in communicating with the database
34. |                 NOTE: For MySQL and MySQLi databases, this setting is only used
35. |                 as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
36. |                 (and in table creation queries made with DB Forge).
37. |                 There is an incompatibility in PHP with mysql_real_escape_string() whi
38. | ch
39. |                 can make your site vulnerable to SQL injection if you are using a
40. |                 multi-byte character set and are running versions lower than these.
41. |                 Sites using Latin-1 or UTF-
42. | 8 database character set and collation are unaffected.
43. | ['swap_pre']     A default table prefix that should be swapped with the dbprefix
44. | ['encrypt']      Whether or not to use an encrypted connection.
45. |
46. |                 'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv' drivers accept TRUE/FALSE
47. |                 'mysqli' and 'pdo/mysqli' drivers accept an array with the following options
48. |                 :
49. |                 'ssl_key'    - Path to the private key file
50. |                 'ssl_cert'   - Path to the public key certificate file
51. |                 'ssl_ca'     - Path to the certificate authority file
52. |                 'ssl_capath' - Path to a directory containing trusted CA certificates i
53. | n PEM format
```

```

51. |         'ssl_cipher' - List of *allowed* ciphers to be used for the encryption,
    |         separated by colons (':')
52. |         'ssl_verify' - TRUE/FALSE; Whether verify the server certificate or not
53. |
54. |         ['compress'] Whether or not to use client compression (MySQL only)
55. |         ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
56. |         - good for ensuring strict SQL while developing
57. |         ['ssl_options'] Used to set various SSL options that can be used when making SSL co
    |         nnections.
58. |         ['failover'] array - A array with 0 or more data for connections if the main should
    |         fail.
59. |         ['save_queries'] TRUE/FALSE - Whether to "save" all executed queries.
60. |         NOTE: Disabling this will also effectively disable both
61. |         $this->db->last_query() and profiling of DB queries.
62. |         When you run a query, with this setting set to TRUE (default),
63. |         CodeIgniter will store the SQL statement for debugging purposes.
64. |         However, this may cause high memory usage, especially if you run
65. |         a lot of SQL queries ... disable this to avoid that problem.
66. |
67. | The $active_group variable lets you choose which connection group to
68. | make active. By default there is only one group (the 'default' group).
69. |
70. | The $query_builder variables lets you determine whether or not to load
71. | the query builder class.
72. */
73. $active_group = 'default';
74. $query_builder = TRUE;
75.
76. $db['default'] = array(
77.     'dsn' => '',
78.     'hostname' => 'localhost',
79.     'username' => 'root',
80.     'password' => '',
81.     'database' => 'elearning',
82.     'dbdriver' => 'mysqli',
83.     'dbprefix' => '',
84.     'pconnect' => FALSE,
85.     'db_debug' => (ENVIRONMENT !== 'production'),
86.     'cache_on' => FALSE,
87.     'cachedir' => '',
88.     'char_set' => 'utf8',
89.     'dbcollat' => 'utf8_general_ci',
90.     'swap_pre' => '',
91.     'encrypt' => FALSE,
92.     'compress' => FALSE,
93.     'stricton' => FALSE,
94.     'failover' => array(),
95.     'save_queries' => TRUE
96. );

```

application\config\doctype.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. $_doctypes = array(
5.     'xhtml11' => '<!DOCTYPE html PUBLIC "-//
    | //W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">',

```

```

6.     'xhtml1-strict' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">',
7.     'xhtml1-trans' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">',
8.     'xhtml1-frame' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
frameset.dtd">',
9.     'xhtml-basic11' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML Basic 1.1//EN" "http://www.w3.org/TR/xhtml-basic/xhtml1-
basic11.dtd">',
10.    'html5' => '<!DOCTYPE html>',
11.    'html4-strict' => '<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">',
12.    'html4-trans' => '<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">',
13.    'html4-frame' => '<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">',
14.    'mathml1' => '<!DOCTYPE math SYSTEM "http://www.w3.org/Math/DTD/mathml1/mathml.dtd"
>',
15.    'mathml2' => '<!DOCTYPE math PUBLIC "-
//W3C//DTD MathML 2.0//EN" "http://www.w3.org/Math/DTD/mathml2/mathml2.dtd">',
16.    'svg10' => '<!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">',
17.    'svg11' => '<!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">',
18.    'svg11-basic' => '<!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.1 Basic//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-
basic.dtd">',
19.    'svg11-tiny' => '<!DOCTYPE svg PUBLIC "-
//W3C//DTD SVG 1.1 Tiny//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11-
tiny.dtd">',
20.    'xhtml-math-svg-xh' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN" "http://www.w3.org/2002/04/xhtml1
-math-svg/xhtml1-math-svg.dtd">',
21.    'xhtml-math-svg-sh' => '<!DOCTYPE svg:svg PUBLIC "-
//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN" "http://www.w3.org/2002/04/xhtml1
-math-svg/xhtml1-math-svg.dtd">',
22.    'xhtml-rdfa-1' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml1-rdfa-1.dtd">',
23.    'xhtml-rdfa-2' => '<!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML+RDFa 1.1//EN" "http://www.w3.org/MarkUp/DTD/xhtml1-rdfa-2.dtd">'
24. );

```

application\config\foreign_chars.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Foreign Characters
7. | -----
8. | This file contains an array of foreign characters for transliteration
9. | conversion used by the Text helper
10. |
11. */
12. $foreign_characters = array(
13.     '/ä|æ|é/' => 'ae',

```

[illegible]

```

75.     '/z|ž|ž|ž|з/' => 'z',
76.     '/Æ|Ǽ/' => 'AE',
77.     '/ß/' => 'ss',
78.     '/IJ/' => 'IJ',
79.     '/ij/' => 'ij',
80.     '/Œ/' => 'OE',
81.     '/f/' => 'f',
82.     '/Ξ/' => 'KS',
83.     '/ξ/' => 'ks',
84.     '/Π/' => 'P',
85.     '/π/' => 'p',
86.     '/Β/' => 'V',
87.     '/β/' => 'v',
88.     '/Μ/' => 'M',
89.     '/μ/' => 'm',
90.     '/Ψ/' => 'PS',
91.     '/ψ/' => 'ps',
92.     '/Ë/' => 'Yo',
93.     '/ë/' => 'yo',
94.     '/Ε/' => 'Ye',
95.     '/ε/' => 'ye',
96.     '/İ/' => 'Yi',
97.     '/Ж/' => 'Zh',
98.     '/ж/' => 'zh',
99.     '/X/' => 'Kh',
100.     '/x/' => 'kh',
101.     '/Ц/' => 'Ts',
102.     '/ц/' => 'ts',
103.     '/Ч/' => 'Ch',
104.     '/ч/' => 'ch',
105.     '/Ш/' => 'Sh',
106.     '/ш/' => 'sh',
107.     '/Щ/' => 'Shch',
108.     '/щ/' => 'shch',
109.     '/Ъ|ь|б|ъ/' => '',
110.     '/Ю/' => 'Yu',
111.     '/ю/' => 'yu',
112.     '/Я/' => 'Ya',
113.     '/я/' => 'ya'
114. );

```

application\config\hooks.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Hooks
7. | -----
8. | This file lets you define "hooks" to extend CI without hacking the core
9. | files. Please see the user guide for info:
10. |
11. | https://codeigniter.com/user\_guide/general/hooks.html
12. |
13. */

```

application\config\index.html

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>403 Forbidden</title>
5. </head>
6. <body>
7.
8. <p>Directory access is forbidden.</p>
9.
10. </body>
11. </html>
```

application\config\memcached.php

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Memcached settings
7. | -----
8. | Your Memcached servers can be specified below.
9. |
10. | See: https://codeigniter.com/user\_guide/libraries/caching.html#memcached
11. |
12. */
13. $config = array(
14.     'default' => array(
15.         'hostname' => '127.0.0.1',
16.         'port' => '11211',
17.         'weight' => '1',
18.     ),
19. );
```

application\config\migration.php

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Enable/Disable Migrations
7. | -----
8. |
9. | Migrations are disabled by default for security reasons.
10. | You should enable migrations whenever you intend to do a schema migration
11. | and disable it back when you're done.
12. |
13. */
14. $config['migration_enabled'] = FALSE;
15.
16. /*
17. | -----
18. | Migration Type
19. | -----
20. |
```

```

21. | Migration file names may be based on a sequential identifier or on
22. | a timestamp. Options are:
23. |
24. | 'sequential' = Sequential migration naming (001_add_blog.php)
25. | 'timestamp'  = Timestamp migration naming (20121031104401_add_blog.php)
26. |               Use timestamp format YYYYMMDDHHIISS.
27. |
28. | Note: If this configuration value is missing the Migration library
29. |       defaults to 'sequential' for backward compatibility with CI2.
30. |
31. */
32. $config['migration_type'] = 'timestamp';
33.
34. /*
35. |-----
36. | Migrations table
37. |-----
38. |
39. | This is the name of the table that will store the current migrations state.
40. | When migrations runs it will store in a database table which migration
41. | level the system is at. It then compares the migration level in this
42. | table to the $config['migration_version'] if they are not the same it
43. | will migrate up. This must be set.
44. |
45. */
46. $config['migration_table'] = 'migrations';
47.
48. /*
49. |-----
50. | Auto Migrate To Latest
51. |-----
52. |
53. | If this is set to TRUE when you load the migrations class and have
54. | $config['migration_enabled'] set to TRUE the system will auto migrate
55. | to your latest migration (whatever $config['migration_version'] is
56. | set to). This way you do not have to call migrations anywhere else
57. | in your code to have the latest migration.
58. |
59. */
60. $config['migration_auto_latest'] = FALSE;
61.
62. /*
63. |-----
64. | Migrations version
65. |-----
66. |
67. | This is used to set migration version that the file system should be on.
68. | If you run $this->migration->current() this is the version that schema will
69. | be upgraded / downgraded to.
70. |
71. */
72. $config['migration_version'] = 0;
73.
74. /*
75. |-----
76. | Migrations Path
77. |-----
78. |
79. | Path to your migrations folder.
80. | Typically, it will be within your application path.
81. | Also, writing permission is required within the migrations path.

```

```

82. |
83. */
84. $config['migration_path'] = APPPATH.'migrations/';

```

application\config\mimes.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | MIME TYPES
7. | -----
8. | This file contains an array of mime types. It is used by the
9. | Upload class to help identify allowed file types.
10. |
11. */
12. return array(
13.     'hqx' => array('application/mac-binhex40', 'application/mac-
binhex', 'application/x-binhex40', 'application/x-mac-binhex40'),
14.     'cpt' => 'application/mac-compactpro',
15.     'csv' => array('text/x-comma-separated-values', 'text/comma-separated-
values', 'application/octet-stream', 'application/vnd.ms-excel', 'application/x-
csv', 'text/x-
csv', 'text/csv', 'application/csv', 'application/excel', 'application/vnd.ms-excel', 't
ext/plain'),
16.     'bin' => array('application/macbinary', 'application/mac-
binary', 'application/octet-stream', 'application/x-binary', 'application/x-
macbinary'),
17.     'dms' => 'application/octet-stream',
18.     'lha' => 'application/octet-stream',
19.     'lzh' => 'application/octet-stream',
20.     'exe' => array('application/octet-stream', 'application/x-msdownload'),
21.     'class' => 'application/octet-stream',
22.     'psd' => array('application/x-photoshop', 'image/vnd.adobe.photoshop'),
23.     'so' => 'application/octet-stream',
24.     'sea' => 'application/octet-stream',
25.     'dll' => 'application/octet-stream',
26.     'oda' => 'application/oda',
27.     'pdf' => array('application/pdf', 'application/force-download', 'application/x-
download', 'binary/octet-stream'),
28.     'ai' => array('application/pdf', 'application/postscript'),
29.     'eps' => 'application/postscript',
30.     'ps' => 'application/postscript',
31.     'smi' => 'application/smil',
32.     'smil' => 'application/smil',
33.     'mif' => 'application/vnd.mif',
34.     'xls' => array('application/vnd.ms-
excel', 'application/msexcel', 'application/x-msexcel', 'application/x-ms-
excel', 'application/x-excel', 'application/x-
dos_ms_excel', 'application/xls', 'application/x-
xls', 'application/excel', 'application/download', 'application/vnd.ms-
office', 'application/msword'),
35.     'ppt' => array('application/powerpoint', 'application/vnd.ms-
powerpoint', 'application/vnd.ms-office', 'application/msword'),
36.     'pptx' => array('application/vnd.openxmlformats-
officedocument.presentationml.presentation', 'application/x-zip', 'application/zip'),
37.     'wbxml' => 'application/wbxml',
38.     'wmlc' => 'application/wmlc',
39.     'dcr' => 'application/x-director',

```

```
40. 'dir' => 'application/x-director',
41. 'dvr' => 'application/x-director',
42. 'dvi' => 'application/x-dvi',
43. 'gtar' => 'application/x-gtar',
44. 'gz' => 'application/x-gzip',
45. 'gzip' => 'application/x-gzip',
46. 'php' => array('application/x-httpd-php', 'application/php', 'application/x-
php', 'text/php', 'text/x-php', 'application/x-httpd-php-source'),
47. 'php4' => 'application/x-httpd-php',
48. 'php3' => 'application/x-httpd-php',
49. 'phtml' => 'application/x-httpd-php',
50. 'phps' => 'application/x-httpd-php-source',
51. 'js' => array('application/x-javascript', 'text/plain'),
52. 'swf' => 'application/x-shockwave-flash',
53. 'sit' => 'application/x-stuffit',
54. 'tar' => 'application/x-tar',
55. 'tgz' => array('application/x-tar', 'application/x-gzip-compressed'),
56. 'z' => 'application/x-compress',
57. 'xhtml' => 'application/xhtml+xml',
58. 'xht' => 'application/xhtml+xml',
59. 'zip' => array('application/x-zip', 'application/zip', 'application/x-zip-
compressed', 'application/s-compressed', 'multipart/x-zip'),
60. 'rar' => array('application/x-rar', 'application/rar', 'application/x-rar-
compressed'),
61. 'mid' => 'audio/midi',
62. 'midi' => 'audio/midi',
63. 'mpga' => 'audio/mpeg',
64. 'mp2' => 'audio/mpeg',
65. 'mp3' => array('audio/mpeg', 'audio/mpg', 'audio/mpeg3', 'audio/mp3'),
66. 'aif' => array('audio/x-aiff', 'audio/aiff'),
67. 'aiff' => array('audio/x-aiff', 'audio/aiff'),
68. 'aifc' => 'audio/x-aiff',
69. 'ram' => 'audio/x-pn-realaudio',
70. 'rm' => 'audio/x-pn-realaudio',
71. 'rpm' => 'audio/x-pn-realaudio-plugin',
72. 'ra' => 'audio/x-realaudio',
73. 'rv' => 'video/vnd.rn-realvideo',
74. 'wav' => array('audio/x-wav', 'audio/wave', 'audio/wav'),
75. 'bmp' => array('image/bmp', 'image/x-bmp', 'image/x-bitmap', 'image/x-
xbitmap', 'image/x-win-bitmap', 'image/x-windows-bmp', 'image/ms-bmp', 'image/x-ms-
bmp', 'application/bmp', 'application/x-bmp', 'application/x-win-bitmap'),
76. 'gif' => 'image/gif',
77. 'jpeg' => array('image/jpeg', 'image/pjpeg'),
78. 'jpg' => array('image/jpeg', 'image/pjpeg'),
79. 'jpe' => array('image/jpeg', 'image/pjpeg'),
80. 'jp2' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
81. 'j2k' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
82. 'jpf' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
83. 'jpg2' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
84. 'jpx' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
85. 'jpm' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
86. 'mj2' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
87. 'mjp2' => array('image/jp2', 'video/mj2', 'image/jpx', 'image/jpm'),
88. 'png' => array('image/png', 'image/x-png'),
89. 'tiff' => 'image/tiff',
90. 'tif' => 'image/tiff',
91. 'css' => array('text/css', 'text/plain'),
92. 'html' => array('text/html', 'text/plain'),
93. 'htm' => array('text/html', 'text/plain'),
94. 'shtml' => array('text/html', 'text/plain'),
95. 'txt' => 'text/plain',
```

```

96.     'text' => 'text/plain',
97.     'log' => array('text/plain', 'text/x-log'),
98.     'rtx' => 'text/richtext',
99.     'rtf' => 'text/rtf',
100.    'xml' => array('application/xml', 'text/xml', 'text/plain'),
101.    'xsl' => array('application/xml', 'text/xsl', 'text/xml'),
102.    'mpeg' => 'video/mpeg',
103.    'mpg' => 'video/mpeg',
104.    'mpe' => 'video/mpeg',
105.    'qt' => 'video/quicktime',
106.    'mov' => 'video/quicktime',
107.    'avi' => array('video/x-
msvideo', 'video/msvideo', 'video/avi', 'application/x-troff-msvideo'),
108.    'movie' => 'video/x-sgi-movie',
109.    'doc' => array('application/msword', 'application/vnd.ms-office'),
110.    'docx' => array('application/vnd.openxmlformats-
officedocument.wordprocessingml.document', 'application/zip', 'application/msword', 'ap
plication/x-zip'),
111.    'dot' => array('application/msword', 'application/vnd.ms-office'),
112.    'dotx' => array('application/vnd.openxmlformats-
officedocument.wordprocessingml.document', 'application/zip', 'application/msword'),
113.    'xlsx' => array('application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet', 'application/zip', 'application/vnd.ms-
excel', 'application/msword', 'application/x-zip'),
114.    'word' => array('application/msword', 'application/octet-stream'),
115.    'xl' => 'application/excel',
116.    'eml' => 'message/rfc822',
117.    'json' => array('application/json', 'text/json'),
118.    'pem' => array('application/x-x509-user-cert', 'application/x-pem-
file', 'application/octet-stream'),
119.    'p10' => array('application/x-pkcs10', 'application/pkcs10'),
120.    'p12' => 'application/x-pkcs12',
121.    'p7a' => 'application/x-pkcs7-signature',
122.    'p7c' => array('application/pkcs7-mime', 'application/x-pkcs7-mime'),
123.    'p7m' => array('application/pkcs7-mime', 'application/x-pkcs7-mime'),
124.    'p7r' => 'application/x-pkcs7-certreqresp',
125.    'p7s' => 'application/pkcs7-signature',
126.    'crt' => array('application/x-x509-ca-cert', 'application/x-x509-user-
cert', 'application/pkix-cert'),
127.    'crl' => array('application/pkix-crl', 'application/pkcs-crl'),
128.    'der' => 'application/x-x509-ca-cert',
129.    'kdb' => 'application/octet-stream',
130.    'pgp' => 'application/pgp',
131.    'gpg' => 'application/gpg-keys',
132.    'sst' => 'application/octet-stream',
133.    'csr' => 'application/octet-stream',
134.    'rsa' => 'application/x-pkcs7',
135.    'cer' => array('application/pkix-cert', 'application/x-x509-ca-cert'),
136.    '3g2' => 'video/3gpp2',
137.    '3gp' => array('video/3gp', 'video/3gpp'),
138.    'mp4' => array('video/mp4', 'video/x-m4v'),
139.    'm4a' => 'audio/x-m4a',
140.    'f4v' => array('video/mp4', 'video/x-f4v'),
141.    'flv' => 'video/x-flv',
142.    'webm' => 'video/webm',
143.    'aac' => array('audio/x-aac', 'audio/aac'),
144.    'm4u' => 'application/vnd.mpegurl',
145.    'm3u' => 'text/plain',
146.    'xspf' => 'application/xspf+xml',
147.    'vlc' => 'application/videolan',
148.    'wmv' => array('video/x-ms-wmv', 'video/x-ms-asf'),

```

```

149.         'au'      => 'audio/x-au',
150.         'ac3'     => 'audio/ac3',
151.         'flac'    => 'audio/x-flac',
152.         'ogg'     => array('audio/ogg', 'video/ogg', 'application/ogg'),
153.         'kmz'     => array('application/vnd.google-
earth.kmz', 'application/zip', 'application/x-zip'),
154.         'kml'     => array('application/vnd.google-
earth.kml+xml', 'application/xml', 'text/xml'),
155.         'ics'     => 'text/calendar',
156.         'ical'    => 'text/calendar',
157.         'zsh'     => 'text/x-scriptzsh',
158.         '7z'      => array('application/x-7z-compressed', 'application/x-
compressed', 'application/x-zip-compressed', 'application/zip', 'multipart/x-zip'),
159.         '7zip'    => array('application/x-7z-compressed', 'application/x-
compressed', 'application/x-zip-compressed', 'application/zip', 'multipart/x-zip'),
160.         'cdr'     => array('application/cdr', 'application/coreldraw', 'application/
x-cdr', 'application/x-coreldraw', 'image/cdr', 'image/x-cdr', 'zz-application/zz-
winassoc-cdr'),
161.         'wma'     => array('audio/x-ms-wma', 'video/x-ms-asf'),
162.         'jar'     => array('application/java-archive', 'application/x-java-
application', 'application/x-jar', 'application/x-compressed'),
163.         'svg'     => array('image/svg+xml', 'application/xml', 'text/xml'),
164.         'vcf'     => 'text/x-vcard',
165.         'srt'     => array('text/srt', 'text/plain'),
166.         'vtt'     => array('text/vtt', 'text/plain'),
167.         'ico'     => array('image/x-icon', 'image/x-
ico', 'image/vnd.microsoft.icon'),
168.         'odc'     => 'application/vnd.oasis.opendocument.chart',
169.         'otc'     => 'application/vnd.oasis.opendocument.chart-template',
170.         'odf'     => 'application/vnd.oasis.opendocument.formula',
171.         'otf'     => 'application/vnd.oasis.opendocument.formula-template',
172.         'odg'     => 'application/vnd.oasis.opendocument.graphics',
173.         'otg'     => 'application/vnd.oasis.opendocument.graphics-template',
174.         'odi'     => 'application/vnd.oasis.opendocument.image',
175.         'oti'     => 'application/vnd.oasis.opendocument.image-template',
176.         'odp'     => 'application/vnd.oasis.opendocument.presentation',
177.         'otp'     => 'application/vnd.oasis.opendocument.presentation-template',
178.         'ods'     => 'application/vnd.oasis.opendocument.spreadsheet',
179.         'ots'     => 'application/vnd.oasis.opendocument.spreadsheet-template',
180.         'odt'     => 'application/vnd.oasis.opendocument.text',
181.         'odm'     => 'application/vnd.oasis.opendocument.text-master',
182.         'ott'     => 'application/vnd.oasis.opendocument.text-template',
183.         'oth'     => 'application/vnd.oasis.opendocument.text-web'
184.     );

```

application\config\profiler.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | Profiler Sections
7. | -----
8. | This file lets you determine whether or not various sections of Profiler
9. | data are displayed when the Profiler is enabled.
10. | Please see the user guide for info:
11. |
12. | https://codeigniter.com/user\_guide/general/profiling.html
13. |

```

```
14. */
```

application\config\routes.php

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | URI ROUTING
7. | -----
8. | This file lets you re-map URI requests to specific controller functions.
9. |
10. | Typically there is a one-to-one relationship between a URL string
11. | and its corresponding controller class/method. The segments in a
12. | URL normally follow this pattern:
13. |
14. |   example.com/class/method/id/
15. |
16. | In some instances, however, you may want to remap this relationship
17. | so that a different class/function is called than the one
18. | corresponding to the URL.
19. |
20. | Please see the user guide for complete details:
21. |
22. |   https://codeigniter.com/user_guide/general/routing.html
23. |
24. | -----
25. | RESERVED ROUTES
26. | -----
27. |
28. | There are three reserved routes:
29. |
30. |   $route['default_controller'] = 'welcome';
31. |
32. | This route indicates which controller class should be loaded if the
33. | URI contains no data. In the above example, the "welcome" class
34. | would be loaded.
35. |
36. |   $route['404_override'] = 'errors/page_missing';
37. |
38. | This route will tell the Router which controller/method to use if those
39. | provided in the URL cannot be matched to a valid route.
40. |
41. |   $route['translate_uri_dashes'] = FALSE;
42. |
43. | This is not exactly a route, but allows you to automatically route
44. | controller and method names that contain dashes. '-' isn't a valid
45. | class or method name character, so it requires translation.
46. | When you set this option to TRUE, it will replace ALL dashes in the
47. | controller and method URI segments.
48. |
49. | Examples: my-controller/index -> my_controller/index
50. |           my-controller/my-method -> my_controller/my_method
51. */
52. $route['default_controller'] = 'auth';
53. $route['404_override'] = '';
54. $route['translate_uri_dashes'] = FALSE;
```

application\config\smileys.php

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5. | -----
6. | SMILEYS
7. | -----
8. | This file contains an array of smileys for use with the emoticon helper.
9. | Individual images can be used to replace multiple smileys. For example:
10. | :-) and :) use the same image replacement.
11. |
12. | Please see user guide for more info:
13. | https://codeigniter.com/user\_guide/helpers/smiley\_helper.html
14. |
15. */
16. $smileys = array(
17.
18. // smiley          image name          width  height  alt
19.
20. ':-)'              => array('grin.gif',      '19',   '19',   'grin'),
21. ':lol:'            => array('lol.gif',      '19',   '19',   'LOL'),
22. ':cheese:'         => array('cheese.gif',   '19',   '19',   'cheese'),
23. ':)'               => array('smile.gif',   '19',   '19',   'smile'),
24. ';-)'              => array('wink.gif',    '19',   '19',   'wink'),
25. ';)'               => array('wink.gif',    '19',   '19',   'wink'),
26. ':smirk:'          => array('smirk.gif',   '19',   '19',   'smirk'),
27. ':roll:'           => array('rolleyes.gif', '19',   '19',   'rolleyes'),
28. ':-S'              => array('confused.gif', '19',   '19',   'confused'),
29. ':wow:'            => array('surprise.gif', '19',   '19',   'surprised'),
30. ':bug:'            => array('bigsurprise.gif', '19',   '19',   'big surprise'),
31. ':-P'              => array('tongue_laugh.gif', '19',   '19',   'tongue laugh'),
32. '%-'               => array('tongue_rolleye.gif', '19',   '19',   'tongue rolleye'),
33. ';-P'              => array('tongue_wink.gif', '19',   '19',   'tongue wink'),
34. ':P'               => array('raspberry.gif', '19',   '19',   'raspberry'),
35. ':blank:'          => array('blank.gif',    '19',   '19',   'blank stare'),
36. ':long:'           => array('longface.gif',   '19',   '19',   'long face'),
37. ':ohh:'            => array('ohh.gif',      '19',   '19',   'ohh'),
38. ':grrr:'           => array('grrr.gif',    '19',   '19',   'grrr'),
39. ':gulp:'           => array('gulp.gif',    '19',   '19',   'gulp'),
40. '8-/'              => array('ohoh.gif',    '19',   '19',   'oh oh'),
41. ':down:'           => array('downer.gif',   '19',   '19',   'downer'),
42. ':red:'            => array('embarrassed.gif', '19',   '19',   'red face'),
43. ':sick:'           => array('sick.gif',     '19',   '19',   'sick'),
44. ':shut:'           => array('shuteye.gif',   '19',   '19',   'shut eye'),
45. ':-/'              => array('hmm.gif',     '19',   '19',   'hmmm'),
46. '>:(('             => array('mad.gif',      '19',   '19',   'mad'),
47. ':mad:'            => array('mad.gif',      '19',   '19',   'mad'),
48. '>:-('            => array('angry.gif',    '19',   '19',   'angry'),
49. ':angry:'          => array('angry.gif',    '19',   '19',   'angry'),
50. ':zip:'            => array('zip.gif',      '19',   '19',   'zipper'),
51. ':kiss:'           => array('kiss.gif',     '19',   '19',   'kiss'),
52. ':ahhh:'           => array('shock.gif',    '19',   '19',   'shock'),
53. ':coolsmile:'      => array('shade_smile.gif', '19',   '19',   'cool smile'),
54. ':coolsmirk:'      => array('shade_smirk.gif', '19',   '19',   'cool smirk'),
55. ':coolgrin:'       => array('shade_grin.gif', '19',   '19',   'cool grin'),
56. ':coolhmm:'        => array('shade_hmm.gif', '19',   '19',   'cool hmm'),
57. ':coolmad:'        => array('shade_mad.gif', '19',   '19',   'cool mad'),
```

```

58.     ':coolcheese:' => array('shade_cheese.gif', '19', '19', 'cool cheese'),
59.     ':vampire:'    => array('vampire.gif',      '19', '19', 'vampire'),
60.     ':snake:'      => array('snake.gif',        '19', '19', 'snake'),
61.     ':exclaim:'    => array('exclaim.gif',      '19', '19', 'exclaim'),
62.     ':question:'   => array('question.gif',     '19', '19', 'question')
63.
64. );

```

application\config\user_agents.php

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. /*
5.  | -----
6.  | USER AGENT TYPES
7.  | -----
8.  | This file contains four arrays of user agent data. It is used by the
9.  | User Agent Class to help identify browser, platform, robot, and
10. | mobile device data. The array keys are used to identify the device
11. | and the array values are used to set the actual name of the item.
12. */
13. $platforms = array(
14.     'windows nt 10.0' => 'Windows 10',
15.     'windows nt 6.3'  => 'Windows 8.1',
16.     'windows nt 6.2'  => 'Windows 8',
17.     'windows nt 6.1'  => 'Windows 7',
18.     'windows nt 6.0'  => 'Windows Vista',
19.     'windows nt 5.2'  => 'Windows 2003',
20.     'windows nt 5.1'  => 'Windows XP',
21.     'windows nt 5.0'  => 'Windows 2000',
22.     'windows nt 4.0'  => 'Windows NT 4.0',
23.     'winnt4.0'        => 'Windows NT 4.0',
24.     'winnt 4.0'       => 'Windows NT',
25.     'winnt'           => 'Windows NT',
26.     'windows 98'      => 'Windows 98',
27.     'win98'           => 'Windows 98',
28.     'windows 95'      => 'Windows 95',
29.     'win95'           => 'Windows 95',
30.     'windows phone'   => 'Windows Phone',
31.     'windows'         => 'Unknown Windows OS',
32.     'android'         => 'Android',
33.     'blackberry'      => 'BlackBerry',
34.     'iphone'          => 'iOS',
35.     'ipad'            => 'iOS',
36.     'ipod'            => 'iOS',
37.     'os x'            => 'Mac OS X',
38.     'ppc mac'         => 'Power PC Mac',
39.     'freebsd'         => 'FreeBSD',
40.     'ppc'             => 'Macintosh',
41.     'linux'           => 'Linux',
42.     'debian'          => 'Debian',
43.     'sunos'           => 'Sun Solaris',
44.     'beos'            => 'BeOS',
45.     'apachebench'     => 'ApacheBench',
46.     'aix'             => 'AIX',
47.     'irix'            => 'Irix',
48.     'osf'             => 'DEC OSF',
49.     'hp-ux'           => 'HP-UX',
50.     'netbsd'          => 'NetBSD',

```

```

51.     'bsdi'           => 'BSDi',
52.     'openbsd'        => 'OpenBSD',
53.     'gnu'             => 'GNU/Linux',
54.     'unix'            => 'Unknown Unix OS',
55.     'symbian'         => 'Symbian OS'
56. );
57.
58.
59. // The order of this array should NOT be changed. Many browsers return
60. // multiple browser types so we want to identify the sub-type first.
61. $browsers = array(
62.     'OPR'              => 'Opera',
63.     'Flock'            => 'Flock',
64.     'Edge'             => 'Edge',
65.     'Chrome'           => 'Chrome',
66.     // Opera 10+ always reports Opera/9.80 and appends Version/<real version> to the us
er agent string
67.     'Opera.*?Version'  => 'Opera',
68.     'Opera'            => 'Opera',
69.     'MSIE'             => 'Internet Explorer',
70.     'Internet Explorer' => 'Internet Explorer',
71.     'Trident.* rv'     => 'Internet Explorer',
72.     'Shiira'           => 'Shiira',
73.     'Firefox'          => 'Firefox',
74.     'Chimera'          => 'Chimera',
75.     'Phoenix'          => 'Phoenix',
76.     'Firebird'         => 'Firebird',
77.     'Camino'           => 'Camino',
78.     'Netscape'         => 'Netscape',
79.     'OmniWeb'          => 'OmniWeb',
80.     'Safari'           => 'Safari',
81.     'Mozilla'          => 'Mozilla',
82.     'Konqueror'        => 'Konqueror',
83.     'icab'             => 'iCab',
84.     'Lynx'             => 'Lynx',
85.     'Links'            => 'Links',
86.     'hotjava'          => 'HotJava',
87.     'amaya'            => 'Amaya',
88.     'IBrowse'          => 'IBrowse',
89.     'Maxthon'          => 'Maxthon',
90.     'Ubuntu'           => 'Ubuntu Web Browser'
91. );
92.
93. $mobiles = array(
94.     // legacy array, old values commented out
95.     'mobileexplorer'   => 'Mobile Explorer',
96.     // 'openwave'        => 'Open Wave',
97.     // 'opera mini'      => 'Opera Mini',
98.     // 'operamini'       => 'Opera Mini',
99.     // 'elaine'          => 'Palm',
100.     'palmsource'        => 'Palm',
101.     // 'digital paths'   => 'Palm',
102.     // 'avantgo'         => 'Avantgo',
103.     // 'xiino'           => 'Xiino',
104.     'palmscape'         => 'Palmscape',
105.     // 'nokia'           => 'Nokia',
106.     // 'ericsson'        => 'Ericsson',
107.     // 'blackberry'      => 'BlackBerry',
108.     // 'motorola'        => 'Motorola'
109.
110.     // Phones and Manufacturers

```

```

111.         'motorola'      => 'Motorola',
112.         'nokia'         => 'Nokia',
113.         'nexus'         => 'Nexus',
114.         'palm'          => 'Palm',
115.         'iphone'        => 'Apple iPhone',
116.         'ipad'          => 'iPad',
117.         'ipod'          => 'Apple iPod Touch',
118.         'sony'          => 'Sony Ericsson',
119.         'ericsson'       => 'Sony Ericsson',
120.         'blackberry'     => 'BlackBerry',
121.         'cocoon'         => 'O2 Cocoon',
122.         'blazer'        => 'Treo',
123.         'lg'             => 'LG',
124.         'amoi'           => 'Amoi',
125.         'xda'            => 'XDA',
126.         'mda'            => 'MDA',
127.         'vario'          => 'Vario',
128.         'htc'            => 'HTC',
129.         'samsung'        => 'Samsung',
130.         'sharp'          => 'Sharp',
131.         'sie-'           => 'Siemens',
132.         'alcatel'        => 'Alcatel',
133.         'benq'           => 'BenQ',
134.         'ipaq'           => 'HP iPaq',
135.         'mot-'           => 'Motorola',
136.         'playstation portable' => 'PlayStation Portable',
137.         'playstation 3'   => 'PlayStation 3',
138.         'playstation vita'   => 'PlayStation Vita',
139.         'hiptop'         => 'Danger Hiptop',
140.         'nec-'           => 'NEC',
141.         'panasonic'       => 'Panasonic',
142.         'philips'         => 'Philips',
143.         'sagem'          => 'Sagem',
144.         'sanyo'           => 'Sanyo',
145.         'spv'            => 'SPV',
146.         'zte'             => 'ZTE',
147.         'sendo'           => 'Sendo',
148.         'nintendo dsi'    => 'Nintendo DSi',
149.         'nintendo ds'     => 'Nintendo DS',
150.         'nintendo 3ds'    => 'Nintendo 3DS',
151.         'wii'             => 'Nintendo Wii',
152.         'open web'        => 'Open Web',
153.         'openweb'         => 'OpenWeb',
154.         'meizu'           => 'Meizu',
155.
156.         // Operating Systems
157.         'android'         => 'Android',
158.         'symbian'          => 'Symbian',
159.         'SymbianOS'        => 'SymbianOS',
160.         'elaine'           => 'Palm',
161.         'series60'         => 'Symbian S60',
162.         'windows ce'       => 'Windows CE',
163.
164.         // Browsers
165.         'obigo'            => 'Obigo',
166.         'netfront'         => 'Netfront Browser',
167.         'openwave'         => 'Openwave Browser',
168.         'mobileexplorer'   => 'Mobile Explorer',
169.         'operamini'        => 'Opera Mini',
170.         'opera mini'       => 'Opera Mini',
171.         'opera mobi'       => 'Opera Mobile',

```

```

172.         'fennec'      => 'Firefox Mobile',
173.
174.         // Other
175.         'digital paths' => 'Digital Paths',
176.         'avantgo'       => 'AvantGo',
177.         'xiino'         => 'Xiino',
178.         'novarra'       => 'Novarra Transcoder',
179.         'vodafone'      => 'Vodafone',
180.         'docomo'        => 'NTT DoCoMo',
181.         'o2'            => 'O2',
182.
183.         // Fallback
184.         'mobile'        => 'Generic Mobile',
185.         'wireless'      => 'Generic Mobile',
186.         'j2me'          => 'Generic Mobile',
187.         'midp'           => 'Generic Mobile',
188.         'cldc'           => 'Generic Mobile',
189.         'up.link'        => 'Generic Mobile',
190.         'up.browser'     => 'Generic Mobile',
191.         'smartphone'     => 'Generic Mobile',
192.         'cellphone'      => 'Generic Mobile'
193.     );
194.
195.     // There are hundreds of bots but these are the most common.
196.     $robots = array(
197.         'googlebot'      => 'Googlebot',
198.         'msnbot'         => 'MSNBot',
199.         'baiduspider'    => 'Baiduspider',
200.         'bingbot'        => 'Bing',
201.         'slurp'          => 'Inktomi Slurp',
202.         'yahoo'          => 'Yahoo',
203.         'ask jeeves'     => 'Ask Jeeves',
204.         'fastcrawler'    => 'FastCrawler',
205.         'infoseek'       => 'InfoSeek Robot 1.0',
206.         'lycos'          => 'Lycos',
207.         'yandex'         => 'YandexBot',
208.         'mediapartners-google' => 'MediaPartners Google',
209.         'CRAZYWEBCRAWLER' => 'Crazy Webcrawler',
210.         'adsbot-google'  => 'AdsBot Google',
211.         'feedfetcher-google' => 'Feedfetcher Google',
212.         'curious george' => 'Curious George',
213.         'ia_archiver'     => 'Alexa Crawler',
214.         'MJ12bot'        => 'Majestic-12',
215.         'Uptimebot'      => 'Uptimebot'
216.     );

```

application\controllers\Activity.php

```

1.  <?php
2.  defined('BASEPATH') or exit('No direct script access allowed');
3.  include APPPATH . 'third_party/vendor/autoload.php';
4.
5.  use PhpOffice\PhpSpreadsheet\Reader\Xlsx;
6.
7.  class Activity extends CI_Controller
8.  {
9.      private $filename = "import_question";
10.     public function __construct()
11.     {
12.         parent::__construct();

