

CodeIgniter 3 Guidance

August 30, 2024

You said:

Act as a codeigniter version 3 developer and guide me ok

ChatGPT said:



Sure! I can help you with CodeIgniter 3 development. Whether you need help with setting up a project, working with controllers, models, views, routing, or any other aspect of CodeIgniter, feel free to ask.

To get started, could you let me know what specific issue or task you're working on? If you have any code examples or specific questions, please share them so I can provide more precise guidance.

You said:

now create a curd app which contains register,login, and all the curd operations along with image upload and view option.

ChatGPT said:



To create a full CRUD (Create, Read, Update, Delete) application in CodeIgniter 3 with registration, login, and image upload functionalities, we'll go step by step. Below is an outline of what we'll build, including database setup, controllers, models, views, and routes.

Step-by-Step Plan

1. Database Setup:
 - Create a MySQL database and set up a table for users and posts.
2. Configuration:
 - Configure the database connection in `application/config/database.php`.

- Configure base URL in `application/config/config.php`.
- 3. **Routes:**
 - Define routes for registration, login, and CRUD operations.
- 4. **Controllers:**
 - **Auth:** Handles registration and login.
 - **Post:** Handles CRUD operations and image upload.
- 5. **Models:**
 - **User_model:** Handles user-related database interactions.
 - **Post_model:** Handles post-related database interactions.
- 6. **Views:**
 - Create views for registration, login, post listing, post creation, editing, and viewing.
- 7. **Image Upload Handling:**
 - Add functionality to upload images and display them.

1. Database Setup

Create a database and the following tables.

sql

Copy code

```
CREATE TABLE users (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) NOT NULL,
    password VARCHAR(255) NOT NULL
);

CREATE TABLE posts (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    user_id INT(11),
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    image VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

2. Configuration

- Database Configuration in `application/config/database.php`:

php

Copy code

```
$db['default'] = array(
    'dsn'      => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'your_database_name',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on'  => FALSE,
    'cachedir'  => '',
    'char_set'  => 'utf8',
    'dbcollat'  => 'utf8_general_ci',
    'swap_pre'  => '',
    'encrypt'   => FALSE,
    'compress'  => FALSE,
    'stricton'  => FALSE,
    'failover'  => array(),
    'save_queries' => TRUE
);
```

- Base URL Configuration in application/config/config.php:

php

Copy code

```
$config['base_url'] = 'http://localhost/your_project_name/';
```

3. Routes

Add the following routes in application/config/routes.php:

php

Copy code

```
$route['default_controller'] = 'auth/login';
$route['register'] = 'auth/register';
$route['login'] = 'auth/login';
```

```
$route['logout'] = 'auth/logout';
```

```
$route['posts'] = 'post/index';
```

```
$route['post/create'] = 'post/create';
```

```
$route['post/edit/(:num)'] = 'post/edit/$1';
```

```
$route['post/delete/(:num)'] = 'post/delete/$1';
```

```
$route['post/view/(:num)'] = 'post/view/$1';
```

4. Controllers

Auth Controller (application/controllers/Auth.php):

php

Copy code

```
<?php
```

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

```
class Auth extends CI_Controller {
```

```
    public function __construct() {
```

```
        parent::__construct();
```

```
        $this->load->model('User_model');
```

```
        $this->load->library('session');
```

```
        $this->load->helper(array('form', 'url'));
```

```
    }
```

```
    public function register() {
```

```
        if ($this->input->post()) {
```

```
            $data = array(
```

```
                'username' => $this->input->post('username'),
```

```
                'password' =>
```

```
password_hash($this->input->post('password'),
```

```
PASSWORD_BCRYPT),
```

```
            );
```

```
            $this->User_model->insert_user($data);
```

```
            redirect('login');
```

```
        } else {
```

```

        $this->load->view('register');
    }
}

public function login() {
    if ($this->input->post()) {
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $user = $this->User_model->get_user($username);

        if ($user && password_verify($password,
$user->password)) {
            $this->session->set_userdata('user_id',
$user->id);
            redirect('posts');
        } else {
            $this->session->set_flashdata('error',
'Invalid credentials');
            redirect('login');
        }
    } else {
        $this->load->view('login');
    }
}

public function logout() {
    $this->session->unset_userdata('user_id');
    redirect('login');
}
}

```

Post Controller (application/controllers/Post.php):

php

Copy code

<?php

