

2 Aim: - First program by edit the view as First-Program

Learning outcome: - Able to understand Laravel Framework.

Requirement of Hardware & Software: -

1 Working PC.

2 Windows Operating System.

3 Xampp server and also PHP.

4 Composer, Laravel 8 and any browser.

Procedure: -

Open your Command prompt and Please enter a composer create project Laravel/Laravel First-Program

Then after write code.

Step1:-

Open CMD and Write code

Code: -

composer create-project laravel/Laravel First-Program

Screenshot of code & Output:

```
C:\Windows\System32\cmd.exe x + v
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\htdocs>composer create-project laravel/laravel First-Program
Creating a "laravel/laravel" project at "./First-Program"
Info from https://repo.packagist.org: #StandWithUkraine
Cannot use laravel/laravel's latest version v10.2.1 as it requires php ^8.1 which is not satisfied by your platform.
Installing laravel/laravel (v9.5.2)
- Installing laravel/laravel (v9.5.2): Extracting archive
Created project in C:\xampp\htdocs\First-Program
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.11.0)
- Locking dflydev/dot-access-data (v3.0.2)
- Locking doctrine/annotations (v1.0.0)
- Locking doctrine/inflector (2.0.6)
- Locking doctrine/instantiator (1.5.0)
- Locking doctrine/lexer (2.1.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (3.2.5)
- Locking fakerphp/faker (v1.22.0)
- Locking filp/whoops (2.15.2)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.1)
- Locking guzzlehttp/guzzle (7.6.1)
- Locking guzzlehttp/promises (1.5.2)
- Locking guzzlehttp/psr7 (2.5.0)
- Locking guzzlehttp/uri-template (v1.0.1)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v9.52.7)
```

Step2:

Start sever:

Code:

php artisan serve

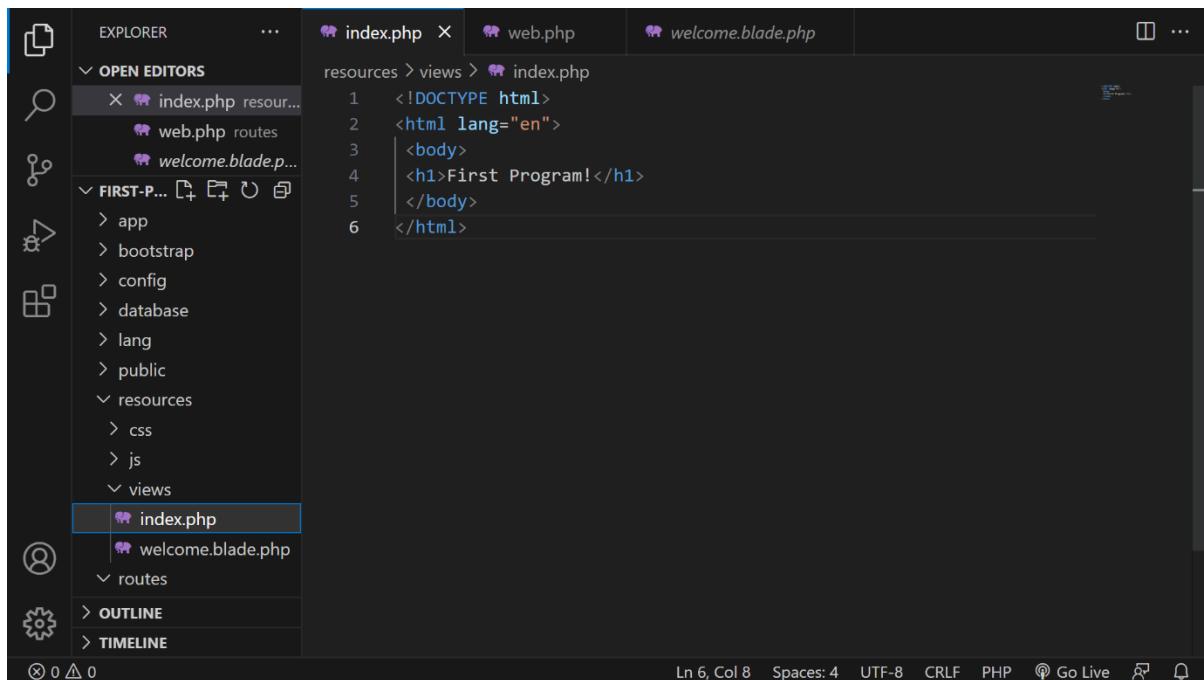
```
C:\Windows\System32\cmd.exe x + v
| make:controller
| make:event
| make:exception
| make:factory
| make:job
| make:listener
| make:mail
| make:middleware
| make:migration
| make:model
| make:notification
| make:observer
| make:policy
| make:provider
| make:request
| make:resource
| make:rule
| make:scope
| make:seeder
| make:test

C:\xampp\htdocs\First-Program>php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

Create a view named index.php in the resources/views directory and save the following code.

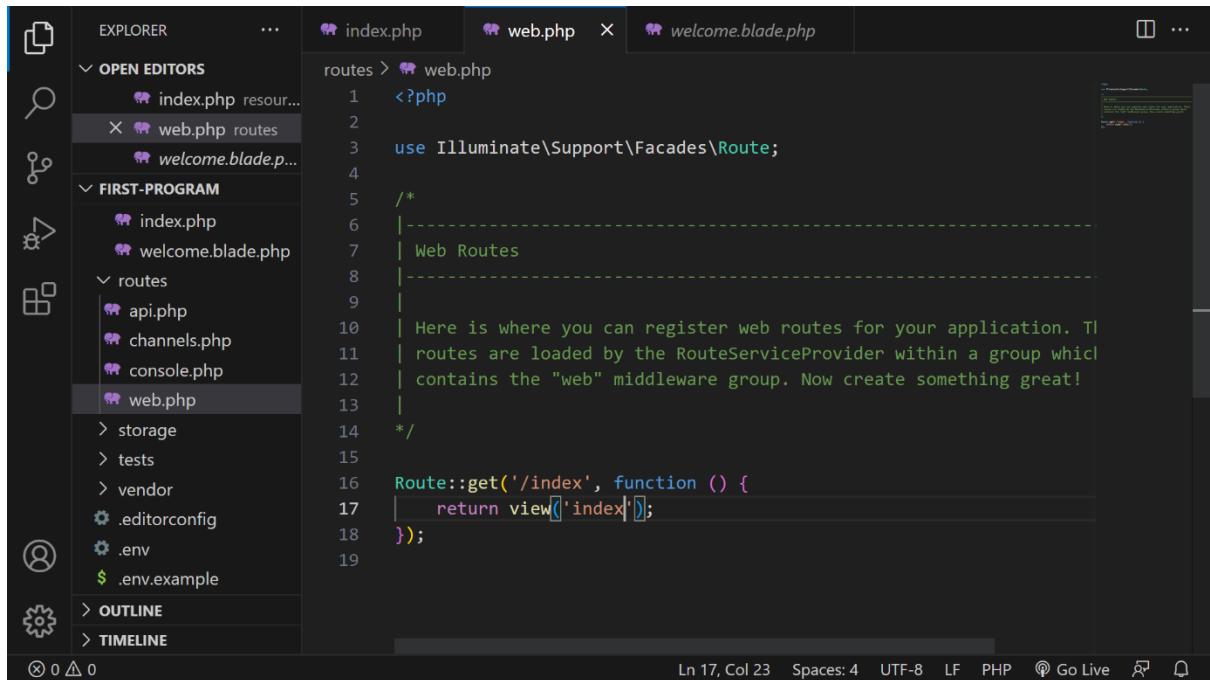
And write code:

```
<!DOCTYPE html>
<html lang="en">
<body>
<h1>First-Program!</h1>
</body>
</html>
```



Add the following code segment in the routes/web.php file to register a new route

