

```
1. <?php
2.
3. class Homepage extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->view('templates/header', array('title' => 'CS164'));
8.         $this->load->view('homepage/index');
9.         $this->load->view('templates/footer');
10.    }
11.
12.    public function lectures()
13.    {
14.        $this->load->view('templates/header', array('title' => 'Lectures'));
15.        $this->load->view('homepage/lectures');
16.        $this->load->view('templates/footer');
17.    }
18.
19.    public function lecture($n)
20.    {
21.        $this->load->view('templates/header', array('title' => 'Lecture ' . $n));
22.        $this->load->view('homepage/lecture', array('n' => $n));
23.        $this->load->view('templates/footer');
24.    }
25.
26.    public function labs()
27.    {
28.        $this->load->view('templates/header', array('title' => 'Labs'));
29.        $this->load->view('homepage/labs');
30.        $this->load->view('templates/footer');
31.    }
32.
33.    public function lab($n)
34.    {
35.        $this->load->view('templates/header', array('title' => 'Lab ' . $n));
36.        $this->load->view('homepage/lab', array('n' => $n));
37.        $this->load->view('templates/footer');
38.    }
39. }
40.
41. ?>
```

```
1. <ul>
2.   <li><a href="labs">Labs</a></li>
3.   <li><a href="lectures">Lectures</a></li>
4.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/0/syllabus.pdf">Syllabus</a></li>
5. </ul>
```

1. <ul>
2.   <li><a href="https://www.cs164.net/Lab\_<?php echo \$n ?>">Activity</a></li>
3. </ul>

1. <ul>
2.   <li><a href="lab/1">Lab 1</a></li>
3.   <li><a href="lab/2">Lab 2</a></li>
4. </ul>

```
1. <ul>
2.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/slides<?php echo $n ?>.pdf">Slides</a></li>
3.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/lecture<?php echo $n ?>.mp4">Video</a></li>
4. </ul>
```

1. <ul>
2.   <li><a href="lecture/0">Lecture 0</a></li>
3.   <li><a href="lecture/1">Lecture 1</a></li>
4. </ul>