```

defined('BASEPATH') OR exit('No direct script access
allowed');

```

```

class Post extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('Post_model');
        $this->load->library('session');
        $this->load->helper(array('form', 'url'));

        if (!$this->session->userdata('user_id')) {
            redirect('login');
        }
    }

    public function index() {
        $data['posts'] = $this->Post_model->get_posts();
        $this->load->view('posts/index', $data);
    }

    public function create() {
        if ($this->input->post()) {
            $config['upload_path'] = './uploads/';
            $config['allowed_types'] = 'gif|jpg|png';
            $this->load->library('upload', $config);

            $image = null;
            if ($this->upload->do_upload('image')) {
                $image = $this->upload->data('file_name');
            }

            $data = array(
                'user_id' =>
$this->session->userdata('user_id'),
                'title' => $this->input->post('title'),
                'content' => $this->input->post('content'),
                'image' => $image,
            );
            $this->Post_model->insert_post($data);
        }
    }
}

```

```

        redirect('posts');
    } else {
        $this->load->view('posts/create');
    }
}

public function edit($id) {
    $data['post'] = $this->Post_model->get_post($id);

    if ($this->input->post()) {
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);

        $image = $data['post']->image;
        if ($this->upload->do_upload('image')) {
            $image = $this->upload->data('file_name');
        }

        $update_data = array(
            'title' => $this->input->post('title'),
            'content' => $this->input->post('content'),
            'image' => $image,
        );
        $this->Post_model->update_post($id, $update_data);
        redirect('posts');
    } else {
        $this->load->view('posts/edit', $data);
    }
}

public function delete($id) {
    $this->Post_model->delete_post($id);
    redirect('posts');
}

public function view($id) {
    $data['post'] = $this->Post_model->get_post($id);

```

```
        $this->load->view('posts/view', $data);
    }
}
```

5. Models

User Model (application/models/User_model.php):

php

Copy code

<?php

```
class User_model extends CI_Model {

    public function insert_user($data) {
        $this->db->insert('users', $data);
    }

    public function get_user($username) {
        return $this->db->get_where('users', array('username'
=> $username))->row();
    }
}
```

Post Model (application/models/Post_model.php):

php

Copy code

<?php

```
class Post_model extends CI_Model {

    public function get_posts() {
        return $this->db->get('posts')->result();
    }

    public function insert_post($data) {
        $this->db->insert('posts', $data);
    }
}
```



```

    public function get_post($id) {
        return $this->db->get_where('posts', array('id' =>
$Id))->row();
    }

    public function update_post($id, $data) {
        $this->db->where('id', $id);
        $this->db->update('posts', $data);
    }

    public function delete_post($id) {
        $this->db->where('id', $id);
        $this->db->delete('posts');
    }
}

```

6. Views

- Register View (application/views/register.php):

html

Copy code

```

<!DOCTYPE html>
<html>
<head>
    <title>Register</title>
</head>
<body>
    <h1>Register</h1>
    <form method="POST" action="">
        <input type="text" name="username"
placeholder="Username" required><br>
        <input type="password" name="password"
placeholder="Password" required><br>
        <button type="submit">Register</button>
    </form>
</body>
</html>

```

- Login View (application/views/login.php):

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h1>Login</h1>
    <form method="POST" action="">
        <input type="text" name="username"
placeholder="Username" required><br>
        <input type="password" name="password"
placeholder="Password" required><br>
        <button type="submit">Login</button>
    </form>
    <?php if($this->session->flashdata('error')): ?>
        <p><?php echo $this->session->flashdata('error');
?></p>
    <?php endif; ?>
</body>
</html>
```

