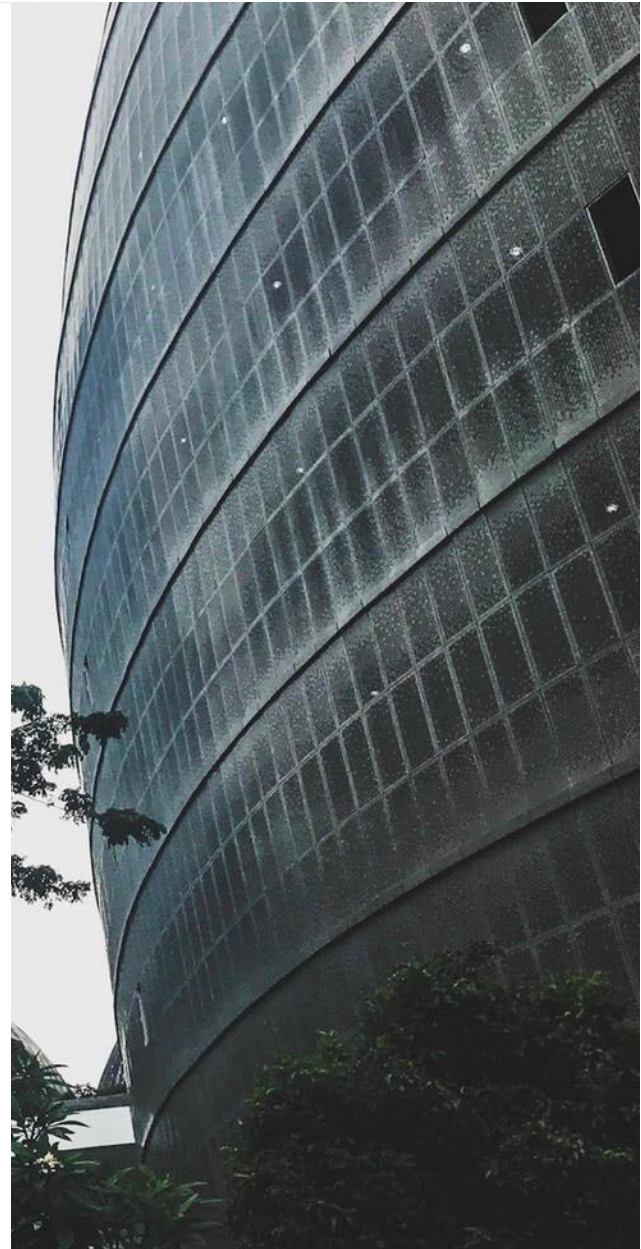


MODUL PRAKTIKUM

IS556 – WEB DESIGN AND DEVELOPMENT
PROGRAM SARJANA S1 SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA**

Gedung B Lantai 5, Kampus UMN
Jl. Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia
Telp: +62-21.5422.0808 (ext. 1803), email: ict.lab@umn.ac.id, web: umn.ac.id

MODULE 10

WEB DEVELOPMENT FRAMEWORK - CODEIGNITER



DESCRIPTION

Web application development using certain framework: CodeIgniter.

WEEKLY LEARNING OUTCOMES (SUB-LEARNING OUTCOMES)

1. SUB-CLO 11: Students are able to build web application using certain framework (C6)

PRACTICAL SUPPORT

1. Visual Studio Code

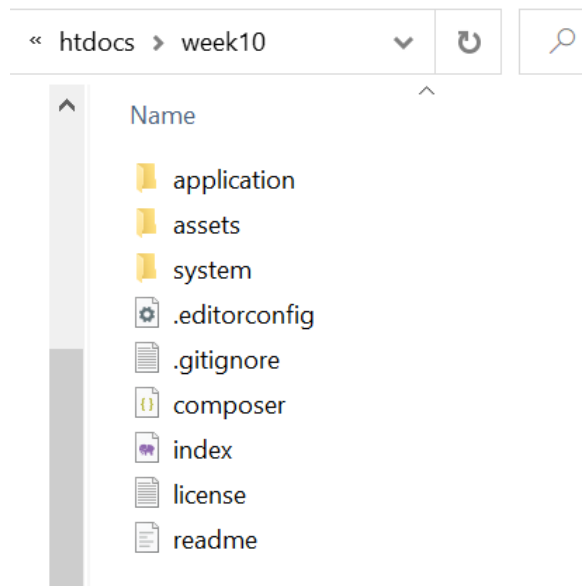
STEPS:

In this exercise, you will modify the last week's exercise with the concept of MVC (Model-View-Controller) using CodeIgniter. In this exercise, we use CodeIgniter version 3.1.13.

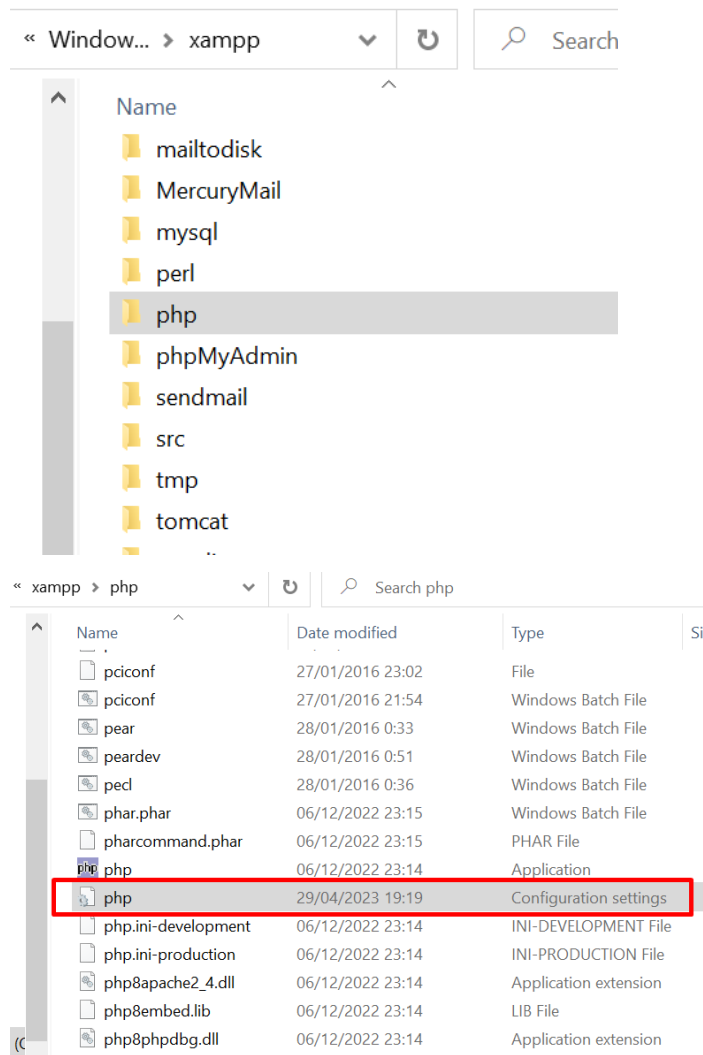
Task 1: Installation

1. You don't need to install CodeIgniter because we use the manual way. Extract the zip file named **bcit-ci-CodeIgniter-3.1.13-0-gbcb17eb** from the supported file and save into xampp → htdocs folder, then rename to **week10**.

You will see the structure of the CodeIgniter like the figure below.



2. Before starting the Apache and MySQL in xampp-control, open the folder named **php** in the xampp folder. Search for a file named **php.ini** and open using text editor or Notepad.