1. `</body>`
2. `</html>`

```
1. <!DOCTYPE html>
2.
3. <html>
4.   <head>
5.     <meta name="viewport" content="width=device-width">
6.     <title><?php echo htmlspecialchars($title) ?></title>
7.   </head>
8.   <body>
9.     <h1><?php echo htmlspecialchars($title) ?></h1>
```



```
1. <?php
2.
3. /*
4. *-----
5. * APPLICATION ENVIRONMENT
6. *-----
7. *
8. * You can load different configurations depending on your
9. * current environment. Setting the environment also influences
10. * things like logging and error reporting.
11. *
12. * This can be set to anything, but default usage is:
13. *
14. *     development
15. *     testing
16. *     production
17. *
18. * NOTE: If you change these, also change the error_reporting() code below
19. *
20. */
21.     define('ENVIRONMENT', 'development');
22. /*
23. *-----
24. * ERROR REPORTING
25. *-----
26. *
27. * Different environments will require different levels of error reporting.
28. * By default development will show errors but testing and live will hide them.
29. */
30.
31. if (defined('ENVIRONMENT'))
32. {
33.     switch (ENVIRONMENT)
34.     {
35.         case 'development':
36.             error_reporting(E_ALL);
37.             break;
38.
39.         case 'testing':
40.         case 'production':
41.             error_reporting(0);
42.             break;
43.
44.         default:
45.             exit('The application environment is not set correctly.');
```

```
49. /*
50. *-----
51. * SYSTEM FOLDER NAME
52. *-----
53. *
54. * This variable must contain the name of your "system" folder.
55. * Include the path if the folder is not in the same directory
56. * as this file.
57. *
58. */
59.     $system_path = '../system';
60.
61. /*
62. *-----
63. * APPLICATION FOLDER NAME
64. *-----
65. *
66. * If you want this front controller to use a different "application"
67. * folder then the default one you can set its name here. The folder
68. * can also be renamed or relocated anywhere on your server. If
69. * you do, use a full server path. For more info please see the user guide:
70. * http://codeigniter.com/user\_guide/general/managing\_apps.html
71. *
72. * NO TRAILING SLASH!
73. *
74. */
75.     $application_folder = '../application';
76.
77. /*
78. * -----
79. * DEFAULT CONTROLLER
80. * -----
81. *
82. * Normally you will set your default controller in the routes.php file.
83. * You can, however, force a custom routing by hard-coding a
84. * specific controller class/function here. For most applications, you
85. * WILL NOT set your routing here, but it's an option for those
86. * special instances where you might want to override the standard
87. * routing in a specific front controller that shares a common CI installation.
88. *
89. * IMPORTANT: If you set the routing here, NO OTHER controller will be
90. * callable. In essence, this preference limits your application to ONE
91. * specific controller. Leave the function name blank if you need
92. * to call functions dynamically via the URI.
93. *
94. * Un-comment the $routing array below to use this feature
95. *
96. */
```

```
97. // The directory name, relative to the "controllers" folder. Leave blank
98. // if your controller is not in a sub-folder within the "controllers" folder
99. // $routing['directory'] = '';
100.
101. // The controller class file name. Example: Mycontroller
102. // $routing['controller'] = '';
103.
104. // The controller function you wish to be called.
105. // $routing['function'] = '';
106.
107.
108. /*
109. * -----
110. * CUSTOM CONFIG VALUES
111. * -----
112. *
113. * The $assign_to_config array below will be passed dynamically to the
114. * config class when initialized. This allows you to set custom config
115. * items or override any default config values found in the config.php file.
116. * This can be handy as it permits you to share one application between
117. * multiple front controller files, with each file containing different
118. * config values.
119. *
120. * Un-comment the $assign_to_config array below to use this feature
121. *
122. */
123. // $assign_to_config['name_of_config_item'] = 'value of config item';
124.
125.
126.
127. // -----
128. // END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
129. // -----
130.
131. /*
132. * -----
133. * Resolve the system path for increased reliability
134. * -----
135. */
136.
137. // Set the current directory correctly for CLI requests
138. if (defined('STDIN'))
139. {
140.     chdir(dirname(__FILE__));
141. }
142.
143. if (realpath($system_path) !== FALSE)
144. {
```

```
145.     $system_path = realpath($system_path).'/';
146. }
147.
148. // ensure there's a trailing slash
149. $system_path = rtrim($system_path, '/').'/';
150.
151. // Is the system path correct?
152. if ( ! is_dir($system_path) )
153. {
154.     exit("Your system folder path does not appear to be set correctly. Please open the following file and correct this: ".pathinfo(
155.     __FILE__, PATHINFO_BASENAME));
156. }
157. /*
158. * -----
159. * Now that we know the path, set the main path constants
160. * -----
161. */
162. // The name of THIS file
163. define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
164.
165. // The PHP file extension
166. // this global constant is deprecated.
167. define('EXT', '.php');
168.
169. // Path to the system folder
170. define('BASEPATH', str_replace("\\", "/", $system_path));
171.
172. // Path to the front controller (this file)
173. define('FCPATH', str_replace(SELF, '', __FILE__));
174.
175. // Name of the "system folder"
176. define('SYSDIR', trim(strchr(trim(BASEPATH, '/'), '/'), '/'));
177.
178.
179. // The path to the "application" folder
180. if (is_dir($application_folder))
181. {
182.     define('APPPATH', $application_folder.'/');
183. }
184. else
185. {
186.     if ( ! is_dir(BASEPATH.$application_folder.'/'))
187.     {
188.         exit("Your application folder path does not appear to be set correctly. Please open the following file and correct this: ".SELF);
189.     }
190.
191.     define('APPPATH', BASEPATH.$application_folder.'/');
```

```
192.     }
193.
194. /*
195.  * -----
196.  * LOAD THE BOOTSTRAP FILE
197.  * -----
198.  *
199.  * And away we go...
200.  *
201. */
202. require_once BASEPATH.'core/CodeIgniter.php' ;
203.
204. /* End of file index.php */
205. /* Location: ./index.php */
```