- Post Index View (application/views/posts/index.php):

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>Posts</title>
</head>
<body>
    <h1>Posts</h1>
```

```

    <a href="<?php echo base_url('post/create'); ?>">Create
Post</a>
    <table border="1">
        <tr>
            <th>Title</th>
            <th>Actions</th>
        </tr>
        <?php foreach($posts as $post): ?>
        <tr>
            <td><?php echo $post->title; ?></td>
            <td>
                <a href="<?php echo
base_url('post/view/' . $post->id); ?>">View</a>
                <a href="<?php echo
base_url('post/edit/' . $post->id); ?>">Edit</a>
                <a href="<?php echo
base_url('post/delete/' . $post->id); ?>">Delete</a>
            </td>
        </tr>
        <?php endforeach; ?>
    </table>
</body>
</html>

```

- Post Create View (application/views/posts/create.php):

```

html
Copy code
<!DOCTYPE html>
<html>
<head>
    <title>Create Post</title>
</head>
<body>
    <h1>Create Post</h1>
    <form method="POST" action=""
enctype="multipart/form-data">

```

```

        <input type="text" name="title" placeholder="Title"
required><br>
        <textarea name="content" placeholder="Content"
required></textarea><br>
        <input type="file" name="image"><br>
        <button type="submit">Create</button>
    </form>
</body>
</html>

```

- Post Edit View (application/views/posts/edit.php):

html

Copy code

```

<!DOCTYPE html>
<html>
<head>
    <title>Edit Post</title>
</head>
<body>
    <h1>Edit Post</h1>
    <form method="POST" action=""
enctype="multipart/form-data">
        <input type="text" name="title" value="<?php echo
$post->title; ?>" required><br>
        <textarea name="content" required><?php echo
$post->content; ?></textarea><br>
        <input type="file" name="image"><br>
        <button type="submit">Update</button>
    </form>
</body>
</html>

```

- Post View (application/views/posts/view.php):

html

Copy code

```

<!DOCTYPE html>

```

```

<html>
<head>
    <title>View Post</title>
</head>
<body>
    <h1><?php echo $post->title; ?></h1>
    <p><?php echo $post->content; ?></p>
    <?php if($post->image): ?>
        
    <?php endif; ?>
</body>
</html>

```

7. Image Upload Handling

Ensure that the uploads directory exists in the root directory and is writable (chmod 777 uploads).

Final Notes

This is a basic CRUD application with registration, login, and image upload functionalities in CodeIgniter 3. You can further enhance it by adding validations, error handling, and security measures.

- a. Would you like to add form validation and error handling next?
- b. Do you need help setting up the environment or running the application?

You said:

Do you need help setting up the environment or running the application? yes i need to

ChatGPT said:



Great! I'll guide you through the process of setting up the environment and running the CodeIgniter 3 application on your local machine. Below are the steps to follow:

1. Install Required Software

Before we start, make sure you have the following installed:

- **XAMPP (or any other PHP development environment):** This includes Apache, MySQL, and PHP, which are necessary to run the CodeIgniter application.
 - You can download XAMPP from [here](#).
- **Composer (optional, but recommended):** Although CodeIgniter 3 doesn't require Composer, it can be helpful for managing PHP dependencies.
 - You can download Composer from [here](#).

2. Set Up XAMPP

1. **Start Apache and MySQL:**
 - Open the XAMPP Control Panel.
 - Start the Apache and MySQL services.
2. **Access phpMyAdmin:**
 - Go to your browser and open `http://localhost/phpmyadmin`.
 - Create a new database for your application. For example, you can name it `codeigniter_app`.

3. Download and Set Up CodeIgniter 3

1. **Download CodeIgniter 3:**
 - Download the latest version of CodeIgniter 3 from [here](#).

Extract the ZIP file to the `htdocs` directory of your XAMPP installation. For example:

`makefile`

`Copy code`

`C:\xampp\htdocs\codeigniter_app`

○

2. **Configure Base URL:**
 - Open `application/config/config.php`.

Set the `$config['base_url']` to your local development URL:

`php`

`Copy code`

`$config['base_url'] = 'http://localhost/codeigniter_app/';`

○

3. **Configure the Database:**

- Open `application/config/database.php`.

Set the database credentials:

php

Copy code