```

```

13.     $this->load->library('form_validation');
14.     $this->load->model('Submenu_model', 'SM');
15.     $this->load->model('Role_model', 'role');
16.     $this->load->model('User_model', 'UM');
17.     $this->load->model('Menu_model', 'menu');
18.     $this->load->model('Approve_model', 'app');
19.     $this->load->model('Activity_model', 'Act');
20.     $this->load->model('Question_model', 'Que');
21.     $this->load->model('Quiz_model', 'quiz');
22.     $this->load->model('Enroll_model', 'Enroll');
23.     $this->load->model('Course_model', 'Course');
24. }
25.
26. public function activityadd()
27. {
28.     $section_id = $this->input->post('section_id');
29.     $act = $this->input->post('activity');
30.     if ($act == 'resource') {
31.         redirect('activity/addresource/' . $section_id);
32.     } else if ($act == 'quiz') {
33.         redirect('activity/addquiz/' . $section_id);
34.     }
35. }
36.
37. public function addquiz($id)
38. {
39.     $this->form_validation-
40. >set_rules('activity_name', 'Quiz Name', 'required|trim');
41.     $this->form_validation->set_rules('start_date', 'Start date', 'required');
42.     $this->form_validation->set_rules('end_date', 'End date', 'required');
43.     $this->form_validation->set_rules('limit', 'Limit time', 'required|is_numeric');
44.     $this->form_validation->set_rules('grade', 'Grade', 'required|is_numeric');
45.     $this->form_validation->set_rules('total_question', 'Total Question', 'required|is_numeric');
46.     if ($this->form_validation->run() == false) {
47.         $data['section_id'] = $id;
48.         $data['title'] = 'My Course';
49.         $emplid = $this->session->userdata('emplid');
50.         $role_id = $this->session->userdata('role_id');
51.         $data['menu'] = $this->menu->getMenu($role_id);
52.         $data['me'] = $this->UM->getUserByEmplid($emplid);
53.         $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
54. >num_rows();
55.         $data['approve'] = $this->app->getApproveByManagerId($data['me']['id'], 3)-
56. >result_array();
57.         $data['course_id'] = $this->Course->getSectionById($id)['course_id'];
58.         $this->load->view('templates/header', $data);
59.         _templatesHeader($data);
60.         $this->load->view('views_course/addquiz', $data);
61.         _templatesFooter();
62.         $this->load->view('templates/footer');
63.     } else {
64.         $start_date = strtotime($this->input->post('start_date'));
65.         $end_date = strtotime($this->input->post('end_date'));
66.         if ($start_date > $end_date) {
67.             $this->session-
68. >set_flashdata('message', 'Start date is greater than the end date!');

```

```

68.         $this->session->set_flashdata('icon', 'warning');
69.         redirect('activity/addquiz/' . $id);
70.     } else {
71.         $data = [
72.             'id' => '',
73.             'section_id' => $id,
74.             'icon' => $this->input->post('type_file'),
75.             'activity_name' => htmlspecialchars($this->input-
>post('activity_name')),
76.             'file_name' => '',
77.             'start_date' => $start_date,
78.             'end_date' => $end_date,
79.             'limit_time' => $this->input->post('limit'),
80.             'grade_to_pass' => $this->input->post('grade'),
81.             'total_question' => $this->input->post('total_question'),
82.             'type' => 'Quiz'
83.         ];
84.         $this->Act->addActivity($data);
85.         $result = $this->Act->getActivityBySectionId($id);
86.         $this->session->set_flashdata('message', 'Quiz successfully added!');
87.         $this->session->set_flashdata('icon', 'success');
88.         redirect('course/open/' . $result[0]['course_id']);
89.     }
90. }
91. }
92.
93. public function addresource($id)
94. {
95.     $data['section_id'] = $id;
96.     $this->form_validation-
>set_rules('activity_name', 'Activity Name', 'required|trim');
97.     if ($this->form_validation->run() == false) {
98.         $data['title'] = 'My Course';
99.         $emplid = $this->session->userdata('emplid');
100.        $role_id = $this->session->userdata('role_id');
101.        $data['menu'] = $this->menu->getMenu($role_id);
102.        $data['me'] = $this->UM->getUserByEmplid($emplid);
103.        $data['notif'] = $this->app-
>getApproveByManagerId($data['me']['id']->num_rows());
104.        $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
105.        $data['course_id'] = $this->Course-
>getSectionById($id)['course_id'];
106.
107.        $this->load->view('templates/header', $data);
108.        _templatesHeader($data);
109.        $this->load->view('views_course/addresource', $data);
110.        _templatesFooter();
111.        $this->load->view('templates/footer');
112.    } else {
113.        $upload = $this->Act->upload_file();
114.        $typefile = $upload['file']['file_ext'];
115.        if ($typefile == '.pdf') {
116.            $icon = 'fas fa-file-pdf';
117.        } elseif ($typefile == '.docx') {
118.            $icon = 'fas fa-file-word';
119.        } elseif ($typefile == '.pptx') {
120.            $icon = 'fas fa-file-powerpoint';
121.        } elseif ($typefile == '.xlsx') {
122.            $icon = 'fas fa-file-excel';

```

```

123.         } elseif ($typefile == '.mp4' || $typefile == '.flv' || $typefile ==
'.wmv') {
124.             $icon = 'fas fa-file-video';
125.         }
126.         if ($upload['result'] == 'success') {
127.             $data = [
128.                 'id' => '',
129.                 'section_id' => $id,
130.                 'icon' => $icon,
131.                 'activity_name' => htmlspecialchars($this->input-
>post('activity_name')),
132.                 'file_name' => $upload['file']['file_name'],
133.                 'start_date' => '',
134.                 'end_date' => '',
135.                 'limit_time' => '',
136.                 'grade_to_pass' => '',
137.                 'type' => 'Resource'
138.             ];
139.
140.             $this->Act->addActivity($data);
141.             $result = $this->Act->getActivityBySectionId($id);
142.             redirect('course/open/' . $result[0]['course_id']);
143.         } else {
144.             $this->session->set_flashdata('error', $upload['error']);
145.             $data['title'] = 'My Course';
146.             $emplid = $this->session->userdata('emplid');
147.             $role_id = $this->session->userdata('role_id');
148.             $data['menu'] = $this->menu->getMenu($role_id);
149.             $data['me'] = $this->UM->getUserByEmplid($emplid);
150.             $data['notif'] = $this->app-
>getApproveByManagerId($data['me']['id'])->num_rows();
151.             $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
152.
153.             $this->load->view('templates/header', $data);
154.             _templatesHeader($data);
155.             $this->load->view('views_course/addresource', $data);
156.             _templatesFooter();
157.             $this->load->view('templates/footer');
158.         }
159.     }
160. }
161.
162. public function activitydetail($id)
163. {
164.     $data['title'] = 'My Course';
165.     $emplid = $this->session->userdata('emplid');
166.     $role_id = $this->session->userdata('role_id');
167.     $data['menu'] = $this->menu->getMenu($role_id);
168.     $data['me'] = $this->UM->getUserByEmplid($emplid);
169.     $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
170.     $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
171.     $data['encrypted_id'] = urlencode($this->encryption->encrypt($id));
172.     $course_id = $this->Act->getActivityById($id)['course_id'];
173.     $data['course'] = $this->Course->getCourseById($course_id);
174.     $check = $this->Enroll->checkUser($course_id, $data['me']['id'], 1);
175.
176.     if ($check > 0 or $data['course']['instructor_id'] == $data['me']['id']
or $role_id == 3) {

```

```

177.         $result = $this->Act->getActivityById($id);
178.         $data['activity'] = $result;
179.         if ($result['type'] == 'Resource') {
180.             redirect('uploads/resource-course/' . $result['file_name']);
181.         } else {
182.             $this->load->view('templates/header', $data);
183.             _templatesHeader($data);
184.             $this->load->view('views_course/activitydetail', $data);
185.             _templatesFooter();
186.             $this->load->view('templates/footer');
187.         }
188.     } else {
189.         $this->session->
190.         >set_flashdata('message', 'You have not enrolled for this course');
191.         $this->session->set_flashdata('icon', 'error');
192.         redirect('course');
193.     }
194.
195.     public function question($id)
196.     {
197.         $data['activity'] = $this->Act->getActivityById($id);
198.         $data['question'] = $this->Que->getQuestionByActId($id)-
199.         >result_array();
200.         $data['title'] = 'My Course';
201.         $emplid = $this->session->userdata('emplid');
202.         $role_id = $this->session->userdata('role_id');
203.         $data['menu'] = $this->menu->getMenu($role_id);
204.         $data['me'] = $this->UM->getUserByEmplid($emplid);
205.         $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
206.         >num_rows();
207.         $data['approve'] = $this->app-
208.         >getApproveByManagerId($data['me']['id'], 3)->result_array();
209.
210.         $this->load->view('templates/header', $data);
211.         _templatesHeader($data);
212.         $this->load->view('views_course/question', $data);
213.         _templatesFooter();
214.         $this->load->view('templates/footer');
215.     }
216.
217.     public function addquestion($id)
218.     {
219.         $this->form_validation->set_rules('question', 'Question', 'required');
220.         $this->form_validation->set_rules('answer_a', 'Question', 'required');
221.         $this->form_validation->set_rules('answer_b', 'Question', 'required');
222.         if ($this->form_validation->run() == false) {
223.             $data['activity'] = $this->Act->getActivityById($id);
224.             $data['title'] = 'My Course';
225.             $emplid = $this->session->userdata('emplid');
226.             $role_id = $this->session->userdata('role_id');
227.             $data['menu'] = $this->menu->getMenu($role_id);
228.             $data['me'] = $this->UM->getUserByEmplid($emplid);
229.             $data['notif'] = $this->app-
230.             >getApproveByManagerId($data['me']['id'])->num_rows();
231.             $data['approve'] = $this->app-
232.             >getApproveByManagerId($data['me']['id'], 3)->result_array();
233.
234.             $this->load->view('templates/header', $data);
235.             _templatesHeader($data);
236.             $this->load->view('views_course/addquestion', $data);

```

```

232.         _templatesFooter();
233.         $this->load->view('templates/footer');
234.     } else {
235.         $data = [
236.             'id' => '',
237.             'act_id' => $id,
238.             'bobot' => 1,
239.             'question' => $this->input->post('question'),
240.             'answer_a' => $this->input->post('answer_a'),
241.             'answer_b' => $this->input->post('answer_b'),
242.             'answer_c' => $this->input->post('answer_c'),
243.             'answer_d' => $this->input->post('answer_d'),
244.             'answer_e' => $this->input->post('answer_e'),
245.             'correct_answer' => $this->input->post('correct_answer')
246.         ];
247.         $this->Que->addQuestion($data);
248.         $this->session->
249.         >set_flashdata('message', 'Question successfully added!');
250.         $this->session->set_flashdata('icon', 'success');
251.         redirect('activity/question/' . $id);
252.     }
253.
254.     public function delete($id)
255.     {
256.         $act = $this->Act->getActivityById($id);
257.         if ($act['file_name'] !== '') {
258.             unlink(FCPATH . 'uploads/resource-course/' . $act['file_name']);
259.         }
260.         $this->Act->deleteAct($id);
261.         $this->Que->deleteQuisByActId($id);
262.         $this->session->
263.         >set_flashdata('message', 'Activity successfully deleted!');
264.         $this->session->set_flashdata('icon', 'success');
265.         redirect('course/open/' . $act['course_id']);
266.     }
267.
268.     public function edit($id)
269.     {
270.         $act = $this->Act->getActivityById($id);
271.         if ($act['type'] == 'Resource') {
272.             redirect('activity/editresource/' . $act['id']);
273.         } else if ($act['type'] == 'Quiz') {
274.             redirect('activity/editquiz/' . $act['id']);
275.         }
276.     }
277.
278.     public function editquiz($id)
279.     {
280.         $this->form_validation->
281.         >set_rules('activity_name', 'Quiz Name', 'required|trim');
282.         $this->form_validation->
283.         >set_rules('start_date', 'Start date', 'required');
284.         $this->form_validation->set_rules('end_date', 'End date', 'required');
285.         $this->form_validation->
286.         >set_rules('limit', 'Limit time', 'required|is_numeric');
287.         $this->form_validation->
288.         >set_rules('grade', 'Grade', 'required|is_numeric');
289.         $this->form_validation->
290.         >set_rules('total_question', 'Total Question', 'required|is_numeric');
291.     }

```

```

286.         $data['act'] = $this->Act->getActivityById($id);
287.         if ($this->form_validation->run() == false) {
288.             $data['section_id'] = $id;
289.             $data['title'] = 'My Course';
290.             $emplid = $this->session->userdata('emplid');
291.             $role_id = $this->session->userdata('role_id');
292.             $data['menu'] = $this->menu->getMenu($role_id);
293.             $data['me'] = $this->UM->getUserByEmplid($emplid);
294.             $data['notif'] = $this->app-
>getApproveByManagerId($data['me']['id']->num_rows();
295.             $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
296.
297.             $this->load->view('templates/header', $data);
298.             _templatesHeader($data);
299.             $this->load->view('views_course/editquiz', $data);
300.             _templatesFooter();
301.             $this->load->view('templates/footer');
302.         } else {
303.             $act = $data['act'];
304.             $start_date = strtotime($this->input->post('start_date'));
305.             $end_date = strtotime($this->input->post('end_date'));
306.
307.             if ($start_date > $end_date) {
308.                 $this->session-
>set_flashdata('message', 'Start date is greater than the end date!');
309.                 $this->session->set_flashdata('icon', 'warning');
310.                 redirect('activity/editquiz/' . $id);
311.             } else {
312.                 $data = [
313.                     'activity_name' => htmlspecialchars($this->input-
>post('activity_name')),
314.                     'start_date' => htmlspecialchars($start_date),
315.                     'end_date' => htmlspecialchars($end_date),
316.                     'limit_time' => htmlspecialchars($this->input-
>post('limit')),
317.                     'grade_to_pass' => htmlspecialchars($this->input-
>post('grade')),
318.                     'total_question' => htmlspecialchars($this->input-
>post('total_question')),
319.                 ];
320.                 $this->Act->updateQuiz($data, $act['id']);
321.                 $this->session-
>set_flashdata('message', 'Quiz successfully updated!');
322.                 $this->session->set_flashdata('icon', 'success');
323.                 redirect('course/open/' . $act['course_id']);
324.             }
325.         }
326.     }
327.
328.     public function deletequestion($id)
329.     {
330.         $q = $this->Que->getQuestionById($id)->row()->act_id;
331.         $this->Que->deleteQuizById($id);
332.         $this->session-
>set_flashdata('message', 'Question successfully deleted!');
333.         $this->session->set_flashdata('icon', 'success');
334.         redirect('activity/question/' . $q);
335.     }
336.
337.     public function editquestion($id)

```

```

338.         {
339.             $this->form_validation->set_rules('question', 'Question', 'required');
340.             $this->form_validation->set_rules('answer_a', 'Answer A', 'required');
341.             $this->form_validation->set_rules('answer_b', 'Answer B', 'required');
342.             $this->form_validation->set_rules('answer_c', 'Answer C', 'required');
343.             $this->form_validation->set_rules('answer_d', 'Answer D', 'required');
344.             $this->form_validation->set_rules('answer_e', 'Answer E', 'required');
345.             $data['q'] = $this->Que->getQuestionById($id)->row_array();
346.             if ($this->form_validation->run() == false) {
347.                 $data['activity'] = $data['q']['act_id'];
348.                 $data['title'] = 'My Course';
349.                 $emplid = $this->session->userdata('emplid');
350.                 $role_id = $this->session->userdata('role_id');
351.                 $data['menu'] = $this->menu->getMenu($role_id);
352.                 $data['me'] = $this->UM->getUserByEmplid($emplid);
353.                 $data['notif'] = $this->app-
>getApproveByManagerId($data['me']['id'])->num_rows();
354.                 $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
355.                 $data['ca'] = ["A", "B", "C", "D", "E"];
356.                 $this->load->view('templates/header', $data);
357.                 _templatesHeader($data);
358.                 $this->load->view('views_course/editquestion', $data);
359.                 _templatesFooter();
360.                 $this->load->view('templates/footer');
361.             } else {
362.                 $que = $data['q'];
363.                 $data = [
364.                     'question' => htmlspecialchars($this->input-
>post('question')),
365.                     'answer_a' => htmlspecialchars($this->input-
>post('answer_a')),
366.                     'answer_b' => htmlspecialchars($this->input-
>post('answer_b')),
367.                     'answer_c' => htmlspecialchars($this->input-
>post('answer_c')),
368.                     'answer_d' => htmlspecialchars($this->input-
>post('answer_d')),
369.                     'answer_e' => htmlspecialchars($this->input-
>post('answer_e')),
370.                     'correct_answer' => $this->input->post('correct_answer')
371.                 ];
372.                 $this->Que->updateQuestion($data, $que['id']);
373.                 $this->session-
>set_flashdata('message', 'Question successfully updated!');
374.                 $this->session->set_flashdata('icon', 'success');
375.                 redirect('activity/question/' . $que['act_id']);
376.             }
377.         }
378.
379.         public function reportquiz($id)
380.         {
381.             $data['title'] = 'My Course';
382.             $emplid = $this->session->userdata('emplid');
383.             $role_id = $this->session->userdata('role_id');
384.             $data['menu'] = $this->menu->getMenu($role_id);
385.             $data['me'] = $this->UM->getUserByEmplid($emplid);
386.             $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
387.             $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();

```

```

388.         $data['question'] = $this->Que->getQuestionByActId($id)->num_rows();
389.         $data['encrypted_id'] = urlencode($this->encryption->encrypt($id));
390.         $course_id = $this->Act->getActivityById($id)['course_id'];
391.         $data['course'] = $this->Course->getCourseById($course_id);
392.         $check = $this->Enroll->checkUser($course_id, $data['me']['id'], 1);
393.
394.         $data['user'] = $this->Enroll->getUserEnrolledById($course_id);
395.
396.         if ($check > 0 or $data['course']['instructor_id'] == $data['me']['id']
or $role_id == 3) {
397.             $result = $this->Act->getActivityById($id);
398.             $data['activity'] = $result;
399.             if ($result['type'] == 'Resource') {
400.                 redirect('uploads/resource-course/' . $result['file_name']);
401.             } else {
402.                 $this->load->view('templates/header', $data);
403.                 _templatesHeader($data);
404.                 $this->load->view('views_course/reportquiz', $data);
405.                 _templatesFooter();
406.                 $this->load->view('templates/footer');
407.             }
408.         } else {
409.             $this->session-
>set_flashdata('message', 'You have not enrolled for this course');
410.             $this->session->set_flashdata('icon', 'error');
411.             redirect('course');
412.         }
413.     }
414.
415.     public function reportprint($id)
416.     {
417.         $course_id = $this->Act->getActivityById($id)['course_id'];
418.         $data['user'] = $this->Enroll->getUserEnrolledById($course_id);
419.         $result = $this->Act->getActivityById($id);
420.         $data['activity'] = $result;
421.
422.         $this->load->view('views_course/reportprint', $data);
423.     }
424.
425.     public function importquestion($act_id)
426.     {
427.         check_session_instructor();
428.         $data = array(); // Buat variabel $data sebagai array
429.         if (isset($_POST['preview'])) { // Jika user menekan tombol Preview pada
form
430.             // lakukan upload file dengan memanggil function upload yang ada di
Siswa_model.php
431.             $upload = $this->Que->upload_question($this->filename);
432.             if ($upload['result'] == "success") { // Jika proses upload sukses
433.
434.                 // Load plugin PHPExcel nya
435.                 $excelreader = new Xlsx();
436.                 $loadexcel = $excelreader->load('excel/' . $this-
>filename . '.xlsx'); // Load file yang tadi diupload ke folder excel
437.                 $sheet = $loadexcel->getActiveSheet()-
>toArray(null, true, true, true);
438.                 // Masukkan variabel $sheet ke dalam array data yang nantinya aka
n di kirim ke file form.php
439.                 // Variabel $sheet tersebut berisi data-
data yang sudah diinput di dalam excel yang sudah di upload sebelumnya
439.                 $data['sheet'] = $sheet;

```

```

440.         } else { // Jika proses upload gagal
441.             $data['upload_error'] = $upload['error']; // Ambil pesan error u
ploadnya untuk dikirim ke file form dan ditampilkan
442.         }
443.     }
444.     $data['title'] = 'Course';
445.     $data['role'] = $this->role->getRole();
446.     $emplid = $this->session->userdata('emplid');
447.     $role_id = $this->session->userdata('role_id');
448.     $data['menu'] = $this->menu->getMenu($role_id);
449.     $data['me'] = $this->UM->getUserByEmplid($emplid);
450.     $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
451.     $data['approve'] = $this->app-
>getApproveByManagerId($data['me']['id'], 3)->result_array();
452.     $this->load->view('templates/header', $data);
453.     _templatesHeader($data);
454.     $this->load->view('views_course/importquestion');
455.     _templatesFooter();
456.     $this->load->view('templates/footer');
457. }
458.
459. public function import($act_id)
460. {
461.     check_session_instructor();
462.     // Load plugin PHPExcel nya
463.     $excelreader = new Xlsx();
464.     $loadexcel = $excelreader->load('excel/' . $this-
>filename . '.xlsx'); // Load file yang telah diupload ke folder excel
465.     $sheet = $loadexcel->getActiveSheet()-
>toArray(null, true, true);
466.     // Buat sebuah variabel array untuk menampung array data yg akan kita in
sert ke database
467.     $data = array();
468.     $numrow = 1;
469.     foreach ($sheet as $row) {
470.         // Cek $numrow apakah lebih dari 1
471.         // Artinya karena baris pertama adalah nama-nama kolom
472.         // Jadi dilewat saja, tidak usah diimport
473.         if ($numrow > 1) {
474.             // Kita push (add) array data ke variabel data
475.             array_push($data, array(
476.                 'id' => '',
477.                 'act_id' => $act_id, // Insert data nis dari kolom A di exce
1
478.                 'bobot' => 1, // Insert data nama dari kolom B di excel
479.                 'question' => $row['A'], // Insert data nama dari kolom B di
excel
480.                 'answer_a' => $row['B'], // Insert data jenis kelamin dari k
olom C di excel
481.                 'answer_b' => $row['C'], // Insert data alamat dari kolom D
di excel
482.                 'answer_c' => $row['D'], // Insert data alamat dari kolom D
di excel
483.                 'answer_d' => $row['E'], // Insert data alamat dari kolom D
di excel
484.                 'answer_e' => $row['F'], // Insert data alamat dari kolom D
di excel
485.                 'correct_answer' => strtoupper($row['G']), // Insert data al
amat dari kolom D di excel

```

```

486.         ));
487.     }
488.     $numrow++; // Tambah 1 setiap kali looping
489.
490. }
491. // Panggil fungsi insert_multiple yg telah kita buat sebelumnya di model
492. $this->Que->import_multiple($data);
493. $this->session->set_flashdata('message', 'Question added successfully!');
494. $this->session->set_flashdata('icon', 'success');
495. redirect("activity/question/" . $act_id); // Redirect ke halaman awal (ke controller siswa fungsi index)
496. }
497.
498. public function deletesection($id)
499. {
500.     $result = $this->Course->getSectionById($id);
501.     $course_id = $result['course_id'];
502.     $act = $this->Act->getActivityBySectionId($id);
503.     foreach ($act as $a) {
504.         if ($a['file_name'] != '') {
505.             unlink(FCPATH . 'uploads/resource-course/' . $a['file_name']);
506.         }
507.         $this->Act->deleteAct($a['id']);
508.     }
509.     $this->Course->deleteSection($id);
510.     $this->session->set_flashdata('message', 'Deleted');
511.     $this->session->set_flashdata('icon', 'success');
512.     redirect('course/open/' . $course_id);
513. }
514. }

```

application\controllers\Admin.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Admin extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         check_session_admin();
10.        $this->load->model('Submenu_model', 'SM');
11.        $this->load->model('Role_model', 'role');
12.        $this->load->model('User_model', 'UM');
13.        $this->load->model('Menu_model', 'menu');
14.        $this->load->model('Approve_model', 'app');
15.        $this->load->model('Approverej_model', 'apprej');
16.    }
17.
18.    public function setting()
19.    {
20.        $data['title'] = 'Setting';
21.        $emplid = $this->session->userdata('emplid');
22.        $role_id = $this->session->userdata('role_id');
23.        $data['menu'] = $this->menu->getMenu($role_id);
24.        $data['me'] = $this->UM->getUserByEmplid($emplid);

```

```

25.         $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
26.
27.         $this->load->view('templates/header', $data);
28.         _templatesHeader($data);
29.         $this->load->view('views_admin/setting', $data);
30.         _templatesFooter();
31.         $this->load->view('templates/footer');
32.     }
33.
34.     public function reset()
35.     {
36.         $this->db->empty_table('t_activity_course');
37.         $this->db->empty_table('t_course');
38.         $this->db->empty_table('t_course_category');
39.         $this->db->empty_table('t_course_enroll');
40.         $this->db->empty_table('t_question');
41.         $this->db->empty_table('t_result_quiz');
42.         $this->db->empty_table('t_section_course');
43.
44.         redirect('admin/setting');
45.     }
46. }

```

application\controllers\Auth.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Auth extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->library('form_validation');
10.        $this->load->model('User_model', 'UM');
11.    }
12.    public function index()
13.    {
14.        is_set_session();
15.        $this->form_validation->set_rules('emplid', 'NPK', 'required|trim');
16.        $this->form_validation->set_rules('password', 'Password', 'required|trim');
17.        if ($this->form_validation->run() == false) {
18.            $this->load->view('views_auth/login');
19.        } else {
20.            $this->_login();
21.        }
22.    }
23.
24.    private function _login()
25.    {
26.        $emplid = $this->input->post('emplid');
27.        $password = $this->input->post('password');
28.
29.        $user = $this->UM->getUserByEmplid($emplid);
30.        //jika user ada
31.        if ($user) {
32.            //jika user aktif
33.            if ($user['is_active'] == 1) {
34.                // cek password

```

```

35.         if (password_verify($password, $user['password'])) {
36.             $data = [
37.                 'emplid' => $user['emplid'],
38.                 'role_id' => $user['role_id'],
39.                 'id' => $user['id']
40.             ];
41.             $this->session->set_userdata($data);
42.             $insert = [
43.                 'id' => '',
44.                 'user_id' => $user['id'],
45.                 'status' => 'Online',
46.                 'time' => time()
47.             ];
48.             $this->db->insert('t_history_log', $insert);
49.             redirect('course/mycourse');
50.         } else {
51.             $this->session->set_flashdata('message', 'Wrong password!');
52.             $this->session->set_flashdata('icon', 'error');
53.             redirect('auth');
54.         }
55.     } else {
56.         $this->session->
>set_flashdata('message', 'This NPK has been suspended!');
57.         $this->session->
>set_flashdata('text', 'Please contact your administrator!');
58.         $this->session->set_flashdata('icon', 'error');
59.         redirect('auth');
60.     }
61. } else {
62.     $this->session->set_flashdata('message', 'NPK is not registered!');
63.     $this->session->
>set_flashdata('text', 'Please contact your administrator!');
64.     $this->session->set_flashdata('icon', 'error');
65.     redirect('auth');
66. }
67. }
68.
69. public function logout()
70. {
71.     $insert = [
72.         'id' => '',
73.         'user_id' => $this->session->userdata('id'),
74.         'status' => 'Offline',
75.         'time' => time()
76.     ];
77.     $this->db->insert('t_history_log', $insert);
78.     $this->session->unset_userdata('emplid');
79.     $this->session->unset_userdata('role_id');
80.     $this->session->unset_userdata('id');
81.     redirect('auth');
82. }
83.
84. public function forgot()
85. {
86.     is_set_session();
87.     $this->form_validation-
>set_rules('email', 'Email', 'trim|required|valid_email');
88.     if ($this->form_validation->run() == false) {
89.         $this->load->view('views_auth/forget_password');
90.     } else {
91.         $email = $this->input->post('email');

```

```

92.         $user = $this->UM->getUserByEmail($email)->row_array();
93.         if ($user) {
94.             $token = base64_encode(random_bytes(32));
95.             $user_token = [
96.                 'email' => $email,
97.                 'token' => $token,
98.                 'date_created' => time()
99.             ];
100.            $this->db->insert('t_user_token', $user_token);
101.            $this->sendEmail($email, $token);
102.            $this->session-
103.            >set_flashdata('message', 'Please check your email to reset password!');
104.            $this->session->set_flashdata('icon', 'success');
105.            redirect('auth/forgot');
106.        } else {
107.            $this->session-
108.            >set_flashdata('message', 'Email is not registered!');
109.            $this->session->set_flashdata('icon', 'error');
110.            redirect('auth/forgot');
111.        }
112.    }
113.    private function _sendEmail($email, $token)
114.    {
115.        $config = [
116.            'protocol' => 'smtp',
117.            'smtp_host' => 'ssl://mail.eiffory.com',
118.            'smtp_user' => 'elearning@eiffory.com',
119.            // 'smtp_pass' => '99HD-QjAP03=',
120.            'smtp_pass' => 'jy@q}k%*)mS=',
121.            'smtp_port' => '465',
122.            'mailtype' => 'html',
123.            'charset' => 'utf-8',
124.            'newline' => "\r\n",
125.        ];
126.
127.        $this->email->initialize($config);
128.
129.        $this->email->from('elearning@eiffory.com', 'Elearning Eiffory');
130.        $this->email->to($email);
131.
132.        $this->email->subject('Reset Password');
133.        $this->email-
134.        >message('Click this link to reset your password : <a href="' . base_url() . 'auth/resetpassword?email=' . $email . '&token=' . urlencode($token) . '">Reset Password</a>');
135.
136.        $this->email->send();
137.    }
138.    public function resetpassword()
139.    {
140.        is_set_session();
141.        $email = $this->input->get('email');
142.        $token = $this->input->get('token');
143.
144.        $user = $this->UM->getUserByEmail($email)->row_array();
145.        if ($user) {
146.            $user_token = $this->db-
147.            >get_where('t_user_token', ['token' => $token])->row_array();
148.            if ($user_token) {

```

```

148.             if (time() - $user_token['date_created'] < (60 * 60 * 24)) {
149.                 $this->session->set_userdata('email', $email);
150.                 $this->load->view('views_auth/resetpassword.php');
151.             } else {
152.                 $this->db->delete('t_user_token', ['email' => $email]);
153.                 $this->session-
>set_flashdata('message', 'Sorry, your link has been expired');
154.                 $this->session-
>set_flashdata('message', ' please enter your email again!');
155.                 $this->session->set_flashdata('icon', 'warning');
156.                 redirect('auth');
157.             }
158.         } else {
159.             $this->session-
>set_flashdata('message', 'Reset password failed. Wrong token!');
160.             $this->session->set_flashdata('icon', 'error');
161.             redirect('auth');
162.         }
163.     } else {
164.         $this->session-
>set_flashdata('message', 'Reset password failed. Wrong email!');
165.         $this->session->set_flashdata('icon', 'error');
166.         redirect('auth');
167.     }
168. }
169.
170. public function changepassword()
171. {
172.     if (!$this->session->userdata('email')) {
173.         redirect('auth');
174.     }
175.     $password = $this->input->post('password1');
176.     $this->form_validation-
>set_rules('password1', 'Password', 'required|min_length[3]|matches[password2]');
177.     $this->form_validation-
>set_rules('password2', 'Password', 'required|matches[password1]');
178.     if ($this->form_validation->run() == false) {
179.         $this->load->view('views_auth/resetpassword.php');
180.     } else {
181.         $update = [
182.             'password' => password_hash($password, PASSWORD_DEFAULT)
183.         ];
184.         $user = $this->UM->getUserByEmail($this->session-
>userdata('email'))->row_array();
185.         $this->UM->updateUser($update, $user['id']);
186.         $this->db->delete('t_user_token', ['email' => $user['email']]);
187.         $this->session->unset_userdata('email');
188.         $this->session-
>set_flashdata('message', 'Password successfully changed!');
189.         $this->session->set_flashdata('icon', 'success');
190.         redirect('auth');
191.     }
192. }
193.
194. public function blocked()
195. {
196.     $this->load->view('views_auth/blocked');
197. }
198. }

```

application\controllers\Chating.php

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Chating extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('Chat_model', 'chatModel');
10.        $this->load->model('Submenu_model', 'SM');
11.        $this->load->model('Role_model', 'role');
12.        $this->load->model('User_model', 'UM');
13.        $this->load->model('Menu_model', 'menu');
14.        $this->load->model('Approve_model', 'app');
15.        $this->load->model('Approverej_model', 'apprej');
16.        $this->load->model('Enroll_model', 'enrollModel');
17.        $this->load->model('Course_model', 'courseModel');
18.        $this->load->model('Log_model', 'logModel');
19.    }
20.
21.    public function chat($id)
22.    {
23.        $data['title'] = 'Chating';
24.        $emplid = $this->session->userdata('emplid');
25.        $role_id = $this->session->userdata('role_id');
26.        $data['menu'] = $this->menu->getMenu($role_id);
27.        $data['me'] = $this->UM->getUserByEmplid($emplid);
28.        $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])->num_rows();
29.        $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])->num_rows();
30.        $data['notif'] = $nwaitapp + $nrejapp;
31.        $instructor = $this->courseModel->getCourseById($id);
32.        $users = [
33.            0 => [
34.                'id' => $instructor['instructor_id'],
35.                'name' => $instructor['name'],
36.                'picture' => $instructor['user_picture'],
37.                'emplid' => $instructor['emplid'],
38.                'role' => 'Instructor'
39.            ]
40.        ];
41.        if ($instructor['name'] == $data['me']['name']) {
42.            $data['users'] = $this->enrollModel->getUserEnrolledById($id);
43.        } else {
44.            $data['users'] = $users;
45.        }
46.        $this->load->view('templates/header', $data);
47.        _templatesHeader($data);
48.        $this->load->view('views_chating/home');
49.        _templatesFooter();
50.        $this->load->view('templates/footer');
51.    }
52.
53.    public function getchat()
54.    {
55.        $meId = $this->input->get('meId');
56.        $emplid = $this->session->userdata('emplid');
57.        $data['me'] = $this->UM->getUserByEmplid($emplid);
```

```

58.     $youId = $this->input->get('youId');
59.     $input = [
60.         'is_read' => 1
61.     ];
62.     $this->chatModel->update($input, $meId, $youId);
63.     $data['chat'] = $this->chatModel->getChat($meId, $youId);
64.     $data['meId'] = $meId;
65.     $data['youId'] = $youId;
66.     $this->load->view('views_chating/chating', $data);
67. }
68.
69. public function getnotifchat()
70. {
71.     $emplid = $this->session->userdata('emplid');
72.     $meId = $this->UM->getUserByEmplid($emplid)['id'];
73.     $notif = $this->chatModel->getNotifChat($meId);
74.     if ($notif == 0) {
75.         echo '';
76.     } else {
77.         echo $notif;
78.     }
79. }
80.
81. public function getnotifchatdetail()
82. {
83.     $meId = $this->input->get('meId');
84.     $youId = $this->input->get('youId');
85.     $notif = $this->chatModel->getNotifChatDetail($meId, $youId);
86.     if ($notif == 0) {
87.         echo '';
88.     } else {
89.         echo $notif;
90.     }
91. }
92.
93. public function sendchat()
94. {
95.     $meId = $this->input->get('meId');
96.     $youId = $this->input->get('youId');
97.     $pesan = $this->input->get('pesan');
98.     $data = [
99.         'user_id_send' => $meId,
100.        'user_id_rcpt' => $youId,
101.        'message' => htmlspecialchars($pesan),
102.        'created_at' => date('Y-m-d H:i:s')
103.    ];
104.    $this->chatModel->addChat($data);
105.    $data['chat'] = $this->chatModel->getChat($meId, $youId);
106.    $emplid = $this->session->userdata('emplid');
107.    $data['me'] = $this->UM->getUserByEmplid($emplid);
108.    $data['meId'] = $meId;
109.    $data['youId'] = $youId;
110.    $this->load->view('views_chating/chating', $data);
111. }
112.
113. public function getstatususer()
114. {
115.     $user_id = $this->input->get('id');
116.     $result = $this->logModel->getLogByUid($user_id);
117.     if (!$result) {
118.         echo '<span class="text-danger">(Never Login)</span>';

```

```

119.         } else if ($result['status'] == 'Online') {
120.             echo '<span class="text-
success">(' . $result['status'] . ')</span>';
121.         } else {
122.             echo '<span class="text-primary">(Last access on ' . date('d-m-
y H:i', $result['time']) . ')</span>';
123.         }
124.     }
125. }

```

application\controllers\Course.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Course extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         check_session();
10.        $this->load->model('User_model', 'UM');
11.        $this->load->model('Org_model', 'org');
12.        $this->load->model('Role_model', 'role');
13.        $this->load->model('Menu_model', 'menu');
14.        $this->load->model('Submenu_model', 'SM');
15.        $this->load->model('Category_model', 'CM');
16.        $this->load->library('form_validation');
17.        $this->load->model('Course_model', 'Course');
18.        $this->load->model('Approve_model', 'app');
19.        $this->load->model('Approverej_model', 'apprej');
20.        $this->load->model('Enroll_model', 'Enroll');
21.        $this->load->model('Activity_model', 'Act');
22.        $this->load->model('Question_model', 'Que');
23.        $this->load->model('Quiz_model', 'quiz');
24.    }
25.    public function index()
26.    {
27.        $data['title'] = 'List Course';
28.        $emplid = $this->session->userdata('emplid');
29.        $role_id = $this->session->userdata('role_id');
30.        $data['menu'] = $this->menu->getMenu($role_id);
31.        $data['me'] = $this->UM->getUserByEmplid($emplid);
32.        $data['cat'] = $this->CM->getCategory();
33.        $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])->num_rows();
34.        $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])->
num_rows();
35.        $data['notif'] = $nwaitapp + $nrejapp;
36.        $data['course'] = $this->Course->getCourse();
37.
38.        $this->load->view('templates/header', $data);
39.        _templatesHeader($data);
40.        $this->load->view('views_course/listcourse', $data);
41.        _templatesFooter();
42.        $this->load->view('templates/footer_course');
43.    }
44.
45.    public function search()

```

```

46.     {
47.         $keyword = $this->input->get('keyword');
48.         $emplid = $this->session->userdata('emplid');
49.         $data['me'] = $this->UM->getUserByEmplid($emplid);
50.         $data['course'] = $this->Course->search($keyword);
51.         $this->load->view('views_course/listsearch', $data);
52.     }
53.
54.     public function createcourse()
55.     {
56.         check_session_instructor();
57.         $this->form_validation->set_rules('title', 'Title', 'required');
58.         if ($this->form_validation->run() == false) {
59.             $data['title'] = 'Create Course';
60.             $emplid = $this->session->userdata('emplid');
61.             $role_id = $this->session->userdata('role_id');
62.             $data['menu'] = $this->menu->getMenu($role_id);
63.             $data['me'] = $this->UM->getUserByEmplid($emplid);
64.             $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
65.             $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
66.             $data['notif'] = $nwaitapp + $nrejapp;
67.
68.             $data['cat'] = $this->CM->getCategory();
69.
70.             $this->load->view('templates/header', $data);
71.             _templatesHeader($data);
72.             $this->load->view('views_course/createcourse', $data);
73.             _templatesFooter();
74.             $this->load->view('templates/footer');
75.         } else {
76.             $upload = $this->Course->upload_cover();
77.             if ($upload['result'] == 'success') {
78.                 $this->Course->addCourse($upload['file']['file_name']);
79.                 $this->session-
>set_flashdata('message', 'Course successfully created!');
80.                 $this->session->set_flashdata('icon', 'success');
81.                 redirect('course');
82.             } else {
83.                 if ($upload['error'] == '<p>You did not select a file to upload.</p>')
{
84.                     $this->Course->addCourse();
85.                     $this->session-
>set_flashdata('message', 'Course successfully created!');
86.                     $this->session->set_flashdata('icon', 'success');
87.                     redirect('course');
88.                 } else {
89.                     $this->session->set_flashdata('error', '<div class="alert alert-
danger" role="alert">' . $upload['error'] . '</div>');
90.                     redirect('course/createcourse');
91.                 }
92.             }
93.         }
94.     }
95.
96.     public function delete($id)
97.     {
98.         $this->Course->delete($id);
99.         $this->session->set_flashdata('message', 'Course successfully deleted!');
100.        $this->session->set_flashdata('icon', 'success');