1. mvc/9/README
- 2.
3. David J. Malan
4. malan@harvard.edu
- 5.
6. Computer Science 164
7. Harvard College
- 8.
9. Builds upon mvc/8, adding support for Labs.
- 10.
11. application/
  12. config/
    13. config.php - CodeIgniter's configuration
    14. routes.php - default routes
    15. ...
  16. controllers/
    17. homepage.php - application controller
  18. errors/
    19. ...
  20. views/
    21. homepage/
      22. index.php - a homepage for the course
      23. lecture.php - a lecture
      24. lectures.php - a list of lectures
  25. templates/
    26. footer.php - pages' footer
    27. header.php - pages' header
  28. index.php - front controller
  29. system/
    30. ...

```
1. <?php
2.
3. class Labs extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->view('templates/header', array('title' => 'Labs'));
8.         $this->load->view('labs/index');
9.         $this->load->view('templates/footer');
10.    }
11.
12.    public function lab($n)
13.    {
14.        $this->load->view('templates/header', array('title' => 'Lab ' . $n));
15.        $this->load->view('labs/lab', array('n' => $n));
16.        $this->load->view('templates/footer');
17.    }
18. }
19.
20. ?>
```

```
1. <?php
2.
3. class Lectures extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->view('templates/header', array('title' => 'Lectures'));
8.         $this->load->view('lectures/index');
9.         $this->load->view('templates/footer');
10.    }
11.
12.    public function lecture($n)
13.    {
14.        $this->load->view('templates/header', array('title' => 'Lecture ' . $n));
15.        $this->load->view('lectures/lecture', array('n' => $n));
16.        $this->load->view('templates/footer');
17.    }
18. }
19.
20. ?>
```



```
1. <?php
2.
3. class Welcome extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->view('templates/header', array('title' => 'CS164'));
8.         $this->load->view('welcome/index');
9.         $this->load->view('templates/footer');
10.    }
11. }
12.
13. ?>
```

```
1. <ul>
2.   <li><a href="labs/lab/1">Lab 1</a></li>
3.   <li><a href="labs/lab/2">Lab 2</a></li>
4. </ul>
```

1. <ul>
2.   <li><a href="https://www.cs164.net/Lab\_<?php echo \$n ?>">Activity</a></li>
3. </ul>

1. <ul>
2.   <li><a href="lectures/lecture/0">Lecture 0</a></li>
3.   <li><a href="lectures/lecture/1">Lecture 1</a></li>
4. </ul>

```
1. <ul>
2.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/slides<?php echo $n ?>.pdf">Slides</a></li>
3.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/lecture<?php echo $n ?>.mp4">Video</a></li>
4. </ul>
```