```
Route::get('/index', function () {
    return view('index');
});
```



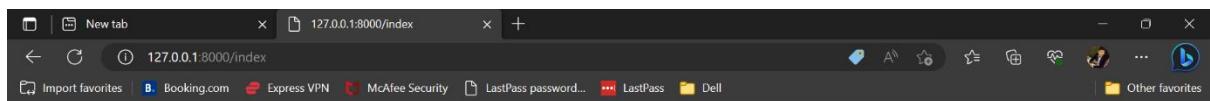
```
index.php routes > web.php
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----|
7 | Web Routes
8 |-----|
9 |
10 | Here is where you can register web routes for your application. The
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/index', function () {
17 |     return view('index');
18 });
19
```

Output in browser:

Go to browser and search

Code:

127.0.0.1:8000/index



First Program!

3 Aim: - Create a web application of minimum 4 blades demonstrating the template in Laravel.

Learning outcome: - Able to understand Laravel Framework.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Open your Command prompt and Please enter a composer create project Laravel/Laravel

Then after write code.

Step1:-

Open CMD and Write code

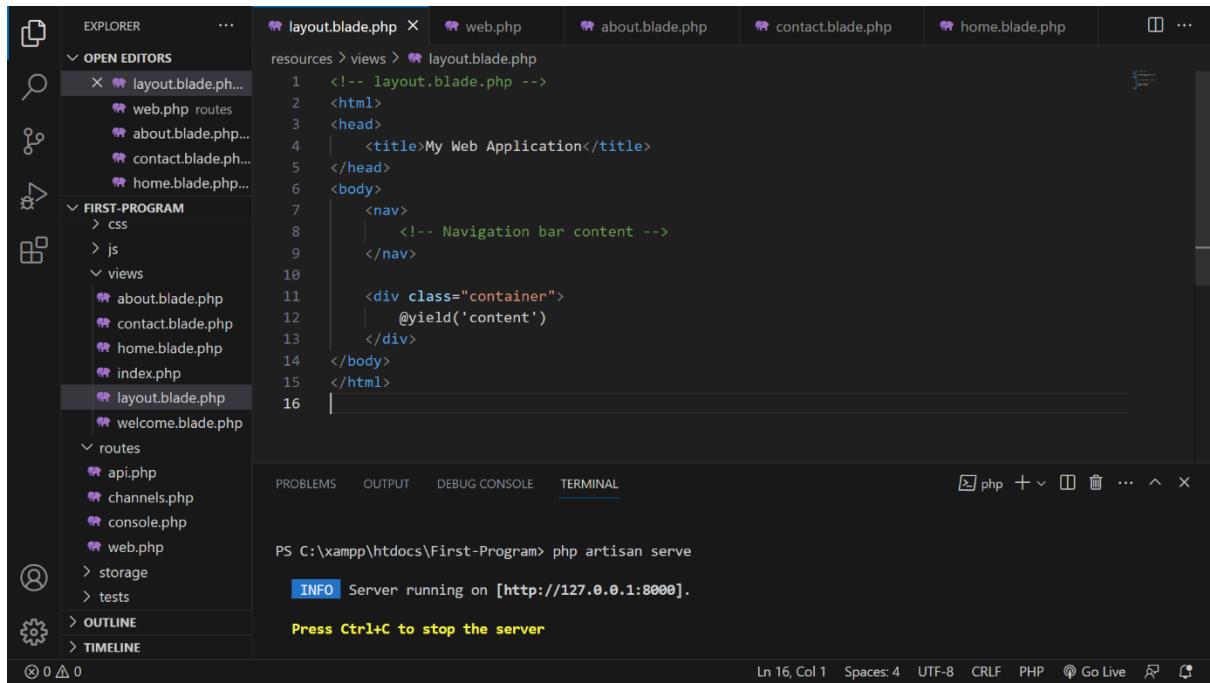
Code: -

```
<!-- layout.blade.php -->

<html>
  <head>
    <title>My Web Application</title>
  </head>
  <body>
    <nav>
      <!-- Navigation bar content -->
    </nav>

    <div class="container">
      @yield('content')
    </div>
  </body>
</html>
```

Screenshot of code:



```
resources > views > layout.blade.php
1  <!-- layout.blade.php -->
2  <html>
3  <head>
4  |   <title>My Web Application</title>
5  </head>
6  <body>
7  |   <nav>
8  |   |   <!-- Navigation bar content -->
9  |   </nav>
10 |   <div class="container">
11 |   |   @yield('content')
12 |   </div>
13 </body>
14 </html>
15
16 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\xampp\htdocs\First-Program> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Ln 16, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live

Step2:

Code:

Web.php

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
/*
```

```
|-----
```

```
| Web Routes
```

```
|-----
```

```
|
```

| Here is where you can register web routes for your application.
These

| routes are loaded by the RouteServiceProvider within a group
which

| contains the "web" middleware group. Now create something
great!

|
*/

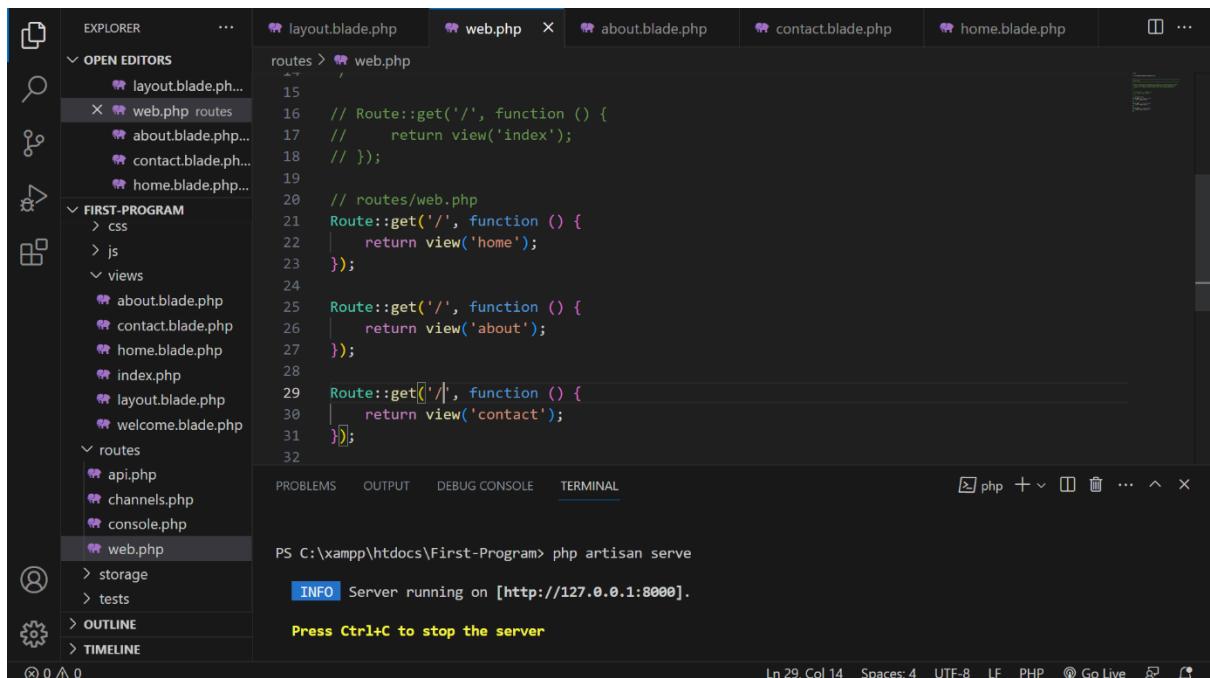
```
// Route::get('/', function () {  
//     return view('index');  
// });
```

```
// routes/web.php  
Route::get('/', function () {  
    return view('home');  
});
```

```
Route::get('/', function () {  
    return view('about');  
});
```

```
Route::get('/', function () {  
    return view('contact');  
});
```

Screenshot of code:



```
routes > web.php
15 // Route::get('/', function () {
16 //     return view('index');
17 // });
18 // routes/web.php
19 // Route::get('/', function () {
20 //     return view('home');
21 // });
22 // Route::get('/', function () {
23 //     return view('about');
24 // });
25 // Route::get(['/|', function () {
26 //     return view('contact');
27 // }]);
28
29
30
31
32
```

PS C:\xampp\htdocs\First-Program> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Step:3

Code:

```
<!-- about.blade.php -->

@extends('layout')