```

```

101.         redirect('course');
102.     }
103.
104.     public function category()
105.     {
106.         check_session_instructor();
107.         $data['title'] = 'Category';
108.         $emplid = $this->session->userdata('emplid');
109.         $role_id = $this->session->userdata('role_id');
110.         $data['menu'] = $this->menu->getMenu($role_id);
111.         $data['me'] = $this->UM->getUserByEmplid($emplid);
112.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
113.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
114.         $data['notif'] = $nwaitapp + $nrejapp;
115.         $data['cat'] = $this->CM->getCategory();
116.
117.         $this->load->view('templates/header', $data);
118.         _templatesHeader($data);
119.         $this->load->view('views_course/category', $data);
120.         _templatesFooter();
121.         $this->load->view('templates/footer');
122.     }
123.
124.     public function addcategory()
125.     {
126.         check_session_instructor();
127.         $this->form_validation-
>set_rules('category', 'Category', 'required|is_unique[t_course_category.category]');
128.         if ($this->form_validation->run() == false) {
129.             $this->session->set_flashdata('message', form_error('category'));
130.             $this->session->set_flashdata('icon', 'error');
131.             redirect('course/category');
132.         } else {
133.             $this->CM->addCategory();
134.             $this->session-
>set_flashdata('message', 'Category successfully added!');
135.             $this->session->set_flashdata('icon', 'success');
136.             redirect('course/category');
137.         }
138.     }
139.     public function mycourse()
140.     {
141.         $data['title'] = 'My Course';
142.         $emplid = $this->session->userdata('emplid');
143.         $role_id = $this->session->userdata('role_id');
144.         $data['menu'] = $this->menu->getMenu($role_id);
145.         $data['me'] = $this->UM->getUserByEmplid($emplid);
146.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
147.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
148.         $data['notif'] = $nwaitapp + $nrejapp;
149.         $data['cat'] = $this->CM->getCategory();
150.         if (!$this->input->get('search')) {
151.             $data['course'] = $this->Course-
>getCourseByEnrolledUser($data['me']['id']);
152.         } else {
153.             $data['course'] = $this->Course-
>getCourseByEnrolledUser($data['me']['id'], 1, $this->input->get('search'));

```

```

154.         }
155.         $this->load->view('templates/header', $data);
156.         _templatesHeader($data);
157.         $this->load->view('views_course/mycourse', $data);
158.         _templatesFooter();
159.         $this->load->view('templates/footer');
160.     }
161.
162.     public function detail()
163.     {
164.         $id = $this->input->post('id');
165.
166.         $result = $this->Course->getCourseById($id);
167.         $date = date('j F Y', $result['date_created']);
168.         $final_result = [
169.             'id' => $result['id'],
170.             'title' => $result['title'],
171.             'category' => $result['category'],
172.             'description' => $result['description'],
173.             'name' => $result['name'],
174.             'picture' => $result['picture'],
175.             'date_created' => $date,
176.             'is_active' => $result['is_active']
177.         ];
178.         echo json_encode($final_result);
179.     }
180.
181.     public function detailcategory()
182.     {
183.         $id = $this->input->post('id');
184.         $result = $this->CM->getCategoryById($id);
185.         echo json_encode($result);
186.     }
187.
188.     public function editcategory()
189.     {
190.         check_session_instructor();
191.         $this->form_validation->set_rules('category', 'Category', 'required');
192.         if ($this->form_validation->run() == false) {
193.             $this->session->set_flashdata('message', form_error('category'));
194.             $this->session->set_flashdata('icon', 'error');
195.             redirect('course/category');
196.         } else {
197.             $this->CM->editCategory();
198.             $this->session->set_flashdata('message', 'Category successfully updated!');
199.             $this->session->set_flashdata('icon', 'success');
200.             redirect('course/category');
201.         }
202.     }
203.
204.     public function deletecategory($id)
205.     {
206.         $this->CM->deleteCategory($id);
207.         $this->session->set_flashdata('message', 'Category successfully deleted!');
208.         $this->session->set_flashdata('icon', 'success');
209.         redirect('course/category');
210.     }
211.
212.     public function selfenroll()

```

```

213.         {
214.             $course_id = $this->input->post('course_id');
215.             $user_id = $this->input->post('user_id');
216.             if ($this->Enroll->checkUser($course_id, $user_id) > 0 or $this->Enroll-
>checkUser($course_id, $user_id, 1) > 0) {
217.                 $this->session-
>set_flashdata('message', 'You have registered in this course');
218.                 $this->session->set_flashdata('icon', 'error');
219.                 redirect('course');
220.             } else {
221.                 $this->Enroll->addUserEnroll();
222.                 $this->session-
>set_flashdata('message', 'You have successfully registered in this course');
223.                 $this->session-
>set_flashdata('text', 'please wait for your manager\'s approval');
224.                 $this->session->set_flashdata('icon', 'success');
225.                 redirect('course');
226.             }
227.         }
228.
229.         public function enroll()
230.         {
231.             $data['title'] = 'User Enroll';
232.             $emplid = $this->session->userdata('emplid');
233.             $role_id = $this->session->userdata('role_id');
234.             $data['menu'] = $this->menu->getMenu($role_id);
235.             $data['me'] = $this->UM->getUserByEmplid($emplid);
236.             $data['cat'] = $this->CM->getCategory();
237.             $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
238.             $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
239.             $data['notif'] = $nwaitapp + $nrejapp;
240.             $instructor_id = $data['me']['id'];
241.             if ($role_id == 3) {
242.                 $data['course'] = $this->Course->getCourse();
243.             } else {
244.                 $data['course'] = $this->Course-
>getCourseByInstructorID($instructor_id);
245.             }
246.
247.             $this->load->view('templates/header', $data);
248.             _templatesHeader($data);
249.             $this->load->view('views_course/enroll', $data);
250.             _templatesFooter();
251.             $this->load->view('templates/footer');
252.         }
253.
254.         public function enrolluser($id)
255.         {
256.             $data['title'] = 'User Enroll';
257.             $emplid = $this->session->userdata('emplid');
258.             $role_id = $this->session->userdata('role_id');
259.             $data['menu'] = $this->menu->getMenu($role_id);
260.             $data['me'] = $this->UM->getUserByEmplid($emplid);
261.             $data['cat'] = $this->CM->getCategory();
262.             $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
263.             $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
264.             $data['notif'] = $nwaitapp + $nrejapp;

```

```

265.         $data['course'] = $this->Course->getCourseById($id);
266.
267.         $data['users'] = $this->UM->getUser();
268.
269.         $data['enrolled'] = $this->Enroll->getUserEnrolledById($id);
270.
271.         $this->load->view('templates/header', $data);
272.         _templatesHeader($data);
273.         $this->load->view('views_course/enrolluser', $data);
274.         _templatesFooter();
275.         $this->load->view('templates/footer');
276.     }
277.
278.     public function unenroll()
279.     {
280.         $userId = $this->input->get('user');
281.         $courseId = $this->input->get('course');
282.
283.         $this->Enroll->unenroll($userId, $courseId);
284.         $this->session->set_flashdata('message', 'User successfully unenrolled!');
285.         $this->session->set_flashdata('icon', 'success');
286.         redirect('course/enrolluser/' . $courseId);
287.     }
288.
289.     public function enrollcourse()
290.     {
291.         $user = $this->input->post('user');
292.         $u = count($user);
293.         $course_id = $this->input->post('id');
294.         for ($i = 0; $i < $u; $i++) {
295.             $data = [
296.                 'id' => '',
297.                 'course_id' => $course_id,
298.                 'user_id' => $user[$i],
299.                 'is_approve' => 1
300.             ];
301.             $this->Enroll->enrollCourse($data);
302.         }
303.         $this->session->set_flashdata('message', 'User successfully enrolled!');
304.         $this->session->set_flashdata('icon', 'success');
305.         redirect('course/enrolluser/' . $course_id);
306.     }
307.
308.     public function notif($id)
309.     {
310.         // $this->app->updateRead($id);
311.         $data['title'] = 'Notifikasi';
312.         $emplid = $this->session->userdata('emplid');
313.         $role_id = $this->session->userdata('role_id');
314.         $data['menu'] = $this->menu->getMenu($role_id);
315.         $data['me'] = $this->UM->getUserByEmplid($emplid);
316.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
317.         >num_rows();
318.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
319.         >num_rows();
320.         $data['notif'] = $nwaitapp + $nrejapp;
321.         $data['app'] = $this->app->getApproveById($id);
322.         $this->load->view('templates/header', $data);
323.         _templatesHeader($data);

```

```

322.         $this->load->view('views_course/notif', $data);
323.         _templatesFooter();
324.         $this->load->view('templates/footer');
325.     }
326.
327.     public function notifdetail()
328.     {
329.         // $this->app->updateRead($id);
330.         $data['title'] = 'Notifikasi';
331.         $emplid = $this->session->userdata('emplid');
332.         $role_id = $this->session->userdata('role_id');
333.         $data['menu'] = $this->menu->getMenu($role_id);
334.         $data['me'] = $this->UM->getUserByEmplid($emplid);
335.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
336.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
337.         $data['notif'] = $nwaitapp + $nrejapp;
338.         $data['appreject'] = $this->apprej-
>getApproveRejectByUserId($data['me']['id'])->result_array();
339.         $data['app'] = $this->app->getApproveByManagerId($data['me']['id'])-
>result_array();
340.         $data['waitapp'] = $this->app->getApproveByUserId($data['me']['id']);
341.         $this->load->view('templates/header', $data);
342.         _templatesHeader($data);
343.         $this->load->view('views_course/notifdetail', $data);
344.         _templatesFooter();
345.         $this->load->view('templates/footer');
346.     }
347.
348.     public function approve($id)
349.     {
350.         $this->app->updateApprove($id);
351.         $this->session->set_flashdata('message', 'Approved');
352.         $this->session->set_flashdata('icon', 'success');
353.         redirect('course/notifdetail');
354.     }
355.
356.     public function detailapp()
357.     {
358.         $id = $this->input->post('id');
359.         $result = $this->Enroll->getEnrollUser($id);
360.         echo json_encode($result);
361.     }
362.
363.     public function open($id)
364.     {
365.         // $this->app->updateRead($id);
366.         $data['title'] = 'My Course';
367.         $emplid = $this->session->userdata('emplid');
368.         $role_id = $this->session->userdata('role_id');
369.         $data['menu'] = $this->menu->getMenu($role_id);
370.         $data['me'] = $this->UM->getUserByEmplid($emplid);
371.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
372.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
373.         $data['notif'] = $nwaitapp + $nrejapp;
374.         $data['course'] = $this->Course->getCourseById($id);
375.         $check = $this->Enroll->checkUser($id, $data['me']['id'], 1);
376.         $data['section'] = $this->Course->section($id);

```

```

377.         if ($check > 0 or $data['course']['instructor_id'] == $data['me']['id']
or $role_id == 3) {
378.             $this->load->view('templates/header', $data);
379.             _templatesHeader($data);
380.             $this->load->view('views_course/courseopen', $data);
381.             _templatesFooter();
382.             $this->load->view('templates/footer');
383.         } else {
384.             $this->session-
>set_flashdata('message', 'You have not enrolled for this course');
385.             $this->session->set_flashdata('icon', 'error');
386.             redirect('course');
387.         }
388.     }
389.
390.     public function sectionadd()
391.     {
392.         $this->form_validation-
>set_rules('section_name', 'Section Name', 'required');
393.         $course_id = htmlspecialchars($this->input->post('course_id'));
394.         if ($this->form_validation->run() == false) {
395.             $this->session-
>set_flashdata('message', form_error('section_name'));
396.             $this->session->set_flashdata('icon', 'error');
397.             redirect('course/open/' . $course_id);
398.         } else {
399.             $this->Course->addSection();
400.             $this->session-
>set_flashdata('message', 'Section successfully added!');
401.             $this->session->set_flashdata('icon', 'success');
402.             redirect('course/open/' . $course_id);
403.         }
404.     }
405.
406.     public function report()
407.     {
408.         $data['title'] = 'Report';
409.         $emplid = $this->session->userdata('emplid');
410.         $role_id = $this->session->userdata('role_id');
411.         $data['menu'] = $this->menu->getMenu($role_id);
412.         $data['me'] = $this->UM->getUserByEmplid($emplid);
413.         $data['cat'] = $this->CM->getCategory();
414.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
415.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
416.         $data['notif'] = $nwaitapp + $nrejapp;
417.         $instructor_id = $data['me']['id'];
418.         if ($role_id == 3) {
419.             $data['course'] = $this->Course->getCourse();
420.         } else {
421.             $data['course'] = $this->Course-
>getCourseByInstructorID($instructor_id);
422.         }
423.
424.         $this->load->view('templates/header', $data);
425.         _templatesHeader($data);
426.         $this->load->view('views_course/report', $data);
427.         _templatesFooter();
428.         $this->load->view('templates/footer');
429.     }

```

```

430.
431.     public function detailreport($id)
432.     {
433.         $data['title'] = 'Report';
434.         $emplid = $this->session->userdata('emplid');
435.         $role_id = $this->session->userdata('role_id');
436.         $data['menu'] = $this->menu->getMenu($role_id);
437.         $data['me'] = $this->UM->getUserByEmplid($emplid);
438.         $data['cat'] = $this->CM->getCategory();
439.         $nwaitapp = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
440.         $nrejapp = $this->apprej->getApproveRejectByUserId($data['me']['id'])-
>num_rows();
441.         $data['notif'] = $nwaitapp + $nrejapp;
442.         $data['course'] = $this->Course->getCourseById($id);
443.
444.         $data['users'] = $this->UM->getUser();
445.
446.         $data['enrolled'] = $this->Enroll->getUserEnrolledById($id);
447.
448.         $this->load->view('templates/header', $data);
449.         _templatesHeader($data);
450.         $this->load->view('views_course/detailreport', $data);
451.         _templatesFooter();
452.         $this->load->view('templates/footer');
453.     }
454.
455.     public function reject()
456.     {
457.         $user_id = $this->input->post('user_id');
458.         $course_id = $this->input->post('course_id');
459.
460.         $data = [
461.             'user_id' => $user_id,
462.             'course_id' => $course_id,
463.             'reason' => htmlspecialchars($this->input->post('reason'))
464.         ];
465.         $this->apprej->addReject($data);
466.         $this->Enroll->unenroll($user_id, $course_id);
467.         $this->session-
>set_flashdata('message', 'Your staff request has been rejected');
468.         $this->session->set_flashdata('icon', 'success');
469.         redirect('course/notifdetail');
470.     }
471.
472.     public function deletereject($id)
473.     {
474.         $this->apprej->delete($id);
475.         $this->session->set_flashdata('message', 'Deleted');
476.         $this->session->set_flashdata('icon', 'success');
477.         redirect('course/notifdetail');
478.     }
479. }

```

application\controllers\index.php

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>403 Forbidden</title>

```

```

5. </head>
6. <body>
7.
8. <p>Directory access is forbidden.</p>
9.
10. </body>
11. </html>

```

application\controllers\Manager.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Manager extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('User_model', 'UM');
10.        $this->load->model('Org_model', 'org');
11.        $this->load->model('Role_model', 'role');
12.        $this->load->model('Menu_model', 'menu');
13.        $this->load->model('Submenu_model', 'SM');
14.        $this->load->model('Manager_model', 'Mgr');
15.        $this->load->model('Approve_model', 'app');
16.        $this->load->library('form_validation');
17.        check_session_admin();
18.    }
19.
20.    public function index()
21.    {
22.        $data['users'] = $this->UM->getUser();
23.        $data['title'] = 'Managers Management';
24.        $emplid = $this->session->userdata('emplid');
25.        $role_id = $this->session->userdata('role_id');
26.        $data['menu'] = $this->menu->getMenu($role_id);
27.        $data['me'] = $this->UM->getUserByEmplid($emplid);
28.        // var_dump($data['menu']);
29.        // die;
30.        $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])->
>num_rows();
31.        $data['org'] = $this->org->getOrg();
32.        $data['managers'] = $this->Mgr->getManager();
33.        $this->load->view('templates/header', $data);
34.        _templatesHeader($data);
35.        $this->load->view('views_manager/index', $data);
36.        _templatesFooter();
37.        $this->load->view('templates/footer');
38.    }
39.
40.    public function getuserbyorg(){
41.        $org_id = $this->input->get('orgid');
42.        echo $org_id;
43.        $data['users'] = $this->Mgr->getUserByOrg($org_id);
44.        $this->load->view('views_manager/manager', $data);
45.    }
46.
47.    public function add()

```

```

48.     {
49.         $this->form_validation->set_rules('manager_id', 'Manager', 'required');
50.         if($this->form_validation->run() == false){
51.             $this->session->set_flashdata('message', 'failed to add Manager!');
52.             $this->session->set_flashdata('icon', 'error');
53.             redirect('manager');
54.         }else{
55.             $this->Mgr->addManager();
56.             $this->session->set_flashdata('message', 'Manager successfully added');
57.             $this->session->set_flashdata('icon', 'success');
58.             redirect('manager');
59.         }
60.     }
61.
62.     public function delete($id)
63.     {
64.         $this->Mgr->delete($id);
65.         $this->session->set_flashdata('message', 'Manager successfully deleted');
66.         $this->session->set_flashdata('icon', 'success');
67.         redirect('manager');
68.     }
69.
70.     public function test()
71.     {
72.         var_dump($this->app->getApproveByManagerId(7)->result_array());
73.     }
74. }

```

application\controllers\Quiz.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Quiz extends CI_Controller
5. {
6.     public function __construct()
7.     {
8.         parent::__construct();
9.         $this->load->model('User_model', 'UM');
10.        $this->load->model('Org_model', 'org');
11.        $this->load->model('Role_model', 'role');
12.        $this->load->model('Menu_model', 'menu');
13.        $this->load->model('Submenu_model', 'SM');
14.        $this->load->model('Activity_model', 'Act');
15.        $this->load->model('Question_model', 'Que');
16.        $this->load->model('Approve_model', 'app');
17.        $this->load->model('Resultquiz_model', 'result');
18.        $this->load->model('Quiz_model', 'quiz');
19.    }
20.
21.    public function index()
22.    {
23.        $key = $this->input->get('key', TRUE);
24.        $id = $this->encryption->decrypt(rawurldecode($key));
25.
26.        $quiz = $this->quiz->getQuizById($id);
27.

```

```

28.     $question = $this->Que->getQuestionByActIdRand($id, $quiz['total_question'])-
>result_array();
29.
30.     $emplid = $this->session->userdata('emplid');
31.     $me = $this->UM->getUserByEmplid($emplid);
32.     $user = $this->UM->getUserById($me['id']);
33.
34.     $result_quiz = $this->quiz->getResultQuizByActIdUserId($id, $user['id']);
35.     $check_user_done = $result_quiz->num_rows();
36.
37.     if ($check_user_done < 1) {
38.         $list_id_question = "";
39.         $list_ans_question = "";
40.
41.         if (!empty($question)) {
42.             foreach ($question as $q) {
43.                 $list_id_question .= $q['id'] . ",";
44.                 $list_ans_question .= $q['id'] . ":",";
45.             }
46.         }
47.         $list_id_question = substr($list_id_question, 0, -1);
48.         $list_ans_question = substr($list_ans_question, 0, -1);
49.         $waktu_selesai = date('Y-m-
d H:i:s', strtotime("+".$quiz['limit_time']." minute"));
50.         $time_mulai = date('Y-m-d H:i:s');
51.
52.         $input = [
53.             'act_id' => $id,
54.             'user_id' => $user['id'],
55.             'list_question' => $list_id_question,
56.             'list_answer' => $list_ans_question,
57.             'total_correct_answer' => 0,
58.             'grade' => 0,
59.             'grade_bobot' => 0,
60.             'start_date' => $time_mulai,
61.             'end_date' => $waktu_selesai,
62.             'status' => 'Y'
63.         ];
64.
65.         $this->db->insert('t_result_quiz', $input);
66.         redirect('quiz/?key=' . urlencode($key), 'location', 301);
67.     }
68.
69.     $q_question = $result_quiz->row_array();
70.
71.     $sequence_question = explode(",", $q_question['list_answer']);
72.
73.     for ($i = 0; $i < sizeof($sequence_question); $i++) {
74.         $qs_question = explode(':', $sequence_question[$i]);
75.         $qs_question1 = empty($qs_question[1]) ? "" : "'{$qs_question[1]
}';";
76.         $get_question = $this->quiz->getQuestion($qs_question1, $qs_question[0]);
77.         $question_sequence_ok[] = $get_question;
78.     }
79.
80.     $detail_test = $q_question;
81.     $question_sequence_ok = $question_sequence_ok;
82.     $arr_answer = [];
83.     $ex_list_answer = explode(",", $detail_test['list_answer']);
84.     foreach ($ex_list_answer as $list) {
85.         $ex_list = explode(":", $list);

```

```

86.         $idq      = $ex_list[0];
87.         $val       = $ex_list[1];
88.
89.         $arr_answer[$idq] = ["answer" => $val];
90.     }
91.     $arr_opsi = ["a", "b", "c", "d", "e"];
92.
93.     $html = '';
94.     $no = 1;
95.     if (!empty($question_sequence_ok)) {
96.         foreach ($question_sequence_ok as $q) {
97.             $html .= '<input type="hidden" name="id_soal_" . $no . " value="' . $q
98.             ['id'] . "'>';
99.             $html .= '<div class="step" id="widget_" . $no . "'>';
100.             $html .= '<div class="text-center"><div class="w-
101.             25"></div></div>' . $q['question'] . '<div class="funkyradio">';
102.             for ($j = 0; $j < 5; $j++) {
103.                 $opsi          = "answer_" . $arr_opsi[$j];
104.                 $checked       = $arr_answer[$q['id']]['answer'] == strtou
105.                 pper($arr_opsi[$j]) ? "checked" : "";
106.                 $pilihan_opsi  = !empty($q[$opsi]) ? $q[$opsi] : "";
107.
108.                 $html .= '<div class="funkyradio-
109.                 success" onclick="return simpan_sementara();">
110.                 <input type="radio" id="opsi_" . strtolower($arr_opsi[$j
111.                 ]) . "_" . $q['id'] . " name="opsi_" . $no . " value="' . strtoupper($arr_opsi[$j]) .
112.                 "' . $checked . "'> <label for="opsi_" . strtolower($arr_opsi[$j]) . "_" . $q['id'] .
113.                 "'><div class="huruf_opsi">' . $arr_opsi[$j] . '</div> <p>' . $pilihan_opsi . '</p><di
114.                 v class="w-25"></div></label></div>';
115.             }
116.             $html .= '</div></div>';
117.             $no++;
118.         }
119.     }
120.
121.     $id_test = $this->encryption->encrypt($detail_test['id']);
122.     $data = [
123.         'user'      => $user,
124.         'judul'     => 'Ujian',
125.         'subjudul'  => 'Lembar Ujian',
126.         'soal'      => $detail_test,
127.         'no'        => $no,
128.         'html'      => $html,
129.         'id_test'   => $id_test
130.     ];
131.
132.     $this->load->view('_templates/topnav/_header.php', $data);
133.     $this->load->view('quiz/sheet');
134.     $this->load->view('_templates/topnav/_footer.php');
135. }
136.
137. public function simpan_satu()
138. {
139.     // Decrypt Id
140.     $id_tes = $this->input->post('id', true);
141.     $id_tes = $this->encryption->decrypt($id_tes);
142.
143.     $input      = $this->input->post(null, true);
144.     $list_jawaban = "";
145.     for ($i = 1; $i < $input['jml_soal']; $i++) {

```

```

138.         $_tjawab      = "opsi_" . $i;
139.         $_tidsoal      = "id_soal_" . $i;
140.         $jawaban_      = emptyempty($_input[$_tjawab]) ? "" : $_input[$_tjawab]
;
141.         $list_jawaban   .= "" . $_input[$_tidsoal] . ":" . $jawaban_ . ",";

142.     }
143.     $list_jawaban       = substr($list_jawaban, 0, -1);
144.     $d_simpan = [
145.         'list_answer' => $list_jawaban
146.     ];
147.
148.     // Simpan jawaban
149.     $this->db->update('t_result_quiz', $d_simpan, ['id' => $id_tes]);
150.     json_encode(['status' => true]);
151. }
152.
153. public function simpan_akhir()
154. {
155.     // Decrypt Id
156.     $id_tes = $this->input->post('id');
157.     $id_tes = $this->encryption->decrypt($id_tes);
158.     // Get Jawaban
159.     $list_jawaban = $this->quiz->getAnswer($id_tes);
160.     // Pecah Jawaban
161.
162.     $pc_jawaban = explode(',', $list_jawaban);
163.     $jumlah_benar      = 0;
164.     $jumlah_salah      = 0;
165.     $nilai_bobot       = 0;
166.     $total_bobot       = 0;
167.     $jumlah_soal       = sizeof($pc_jawaban);
168.
169.     foreach ($pc_jawaban as $jwb) {
170.         $pc_dt          = explode(":", $jwb);
171.         $id_soal        = $pc_dt[0];
172.         $jawaban        = $pc_dt[1];
173.
174.         $cek_jwb        = $this->quiz->getQuestionById($id_soal);
175.         $total_bobot    = $total_bobot + $cek_jwb['bobot'];
176.
177.         $jawaban == $cek_jwb['correct_answer'] ? $jumlah_benar++ : $jumlah_s
alah++;
178.     }
179.
180.     $nilai = ($jumlah_benar / $jumlah_soal) * 100;
181.     $nilai_bobot = ($total_bobot / $jumlah_soal) * 100;
182.
183.     $d_update = [
184.         'total_correct_answer' => $jumlah_benar,
185.         'grade'                => number_format(floor($nilai), 0),
186.         'grade_bobot'          => number_format(floor($nilai_bobot), 0),
187.         'status'               => 'N'
188.     ];
189.
190.     $this->db->update('t_result_quiz', $d_update, ['id' => $id_tes]);
191.     $data = ['status' => TRUE, 'data' => $d_update, 'id' => $id_tes];
192.     $this->output->set_content_type('application/json')-
>set_output(json_encode($data));
193. }
194.

```

```

195.         public function list($id)
196.         {
197.             $act_id = $this->db->get_where('t_result_quiz', ['id' => $id])-
>row_array();
198.             redirect('activity/activitydetail/' . $act_id['act_id']);
199.         }
200.     }

```

application\controllers\User.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3. include APPPATH . 'third_party/vendor/autoload.php';
4.
5. use PhpOffice\PhpSpreadsheet\Reader\Xlsx;
6.
7. class User extends CI_Controller
8. {
9.     private $filename = "import_users";
10.
11.     public function __construct()
12.     {
13.         parent::__construct();
14.         $this->load->model('User_model', 'UM');
15.         $this->load->model('Org_model', 'org');
16.         $this->load->model('Role_model', 'role');
17.         $this->load->model('Menu_model', 'menu');
18.         $this->load->model('Submenu_model', 'SM');
19.         $this->load->model('Course_model', 'Course');
20.         $this->load->model('Approve_model', 'app');
21.         $this->load->library('form_validation');
22.     }
23.     public function index()
24.     {
25.         check_session_admin();
26.         $data['users'] = $this->UM->getUser();
27.         $data['title'] = 'Users Management';
28.         $emplid = $this->session->userdata('emplid');
29.         $role_id = $this->session->userdata('role_id');
30.         $data['menu'] = $this->menu->getMenu($role_id);
31.         $data['me'] = $this->UM->getUserByEmplid($emplid);
32.         // var_dump($data['menu']);
33.         // die;
34.         $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
35.         $this->load->view('templates/header', $data);
36.         _templatesHeader($data);
37.         $this->load->view('views_user/user', $data);
38.         _templatesFooter();
39.         $this->load->view('templates/footer');
40.     }
41.
42.     public function status($id)
43.     {
44.         check_session_admin();
45.         $this->UM->updateIs_Active($id);
46.         redirect('user');
47.     }

```

```

48.
49.     public function adduser()
50.     {
51.         check_session_admin();
52.         $this->form_validation->
>set_rules('emplid', 'NPK', 'required|trim|is_unique[t_user.emplid]', [
53.             'is_unique' => 'This NPK already exists!'
54.         ]);
55.         $this->form_validation->set_rules('name', 'Name', 'required|trim');
56.         $this->form_validation->
>set_rules('email', 'Email', 'required|trim|valid_email|is_unique[t_user.email]', [
57.             'is_unique' => 'This Email already exists!'
58.         ]);
59.         $this->form_validation->set_rules('phone', 'Phone', 'trim|is_numeric');
60.         $this->form_validation->set_rules('org', 'Organization', 'required');
61.         $this->form_validation->set_rules('role', 'Role', 'required');
62.         $this->form_validation->
>set_rules('password1', 'Password', 'required|trim|matches[password2]', [
63.             'matches' => 'Password don\'t matches!'
64.         ]);
65.         $this->form_validation->
>set_rules('password2', 'Password', 'required|trim|matches[password1]');
66.
67.         if ($this->form_validation->run() == FALSE) {
68.             $data['title'] = 'Users';
69.             $data['org'] = $this->org->getOrg();
70.             $data['role'] = $this->role->getRole();
71.             $emplid = $this->session->userdata('emplid');
72.             $role_id = $this->session->userdata('role_id');
73.             $data['menu'] = $this->menu->getMenu($role_id);
74.             $data['me'] = $this->UM->getUserByEmplid($emplid);
75.             $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
76.
77.             $this->load->view('templates/header', $data);
78.             _templatesHeader($data);
79.             $this->load->view('views_user/adduser', $data);
80.             _templatesFooter();
81.             $this->load->view('templates/footer');
82.         } else {
83.             $this->UM->addUser();
84.             $this->session->set_flashdata('message', 'User added successfully!');
85.             $this->session->set_flashdata('icon', 'success');
86.             redirect('user');
87.         }
88.     }
89.
90.     public function importuser()
91.     {
92.         check_session_admin();
93.         $data = array(); // Buat variabel $data sebagai array
94.         if (isset($_POST['preview'])) { // Jika user menekan tombol Preview pada form
95.             // lakukan upload file dengan memanggil function upload yang ada di Siswa_m
odel.php
96.             $upload = $this->UM->upload_user($this->filename);
97.             if ($upload['result'] == "success") { // Jika proses upload sukses
98.
99.                 // Load plugin PHPExcel nya
                 $excelreader = new Xlsx();

```

```

100.         $loadexcel = $excelreader->load('excel/' . $this-
>filename . '.xlsx'); // Load file yang tadi diupload ke folder excel
101.         $sheet = $loadexcel->getActiveSheet()-
>toArray(null, true, true);
102.         // Masukan variabel $sheet ke dalam array data yang nantinya aka
n di kirim ke file form.php
103.         // Variabel $sheet tersebut berisi data-
data yang sudah diinput di dalam excel yang sudah di upload sebelumnya
104.         $data['sheet'] = $sheet;
105.     } else { // Jika proses upload gagal
106.         $data['upload_error'] = $upload['error']; // Ambil pesan error u
ploadnya untuk dikirim ke file form dan ditampilkan
107.     }
108. }
109. $data['title'] = 'Users';
110. $data['role'] = $this->role->getRole();
111. $emplid = $this->session->userdata('emplid');
112. $role_id = $this->session->userdata('role_id');
113. $data['menu'] = $this->menu->getMenu($role_id);
114. $data['me'] = $this->UM->getUserByEmplid($emplid);
115. $data['notif'] = $this->app->getApproveByManagerId($data['me']['id'])-
>num_rows();
116. $this->load->view('templates/header', $data);
117. _templatesHeader($data);
118. $this->load->view('views_user/importuser', $data);
119. _templatesFooter();
120. $this->load->view('templates/footer');
121. }
122.
123. public function import()
124. {
125.     check_session_admin();
126.     // Load plugin PHPExcel nya
127.     $excelreader = new Xlsx();
128.     $loadexcel = $excelreader->load('excel/' . $this-
>filename . '.xlsx'); // Load file yang telah diupload ke folder excel
129.     $sheet = $loadexcel->getActiveSheet()-
>toArray(null, true, true);
130.     // Buat sebuah variabel array untuk menampung array data yg akan kita in
sert ke database
131.     $data = array();
132.     $numrow = 1;
133.     foreach ($sheet as $row) {
134.         // Cek $numrow apakah lebih dari 1
135.         // Artinya karena baris pertama adalah nama-nama kolom
136.         // Jadi dilewat saja, tidak usah diimport
137.
138.         if ($this->UM-
>getUserByEmplidOrEmail($row['A'], $row['D']) > 0 or $row['A'] == "" or $row['B'] == ""
or $row['C'] == "" or $row['D'] == "" or $row['E'] == "" or $row['F'] == "") {
139.             continue;
140.         } else {
141.             if ($numrow > 1) {
142.                 // Kita push (add) array data ke variabel data
143.                 $org = explode(".", $row['E']);
144.                 $role = explode(".", $row['F']);
145.                 array_push($data, array(
146.                     'id' => '',
147.                     'emplid' => $row['A'], // Insert data nis dari kolom A d
i excel

```

```

148.         'name' => $row['C'], // Insert data nama dari kolom B di
        excel
149.         'password' => password_hash($row['B'], PASSWORD_DEFAULT)
        , // Insert data nama dari kolom B di excel
150.         'email' => $row['D'], // Insert data jenis kelamin dari
        kolom C di excel
151.         'org_id' => $org[0], // Insert data alamat dari kolom D
        di excel
152.         'phone' => '', // Insert data alamat dari kolom D di exc
        el
153.         'picture' => 'default.jpg', // Insert data alamat dari k
        olom D di excel
154.         'is_active' => 1, // Insert data alamat dari kolom D di
        excel
155.         'about_me' => '', // Insert data alamat dari kolom D di
        excel
156.         'role_id' => $role[0], // Insert data alamat dari kolom
        D di excel
157.     ));
158.     }
159.     $numrow++; // Tambah 1 setiap kali looping
160. }
161. }
162. // Panggil fungsi insert_multiple yg telah kita buat sebelumnya di model

163. $this->UM->import_multiple($data);
164. $this->session->set_flashdata('message', 'Users added successfully!');
165. $this->session->set_flashdata('icon', 'success');
166. redirect("user"); // Redirect ke halaman awal (ke controller siswa fungs
        i index)
167. }
168.
169. public function edituser($id)
170. {
171.     check_session_admin();
172.     $this->form_validation->set_rules('emplid', 'NPK', 'required|trim', [
173.         'is_unique' => 'This NPK already exists!'
174.     ]);
175.     $this->form_validation->set_rules('name', 'Name', 'required|trim');
176.     $this->form_validation-
        >set_rules('email', 'Email', 'required|trim|valid_email', [
177.         'is_unique' => 'This Email already exists!'
178.     ]);
179.     $this->form_validation-
        >set_rules('phone', 'Phone', 'trim|is_numeric');
180.     $this->form_validation->set_rules('org', 'Organization', 'required');
181.     $this->form_validation->set_rules('role', 'Role', 'required');
182.     $this->form_validation-
        >set_rules('password1', 'Password', 'trim|matches[password2]', [
183.         'matches' => 'Password don\'t matches!'
184.     ]);
185.     $this->form_validation-
        >set_rules('password2', 'Password', 'trim|matches[password1]');
186.
187.     if ($this->form_validation->run() == FALSE) {
188.         $data['user'] = $this->UM->getUserById($id);
189.         $data['title'] = 'Users';
190.         $data['org'] = $this->org->getOrg();
191.         $data['role'] = $this->role->getRole();
192.         $emplid = $this->session->userdata('emplid');
193.         $role_id = $this->session->userdata('role_id');

```

```

194.         $data['menu'] = $this->menu->getMenu($role_id);
195.         $data['me'] = $this->UM->getUserByEmplid($emplid);
196.         $data['notif'] = $this->app-
>getApproveByManagerId($data['me']['id']->num_rows();
197.
198.         $this->load->view('templates/header', $data);
199.         _templatesHeader($data);
200.         $this->load->view('views_user/edituser', $data);
201.         _templatesFooter();
202.         $this->load->view('templates/footer');
203.     } else {
204.         $npk = $this->input->post('emplid');
205.         $email = $this->input->post('email');
206.         if ($this->UM->getUserByEmplidOrEmail($npk, $email) <= 1) {
207.             if ($this->input->post('password1') == '') {
208.                 $data = [
209.                     'id' => $this->input->post('id', TRUE),
210.                     'emplid' => htmlspecialchars($this->input-
>post('emplid', TRUE)),
211.                     'name' => htmlspecialchars($this->input-
>post('name', TRUE)),
212.                     'email' => htmlspecialchars($this->input-
>post('email', TRUE)),
213.                     'org_id' => htmlspecialchars($this->input-
>post('org', TRUE)),
214.                     'phone' => htmlspecialchars($this->input-
>post('phone', TRUE)),
215.                     'about_me' => htmlspecialchars($this->input-
>post('about')),
216.                     'role_id' => htmlspecialchars($this->input-
>post('role', TRUE))
217.                 ];
218.             } else {
219.                 $data = [
220.                     'id' => $this->input->post('id', TRUE),
221.                     'emplid' => $this->input->post('emplid', TRUE),
222.                     'name' => $this->input->post('name', TRUE),
223.                     'password' => password_hash(htmlspecialchars($this-
>input->post('password1', TRUE)), PASSWORD_DEFAULT),
224.                     'email' => htmlspecialchars($this->input-
>post('email', TRUE)),
225.                     'org_id' => htmlspecialchars($this->input-
>post('org', TRUE)),
226.                     'phone' => htmlspecialchars($this->input-
>post('phone', TRUE)),
227.                     'about_me' => htmlspecialchars($this->input-
>post('about')),
228.                     'role_id' => htmlspecialchars($this->input-
>post('role', TRUE))
229.                 ];
230.             }
231.             $this->UM->updateUser($data, $id);
232.             $this->session-
>set_flashdata('message', 'User successfully changed!');
233.             $this->session->set_flashdata('icon', 'success');
234.             redirect('user');
235.         } else {
236.             $this->session-
>set_flashdata('message', 'Sorry, user failed to change');
237.             $this->session-
>set_flashdata('text', 'NPK or email already exists');

```

```

238.         $this->session->set_flashdata('icon', 'error');
239.         redirect('user');
240.     }
241. }
242. }
243.
244. public function myprofile()
245. {
246.     $this->form_validation->set_rules('name', 'Name', 'required|trim');
247.     $this->form_validation-
248. >set_rules('email', 'Email', 'required|trim|valid_email');
249.     $this->form_validation-
250. >set_rules('phone', 'Phone', 'required|is_numeric');
251.     $emplid = $this->session->userdata('emplid');
252.     $role_id = $this->session->userdata('role_id');
253.     $data['me'] = $this->UM->getUserByEmplid($emplid);
254.     $data['user'] = $this->UM->getUserById($data['me']['id']);
255.     $data['mycourse'] = $this->Course->getMyCourse($data['me']['id'])-
256. >num_rows();
257.     if ($this->form_validation->run() == false) {
258.         $data['title'] = 'My Profile';
259.         $data['org'] = $this->org->getOrg();
260.         $data['role'] = $this->role->getRole();
261.
262.         $data['menu'] = $this->menu->getMenu($role_id);
263.
264.         $data['notif'] = $this->app-
265. >getApproveByManagerId($data['me']['id'])->num_rows();
266.
267.         $this->load->view('templates/header', $data);
268.         _templatesHeader($data);
269.         $this->load->view('views_user/myprofile', $data);
270.         _templatesFooter();
271.         $this->load->view('templates/footer');
272.     } else {
273.         $npk = $this->input->post('emplid');
274.         $email = $this->input->post('email');
275.         $id = $this->input->post('id');
276.         if ($this->UM->getUserByEmplidOrEmail($npk, $email) <= 1) {
277.             //
278.             $upload_image = $_FILES['img']['name'];
279.             if ($upload_image) {
280.                 $upload = $this->UM->upload_profile();
281.                 if ($upload['result'] == 'success') {
282.                     $old_image = $data['user']['picture'];
283.                     if ($old_image != 'default.jpg') {
284.                         unlink(FCPATH . 'uploads/profile/' . $old_image);
285.                     }
286.
287.                     $new_image = $upload['file']['file_name'];
288.                     $data = [
289.                         'id' => $id,
290.                         'emplid' => htmlspecialchars($this->input-
291. >post('emplid', TRUE)),
292.                         'name' => htmlspecialchars($this->input-
293. >post('name', TRUE)),
294.                         'email' => htmlspecialchars($this->input-
295. >post('email', TRUE)),
296.                         'about_me' => htmlspecialchars($this->input-
297. >post('about me', TRUE)),