Find this snippet code `;extension=intl` and remove the `;`. After that, save and close the file.

```
extension=bz2
extension=curl
;extension=ffi
;extension=ftp
extension=fileinfo
;extension=gd
extension=gettext
;extension=gmp
extension=intl
;extension=imap
;extension=ldap
extension=mbstring
extension=exif ; Must be at
extension=mysqli
;extension=oci8_12c ; Use with
```

- Now, you can start the Apache and MySQL.
- Back to the CodeIgniter file (now week10), go to application → config, then open the file **config.php**
Set the **base_url** with adding the path <https://localhost/week10/>

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|   http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try to guess the protocol and
| path to your installation, but due to security concerns the hostname will
| be set to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
*/
$config['base_url'] = 'https://localhost/week10/';
```

Save the file, then open the browser and direct to <https://localhost/week10/>

5. If an error occurs, go to the folder week10 → system → core, and add this snipped code
#[\AllowDynamicProperties] to the files **URI.php, Router.php, Loader.php, Controller.php**.
 Go to system → database, open the **DB_driver.php** and add the snipped code too.

```
/**
 * URI Class
 *
 * Parses URIs and determines routing
 *
 * @package      CodeIgniter
 * @subpackage   Libraries
 * @category     URI
 * @author       EllisLab Dev Team
 * @link         https://codeigniter.com/userguide3/libraries/uri.html
 */
#[\AllowDynamicProperties]

class CI_URI {
```

```
/**
 * Application Controller Class
 *
 * This class object is the super class that every library in
 * CodeIgniter will be assigned to.
 *
 * @package      CodeIgniter
 * @subpackage    Libraries
 * @category      Libraries
 * @author        EllisLab Dev Team
 * @link          https://codeigniter.com/userguide3/general/controllers.html
 */

#[\AllowDynamicProperties]
class CI_Controller {
```

```
/**
 * Loader Class
 *
 * Loads framework components.
 *
 * @package      CodeIgniter
 * @subpackage    Libraries
 * @category      Loader
 * @author        EllisLab Dev Team
 * @link          https://codeigniter.com/userguide3/libraries/loader.html
 */

#[\AllowDynamicProperties]
class CI_Loader {
```

```
/**
 * Router Class
 *
 * Parses URIs and determines routing
 *
 * @package      CodeIgniter
 * @subpackage    Libraries
 * @category      Libraries
 * @author        EllisLab Dev Team
 * @link          https://codeigniter.com/userguide3/general/routing.html
 */

#[\AllowDynamicProperties]
class CI_Router {
```

```
/**
 * Database Driver Class
 *
 * This is the platform-independent base DB implementation class.
 * This class will not be called directly. Rather, the adapter
 * class for the specific database will extend and instantiate it.
 *
 * @package      CodeIgniter
 * @subpackage    Drivers
 * @category      Database
 * @author        EllisLab Dev Team
 * @link          https://codeigniter.com/userguide3/database/
 */

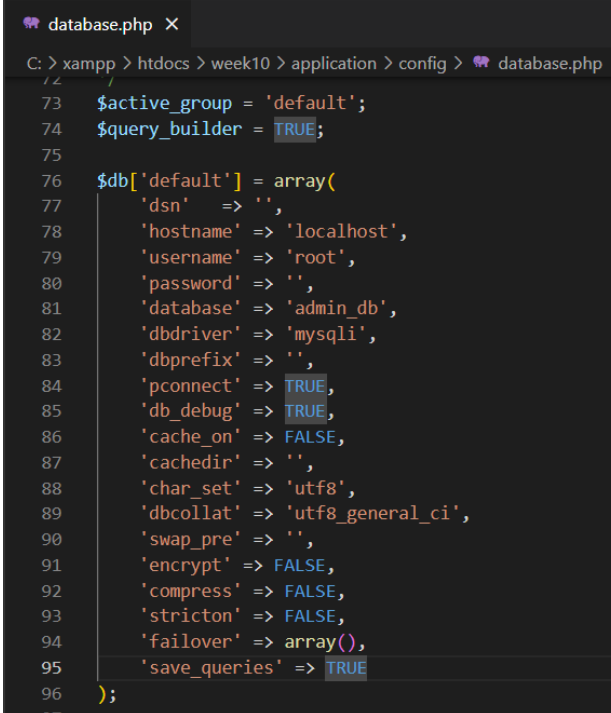
#[\AllowDynamicProperties]
abstract class CI_DB_driver {
```

Note: Look at the image for the code placement.

Task 2: Database Configuration

We will use the same database as the last week's exercise. So, you just need to configuration the database.

1. Go to week10 → application → config, open the file **database.php**.
2. Set the hostname, root, password, and database name. Don't forget to save the file.



