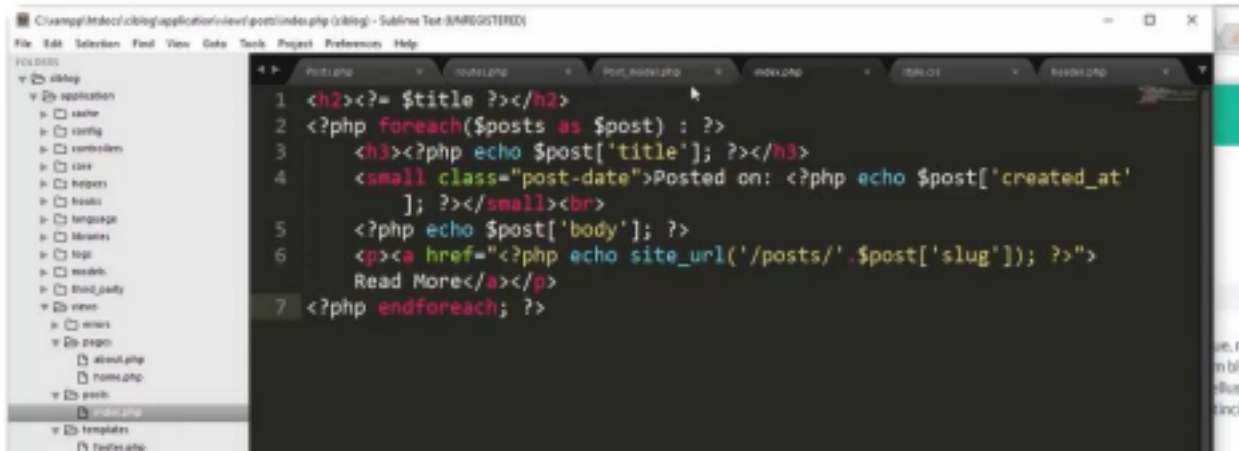
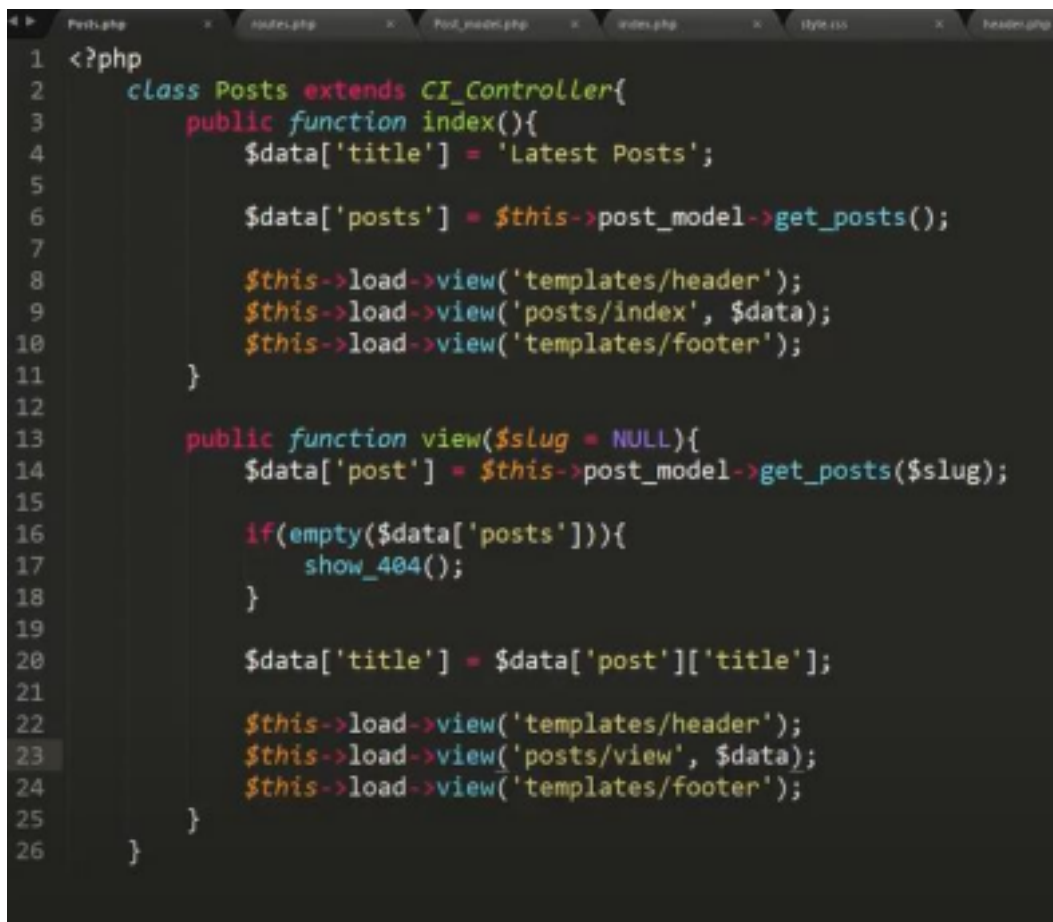