```
$db['default'] = array(
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'codeigniter_app',
    'dbdriver' => 'mysqli',
);
```

○

4. Import the Database

1. Create the Database:

- In phpMyAdmin, select your newly created database (codeigniter_app).

2. Import Tables:

- Use the SQL queries provided earlier to create the users and posts tables. You can do this by going to the SQL tab in phpMyAdmin and pasting the queries:

sql

Copy code

```
CREATE TABLE users (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(100) NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE posts (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    user_id INT(11),
    title VARCHAR(255) NOT NULL,
    content TEXT NOT NULL,
    image VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
```

);

3.

5. Run the Application

1. Access the Application:

- Open your browser and go to `http://localhost/codeigniter_app/`.
- You should see the login page.

2. Register a New User:

- Go to the registration page at `http://localhost/codeigniter_app/register`.
- After registering, log in with the new credentials.

3. Test CRUD Operations:

- After logging in, you can create, view, edit, and delete posts.

6. Troubleshooting Common Issues

- **404 Page Not Found:**
 - Ensure your routes are correctly defined in `application/config/routes.php`.
 - Make sure the URL matches your controller names and methods.
- **Database Connection Issues:**
 - Double-check the database credentials in `application/config/database.php`.
 - Ensure that MySQL is running in XAMPP.
- **Permission Errors:**
 - Ensure that the uploads directory has the correct permissions (`chmod 777 uploads` on Linux).

Next Steps

Once you have the application up and running, you can start customizing it or adding new features. If you encounter any issues or need further help, feel free to ask!

a. Do you need help with adding specific features or fixing any errors during setup?

b. Would you like to learn how to deploy this application to a live server?

You said:

Would you like to add form validation and error handling next? yes please

ChatGPT said:



Adding form validation and error handling is essential for ensuring that your application is robust and secure. I'll guide you through how to implement form validation and handle errors in your registration, login, and post creation forms in CodeIgniter 3.

1. Form Validation in CodeIgniter

CodeIgniter 3 provides a built-in form validation library that makes it easy to validate form data. You'll need to load the form validation library in your controllers and define validation rules for each form.

2. Adding Form Validation and Error Handling

Auth Controller (Registration & Login)

Updated Auth.php Controller:

php

Copy code

```
<?php
```

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

```
class Auth extends CI_Controller {  
  
    public function __construct() {  
        parent::__construct();  
        $this->load->model('User_model');  
        $this->load->library(array('form_validation',  
'session'));  
        $this->load->helper(array('form', 'url'));  
    }  
  
    public function register() {  
        // Define validation rules
```

```

        $this->form_validation->set_rules('username',
'Username', 'required|is_unique[users.username]');
        $this->form_validation->set_rules('password',
'Password', 'required|min_length[6]');

        if ($this->form_validation->run() == FALSE) {
            // Validation failed, load the registration form
with errors
            $this->load->view('register');
        } else {
            // Validation passed, proceed with user
registration
            $data = array(
                'username' => $this->input->post('username'),
                'password' =>
password_hash($this->input->post('password'),
PASSWORD_BCRYPT),
            );
            $this->User_model->insert_user($data);
            $this->session->set_flashdata('success',
'Registration successful! Please log in. ');
            redirect('login');
        }
    }

    public function login() {
        // Define validation rules
        $this->form_validation->set_rules('username',
'Username', 'required');
        $this->form_validation->set_rules('password',
'Password', 'required');

        if ($this->form_validation->run() == FALSE) {
            // Validation failed, load the login form with
errors
            $this->load->view('login');
        } else {
            // Validation passed, proceed with login

```