```

```

290.                                     'phone' => htmlspecialchars($this->input-
>post('phone', TRUE)),
291.                                     'org_id' => htmlspecialchars($this->input-
>post('org_id', TRUE)),
292.                                     'picture' => $new_image
293.                                     ];
294.                                     } else {
295.                                         echo $upload['errors'];
296.                                     }
297.                                     } else {
298.                                         $data = [
299.                                             'id' => $id,
300.                                             'emplid' => htmlspecialchars($this->input-
>post('emplid', TRUE)),
301.                                             'name' => htmlspecialchars($this->input-
>post('name', TRUE)),
302.                                             'email' => htmlspecialchars($this->input-
>post('email', TRUE)),
303.                                             'about_me' => htmlspecialchars($this->input-
>post('about_me', TRUE)),
304.                                             'phone' => htmlspecialchars($this->input-
>post('phone', TRUE)),
305.                                             'org_id' => htmlspecialchars($this->input-
>post('org_id', TRUE)),
306.                                         ];
307.                                     }
308.
309.                                     $this->UM->updateUser($data, $id);
310.                                     $this->session-
>set_flashdata('message', 'User successfully changed!');
311.                                     $this->session->set_flashdata('icon', 'success');
312.                                     redirect('user/myprofile');
313.                                     } else {
314.                                         $this->session-
>set_flashdata('message', 'Sorry, user failed to change');
315.                                         $this->session-
>set_flashdata('text', 'NPK or email already exists');
316.                                         $this->session->set_flashdata('icon', 'error');
317.                                         redirect('user/myprofile');
318.                                     }
319.                                 }
320.                            }
321.
322.                            public function delete($id)
323.                            {
324.                                check_session_admin();
325.                                $this->UM->deleteUser($id);
326.                                $this->session-
>set_flashdata('message', 'User successfully deleted!');
327.                                $this->session->set_flashdata('icon', 'success');
328.                                redirect('user');
329.                            }
330.
331.                            public function changespassword()
332.                            {
333.                                $id = $this->input->post('id');
334.                                $password = $this->input->post('password');
335.                                $this->form_validation-
>set_rules('password', 'Password', 'required|min_length[4]|matches[c_password]');
336.                                $this->form_validation-
>set_rules('c_password', 'Confirm Password', 'required|matches[password]');

```

```

337.         if ($this->form_validation->run() == false) {
338.             $this->session->set_flashdata('message', form_error('password'));
339.             $this->session->set_flashdata('icon', 'error');
340.             redirect('user/myprofile');
341.         } else {
342.             $data = [
343.                 'password' => password_hash($password, PASSWORD_DEFAULT)
344.             ];
345.
346.             $this->UM->updateUser($data, $id);
347.             $this->session-
>set_flashdata('message', 'Password successfully changed');
348.             $this->session->set_flashdata('icon', 'success');
349.             redirect('user/myprofile');
350.         }
351.     }
352. }

```

application\core\index.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>403 Forbidden</title>
5. </head>
6. <body>
7.
8. <p>Directory access is forbidden.</p>
9.
10. </body>
11. </html>

```

application\helpers\index.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>403 Forbidden</title>
5. </head>
6. <body>
7.
8. <p>Directory access is forbidden.</p>
9.
10. </body>
11. </html>

```

application\helpers\learning_helper.php

```

1. <?php
2.
3. function check_session_admin()
4. {
5.     $app = get_instance();
6.     if (!$app->session->userdata('emplid')) {
7.         redirect('auth');
8.     } else {

```

```

9.         $role_id = $app->session->userdata('role_id');
10.
11.         if ($role_id != 3) {
12.             redirect('auth/blocked');
13.         }
14.     }
15. }
16.
17. function check_session()
18. {
19.     $app = get_instance();
20.     if (!$app->session->userdata('emplid')) {
21.         redirect('auth');
22.     }
23. }
24.
25. function is_set_session()
26. {
27.     $app = get_instance();
28.     if ($app->session->userdata('emplid')) {
29.         redirect('course/mycourse');
30.     }
31. }
32.
33. function check_session_instructor()
34. {
35.     $app = get_instance();
36.     if (!$app->session->userdata('emplid')) {
37.         redirect('auth');
38.     } else {
39.         $role_id = $app->session->userdata('role_id');
40.
41.         if ($role_id != 2 and $role_id != 3) {
42.             redirect('auth/blocked');
43.         }
44.     }
45. }
46.
47. function _templatesHeader($data)
48. {
49.     $app = get_instance();
50.     $app->load->view('templates/_header', $data);
51.     $app->load->view('templates/_sidebar', $data);
52. }
53.
54. function _templatesFooter()
55. {
56.     $app = get_instance();
57.     $app->load->view('templates/_footer');
58. }

```

application\models\Activity_model.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Activity_model extends CI_Model
5. {
6.     public function upload_file()
7.     {

```

```

8.         $new_file_name = date("mdY") . "_" . time();
9.         $config['upload_path'] = './uploads/resource-course';
10.        $config['allowed_types'] = 'pdf|docx|pptx|xlsx|mp4|flv|wmv';
11.        $config['max_size'] = '100000';
12.        $config['file_name'] = $new_file_name;
13.
14.        $this->upload->initialize($config); // Load konfigurasi uploadnya
15.        if ($this->upload-
>do_upload('upload')) { // Lakukan upload dan Cek jika proses upload berhasil
16.            // Jika berhasil :
17.            $return = array(
18.                'result' => 'success',
19.                'file' => $this->upload->data(),
20.                'error' => ''
21.            );
22.            return $return;
23.        } else { // Jika gagal :
24.            $return = array('result' => 'failed', 'file' => '', 'error' => $this-
>upload->display_errors());
25.            return $return;
26.        }
27.    }
28.
29.    public function addActivity($data)
30.    {
31.        $this->db->insert('t_activity_course', $data);
32.    }
33.
34.    public function getActivityBySectionId($id)
35.    {
36.        return $this->db->get_where('view_activity', ['section_id' => $id])-
>result_array();
37.    }
38.
39.    public function getActivityById($id)
40.    {
41.        return $this->db->get_where('view_activity', ['id' => $id])>row_array();
42.    }
43.
44.    public function updateQuiz($data, $id)
45.    {
46.        $this->db->update('t_activity_course', $data, ['id' => $id]);
47.    }
48.
49.    public function deleteAct($id)
50.    {
51.        $this->db->delete('t_activity_course', ['id' => $id]);
52.    }
53. }

```

application\models\Approve_model.php

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Approve_model extends CI_Model
5. {
6.     public function getApproveByManagerId($id, $limit = NULL)

```

```

7.     {
8.         return $this->db->get_where('view_approve', ['manager_id' => $id], $limit);
9.     }
10.
11.    public function getApproveById($id)
12.    {
13.        return $this->db->get_where('view_approve', ['id' => $id])->row_array();
14.    }
15.
16.    public function updateApprove($id)
17.    {
18.        $this->db->update('t_course_enroll', ['is_approve' => 1], ['id' => $id]);
19.    }
20.
21.    public function getApproveByUserId($user_id)
22.    {
23.        return $this->db->get_where('view_approve', ['user_id' => $user_id])-
24.        >result_array();
25.    }

```

application\models\Approverej_model

```

1.  <?php
2.  defined('BASEPATH') or exit('No direct script access allowed');
3.
4.  class Approverej_model extends CI_Model
5.  {
6.      public function getApproveRejectByUserId($user_id)
7.      {
8.          $this->db->select('t_course_rejected.id,
9.                          t_course_rejected.course_id,
10.                         t_course_rejected.user_id,
11.                         reason,
12.                         t_course.title,
13.                         t_course.date_created,
14.                         t_course_category.category');
15.          $this->db->from('t_course_rejected');
16.          $this->db->join('t_user', 't_user.id = t_course_rejected.user_id', 'left');
17.          $this->db->join('t_course', 't_course.id = t_course_rejected.course_id', 'left');
18.          $this->db->join('t_course_category', 't_course_category.id = t_course.category_id', 'left');
19.          $this->db->where('t_course_rejected.user_id', $user_id);
20.          return $this->db->get();
21.      }
22.
23.      public function addReject($data)
24.      {
25.          $this->db->insert('t_course_rejected', $data);
26.      }
27.
28.      public function delete($id)
29.      {
30.          $this->db->delete('t_course_rejected', ['id' => $id]);
31.      }
32. }

```

application\models\Category_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Category_model extends CI_Model
5. {
6.     public function getCategory()
7.     {
8.         return $this->db->get('t_course_category')->result_array();
9.     }
10.
11.    public function addCategory()
12.    {
13.        $data = [
14.            'id' => '',
15.            'category' => $this->input->post('category')
16.        ];
17.
18.        $this->db->insert('t_course_category', $data);
19.    }
20.
21.    public function getCategoryById($id)
22.    {
23.        return $this->db->get_where('t_course_category', ['id' => $id])->row_array();
24.    }
25.
26.    public function editCategory()
27.    {
28.        $id = $this->input->post('id');
29.        $data = [
30.            'id' => $id,
31.            'category' => $this->input->post('category')
32.        ];
33.
34.        $this->db->update('t_course_category', $data, ['id' => $id]);
35.    }
36.
37.    public function deleteCategory($id)
38.    {
39.        $this->db->delete('t_course_category', ['id' => $id]);
40.    }
41. }
```

application\models\Chat_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Chat_model extends CI_Model
5. {
6.     public function getChat($meId, $youId)
7.     {
8.         $this->db->select('*');
9.         $this->db->from('t_chating');
10.        $where = "user_id_send='{ $meId }' AND user_id_rcpt='{ $youId }' OR user_id_send='{ $youId }' AND user_id_rcpt='{ $meId }'";
```

```

11.         $this->db->where($where);
12.         $this->db->order_by('created_at', 'ASC');
13.         return $this->db->get()->result_array();
14.     }
15.
16.     public function getChatEnd($meId, $youId)
17.     {
18.         $this->db->order_by('created_at', 'DESC');
19.         return $this->db->get_where('t_chating', ['user_id_send' => $youId, 'user_id_rcpt' => $meId], 1)->row_array();
20.     }
21.
22.     public function addChat($data)
23.     {
24.         $this->db->insert('t_chating', $data);
25.     }
26.
27.     public function update($data, $meId, $youId)
28.     {
29.         $where = "user_id_send='{ $youId }' AND user_id_rcpt='{ $meId }'";
30.         $this->db->update('t_chating', $data, $where);
31.     }
32.
33.     public function getNotifChat($meId)
34.     {
35.         return $this->db->get_where('t_chating', ['user_id_rcpt' => $meId, 'is_read' => 0])->num_rows();
36.     }
37.
38.     public function getNotifChatDetail($meId, $youId)
39.     {
40.         $where = "user_id_send='{ $youId }' AND user_id_rcpt='{ $meId }' AND is_read=0";
41.         return $this->db->get_where('t_chating', $where)->num_rows();
42.     }
43. }

```

application\models\Course_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Course_model extends CI_Model
5. {
6.     public function upload_cover()
7.     {
8.         $new_file_name = date("mdY") . "_" . time();
9.         $config['upload_path'] = './uploads/cover-course';
10.        $config['allowed_types'] = 'jpg|jpeg|png';
11.        $config['max_size'] = '10000';
12.        $config['file_name'] = $new_file_name;
13.
14.        $this->upload->initialize($config); // Load konfigurasi uploadnya
15.        if ($this->upload->do_upload('cover')) { // Lakukan upload dan Cek jika proses upload berhasil
16.            // Jika berhasil :
17.            $return = array(
18.                'result' => 'success',

```

```

19.         'file' => $this->upload->data(),
20.         'error' => ''
21.     );
22.     return $return;
23. } else { // Jika gagal :
24.     $return = array('result' => 'failed', 'file' => '', 'error' => $this-
>upload->display_errors());
25.     return $return;
26. }
27. }
28.
29. public function addCourse($picture = 'default.jpg')
30. {
31.     $data = [
32.         'id' => '',
33.         'title' => htmlspecialchars($this->input->post('title')),
34.         'category_id' => $this->input->post('cat'),
35.         'description' => $this->input->post('desc'),
36.         'instructor_id' => $this->input->post('instructor_id'),
37.         'picture' => $picture,
38.         'date_created' => time(),
39.         'is_active' => 1,
40.     ];
41.
42.     $this->db->insert('t_course', $data);
43. }
44.
45. public function getCourse()
46. {
47.     $this->db-
>select('t_course.id, title, category, description, name, t_course.picture, date_create
d, t_course.is_active');
48.     $this->db->from('t_course');
49.     $this->db->join('t_course_category', 't_course_category.id = category_id');
50.     $this->db->join('t_user', 't_user.id = instructor_id');
51.     return $this->db->get()->result_array();
52. }
53.
54. public function search($keyword)
55. {
56.     $this->db-
>select('t_course.id, title, category, description, name, t_course.picture, date_create
d, t_course.is_active');
57.     $this->db->from('t_course');
58.     $this->db->join('t_course_category', 't_course_category.id = category_id');
59.     $this->db->join('t_user', 't_user.id = instructor_id');
60.     $this->db->like('title', $keyword);
61.     $this->db->or_like('category', $keyword);
62.     $this->db->or_like('description', $keyword);
63.     $this->db->or_like('name', $keyword);
64.     return $this->db->get()->result_array();
65. }
66.
67. public function getCourseByEnrolledUser($user_id, $is_approve = 1, $category = null
)
68. {
69.     $this->db-
>select('t_course.id, title, category, description, name, t_course.picture, date_create
d, t_course.is_active, t_course_enroll.user_id, is_approve');
70.     $this->db->from('t_course');
71.     $this->db->join('t_course_category', 't_course_category.id = category_id');

```

```

72.     $this->db->join('t_user', 't_user.id = instructor_id');
73.     $this->db->join('t_course_enroll', 't_course.id = course_id');
74.     $this->db->where('t_course_enroll.user_id', $user_id);
75.     $this->db->where('is_approve', $is_approve);
76.     if ($category !== null) {
77.         $this->db->where('category', $category);
78.     }
79.     return $this->db->get()->result_array();
80. }
81.
82. public function getCourseById($id)
83. {
84.     $this->db->
85.     >select('t_course.id, title, category, t_user.picture as user_picture, emplid, instruct
86.     or_id, description, name, t_course.picture, date_created, t_course.is_active');
87.     $this->db->from('t_course');
88.     $this->db->join('t_course_category', 't_course_category.id = category_id');
89.     $this->db->join('t_user', 't_user.id = instructor_id');
90.     $this->db->where('t_course.id', $id);
91.     return $this->db->get()->row_array();
92. }
93.
94. public function getCourseByInstructorId($id)
95. {
96.     $this->db->
97.     >select('t_course.id, title, category, description, name, t_course.picture, date_create
98.     d, t_course.is_active');
99.     $this->db->from('t_course');
100.    $this->db->join('t_course_category', 't_course_category.id = category_id');
101.    $this->db->join('t_user', 't_user.id = instructor_id');
102.    $this->db->where('instructor_id', $id);
103.    return $this->db->get()->result_array();
104. }
105.
106. public function delete($id)
107. {
108.     $this->db->delete('t_course', ['id' => $id]);
109. }
110.
111. public function section($id)
112. {
113.     return $this->db->get_where('t_section_course', ['course_id' => $id])-
114.     >result_array();
115. }
116.
117. public function addSection()
118. {
119.     $data = [
120.         'id' => '',
121.         'course_id' => $this->input->post('course_id'),
122.         'section_name' => htmlspecialchars($this->input-
123.         >post('section_name'))
124.     ];
125.     $this->db->insert('t_section_course', $data);
126. }
127.
128. public function deleteSection($id)
129. {
130.     $this->db->delete('t_section_course', ['id' => $id]);
131. }

```

```

127.         public function getSectionById($sec_id)
128.         {
129.             return $this->db->get_where('t_section_course', ['id' => $sec_id])-
>row_array();
130.         }
131.
132.         public function getMyCourse($id)
133.         {
134.             return $this->db-
>get_where('t_course_enroll', ['user_id' => $id, 'is_approve' => 1]);
135.         }
136.     }

```

application\models\Enroll_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Enroll_model extends CI_Model
5. {
6.     public function enrollCourse($data)
7.     {
8.         $this->db->insert('t_course_enroll', $data);
9.     }
10.
11.     public function getUserEnrolledById($id)
12.     {
13.         $this->db-
>select('t_course_enroll.id, t_user.id, emplid, name, role, t_user.picture, email, is_a
pprove');
14.         $this->db->from('t_course_enroll');
15.         $this->db->join('t_course', 'course_id = t_course.id', 'left');
16.         $this->db->join('t_user', 'user_id = t_user.id', 'left');
17.         $this->db->join('t_role', 't_user.role_id = t_role.id', 'left');
18.         $this->db->where('course_id', $id);
19.         $this->db->order_by('t_user.name', 'ASC');
20.         return $this->db->get()->result_array();
21.     }
22.
23.     public function checkUser($course_id, $user_id, $is_approve = 0)
24.     {
25.         $result = $this->db->get_where('t_course_enroll', [
26.             'course_id' => $course_id,
27.             'user_id' => $user_id,
28.             'is_approve' => $is_approve
29.         ]);
30.         return $result->num_rows();
31.     }
32.
33.     public function addUserEnroll()
34.     {
35.         $user_id = $this->input->post('user_id');
36.         $course_id = $this->input->post('course_id');
37.         $data = [
38.             'id' => '',
39.             'course_id' => $course_id,
40.             'user_id' => $user_id,
41.             'is_approve' => 0

```

```

42.     ];
43.
44.     $this->db->insert('t_course_enroll', $data);
45. }
46.
47. public function unenroll($userId, $courseId)
48. {
49.     $this->db->
>delete('t_course_enroll', ['user_id' => $userId, 'course_id' => $courseId]);
50. }
51.
52. public function getUserEnrollByCourseAndActId($course_id, $act_id)
53. {
54.     $this->db->
>select('view_enroll.course_id, view_enroll.act_id, view_enroll.user_id, t_user.name, t
_user.emplid, t_user.email');
55.     $this->db->from('view_enroll');
56.     $this->db->join('t_user', 'view_enroll.user_id = t_user.id', 'left');
57.     $this->db->where('view_enroll.course_id', $course_id);
58.     // $this->db->or_where('view_enroll.act_id', $act_id);
59.     return $this->db->get()->result_array();
60. }
61.
62. public function getEnrollUser($id)
63. {
64.     return $this->db->get_where('t_course_enroll', ['id' => $id])->row_array();
65. }
66. }

```

application\models\index.html

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <title>403 Forbidden</title>
5. </head>
6. <body>
7.
8. <p>Directory access is forbidden.</p>
9.
10. </body>
11. </html>

```

application\models\Log_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Log_model extends CI_Model
5. {
6.     public function getLogByUid($uid)
7.     {
8.         $this->db->order_by('time', 'DESC');
9.         return $this->db->get_where('t_history_log', ['user_id' => $uid], 1)-
>row_array();
10.     }
11. }

```

application\models\Manager_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Manager_model extends CI_Model
5. {
6.     public function getManager()
7.     {
8.         $this->db->select('t_manager.id, t_manager.org_id, t_org.org, t_manager.manager_id, t_user.name')
9.         ;
10.        $this->db->from('t_manager');
11.        $this->db->join('t_org', 't_manager.org_id = t_org.id', 'left');
12.        $this->db->join('t_user', 't_manager.manager_id = t_user.id', 'left');
13.        return $this->db->get()->result_array();
14.    }
15.
16.    public function checkMgr($org_id = null, $mgr_id = null)
17.    {
18.        $this->db->select('*');
19.        $this->db->from('t_manager');
20.        $this->db->where('org_id =', $org_id);
21.        $this->db->or_where('manager_id =', $mgr_id);
22.        return $this->db->get()->num_rows();
23.    }
24.
25.    public function addManager()
26.    {
27.        $data = [
28.            'id' => '',
29.            'org_id' => $this->input->post('org_id'),
30.            'manager_id' => $this->input->post('manager_id')
31.        ];
32.        $this->db->insert('t_manager', $data);
33.    }
34.
35.    public function delete($id)
36.    {
37.        $this->db->delete('t_manager', ['id' => $id]);
38.    }
39.
40.    public function getUserByOrg($org){
41.        return $this->db->get_where('t_user', ['org_id' => $org])->result_array();
42.    }
43. }
```

application\models\Menu_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Menu_model extends CI_Model
5. {
6.     public function getMenu($id)
7.     {
8.         $this->db->select('t_user_menu.id, menu');
```

```

9.         $this->db->from('t_user_menu');
10.        $this->db-
>join('t_user_access_menu', 't_user_menu.id = t_user_access_menu.menu_id');
11.        $this->db->where('t_user_access_menu.role_id', $id);
12.        $this->db->order_by('t_user_access_menu.menu_id', 'DESC');
13.        return $this->db->get()->result_array();
14.    }
15. }

```

application\models\Org_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Org_model extends CI_Model
5. {
6.     public function getOrg()
7.     {
8.         return $this->db->get('t_org')->result_array();
9.     }
10. }

```

application\models\Question_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Question_model extends CI_Model
5. {
6.     public function addQuestion($data)
7.     {
8.         $this->db->insert('t_question', $data);
9.     }
10.
11.     public function updateQuestion($data, $id)
12.     {
13.         $this->db->update('t_question', $data, ['id' => $id]);
14.     }
15.
16.     public function getQuestionByActId($id)
17.     {
18.         return $this->db->get_where('t_question', ['act_id' => $id]);
19.     }
20.
21.     public function getQuestionByActIdRand($id, $total_question)
22.     {
23.         $this->db->order_by('id', 'RANDOM');
24.         return $this->db-
>get_where('t_question', ['act_id' => $id], $total_question);
25.     }
26.
27.     public function getQuestionById($id)
28.     {
29.         return $this->db->get_where('t_question', ['id' => $id]);
30.     }
31.
32.     public function deleteQuisByActId($id)

```

```

33.     {
34.         $this->db->delete('t_question', ['act_id' => $id]);
35.     }
36.
37.     public function deleteQuizById($id)
38.     {
39.         $this->db->delete('t_question', ['id' => $id]);
40.     }
41.
42.     public function upload_question($filename)
43.     {
44.         $config['upload_path'] = './excel/';
45.         $config['allowed_types'] = 'xlsx';
46.         $config['max_size'] = '2048';
47.         $config['overwrite'] = true;
48.         $config['file_name'] = $filename;
49.
50.         $this->upload->initialize($config); // Load konfigurasi uploadnya
51.         if ($this->upload-
>do_upload('question')) { // Lakukan upload dan Cek jika proses upload berhasil
52.             // Jika berhasil :
53.             $return = array('result' => 'success', 'file' => $this->upload-
>data(), 'error' => '');
54.             return $return;
55.         } else { // Jika gagal :
56.             $return = array('result' => 'failed', 'file' => '', 'error' => $this-
>upload->display_errors());
57.             return $return;
58.         }
59.     }
60.
61.     public function import_multiple($data)
62.     {
63.         $this->db->insert_batch('t_question', $data);
64.     }
65. }

```

application\models\Quiz_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Quiz_model extends CI_Model
5. {
6.     public function getQuizById($id)
7.     {
8.         return $this->db->get_where('t_activity_course', ['id' => $id])->row_array();
9.     }
10.
11.     public function getQuestionByQuizId($id)
12.     {
13.         return $this->db->get_where('t_question', ['act_id' => $id])->result_array();
14.     }
15.
16.     public function getResultQuizByActIdUserId($act_id, $user_id)
17.     {
18.         return $this->db-
>get_where('t_result_quiz', ['act_id' => $act_id, 'user_id' => $user_id]);

```

```

19.     }
20.
21.     public function getQuestion($qs_question_i1, $qs_question_i0)
22.     {
23.         $this->db->select("*, {$qs_question_i1} AS correct_answer");
24.         $this->db->from('t_question');
25.         $this->db->where('id', $qs_question_i0);
26.         return $this->db->get()->row_array();
27.     }
28.
29.     public function getAnswer($id_tes)
30.     {
31.         $this->db->select('list_answer');
32.         $this->db->from('t_result_quiz');
33.         $this->db->where('id', $id_tes);
34.         return $this->db->get()->row_array()['list_answer'];
35.     }
36.
37.     public function getQuestionById($id)
38.     {
39.         return $this->db->get_where('t_question', ['id' => $id])->row_array();
40.     }
41. }

```

application\models\Resultquiz_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Resultquiz_model extends CI_Model
5. {
6.     // public function getResultByActIdAndUserId($id, $user_id)
7.     // {
8.     //     return $this->db-
9.     >get_where('t_result_quiz', ['act_id' => $id, 'user_id' => $user_id]);
10.    // }
11.    public function addResult($data)
12.    {
13.        $this->db->insert('t_result_quiz', $data);
14.    }
15. }

```

application\models\Role_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Role_model extends CI_Model
5. {
6.     public function getRole()
7.     {
8.         return $this->db->get('t_role')->result_array();
9.     }
10. }

```

application\models\Siswa_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Siswa_model extends CI_Model
5. {
6.     public function view()
7.     {
8.         return $this->db->get('siswa')-
>result(); // Tampilkan semua data yang ada di tabel siswa
9.     }
10.
11.     // Fungsi untuk melakukan proses upload file
12.     public function upload_file($filename)
13.     {
14.         $this->load->library('upload'); // Load librari upload
15.         $config['upload_path'] = './excel/';
16.         $config['allowed_types'] = 'xlsx';
17.         $config['max_size'] = '2048';
18.         $config['overwrite'] = true;
19.         $config['file_name'] = $filename;
20.
21.         $this->upload->initialize($config); // Load konfigurasi uploadnya
22.         if ($this->upload-
>do_upload('file')) { // Lakukan upload dan Cek jika proses upload berhasil
23.             // Jika berhasil :
24.             $return = array('result' => 'success', 'file' => $this->upload-
>data(), 'error' => '');
25.             return $return;
26.         } else { // Jika gagal :
27.             $return = array('result' => 'failed', 'file' => '', 'error' => $this-
>upload->display_errors());
28.             return $return;
29.         }
30.     }
31.
32.     public function insert_multiple($data)
33.     {
34.         $this->db->insert_batch('siswa', $data);
35.     }
36.
37.     public function get_data($data)
38.     {
39.         return $this->db->get_where('siswa', $data)->row_array();
40.     }
41.
42.     public function get_siswa($data)
43.     {
44.         return $this->db->get_where('siswa', ['nis' => $data])->num_rows();
45.     }
46. }
```

application\models\Submenu_model

```
1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
```

```

3.
4. class Submenu_model extends CI_Model
5. {
6.     public function getSubMenu($id)
7.     {
8.         $this->db->select('*');
9.         $this->db->from('t_user_sub_menu');
10.        $this->db->where('menu_id', $id);
11.        return $this->db->get()->result_array();
12.    }
13. }

```

application\models\Tabel1_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class Tabel1_model extends CI_Model
5. { }

```

application\models\User_model

```

1. <?php
2. defined('BASEPATH') or exit('No direct script access allowed');
3.
4. class User_model extends CI_Model
5. {
6.     public function getUser($order = 'ASC')
7.     {
8.         $this->db->
9.         >select('t_user.id, emplid, name, password, email, role, org, phone, picture, is_active
10.         , about_me');
11.         $this->db->from('t_user');
12.         $this->db->join('t_org', 'org_id = t_org.id', 'left');
13.         $this->db->join('t_role', 'role_id = t_role.id', 'left');
14.         $this->db->order_by('name', $order);
15.         return $this->db->get()->result_array();
16.     }
17.
18.     public function getUserById($id)
19.     {
20.         $this->db->
21.         >select('t_user.id, emplid, name, password, email, role, org, phone, picture, is_active
22.         , about_me, org_id, role_id');
23.         $this->db->from('t_user');
24.         $this->db->join('t_org', 'org_id = t_org.id', 'left');
25.         $this->db->join('t_role', 'role_id = t_role.id', 'left');
26.         $this->db->where('t_user.id', $id);
27.         return $this->db->get()->row_array();
28.     }
29.
30.     public function getUserByEmplid($emplid)
31.     {
32.         return $this->db->get_where('t_user', array('emplid' => $emplid))-
33.         >row_array();
34.     }
35. }

```

```

29.     }
30.
31.     public function getUserByEmplidOrEmail($emplid, $email)
32.     {
33.         $this->db->select('*');
34.         $this->db->where('emplid', $emplid);
35.         $this->db->or_where('email', $email);
36.         $this->db->get('t_user');
37.         return $this->db->affected_rows();
38.     }
39.
40.     public function getUserByEmail($email)
41.     {
42.         return $this->db->get_where('t_user', ['email' => $email, 'is_active' => 1]);
43.     }
44.
45.     public function updateIs_Active($id)
46.     {
47.         $user = $this->getUserById($id);
48.         if ($user['is_active'] == 1) {
49.             $this->db->where('id', $id);
50.             $this->db->update('t_user', array('is_active' => 0));
51.         } else {
52.             $this->db->where('id', $id);
53.             $this->db->update('t_user', array('is_active' => 1));
54.         }
55.     }
56.
57.     public function addUser()
58.     {
59.         $data = [
60.             'id' => '',
61.             'emplid' => htmlspecialchars($this->input->post('emplid', TRUE)),
62.             'name' => htmlspecialchars($this->input->post('name', TRUE)),
63.             'password' => password_hash(htmlspecialchars($this->input->
64.             >post('password1', TRUE)), PASSWORD_DEFAULT),
65.             'email' => htmlspecialchars($this->input->post('email', TRUE)),
66.             'org_id' => htmlspecialchars($this->input->post('org', TRUE)),
67.             'phone' => htmlspecialchars($this->input->post('phone', TRUE)),
68.             'picture' => 'default.jpg',
69.             'is_active' => 1,
70.             'about_me' => htmlspecialchars($this->input->post('about')),
71.             'role_id' => htmlspecialchars($this->input->post('role', TRUE))
72.         ];
73.         $this->db->insert('t_user', $data);
74.     }
75.
76.     public function updateUser($data, $id)
77.     {
78.         $this->db->update('t_user', $data, array('id' => $id));
79.     }
80.
81.     public function deleteUser($id)
82.     {
83.         $this->db->delete('t_user', ['id' => $id]);
84.     }
85.
86.     public function upload_user($filename)
87.     {
88.         $config['upload_path'] = './excel/';

```

```

89.         $config['allowed_types'] = 'xlsx';
90.         $config['max_size'] = '2048';
91.         $config['overwrite'] = true;
92.         $config['file_name'] = $filename;
93.
94.         $this->upload->initialize($config); // Load konfigurasi uploadnya
95.         if ($this->upload-
>do_upload('file')) { // Lakukan upload dan Cek jika proses upload berhasil
96.             // Jika berhasil :
97.             $return = array('result' => 'success', 'file' => $this->upload-
>data(), 'error' => '');
98.             return $return;
99.         } else { // Jika gagal :
100.             $return = array('result' => 'failed', 'file' => '', 'error' => $this
->upload->display_errors());
101.             return $return;
102.         }
103.     }
104.
105.     public function import_multiple($data)
106.     {
107.         $this->db->insert_batch('t_user', $data);
108.     }
109.
110.     public function upload_profile()
111.     {
112.         $new_file_name = date("mdY") . "_" . time();
113.         $config['upload_path'] = './uploads/profile';
114.         $config['allowed_types'] = 'jpg|jpeg|png';
115.         $config['max_size'] = '5000';
116.         $config['file_name'] = $new_file_name;
117.
118.         $this->upload->initialize($config); // Load konfigurasi uploadnya
119.         if ($this->upload-
>do_upload('img')) { // Lakukan upload dan Cek jika proses upload berhasil
120.             // Jika berhasil :
121.             $return = array(
122.                 'result' => 'success',
123.                 'file' => $this->upload->data(),
124.                 'error' => ''
125.             );
126.             return $return;
127.         } else { // Jika gagal :
128.             $return = array('result' => 'failed', 'file' => '', 'error' => $this
->upload->display_errors());
129.             return $return;
130.         }
131.     }
132. }

```

application\third_party\vendor\autoload

```

1. <?php
2.
3. // autoload.php @generated by Composer
4.
5. require_once __DIR__ . '/composer/autoload_real.php';
6.
7. return ComposerAutoloaderInit5bbff88ed03addcf6b3f7331b776e204::getLoader();

```

application\third_party\vendor\composer\autoload_classmap

```
1. <?php
2.
3. // autoload_classmap.php @generated by Composer
4.
5. $vendorDir = dirname(dirname(__FILE__));
6. $baseDir = dirname($vendorDir);
7.
8. return array(
9. );
```

application\third_party\vendor\composer\autoload_files

```
1. <?php
2.
3. // autoload_files.php @generated by Composer
4.
5. $vendorDir = dirname(dirname(__FILE__));
6. $baseDir = dirname($vendorDir);
7.
8. return array(
9.     '0e6d7bf4a5811bfa5cf40c5ccd6fae6a' => $vendorDir . '/symfony/polyfill-
mbstring/bootstrap.php',
10.    'abede361264e2ae69ec1eee813a101af' => $vendorDir . '/markbaker/complex/classes/src/
functions/abs.php',
11.    '21a5860fbef5be28db5ddfbc3cca67c4' => $vendorDir . '/markbaker/complex/classes/src/
functions/acos.php',
12.    '1546e3f9d127f2a9bb2d1b6c31c26ef1' => $vendorDir . '/markbaker/complex/classes/src/
functions/acosh.php',
13.    'd2516f7f4fba5ea5905f494b4a8262e0' => $vendorDir . '/markbaker/complex/classes/src/
functions/acot.php',
14.    '4511163d560956219b96882c0980b65e' => $vendorDir . '/markbaker/complex/classes/src/
functions/acoth.php',
15.    'c361f5616dc2a8da4fa3e137077cd4ea' => $vendorDir . '/markbaker/complex/classes/src/
functions/acsc.php',
16.    '02d68920fc98da71991ce569c91df0f6' => $vendorDir . '/markbaker/complex/classes/src/
functions/acsch.php',
17.    '88e19525eae308b4a6aa3419364875d3' => $vendorDir . '/markbaker/complex/classes/src/
functions/argument.php',
18.    '60e8e2d0827b58bfc904f13957e51849' => $vendorDir . '/markbaker/complex/classes/src/
functions/asec.php',
19.    '13d2f040713999eab66c359b4d79871d' => $vendorDir . '/markbaker/complex/classes/src/
functions/asech.php',
20.    '838ab38beb32c68a79d3cd2c007d5a04' => $vendorDir . '/markbaker/complex/classes/src/
functions/asin.php',
21.    'bb28eccd0f8f008333a1b3c163d604ac' => $vendorDir . '/markbaker/complex/classes/src/
functions/asinh.php',
22.    '9e483de83558c98f7d3feaa402c78cb3' => $vendorDir . '/markbaker/complex/classes/src/
functions/atan.php',
23.    '36b74b5b765ded91ee58c8ee3c0e85e3' => $vendorDir . '/markbaker/complex/classes/src/
functions/atanh.php',
24.    '05c15ee9510da7fd6bf6136f436500c0' => $vendorDir . '/markbaker/complex/classes/src/
functions/conjugate.php',
25.    'd3208dfbce2505e370788f9f22f6785f' => $vendorDir . '/markbaker/complex/classes/src/
functions/cos.php',
```

```
26. '141cf1fb3a3046f8b64534b0ebab33ca' => $vendorDir . '/markbaker/complex/classes/src/
functions/cosh.php',
27. 'be660df75fd0dbe7fa7c03b7434b3294' => $vendorDir . '/markbaker/complex/classes/src/
functions/cot.php',
28. '01e31ea298a51bc9e91517e3ce6b9e76' => $vendorDir . '/markbaker/complex/classes/src/
functions/coth.php',
29. '803ddd97f7b1da68982a7b087c3476f6' => $vendorDir . '/markbaker/complex/classes/src/
functions/csc.php',
30. '3001cdfd101ec3c32da34ee43c2e149b' => $vendorDir . '/markbaker/complex/classes/src/
functions/csch.php',
31. '77b2d7629ef2a93fabb8c56754a91051' => $vendorDir . '/markbaker/complex/classes/src/
functions/exp.php',
32. '4a4471296dec796c21d4f4b6552396a9' => $vendorDir . '/markbaker/complex/classes/src/
functions/inverse.php',
33. 'c3e9897e1744b88deb56fcdc39d34d85' => $vendorDir . '/markbaker/complex/classes/src/
functions/ln.php',
34. 'a83cacf2de942cff288de15a83afd26d' => $vendorDir . '/markbaker/complex/classes/src/
functions/log2.php',
35. '6a861dacc9ee2f3061241d4c7772fa21' => $vendorDir . '/markbaker/complex/classes/src/
functions/log10.php',
36. '4d2522d968c8ba78d6c13548a1b4200e' => $vendorDir . '/markbaker/complex/classes/src/
functions/negative.php',
37. 'fd587ca933fc0447fa5ab4843bdd97f7' => $vendorDir . '/markbaker/complex/classes/src/
functions/pow.php',
38. '383ef01c62028fc78cd4388082fce3c2' => $vendorDir . '/markbaker/complex/classes/src/
functions/rho.php',
39. '150fbd1b95029dc47292da97ecab9375' => $vendorDir . '/markbaker/complex/classes/src/
functions/sec.php',
40. '549abd9bae174286d660bdaa07407c68' => $vendorDir . '/markbaker/complex/classes/src/
functions/sech.php',
41. '6bfbf5eaea6b17a0ed85cb21ba80370c' => $vendorDir . '/markbaker/complex/classes/src/
functions/sin.php',
42. '22efe13f1a497b8e199540ae2d9dc59c' => $vendorDir . '/markbaker/complex/classes/src/
functions/sinh.php',
43. 'e90135ab8e787795a509ed7147de207d' => $vendorDir . '/markbaker/complex/classes/src/
functions/sqrt.php',
44. 'bb0a7923ffc6a90919cd64ec54ff06bc' => $vendorDir . '/markbaker/complex/classes/src/
functions/tan.php',
45. '2d302f32ce0fd4e433dd91c5bb404a28' => $vendorDir . '/markbaker/complex/classes/src/
functions/tanh.php',
46. '24dd4658a952171a4ee79218c4f9fd06' => $vendorDir . '/markbaker/complex/classes/src/
functions/theta.php',
47. 'e49b7876281d6f5bc39536dde96d1f4a' => $vendorDir . '/markbaker/complex/classes/src/
operations/add.php',
48. '47596e02b43cd6da7700134fd08f88cf' => $vendorDir . '/markbaker/complex/classes/src/
operations/subtract.php',
49. '883af48563631547925fa4c3b48ead07' => $vendorDir . '/markbaker/complex/classes/src/
operations/multiply.php',
50. 'f190e3308e6ca23234a2875edc985c03' => $vendorDir . '/markbaker/complex/classes/src/
operations/divideby.php',
51. 'ac9e33ce6841aa5bf5d16d465a2f03a7' => $vendorDir . '/markbaker/complex/classes/src/
operations/divideinto.php',
52. '9d8e013a5160a09477beb8e44f8ae97b' => $vendorDir . '/markbaker/matrix/classes/src/f
unctions/adjoint.php',
53. '6e78d1bdea6248d6aa117229efae50f2' => $vendorDir . '/markbaker/matrix/classes/src/f
unctions/antidiagonal.php',
54. '4623d87924d94f5412fe5afb1cef31d' => $vendorDir . '/markbaker/matrix/classes/src/f
unctions/cofactors.php',
55. '901fd1f6950a637ca85f66b701a45e13' => $vendorDir . '/markbaker/matrix/classes/src/f
unctions/determinant.php',
```