View Single Page



```
1 <h2><?=$title ?></h2>
2 <?php foreach($posts as $post) : ?>
3     <h3><?php echo $post['title']; ?></h3>
4     <small class="post-date">Posted on: <?php echo $post['created_at'
5         ]; ?></small><br>
6     <?php echo $post['body']; ?>
7     <p><a href="<?php echo site_url('/posts/'.$post['slug']); ?>">
    Read More</a></p>
8 <?php endforeach; ?>
```

Figure 1. Site_URL (view)



```
1 <?php
2 class Posts extends CI_Controller{
3     public function index(){
4         $data['title'] = 'Latest Posts';
5
6         $data['posts'] = $this->post_model->get_posts();
7
8         $this->load->view('templates/header');
9         $this->load->view('posts/index', $data);
10        $this->load->view('templates/footer');
11    }
12
13    public function view($slug = NULL){
14        $data['post'] = $this->post_model->get_posts($slug);
15
16        if(empty($data['posts'])){
17            show_404();
18        }
19
20        $data['title'] = $data['post']['title'];
21
22        $this->load->view('templates/header');
23        $this->load->view('posts/view', $data);
24        $this->load->view('templates/footer');
25    }
26 }
```

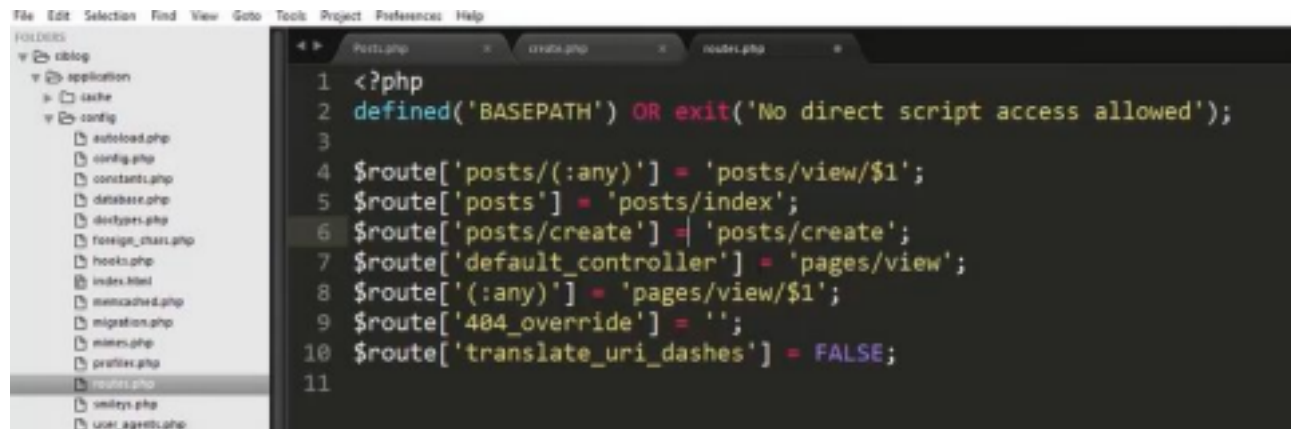
Figure 2. Controller- public function for view

Figure 3. View.php for single display

Figure 4. Change the route (Line 4 route to the view.php)