```

        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $user = $this->User_model->get_user($username);

        if ($user && password_verify($password,
$user->password)) {
            $this->session->set_userdata('user_id',
$user->id);
            redirect('posts');
        } else {
            $this->session->set_flashdata('error',
'Invalid credentials');
            redirect('login');
        }
    }

    public function logout() {
        $this->session->unset_userdata('user_id');
        redirect('login');
    }
}

```

Post Controller (Create & Edit)

Updated Post.php Controller:

php

Copy code

```
<?php
```

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

```

class Post extends CI_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('Post_model');
    }
}

```

```

        $this->load->library(array('form_validation',
'session')));
        $this->load->helper(array('form', 'url'));

        if (!$this->session->userdata('user_id')) {
            redirect('login');
        }
    }

    public function create() {
        // Define validation rules
        $this->form_validation->set_rules('title', 'Title',
'required');
        $this->form_validation->set_rules('content',
'Content', 'required');

        if ($this->form_validation->run() == FALSE) {
            // Validation failed, load the creation form with
errors
            $this->load->view('posts/create');
        } else {
            // Image upload configuration
            $config['upload_path'] = './uploads/';
            $config['allowed_types'] = 'gif|jpg|png';
            $this->load->library('upload', $config);

            $image = null;
            if ($this->upload->do_upload('image')) {
                $image = $this->upload->data('file_name');
            }

            // Insert post into database
            $data = array(
                'user_id' =>
$this->session->userdata('user_id'),
                'title' => $this->input->post('title'),
                'content' => $this->input->post('content'),
                'image' => $image,

```

```

        );
        $this->Post_model->insert_post($data);
        redirect('posts');
    }
}

public function edit($id) {
    $data['post'] = $this->Post_model->get_post($id);

    // Define validation rules
    $this->form_validation->set_rules('title', 'Title',
'required');
    $this->form_validation->set_rules('content',
'Content', 'required');

    if ($this->form_validation->run() == FALSE) {
        // Validation failed, load the edit form with
errors
        $this->load->view('posts/edit', $data);
    } else {
        // Image upload configuration
        $config['upload_path'] = './uploads/';
        $config['allowed_types'] = 'gif|jpg|png';
        $this->load->library('upload', $config);

        $image = $data['post']->image;
        if ($this->upload->do_upload('image')) {
            $image = $this->upload->data('file_name');
        }

        // Update post in the database
        $update_data = array(
            'title' => $this->input->post('title'),
            'content' => $this->input->post('content'),
            'image' => $image,
        );
        $this->Post_model->update_post($id, $update_data);
        redirect('posts');
    }
}

```

```

    }
}

public function delete($id) {
    $this->Post_model->delete_post($id);
    redirect('posts');
}

public function view($id) {
    $data['post'] = $this->Post_model->get_post($id);
    $this->load->view('posts/view', $data);
}
}

```

3. Updating Views to Display Validation Errors

Register View (register.php):

html

Copy code

```

<!DOCTYPE html>
<html>
<head>
    <title>Register</title>
</head>
<body>
    <h1>Register</h1>
    <?php echo validation_errors(); ?>
    <form method="POST" action="">
        <input type="text" name="username"
placeholder="Username" required value="<?php echo
set_value('username'); ?>"><br>
        <input type="password" name="password"
placeholder="Password" required><br>
        <button type="submit">Register</button>
    </form>
</body>
</html>

```

Login View (login.php):

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h1>Login</h1>
    <?php echo validation_errors(); ?>
    <?php if($this->session->flashdata('error')): ?>
        <p><?php echo $this->session->flashdata('error');
?></p>
    <?php endif; ?>
    <form method="POST" action="">
        <input type="text" name="username"
placeholder="Username" required value="<?php echo
set_value('username'); ?>"><br>
        <input type="password" name="password"
placeholder="Password" required><br>
        <button type="submit">Login</button>
    </form>
</body>
</html>
```