```

56.     '83057abc0e4acc99ba80154ee5d02a49' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/diagonal.php',
57.     '07b7fd7a434451149b4fd477fca0ce06' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/identity.php',
58.     'c8d43b340583e07ae89f2a3baef2cf89' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/inverse.php',
59.     '499bb10ed7a3aee2ba4c09a31a85e8d1' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/minors.php',
60.     '1cad2e6414d652e8b1c64e8967f6f37d' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/trace.php',
61.     '95a7f134ac17161d07def442b3b737e8' => $vendorDir . '/markbaker/matrix/classes/src/f
    unctions/transpose.php',
62.     'b3a6bc628377118d4b4b8ba08d1eb949' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/add.php',
63.     '5fef6d0e407f3f8887266dfa4a6c534c' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/directsum.php',
64.     '684ba247e1385946e3babdaa054119de' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/subtract.php',
65.     'aa53dcba601214d17ad405b7c291b7e8' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/multiply.php',
66.     '75c79eb1b25749b05a47976f32b0d8a2' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/divideby.php',
67.     '6ab8ad87a734f276a6bcd5a0fe1289be' => $vendorDir . '/markbaker/matrix/classes/src/o
    perations/divideinto.php',
68. );

```

application\third_party\vendor\composer\autoload_namespaces

```

1.  <?php
2.
3.  // autoload_namespaces.php @generated by Composer
4.
5.  $vendorDir = dirname(dirname(__FILE__));
6.  $baseDir = dirname($vendorDir);
7.
8.  return array(
9.  );

```

application\third_party\vendor\composer\autoload_psr4

```

1.  <?php
2.
3.  // autoload_psr4.php @generated by Composer
4.
5.  $vendorDir = dirname(dirname(__FILE__));
6.  $baseDir = dirname($vendorDir);
7.
8.  return array(
9.      'ZipStream\\' => array($vendorDir . '/maennchen/zipstream-php/src'),
10.     'Symfony\\Polyfill\\Mbstring\\' => array($vendorDir . '/symfony/polyfill-
    mbstring'),
11.     'Psr\\SimpleCache\\' => array($vendorDir . '/psr/simple-cache/src'),
12.     'Psr\\Http\\Message\\' => array($vendorDir . '/psr/http-
    factory/src', $vendorDir . '/psr/http-message/src'),
13.     'Psr\\Http\\Client\\' => array($vendorDir . '/psr/http-client/src'),

```

```

14.     'PhpOffice\\PhpSpreadsheet\\' => array($vendorDir . '/phpoffice/phpspreadsheet/src/
    PhpSpreadsheet'),
15.     'MyCLabs\\Enum\\' => array($vendorDir . '/myclabs/php-enum/src'),
16.     'Matrix\\' => array($vendorDir . '/markbaker/matrix/classes/src'),
17.     'Complex\\' => array($vendorDir . '/markbaker/complex/classes/src'),
18. );

```

application\third_party\vendor\composer\autoload_real

```

1.  <?php
2.
3.  // autoload_real.php @generated by Composer
4.
5.  class ComposerAutoloaderInit5bbff88ed03addcf6b3f7331b776e204
6.  {
7.      private static $loader;
8.
9.      public static function loadClassLoader($class)
10.     {
11.         if ('Composer\Autoload\ClassLoader' === $class) {
12.             require __DIR__ . '/ClassLoader.php';
13.         }
14.     }
15.
16.     public static function getLoader()
17.     {
18.         if (null !== self::$loader) {
19.             return self::$loader;
20.         }
21.
22.         spl_autoload_register(array('ComposerAutoloaderInit5bbff88ed03addcf6b3f7331b776
e204', 'loadClassLoader'), true, true);
23.         self::$loader = $loader = new \Composer\Autoload\ClassLoader();
24.         spl_autoload_unregister(array('ComposerAutoloaderInit5bbff88ed03addcf6b3f7331b7
76e204', 'loadClassLoader'));
25.
26.         $useStaticLoader = PHP_VERSION_ID >= 50600 && !defined('HHVM_VERSION') && (!fun
ction_exists('zend_loader_file_encoded') || !zend_loader_file_encoded());
27.         if ($useStaticLoader) {
28.             require_once __DIR__ . '/autoload_static.php';
29.
30.             call_user_func(\Composer\Autoload\ComposerStaticInit5bbff88ed03addcf6b3f733
1b776e204::getInitializer($loader));
31.         } else {
32.             $map = require __DIR__ . '/autoload_namespaces.php';
33.             foreach ($map as $namespace => $path) {
34.                 $loader->set($namespace, $path);
35.             }
36.
37.             $map = require __DIR__ . '/autoload_psr4.php';
38.             foreach ($map as $namespace => $path) {
39.                 $loader->setPsr4($namespace, $path);
40.             }
41.
42.             $classMap = require __DIR__ . '/autoload_classmap.php';
43.             if ($classMap) {
44.                 $loader->addClassMap($classMap);
45.             }

```

```

46.     }
47.
48.     $loader->register(true);
49.
50.     if ($useStaticLoader) {
51.         $includeFiles = Composer\Autoload\ComposerStaticInit5bbff88ed03addcf6b3f733
1b776e204::$files;
52.     } else {
53.         $includeFiles = require __DIR__ . '/autoload_files.php';
54.     }
55.     foreach ($includeFiles as $fileIdentifier => $file) {
56.         composerRequire5bbff88ed03addcf6b3f7331b776e204($fileIdentifier, $file);
57.     }
58.
59.     return $loader;
60. }
61. }
62.
63. function composerRequire5bbff88ed03addcf6b3f7331b776e204($fileIdentifier, $file)
64. {
65.     if (empty($GLOBALS['__composer_autoload_files'][$fileIdentifier])) {
66.         require $file;
67.
68.         $GLOBALS['__composer_autoload_files'][$fileIdentifier] = true;
69.     }
70. }

```

application\third_party\vendor\composer\autoload_static

```

1. <?php
2.
3. // autoload_static.php @generated by Composer
4.
5. namespace Composer\Autoload;
6.
7. class ComposerStaticInit5bbff88ed03addcf6b3f7331b776e204
8. {
9.     public static $files = array (
10.         '0e6d7bf4a5811bfa5cf40c5ccd6fae6a' => __DIR__ . '/../..' . '/symfony/polyfill-
mbstring/bootstrap.php',
11.         'abede361264e2ae69ec1eee813a101af' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/abs.php',
12.         '21a5860fbef5be28db5ddfb3cca67c4' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acos.php',
13.         '1546e3f9d127f2a9bb2d1b6c31c26ef1' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acosh.php',
14.         'd2516f7f4fba5ea5905f494b4a8262e0' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acot.php',
15.         '4511163d560956219b96882c0980b65e' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acoth.php',
16.         'c361f5616dc2a8da4fa3e137077cd4ea' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acsc.php',
17.         '02d68920fc98da71991ce569c91df0f6' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/acsch.php',
18.         '88e19525eae308b4a6aa3419364875d3' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/argument.php',
19.         '60e8e2d0827b58bfc904f13957e51849' => __DIR__ . '/../..' . '/markbaker/complex/cl
asses/src/functions/asec.php',

```

```
20. '13d2f040713999eab66c359b4d79871d' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/asech.php',
21. '838ab38beb32c68a79d3cd2c007d5a04' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/asin.php',
22. 'bb28eccd0f8f008333a1b3c163d604ac' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/asinh.php',
23. '9e483de83558c98f7d3feaa402c78cb3' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/atan.php',
24. '36b74b5b765ded91ee58c8ee3c0e85e3' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/atanh.php',
25. '05c15ee9510da7fd6bf6136f436500c0' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/conjugate.php',
26. 'd3208dfbce2505e370788f9f22f6785f' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/cos.php',
27. '141cf1fb3a3046f8b64534b0ebab33ca' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/cosh.php',
28. 'be660df75fd0dbe7fa7c03b7434b3294' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/cot.php',
29. '01e31ea298a51bc9e91517e3ce6b9e76' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/coth.php',
30. '803ddd97f7b1da68982a7b087c3476f6' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/csc.php',
31. '3001cdfd101ec3c32da34ee43c2e149b' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/csch.php',
32. '77b2d7629ef2a93fabb8c56754a91051' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/exp.php',
33. '4a4471296dec796c21d4f4b6552396a9' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/inverse.php',
34. 'c3e9897e1744b88deb56fcdc39d34d85' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/ln.php',
35. 'a83cacf2de942cff288de15a83afd26d' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/log2.php',
36. '6a861dacc9ee2f3061241d4c7772fa21' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/log10.php',
37. '4d2522d968c8ba78d6c13548a1b4200e' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/negative.php',
38. 'fd587ca933fc0447fa5ab4843bdd97f7' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/pow.php',
39. '383ef01c62028fc78cd4388082fce3c2' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/rho.php',
40. '150fbd1b95029dc47292da97ecab9375' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/sec.php',
41. '549abd9bae174286d660bdaa07407c68' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/sech.php',
42. '6bfbf5eaea6b17a0ed85cb21ba80370c' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/sin.php',
43. '22efe13f1a497b8e199540ae2d9dc59c' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/sinh.php',
44. 'e90135ab8e787795a509ed7147de207d' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/sqrt.php',
45. 'bb0a7923ffc6a90919cd64ec54ff06bc' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/tan.php',
46. '2d302f32ce0fd4e433dd91c5bb404a28' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/tanh.php',
47. '24dd4658a952171a4ee79218c4f9fd06' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/functions/theta.php',
48. 'e49b7876281d6f5bc39536dde96d1f4a' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/operations/add.php',
49. '47596e02b43cd6da7700134fd08f88cf' => __DIR__ . '/../' . '/markbaker/complex/cla
sses/src/operations/subtract.php',
```

```

50.         '883af48563631547925fa4c3b48ead07' => __DIR__ . '/../' . '/markbaker/complex/classes/src/operations/multiply.php',
51.         'f190e3308e6ca23234a2875edc985c03' => __DIR__ . '/../' . '/markbaker/complex/classes/src/operations/divideby.php',
52.         'ac9e33ce6841aa5bf5d16d465a2f03a7' => __DIR__ . '/../' . '/markbaker/complex/classes/src/operations/divideinto.php',
53.         '9d8e013a5160a09477beb8e44f8ae97b' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/adjoint.php',
54.         '6e78d1bdea6248d6aa117229efae50f2' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/antidiagonal.php',
55.         '4623d87924d94f5412fe5afbf1cef31d' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/cofactors.php',
56.         '901fd1f6950a637ca85f66b701a45e13' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/determinant.php',
57.         '83057abc0e4acc99ba80154ee5d02a49' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/diagonal.php',
58.         '07b7fd7a434451149b4fd477fca0ce06' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/identity.php',
59.         'c8d43b340583e07ae89f2a3baef2cf89' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/inverse.php',
60.         '499bb10ed7a3aee2ba4c09a31a85e8d1' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/minors.php',
61.         '1cad2e6414d652e8b1c64e8967f6f37d' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/trace.php',
62.         '95a7f134ac17161d07def442b3b737e8' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/functions/transpose.php',
63.         'b3a6bc628377118d4b4b8ba08d1eb949' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/add.php',
64.         '5fef6d0e407f3f8887266dfa4a6c534c' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/directsum.php',
65.         '684ba247e1385946e3babdaa054119de' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/subtract.php',
66.         'aa53dcba601214d17ad405b7c291b7e8' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/multiply.php',
67.         '75c79eb1b25749b05a47976f32b0d8a2' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/divideby.php',
68.         '6ab8ad87a734f276a6bcd5a0fe1289be' => __DIR__ . '/../' . '/markbaker/matrix/classes/src/operations/divideinto.php',
69.     );
70.
71.     public static $prefixLengthsPsr4 = array (
72.         'Z' =>
73.         array (
74.             'ZipStream\\' => 10,
75.         ),
76.         'S' =>
77.         array (
78.             'Symfony\\Polyfill\\Mbstring\\' => 26,
79.         ),
80.         'P' =>
81.         array (
82.             'Psr\\SimpleCache\\' => 16,
83.             'Psr\\Http\\Message\\' => 17,
84.             'Psr\\Http\\Client\\' => 16,
85.             'PhpOffice\\PhpSpreadsheet\\' => 25,
86.         ),
87.         'M' =>
88.         array (
89.             'MyCLabs\\Enum\\' => 13,
90.             'Matrix\\' => 7,
91.         ),

```

```

92.         'C' =>
93.         array (
94.             'Complex\\' => 8,
95.         ),
96.     );
97.
98.     public static $prefixDirsPsr4 = array (
99.         'ZipStream\\' =>
100.         array (
101.             0 => __DIR__ . '/../' . '/maennchen/zipstream-php/src',
102.         ),
103.         'Symfony\\Polyfill\\Mbstring\\' =>
104.         array (
105.             0 => __DIR__ . '/../' . '/symfony/polyfill-mbstring',
106.         ),
107.         'Psr\\SimpleCache\\' =>
108.         array (
109.             0 => __DIR__ . '/../' . '/psr/simple-cache/src',
110.         ),
111.         'Psr\\Http\\Message\\' =>
112.         array (
113.             0 => __DIR__ . '/../' . '/psr/http-factory/src',
114.             1 => __DIR__ . '/../' . '/psr/http-message/src',
115.         ),
116.         'Psr\\Http\\Client\\' =>
117.         array (
118.             0 => __DIR__ . '/../' . '/psr/http-client/src',
119.         ),
120.         'PhpOffice\\PhpSpreadsheet\\' =>
121.         array (
122.             0 => __DIR__ . '/../' . '/phpoffice/phpspreadsheet/src/PhpSpreadsheet',
123.         ),
124.         'MyCLabs\\Enum\\' =>
125.         array (
126.             0 => __DIR__ . '/../' . '/myclabs/php-enum/src',
127.         ),
128.         'Matrix\\' =>
129.         array (
130.             0 => __DIR__ . '/../' . '/markbaker/matrix/classes/src',
131.         ),
132.         'Complex\\' =>
133.         array (
134.             0 => __DIR__ . '/../' . '/markbaker/complex/classes/src',
135.         ),
136.     );
137.
138.     public static function getInitializer(ClassLoader $loader)
139.     {
140.         return \Closure::bind(function () use ($loader) {
141.             $loader->
142.                 >prefixLengthsPsr4 = ComposerStaticInit5bbff88ed03addcf6b3f7331b776e204::$prefixLengthsPsr4;
143.             $loader->
144.                 >prefixDirsPsr4 = ComposerStaticInit5bbff88ed03addcf6b3f7331b776e204::$prefixDirsPsr4;
145.         }, null, ClassLoader::class);
146.     }

```

application\third_party\vendor\composer\ClassLoader

```
1. <?php
2.
3. /*
4.  * This file is part of Composer.
5.  *
6.  * (c) Nils Adermann <naderman@naderman.de>
7.  *     Jordi Boggiano <j.boggiano@seld.be>
8.  *
9.  * For the full copyright and license information, please view the LICENSE
10. * file that was distributed with this source code.
11. */
12.
13. namespace Composer\Autoload;
14.
15. /**
16.  * ClassLoader implements a PSR-0, PSR-4 and classmap class loader.
17.  *
18.  *     $loader = new \Composer\Autoload\ClassLoader();
19.  *
20.  *     // register classes with namespaces
21.  *     $loader->add('Symfony\Component', __DIR__.'/component');
22.  *     $loader->add('Symfony',       __DIR__.'/framework');
23.  *
24.  *     // activate the autoloader
25.  *     $loader->register();
26.  *
27.  *     // to enable searching the include path (eg. for PEAR packages)
28.  *     $loader->setUseIncludePath(true);
29.  *
30.  * In this example, if you try to use a class in the Symfony\Component
31.  * namespace or one of its children (Symfony\Component\Console for instance),
32.  * the autoloader will first look for the class under the component/
33.  * directory, and it will then fallback to the framework/ directory if not
34.  * found before giving up.
35.  *
36.  * This class is loosely based on the Symfony UniversalClassLoader.
37.  *
38.  * @author Fabien Potencier <fabien@symfony.com>
39.  * @author Jordi Boggiano <j.boggiano@seld.be>
40.  * @see    http://www.php-fig.org/psr/psr-0/
41.  * @see    http://www.php-fig.org/psr/psr-4/
42.  */
43. class ClassLoader
44. {
45.     // PSR-4
46.     private $prefixLengthsPsr4 = array();
47.     private $prefixDirsPsr4 = array();
48.     private $fallbackDirsPsr4 = array();
49.
50.     // PSR-0
51.     private $prefixesPsr0 = array();
52.     private $fallbackDirsPsr0 = array();
53.
54.     private $useIncludePath = false;
55.     private $classMap = array();
56.     private $classMapAuthoritative = false;
```

```

57.     private $missingClasses = array();
58.     private $apcuPrefix;
59.
60.     public function getPrefixes()
61.     {
62.         if (!empty($this->prefixesPsr0)) {
63.             return call_user_func_array('array_merge', $this->prefixesPsr0);
64.         }
65.
66.         return array();
67.     }
68.
69.     public function getPrefixesPsr4()
70.     {
71.         return $this->prefixDirsPsr4;
72.     }
73.
74.     public function getFallbackDirs()
75.     {
76.         return $this->fallbackDirsPsr0;
77.     }
78.
79.     public function getFallbackDirsPsr4()
80.     {
81.         return $this->fallbackDirsPsr4;
82.     }
83.
84.     public function getClassMap()
85.     {
86.         return $this->classMap;
87.     }
88.
89.     /**
90.      * @param array $classMap Class to filename map
91.      */
92.     public function addClassMap(array $classMap)
93.     {
94.         if ($this->classMap) {
95.             $this->classMap = array_merge($this->classMap, $classMap);
96.         } else {
97.             $this->classMap = $classMap;
98.         }
99.     }
100.
101.     /**
102.      * Registers a set of PSR-0 directories for a given prefix, either
103.      * appending or prepending to the ones previously set for this prefix.
104.      *
105.      * @param string $prefix The prefix
106.      * @param array|string $paths The PSR-0 root directories
107.      * @param bool $prepend Whether to prepend the directories
108.      */
109.     public function add($prefix, $paths, $prepend = false)
110.     {
111.         if (!$prefix) {
112.             if ($prepend) {
113.                 $this->fallbackDirsPsr0 = array_merge(
114.                     (array) $paths,
115.                     $this->fallbackDirsPsr0
116.                 );
117.             } else {

```

```

118.         $this->fallbackDirsPsr0 = array_merge(
119.             $this->fallbackDirsPsr0,
120.             (array) $paths
121.         );
122.     }
123.
124.     return;
125. }
126.
127. $first = $prefix[0];
128. if (!isset($this->prefixesPsr0[$first][$prefix])) {
129.     $this->prefixesPsr0[$first][$prefix] = (array) $paths;
130.
131.     return;
132. }
133. if ($prepend) {
134.     $this->prefixesPsr0[$first][$prefix] = array_merge(
135.         (array) $paths,
136.         $this->prefixesPsr0[$first][$prefix]
137.     );
138. } else {
139.     $this->prefixesPsr0[$first][$prefix] = array_merge(
140.         $this->prefixesPsr0[$first][$prefix],
141.         (array) $paths
142.     );
143. }
144. }
145.
146. /**
147.  * Registers a set of PSR-4 directories for a given namespace, either
148.  * appending or prepending to the ones previously set for this namespace.
149.  *
150.  * @param string      $prefix The prefix/namespace, with trailing '\\'
151.  * @param array|string $paths  The PSR-4 base directories
152.  * @param bool        $prepend Whether to prepend the directories
153.  *
154.  * @throws \InvalidArgumentException
155.  */
156. public function addPsr4($prefix, $paths, $prepend = false)
157. {
158.     if (!$prefix) {
159.         // Register directories for the root namespace.
160.         if ($prepend) {
161.             $this->fallbackDirsPsr4 = array_merge(
162.                 (array) $paths,
163.                 $this->fallbackDirsPsr4
164.             );
165.         } else {
166.             $this->fallbackDirsPsr4 = array_merge(
167.                 $this->fallbackDirsPsr4,
168.                 (array) $paths
169.             );
170.         }
171.     } elseif (!isset($this->prefixDirsPsr4[$prefix])) {
172.         // Register directories for a new namespace.
173.         $length = strlen($prefix);
174.         if ('\\' !== $prefix[$length - 1]) {
175.             throw new \InvalidArgumentException("A non-empty PSR-
176. 4 prefix must end with a namespace separator.");
177.         }
178.         $this->prefixLengthsPsr4[$prefix[0]][$prefix] = $length;

```

```

178.         $this->prefixDirsPsr4[$prefix] = (array) $paths;
179.     } elseif ($prepend) {
180.         // Prepend directories for an already registered namespace.
181.         $this->prefixDirsPsr4[$prefix] = array_merge(
182.             (array) $paths,
183.             $this->prefixDirsPsr4[$prefix]
184.         );
185.     } else {
186.         // Append directories for an already registered namespace.
187.         $this->prefixDirsPsr4[$prefix] = array_merge(
188.             $this->prefixDirsPsr4[$prefix],
189.             (array) $paths
190.         );
191.     }
192. }
193.
194. /**
195.  * Registers a set of PSR-0 directories for a given prefix,
196.  * replacing any others previously set for this prefix.
197.  *
198.  * @param string $prefix The prefix
199.  * @param array|string $paths The PSR-0 base directories
200.  */
201. public function set($prefix, $paths)
202. {
203.     if (!$prefix) {
204.         $this->fallbackDirsPsr0 = (array) $paths;
205.     } else {
206.         $this->prefixesPsr0[$prefix[0]][$prefix] = (array) $paths;
207.     }
208. }
209.
210. /**
211.  * Registers a set of PSR-4 directories for a given namespace,
212.  * replacing any others previously set for this namespace.
213.  *
214.  * @param string $prefix The prefix/namespace, with trailing '\\'
215.  * @param array|string $paths The PSR-4 base directories
216.  *
217.  * @throws \InvalidArgumentException
218.  */
219. public function setPsr4($prefix, $paths)
220. {
221.     if (!$prefix) {
222.         $this->fallbackDirsPsr4 = (array) $paths;
223.     } else {
224.         $length = strlen($prefix);
225.         if ('\\' !== $prefix[$length - 1]) {
226.             throw new \InvalidArgumentException("A non-empty PSR-
227. 4 prefix must end with a namespace separator.");
228.         }
229.         $this->prefixLengthsPsr4[$prefix[0]][$prefix] = $length;
230.         $this->prefixDirsPsr4[$prefix] = (array) $paths;
231.     }
232. }
233.
234. /**
235.  * Turns on searching the include path for class files.
236.  *
237.  * @param bool $useIncludePath
238.  */

```

```

238.     public function setUseIncludePath($useIncludePath)
239.     {
240.         $this->useIncludePath = $useIncludePath;
241.     }
242.
243.     /**
244.      * Can be used to check if the autoloader uses the include path to check
245.      * for classes.
246.      *
247.      * @return bool
248.      */
249.     public function getUseIncludePath()
250.     {
251.         return $this->useIncludePath;
252.     }
253.
254.     /**
255.      * Turns off searching the prefix and fallback directories for classes
256.      * that have not been registered with the class map.
257.      *
258.      * @param bool $classMapAuthoritative
259.      */
260.     public function setClassMapAuthoritative($classMapAuthoritative)
261.     {
262.         $this->classMapAuthoritative = $classMapAuthoritative;
263.     }
264.
265.     /**
266.      * Should class lookup fail if not found in the current class map?
267.      *
268.      * @return bool
269.      */
270.     public function isClassMapAuthoritative()
271.     {
272.         return $this->classMapAuthoritative;
273.     }
274.
275.     /**
276.      * APCu prefix to use to cache found/not-
277.      * found classes, if the extension is enabled.
278.      *
279.      * @param string|null $apcuPrefix
280.      */
281.     public function setApcuPrefix($apcuPrefix)
282.     {
283.         $this->apcuPrefix = function_exists('apcu_fetch') && filter_var(ini_get('apc.enabled'), FILTER_VALIDATE_BOOLEAN) ? $apcuPrefix : null;
284.     }
285.
286.     /**
287.      * The APCu prefix in use, or null if APCu caching is not enabled.
288.      *
289.      * @return string|null
290.      */
291.     public function getApcuPrefix()
292.     {
293.         return $this->apcuPrefix;
294.     }
295.
296.     /**

```

```

296.         * Registers this instance as an autoloader.
297.         *
298.         * @param bool $prepend Whether to prepend the autoloader or not
299.         */
300.     public function register($prepend = false)
301.     {
302.         spl_autoload_register(array($this, 'loadClass'), true, $prepend);
303.     }
304.
305.     /**
306.     * Unregisters this instance as an autoloader.
307.     */
308.     public function unregister()
309.     {
310.         spl_autoload_unregister(array($this, 'loadClass'));
311.     }
312.
313.     /**
314.     * Loads the given class or interface.
315.     *
316.     * @param string $class The name of the class
317.     * @return bool|null True if loaded, null otherwise
318.     */
319.     public function loadClass($class)
320.     {
321.         if ($file = $this->findFile($class)) {
322.             includeFile($file);
323.
324.             return true;
325.         }
326.     }
327.
328.     /**
329.     * Finds the path to the file where the class is defined.
330.     *
331.     * @param string $class The name of the class
332.     *
333.     * @return string|false The path if found, false otherwise
334.     */
335.     public function findFile($class)
336.     {
337.         // class map lookup
338.         if (isset($this->classMap[$class])) {
339.             return $this->classMap[$class];
340.         }
341.         if ($this->classMapAuthoritative || isset($this->missingClasses[$class])) {
342.             return false;
343.         }
344.         if (null !== $this->apcuPrefix) {
345.             $file = apcu_fetch($this->apcuPrefix.$class, $hit);
346.             if ($hit) {
347.                 return $file;
348.             }
349.         }
350.
351.         $file = $this->findFileWithExtension($class, '.php');
352.
353.         // Search for Hack files if we are running on HHVM
354.         if (false === $file && defined('HHVM_VERSION')) {
355.             $file = $this->findFileWithExtension($class, '.hh');

```

```

356.         }
357.
358.         if (null !== $this->apcuPrefix) {
359.             apcu_add($this->apcuPrefix.$class, $file);
360.         }
361.
362.         if (false === $file) {
363.             // Remember that this class does not exist.
364.             $this->missingClasses[$class] = true;
365.         }
366.
367.         return $file;
368.     }
369.
370.     private function findFileWithExtension($class, $ext)
371.     {
372.         // PSR-4 lookup
373.         $logicalPathPsr4 = strrtr($class, '\\', DIRECTORY_SEPARATOR) . $ext;
374.
375.         $first = $class[0];
376.         if (isset($this->prefixLengthsPsr4[$first])) {
377.             $subPath = $class;
378.             while (false !== $lastPos = strrpos($subPath, '\\')) {
379.                 $subPath = substr($subPath, 0, $lastPos);
380.                 $search = $subPath . '\\';
381.                 if (isset($this->prefixDirsPsr4[$search])) {
382.                     $pathEnd = DIRECTORY_SEPARATOR . substr($logicalPathPsr4, $l
astPos + 1);
383.                     foreach ($this->prefixDirsPsr4[$search] as $dir) {
384.                         if (file_exists($file = $dir . $pathEnd)) {
385.                             return $file;
386.                         }
387.                     }
388.                 }
389.             }
390.         }
391.
392.         // PSR-4 fallback dirs
393.         foreach ($this->fallbackDirsPsr4 as $dir) {
394.             if (file_exists($file = $dir . DIRECTORY_SEPARATOR . $logicalPathPsr
4)) {
395.                 return $file;
396.             }
397.         }
398.
399.         // PSR-0 lookup
400.         if (false !== $pos = strrpos($class, '\\')) {
401.             // namespaced class name
402.             $logicalPathPsr0 = substr($logicalPathPsr4, 0, $pos + 1)
. strrtr(substr($logicalPathPsr4, $pos + 1), '_', DIRECTORY_SEPAR
ATOR);
403.         } else {
404.             // PEAR-like class name
405.             $logicalPathPsr0 = strrtr($class, '_', DIRECTORY_SEPARATOR) . $ext;
406.         }
407.
408.         if (isset($this->prefixesPsr0[$first])) {
409.             foreach ($this->prefixesPsr0[$first] as $prefix => $dirs) {
410.                 if (0 === strpos($class, $prefix)) {
411.                     foreach ($dirs as $dir) {
412.

```

```

413.                                     if (file_exists($file = $dir . DIRECTORY_SEPARATOR . $lo
gicalPathPsr0)) {
414.                                         return $file;
415.                                     }
416.                                 }
417.                            }
418.                        }
419.                    }
420.
421.                // PSR-0 fallback dirs
422.                foreach ($this->fallbackDirsPsr0 as $dir) {
423.                    if (file_exists($file = $dir . DIRECTORY_SEPARATOR . $logicalPathPsr
0)) {
424.                        return $file;
425.                    }
426.                }
427.
428.                // PSR-0 include paths.
429.                if ($this-
>useIncludePath && $file = stream_resolve_include_path($logicalPathPsr0)) {
430.                    return $file;
431.                }
432.
433.                return false;
434.            }
435.        }
436.
437.        /**
438.         * Scope isolated include.
439.         *
440.         * Prevents access to $this/self from included files.
441.         */
442.        function includeFile($file)
443.        {
444.            include $file;
445.        }

```

application\views\quiz\ sheet

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.
4.         <div class="row">
5.             <div class="col-sm-9">
6.                 <form action="" method="post" id="ujian">
7.                     <input type="hidden" name="id" id="id" value="<?= $id_test ?>">
8.                     <div class="box box-success">
9.                         <div class="box-header with-border">
10.                            <h3 class="box-title"><span class="badge bg-
blue">Question #<span id="soalke"></span> </span></h3>
11.                            <div class="box-tools pull-right">
12.                                <span class="badge bg-
red">Time <span class="sisawaktu" data-time="<?= $soal['end_date'] ?>"></span></span>
13.                                <button type="button" class="btn btn-box-tool" data-
widget="collapse"><i class="fa fa-minus"></i>
14.                                </button>
15.                            </div>
16.                        </div>
17.                        <div class="box-body">
18.                            <?= $html ?>

```

```

19.             </div>
20.             <div class="box-footer text-center">
21.                 <a class="action back btn btn-
info" rel="0" onclick="return back();"><i class="glyphicon glyphicon-chevron-
left"></i> Back</a>
22.                 <a class="action next btn btn-
info" rel="2" onclick="return next();"><i class="glyphicon glyphicon-chevron-
right"></i> Next</a>
23.                 <a class="selesai action submit btn btn-
danger" onclick="return simpan_akhir();"><i class="glyphicon glyphicon-
stop"></i> Selesai</a>
24.                 <input type="hidden" name="jml_soal" id="jml_soal" value="<
?>= $no; ?>">
25.             </div>
26.         </div>
27.     </form>
28. </div>
29.     <div class="col-sm-3">
30.         <div class="box box-success">
31.             <div class="box-header with-border">
32.                 <h3 class="box-title">Question Navigation</h3>
33.                 <div class="box-tools pull-right">
34.                     <button type="button" class="btn btn-box-tool" data-
widget="collapse"><i class="fa fa-minus"></i>
35.                 </button>
36.             </div>
37.         </div>
38.         <div class="box-body text-center" id="tampil_jawaban">
39.         </div>
40.     </div>
41. </div>
42. </div>
43. </div>
44. </div>
45.
46.
47. <script type="text/javascript">
48.     var base_url = "<?>= base_url(); ?>";
49.     var id_tes = "<?>= $id_test; ?>";
50.     var widget = $(''.step');
51.     var total_widget = widget.length;
52.     console.log(base_url)
53. </script>
54.
55. <script src="<?>= base_url() ?>assets/dist/js/app/ujian/sheet.js"></script>

```

application\views\views_admin\setting

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="page-header">
4.             <div class="row align-items-end">
5.                 <div class="col-lg-8">
6.                     <div class="page-header-title">
7.                         <i class="ik ik-settings bg-blue"></i>
8.                     <div class="d-inline">
9.                         <h5>Setting</h5>
10.                    <span>Access by: Administrator</span>

```

```

11.         </div>
12.     </div>
13. </div>
14. <div class="col-lg-4">
15.     <nav class="breadcrumb-container" aria-label="breadcrumb">
16.         <ol class="breadcrumb">
17.             <li class="breadcrumb-item">
18.                 <a href="{%= base_url('course'); %}"><i class="ik ik-
home"></i></a>
19.             </li>
20.             <li class="breadcrumb-item active" aria-
current="page">Setting</li>
21.         </ol>
22.     </nav>
23. </div>
24. </div>
25. </div>
26. <div class="row">
27.     <div class="col-md-12">
28.         <div class="card">
29.             <div class="card-header">
30.                 <h3>Setting</h3>
31.             </div>
32.             <div class="card-block">
33.                 <div class="alert alert-danger" role="alert">
34.                     <strong>If you press the Reset button, all data related to
the course will be lost</strong>
35.                 </div>
36.                 <a href="{%= base_url('admin/reset'); %}" class="btn btn-
danger">RESET</a>
37.             </div>
38.         </div>
39.
40.     </div>
41. </div>
42. </div>
43. </div>

```

application\views\views_auth\ blocked

```

1. <!doctype html>
2. <html lang="en">
3.
4. <head>
5.     <link rel="icon" href="{%= base_url(); %}vendor/src/img/rekind.png" />
6.     <meta charset="utf-8">
7.     <title>Page Not Found</title>
8.     <meta name="viewport" content="width=device-width, initial-scale=1">
9.     <link href="https://fonts.googleapis.com/css?family=Nunito+Sans:300,400,600,700,800"
rel="stylesheet">
10.    <style>
11.        *,
12.        body {
13.            -webkit-font-smoothing: antialiased;
14.            text-rendering: optimizeLegibility;
15.            -moz-osx-font-smoothing: grayscale;
16.        }
17.
18.        * {
19.            line-height: 1.2;

```

```
20.         margin: 0;
21.     }
22.
23.     html {
24.         color: #888;
25.         display: table;
26.         font-family: 'Nunito Sans', sans-serif;
27.         height: 100%;
28.         text-align: center;
29.         width: 100%;
30.     }
31.
32.     .h1,
33.     .h2,
34.     .h3,
35.     .h4,
36.     .h5,
37.     .h6,
38.     h1,
39.     h2,
40.     h3,
41.     h4,
42.     h5,
43.     h6 {
44.         font-family: 'Nunito Sans', sans-serif;
45.     }
46.
47.     body {
48.         display: table-cell;
49.         vertical-align: middle;
50.         margin: 2em auto;
51.     }
52.
53.     h1 {
54.         color: #ef4153;
55.         text-
shadow: rgba(235, 82, 93, 0.3) 5px 1px, rgba(235, 82, 93, 0.2) 10px 3px;
56.         font-size: 150px;
57.         font-weight: 800;
58.         margin-bottom: 10px;
59.         letter-spacing: 2px;
60.     }
61.
62.     h4 {
63.         color: #4a5361;
64.         text-transform: capitalize;
65.         font-size: 28px;
66.     }
67.
68.     p {
69.         margin: 0 auto;
70.         max-width: 790px;
71.         margin-top: 20px;
72.         color: #666;
73.         margin-bottom: 10px;
74.         font-size: 15px;
75.         line-height: 20px;
76.     }
77.
78.     a {
79.         display: inline-block;
```

```

80.         padding: 8px 15px;
81.         background-color: #ef4153;
82.         color: #fff;
83.         text-decoration: none;
84.         border-radius: 4px;
85.         margin-top: 20px;
86.     }
87.
88.     @media only screen and (max-width: 280px) {
89.
90.         body,
91.         p {
92.             width: 95%;
93.         }
94.
95.         h1 {
96.             font-size: 1.5em;
97.             margin: 0 0 0.3em;
98.         }
99.
100.    }
101.    </style>
102. </head>
103.
104. <body>
105.     <h1>404</h1>
106.     <h4>Page Not Found</h4>
107.     <p>It's looking like you may have taken a wrong turn. Don't worry... it happens to the best of us. Here's a little tip that might help you get back on track.</p>
108.     <a href="<%= base_url('course/mycourse') ?>">Return to Home</a>
109. </body>
110.
111. </html>
112. <!--
- IE needs 512+ bytes: http://blogs.msdn.com/b/ieinternals/archive/2010/08/19/http-error-pages-in-internet-explorer.aspx -->

```

application\views\views_auth\forget_password

```

1. <!doctype html>
2. <html class="no-js" lang="en">
3.
4. <head>
5.     <meta charset="utf-8">
6.     <meta http-equiv="x-ua-compatible" content="ie=edge">
7.     <title>Forgot Password | E-Learning Rekind</title>
8.     <meta name="description" content="">
9.     <meta name="keywords" content="">
10.    <meta name="viewport" content="width=device-width, initial-scale=1">
11.
12.    <link rel="icon" href="<%= base_url(); ?>vendor/src/img/rekind.png" />
13.
14.    <link href="https://fonts.googleapis.com/css?family=Nunito+Sans:300,400,600,700,800" rel="stylesheet">
15.
16.    <link rel="stylesheet" href="<%= base_url('vendor/') ?>plugins/bootstrap/dist/css/bootstrap.min.css">

```

```

17. <link rel="stylesheet" href="<?= base_url('vendor/') ?>plugins/fontawesome-
    free/css/all.min.css">
18. <link rel="stylesheet" href="<?= base_url('vendor/') ?>plugins/ionicons/dist/css/ion
    icons.min.css">
19. <link rel="stylesheet" href="<?= base_url('vendor/') ?>plugins/icon-
    kit/dist/css/iconkit.min.css">
20. <link rel="stylesheet" href="<?= base_url('vendor/') ?>plugins/perfect-
    scrollbar/css/perfect-scrollbar.css">
21. <link rel="stylesheet" href="<?= base_url('vendor/') ?>dist/css/theme.min.css">
22. <link rel="stylesheet" href="<?= base_url('vendor/') ?>dist/css/style.css">
23. <script src="<?= base_url('vendor/') ?>src/js/vendor/modernizr-
    2.8.3.min.js"></script>
24. </head>
25.
26. <body>
27. <!--[if lt IE 8]>
28. <p class="browserupgrade">You are using an <strong>outdated</strong> browse
    r. Please <a href="http://browsehappy.com/">upgrade your browser</a> to improve your ex
    perience.</p>
29. <![endif]-->
30. <div class="auth-wrapper">
31. <div class="container-fluid h-100">
32. <div class="flash-data" data-flashdata="<?= $this->session-
    >flashdata('message'); ?>"></div>
33. <div class="flash-data-icon" data-icon="<?= $this->session-
    >flashdata('icon'); ?>"></div>
34. <div class="flash-data-text" data-text="<?= $this->session-
    >flashdata('text'); ?>"></div>
35. <div class="row flex-row h-100 bg-white">
36. <div class="col-xl-8 col-lg-6 col-md-5 p-0 d-md-block d-lg-block d-sm-
    none d-none">
37. <div class="lavalite-bg" style="background-
    image: url('<?= base_url('vendor/') ?>img/auth/login-bg-rekind.jpg')">
38. <div class="bg-overlay"></div>
39. </div>
40. </div>
41. <div class="col-xl-4 col-lg-6 col-md-7 my-auto p-0">
42. <div class="authentication-form mx-auto">
43. <div class="logo-centered">
44. <a href="..../index.html"></a>
45. </div>
46. <h3>Forget Password</h3>
47. <p>Happy to see you again!</p>
48. <form action="<?= base_url('auth/forgot') ?>" method="post">
49. <div class="form-group">
50. <input type="text" name="email" class="form-
    control" placeholder="Email">
51. <i class="ik ik-mail"></i>
52. <small class="text-
    danger"><?= form_error('email'); ?></small>
53. </div>
54. <div class="row">
55. <div class="col text-right">
56. <a href="<?= base_url('auth') ?>">Back to Login</a>
57. </div>
58. </div>
59. <div class="sign-btn text-center">
60. <button type="submit" class="btn btn-
    primary">Submit</button>

```