Create new record

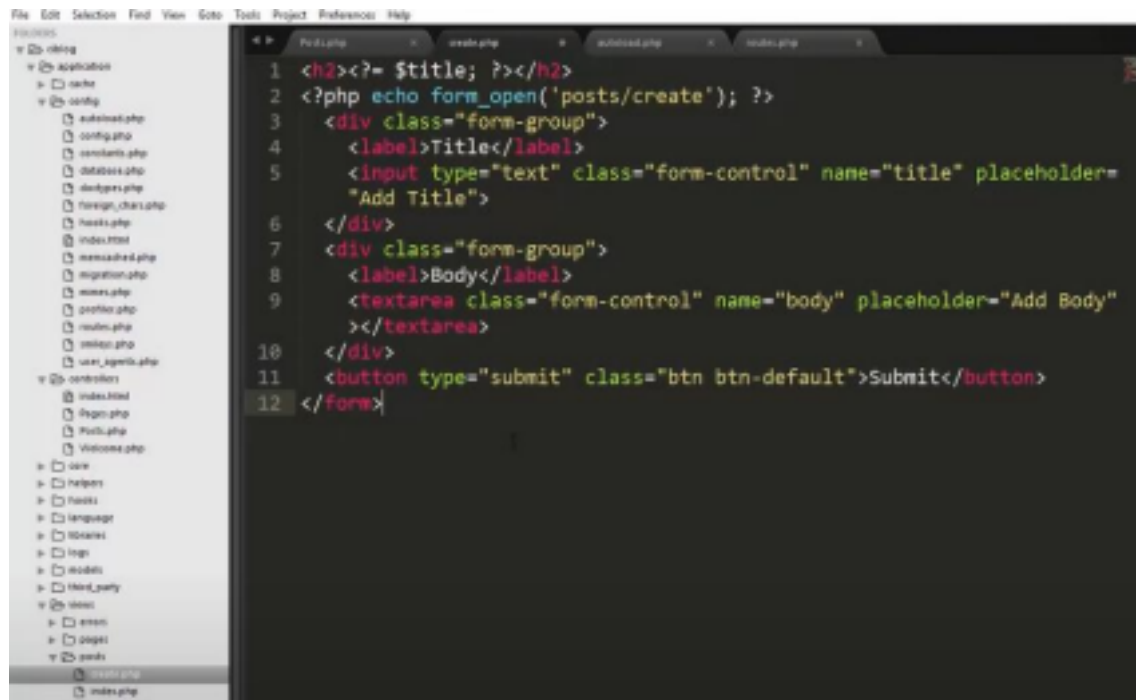
Figure 5. Create a public function for creating new record



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  application
  cache
  config
    autoload.php
    config.php
    constants.php
    database.php
    doctypes.php
    foreign_chars.php
    hooks.php
    index.html
    memcached.php
    migration.php
    minify.php
    profiler.php
    routes.php
    smarty.php
    user_agents.php
  Posts.php
  create.php
  routes.php