Create Post View (create.php):

html

Copy code

```
<!DOCTYPE html>
<html>
<head>
    <title>Create Post</title>
</head>
<body>
    <h1>Create Post</h1>
    <?php echo validation_errors(); ?>
```

```

        <form method="POST" action=""
enctype="multipart/form-data">
            <input type="text" name="title" placeholder="Title"
required value="<?php echo set_value('title'); ?>"><br>
            <textarea name="content" placeholder="Content"
required><?php echo set_value('content'); ?></textarea><br>
            <input type="file" name="image"><br>
            <button type="submit">Create</button>
        </form>
</body>
</html>

```

Edit Post View (edit.php):

```

<!DOCTYPE html>
<html>
<head>
    <title>Edit Post</title>
</head>
<body>
    <h1>Edit Post</h1>
    <?php echo validation_errors(); ?>
    <form method="POST" action=""
enctype="multipart/form-data">
        <input type="text" name="title" value="<?php echo
set_value('title', $post->title); ?>" required><br>
        <textarea name="content" required><?php echo
set_value('content', $post->content); ?></textarea><br>

        <!-- Display current image if available -->
        <?php if ($post->image): ?>
            <br>
        <?php endif; ?>

        <input type="file" name="image"><br>

```



```
        <button type="submit">Update</button>
    </form>
</body>
</html>
```

how to remove index.php in CI?

ChatGPT said:



To remove `index.php` from the URLs in a CodeIgniter 3 application, you'll need to configure the `.htaccess` file and make a few changes to the configuration files. Follow these steps:

1. Enable mod_rewrite in Apache

If you're using XAMPP, WAMP, or a similar local server environment, ensure that the `mod_rewrite` module is enabled in Apache.

- Open your `httpd.conf` file located in the Apache configuration directory. For example, in XAMPP, it's typically located at `C:\xampp\apache\conf\httpd.conf`.

Find and uncomment the following line by removing the `#` at the beginning:

apache

Copy code

```
LoadModule rewrite_module modules/mod_rewrite.so
```

-

Also, search for the following block and ensure that `AllowOverride` is set to `All`:

apache

Copy code

```
<Directory "C:/xampp/htdocs">
    Options Indexes FollowSymLinks Includes ExecCGI
```

```
    AllowOverride All
    Require all granted
</Directory>
```

-
- Save the `httpd.conf` file and restart Apache.

2. Update the .htaccess File

Create or update the `.htaccess` file in the root directory of your CodeIgniter application (where the `index.php` file is located). Add the following code:

apache

Copy code

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /your_project_name/

    # Redirect Trailing Slashes...
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)/$ /$1 [L,R=301]

    # Rewrite to remove index.php
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule ^(.*)$ index.php/$1 [L]
</IfModule>
```

- Replace `/your_project_name/` with the base directory name of your project. If your project is directly in the `htdocs` folder (i.e., accessible at `http://localhost/`), you can leave `RewriteBase` as `/`:

apache

Copy code

```
RewriteBase /
```

3. Update the config.php File

Edit `application/config/config.php` and update the `index_page` configuration to remove `index.php`:

php

Copy code

```
$config['index_page'] = '';
```

4. Set the `uri_protocol`

In the same `config.php` file, ensure that the `uri_protocol` is correctly set. Try setting it to `REQUEST_URI`:

php

Copy code

```
$config['uri_protocol'] = 'REQUEST_URI';
```

5. Test the Application

Now, visit your application in the browser without `index.php` in the URL. For example, instead of:

arduino

Copy code

```
http://localhost/your_project_name/index.php/posts
```

It should be accessible at:

arduino

Copy code

```
http://localhost/your_project_name/posts
```

Troubleshooting:

- **404 Errors:** If you encounter 404 errors, double-check that `mod_rewrite` is enabled and the `.htaccess` file is in the correct location.
- **Ensure `.htaccess` is supported:** Some servers disable `.htaccess` overrides by default. Ensure your Apache configuration allows `.htaccess` files to be used.