1. `</body>`
2. `</html>`

```
1. <!DOCTYPE html>
2.
3. <html>
4.   <head>
5.     <meta name="viewport" content="width=device-width">
6.     <title><?php echo htmlspecialchars($title) ?></title>
7.   </head>
8.   <body>
9.     <h1><?php echo htmlspecialchars($title) ?></h1>
```

```
1. <ul>
2.   <li><a href="labs">Labs</a></li>
3.   <li><a href="lectures">Lectures</a></li>
4.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/0/syllabus.pdf">Syllabus</a></li>
5. </ul>
```



```
1. <?php
2.
3. /*
4. *-----
5. * APPLICATION ENVIRONMENT
6. *-----
7. *
8. * You can load different configurations depending on your
9. * current environment. Setting the environment also influences
10. * things like logging and error reporting.
11. *
12. * This can be set to anything, but default usage is:
13. *
14. *     development
15. *     testing
16. *     production
17. *
18. * NOTE: If you change these, also change the error_reporting() code below
19. *
20. */
21.     define('ENVIRONMENT', 'development');
22. /*
23. *-----
24. * ERROR REPORTING
25. *-----
26. *
27. * Different environments will require different levels of error reporting.
28. * By default development will show errors but testing and live will hide them.
29. */
30.
31. if (defined('ENVIRONMENT'))
32. {
33.     switch (ENVIRONMENT)
34.     {
35.         case 'development':
36.             error_reporting(E_ALL);
37.             break;
38.
39.         case 'testing':
40.         case 'production':
41.             error_reporting(0);
42.             break;
43.
44.         default:
45.             exit('The application environment is not set correctly.');
```

```
49. /*
50. *-----
51. * SYSTEM FOLDER NAME
52. *-----
53. *
54. * This variable must contain the name of your "system" folder.
55. * Include the path if the folder is not in the same directory
56. * as this file.
57. *
58. */
59.     $system_path = '../system';
60.
61. /*
62. *-----
63. * APPLICATION FOLDER NAME
64. *-----
65. *
66. * If you want this front controller to use a different "application"
67. * folder then the default one you can set its name here. The folder
68. * can also be renamed or relocated anywhere on your server. If
69. * you do, use a full server path. For more info please see the user guide:
70. * http://codeigniter.com/user\_guide/general/managing\_apps.html
71. *
72. * NO TRAILING SLASH!
73. *
74. */
75.     $application_folder = '../application';
76.
77. /*
78. * -----
79. * DEFAULT CONTROLLER
80. * -----
81. *
82. * Normally you will set your default controller in the routes.php file.
83. * You can, however, force a custom routing by hard-coding a
84. * specific controller class/function here. For most applications, you
85. * WILL NOT set your routing here, but it's an option for those
86. * special instances where you might want to override the standard
87. * routing in a specific front controller that shares a common CI installation.
88. *
89. * IMPORTANT: If you set the routing here, NO OTHER controller will be
90. * callable. In essence, this preference limits your application to ONE
91. * specific controller. Leave the function name blank if you need
92. * to call functions dynamically via the URI.
93. *
94. * Un-comment the $routing array below to use this feature
95. *
96. */
```

```
97. // The directory name, relative to the "controllers" folder. Leave blank
98. // if your controller is not in a sub-folder within the "controllers" folder
99. // $routing['directory'] = '';
100.
101. // The controller class file name. Example: Mycontroller
102. // $routing['controller'] = '';
103.
104. // The controller function you wish to be called.
105. // $routing['function'] = '';
106.
107.
108. /*
109. * -----
110. * CUSTOM CONFIG VALUES
111. * -----
112. *
113. * The $assign_to_config array below will be passed dynamically to the
114. * config class when initialized. This allows you to set custom config
115. * items or override any default config values found in the config.php file.
116. * This can be handy as it permits you to share one application between
117. * multiple front controller files, with each file containing different
118. * config values.
119. *
120. * Un-comment the $assign_to_config array below to use this feature
121. *
122. */
123. // $assign_to_config['name_of_config_item'] = 'value of config item';
124.
125.
126.
127. // -----
128. // END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
129. // -----
130.
131. /*
132. * -----
133. * Resolve the system path for increased reliability
134. * -----
135. */
136.
137. // Set the current directory correctly for CLI requests
138. if (defined('STDIN'))
139. {
140.     chdir(dirname(__FILE__));
141. }
142.
143. if (realpath($system_path) !== FALSE)
144. {
```

```
145.     $system_path = realpath($system_path).'/';
146. }
147.
148. // ensure there's a trailing slash
149. $system_path = rtrim($system_path, '/').'/';
150.
151. // Is the system path correct?
152. if ( ! is_dir($system_path) )
153. {
154.     exit("Your system folder path does not appear to be set correctly. Please open the following file and correct this: ".pathinfo(
155.     __FILE__, PATHINFO_BASENAME));
156. }
157. /*
158. * -----
159. * Now that we know the path, set the main path constants
160. * -----
161. */
162. // The name of THIS file
163. define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
164.
165. // The PHP file extension
166. // this global constant is deprecated.
167. define('EXT', '.php');
168.
169. // Path to the system folder
170. define('BASEPATH', str_replace("\\", "/", $system_path));
171.
172. // Path to the front controller (this file)
173. define('FCPATH', str_replace(SELF, '', __FILE__));
174.
175. // Name of the "system folder"
176. define('SYSDIR', trim(strchr(trim(BASEPATH, '/'), '/'), '/'));
177.
178.
179. // The path to the "application" folder
180. if (is_dir($application_folder))
181. {
182.     define('APPPATH', $application_folder.'/');
183. }
184. else
185. {
186.     if ( ! is_dir(BASEPATH.$application_folder.'/'))
187.     {
188.         exit("Your application folder path does not appear to be set correctly. Please open the following file and correct this: ".SELF);
189.     }
190.
191.     define('APPPATH', BASEPATH.$application_folder.'/');
```

```
192.     }
193.
194. /*
195.  * -----
196.  * LOAD THE BOOTSTRAP FILE
197.  * -----
198.  *
199.  * And away we go...
200.  *
201. */
202. require_once BASEPATH.'core/CodeIgniter.php' ;
203.
204. /* End of file index.php */
205. /* Location: ./index.php */
```