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 $route['posts/(.*)'] = 'posts/view/$1';
5 $route['posts'] = 'posts/index';
6 $route['posts/create'] = 'posts/create';
7 $route['default_controller'] = 'pages/view';
8 $route['(:any)'] = 'pages/view/$1';
9 $route['404_override'] = '';
10 $route['translate_uri_dashes'] = FALSE;
11
```

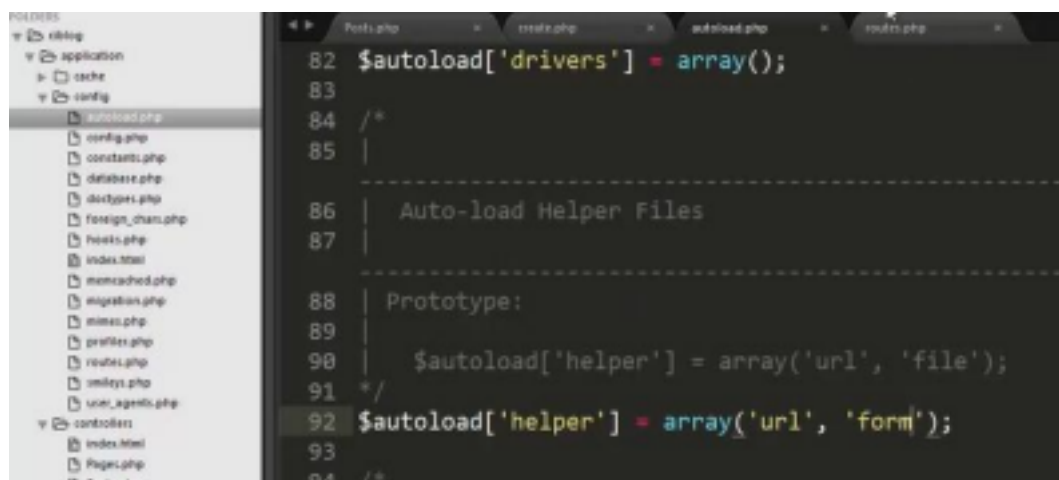
Figure 6. Line 6- create the new route



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  application
  cache
  config
    autoload.php
    config.php
    constants.php
    database.php
    doctypes.php
    foreign_chars.php
    hooks.php
    index.html
    memcached.php
    migration.php
    minify.php
    profiler.php
    routes.php
    smarty.php
    user_agents.php
  controllers
    index.html
    login.php
    Posts.php
    welcome.php
  core
  helpers
  hooks
  language
  libraries
  logs
  models
  third_party
  views
    errors
    pages
    posts
    posts.php
    index.php
    welcome.php

1 <h2><?php echo $title; ?></h2>
2 <?php echo form_open('posts/create'); ?>
3 <div class="form-group">
4   <label>Title</label>
5   <input type="text" class="form-control" name="title" placeholder="Add Title">
6 </div>
7 <div class="form-group">
8   <label>Body</label>
9   <textarea class="form-control" name="body" placeholder="Add Body">
10 </div>
11 <button type="submit" class="btn btn-default">Submit</button>
12 </form>
```

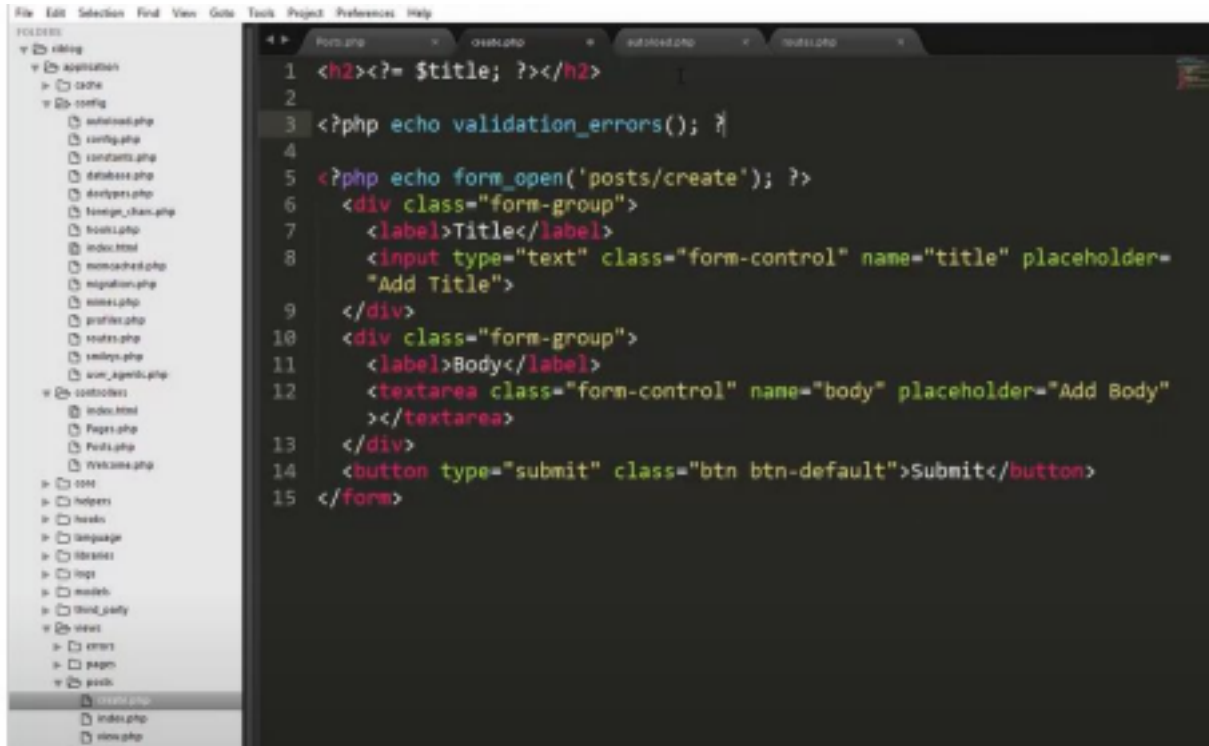
Figure 7. Create the form



```
FOLDERS
  application
  cache
  config
    autoload.php
    config.php
    constants.php
    database.php
    doctypes.php
    foreign_chars.php
    hooks.php
    index.html
    memcached.php
    migration.php
    minify.php
    profiler.php
    routes.php
    smarty.php
    user_agents.php
  controllers
    index.html
    login.php
    Posts.php
    welcome.php

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', 'form');
93
94 /*
```