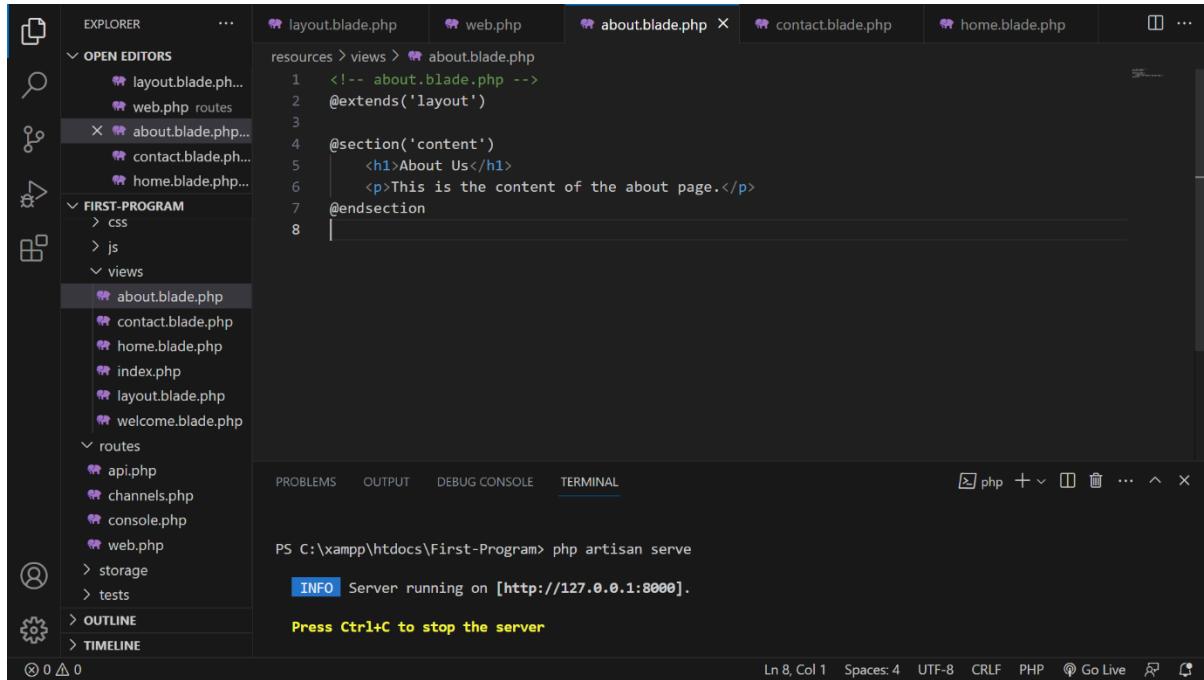
@section('content')

<h1>About Us</h1>

<p>This is the content of the about page.</p>

@endsection
```

Screenshot of code:



```
resources > views > about.blade.php
1 <!-- about.blade.php -->
2 @extends('layout')
3
4 @section('content')
5 <h1>About Us</h1>
6 <p>This is the content of the about page.</p>
7 @endsection
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\xampp\htdocs\First-Program> php artisan serve

INFO Server running on [<http://127.0.0.1:8000>].

Press Ctrl+C to stop the server

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡ Go Live ⚡

Step:4

Code:

```
<!-- contact.blade.php -->

@extends('layout')

@section('content')

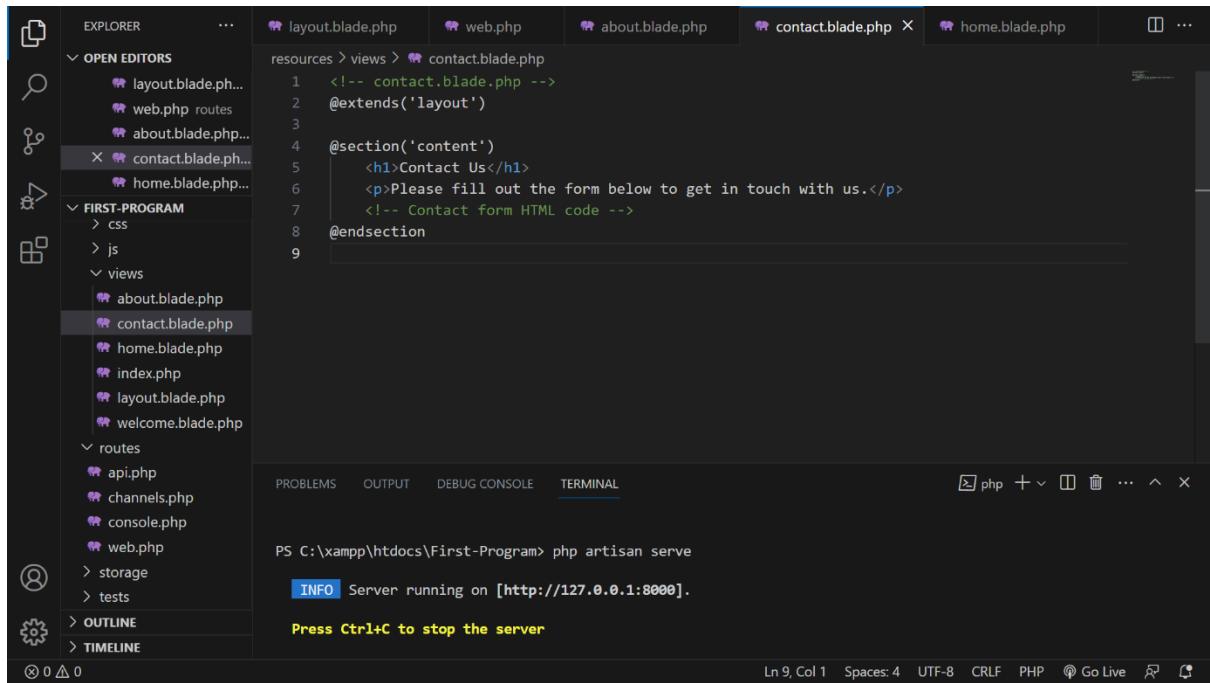
<h1>Contact Us</h1>

<p>Please fill out the form below to get in touch with us.</p>

<!-- Contact form HTML code -->

@endsection
```

Screenshot of code:



resources > views > contact.blade.php

```
1 <!-- contact.blade.php -->
2 @extends('layout')
3
4 @section('content')
5 <h1>Contact Us</h1>
6 <p>Please fill out the form below to get in touch with us.</p>
7 <!-- Contact form HTML code -->
8 @endsection
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\xampp\htdocs\First-Program> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Ln 9, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live

Step:5

Code:

```
<!-- home.blade.php -->

@extends('layout')

@section('content')

<h1>Welcome to the Home Page</h1>

<p>This is the content of the home page.</p>

@endsection
```

Screenshot of code:

The screenshot shows the VS Code interface. The Explorer sidebar on the left lists files and folders, including 'FIRST-PROGRAM' and 'views'. The 'views' folder contains 'about.blade.php', 'contact.blade.php', 'home.blade.php', 'index.php', 'layout.blade.php', and 'welcome.blade.php'. The 'home.blade.php' file is open in the editor, showing the following code:

```
resources > views > home.blade.php
1 <!-- home.blade.php -->
2 @extends('layout')
3
4 @section('content')
5 <h1>Welcome to the Home Page</h1>
6 <p>This is the content of the home page.</p>
7 @endsection
8
```

The terminal at the bottom shows the command 'php artisan serve' being run, and it outputs 'INFO Server running on [http://127.0.0.1:8000]. Press Ctrl+C to stop the server'.

Step:6

Code:

php artisan serve

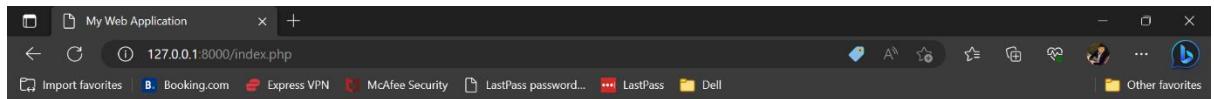
The screenshot shows the VS Code interface with the terminal tab selected. The terminal output shows the command 'php artisan serve' being run, and it outputs 'INFO Server running on [http://127.0.0.1:8000]. Press Ctrl+C to stop the server'.

Output in browser:

Go to browser and search

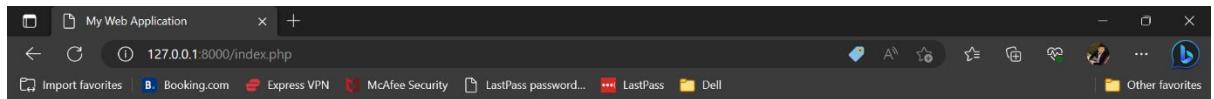
Code:

127.0.0.1:8000/index



Welcome to the Home Page

This is the content of the home page.



About Us

This is the content of the about page.



Contact Us

Please fill out the form below to get in touch with us.

4 Aim: - Create a Web Application which demonstrate routing through controller.

Learning outcome: - Able to understand Laravel Framework.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Open your Command prompt and Please enter a composer create project Laravel/Laravel

Then after write code.

Step1:-

Open CMD and Write code

First create a controller

Php artisan make: Controller MyController

Then go to MyController.php

Then after write code.

Code MyController.php: -

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class MyController extends Controller
{
    public function index()
    {
        echo 'Hello Rakesh Kumar This is Index method to print Containt';
    }

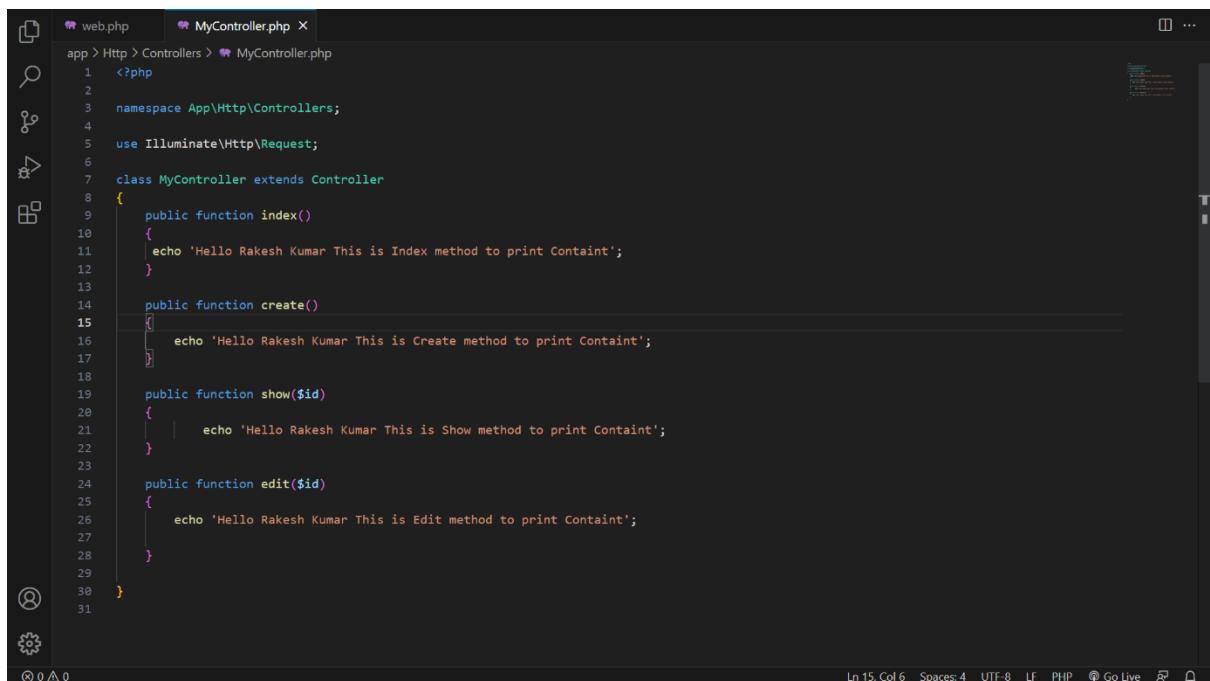
    public function create()
    {
        echo 'Hello Rakesh Kumar This is Create method to print
Containt';
    }

    public function show($id)
    {
        echo 'Hello Rakesh Kumar This is Show method to print
Containt';
    }

    public function edit($id)
    {
        echo 'Hello Rakesh Kumar This is Edit method to print Containt';
    }
}
```

```
}
```

Screenshot of code MyController.php:



The screenshot shows a code editor with a dark theme. The file being edited is `MyController.php`, located in the `app/Http/Controllers` directory. The code defines a `MyController` class that extends `Controller`. It contains four methods: `index()`, `create()`, `show($id)`, and `edit($id)`. Each method contains an `echo` statement that prints a message to the console. The code editor interface includes a sidebar with file navigation, a status bar at the bottom, and a floating toolbar on the right.

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class MyController extends Controller
{
    public function index()
    {
        echo 'Hello Rakesh Kumar This is Index method to print Containt';
    }
    public function create()
    {
        echo 'Hello Rakesh Kumar This is Create method to print Containt';
    }
    public function show($id)
    {
        echo 'Hello Rakesh Kumar This is Show method to print Containt';
    }
    public function edit($id)
    {
        echo 'Hello Rakesh Kumar This is Edit method to print contain';
    }
}
```

Code Web.php:

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
// |-----  
// | Web Routes  
// |-----  
// |
```

```
// | Here is where you can register web routes for your application.  
These
```

```
// | routes are loaded by the RouteServiceProvider within a group  
which
```

```
// | contains the "web" middleware group. Now create something  
great!
```

```
// |
```

```
use App\Http\Controllers\MyController;
```

```
Route::get('/index', [MyController::class, 'index']);
```

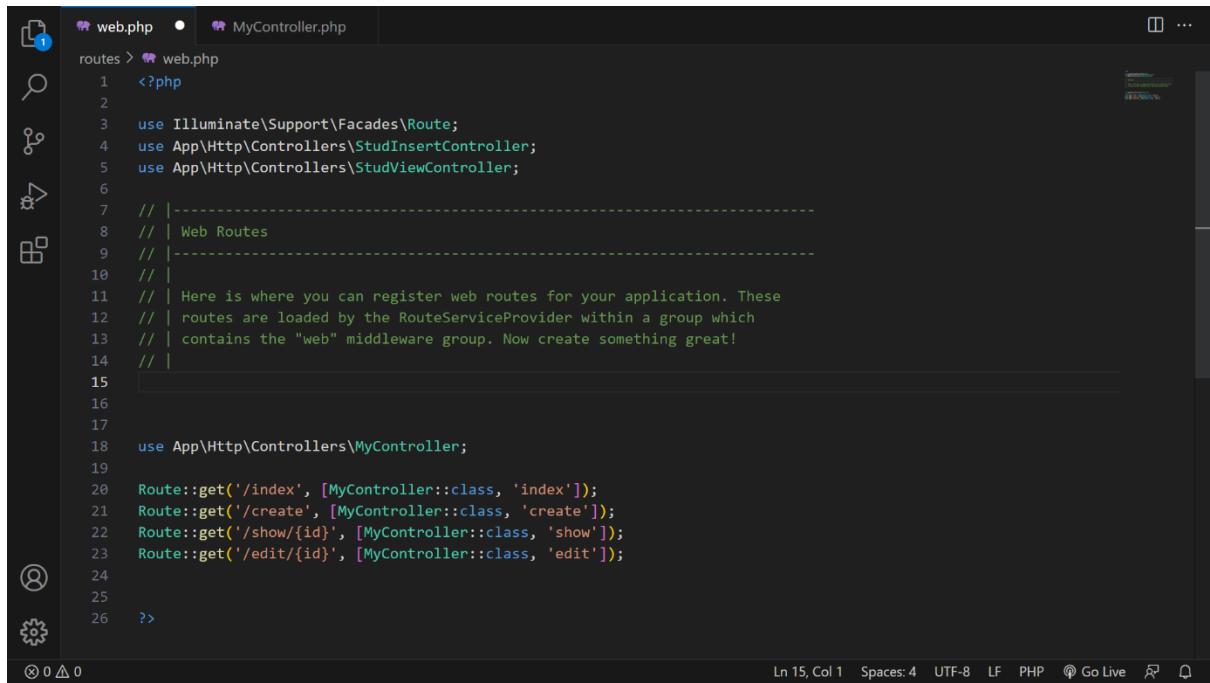
```
Route::get('/create', [MyController::class, 'create']);
```

```
Route::get('/show/{id}', [MyController::class, 'show']);
```

```
Route::get('/edit/{id}', [MyController::class, 'edit']);
```

```
?>
```