---

1. mvc/10/README
- 2.
3. David J. Malan
4. malan@harvard.edu
- 5.
6. Computer Science 164
7. Harvard College
- 8.
9. Improves upon mvc/9, refactoring code into separate controllers.
- 10.
11. application/
  12. config/
    13. config.php - CodeIgniter's configuration
    14. routes.php - default routes
    15. ...
  16. controllers/
    17. labs.php - application controller
    18. lectures.php - application controller
    19. welcome.php - application controller
  20. errors/
    21. ...
  22. views/
    23. homepage/
      24. index.php - a homepage for the course
      25. lecture.php - a lecture
      26. lectures.php - a list of lectures
    27. templates/
      28. footer.php - pages' footer
      29. header.php - pages' header
  30. index.php - front controller
  31. system/
    32. ...

```
1. <?php
2.
3. class Labs extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->model('Lab');
8.         $labs = $this->Lab->get_labs();
9.         $this->load->view('templates/header', array('title' => 'Labs'));
10.        $this->load->view('labs/index', array('labs' => $labs));
11.        $this->load->view('templates/footer');
12.    }
13.
14.    public function lab($n)
15.    {
16.        $this->load->view('templates/header', array('title' => 'Lab ' . $n));
17.        $this->load->view('labs/lab', array('n' => $n));
18.        $this->load->view('templates/footer');
19.    }
20. }
21.
22. ?>
```

```
1. <?php
2.
3. class Lectures extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->model('Lecture');
8.         $lectures = $this->Lecture->get_lectures();
9.         $this->load->view('templates/header', array('title' => 'Lectures'));
10.        $this->load->view('lectures/index', array('lectures' => $lectures));
11.        $this->load->view('templates/footer');
12.    }
13.
14.    public function lecture($n)
15.    {
16.        $this->load->view('templates/header', array('title' => 'Lecture ' . $n));
17.        $this->load->view('lectures/lecture', array('n' => $n));
18.        $this->load->view('templates/footer');
19.    }
20. }
21.
22. ?>
```



```
1. <?php
2.
3. class Welcome extends CI_Controller {
4.
5.     public function index()
6.     {
7.         $this->load->view('templates/header', array('title' => 'CS164'));
8.         $this->load->view('welcome/index');
9.         $this->load->view('templates/footer');
10.    }
11. }
12.
13. ?>
```

```
1. <?php
2.
3. /**
4.  * @file
5.  *
6.  * Model for labs.
7.  *
8.  * @author David J. Malan <malan@harvard.edu>
9.  * @version 3.0
10. */
11.
12. class Lab extends CI_Model {
13.
14.     /**
15.      * Returns labs.
16.      *
17.      * @return array
18.      */
19.     public function get_labs()
20.     {
21.         $this->db->select('id, name')->from('labs')->order_by('id asc');
22.         return $this->db->get()->result();
23.     }
24. }
25.
26. ?>
```

```
1. <?php
2.
3. /**
4.  * @file
5.  *
6.  * Model for labs.
7.  *
8.  * @author David J. Malan <malan@harvard.edu>
9.  * @version 3.0
10. */
11.
12. class Lecture extends CI_Model {
13.
14.     /**
15.      * Returns lectures.
16.      *
17.      * @return array
18.      */
19.     public function get_lectures()
20.     {
21.         $this->db->select('id, name')->from('lectures')->order_by('id asc');
22.         return $this->db->get()->result();
23.     }
24. }
25.
26. ?>
```

```
1. <ul>
2.   <?php foreach ($labs as $lab): ?>
3.     <li><a href="labs/lab/<?= $lab->id ?>"><?= htmlspecialchars($lab->name) ?></a></li>
4.   <?php endforeach ?>
5. </ul>
```

1. <ul>
2.   <li><a href="https://www.cs164.net/Lab\_<?php echo \$n ?>">Activity</a></li>
3. </ul>

```
1. <ul>
2.   <?php foreach ($lectures as $lecture): ?>
3.     <li><a href="lectures/lecture/<?= $lecture->id ?>"><?= htmlspecialchars($lecture->name) ?></a></li>
4.   <?php endforeach ?>
5. </ul>
```

```
1. <ul>
2.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/slides<?php echo $n ?>.pdf">Slides</a></li>
3.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/<?php echo $n ?>/lecture<?php echo $n ?>.mp4">Video</a></li>
4. </ul>
```