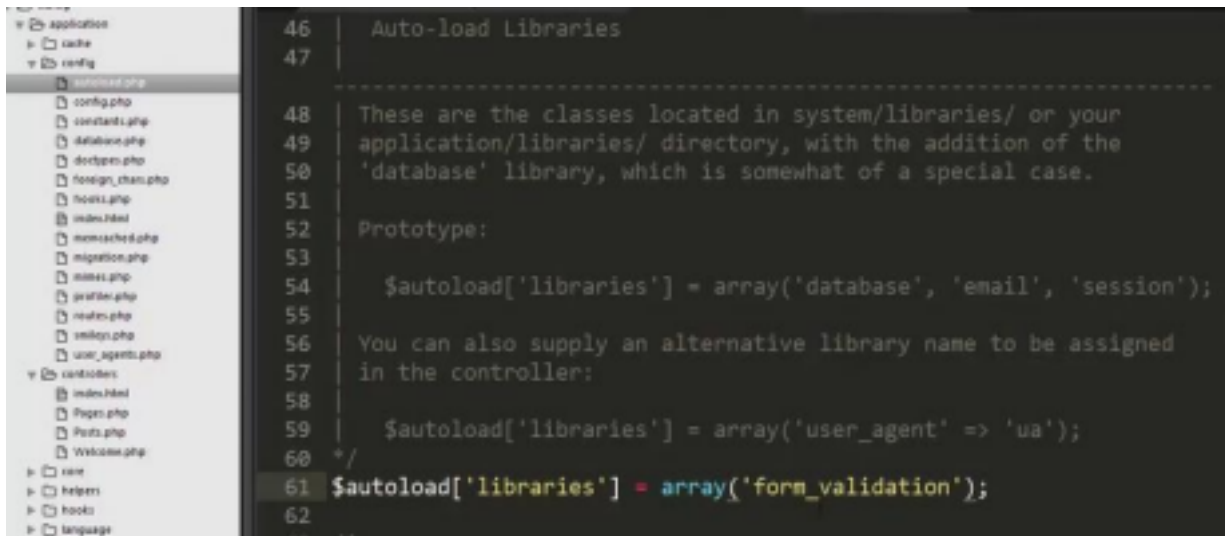
Figure 8. Add the form helper in the autoload config



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'application', 'cache', 'config', 'controllers', 'core', 'helpers', 'hooks', 'language', 'libraries', 'logs', 'models', 'third_party', 'views', and 'pages'. The code editor shows the 'autoload.php' file with the following code:

```
1 <h2><?=$title; ?></h2>
2
3 <?php echo validation_errors(); ?>
4
5 <?php echo form_open('posts/create'); ?>
6 <div class="form-group">
7   <label>Title</label>
8   <input type="text" class="form-control" name="title" placeholder=
  "Add Title">
9 </div>
10 <div class="form-group">
11   <label>Body</label>
12   <textarea class="form-control" name="body" placeholder="Add Body"
  ></textarea>
13 </div>
14 <button type="submit" class="btn btn-default">Submit</button>
15 </form>
```