Screenshot of code Web.php:



```
routes > routes/web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\StudInsertController;
5  use App\Http\Controllers\StudViewController;
6
7  // | -----
8  // | Web Routes
9  // | -----
10 // |
11 // | Here is where you can register web routes for your application. These
12 // | routes are loaded by the RouteServiceProvider within a group which
13 // | contains the "web" middleware group. Now create something great!
14 // |
15
16
17
18 use App\Http\Controllers\MyController;
19
20 Route::get('/index', [MyController::class, 'index']);
21 Route::get('/create', [MyController::class, 'create']);
22 Route::get('/show/{id}', [MyController::class, 'show']);
23 Route::get('/edit/{id}', [MyController::class, 'edit']);
24
25
26 ?>
```

Ln 15, Col 1 Spaces: 4 UTF-8 LF PHP ⌂ Go Live ⌂ ⌂

Then go Browser and search code of calling:

127.0.0.1:8000/index

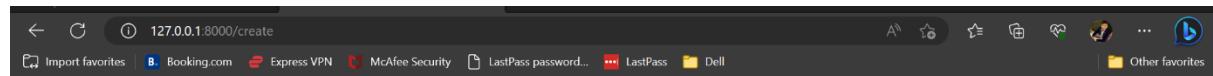
Output:



Hello Rakesh Kumar This is Index method to print Containt

127.0.0.1:8000/create

Output:



Hello Rakesh Kumar This is Create method to print Containt

[**127.0.0.1:8000/show/{id}**](http://127.0.0.1:8000/show/{id})

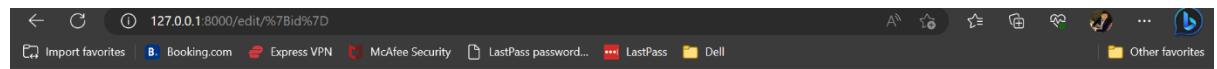
Output:



Hello Rakesh Kumar This is Show method to print Containt

[**127.0.0.1:8000/edit/{id}**](http://127.0.0.1:8000/edit/{id})

Output:



Hello Rakesh Kumar This is Edit method to print Containt

5 **Aim:** - Create a Form which inserts the data in the database.

Learning outcome: - Able to understand in Laravel Framework.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:-

Open your Command prompt and Please enter a Laravel command

```
composer create-project --prefer-dist
laravel/laravel form-data
```

After you download it and put it in your **htdocs** folder if you're using **xampp**. Then run the following command:

```
composer update
```

Once done above we need to install the **Laravel Collective Package**, run the following command below:

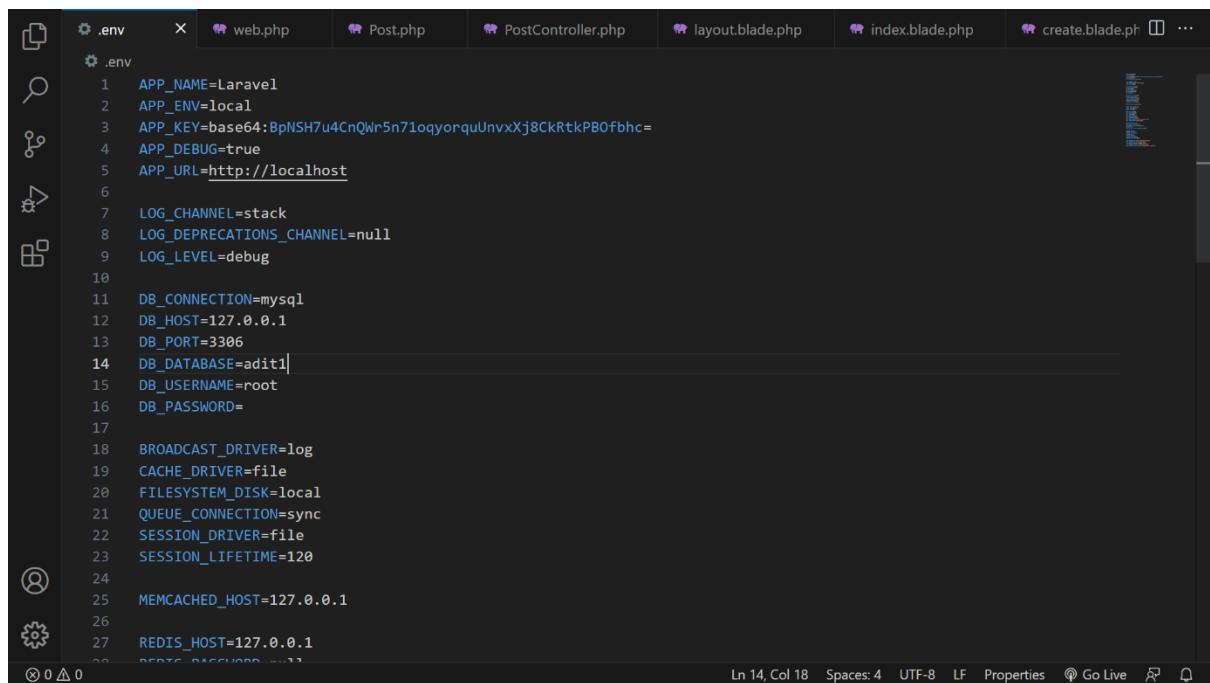
```
composer require laravelcollective/html
```

Step 2: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:BpNSH7u4CnQWr5n71oqyorquUnvxXj8CkRtkPBOfbhc=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=adit1
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=
```

Step 3: Create C:\xampp\htdocs\form-data\app\Http\Controllers\StudInsertController.php

Code:

```
<?php
```

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use DB;

class StudInsertController extends Controller
{
    public function insertform()
    {
        return view('stud_create');
    }

    public function insert(Request $request)
    {
        $first_name = $request->input('first_name');
        $last_name = $request->input('last_name');
        $city_name = $request->input('city_name');
        $email = $request->input('email');

        $data = array(
            'first_name' => $first_name,
            'last_name' => $last_name,
            'city_name' => $city_name,
            'email' => $email
        )
    }
}
```

```
});
```

```
DB::table('student_details')->insert($data);
```

```
echo "Record inserted successfully.<br/>";
```

```
echo '<a href="/insertform">Click Here</a> To Go Back.';
```

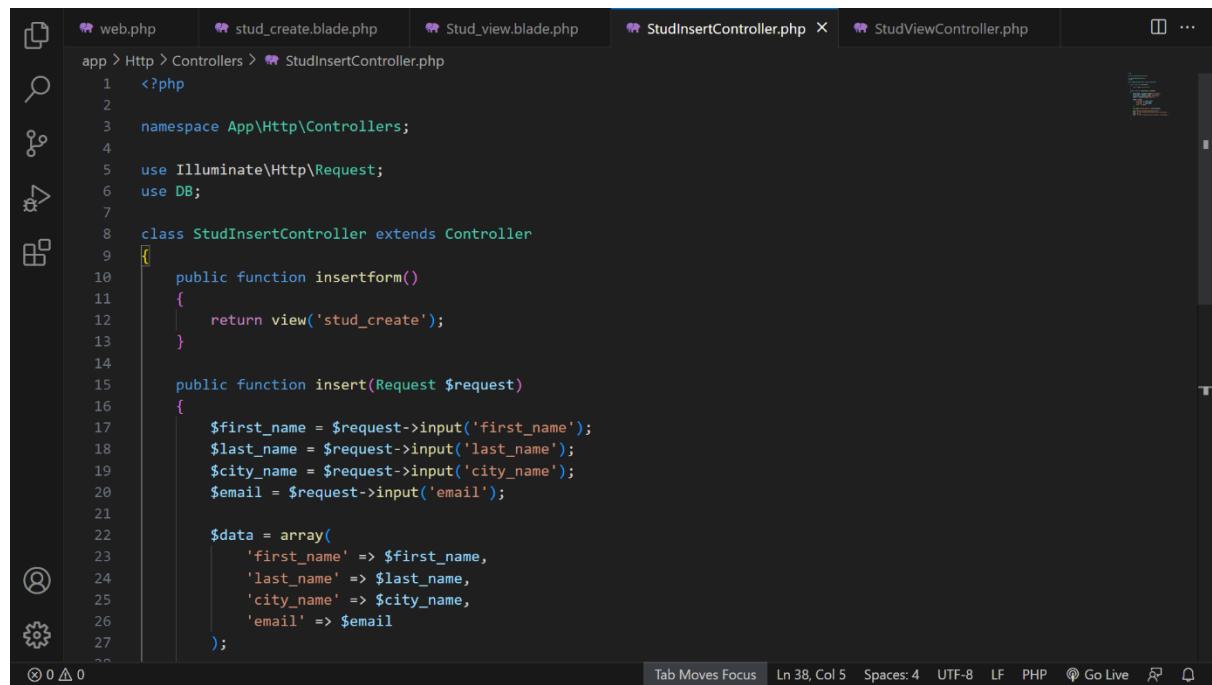
```
echo '<br>';
```

```
echo '<a href="/stud_view">Click Here</a> To View Data.';
```

```
}
```

```
}
```