1. `</body>`
2. `</html>`



```
1. <!DOCTYPE html>
2.
3. <html>
4.   <head>
5.     <meta name="viewport" content="width=device-width">
6.     <title><?php echo htmlspecialchars($title) ?></title>
7.   </head>
8.   <body>
9.     <h1><?php echo htmlspecialchars($title) ?></h1>
```

```
1. <ul>
2.   <li><a href="labs">Labs</a></li>
3.   <li><a href="lectures">Lectures</a></li>
4.   <li><a href="http://cdn.cs164.net/2012/spring/lectures/0/syllabus.pdf">Syllabus</a></li>
5. </ul>
```

```
1. <?php
2.
3. /*
4. *-----
5. * APPLICATION ENVIRONMENT
6. *-----
7. *
8. * You can load different configurations depending on your
9. * current environment. Setting the environment also influences
10. * things like logging and error reporting.
11. *
12. * This can be set to anything, but default usage is:
13. *
14. *     development
15. *     testing
16. *     production
17. *
18. * NOTE: If you change these, also change the error_reporting() code below
19. *
20. */
21.     define('ENVIRONMENT', 'development');
22. /*
23. *-----
24. * ERROR REPORTING
25. *-----
26. *
27. * Different environments will require different levels of error reporting.
28. * By default development will show errors but testing and live will hide them.
29. */
30.
31. if (defined('ENVIRONMENT'))
32. {
33.     switch (ENVIRONMENT)
34.     {
35.         case 'development':
36.             error_reporting(E_ALL);
37.             break;
38.
39.         case 'testing':
40.         case 'production':
41.             error_reporting(0);
42.             break;
43.
44.         default:
45.             exit('The application environment is not set correctly.');
```

```
49. /*
50. *-----
51. * SYSTEM FOLDER NAME
52. *-----
53. *
54. * This variable must contain the name of your "system" folder.
55. * Include the path if the folder is not in the same directory
56. * as this file.
57. *
58. */
59.     $system_path = '../system';
60.
61. /*
62. *-----
63. * APPLICATION FOLDER NAME
64. *-----
65. *
66. * If you want this front controller to use a different "application"
67. * folder then the default one you can set its name here. The folder
68. * can also be renamed or relocated anywhere on your server. If
69. * you do, use a full server path. For more info please see the user guide:
70. * http://codeigniter.com/user\_guide/general/managing\_apps.html
71. *
72. * NO TRAILING SLASH!
73. *
74. */
75.     $application_folder = '../application';
76.
77. /*
78. * -----
79. * DEFAULT CONTROLLER
80. * -----
81. *
82. * Normally you will set your default controller in the routes.php file.
83. * You can, however, force a custom routing by hard-coding a
84. * specific controller class/function here. For most applications, you
85. * WILL NOT set your routing here, but it's an option for those
86. * special instances where you might want to override the standard
87. * routing in a specific front controller that shares a common CI installation.
88. *
89. * IMPORTANT: If you set the routing here, NO OTHER controller will be
90. * callable. In essence, this preference limits your application to ONE
91. * specific controller. Leave the function name blank if you need
92. * to call functions dynamically via the URI.
93. *
94. * Un-comment the $routing array below to use this feature
95. *
96. */
```

```
97. // The directory name, relative to the "controllers" folder. Leave blank
98. // if your controller is not in a sub-folder within the "controllers" folder
99. // $routing['directory'] = '';
100.
101. // The controller class file name. Example: Mycontroller
102. // $routing['controller'] = '';
103.
104. // The controller function you wish to be called.
105. // $routing['function'] = '';
106.
107.
108. /*
109. * -----
110. * CUSTOM CONFIG VALUES
111. * -----
112. *
113. * The $assign_to_config array below will be passed dynamically to the
114. * config class when initialized. This allows you to set custom config
115. * items or override any default config values found in the config.php file.
116. * This can be handy as it permits you to share one application between
117. * multiple front controller files, with each file containing different
118. * config values.
119. *
120. * Un-comment the $assign_to_config array below to use this feature
121. *
122. */
123. // $assign_to_config['name_of_config_item'] = 'value of config item';
124.
125.
126.
127. // -----
128. // END OF USER CONFIGURABLE SETTINGS. DO NOT EDIT BELOW THIS LINE
129. // -----
130.
131. /*
132. * -----
133. * Resolve the system path for increased reliability
134. * -----
135. */
136.
137. // Set the current directory correctly for CLI requests
138. if (defined('STDIN'))
139. {
140.     chdir(dirname(__FILE__));
141. }
142.
143. if (realpath($system_path) !== FALSE)
144. {
```

```
145.     $system_path = realpath($system_path).'/';