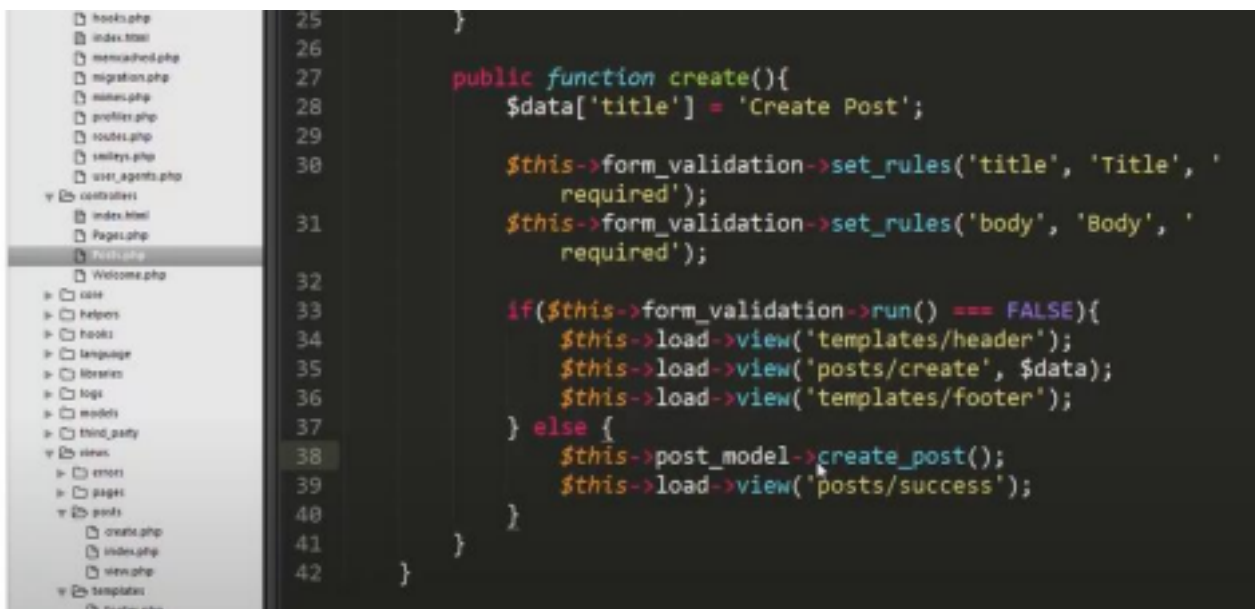
Figure 9. Add the form validation (line 3)



The screenshot shows the same IDE with the 'autoload.php' file. The code is as follows:

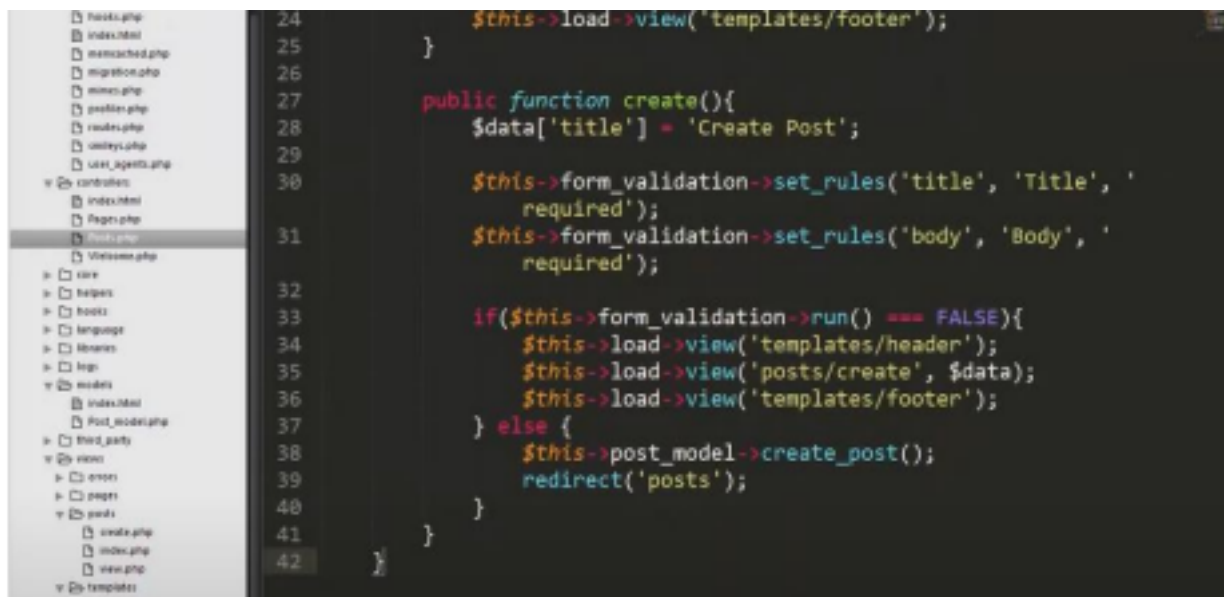
```
46 Auto-load Libraries
47
48 -----
49 These are the classes located in system/libraries/ or your
50 application/libraries/ directory, with the addition of the
51 'database' library, which is somewhat of a special case.
52 Prototype:
53
54 $autoload['libraries'] = array('database', 'email', 'session');
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('form_validation');
62
```