```

61.             </div>
62.         </form>
63.     </div>
64. </div>
65. </div>
66. </div>
67. </div>
68. <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
69. <script>
70.     window.jQuery || document.write('<script src="<%= base_url('vendor/') ?>src/js/
    vendor/jquery-3.3.1.min.js"></script>')
71. </script>
72. <script src="<%= base_url('vendor/') ?>plugins/popper.js/dist/umd/popper.min.js"></
    script>
73. <script src="<%= base_url('vendor/') ?>plugins/bootstrap/dist/js/bootstrap.min.js">
    </script>
74. <script src="<%= base_url('vendor/') ?>plugins/perfect-scrollbar/dist/perfect-
    scrollbar.min.js"></script>
75. <script src="<%= base_url('vendor/') ?>plugins/screenfull/dist/screenfull.js"></scr
    ipt>
76. <script src="<%= base_url('vendor/') ?>dist/js/theme.js"></script>
77. <script src="<%= base_url('vendor/') ?>sweetalert/js/sweetalert2.all.min.js"></scri
    pt>
78. <script src="<%= base_url() ?>js/script.js"></script>
79. <!-- Google Analytics: change UA-XXXXX-X to be your site's ID. -->
80. <script>
81.     (function(b, o, i, l, e, r) {
82.         b.GoogleAnalyticsObject = l;
83.         b[l] || (b[l] =
84.             function() {
85.                 (b[l].q = b[l].q || []).push(arguments)
86.             });
87.         b[l].l = +new Date;
88.         e = o.createElement(i);
89.         r = o.getElementsByTagName(i)[0];
90.         e.src = 'https://www.google-analytics.com/analytics.js';
91.         r.parentNode.insertBefore(e, r)
92.     })(window, document, 'script', 'ga');
93.     ga('create', 'UA-XXXXX-X', 'auto');
94.     ga('send', 'pageview');
95. </script>
96. </body>
97.
98. </html>

```

application\views\views_auth\login

```

1. <!doctype html>
2. <html class="no-js" lang="en">
3.
4. <head>
5.     <meta charset="utf-8">
6.     <meta http-equiv="x-ua-compatible" content="ie=edge">
7.     <title>Login | E-Learning Rekind</title>
8.     <meta name="description" content="">
9.     <meta name="keywords" content="">
10.    <meta name="viewport" content="width=device-width, initial-scale=1">
11.

```



```

55.         <input type="password" name="password" class="form-
control" placeholder="Password">
56.         <?= form_error('password', '<small class="text-
danger">', '</small>'); ?>
57.         <i class="ik ik-lock"></i>
58.     </div>
59.     <div class="row">
60.         <div class="col text-right">
61.             <a href="<?= base_url('auth/forgot') ?>">Forgot Pas
sword ?</a>
62.         </div>
63.     </div>
64.     <div class="sign-btn text-center">
65.         <button type="submit" class="btn btn-
primary">Sign In</button>
66.     </div>
67. </form>
68. </div>
69. </div>
70. </div>
71. </div>
72. </div>
73. <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
74. <script>
75.     window.jQuery || document.write('<script src="<?= base_url('vendor/') ?>src/js/
vendor/jquery-3.3.1.min.js"></script>')
76. </script>
77. <script src="<?= base_url('vendor/') ?>plugins/popper.js/dist/umd/popper.min.js"></
script>
78. <script src="<?= base_url('vendor/') ?>plugins/bootstrap/dist/js/bootstrap.min.js">
</script>
79. <script src="<?= base_url('vendor/') ?>plugins/perfect-scrollbar/dist/perfect-
scrollbar.min.js"></script>
80. <script src="<?= base_url('vendor/') ?>plugins/screenfull/dist/screenfull.js"></scr
ipt>
81. <script src="<?= base_url('vendor/') ?>dist/js/theme.js"></script>
82. <script src="<?= base_url('vendor/') ?>sweetalert2.js/sweetalert2.all.min.js"></scri
pt>
83. <script src="<?= base_url() ?>js/script.js"></script>
84. <!-- Google Analytics: change UA-XXXXX-X to be your site's ID. -->
85. <script>
86.     (function(b, o, i, l, e, r) {
87.         b.GoogleAnalyticsObject = l;
88.         b[l] || (b[l] =
89.             function() {
90.                 (b[l].q = b[l].q || []).push(arguments)
91.             });
92.         b[l].l = +new Date;
93.         e = o.createElement(i);
94.         r = o.getElementsByTagName(i)[0];
95.         e.src = 'https://www.google-analytics.com/analytics.js';
96.         r.parentNode.insertBefore(e, r)
97.     }(window, document, 'script', 'ga'));
98.     ga('create', 'UA-XXXXX-X', 'auto');
99.     ga('send', 'pageview');
100. </script>
101. </body>
102.
103. </html>

```



```

44.         <a href="../index.html"></a>
45.     </div>
46.     <h3>Reset Password</h3>
47.     <p>Hallo <strong><?= $this->session-
>userdata('email') ?></strong>, Please enter new password!</p>
48.     <form action="<?= base_url('auth/changepassword') ?>" method="p
ost">
49.         <div class="form-group">
50.             <input type="password" name="password1" class="form-
control" placeholder="New Password">
51.             <i class="ik ik-lock"></i>
52.             <small class="text-
danger"><?= form_error('password1') ?></small>
53.         </div>
54.         <div class="form-group">
55.             <input type="password" name="password2" class="form-
control" placeholder="Repeat Password">
56.             <i class="ik ik-lock"></i>
57.         </div>
58.         <div class="row">
59.             <div class="col text-right">
60.                 <a href="<?= base_url('auth') ?>">Back to Login</a>
61.             </div>
62.         </div>
63.         <div class="sign-btn text-center">
64.             <button type="submit" class="btn btn-
primary">Submit</button>
65.         </div>
66.     </form>
67. </div>
68. </div>
69. </div>
70. </div>
71. </div>
72. <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
73. <script>
74.     window.jQuery || document.write('<script src="<?= base_url('vendor/') ?>src/js/
vendor/jquery-3.3.1.min.js"></script>')
75. </script>
76. <script src="<?= base_url('vendor/') ?>plugins/popper.js/dist/umd/popper.min.js"></
script>
77. <script src="<?= base_url('vendor/') ?>plugins/bootstrap/dist/js/bootstrap.min.js">
</script>
78. <script src="<?= base_url('vendor/') ?>plugins/perfect-scrollbar/dist/perfect-
scrollbar.min.js"></script>
79. <script src="<?= base_url('vendor/') ?>plugins/screenfull/dist/screenfull.js"></scr
ipt>
80. <script src="<?= base_url('vendor/') ?>dist/js/theme.js"></script>
81. <script src="<?= base_url('vendor/') ?>sweetalert/js/sweetalert2.all.min.js"></scri
pt>
82. <script src="<?= base_url() ?>js/script.js"></script>
83. <!-- Google Analytics: change UA-XXXXX-X to be your site's ID. -->
84. <script>
85.     (function(b, o, i, l, e, r) {
86.         b.GoogleAnalyticsObject = l;
87.         b[l] || (b[l] =
88.             function() {
89.                 (b[l].q = b[l].q || []).push(arguments)
90.             });

```

```

91.         b[1].l = +new Date;
92.         e = o.createElement(i);
93.         r = o.getElementsByTagName(i)[0];
94.         e.src = 'https://www.google-analytics.com/analytics.js';
95.         r.parentNode.insertBefore(e, r)
96.     }(window, document, 'script', 'ga'));
97.     ga('create', 'UA-XXXXX-X', 'auto');
98.     ga('send', 'pageview');
99. </script>
100. </body>
101.
102. </html>

```

application\views\views_chating\ chating

```

1. <?php foreach ($chat as $c) : ?>
2.     <?php
3.         if ($c['user_id_send'] == $me['id']) : ?>
4.             <div class="outgoing_msg">
5.                 <div class="sent_msg">
6.                     <p><?=$c['message'] ?></p>
7.                     <span class="time_date"><?=$c['created_at'] ?></span> .
8.                     ' | ' . date('F d', strtotime($c['created_at'])) ?></span>
9.                 </div>
10.             </div>
11.         <?php else :
12.             $picture = $this->UM->getUserById($youId)['picture'];
13.             ?>
14.             <div class="incoming_msg">
15.                 <div class="incoming_msg_img">  </div>
17.                 <div class="received_msg">
18.                     <div class="received_withd_msg">
19.                         <p><?=$c['message'] ?></p>
20.                         <span class="time_date"><?=$c['created_at'] ?></span> .
21.                         ' | ' . date('F d', strtotime($c['created_at'])) ?></span>
22.                     </div>
23.                 </div>
24.             </div>
25.         <?php endif; ?>
26.     <?php endforeach; ?>

```

application\views\views_chating\ home

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="page-header">
4.             <div class="row align-items-end">
5.                 <div class="col-lg-8">
6.                     <div class="page-header-title">
7.                         <i class="ik ik-message-square bg-blue"></i>
8.                         <div class="d-inline">
9.                             <h5>Chating</h5>
10.                            <span>Access by: User</span>
11.                        </div>
12.                    </div>
13.                </div>
14.                <div class="col-lg-4">
15.                    <nav class="breadcrumb-container" aria-label="breadcrumb">

```

```

16.         <ol class="breadcrumb">
17.             <li class="breadcrumb-item">
18.                 <a href="<? = base_url('course') ?>"><i class="ik ik-
home"></i></a>
19.             </li>
20.             <li class="breadcrumb-item">
21.                 <a href="<? = base_url('course/open/') . $this->uri-
>segment(3, 0); ?>">My Course</a>
22.             </li>
23.             <li class="breadcrumb-item active" aria-
current="page">Chating</li>
24.         </ol>
25.     </nav>
26. </div>
27. </div>
28. </div>
29. <div class="row">
30.     <div class="col-md-5">
31.         <div class="inbox_people">
32.             <div class="headind_srch">
33.                 <div class="recent_heading">
34.                     <h4>Recent</h4>
35.                 </div>
36.                 <div class="srch_bar">
37.                     <div class="stylish-input-group">
38.                         <input type="text" class="search-
bar" placeholder="Search">
39.                         <span class="input-group-addon">
40.                             <button type="button"> <i class="fa fa-
search" aria-hidden="true"></i> </button>
41.                         </span> </div>
42.                     </div>
43.                 </div>
44.                 <div class="inbox_chat">
45.                     <?php
46.                         <foreach ($users as $u) : ?>
47.                             <?php
48.                                 $status = $this->logModel->getLogByUid($u['id']);
49.                                 $noHP = $this->UM-
>getUserByEmplid($u['emplid'])['phone'];
50.                             ?>
51.                             <div class="chat_list" data-id="<? = $u['id']; ?>">
52.                                 <div class="chat_people">
53.                                     <div class="chat_img">  </div>
54.                                     <div class="chat_ib">
55.                                         <h5><? = $u['name'] ?> <strong id="status_<? = $u
['id'] ?>"></strong></h5>
56.                                         <p><? = $u['role'] . ' (' . $u['emplid'] . ') '
?> <a href="https://api.whatsapp.com/send?phone=62<? = substr($noHP, 1) ?>" class="btn s
ocial-btn btn-success"><i class="fab fa-whatsapp"></i></a> <span style="float: inline-
end;" class="badge badge-danger" id="inbox_<? = $u['id'] ?>"></span></p>
57.                                         </div>
58.                                     </div>
59.                                 </div>
60.                             <script>
61.                                 setInterval(function() {
62.                                     <const userId = '<? = $u['id'] ?>'
63.                                     <const meId = '<? = $me['id'] ?>'
64.                                     <const selector = '#status_' + '<? = $u['id'] ?>'
65.                                     <const notif = '#inbox_' + '<? = $u['id'] ?>'

```

```

66.                $(selector).load(base_url + 'chating/getstatususer?
id=' + userId)
67.                $(notif).load(base_url + 'chating/getnotifchatdetail?meId=' + meId + '&youId=' + userId)
68.                }, 1000)
69.            </script>
70.
71.            <?php endforeach; ?>
72.        </div>
73.    </div>
74. </div>
75. <div class="col-md-7" id="message">
76.     <div class="mesgs">
77.         <div class="msg_history" id="msg_history">
78.         </div>
79.         <div class="type_msg">
80.             <div class="input_msg_write" id="input_msg_write">
81.                 <input type="hidden" id="meId">
82.                 <input type="hidden" id="youId">
83.                 <input type="text" class="write_msg" id="textwrite" value="
" placeholder="Type a message" />
84.                 <button class="msg_send_btn" onclick="return send()" type="
button"><i class="ik ik-send" aria-hidden="true"></i></button>
85.             </div>
86.         </div>
87.     </div>
88. </div>
89. </div>
90. </div>
91. </div>

```

application\views\views_course\activitydetail

```

1. <div class="main-content">
2.
3.     <div class="container-fluid">
4.         <?php $check_result = $this->quiz-
>getResultQuizByActIdUserId($activity['id'], $me['id']); ?>
5.         <div class="page-header">
6.             <div class="row align-items-end">
7.                 <div class="col-lg-8">
8.                     <div class="page-header-title">
9.                         <i class="ik ik-edit bg-blue"></i>
10.                     <div class="d-inline">
11.                         <h5>My Course</h5>
12.                         <span>Access by: User</span>
13.                     </div>
14.                 </div>
15.             </div>
16.             <div class="col-lg-4">
17.                 <nav class="breadcrumb-container" aria-label="breadcrumb">
18.                     <ol class="breadcrumb">
19.                         <li class="breadcrumb-item">
20.                             <a href="<?= base_url('course') ?>"><i class="ik ik-
home"></i></a>
21.                         </li>
22.                         <li class="breadcrumb-
item"><a href="<?= base_url('course/open/' . $activity['course_id']) ?>">My Course</a><
/li>

```

```

23.         <li class="breadcrumb-item active" aria-
current="page">Quiz</li>
24.     </ol>
25. </nav>
26. </div>
27. </div>
28. </div>
29. <?php if ($me['role_id'] == 2 or $me['role_id'] == 3) : ?>
30.     <a href="<? = base_url('activity/reportquiz/' . $activity['id']) ?>" class="
btn btn-success mb-3"><i class="fas fa-book"></i> Report</a>
31. <?php endif; ?>
32. <span id="id_ujian" data-key="<? = $encrypted_id ?>"></span>
33. <div class="card text-center">
34.     <div class="card-header">
35.         <h5><? = $activity['activity_name'] ?></h5>
36.     </div>
37.     <div class="card-body">
38.         <p class="card-
text">This quiz will be open on <strong><? = date('d F Y h:i A', $activity['start_date']
) ?></strong></p>
39.         <p class="card-
text">This quiz will be close on <strong><? = date('d F Y h:i A', $activity['end_date'])
?></strong></p>
40.         <p class="card-
text">Grade to pass: <strong><? = $activity['grade_to_pass'] ?></strong></p>
41.         <p class="card-
text">Limit: <strong><? = $activity['limit_time'] . ' Minutes' ?></strong></p>
42.         <p class="card-
text">Total Question: <strong><? = $activity['total_question'] ?></strong></p>
43.         <?php if ((int) $activity['start_date'] < time() and $check_result-
>num_rows() < 1 and (int) $activity['end_date'] > time()) : ?>
44.             <a href="<? = base_url('quiz/?key=' . $encrypted_id) ?>" class="btn
btn-primary mr-3">Attempts now</a>
45.         <?php else : ?>
46.             <a href="<? = base_url('course/open/' . $activity['course_id']) ?>"
class="btn btn-primary mr-3">Back to the course</a>
47.         <?php endif ?>
48.         <?php if ($me['role_id'] == 2 or $me['role_id'] == 3) : ?>
49.             <a href="<? = base_url('activity/question/' . $activity['id']) ?>" c
lass="btn btn-primary">Questions Bank</a>
50.         <?php endif; ?>
51.     </div>
52. </div>
53. <?php if ($check_result->num_rows() == 1) : $result = $check_result-
>row_array(); ?>
54.     <div class="card">
55.         <div class="card-header">
56.             Result
57.         </div>
58.         <div class="card-body">
59.             <table class="table table-striped">
60.                 <thead>
61.                     <tr>
62.                         <th scope="col">#</th>
63.                         <th scope="col">Total Correct Answer</th>
64.                         <th scope="col">Grade</th>
65.                         <th scope="col">Result</th>
66.                     </tr>
67.                 </thead>
68.                 <tbody>
69.                     <tr>

```

```

70.         <th scope="row">1</th>
71.         <td><?= $result['total_correct_answer'] ?>/<?= $activit
y['total_question'] ?></td>
72.         <td><?= $result['grade'] ?>/100.00</td>
73.         <?php if ((int) $result['grade'] >= (int) $activity['gr
ade_to_pass']) : ?>
74.             <td><span class="badge badge-pill badge-
success">Passed</span></td>
75.             <?php else : ?>
76.             <td><span class="badge badge-pill badge-
danger">Not Passed</span></td>
77.             <?php endif; ?>
78.         </tr>
79.     </tbody>
80. </table>
81. </div>
82. </div>
83.
84. <?php endif; ?>
85. </div>
86. </div>

```

application\views\views_course\addquiz

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.
4.         <div class="page-header">
5.             <div class="row align-items-end">
6.                 <div class="col-lg-8">
7.                     <div class="page-header-title">
8.                         <i class="ik ik-edit bg-blue"></i>
9.                         <div class="d-inline">
10.                            <h5>Add Quiz</h5>
11.                            <span>Access by: Instructor</span>
12.                        </div>
13.                    </div>
14.                </div>
15.                <div class="col-lg-4">
16.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
17.                        <ol class="breadcrumb">
18.                            <li class="breadcrumb-item">
19.                                <a href="<?= base_url('course') ?>"><i class="ik ik-
home"></i></a>
20.                            </li>
21.                            <li class="breadcrumb-
item"><a href="<?= base_url('course/open/') . $course_id ?>">My Course</a></li>
22.                            <li class="breadcrumb-item active" aria-
current="page">Add Quiz</li>
23.                        </ol>
24.                    </nav>
25.                </div>
26.            </div>
27.        </div>
28.        <div class="flash-data" data-flashdata="<?= $this->session-
>flashdata('message'); ?>"></div>
29.        <div class="flash-data-icon" data-icon="<?= $this->session-
>flashdata('icon'); ?>"></div>

```



```

69.         <label for="">Grade to pass <span class="text-
danger">*</span></label>
70.         <input type="number" max="100" min="0" onkeypress="return is_numeric(event)" value="<? set_value('grade') ?>" class="form-control" name="grade" placeholder="Grade to pass">
71.         <small class="text-danger"><? form_error('grade'); ?></small>
72.     </div>
73.     <div class="form-group">
74.         <label for="">Total Question <span class="text-
danger">*</span></label>
75.         <input type="text" onkeypress="return is_numeric(event)" value="<? set_value('total_question') ?>" class="form-control" name="total_question" placeholder="Total Question">
76.         <small class="text-danger"><? form_error('total_question'); ?></small>
77.     </div>
78.     <div class="d-flex flex-row-reverse">
79.         <button type="submit" class="btn btn-primary mr-2">Save</button>
80.     </div>
81. </form>
82. </div>
83. </div>
84. </div>
85. </div>
86. </div>
87. </div>

```

application\views\views_course\addresource

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.
4.         <div class="page-header">
5.             <div class="row align-items-end">
6.                 <div class="col-lg-8">
7.                     <div class="page-header-title">
8.                         <i class="ik ik-edit bg-blue"></i>
9.                         <div class="d-inline">
10.                            <h5>Add Resource</h5>
11.                            <span>Access by: Instructor</span>
12.                        </div>
13.                    </div>
14.                </div>
15.                <div class="col-lg-4">
16.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
17.                        <ol class="breadcrumb">
18.                            <li class="breadcrumb-item">
19.                                <a href="<? base_url('course') ?>"><i class="ik ik-
home"></i></a>
20.                            </li>
21.                            <li class="breadcrumb-
item"><a href="<? base_url('course/open/') . $course_id ?>">My Course</a></li>
22.                            <li class="breadcrumb-item active" aria-
current="page">Add Resource</li>
23.                        </ol>
24.                    </nav>

```

```

25.         </div>
26.     </div>
27. </div>
28.     <?php if ($this->session->flashdata('error')) : ?>
29.         <div class="alert bg-danger alert-danger text-
white" role="alert"><?=$this->session->flashdata('error'); ?></div>
30.     <?php endif; ?>
31.     <div class="row">
32.         <div class="col-md-12">
33.             <div class="card">
34.                 <div class="card-header">
35.                     <h3>Form add activity</h3>
36.                 </div>
37.                 <div class="card-body">
38.                     <?php echo form_open_multipart('activity/addresource/' . $secti
on_id); ?>
39.                     <div class="form-group">
40.                         <label for="exampleInputName1">Activity Name <span class="t
ext-danger">*</span></label>
41.                         <input type="text" class="form-
control" value="<?=$ set_value('activity_name') ?>" name="activity_name" id="exampleInpu
tName1" placeholder="Activity Name">
42.                         <small class="text-
danger"><?=$ form_error('activity_name'); ?></small>
43.                     </div>
44.                     <div class="form-group">
45.                         <label>Upload File</label>
46.                         <input type="file" name="upload" class="file-upload-
default">
47.                         <div class="input-group col-xs-12">
48.                             <input type="text" class="form-control file-upload-
info" disabled placeholder="Upload File">
49.                             <span class="input-group-append">
50.                                 <button class="file-upload-browse btn btn-
primary" type="button">Upload</button>
51.                             </span>
52.                         </div>
53.                         <small class="text-
danger"><?=$ form_error('upload'); ?></small>
54.                     </div>
55.                     <div class="d-flex flex-row-reverse">
56.                         <button type="submit" class="btn btn-primary mr-
2">Save</button>
57.                     </div>
58.                 </form>
59.             </div>
60.         </div>
61.     </div>
62. </div>
63. </div>
64. </div>

```

application\views\views_course\category

```

1. <!-- Awal Main Content -->
2. <div class="main-content">
3.     <div class="container-fluid">

```

[illegible]

```

58.         <a class="btn-action edit" data-
toggle="modal" data-target="#editLayoutItem" id="edit" data-
id="<?=$c['id']; ?>"><i class="ik ik-edit-2"></i></a>
59.         <a href="<?=$base_url('course/deletecat
egory/') . $c['id'] ?>" class="btn-delete btn-action" data-
value="category"><i class="ik ik-trash-2"></i></a>
60.     </div>
61. </td>
62. </tr>
63. <?php endforeach; ?>
64. </tbody>
65. </table>
66.
67. </div>
68. </div>
69. </div>
70. <div class="col-sm-6">
71.
72. </div>
73. </div>
74.
75. </div>
76. </div>
77. <!-- Akhir Main Content -->
78.
79. <div class="modal fade edit-layout-modal" id="editLayoutItem" tabindex="-
1" role="dialog" aria-labelledby="editLayoutItemLabel" aria-hidden="true">
80.     <div class="modal-dialog" role="document">
81.         <div class="modal-content">
82.             <div class="modal-header">
83.                 <h5 class="modal-
title" id="editLayoutItemLabel">Add Course Category</h5>
84.                 <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">x</span></button>
85.             </div>
86.             <form action="<?=$base_url('course/addcategory') ?>" method="post">
87.                 <div class="modal-body description">
88.                     <div class="card-body">
89.                         <form class="forms-sample">
90.                             <div class="form-group">
91.                                 <label for="cat">Category</label>
92.                                 <input type="text" name="category" class="form-
control" id="cat" placeholder="Category">
93.                                 <input type="hidden" name="id" class="form-
control" id="catId">
94.                             </div>
95.                         </form>
96.                     </div>
97.                     <div class="modal-footer">
98.                         <button type="submit" class="btn btn-primary">Save</button>
99.                     </div>
100.                 </form>
101.             </div>
102.         </div>
103.     </div>
104. </div>

```

application\views\views_course\courseopen

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="flash-data" data-flashdata="<?= $this->session-
4.             >flashdata('message'); ?>"></div>
5.         <div class="flash-data-icon" data-icon="<?= $this->session-
6.             >flashdata('icon'); ?>"></div>
7.         <div class="flash-data-text" data-text="<?= $this->session-
8.             >flashdata('text'); ?>"></div>
9.         <div class="page-header">
10.            <div class="row align-items-end">
11.                <div class="col-lg-8">
12.                    <div class="page-header-title">
13.                        <i class="ik ik-edit bg-blue"></i>
14.                        <div class="d-inline">
15.                            <h5>My Course</h5>
16.                            <span><?= $course['title'] ?></span>
17.                        </div>
18.                    </div>
19.                    <div class="col-lg-4">
20.                        <nav class="breadcrumb-container" aria-label="breadcrumb">
21.                            <ol class="breadcrumb">
22.                                <li class="breadcrumb-item">
23.                                    <a href="<?= base_url('course') ?>"><i class="ik ik-
24.                                        home"></i></a>
25.                                </li>
26.                                <li class="breadcrumb-item active" aria-
27.                                    current="page">My Course</li>
28.                            </ol>
29.                        </nav>
30.                    </div>
31.                </div>
32.                <div class="card">
33.                    <div class="card-header">
34.                        <h5 class="mr-3"><?= $sec['section_name'] ?></h5>
35.                        <?php if ($me['role_id'] == 2 or $me['role_id'] == 3) : ?>
36.                            <button class="btn btn-primary mb-4 mr-4" data-toggle="modal" data-
37.                                target="#demoModal"><i class="ik ik-plus"></i> Add section</button>
38.                            <?php endif; ?>
39.                            <a href="<?= base_url('chating/chat/') . $this->uri-
40.                                >segment(3, 0); ?>" class="btn btn-success mb-4"><i class="ik ik-message-
41.                                    square"></i> Message <span class="badge badge-light" id="inbox"></span></a>
42.                        <?php foreach ($section as $sec) : ?>
43.                            <div class="card">
44.                                <div class="card-header">
45.                                    <h5 class="mr-3"><?= $sec['section_name'] ?></h5>
46.                                    <?php if ($me['role_id'] == 2 or $me['role_id'] == 3) : ?>
47.                                        <button class="btn btn-primary btn-add" data-
48.                                            toggle="modal" data-target="#activity" data-id="<?= $sec['id']; ?>"><i class="ik ik-
49.                                                plus"></i> add activity</button>
50.                                        <a href="<?= base_url('activity/deletesection/') . $sec['id'];
51.                                            ?>" class="btn btn-danger ml-4" data-value="section"><i class="ik ik-trash-
52.                                                2"></i> Delete</a>
53.                                    <?php endif; ?>
54.                                </div>
55.                                <div class="card-body">
56.                                    <?php $act = $this->Act->getActivityById($sec['id']);
57.                                    <?php foreach ($act as $act) : ?>
58.                                        <div class="item-activity">
59.                                            <a href="<?= base_url('activity/activitydetail/') . $act['i
60.                                                d'] ?>"><i class="<?= $act['icon'] ?>"></i><span><?= $act['activity_name'] ?></span></a>
61.                                        </div>
62.                                    </div>
63.                                </div>
64.                            </div>
65.                        </div>
66.                    </div>
67.                </div>
68.            </div>
69.        </div>
70.    </div>

```

```

48.         <?php if ($me['role_id'] == 2 or $me['role_id'] == 3) : ?>
49.             <div class="item-actions">
50.                 <?= $act['type'] == 'Quiz' ? '<a href="' . base_url('ac
activity/edit/') . $act['id'] . '"><i class="ik ik-edit-2"></i></a>' : '' ?>
51.                 <a href="<?= base_url('activity/delete/') . $act['id'];
?>" class="btn-delete" data-value="activity"><i class="ik ik-trash-2"></i></a>
52.             </div>
53.         <?php endif; ?>
54.     <?php endforeach; ?>
55. </div>
56. </div>
57. <?php endforeach; ?>
58. </div>
59. </div>
60.
61. <div class="modal fade" id="demoModal" tabindex="-1" role="dialog" aria-
labelledby="demoModallabel" aria-hidden="true">
62.     <div class="modal-dialog" role="document">
63.         <div class="modal-content">
64.             <form action="<?= base_url('course/sectionadd') ?>" method="post">
65.                 <div class="modal-header">
66.                     <h5 class="modal-title" id="demoModallabel">Modal title</h5>
67.                     <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true"><!--></span></button>
68.                 </div>
69.                 <div class="modal-body">
70.                     <div class="form-group">
71.                         <input type="hidden" name="course_id" class="form-
control" value="<?= $this->uri->segment(3, 0) ?>">
72.                     </div>
73.                     <div class="form-group">
74.                         <label>Section Name</label><br>
75.                         <input type="text" name="section_name" class="form-control">
76.                     </div>
77.                 </div>
78.                 <div class="modal-footer">
79.                     <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
80.                     <button type="submit" class="btn btn-primary">Save</button>
81.                 </div>
82.             </form>
83.         </div>
84.     </div>
85. </div>
86.
87. <div class="modal fade" id="activity" tabindex="-1" role="dialog" aria-
labelledby="demoModallabel" aria-hidden="true">
88.     <div class="modal-dialog" role="document">
89.         <div class="modal-content">
90.             <form action="<?= base_url('activity/activityadd') ?>" method="post">
91.                 <div class="modal-header">
92.                     <h5 class="modal-title" id="demoModallabel">Activity</h5>
93.                     <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true"><!--></span></button>
94.                 </div>
95.                 <ul class="list-group list-group-flush">
96.                     <li class="list-group-item">
97.                         <div class="form-check">
98.                             <input class="form-check-
input" type="radio" name="activity" id="exampleRadios1" value="resource" checked>

```

```

99.         <input class="form-
control" type="hidden" name="section_id" id="sectionId">
100.             <label class="form-check-
label" for="exampleRadios1">
101.                 Resource
102.             </label>
103.         </div>
104.     </li>
105.     <li class="list-group-item">
106.         <div class="form-check">
107.             <input class="form-check-
input" type="radio" name="activity" id="exampleRadios2" value="quiz" checked>
108.             <label class="form-check-
label" for="exampleRadios2">
109.                 Quiz
110.             </label>
111.         </div>
112.     </li>
113. </ul>
114. <div class="modal-footer">
115.     <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
116.     <button type="submit" class="btn btn-
primary">Save</button>
117. </div>
118. </form>
119. </div>
120. </div>
121. </div>
122.
123. <script>
124.     setInterval(function() {
125.         $('#inbox').load(base_url + 'chating/getnotifchat')
126.     }, 1000)
127. </script>

```

```
application\views\views course\enroll
```

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="page-header">
4.             <div class="flash-data" data-flashdata="{<?= $this->session-
>flashdata('message'); ?>}"></div>
5.             <div class="flash-data-icon" data-icon="{<?= $this->session-
>flashdata('icon'); ?>}"></div>
6.             <div class="flash-data-text" data-text="{<?= $this->session-
>flashdata('text'); ?>}"></div>
7.             <div class="row align-items-end">
8.                 <div class="col-lg-8">
9.                     <div class="page-header-title">
10.                        <i class="ik ik-user-plus bg-blue"></i>
11.                        <div class="d-inline">
12.                            <h5>Course Enroll</h5>
13.                            <span>Access by: Instructor</span>
14.                        </div>
15.                    </div>
16.                </div>
17.                <div class="col-lg-4">
18.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
19.                        <ol class="breadcrumb">

```

```

20.             <li class="breadcrumb-item">
21.                 <a href="{<?= base_url('course') ?>}"><i class="ik ik-
home"></i></a>
22.             </li>
23.             <li class="breadcrumb-item active" aria-
current="page">Course Enroll</li>
24.         </ol>
25.     </nav>
26. </div>
27. </div>
28. </div>
29.
30.     <div class="row">
31.         <div class="col-sm-12">
32.             <div class="card">
33.                 <div class="card-header">
34.                     <h3>Data Course</h3>
35.                 </div>
36.                 <div class="card-body">
37.                     <table id="data_table" class="table">
38.                         <thead>
39.                             <tr>
40.                                 <th>No</th>
41.                                 <th class="nosort">Title</th>
42.                                 <th>Instructor</th>
43.                                 <th>Date</th>
44.                                 <th>Category</th>
45.                                 <th class="nosort mr-0"></th>
46.                             </tr>
47.                         </thead>
48.                         <tbody>
49.                             <?php $i = 1;
50.                             foreach ($course as $c) : ?>
51.                                 <tr>
52.                                     <td><?= $i++; ?></td>
53.                                     <td><?= $c['title'] ?></td>
54.                                     <td><?= $c['name'] ?></td>
55.                                     <td><?php date_default_timezone_set('Asia/Jakar
ta');
56.                                     echo date('j F Y', $c['date_created']) ?></
td>
57.                                     <td><?= $c['category'] ?></td>
58.                                     <td><a href="{<?= base_url('course/enrolluser/'
. $c['id'] ?>}" class="btn btn-primary">Enroll</a></td>
59.                                 </tr>
60.                             <?php endforeach; ?>
61.                         </tbody>
62.                     </table>
63.                 </div>
64.             </div>
65.         </div>
66.     </div>
67. </div>
68. </div>
69. </div>

```

application\views\views_course\enrolluser

```

1. <div class="main-content">
2.     <div class="container-fluid">

```

```

3.         <div class="page-header">
4.             <div class="flash-data" data-flashdata="<?= $this->session-
>flashdata('message'); ?>"></div>
5.             <div class="flash-data-icon" data-icon="<?= $this->session-
>flashdata('icon'); ?>"></div>
6.             <div class="flash-data-text" data-text="<?= $this->session-
>flashdata('text'); ?>"></div>
7.             <div class="row align-items-end">
8.                 <div class="col-lg-8">
9.                     <div class="page-header-title">
10.                        <i class="ik ik-user-plus bg-blue"></i>
11.                        <div class="d-inline">
12.                            <h5>Users Enroll</h5>
13.                            <span>Access by: Instructor</span>
14.                        </div>
15.                    </div>
16.                </div>
17.                <div class="col-lg-4">
18.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
19.                        <ol class="breadcrumb">
20.                            <li class="breadcrumb-item">
21.                                <a href="<?= base_url('course') ?>"><i class="ik ik-
home"></i></a>
22.                            </li>
23.                            <li class="breadcrumb-item">
24.                                <a href="<?= base_url('course/enroll') ?>">Course Enrol
l</a>
25.                            </li>
26.                            <li class="breadcrumb-item active" aria-
current="page">Users Enroll</li>
27.                        </ol>
28.                    </nav>
29.                </div>
30.            </div>
31.        </div>
32.        <div class="row">
33.            <div class="flash-data" data-flashdata="<?= $this->session-
>flashdata('message'); ?>"></div>
34.            <div class="flash-data-icon" data-icon="<?= $this->session-
>flashdata('icon'); ?>"></div>
35.            <div class="flash-data-text" data-text="<?= $this->session-
>flashdata('text'); ?>"></div>
36.            <div class="col-md-8">
37.                <div class="card mb-3">
38.                    
39.                    <div class="card-body">
40.                        <h5 class="card-
title"><?= $course['title'] ?> <span class="badge badge-pill badge-
primary"><?= $course['category'] ?></span></h5>
41.                        <h5 class="card-
title">Instructor : <?= $course['name'] ?></h5>
42.                        <p class="card-text"><small class="text-
muted"><?= date('j F Y', $course['date_created']) ?></small></p>
43.                    </div>
44.                    <button class="btn btn-primary mb-3 mx-3" data-toggle="modal" data-
target="#demoModal"><i class="ik ik-user-plus"></i> Enroll</button>
45.                </div>
46.            </div>
47.        </div>
48.