146. }
147.
148. // ensure there's a trailing slash
149. $system_path = rtrim($system_path, '/').'/';
150.
151. // Is the system path correct?
152. if ( ! is_dir($system_path) )
153. {
154.     exit("Your system folder path does not appear to be set correctly. Please open the following file and correct this: ".pathinfo(
155.     __FILE__, PATHINFO_BASENAME));
156. }
157. /*
158. * -----
159. * Now that we know the path, set the main path constants
160. * -----
161. */
162. // The name of THIS file
163. define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
164.
165. // The PHP file extension
166. // this global constant is deprecated.
167. define('EXT', '.php');
168.
169. // Path to the system folder
170. define('BASEPATH', str_replace("\\", "/", $system_path));
171.
172. // Path to the front controller (this file)
173. define('FCPATH', str_replace(SELF, '', __FILE__));
174.
175. // Name of the "system folder"
176. define('SYSDIR', trim(strchr(trim(BASEPATH, '/'), '/'), '/'));
177.
178.
179. // The path to the "application" folder
180. if (is_dir($application_folder))
181. {
182.     define('APPPATH', $application_folder.'/');
183. }
184. else
185. {
186.     if ( ! is_dir(BASEPATH.$application_folder.'/'))
187.     {
188.         exit("Your application folder path does not appear to be set correctly. Please open the following file and correct this: ".SELF);
189.     }
190.
191.     define('APPPATH', BASEPATH.$application_folder.'/');
```

```
192.     }
193.
194. /*
195.  * -----
196.  * LOAD THE BOOTSTRAP FILE
197.  * -----
198.  *
199.  * And away we go...
200.  *
201. */
202. require_once BASEPATH.'core/CodeIgniter.php' ;
203.
204. /* End of file index.php */
205. /* Location: ./index.php */
```

```
1.
2. /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
3. /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
4. /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
5. /*!40101 SET NAMES utf8 */;
6. /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
7. /*!40103 SET TIME_ZONE='+00:00' */;
8. /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
9. /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
10. /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
11. /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
12. DROP TABLE IF EXISTS `labs`;
13. /*!40101 SET @saved_cs_client = @@character_set_client */;
14. /*!40101 SET character_set_client = utf8 */;
15. CREATE TABLE `labs` (
16.   `id` int(11) NOT NULL,
17.   `name` varchar(128) NOT NULL,
18.   PRIMARY KEY (`id`)
19. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
20. /*!40101 SET character_set_client = @saved_cs_client */;
21.
22. LOCK TABLES `labs` WRITE;
23. /*!40000 ALTER TABLE `labs` DISABLE KEYS */;
24. INSERT INTO `labs` VALUES (1,'Lab 1'),(2,'Lab 2');
25. /*!40000 ALTER TABLE `labs` ENABLE KEYS */;
26. UNLOCK TABLES;
27. DROP TABLE IF EXISTS `lectures`;
28. /*!40101 SET @saved_cs_client = @@character_set_client */;
29. /*!40101 SET character_set_client = utf8 */;
30. CREATE TABLE `lectures` (
31.   `id` int(11) NOT NULL,
32.   `name` varchar(128) NOT NULL,
33.   PRIMARY KEY (`id`)
34. ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
35. /*!40101 SET character_set_client = @saved_cs_client */;
36.
37. LOCK TABLES `lectures` WRITE;
38. /*!40000 ALTER TABLE `lectures` DISABLE KEYS */;
39. INSERT INTO `lectures` VALUES (1,'Lecture 1'),(2,'Lecture 2');
40. /*!40000 ALTER TABLE `lectures` ENABLE KEYS */;
41. UNLOCK TABLES;
42. /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
43.
44. /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
45. /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
46. /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
47. /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
48. /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```



```
49. /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
50. /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

---

1. mvc/11/README
- 2.
3. David J. Malan
4. malan@harvard.edu
- 5.
6. Computer Science 164
7. Harvard College
- 8.
9. Improves upon mvc/10, introducing models.
- 10.
11. application/
  12. config/
    13. autoload.php - CodeIgniter's configuration
    14. config.php - CodeIgniter's configuration
    15. database.php - CodeIgniter's configuration
    16. ...
  17. controllers/
    18. labs.php - application controller
    19. lectures.php - application controller
    20. welcome.php - application controller
  21. errors/
    22. ...
  23. views/
    24. homepage/
      25. index.php - a homepage for the course
      26. lecture.php - a lecture
      27. lectures.php - a list of lectures
  28. templates/
    29. footer.php - pages' footer
    30. header.php - pages' header
31. index.php - front controller
32. system/
  33. ...