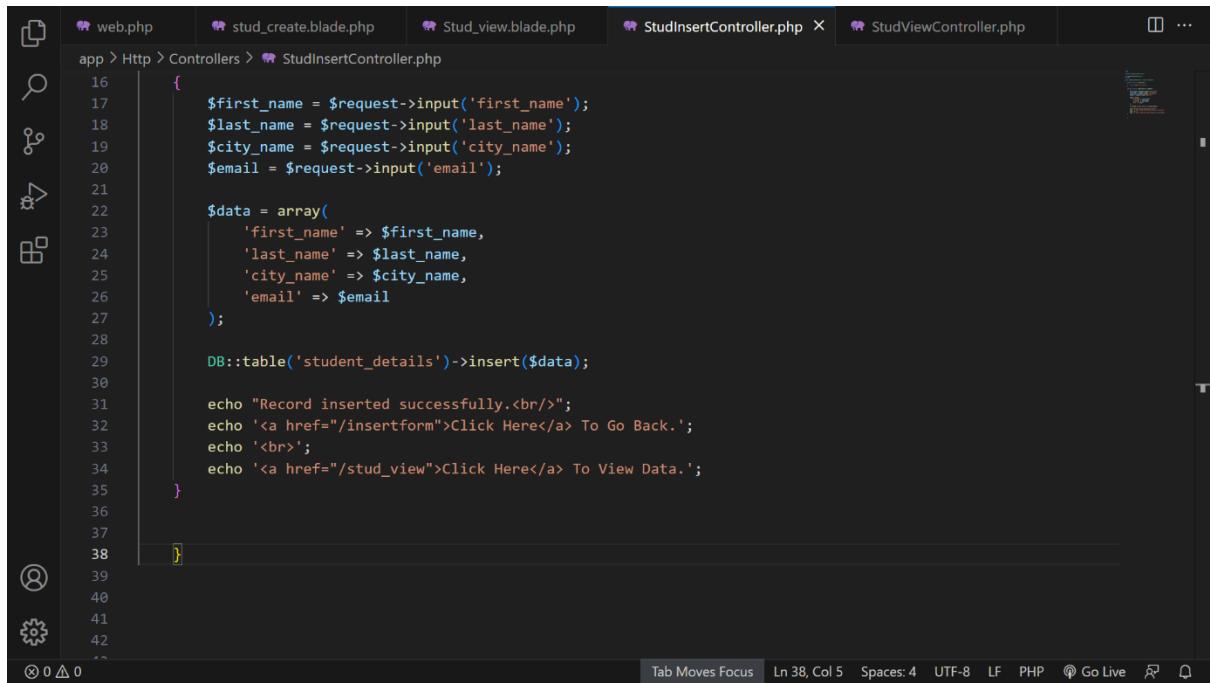
Screenshot of code:



The screenshot shows a code editor with a dark theme. The top navigation bar includes tabs for 'web.php', 'stud_create.blade.php', 'Stud_view.blade.php', 'StudInsertController.php' (which is the active tab), and 'StudViewController.php'. The code editor displays the following PHP code:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use DB;
7
8 class StudInsertController extends Controller
9 {
10     public function insertform()
11     {
12         return view('stud_create');
13     }
14
15     public function insert(Request $request)
16     {
17         $first_name = $request->input('first_name');
18         $last_name = $request->input('last_name');
19         $city_name = $request->input('city_name');
20         $email = $request->input('email');
21
22         $data = array(
23             'first_name' => $first_name,
24             'last_name' => $last_name,
25             'city_name' => $city_name,
26             'email' => $email
27         );
28     }
29
30 }
```

The code is a Laravel controller named 'StudInsertController'. It contains two methods: 'insertform()' which returns a view named 'stud_create', and 'insert()' which handles an incoming request. Inside 'insert()', it retrieves input from the request for 'first_name', 'last_name', 'city_name', and 'email', and then creates an array '\$data' with these values. The code editor interface includes various icons for file operations and a status bar at the bottom.



```
web.php  stud_create.blade.php  Stud_view.blade.php  StudInsertController.php  StudViewController.php  ...  
app > Http > Controllers > StudInsertController.php  
16  {  
17      $first_name = $request->input('first_name');  
18      $last_name = $request->input('last_name');  
19      $city_name = $request->input('city_name');  
20      $email = $request->input('email');  
21  
22      $data = array(  
23          'first_name' => $first_name,  
24          'last_name' => $last_name,  
25          'city_name' => $city_name,  
26          'email' => $email  
27      );  
28  
29      DB::table('student_details')->insert($data);  
30  
31      echo "Record inserted successfully.<br/>";  
32      echo '<a href="/insertform">Click Here</a> To Go Back.';  
33      echo '<br>';  
34      echo '<a href="/stud_view">Click Here</a> To View Data.';  
35  }  
36  
37  }  
38 }  
39  
40  
41  
42  
Tab Moves Focus | Ln 38, Col 5 | Spaces: 4 | UTF-8 | LF | PHP | Go Live |  
```

Step 4: Create C:\xampp\htdocs\form-data\app\Http\Controllers\StudViewController.php

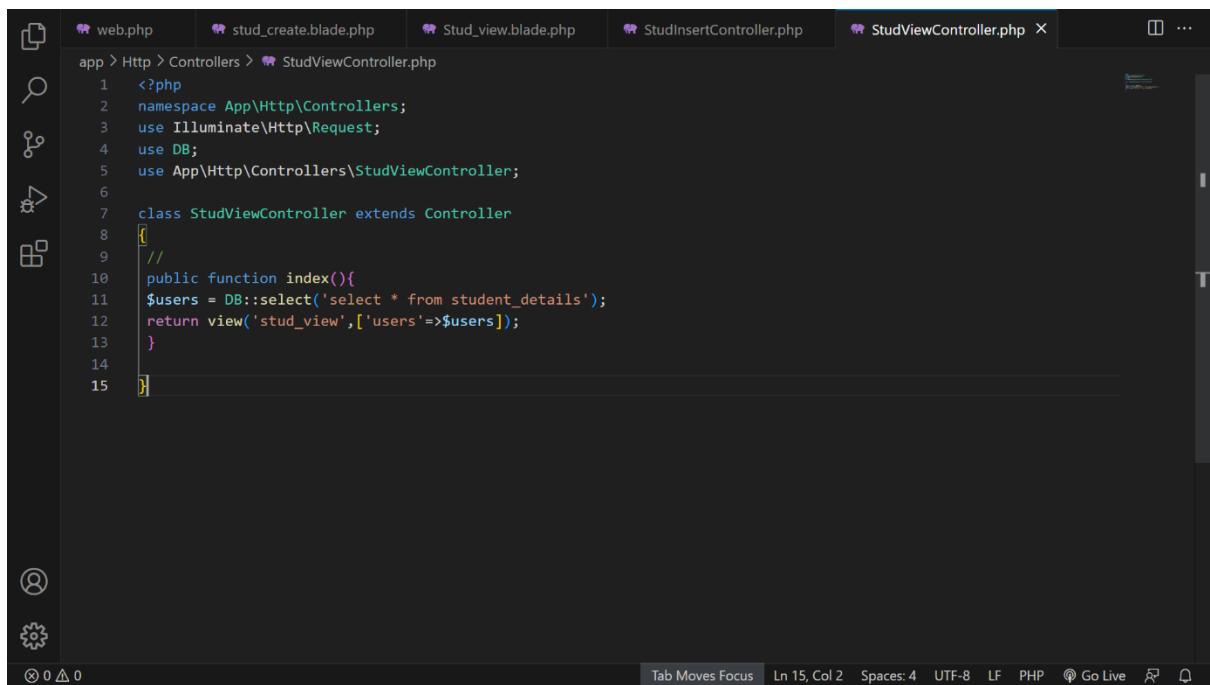
Code:

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
use DB;  
use App\Http\Controllers\StudViewController;
```

```
class StudViewController extends Controller  
{  
//  
public function index(){  
$users = DB::select('select * from student_details');
```

```
return view('stud_view',['users'=>$users]);  
}  
  
}
```