```

```

49.         <div class="row">
50.             <div class="col">
51.                 <div class="card">
52.                     <div class="card-header">
53.                         <h3>Users Enrolled</h3>
54.                     </div>
55.                     <div class="card-body">
56.                         <table id="data_table" class="table">
57.                             <thead>
58.                                 <tr>
59.                                     <th>No</th>
60.                                     <th>NPK</th>
61.                                     <th>Name</th>
62.                                     <th>Email</th>
63.                                     <th>Approval Status</th>
64.                                     <th></th>
65.                             </thead>
66.                             <tbody>
67.                                 <?php $i = 1;
68.                                 foreach ($enrolled as $e) : ?>
69.                                     <tr>
70.                                         <td><?= $i++; ?></td>
71.                                         <td><?= $e['emplid']; ?></td>
72.                                         <td><?= $e['name']; ?></td>
73.                                         <td><?= $e['email']; ?></td>
74.                                         <?php if ($e['is_approve'] == 1) : ?>
75.                                             <td>Approved</td>
76.                                         <?php else : ?>
77.                                             <td>Waiting for Approval</td>
78.                                         <?php endif; ?>
79.                                         <td>
80.                                             <div class="table-actions">
81.                                                 <a href="<?= base_url('course/unenroll/
?user=' . $e['id'] . '&course=') . $this->uri->segment(3, 0); ?>"><i class="ik ik-
x"></i></a>
82.                                             </div>
83.                                         </td>
84.                                     </tr>
85.                                 <?php endforeach; ?>
86.                             </tbody>
87.                         </table>
88.
89.                     </div>
90.                 </div>
91.             </div>
92.         </div>
93.     </div>
94. </div>
95.
96.
97. <div class="modal fade" id="demoModal" tabindex="-1" role="dialog" aria-
labelledby="demoModallabel" aria-hidden="true">
98.     <div class="modal-dialog" role="document">
99.         <div class="modal-content">
100.             <form action="<?= base_url('course/enrollcourse') ?>" method="post">
101.
102.                 <div class="modal-header">
103.                     <h5 class="modal-
title" id="demoModallabel">Users Enroll</h5>
103.                     <button type="button" class="close" data-
dismiss="modal" aria-label="Close"><span aria-hidden="true">x</span></button>

```

```

104.         </div>
105.         <div class="modal-body">
106.
107.             <div class="form-group">
108.                 <select name="user[]" class="form-
control select2" style="width:400px" multiple="multiple">
109.                     <?php foreach ($users as $u) : ?>
110.                         <?php if ($this->Enroll->checkUser($this->uri-
>segment(3, 0), $u['id'], 1) > 0 or $u['emplid'] == $this->session-
>userdata('emplid')) {
111.                             continue;
112.                         } ?>
113.                         <option value="<?= $u['id'] ?>"><?= $u['name'] ?
></option>
114.
115.                     <?php endforeach; ?>
116.                 </select>
117.             </div>
118.             <div class="form-group">
119.                 <input type="hidden" name="id" value="<?= $this->uri-
>segment(3, 0); ?>">
120.
121.             </div>
122.             <div class="modal-footer">
123.                 <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
124.                 <button type="submit" class="btn btn-
primary">Enroll</button>
125.             </div>
126.         </form>
127.     </div>
128. </div>
129. </div>

```

```
application\views\views course\importquestion
```

```

1. <div class="main-content">
2.   <div class="container-fluid">
3.     <div class="page-header">
4.       <div class="row align-items-end">
5.         <div class="col-lg-8">
6.           <div class="page-header-title">
7.             <i class="ik ik-users bg-blue"></i>
8.             <div class="d-inline">
9.               <h5>Data Users</h5>
10.              <span>lorem ipsum dolor sit amet, consectetur adipisicing e
11.                lit</span>
12.            </div>
13.          </div>
14.        <div class="col-lg-4">
15.          <nav class="breadcrumb-container" aria-label="breadcrumb">
16.            <ol class="breadcrumb">
17.              <li class="breadcrumb-item">
18.                <a href="..index.html"><i class="ik ik-
19.                  home"></i></a>
20.              </li>
21.              <li class="breadcrumb-item">

```

```

21.             <a href="#">Pages</a>
22.         </li>
23.         <li class="breadcrumb-item active" aria-
current="page">Data Users</li>
24.     </ol>
25. </nav>
26. </div>
27. </div>
28. </div>
29.
30.     <div class="row">
31.         <div class="col-sm-12">
32.             <a href="<?= base_url('excel/format_question.xlsx') ?>" class="btn btn-
primary mb-4"> <i class="ik ik-file"></i>Download template</a>
33.         </div>
34.     </div>
35.
36.     <div class="row">
37.         <div class="col-sm-12">
38.             <div class="card">
39.                 <div class="card-header">
40.                     <h3>Data Users</h3>
41.                 </div>
42.                 <div class="card-body">
43.                     <?php echo form_open_multipart('activity/importquestion/' . $th
is->uri->segment(3, 0)); ?>
44.                     <div class="row">
45.                         <div class="col-md-8">
46.                             <div class="form-group">
47.                                 <label>File upload</label>
48.                                 <input type="file" name="question" class="file-
upload-default">
49.                                 <div class="input-group col-xs-12">
50.                                     <input type="text" class="form-control file-
upload-info" disabled placeholder="Upload Image">
51.                                     <span class="input-group-append">
52.                                         <button class="file-upload-browse btn btn-
primary" type="button">Upload</button>
53.                                         <button type="submit" class="btn btn-
primary ml-3" name="preview">Preview</button>
54.                                     </span>
55.                                 </div>
56.                             </div>
57.                         </div>
58.                     </div>
59.                 </form>
60.                 <?php
61.                 if (isset($_POST['preview'])) :
62.                     if (isset($upload_error)) :
63.                         ?>
64.                         <div class="alert alert-danger" role="alert">
65.                             <?= $upload_error; ?>
66.                         </div>
67.                     <?php
68.                     die;
69.                 endif; ?>
70.
71.                 <form action="<?= base_url('activity/import/' . $this->uri-
>segment(3, 0)) ?>" method="post">
72.                     <div class="alert alert-
danger" id="kosong" role="alert">

```

```

73.         Please complete the data! (<span id="jumlah_kosong"
></span>)
74.     </div>
75.     <table id="data_table" class="table">
76.         <thead>
77.             <tr>
78.                 <th class="nosort">Question</th>
79.                 <th>Answer A</th>
80.                 <th>Answer B</th>
81.                 <th>Answer C</th>
82.                 <th>Answer D</th>
83.                 <th>Answer E</th>
84.                 <th>Correct Answer</th>
85.                 <th class="nosort mr-0"></th>
86.             </tr>
87.         </thead>
88.         <?php
89.             $numrow = 1;
90.             $kosong = 0;
91.
92.             foreach ($sheet as $row) :
93.
94.                 // Lewat data pada baris ini (masuk ke looping
selanjutnya / baris selanjutnya)
95.                 // Cek $numrow apakah lebih dari 1
96.                 // Artinya karena baris pertama adalah nama-
nama kolom
97.                 // Jadi dilewat saja, tidak usah diimport
98.
99.                 if ($numrow > 1) : ?>
100.                     <?php
// Validasi apakah semua data telah
diisi
101.                     $question_td = (!empty($row['A'
])) ? "" : " style='background: #E07171;"; // Jika Question kosong, beri warna merah
102.                     $answer_a_td = (!empty($row['B'
])) ? "" : " style='background: #E07171;"; // Jika Jawaban A kosong, beri warna merah
103.                     $answer_b_td = (!empty($row['C'
])) ? "" : " style='background: #E07171;"; // Jika Nama kosong, beri warna merah
104.                     $answer_c_td = (!empty($row['D'
])) ? "" : " style='background: #E07171;"; // Jika Jenis Kelamin kosong, beri warna me
rah
105.                     $answer_d_td = (!empty($row['E'
])) ? "" : " style='background: #E07171;"; // Jika Alamat kosong, beri warna merah
106.                     $answer_e_td = (!empty($row['F'
])) ? "" : " style='background: #E07171;"; // Jika Alamat kosong, beri warna merah
107.                     $correct_answer_td = (!empty($r
ow['G'])) ? "" : " style='background: #E07171;"; // Jika Alamat kosong, beri warna mer
ah
108.                     // Jika salah satu data ada yang kos
ong
109.                     if ($row['A'] == "" or $row['B'] ==
"" or $row['C'] == "" or $row['D'] == "" or $row['E'] == "" or $row['F'] == "" or $row[
'G'] == "") {
110.                         $kosong++; // Tambah 1 variabel
$kosong

```

```

111.         }
112.         ?>
113.         <tbody>
114.             <tr>
115.                 <td <?= $question_td ?><?=
    $row['A'] ?></td>
116.                 <td <?= $answer_a_td ?><?=
    $row['B'] ?></td>
117.                 <td <?= $answer_b_td ?><?=
    $row['C'] ?></td>
118.                 <td <?= $answer_c_td ?><?=
    $row['D'] ?></td>
119.                 <td <?= $answer_d_td ?><?=
    $row['E'] ?></td>
120.                 <td <?= $answer_e_td ?><?=
    $row['F'] ?></td>
121.                 <td <?= $correct_answer_td ?
    ><?= $row['G'] ?></td>
122.             </tr>
123.         </tbody>
124.     <?php endif;
125.     $numrow++; ?>
126. <?php endforeach; ?>
127. </table>
128. <?php
129.     if ($kosong > 0) { ?>
130.         <script>
131.             $(document).ready(function() {
132.                 // Ubah isi dari tag span dengan id
    jumlah_kosong dengan isi dari variabel kosong
133.                 $("#jumlah_kosong").html('<?php echo
    $kosong; ?>');
134.                 $("#kosong").show(); // Munculkan al
    ert validasi kosong
135.             });
136.         </script>
137.     <?php } else { ?>
138.         <button type='submit' class="btn btn-
    primary" name='import'>Import</button>
139.     <?php } ?>
140. </form>
141. <?php endif; ?>
142. </div>
143. </div>
144. </div>
145. </div>
146. </div>
147. </div>

```

application\views\views_course\listcourse

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="page-header">
4.             <div class="flash-data" data-flashdata="<?= $this->session-
    >flashdata('message'); ?>"></div>
5.             <div class="flash-data-icon" data-icon="<?= $this->session-
    >flashdata('icon'); ?>"></div>

```

```

6.         <div class="flash-data-text" data-text="<?=$this->session-
>flashdata('text'); ?>"></div>
7.         <div class="row align-items-end">
8.             <div class="col-lg-8">
9.                 <div class="page-header-title">
10.                     <i class="ik ik-file-text bg-blue"></i>
11.                     <div class="d-inline">
12.                         <h5>List Course</h5>
13.                         <span>Access by: User</span>
14.                     </div>
15.                 </div>
16.             </div>
17.             <div class="col-lg-4">
18.                 <nav class="breadcrumb-container" aria-label="breadcrumb">
19.                     <ol class="breadcrumb">
20.                         <li class="breadcrumb-item">
21.                             <a href="<?=$base_url('course') ?>"><i class="ik ik-
home"></i></a>
22.                             </li>
23.                             <li class="breadcrumb-item active" aria-
current="page">List Course</li>
24.                         </ol>
25.                     </nav>
26.                 </div>
27.             </div>
28.         </div>
29.         <div class="row">
30.             <div class="col-md-12">
31.                 <div class="separator mb-20"></div>
32.                 <div class="mb-2 clearfix">
33.                     <a class="btn pt-0 pl-0 d-md-none d-lg-none" data-
toggle="collapse" href="#displayOptions" role="button" aria-expanded="true" aria-
controls="displayOptions">
34.                         Display Options
35.                         <i class="ik ik-chevron-down align-middle"></i>
36.                     </a>
37.                 </div>
38.                 <div class="collapse d-md-block display-options" id="displayOptions">
39.                     <div class="d-block d-md-inline-block">
40.                         <div class="search d-inline-block mr-1 mb-1 align-top">
41.                             <div class="form-group">
42.                                 <div class="form-inline">
43.                                     <input type="text" placeholder="Search..." class="
form-control" id="keyword">
44.                                 </div>
45.                             </div>
46.                         </div>
47.                     </div>
48.                 </div>
49.                 <div class="row layout-wrap" id="layout-wrap">
50.                     <?php if (!$course) {
51.                         echo "<h1>Data not found!</h1>";
52.                     } ?>
53.                     <?php foreach ($course as $course) : ?>
54.                         <div class="col-12 list-item list-item-thumb">
55.                             <div class="card d-flex flex-row mb-3">
56.                                 <a class="d-flex card-img" href="#editLayoutItem" data-
toggle="modal" data-target="#editLayoutItem">
57.                                     
58.                                 </a>

```

```

59.         <div class="d-flex flex-grow-1 min-width-zero card-
content">
60.             <div class="card-body align-self-center d-
flex flex-column flex-md-row justify-content-between min-width-zero align-items-md-
center">
61.                 <a class="list-item-heading mb-1 truncate w-
40 w-xs-100" data-id="<?= $course['id']; ?>" href="#editLayoutItem" data-
toggle="modal" data-target="#editLayoutItem">
62.                     <?= $course['title']; ?>
63.                 </a>
64.                 <p class="mb-1 text-small category w-15 w-xs-
100"><?= $course['name'] ?></p>
65.                 <p class="mb-1 text-muted text-small date w-
15 w-xs-100"><?php date_default_timezone_set('Asia/Jakarta');
66.                     echo date('j F Y', $course['date_created']) ?></p>
67.                 <div class="w-15 w-xs-100">
68.                     <span class="badge badge-pill badge-
primary"><?= $course['category'] ?></span>
69.                 </div>
70.             </div>
71.             <?php if ($me['role_id'] == 3 or $me['role_id'] ==
2 and $me['name'] == $course['name']) : ?>
72.                 <div class="list-actions">
73.                     <a href="<?= base_url('course/delete/') . $
course['id']; ?>" class="btn-delete" data-value="course"><i class="ik ik-trash-
2"></i></a>
74.                 </div>
75.                 <?php endif; ?>
76.             </div>
77.             <?php if ($this->Enroll-
>checkUser($course['id'], $me['id'], 1) > 0 or $course['name'] == $me['name'] or $me['r
ole_id'] == 3) : ?>
78.                 <a class="btn btn-icon btn-dark mt-1 mr-
1" href="<?= base_url('course/open/') . $course['id']; ?>"><i class="fas fa-sign-in-
alt"></i></a>
79.                 <?php endif; ?>
80.             </div>
81.         </div>
82.     <?php endforeach; ?>
83. </div>
84. </div>
85. </div>
86. </div>
87. </div>
88.
89.
90.
91. <div class="modal fade edit-layout-modal" id="editLayoutItem" tabindex="-
1" role="dialog" aria-labelledby="editLayoutItemLabel" aria-hidden="true">
92.     <div class="modal-dialog" role="document">
93.         <div class="modal-content">
94.             <div class="modal-header">
95.                 <h5 class="modal-
title" id="editLayoutItemLabel">Sed id mi non quam iaculis pulvinar.</h5>
96.                 <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">x</span></button>
97.             </div>
98.             <div class="modal-body description">
99.                 <form action="<?= base_url('course/selfenroll') ?>" method="post">
100.                     <div class="card" style="width: 100%;height: 100%;">

```

```

101.         
102.         <div class="card-body">
103.             <h5 class="card-title">Card title</h5>
104.             <h6 class="card-subtitle mb-2 text-
muted">Card subtitle</h6>
105.             <input type="hidden" name="user_id" id="user_id" val
ue="<?= $me['id'] ?>">
106.             <input type="hidden" name="course_id" id="course_id"
>
107.             <p class="card-
text">Some quick example text to build on the card title and make up the bulk of the ca
rd's content.</p>
108.         </div>
109.     </div>
110. </div>
111. <div class="modal-footer">
112.     <button type="submit" class="btn btn-primary">Enroll</button>
113. </div>
114. </form>
115. </div>
116. </div>
117. </div>

```

application\views\views_course\listsearch

```

1. <?php foreach ($course as $course) : ?>
2.     <div class="col-12 list-item list-item-thumb">
3.         <div class="card d-flex flex-row mb-3">
4.             <a class="d-flex card-img" href="#editLayoutItem" data-toggle="modal" data-
target="#editLayoutItem">
5.                 
6.             </a>
7.             <div class="d-flex flex-grow-1 min-width-zero card-content">
8.                 <div class="card-body align-self-center d-flex flex-column flex-md-
row justify-content-between min-width-zero align-items-md-center">
9.                     <a class="list-item-heading mb-1 truncate w-40 w-xs-100" data-
id="<?= $course['id']; ?>" href="#editLayoutItem" data-toggle="modal" data-
target="#editLayoutItem">
10.                         <?= $course['title']; ?>
11.                     </a>
12.                     <p class="mb-1 text-small category w-15 w-xs-
100"><?= $course['name'] ?></p>
13.                     <p class="mb-1 text-muted text-small date w-15 w-xs-
100"><?php date_default_timezone_set('Asia/Jakarta');
14.                                             echo da
te('j F Y', $course['date_created']) ?></p>
15.                     <div class="w-15 w-xs-100">
16.                         <span class="badge badge-pill badge-
primary"><?= $course['category'] ?></span>
17.                     </div>
18.                 </div>
19.                 <?php if ($me['role_id'] == 3 or $me['role_id'] == 2 and $me['name'] ==
$course['name']) : ?>
20.                     <div class="list-actions">
21.                         <a href="#editLayoutItem"><i class="ik ik-eye"></i></a>
22.                         <a href="layout-edit-item.html"><i class="ik ik-edit-
2"></i></a>

```



```

13.         <span>Access by: User</span>
14.     </div>
15. </div>
16. </div>
17. <div class="col-lg-4">
18.     <nav class="breadcrumb-container" aria-label="breadcrumb">
19.         <ol class="breadcrumb">
20.             <li class="breadcrumb-item">
21.                 <a href="{%= base_url('course') %}"><i class="ik ik-
home"></i></a>
22.             </li>
23.             <li class="breadcrumb-item active" aria-
current="page">My Course</li>
24.         </ol>
25.     </nav>
26. </div>
27. </div>
28. </div>
29. <div class="row">
30.     <div class="col-md-12">
31.         <div class="mb-2 clearfix">
32.             <a class="btn pt-0 pl-0 d-md-none d-lg-none" data-
toggle="collapse" href="#displayOptions" role="button" aria-expanded="true" aria-
controls="displayOptions">
33.                 Display Options
34.                 <i class="ik ik-chevron-down align-middle"></i>
35.             </a>
36.             <div class="collapse d-md-block display-
options" id="displayOptions">
37.                 <div class="d-block d-md-inline-block">
38.                     <span>Category</span>
39.                     <div class="search-sm d-inline-block mr-1 mb-1 align-
top">
40.                         <form action="" method="GET">
41.                             <select name="search" class="form-
control select2 mr-10">
42.                                 <?php foreach ($cat as $c) : ?>
43.                                     <option value="{%= $c['category'] %}"><? = $
c['category'] %></option>
44.                                 <?php endforeach; ?>
45.                             </select>
46.                         </div>
47.                         <button type="submit" class="btn btn-primary mt-
1">Search</button>
48.                     </form>
49.                 </div>
50.
51.             </div>
52.         </div>
53.         <div class="separator mb-20"></div>
54.
55.         <div class="row layout-wrap" id="layout-wrap">
56.             <?php if (!$course) {
57.                 echo "<h1>Data not found!</h1>";
58.             } ?>
59.             <?php foreach ($course as $course) : ?>
60.                 <div class="col-12 list-item list-item-thumb">
61.                     <div class="card d-flex flex-row mb-3">
62.                         <a class="d-flex card-img" href="#editLayoutItem" data-
toggle="modal" data-target="#editLayoutItem">

```

```

63.         
64.     </a>
65.     <div class="d-flex flex-grow-1 min-width-zero card-
content">
66.         <div class="card-body align-self-center d-
flex flex-column flex-md-row justify-content-between min-width-zero align-items-md-
center">
67.             <p class="list-item-heading mb-1 truncate w-
40 w-xs-100" data-id="<?= $course['id']; ?>" href="#editLayoutItem" data-
toggle="modal" data-target="#editLayoutItem">
68.                 <?= $course['title']; ?>
69.             </p>
70.             <p class="mb-1 text-small category w-15 w-xs-
100"><?= $course['name'] ?></p>
71.             <p class="mb-1 text-muted text-small date w-
15 w-xs-100"><?php date_default_timezone_set('Asia/Jakarta');
72.                 echo date('j F Y', $course['date_created']) ?></p>
73.             <div class="w-15 w-xs-100">
74.                 <span class="badge badge-pill badge-
primary"><?= $course['category'] ?></span>
75.             </div>
76.             </div>
77.             <?php if ($me['role_id'] == 3 or $me['role_id'] ==
2 and $me['name'] == $course['name']) : ?>
78.                 <div class="list-actions">
79.                     <a href="<?= base_url('course/delete/') . $
course['id']; ?>" class="btn-delete" data-value="course"><i class="ik ik-trash-
2"></i></a>
80.                 </div>
81.                 <?php endif; ?>
82.             </div>
83.             <?php if ($this->Enroll-
>checkUser($course['id'], $me['id'], 1) > 0 or $course['name'] == $me['name'] or $me['r
ole_id'] == 3) : ?>
84.                 <a class="btn btn-icon btn-dark mt-1 mr-
1" href="<?= base_url('course/open/') . $course['id']; ?>"><i class="fas fa-sign-in-
alt"></i></a>
85.             <?php endif; ?>
86.         </div>
87.     </div>
88. <?php endforeach; ?>
89. </div>
90.
91. </div>
92. </div>
93. </div>
94. </div>

```

application\views\views_course\notif

```

1. <div class="main-content">
2.     <div class="container-fluid">
3.         <div class="col-12 list-item list-item-thumb">
4.             <div class="card d-flex flex-row mb-3">
5.                 <div class="d-flex flex-grow-1 min-width-zero card-content">

```

```

6.         <div class="card-body align-self-center d-flex flex-column flex-md-
row justify-content-between min-width-zero align-items-md-center">
7.             <p class="list-item-heading mb-1 truncate w-40 w-xs-
100"><?= $app['title']; ?></p>
8.             <p class="mb-1 text-small category w-15 w-xs-
100"><?= $app['name'] ?></p>
9.             <p class="mb-1 text-muted text-small date w-15 w-xs-100"></p>
10.            <div class="w-15 w-xs-100">
11.                <span class="badge badge-pill badge-
primary"><?= $app['category'] ?></span>
12.            </div>
13.            <div class="w-15 w-xs-100">
14.                <a href="<?= base_url('course/approve/') . $app['id']; ?>"
class="badge badge-pill badge-success">Approve</a>
15.            </div>
16.        </div>
17.    </div>
18. </div>
19. </div>
20. </div>
21. </div>

```

application\views\views_course\notifdetail

[illegible]

```

26.         <span class="badge badge-pill badge-
primary"><?= $app['category'] ?></span>
27.     </div>
28.     <div class="w-15 w-xs-100">
29.         <a href="<?= base_url('course/approve/'
) . $app['id']; ?>" class="badge badge-pill badge-success">Approve</a>
30.         <a href="" class="badge badge-
pill badge-danger reject" data-toggle="modal" data-target="#demoModal" data-
id="<?= $app['id']; ?>">Reject</a>
31.     </div>
32. </div>
33. </div>
34. </div>
35. </div>
36. <?php endforeach; ?>
37. </div>
38. </div>
39. <?php endif; ?>
40. <?php if ($waitapp) : ?>
41.     <div class="card">
42.         <div class="card-header">
43.             <h3>Courses Awaiting Approval</h3>
44.         </div>
45.         <div class="card-body">
46.             <?php foreach ($waitapp as $wa) : ?>
47.                 <div class="col-12 list-item list-item-thumb">
48.                     <div class="card d-flex flex-row mb-3">
49.                         <div class="d-flex flex-grow-1 min-width-zero card-
content">
50.                             <div class="card-body align-self-center d-
flex flex-column flex-md-row justify-content-between min-width-zero align-items-md-
center">
51.                                 <p class="list-item-heading mb-
1 truncate w-40 w-xs-100"><?= $wa['title']; ?></p>
52.                                 <p class="mb-1 text-small category w-15 w-
xs-100"></p>
53.                                 <p class="mb-1 text-muted text-
small date w-15 w-xs-100"></p>
54.                                 <div class="w-15 w-xs-100">
55.                                     <span class="badge badge-pill badge-
primary"><?= $wa['category'] ?></span>
56.                                 </div>
57.                                 <div class="w-15 w-xs-100">
58.                                     <span class="badge badge-pill badge-
danger">Waiting For Approval</span>
59.                                 </div>
60.                             </div>
61.                         </div>
62.                     </div>
63.                 </div>
64.             <?php endforeach; ?>
65.         </div>
66.     </div>
67. <?php endif; ?>
68. <?php if ($appreject) : ?>
69.     <div class="card">
70.         <div class="card-header">
71.             <h3>Rejected Course</h3>
72.         </div>
73.         <div class="card-body">
74.             <?php foreach ($appreject as $ar) : ?>

```

```

75.             <div class="col-12 list-item list-item-thumb">
76.                 <div class="card d-flex flex-row mb-3">
77.                     <div class="d-flex flex-grow-1 min-width-zero card-
content">
78.                         <div class="card-body align-self-center d-
flex flex-column flex-md-row justify-content-between min-width-zero align-items-md-
center">
79.                             <p class="list-item-heading mb-
1 truncate w-40 w-xs-100"><?= $ar['title']; ?></p>
80.                             <p class="mb-1 text-small category w-15 w-
xs-100"><?= $ar['reason'] ?></p>
81.                             <p class="mb-1 text-muted text-
small date w-15 w-xs-100"></p>
82.                             <div class="w-15 w-xs-100">
83.                                 <span class="badge badge-pill badge-
primary"><?= $ar['category'] ?></span>
84.                             </div>
85.                             <div class="w-15 w-xs-100">
86.                                 <span class="badge badge-pill badge-
danger">Rejected</span>
87.                             </div>
88.                             <div class="list-actions">
89.                                 <a href="<?= base_url('course/deleterej
ect/') . $ar['id']; ?>" class="btn-delete" data-value="data"><i class="ik ik-trash-
2"></i></a>
90.                             </div>
91.                         </div>
92.                     </div>
93.                 </div>
94.             </div>
95.             <?php endforeach; ?>
96.         </div>
97.     </div>
98. <?php endif; ?>
99. </div>
100. </div>
101. </div>
102.
103.     <div class="modal fade" id="demoModal" tabindex="-1" role="dialog" aria-
labelledby="demoModallabel" aria-hidden="true">
104.         <div class="modal-dialog" role="document">
105.             <div class="modal-content">
106.                 <div class="modal-header">
107.                     <h5 class="modal-title" id="demoModallabel">Reject</h5>
108.                     <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">x</span></button>
109.                 </div>
110.                 <form action="<?= base_url('course/reject') ?>" method="post">
111.                     <div class="modal-body">
112.                         <input type="hidden" name="user_id" id="user_id">
113.                         <input type="hidden" name="course_id" id="course_id">
114.                         <input type="hidden" name="id" id="id">
115.                         <div class="form-group">
116.                             <label for="">Reason <span class="text-
danger">*</span></label>
117.                             <input type="text" required maxlength="65" minlength="10
" class="form-control" name="reason" placeholder="Reason ..">
118.                         </div>
119.                     </div>
120.                 <div class="modal-footer">

```

```

121.         <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
122.         <button type="submit" class="btn btn-
primary">Save changes</button>
123.     </div>
124. </form>
125. </div>
126. </div>
127. </div>
128.
129. <script>
130.     var base_url = "<?= base_url() ?>"
131. </script>

```

application\views\views_course\reportprint

```

1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.     <meta charset="UTF-8">
6.     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8.     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css
/bootstrap.min.css" integrity="sha384-
Vkoo8x4CGsO3+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymou
s">
9.     <title>Report - <?= $activity['activity_name'] ?></title>
10.     <script>
11.         window.print()
12.     </script>
13. </head>
14.
15. <body>
16.     <div class="container">
17.         <div class="card mb-3">
18.             <div class="card-header">
19.                 <h5><?= $activity['title'] ?></h5>
20.             </div>
21.             <div class="card-body">
22.                 <p class="card-
text">Quiz Name <strong><?= $activity['activity_name'] ?></strong></p>
23.                 <p class="card-
text">Opened <strong><?= date('d F Y h:i A', $activity['start_date']) ?></strong></p>
24.                 <p class="card-
text">Closed <strong><?= date('d F Y h:i A', $activity['end_date']) ?></strong></p>
25.                 <p class="card-
text">Grade to pass: <strong><?= $activity['grade_to_pass'] ?></strong></p>
26.                 <p class="card-
text">Limit: <strong><?= $activity['limit_time'] . ' Minutes' ?></strong></p>
27.             </div>
28.         </div>
29.     </div>
30.
31.     <div class="row">
32.         <div class="col">
33.             <div class="card">
34.                 <div class="card-header">
35.                     <h3>Report User</h3>
36.                 </div>

```

```

37.         <div class="card-body">
38.             <table id="data_table" class="table">
39.                 <thead>
40.                     <tr>
41.                         <th>No</th>
42.                         <th>NPK</th>
43.                         <th>Name</th>
44.                         <th>Email</th>
45.                         <th>Grade</th>
46.                         <th>Result</th>
47.                     </thead>
48.                     <tbody>
49.                         <?php $i = 1;
50.                         foreach ($user as $u) :
51.
52.                             $actId = $this->uri->segment(3, 0);
53.                             $result = $this->quiz-
>getResultQuizByActIdUserId($actId, $u['id'])->row_array();
54.                             <?>
55.                             <tr>
56.                                 <td><?= $i++ ?></td>
57.                                 <td><?= $u['emplid'] ?></td>
58.                                 <td><?= $u['name'] ?></td>
59.                                 <td><?= $u['email'] ?></td>
60.                                 <?php if ($result['grade'] != '') : ?>
61.                                     <td><?= $result['grade'] ?></td>
62.                                 <?php else : ?>
63.                                     <td>Not Attempts</td>
64.                                 <?php endif; ?>
65.                                 <?php if ($result['grade'] > $activity['grade_t
o_pass']) : ?>
66.                                     <td>Passed</td>
67.                                 <?php else : ?>
68.                                     <td>Not Passed</td>
69.                                 <?php endif; ?>
70.                             </tr>
71.                         <?php endforeach; ?>
72.                     </tbody>
73.                 </table>
74.
75.             </div>
76.         </div>
77.     </div>
78. </div>
79. </div>
80.
81.     <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymou
s"></script>
82.     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymou
s"></script>
83.     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js
" integrity="sha384-
wFSDf2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6" crossorigin="anonymou
s"></script>
84.
85. </body>
86.
87. </html>

```

application\views\views_course\reportquiz

```
1. <div class="main-content">
2.
3.     <div class="container-fluid">
4.         <div class="page-header">
5.             <div class="row align-items-end">
6.                 <div class="col-lg-8">
7.                     <div class="page-header-title">
8.                         <i class="ik ik-edit bg-blue"></i>
9.                         <div class="d-inline">
10.                            <h5>Report Course</h5>
11.                            <span>Access by: Instructor</span>
12.                        </div>
13.                    </div>
14.                </div>
15.                <div class="col-lg-4">
16.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
17.                        <ol class="breadcrumb">
18.                            <li class="breadcrumb-item">
19.                                <a href="{%= base_url('course') ?%}"><i class="ik ik-
home"></i></a>
20.                            </li>
21.                            <li class="breadcrumb-
item"><a href="{%= base_url('course/open/' . $activity['course_id']) ?%}">My Course</a><
/li>
22.                            <li class="breadcrumb-item active" aria-
current="page">Quiz</li>
23.                        </ol>
24.                    </nav>
25.                </div>
26.            </div>
27.        </div>
28.
29.        <span id="id_ujian" data-key="{%= $encrypted_id ?%}"></span>
30.        <div class="card text-center">
31.            <div class="card-header">
32.                <h5><%= $activity['activity_name'] ?%></h5>
33.            </div>
34.            <div class="card-body">
35.                <p class="card-
text">This quiz will be open on <strong><%= date('d F Y h:i A', $activity['start_date']
) ?%></strong></p>
36.                <p class="card-
text">This quiz will be close on <strong><%= date('d F Y h:i A', $activity['end_date'])
?%></strong></p>
37.                <p class="card-
text">Grade to pass: <strong><%= $activity['grade_to_pass'] ?%></strong></p>
38.                <p class="card-
text">Limit: <strong><%= $activity['limit_time'] . ' Minutes' ?%></strong></p>
39.                <p class="card-
text">Total Question: <strong><%= $activity['total_question'] ?%></strong></p>
40.            </div>
41.        </div>
42.
43.        <a href="{%= base_url('activity/reportprint/') . $this->uri-
>segment(3, 0); ?%}" target="_blank" class="btn btn-info mb-3"><i class="fas fa-
print"></i> Print</a>
44.    <div class="row">
45.        <div class="col">
```

```

46.         <div class="card">
47.             <div class="card-header">
48.                 <h3>Report User</h3>
49.             </div>
50.             <div class="card-body">
51.                 <table id="data_table" class="table">
52.                     <thead>
53.                         <tr>
54.                             <th>No</th>
55.                             <th>NPK</th>
56.                             <th>Name</th>
57.                             <th>Email</th>
58.                             <th>Grade</th>
59.                             <th>Result</th>
60.                         </thead>
61.                         <tbody>
62.                             <?php $i = 1;
63.                             foreach ($user as $u) :
64.
65.                                 $actId = $this->uri->segment(3, 0);
66.                                 $result = $this->quiz-
>getResultQuizByActIdUserId($actId, $u['id'])->row_array();
67.                                 ?>
68.                                 <tr>
69.                                     <td><?= $i++ ?></td>
70.                                     <td><?= $u['emplid'] ?></td>
71.                                     <td><?= $u['name'] ?></td>
72.                                     <td><?= $u['email'] ?></td>
73.                                     <?php if (!$result) : ?>
74.                                         <td>Not Attempts</td>
75.                                     <?php else : ?>
76.                                         <td><?= $result['grade'] ?></td>
77.                                     <?php endif; ?>
78.                                     <?php if ($result) : ?>
79.                                         <?php if ($result['grade'] >= $activity['gr
ade_to_pass']) : ?>
80.                                             <td><span class="badge badge-
pill badge-success">Passed</span></td>
81.                                         <?php else : ?>
82.                                             <td><span class="badge badge-
pill badge-danger">Not Passed</span></td>
83.                                         <?php endif; ?>
84.                                         <?php else : ?>
85.                                             <td><span class="badge badge-pill badge-
danger">Not Passed</span></td>
86.                                         <?php endif; ?>
87.                                     </tr>
88.                                 <?php endforeach; ?>
89.                             </tbody>
90.                         </table>
91.
92.                     </div>
93.                 </div>
94.             </div>
95.         </div>
96.     </div>
97. </div>

```

application\views\views_manager\index

```

1. <!-- Awal Main Content -->
2. <div class="main-content">
3.     <div class="container-fluid">
4.         <div class="flash-data" data-flashdata="<?= $this->session-
>flashdata('message'); ?>"></div>
5.         <div class="flash-data-icon" data-icon="<?= $this->session-
>flashdata('icon'); ?>"></div>
6.         <div class="flash-data-text" data-text="<?= $this->session-
>flashdata('text'); ?>"></div>
7.         <div class="page-header">
8.             <div class="row align-items-end">
9.                 <div class="col-lg-8">
10.                    <div class="page-header-title">
11.                        <i class="ik ik-users bg-blue"></i>
12.                        <div class="d-inline">
13.                            <h5>Data Manager</h5>
14.                            <span>Access by: Administrator</span>
15.                        </div>
16.                    </div>
17.                </div>
18.                <div class="col-lg-4">
19.                    <nav class="breadcrumb-container" aria-label="breadcrumb">
20.                        <ol class="breadcrumb">
21.                            <li class="breadcrumb-item">
22.                                <a href="<?= base_url('course'); ?>"><i class="ik ik-
home"></i></a>
23.                                </li>
24.                                <li class="breadcrumb-item active" aria-
current="page">Data Users</li>
25.                            </ol>
26.                        </nav>
27.                    </div>
28.                </div>
29.            </div>
30.
31.            <div class="row">
32.                <div class="col-sm-12">
33.                    <button class="btn btn-primary mb-4" data-toggle="modal" data-
target="#demoModal"> <i class=" ik ik-plus"></i>Add Manager</button>
34.                </div>
35.            </div>
36.
37.            <div class="row">
38.                <div class="col-sm-12">
39.                    <div class="card">
40.                        <div class="card-header">
41.                            <h3>Data Users</h3>
42.                        </div>
43.                        <div class="card-body">
44.                            <table id="data_table" class="table">
45.                                <thead>
46.                                    <tr>
47.                                        <th>No</th>
48.                                        <th class="nosort">Organization</th>
49.                                        <th>Manager</th>
50.                                        <th class="nosort mr-0"></th>
51.                                    </tr>
52.                                </thead>
53.                                <tbody>
54.
55.                                <?php

```

```

56.             $i = 1;
57.             foreach ($managers as $m) : ?>
58.                 <tr>
59.                     <td><?=$i++; ?></td>
60.                     <td><?=$m['org']; ?></td>
61.                     <td><?=$m['name']; ?></td>
62.                     <td>
63.                         <div class="table-actions">
64.                             <a href="<?= base_url('manager/delete/'
) . $m['id']; ?>" class="btn-delete" data-value="manager"><i class="ik ik-trash-
2"></i></a>
65.                         </div>
66.                     </td>
67.                 </tr>
68.                 <?php endforeach; ?>
69.             </tbody>
70.         </table>
71.
72.     </div>
73. </div>
74. </div>
75. </div>
76.
77. </div>
78. </div>
79. <!-- Akhir Main Content -->
80.
81.
82.
83. <div class="modal fade" id="demoModal" tabindex="-1" role="dialog" aria-
labelledby="demoModallabel" aria-hidden="true">
84.     <div class="modal-dialog" role="document">
85.         <div class="modal-content">
86.             <form action="<?= base_url('manager/add') ?>" method="post">
87.                 <div class="modal-header">
88.                     <h5 class="modal-title" id="demoModallabel">Relationship Manager-
Organization</h5>
89.                     <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">x</span></button>
90.                 </div>
91.                 <div class="modal-body">
92.                     <div class="form-group">
93.                         <label for="org">Organization</label>
94.                         <select name="org_id" id="org" class="form-
control select2" style="width:400px">
95.                             <option value="" selected hidden>Choose ..</option>
96.                             <?php foreach ($org as $o) : ?>
97.                                 <?php if ($this->Mgr->checkMgr($o['id']) > 0) {
98.                                     continue;
99.                                 } ?>
100.                                <option value="<?= $o['id'] ?>"><?= $o['org'] ?>
</option>
101.                                <?php endforeach; ?>
102.                            </select>
103.                        </div>
104.                        <div class="form-group">
105.                            <label for="manager">Manager</label><br>
106.                            <select name="manager_id" id="manager" class="form-
control select2" style="width:400px">
107.                                <option selected hidden>Choose ..</option>
108.                                </select>

```

```

109.             </div>
110.         </div>
111.         <div class="modal-footer">
112.             <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
113.             <button type="submit" class="btn btn-
primary">Save</button>
114.         </div>
115.     </form>
116. </div>
117. </div>
118. </div>

```

application\views\views_manager\ manager

```

1.     <option value="" selected hidden>Choose ..</option>
2. <?php foreach ($users as $u) : ?>
3.     <?php if ($this->Mgr->checkMgr(null, $u['id']) > 0) {
4.         continue;
5.     } ?>
6.     <option value="<?= $u['id'] ?>"><?= $u['name'] ?></option>
7. <?php endforeach; ?>

```

css\ chating

```

1. img {
2.     max-width: 100%;
3. }
4.
5. .inbox_people {
6.     background: #f8f8f8 none repeat scroll 0 0;
7.     overflow: hidden;
8.     border: 1px solid #c4c4c4;
9.     margin-bottom: 20px !important;
10. }
11.
12. .inbox_msg {
13.     border: 1px solid #c4c4c4;
14.     clear: both;
15.     overflow: hidden;
16. }
17.
18. .top_spac {
19.     margin: 20px 0 0;
20. }
21.
22. .chat_list:hover {
23.     cursor: pointer;
24. }
25.
26. .recent_heading {
27.     float: left;
28.     width: 40%;
29. }
30.
31. .srch_bar {
32.     display: inline-block;

```

```
33.     text-align: right;
34.     width: 60%;
35. }
36.
37. .headind_srch {
38.     padding: 10px 29px 10px 20px;
39.     overflow: hidden;
40.     border-bottom: 1px solid #c4c4c4;
41. }
42.
43. .recent_heading h4 {
44.     color: #05728f;
45.     font-size: 21px;
46.     margin: auto;
47. }
48.
49. .srch_bar input {
50.     border: 1px solid #cdcdcd;
51.     border-width: 0 0 1px 0;
52.     width: 80%;
53.     padding: 2px 0 4px 6px;
54.     background: none;
55. }
56.
57. .srch_bar .input-group-addon button {
58.     background: rgba(0, 0, 0, 0) none repeat scroll 0 0;
59.     border: medium none;
60.     padding: 0;
61.     color: #707070;
62.     font-size: 18px;
63. }
64.
65. .srch_bar .input-group-addon {
66.     margin: 0 0 0 -27px;
67. }
68.
69. .chat_ib h5 {
70.     font-size: 15px;
71.     color: #464646;
72.     margin: 0 0 8px 0;
73. }
74.
75. .chat_ib h5 span {
76.     font-size: 13px;
77.     /* float: right; */
78. }
79.
80. .chat_ib p {
81.     font-size: 14px;
82.     color: #989898;
83.     margin: auto
84. }
85.
86.
87. .chat_img{
88.     float: left;
89. }
90.
91. .chat_img img {
92.     width: 40px;
93.     height: 40px;
```

```
94.     border-radius: 100%;
95. }
96.
97. .chat_ib {
98.     float: left;
99.     padding: 0 0 0 15px;
100.     width: 88%;
101. }
102.
103.     .chat_people {
104.         overflow: hidden;
105.         clear: both;
106.     }
107.
108.     .chat_list {
109.         border-bottom: 1px solid #c4c4c4;
110.         margin: 0;
111.         padding: 18px 16px 10px;
112.     }
113.
114.     .inbox_chat {
115.         height: 435px;
116.         width: 100%;
117.         overflow-y: scroll;
118.     }
119.
120.     .active_chat {
121.         background: #ebebeb;
122.     }
123.
124.     .chat_list:active {
125.         background: #ebebeb;
126.     }
127.
128.     .incoming_msg_img {
129.         display: inline-block;
130.         width: 6%;
131.         margin: 26px 0 26px;
132.     }
133.
134.     .received_msg {
135.         display: inline-block;
136.         padding: 0 0 0 10px;
137.         vertical-align: top;
138.         width: 92%;
139.     }
140.
141.     .received_withd_msg p {
142.         background: #ebebeb none repeat scroll 0 0;
143.         border-radius: 3px;
144.         color: #646464;
145.         font-size: 14px;
146.         margin: 0;
147.         padding: 5px 10px 5px 12px;
148.         width: 100%;
149.     }
150.
151.     .time_date {
152.         color: #747474;
153.         display: block;
154.         font-size: 12px;
```