Figure 10. Load the form validation in the \$autoload['libraries']



```
25 }
26
27 public function create(){
28     $data['title'] = 'Create Post';
29
30     $this->form_validation->set_rules('title', 'Title', '
31         required');
32     $this->form_validation->set_rules('body', 'Body', '
33         required');
34
35     if($this->form_validation->run() == FALSE){
36         $this->load->view('templates/header');
37         $this->load->view('posts/create', $data);
38         $this->load->view('templates/footer');
39     } else {
40         $this->post_model->create_post();
41         $this->load->view('posts/success');
42     }
43 }
```

Figure 11. Add the form validation rule



```
24 $this->load->view('templates/footer');
25 }
26
27 public function create(){
28     $data['title'] = 'Create Post';
29
30     $this->form_validation->set_rules('title', 'Title', '
31         required');
32     $this->form_validation->set_rules('body', 'Body', '
33         required');
34
35     if($this->form_validation->run() == FALSE){
36         $this->load->view('templates/header');
37         $this->load->view('posts/create', $data);
38         $this->load->view('templates/footer');
39     } else {
40         $this->post_model->create_post();
41         redirect('posts');
42     }
43 }
```

Figure 12. Create the model for the creating the post

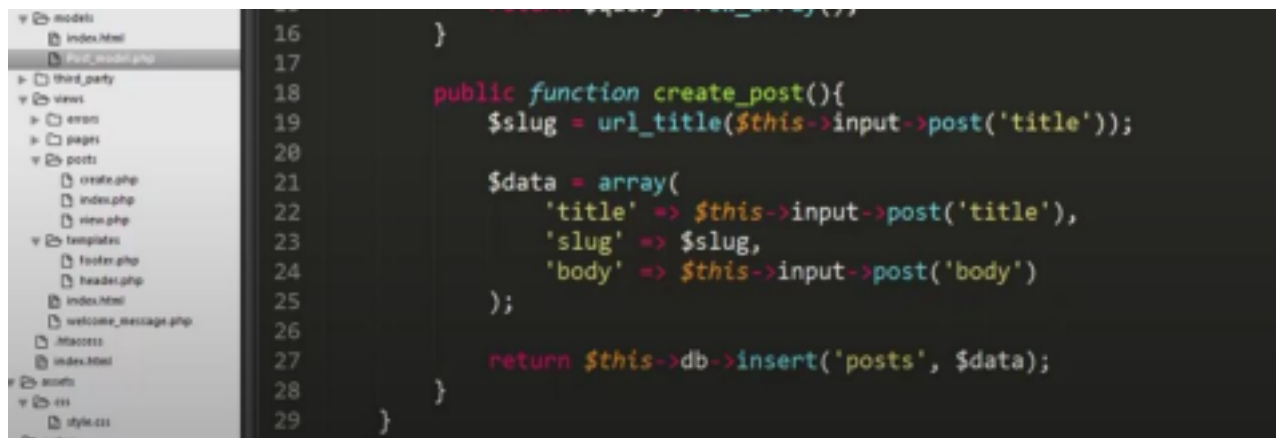


Figure 13. Post model create