```
database.php X
C: > xampp > htdocs > week10 > application > config > database.php
12
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' => 'admin_db',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => TRUE,
85     'db_debug' => TRUE,
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 );
97
```

Task 3: Model-View-Controller

1. Go to application → controllers, create a php file named **Users.php** in this folder.

```

Users.php X
C:\xampp\htdocs> week10 > application > controllers > Users.php
1  <!-- The controller consists of classes that are used to handle HTTP requests,
2  so all the user requests via views will be processed by the controller. -->
3
4  <?php
5
6  class Users extends CI_Controller {
7  // the class name must be the same as the php file name
8      public function __construct()
9      {
10         parent::__construct();
11         $this->load->model('user_model');
12         $this->load->helper('url_helper');
13     }
14
15     public function index()
16     {
17         $this->load->view('templates/header');
18         $this->load->view('users/login');
19         $this->load->view('templates/footer');
20     }
21
22     public function login_action()
23     {
24
25         $username = $this->input->post("email");
26         $password = $this->input->post("password");
27         $data = $this->user_model->get_login($username,$password);
28
29         foreach ($data as $val):
30             if ($val['user_role'] == 'admin'){
31                 $this->login_admin();
32             }else if($val['user_role'] == 'member'){
33                 //code here
34             }else if($val['user_role'] == ''){
35                 $this->load->view('templates/header');
36                 $this->load->view('users/login');
37                 $this->load->view('templates/footer');
38             }
39         endforeach;
40
41     }
42
43     public function login_admin()
44     {
45         $data['user_data'] = $this->user_model->get_users();
46         $data['title'] = 'User Data';
47
48         $this->load->view('templates/header');
49         $this->load->view('users/login_admin', $data);
50         $this->load->view('templates/footer');
51     }
52
53     public function login_member()
54     {
55         //code here
56     }
57
58 }

```

- Go to application → models, create a php file named **User_model.php** in this folder.

```
User_model.php X
C: > xampp > htdocs > week10 > application > models > User_model.php
1  <!-- Models are used to handle database connections. -->
2
3  <?php
4  class User_model extends CI_Model {
5      // The class name must be the same as the file name
6
7      public function __construct()
8      {
9          $this->load->database();
10     }
11
12     public function get_login($username, $password)
13     {
14         $query = $this->db->get_where('user_login', array('email' => $username, 'password' => md5($password)));
15         return $query->result_array();
16     }
17
18     public function get_users()
19     {
20         $query = $this->db->get('user_data');
21         return $query->result_array();
22     }
23
24 }
```

3. Go to application → views, copy-paste the files **footer.php** and **header.php** into the folder **templates**.
4. In the folder views, create a folder named “users”, then put the files **login.php**, **login_admin.php**, and **login_member.php** into the folder.
5. Re-open the file Users.php in the folder controller, modify the comment section "code here" to add a condition if logged in as a "member".

SUBMISSION

1. Submit files consisting of:
 - a. File project (include HTML file and all used objects (image, dll)).
2. Compress file (.zip) and named **COURSECODE_CLASS_NIM_NAME_WEEK KE-XX.zip** (example: IS556_A_13110310017_Monika Evelin Johan_Week-01.zip).

REFERENCES

- Robins, J. N. (2018). Learning Web Design: A Beginner’s Guide to HTML, CSS, JavaScript, and Web Graphics 5th Ed. O’Reilly Media.
- Adrian W. West, Steve Prettyman - Practical PHP 7, MySQL 8, and MariaDB Website Databases_ A Simplified Approach to Developing Database-Driven Websites (2018, Apress) - libgen.lc
- <https://codeigniter.com/>