```
155.         margin: 8px 0 0;
156.     }
157.
158.     .received_withd_msg {
159.         width: 57%;
160.     }
161.
162.     .msgs {
163.         padding: 30px 15px 7px 25px;
164.         border: 1px solid #c4c4c4;
165.     }
166.
167.     .sent_msg p {
168.         background: #05728f none repeat scroll 0 0;
169.         border-radius: 3px;
170.         font-size: 14px;
171.         margin: 0;
172.         color: #fff;
173.         padding: 5px 10px 5px 12px;
174.         width: 100%;
175.     }
176.
177.     .outgoing_msg {
178.         overflow: hidden;
179.         margin: 0 0 26px;
180.     }
181.
182.     .sent_msg {
183.         float: right;
184.         width: 46%;
185.     }
186.
187.     .input_msg_write input {
188.         background: rgba(0, 0, 0, 0) none repeat scroll 0 0;
189.         border: medium none;
190.         color: #4c4c4c;
191.         font-size: 15px;
192.         min-height: 48px;
193.         width: 100%;
194.     }
195.
196.     .type_msg {
197.         border-top: 1px solid #c4c4c4;
198.         position: relative;
199.     }
200.
201.     .msg_send_btn {
202.         background: #05728f none repeat scroll 0 0;
203.         border: medium none;
204.         border-radius: 50%;
205.         color: #fff;
206.         cursor: pointer;
207.         font-size: 17px;
208.         height: 33px;
209.         position: absolute;
210.         right: 0;
211.         top: 11px;
212.         width: 33px;
213.     }
214.
215.     .messaging {
```

```

216.         padding: 0 0 50px 0;
217.     }
218.
219.     .msg_history {
220.         height: 400px;
221.         overflow-y: auto;
222.     }

```

css\style

```

1.  .card .card-body .item-activity a i{
2.      position: relative;
3.      margin-right: 10px;
4.      font-size: 30px;
5.      color: red;
6.  }
7.
8.  .card .card-body .item-activity{
9.      margin-bottom: 15px;
10.     line-height: 15px;
11. }
12. .card .card-body .item-actions:hover{
13.     border-bottom: 1px solid black;
14.     width: 900px;
15.     text-align: right;
16. }
17. .card .card-body .item-actions{
18.     float: right;
19.     margin-top: -30px;
20.     font-size: 17px;
21. }
22. .card .card-body .item-actions a{
23.     margin-left: 7px;
24. }
25. .card .card-body .item-activity a span{
26.     font-size: 16px;
27. }
28.
29. .form-horizontal .form-group .select2 {
30.     width: 570px;
31. }

```

system\core\Config

```

1.  <?php
2.  /**
3.   * CodeIgniter
4.   *
5.   * An open source application development framework for PHP
6.   *
7.   * This content is released under the MIT License (MIT)
8.   *
9.   * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10.  *
11.  * Permission is hereby granted, free of charge, to any person obtaining a copy
12.  * of this software and associated documentation files (the "Software"), to deal
13.  * in the Software without restriction, including without limitation the rights

```

```

14. * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15. * copies of the Software, and to permit persons to whom the Software is
16. * furnished to do so, subject to the following conditions:
17. *
18. * The above copyright notice and this permission notice shall be included in
19. * all copies or substantial portions of the Software.
20. *
21. * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22. * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23. * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24. * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25. * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26. * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27. * THE SOFTWARE.
28. *
29. * @package CodeIgniter
30. * @author EllisLab Dev Team
31. * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32. * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33. * @license https://opensource.org/licenses/MIT MIT License
34. * @link https://codeigniter.com
35. * @since Version 1.0.0
36. * @filesource
37. */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.
40. /**
41. * Config Class
42. *
43. * This class contains functions that enable config files to be managed
44. *
45. * @package CodeIgniter
46. * @subpackage Libraries
47. * @category Libraries
48. * @author EllisLab Dev Team
49. * @link https://codeigniter.com/user\_guide/libraries/config.html
50. */
51. class CI_Config {
52.
53.     /**
54.      * List of all loaded config values
55.      *
56.      * @var array
57.      */
58.     public $config = array();
59.
60.     /**
61.      * List of all loaded config files
62.      *
63.      * @var array
64.      */
65.     public $is_loaded = array();
66.
67.     /**
68.      * List of paths to search when trying to load a config file.
69.      *
70.      * @used-by CI_Loader
71.      * @var array
72.      */
73.     public $_config_paths = array(APPPATH);

```

```

74.
75.     // -----
76.
77.     /**
78.      * Class constructor
79.      *
80.      * Sets the $config data from the primary config.php file as a class variable.
81.      *
82.      * @return void
83.      */
84.     public function __construct()
85.     {
86.         $this->config =& get_config();
87.
88.         // Set the base_url automatically if none was provided
89.         if (empty($this->config['base_url']))
90.         {
91.             if (isset($_SERVER['SERVER_ADDR']))
92.             {
93.                 if (strpos($_SERVER['SERVER_ADDR'], ':') !== FALSE)
94.                 {
95.                     $server_addr = '[' . $_SERVER['SERVER_ADDR'] . ']';
96.                 }
97.                 else
98.                 {
99.                     $server_addr = $_SERVER['SERVER_ADDR'];
100.                }
101.
102.                $base_url = (is_https() ? 'https' : 'http') . '://' . $server_addr
103.                    . substr($_SERVER['SCRIPT_NAME'], 0, strpos($_SERVER['SCRIPT_
NAME'], basename($_SERVER['SCRIPT_FILENAME'])));
104.            }
105.            else
106.            {
107.                $base_url = 'http://localhost/';
108.            }
109.
110.            $this->set_item('base_url', $base_url);
111.        }
112.
113.        log_message('info', 'Config Class Initialized');
114.    }
115.
116.    // -----
117.
118.    /**
119.     * Load Config File
120.     *
121.     * @param string $file Configuration file name
122.     * @param bool $use_sections Whether configuration values should
be loaded into their own section
123.     * @param bool $fail_gracefully Whether to just return FALSE or disp
lay an error message
124.     * @return bool TRUE if the file was loaded correctly or FALSE on failur
e
125.     */
126.     public function load($file = '', $use_sections = FALSE, $fail_gracefully = F
ALSE)
127.     {
128.         $file = ($file === '') ? 'config' : str_replace('.php', '', $file);
129.         $loaded = FALSE;

```

```

130.
131.         foreach ($this->_config_paths as $path)
132.         {
133.             foreach (array($file, ENVIRONMENT.DIRECTORY_SEPARATOR.$file) as $location)
134.             {
135.                 $file_path = $path.'config/'.$location.'.php';
136.                 if (in_array($file_path, $this->is_loaded, TRUE))
137.                 {
138.                     return TRUE;
139.                 }
140.
141.                 if ( ! file_exists($file_path))
142.                 {
143.                     continue;
144.                 }
145.
146.                 include($file_path);
147.
148.                 if ( ! isset($config) OR ! is_array($config))
149.                 {
150.                     if ($fail_gracefully === TRUE)
151.                     {
152.                         return FALSE;
153.                     }
154.
155.                     show_error('Your '.$file_path.' file does not appear to contain a valid configuration array.');
```

```

156.                 }
157.
158.                 if ($use_sections === TRUE)
159.                 {
160.                     $this->config[$file] = isset($this->config[$file])
161.                         ? array_merge($this->config[$file], $config)
162.                         : $config;
163.                 }
164.                 else
165.                 {
166.                     $this->config = array_merge($this->config, $config);
167.                 }
168.
169.                 $this->is_loaded[] = $file_path;
170.                 $config = NULL;
171.                 $loaded = TRUE;
172.                 log_message('debug', 'Config file loaded: '.$file_path);
173.             }
174.         }
175.
176.         if ($loaded === TRUE)
177.         {
178.             return TRUE;
179.         }
180.         elseif ($fail_gracefully === TRUE)
181.         {
182.             return FALSE;
183.         }
184.
185.         show_error('The configuration file '.$file.'.php does not exist.');
```

```

186.     }
187.
188.     // -----

```

```

189.
190.     /**
191.     * Fetch a config file item
192.     *
193.     * @param string $item Config item name
194.     * @param string $index Index name
195.     * @return string|null The configuration item or NULL if the item doesn't e
xist
196.     */
197.     public function item($item, $index = '')
198.     {
199.         if ($index == '')
200.         {
201.             return isset($this->config[$item]) ? $this->config[$item] : NULL;
202.         }
203.
204.         return isset($this->config[$index], $this-
>config[$index][$item]) ? $this->config[$index][$item] : NULL;
205.     }
206.
207.     // -----
208.
209.     /**
210.     * Fetch a config file item with slash appended (if not empty)
211.     *
212.     * @param string $item Config item name
213.     * @return string|null The configuration item or NULL if the item doesn't e
xist
214.     */
215.     public function slash_item($item)
216.     {
217.         if ( ! isset($this->config[$item]))
218.         {
219.             return NULL;
220.         }
221.         elseif (trim($this->config[$item]) === '')
222.         {
223.             return '';
224.         }
225.
226.         return rtrim($this->config[$item], '/') . '/';
227.     }
228.
229.     // -----
230.
231.     /**
232.     * Site URL
233.     *
234.     * Returns base_url . index_page [. uri_string]
235.     *
236.     * @uses CI_Config::uri_string()
237.     *
238.     * @param string|string[] $uri URI string or an array of segments
239.     * @param string $protocol
240.     * @return string
241.     */
242.     public function site_url($uri = '', $protocol = NULL)
243.     {
244.         $base_url = $this->slash_item('base_url');
245.
246.         if (isset($protocol))

```

```

247.         {
248.             // For protocol-relative links
249.             if ($protocol === '')
250.             {
251.                 $base_url = substr($base_url, strpos($base_url, '///'));
252.             }
253.             else
254.             {
255.                 $base_url = $protocol.substr($base_url, strpos($base_url, ':///')
256.             );
257.         }
258.
259.         if (empty($uri))
260.         {
261.             return $base_url.$this->item('index_page');
262.         }
263.
264.         $uri = $this->_uri_string($uri);
265.
266.         if ($this->item('enable_query_strings') === FALSE)
267.         {
268.             $suffix = isset($this->config['url_suffix']) ? $this-
>config['url_suffix'] : '';
269.
270.             if ($suffix !== '')
271.             {
272.                 if (($offset = strpos($uri, '?')) !== FALSE)
273.                 {
274.                     $uri = substr($uri, 0, $offset).$suffix.substr($uri, $offset
275.                 );
276.                 }
277.                 else
278.                 {
279.                     $uri .= $suffix;
280.                 }
281.
282.                 return $base_url.$this->slash_item('index_page').$uri;
283.             }
284.             elseif (strpos($uri, '?') === FALSE)
285.             {
286.                 $uri = '?'.$uri;
287.             }
288.
289.             return $base_url.$this->item('index_page').$uri;
290.         }
291.
292.         // -----
293.
294.         /**
295.          * Base URL
296.          *
297.          * Returns base_url [. uri_string]
298.          *
299.          * @uses    CI_Config::_uri_string()
300.          *
301.          * @param    string|string[] $uri    URI string or an array of segments
302.          * @param    string $protocol
303.          * @return    string
304.          */

```

```

305.     public function base_url($uri = '', $protocol = NULL)
306.     {
307.         $base_url = $this->slash_item('base_url');
308.
309.         if (isset($protocol))
310.         {
311.             // For protocol-relative links
312.             if ($protocol === '')
313.             {
314.                 $base_url = substr($base_url, strpos($base_url, '///'));
315.             }
316.             else
317.             {
318.                 $base_url = $protocol.substr($base_url, strpos($base_url, ':///'))
319.             };
320.         }
321.
322.         return $base_url.$this->_uri_string($uri);
323.     }
324.
325.     // -----
326.
327.     /**
328.     * Build URI string
329.     *
330.     * @used-by CI_Config::site_url()
331.     * @used-by CI_Config::base_url()
332.     *
333.     * @param string|string[] $uri    URI string or an array of segments
334.     * @return string
335.     */
336.     protected function _uri_string($uri)
337.     {
338.         if ($this->item('enable_query_strings') === FALSE)
339.         {
340.             is_array($uri) && $uri = implode('/', $uri);
341.             return ltrim($uri, '/');
342.         }
343.         elseif (is_array($uri))
344.         {
345.             return http_build_query($uri);
346.         }
347.
348.         return $uri;
349.     }
350.
351.     // -----
352.
353.     /**
354.     * System URL
355.     *
356.     * @deprecated 3.0.0 Encourages insecure practices
357.     * @return string
358.     */
359.     public function system_url()
360.     {
361.         $x = explode('/', preg_replace('|/*(.+?)/*$|', '\\1', BASEPATH));
362.         return $this->slash_item('base_url').end($x).'/';
363.     }
364.

```

```

365.         // -----
366.
367.         /**
368.          * Set a config file item
369.          *
370.          * @param string $item Config item key
371.          * @param string $value Config item value
372.          * @return void
373.          */
374.         public function set_item($item, $value)
375.         {
376.             $this->config[$item] = $value;
377.         }
378.
379.     }

```

system\core\Controller

```

1. <?php
2. /**
3.  * CodeIgniter
4.  *
5.  * An open source application development framework for PHP
6.  *
7.  * This content is released under the MIT License (MIT)
8.  *
9.  * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10. *
11. * Permission is hereby granted, free of charge, to any person obtaining a copy
12. * of this software and associated documentation files (the "Software"), to deal
13. * in the Software without restriction, including without limitation the rights
14. * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15. * copies of the Software, and to permit persons to whom the Software is
16. * furnished to do so, subject to the following conditions:
17. *
18. * The above copyright notice and this permission notice shall be included in
19. * all copies or substantial portions of the Software.
20. *
21. * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22. * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23. * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24. * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25. * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26. * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27. * THE SOFTWARE.
28. *
29. * @package CodeIgniter
30. * @author EllisLab Dev Team
31. * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32. * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33. * @license https://opensource.org/licenses/MIT MIT License
34. * @link https://codeigniter.com
35. * @since Version 1.0.0
36. * @filesource
37. */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.

```

```

40. /**
41.  * Application Controller Class
42.  *
43.  * This class object is the super class that every library in
44.  * CodeIgniter will be assigned to.
45.  *
46.  * @package      CodeIgniter
47.  * @subpackage    Libraries
48.  * @category      Libraries
49.  * @author        EllisLab Dev Team
50.  * @link          https://codeigniter.com/user_guide/general/controllers.html
51.  */
52. class CI_Controller {
53.
54.     /**
55.      * Reference to the CI singleton
56.      *
57.      * @var object
58.      */
59.     private static $instance;
60.
61.     /**
62.      * CI_Loader
63.      *
64.      * @var CI_Loader
65.      */
66.     public $load;
67.
68.     /**
69.      * Class constructor
70.      *
71.      * @return void
72.      */
73.     public function __construct()
74.     {
75.         self::$instance =& $this;
76.
77.         // Assign all the class objects that were instantiated by the
78.         // bootstrap file (CodeIgniter.php) to local class variables
79.         // so that CI can run as one big super object.
80.         foreach (is_loaded() as $var => $class)
81.         {
82.             $this->$var =& load_class($class);
83.         }
84.
85.         $this->load =& load_class('Loader', 'core');
86.         $this->load->initialize();
87.         log_message('info', 'Controller Class Initialized');
88.     }
89.
90.     // -----
91.
92.     /**
93.      * Get the CI singleton
94.      *
95.      * @static
96.      * @return object
97.      */
98.     public static function &get_instance()
99.     {
100.         return self::$instance;

```

```

101.         }
102.
103.     }

```

system\core\compat\hash

```

1.  <?php
2.  /**
3.   * CodeIgniter
4.   *
5.   * An open source application development framework for PHP
6.   *
7.   * This content is released under the MIT License (MIT)
8.   *
9.   * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10.  *
11.  * Permission is hereby granted, free of charge, to any person obtaining a copy
12.  * of this software and associated documentation files (the "Software"), to deal
13.  * in the Software without restriction, including without limitation the rights
14.  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15.  * copies of the Software, and to permit persons to whom the Software is
16.  * furnished to do so, subject to the following conditions:
17.  *
18.  * The above copyright notice and this permission notice shall be included in
19.  * all copies or substantial portions of the Software.
20.  *
21.  * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22.  * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23.  * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24.  * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25.  * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26.  * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27.  * THE SOFTWARE.
28.  *
29.  * @package CodeIgniter
30.  * @author   EllisLab Dev Team
31.  * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32.  * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33.  * @license https://opensource.org/licenses/MIT MIT License
34.  * @link    https://codeigniter.com
35.  * @since   Version 3.0.0
36.  * @filesource
37.  */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.
40. /**
41.  * PHP ext/hash compatibility package
42.  *
43.  * @package      CodeIgniter
44.  * @subpackage   CodeIgniter
45.  * @category     Compatibility
46.  * @author       Andrey Andreev
47.  * @link         https://codeigniter.com/user\_guide/
48.  * @link         http://php.net/hash
49.  */
50.
51. // -----

```

```

52.
53. if (is_php('5.6'))
54. {
55.     return;
56. }
57.
58. // -----
59.
60. if ( ! function_exists('hash_equals'))
61. {
62.     /**
63.      * hash_equals()
64.      *
65.      * @link    http://php.net/hash_equals
66.      * @param   string  $known_string
67.      * @param   string  $user_string
68.      * @return  bool
69.      */
70.     function hash_equals($known_string, $user_string)
71.     {
72.         if ( ! is_string($known_string))
73.         {
74.             trigger_error('hash_equals(): Expected known_string to be a string, '.strto
lower(gettype($known_string)).' given', E_USER_WARNING);
75.             return FALSE;
76.         }
77.         elseif ( ! is_string($user_string))
78.         {
79.             trigger_error('hash_equals(): Expected user_string to be a string, '.strtol
ower(gettype($user_string)).' given', E_USER_WARNING);
80.             return FALSE;
81.         }
82.         elseif (($length = strlen($known_string)) !== strlen($user_string))
83.         {
84.             return FALSE;
85.         }
86.
87.         $diff = 0;
88.         for ($i = 0; $i < $length; $i++)
89.         {
90.             $diff |= ord($known_string[$i]) ^ ord($user_string[$i]);
91.         }
92.
93.         return ($diff === 0);
94.     }
95. }
96.
97. // -----
98.
99. if (is_php('5.5'))
100. {
101.     return;
102. }
103.
104. // -----
105.
106. if ( ! function_exists('hash_pbkdf2'))
107. {
108.     /**
109.      * hash_pbkdf2()
110.      *

```

```

111.         * @link      http://php.net/hash_pbkdf2
112.         * @param     string $algo
113.         * @param     string $password
114.         * @param     string $salt
115.         * @param     int $iterations
116.         * @param     int $length
117.         * @param     bool $raw_output
118.         * @return    string
119.         */
120.         function hash_pbkdf2($algo, $password, $salt, $iterations, $length = 0, $raw
_output = FALSE)
121.         {
122.             if ( ! in_array(strtolower($algo), hash_algos(), TRUE))
123.             {
124.                 trigger_error('hash_pbkdf2(): Unknown hashing algorithm: '.$algo, E_
USER_WARNING);
125.                 return FALSE;
126.             }
127.
128.             if (($type = gettype($iterations)) !== 'integer')
129.             {
130.                 if ($type === 'object' && method_exists($iterations, '__toString'))
131.                 {
132.                     $iterations = (string) $iterations;
133.                 }
134.
135.                 if (is_string($iterations) && is_numeric($iterations))
136.                 {
137.                     $iterations = (int) $iterations;
138.                 }
139.                 else
140.                 {
141.                     trigger_error('hash_pbkdf2() expects parameter 4 to be long, '.$
type.' given', E_USER_WARNING);
142.                     return NULL;
143.                 }
144.             }
145.
146.             if ($iterations < 1)
147.             {
148.                 trigger_error('hash_pbkdf2(): Iterations must be a positive integer:
'.$iterations, E_USER_WARNING);
149.                 return FALSE;
150.             }
151.
152.             if (($type = gettype($length)) !== 'integer')
153.             {
154.                 if ($type === 'object' && method_exists($length, '__toString'))
155.                 {
156.                     $length = (string) $length;
157.                 }
158.
159.                 if (is_string($length) && is_numeric($length))
160.                 {
161.                     $length = (int) $length;
162.                 }
163.                 else
164.                 {
165.                     trigger_error('hash_pbkdf2() expects parameter 5 to be long, '.$
type.' given', E_USER_WARNING);

```

```

166.         return NULL;
167.     }
168. }
169.
170.     if ($length < 0)
171.     {
172.         trigger_error('hash_pbkdf2(): Length must be greater than or equal t
o 0: '.$length, E_USER_WARNING);
173.         return FALSE;
174.     }
175.
176.     $hash_length = defined('MB_OVERLOAD_STRING')
177.         ? mb_strlen(hash($algo, NULL, TRUE), '8bit')
178.         : strlen(hash($algo, NULL, TRUE));
179.     emptyempty($length) && $length = $hash_length;
180.
181.     // Pre-hash password inputs longer than the algorithm's block size
182.     // (i.e. prepare HMAC key) to mitigate potential DoS attacks.
183.     static $block_sizes;
184.     emptyempty($block_sizes) && $block_sizes = array(
185.         'gost' => 32,
186.         'haval128,3' => 128,
187.         'haval160,3' => 128,
188.         'haval192,3' => 128,
189.         'haval224,3' => 128,
190.         'haval256,3' => 128,
191.         'haval128,4' => 128,
192.         'haval160,4' => 128,
193.         'haval192,4' => 128,
194.         'haval224,4' => 128,
195.         'haval256,4' => 128,
196.         'haval128,5' => 128,
197.         'haval160,5' => 128,
198.         'haval192,5' => 128,
199.         'haval224,5' => 128,
200.         'haval256,5' => 128,
201.         'md2' => 16,
202.         'md4' => 64,
203.         'md5' => 64,
204.         'ripemd128' => 64,
205.         'ripemd160' => 64,
206.         'ripemd256' => 64,
207.         'ripemd320' => 64,
208.         'salsa10' => 64,
209.         'salsa20' => 64,
210.         'sha1' => 64,
211.         'sha224' => 64,
212.         'sha256' => 64,
213.         'sha384' => 128,
214.         'sha512' => 128,
215.         'snefru' => 32,
216.         'snefru256' => 32,
217.         'tiger128,3' => 64,
218.         'tiger160,3' => 64,
219.         'tiger192,3' => 64,
220.         'tiger128,4' => 64,
221.         'tiger160,4' => 64,
222.         'tiger192,4' => 64,
223.         'whirlpool' => 64
224.     );
225.

```

```

226.         if (isset($block_sizes[$algo], $password[$block_sizes[$algo]]))
227.         {
228.             $password = hash($algo, $password, TRUE);
229.         }
230.
231.         $hash = '';
232.         // Note: Blocks are NOT 0-indexed
233.         for ($bc = (int) ceil($length / $hash_length), $bi = 1; $bi <= $bc; $bi++)
234.         {
235.             $key = $derived_key = hash_hmac($algo, $salt.pack('N', $bi), $password, TRUE);
236.             for ($i = 1; $i < $iterations; $i++)
237.             {
238.                 $derived_key ^= $key = hash_hmac($algo, $key, $password, TRUE);
239.             }
240.
241.             $hash .= $derived_key;
242.         }
243.
244.         // This is not RFC-compatible, but we're aiming for natural PHP compatibility
245.         if ( ! $raw_output)
246.         {
247.             $hash = bin2hex($hash);
248.         }
249.
250.         return defined('MB_OVERLOAD_STRING')
251.             ? mb_substr($hash, 0, $length, '8bit')
252.             : substr($hash, 0, $length);
253.     }
254. }

```

system\core\compat\mbstring

```

1.  <?php
2.  /**
3.   * CodeIgniter
4.   *
5.   * An open source application development framework for PHP
6.   *
7.   * This content is released under the MIT License (MIT)
8.   *
9.   * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10.  *
11.  * Permission is hereby granted, free of charge, to any person obtaining a copy
12.  * of this software and associated documentation files (the "Software"), to deal
13.  * in the Software without restriction, including without limitation the rights
14.  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15.  * copies of the Software, and to permit persons to whom the Software is
16.  * furnished to do so, subject to the following conditions:
17.  *
18.  * The above copyright notice and this permission notice shall be included in
19.  * all copies or substantial portions of the Software.
20.  *
21.  * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22.  * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

```

```

23. * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24. * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25. * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26. * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27. * THE SOFTWARE.
28. *
29. * @package CodeIgniter
30. * @author EllisLab Dev Team
31. * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32. * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (ht
    tps://bcit.ca/)
33. * @license https://opensource.org/licenses/MIT MIT License
34. * @link https://codeigniter.com
35. * @since Version 3.0.0
36. * @filesource
37. */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.
40. /**
41. * PHP ext/mbstring compatibility package
42. *
43. * @package CodeIgniter
44. * @subpackage CodeIgniter
45. * @category Compatibility
46. * @author Andrey Andreev
47. * @link https://codeigniter.com/user_guide/
48. * @link http://php.net/mbstring
49. */
50.
51. // -----
52.
53. if (MB_ENABLED === TRUE)
54. {
55.     return;
56. }
57.
58. // -----
59.
60. if ( ! function_exists('mb_strlen'))
61. {
62.     /**
63.      * mb_strlen()
64.      *
65.      * WARNING: This function WILL fall-back to strlen()
66.      * if iconv is not available!
67.      *
68.      * @link http://php.net/mb_strlen
69.      * @param string $str
70.      * @param string $encoding
71.      * @return int
72.      */
73.     function mb_strlen($str, $encoding = NULL)
74.     {
75.         if (ICONV_ENABLED === TRUE)
76.         {
77.             return iconv_strlen($str, isset($encoding) ? $encoding : config_item('chars
                et'));
78.         }
79.
80.         log_message('debug', 'Compatibility (mbstring): iconv_strlen() is not available
                , falling back to strlen().');

```

```

81.         return strlen($str);
82.     }
83. }
84.
85. // -----
86.
87. if ( ! function_exists('mb_strpos'))
88. {
89.     /**
90.      * mb_strpos()
91.      *
92.      * WARNING: This function WILL fall-back to strpos()
93.      * if iconv is not available!
94.      *
95.      * @link    http://php.net/mb_strpos
96.      * @param   string $haystack
97.      * @param   string $needle
98.      * @param   int $offset
99.      * @param   string $encoding
100.     * @return  mixed
101.     */
102.     function mb_strpos($haystack, $needle, $offset = 0, $encoding = NULL)
103.     {
104.         if (ICONV_ENABLED === TRUE)
105.         {
106.             return iconv_strpos($haystack, $needle, $offset, isset($encoding) ?
$encoding : config_item('charset'));
107.         }
108.
109.         log_message('debug', 'Compatibility (mbstring): iconv_strpos() is not av
ailable, falling back to strpos().');
110.         return strpos($haystack, $needle, $offset);
111.     }
112. }
113.
114. // -----
115.
116. if ( ! function_exists('mb_substr'))
117. {
118.     /**
119.      * mb_substr()
120.      *
121.      * WARNING: This function WILL fall-back to substr()
122.      * if iconv is not available.
123.      *
124.      * @link    http://php.net/mb_substr
125.      * @param   string $str
126.      * @param   int $start
127.      * @param   int $length
128.      * @param   string $encoding
129.      * @return  string
130.     */
131.     function mb_substr($str, $start, $length = NULL, $encoding = NULL)
132.     {
133.         if (ICONV_ENABLED === TRUE)
134.         {
135.             isset($encoding) OR $encoding = config_item('charset');
136.             return iconv_substr(
137.                 $str,
138.                 $start,

```

```

139.             isset($length) ? $length : iconv_strlen($str, $encoding), // NUL
140.             $encoding
141.         );
142.     }
143.
144.     log_message('debug', 'Compatibility (mbstring): iconv_substr() is not available, falling back to substr().');
145.     return isset($length)
146.         ? substr($str, $start, $length)
147.         : substr($str, $start);
148. }
149. }

```

system\core\compat\password

```

1.  <?php
2.  /**
3.   * CodeIgniter
4.   *
5.   * An open source application development framework for PHP
6.   *
7.   * This content is released under the MIT License (MIT)
8.   *
9.   * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10.  *
11.  * Permission is hereby granted, free of charge, to any person obtaining a copy
12.  * of this software and associated documentation files (the "Software"), to deal
13.  * in the Software without restriction, including without limitation the rights
14.  * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15.  * copies of the Software, and to permit persons to whom the Software is
16.  * furnished to do so, subject to the following conditions:
17.  *
18.  * The above copyright notice and this permission notice shall be included in
19.  * all copies or substantial portions of the Software.
20.  *
21.  * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22.  * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23.  * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24.  * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25.  * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26.  * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27.  * THE SOFTWARE.
28.  *
29.  * @package CodeIgniter
30.  * @author  EllisLab Dev Team
31.  * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32.  * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33.  * @license https://opensource.org/licenses/MIT MIT License
34.  * @link    https://codeigniter.com
35.  * @since   Version 3.0.0
36.  * @filesource
37.  */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.
40. /**
41.  * PHP ext/standard/password compatibility package

```

```

42. *
43. * @package      CodeIgniter
44. * @subpackage   CodeIgniter
45. * @category     Compatibility
46. * @author       Andrey Andreev
47. * @link         https://codeigniter.com/user_guide/
48. * @link         http://php.net/password
49. */
50.
51. // -----
52.
53. if (is_php('5.5') OR ! defined('CRYPT_BLOWFISH') OR CRYPT_BLOWFISH !== 1 OR defined('HH
    VM_VERSION'))
54. {
55.     return;
56. }
57.
58. // -----
59.
60. defined('PASSWORD_BCRYPT') OR define('PASSWORD_BCRYPT', 1);
61. defined('PASSWORD_DEFAULT') OR define('PASSWORD_DEFAULT', PASSWORD_BCRYPT);
62.
63. // -----
64.
65. if ( ! function_exists('password_get_info'))
66. {
67.     /**
68.      * password_get_info()
69.      *
70.      * @link      http://php.net/password_get_info
71.      * @param     string $hash
72.      * @return     array
73.      */
74.     function password_get_info($hash)
75.     {
76.         return (strlen($hash) < 60 OR sscanf($hash, '$2y$d', $hash) !== 1)
77.             ? array('algo' => 0, 'algoName' => 'unknown', 'options' => array())
78.             : array('algo' => 1, 'algoName' => 'bcrypt', 'options' => array('cost' => $
    hash));
79.     }
80. }
81.
82. // -----
83.
84. if ( ! function_exists('password_hash'))
85. {
86.     /**
87.      * password_hash()
88.      *
89.      * @link      http://php.net/password_hash
90.      * @param     string $password
91.      * @param     int $algo
92.      * @param     array $options
93.      * @return     mixed
94.      */
95.     function password_hash($password, $algo, array $options = array())
96.     {
97.         static $func_overload;
98.         isset($func_overload) OR $func_overload = (extension_loaded('mbstring') && ini_
    get('mbstring.func_overload'));
99.

```

```

100.         if ($algo != 1)
101.         {
102.             trigger_error('password_hash(): Unknown hashing algorithm: ' . (int) $
                algo, E_USER_WARNING);
103.             return NULL;
104.         }
105.
106.         if (isset($options['cost']) && ($options['cost'] < 4 OR $options['cost']
                > 31))
107.         {
108.             trigger_error('password_hash(): Invalid bcrypt cost parameter specif
                ied: ' . (int) $options['cost'], E_USER_WARNING);
109.             return NULL;
110.         }
111.
112.         if (isset($options['salt']) && ($saltlen = ($func_overload ? mb_strlen($
                options['salt'], '8bit') : strlen($options['salt']))) < 22)
113.         {
114.             trigger_error('password_hash(): Provided salt is too short: ' . $saltl
                en . ' expecting 22', E_USER_WARNING);
115.             return NULL;
116.         }
117.         elseif ( ! isset($options['salt']))
118.         {
119.             if (function_exists('random_bytes'))
120.             {
121.                 try
122.                 {
123.                     $options['salt'] = random_bytes(16);
124.                 }
125.                 catch (Exception $e)
126.                 {
127.                     log_message('error', 'compat/password: Error while trying to
                        use random_bytes(): ' . $e->getMessage());
128.                     return FALSE;
129.                 }
130.             }
131.             elseif (defined('MCRYPT_DEV_URANDOM'))
132.             {
133.                 $options['salt'] = mcrypt_create_iv(16, MCRYPT_DEV_URANDOM);
134.             }
135.             elseif (DIRECTORY_SEPARATOR === '/' && (is_readable($dev = '/dev/ara
                ndom') OR is_readable($dev = '/dev/urandom')))
136.             {
137.                 if (($fp = fopen($dev, 'rb')) === FALSE)
138.                 {
139.                     log_message('error', 'compat/password: Unable to open ' . $dev
                        . ' for reading.');
```

```

151.         log_message('error', 'compat/password: Error while readi
ng from '.$dev.'.');
152.         return FALSE;
153.     }
154.     $options['salt'] .= $read;
155. }
156.
157.     fclose($fp);
158. }
159. elseif (function_exists('openssl_random_pseudo_bytes'))
160. {
161.     $is_secure = NULL;
162.     $options['salt'] = openssl_random_pseudo_bytes(16, $is_secure);
163.
164.     if ($is_secure !== TRUE)
165.     {
166.         log_message('error', 'compat/password: openssl_random_pseudo
_bytes() set the $crypto_strong flag to FALSE');
167.         return FALSE;
168.     }
169. else
170. {
171.     log_message('error', 'compat/password: No CSPRNG available.');
```

```

206.
207.         if ($algo !== $info['algo'])
208.         {
209.             return TRUE;
210.         }
211.         elseif ($algo === 1)
212.         {
213.             $options['cost'] = isset($options['cost']) ? (int) $options['cost']
214. : 10;
215.             return ($info['options']['cost'] !== $options['cost']);
216.         }
217.         // Odd at first glance, but according to a comment in PHP's own unit tes
218.         // because it is an unknown algorithm - it's valid and therefore doesn't
219.         // need rehashing.
220.         return FALSE;
221.     }
222. }
223.
224. // -----
225.
226. if ( ! function_exists('password_verify'))
227. {
228.     /**
229.      * password_verify()
230.      *
231.      * @link    http://php.net/password_verify
232.      * @param   string $password
233.      * @param   string $hash
234.      * @return  bool
235.      */
236.     function password_verify($password, $hash)
237.     {
238.         if (strlen($hash) !== 60 OR strlen($password = crypt($password, $hash))
239. !== 60)
240.         {
241.             return FALSE;
242.         }
243.         $compare = 0;
244.         for ($i = 0; $i < 60; $i++)
245.         {
246.             $compare |= (ord($password[$i]) ^ ord($hash[$i]));
247.         }
248.
249.         return ($compare === 0);
250.     }
251. }

```

system\core\compat\standard

```

1. <?php
2. /**
3.  * CodeIgniter
4.  *
5.  * An open source application development framework for PHP

```

```

6.  *
7.  * This content is released under the MIT License (MIT)
8.  *
9.  * Copyright (c) 2014 - 2019, British Columbia Institute of Technology
10. *
11. * Permission is hereby granted, free of charge, to any person obtaining a copy
12. * of this software and associated documentation files (the "Software"), to deal
13. * in the Software without restriction, including without limitation the rights
14. * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
15. * copies of the Software, and to permit persons to whom the Software is
16. * furnished to do so, subject to the following conditions:
17. *
18. * The above copyright notice and this permission notice shall be included in
19. * all copies or substantial portions of the Software.
20. *
21. * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
22. * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
23. * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
24. * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
25. * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
26. * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
27. * THE SOFTWARE.
28. *
29. * @package CodeIgniter
30. * @author EllisLab Dev Team
31. * @copyright Copyright (c) 2008 - 2014, EllisLab, Inc. (https://ellislab.com/)
32. * @copyright Copyright (c) 2014 - 2019, British Columbia Institute of Technology (https://bcit.ca/)
33. * @license https://opensource.org/licenses/MIT MIT License
34. * @link https://codeigniter.com
35. * @since Version 3.0.0
36. * @filesource
37. */
38. defined('BASEPATH') OR exit('No direct script access allowed');
39.
40. /**
41.  * PHP ext/standard compatibility package
42.  *
43.  * @package CodeIgniter
44.  * @subpackage CodeIgniter
45.  * @category Compatibility
46.  * @author Andrey Andreev
47.  * @link https://codeigniter.com/user\_guide/
48.  */
49.
50. // -----
51.
52. if (is_php('5.5'))
53. {
54.     return;
55. }
56.
57. // -----
58.
59. if ( ! function_exists('array_column'))
60. {
61.     /**
62.      * array_column()
63.      *
64.      * @link http://php.net/array\_column
65.      * @param array $array

```

```

66.      * @param mixed $column_key
67.      * @param mixed $index_key
68.      * @return array
69.      */
70.      function array_column(array $array, $column_key, $index_key = NULL)
71.      {
72.          if ( ! in_array($type = gettype($column_key), array('integer', 'string', 'NULL'
73.          ), TRUE))
74.          {
75.              if ($type === 'double')
76.              {
77.                  $column_key = (int) $column_key;
78.              }
79.              elseif ($type === 'object' && method_exists($column_key, '__toString'))
80.              {
81.                  $column_key = (string) $column_key;
82.              }
83.              else
84.              {
85.                  trigger_error('array_column(): The column key should be either a string
86.                  or an integer', E_USER_WARNING);
87.                  return FALSE;
88.              }
89.          }
90.          if ( ! in_array($type = gettype($index_key), array('integer', 'string', 'NULL'
91.          ), TRUE))
92.          {
93.              if ($type === 'double')
94.              {
95.                  $index_key = (int) $index_key;
96.              }
97.              elseif ($type === 'object' && method_exists($index_key, '__toString'))
98.              {
99.                  $index_key = (string) $index_key;
100.             }
101.             else
102.             {
103.                 trigger_error('array_column(): The index key should be either a
104.                 string or an integer', E_USER_WARNING);
105.                 return FALSE;
106.             }
107.         }
108.         $result = array();
109.         foreach ($array as $a)
110.         {
111.             if ($column_key === NULL)
112.             {
113.                 $value = $a;
114.             }
115.             elseif (is_array($a) && array_key_exists($column_key, $a))
116.             {
117.                 $value = $a[$column_key];
118.             }
119.             else
120.             {
121.                 continue;
122.             }
123.             if ($index_key === NULL OR ! array_key_exists($index_key, $a))

```

```

123.         {
124.             $result[] = $value;
125.         }
126.         else
127.         {
128.             $result[$a[$index_key]] = $value;
129.         }
130.     }
131.
132.     return $result;
133. }
134. }
135.
136. // -----
137.
138. if (is_php('5.4'))
139. {
140.     return;
141. }
142.
143. // -----
144.
145. if ( ! function_exists('hex2bin'))
146. {
147.     /**
148.      * hex2bin()
149.      *
150.      * @link    http://php.net/hex2bin
151.      * @param   string $data
152.      * @return   string
153.      */
154.     function hex2bin($data)
155.     {
156.         if (in_array($type = gettype($data), array('array', 'double', 'object',
157. 'resource'), TRUE))
158.         {
159.             if ($type === 'object' && method_exists($data, '__toString'))
160.             {
161.                 $data = (string) $data;
162.             }
163.             else
164.             {
165.                 trigger_error('hex2bin() expects parameter 1 to be string, '.$ty
166. pe.' given', E_USER_WARNING);
167.                 return NULL;
168.             }
169.         }
170.         if (strlen($data) % 2 !== 0)
171.         {
172.             trigger_error('Hexadecimal input string must have an even length', E
173. _USER_WARNING);
174.             return FALSE;
175.         }
176.         elseif ( ! preg_match('/^[0-9a-f]*$/i', $data))
177.         {
178.             trigger_error('Input string must be hexadecimal string', E_USER_WARN
179. ING);
180.             return FALSE;
181.         }
182.     }
183. }

```

```
180.         return pack('H*', $data);
181.     }
182. }
```