Screenshot of code:



The screenshot shows a code editor with a dark theme. The top navigation bar includes tabs for 'web.php', 'stud_create.blade.php', 'Stud_view.blade.php', 'StudInsertController.php', 'StudViewController.php', and a closed tab. On the left, there are several icons for file operations like copy, paste, search, and refresh. The main code area contains the following PHP code:

```
app > Http > Controllers > StudViewController.php
1  <?php
2  namespace App\Http\Controllers;
3  use Illuminate\Http\Request;
4  use DB;
5  use App\Http\Controllers\StudViewController;
6
7  class StudViewController extends Controller
8  {
9      //
10     public function index(){
11         $users = DB::select('select * from student_details');
12         return view('stud_view',['users'=>$users]);
13     }
14 }
15 }
```

The status bar at the bottom shows 'Tab Moves Focus' and 'Ln 15, Col 2'. Other status indicators include 'Spaces: 4', 'UTF-8', 'LF', 'PHP', 'Go Live', and a refresh icon.

Step 5: Create C:\xampp\htdocs\form-data\resources\views\stud_create.blade.php

Code:

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<title>Student Management</title>  
  
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

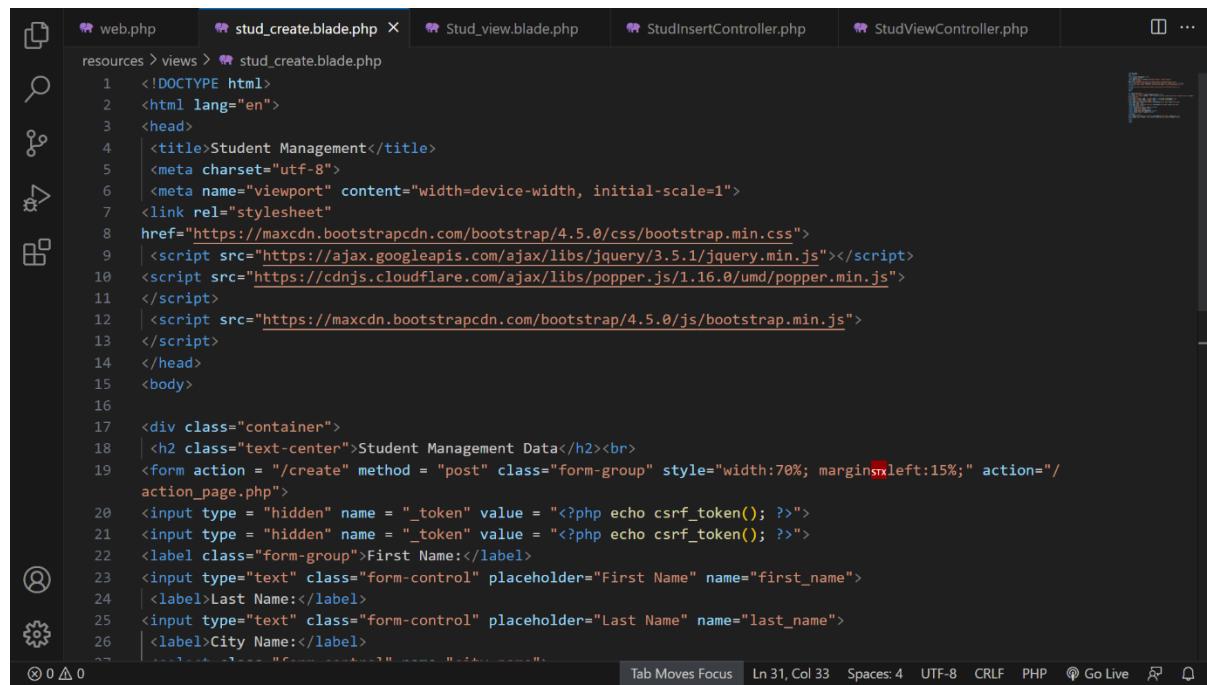
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js">
</script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">
</script>
</head>
<body>
```

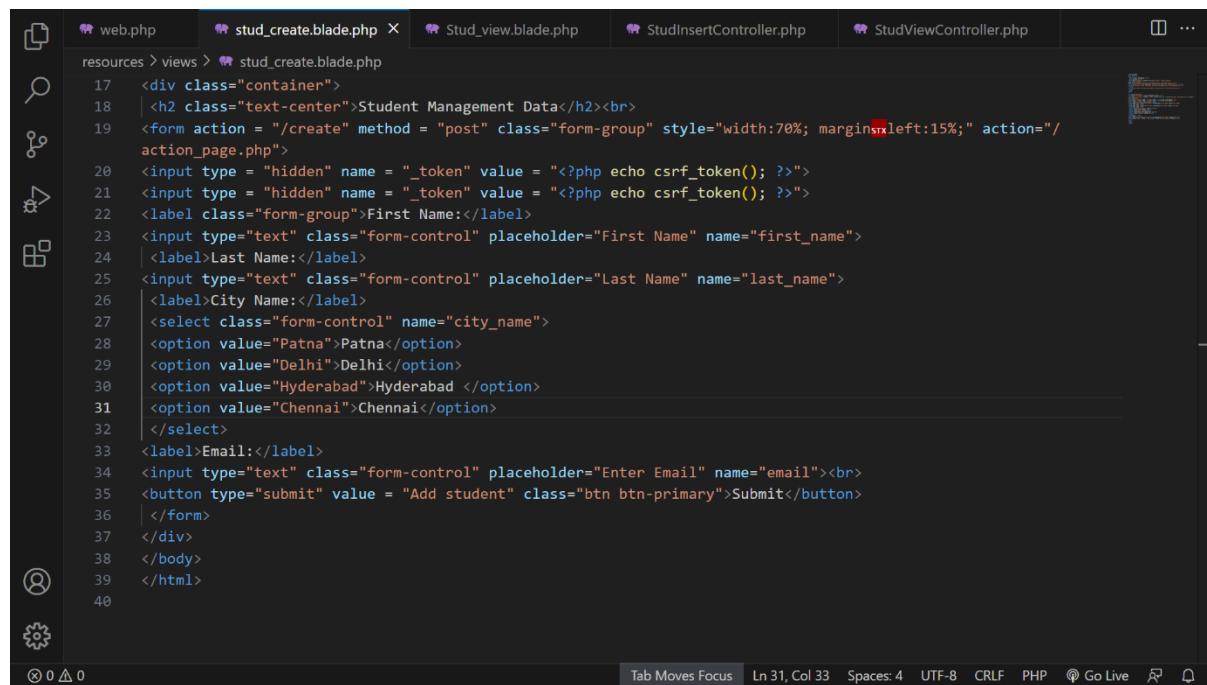
```
<div class="container">
<h2 class="text-center">Student Management Data</h2><br>
<form action = "/create" method = "post" class="form-group"
style="width:70%; margin-left:15%; action="/action_page.php">
<input type = "hidden" name = "_token" value = "<?php echo
csrf_token(); ?>">
<input type = "hidden" name = "_token" value = "<?php echo
csrf_token(); ?>">
```

```
<label class="form-group">First Name:</label>
<input type="text" class="form-control" placeholder="First Name"
name="first_name">
<label>Last Name:</label>
<input type="text" class="form-control" placeholder="Last Name"
name="last_name">
<label>City Name:</label>
<select class="form-control" name="city_name">
<option value="Patna">Patna</option>
<option value="Delhi">Delhi</option>
<option value="Hyderabad">Hyderabad </option>
<option value="Chennai">Chennai</option>
</select>
<label>Email:</label>
<input type="text" class="form-control" placeholder="Enter Email"
name="email"><br>
<button type="submit" value = "Add student" class="btn btn-primary">Submit</button>
</form>
</div>
</body>
</html>
```

Screenshot of code:



```
resources > views > stud_create.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <title>Student Management</title>
5  <meta charset="utf-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1">
7  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
8  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@1.16.0/dist/umd/popper.min.js">
10 </script>
11 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">
12 </script>
13 </head>
14 <body>
15
16 <div class="container">
17 <h2 class="text-center">Student Management Data</h2><br>
18 <form action = "/create" method = "post" class="form-group" style="width:70%; margin-left:15%;" action="/action_page.php">
19 <input type = "hidden" name = "_token" value = "<?php echo csrf_token(); ?>">
20 <input type = "hidden" name = "_token" value = "<?php echo csrf_token(); ?>">
21 <label class="form-group">First Name:</label>
22 <input type="text" class="form-control" placeholder="First Name" name="first_name">
23 <label>Last Name:</label>
24 <input type="text" class="form-control" placeholder="Last Name" name="last_name">
25 <label>City Name:</label>
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```



```
resources > views > stud_create.blade.php
17 <div class="container">
18 <h2 class="text-center">Student Management Data</h2><br>
19 <form action = "/create" method = "post" class="form-group" style="width:70%; margin-left:15%;" action="/action_page.php">
20 <input type = "hidden" name = "_token" value = "<?php echo csrf_token(); ?>">
21 <input type = "hidden" name = "_token" value = "<?php echo csrf_token(); ?>">
22 <label class="form-group">First Name:</label>
23 <input type="text" class="form-control" placeholder="First Name" name="first_name">
24 <label>Last Name:</label>
25 <input type="text" class="form-control" placeholder="Last Name" name="last_name">
26 <label>City Name:</label>
27 <select class="form-control" name="city_name">
28 <option value="Patna">Patna</option>
29 <option value="Delhi">Delhi</option>
30 <option value="Hyderabad">Hyderabad </option>
31 <option value="Chennai">Chennai</option>
32 </select>
33 <label>Email:</label>
34 <input type="text" class="form-control" placeholder="Enter Email" name="email"><br>
35 <button type="submit" value = "Add student" class="btn btn-primary">Submit</button>
36
37 </div>
38 </body>
39 </html>
40
```

Step 6: Create C:\xampp\htdocs\form-data\resources\views\stud_view.blade.php

Code:

```
<!DOCTYPE html>
```

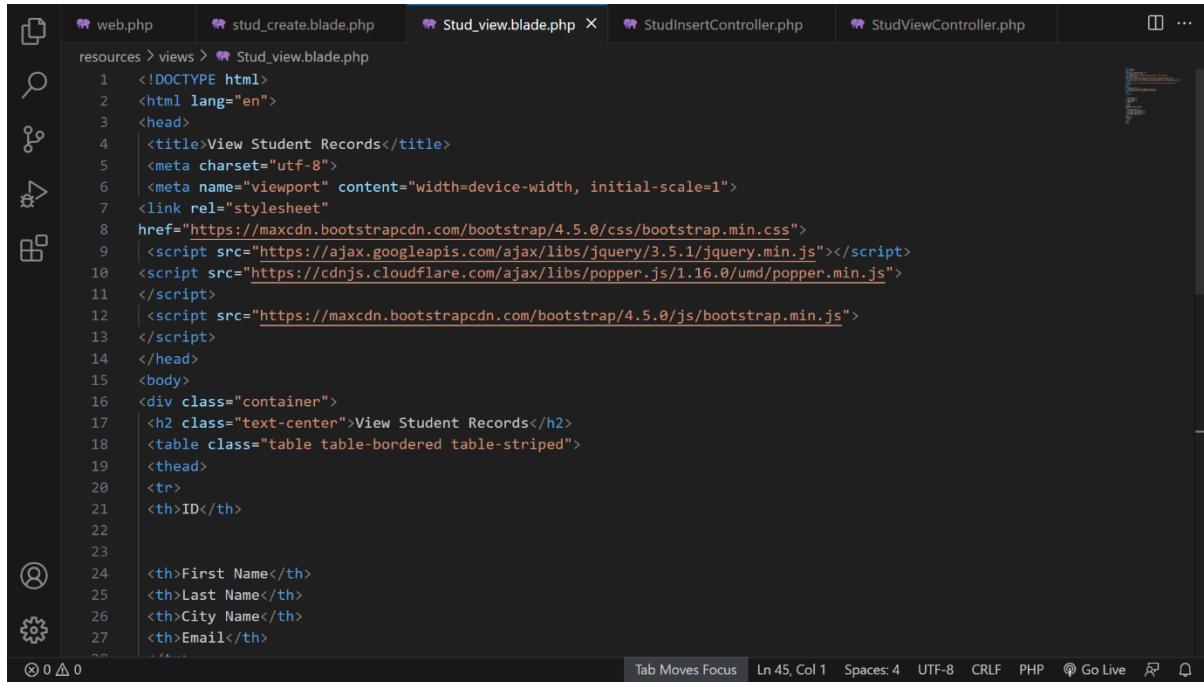
```
<html lang="en">
<head>
<title>View Student Records</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
<script
      src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"
      ></script>
<script
      src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/umd/popper.min.js">
</script>
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">
</script>
</head>
<body>
<div class="container">
<h2 class="text-center">View Student Records</h2>
<table class="table table-bordered table-striped">
<thead>
```

```
<tr>
<th>ID</th>

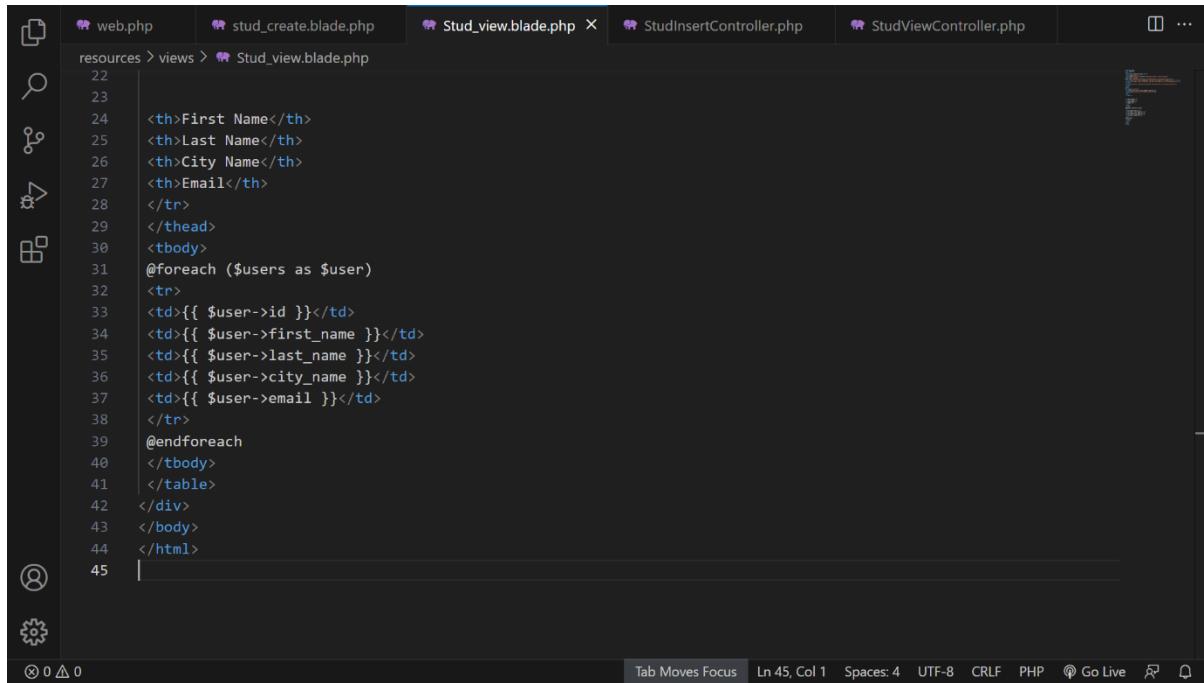
<th>First Name</th>
<th>Last Name</th>
<th>City Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
@foreach ($users as $user)
<tr>
<td>{{ $user->id }}</td>
<td>{{ $user->first_name }}</td>
<td>{{ $user->last_name }}</td>
<td>{{ $user->city_name }}</td>
<td>{{ $user->email }}</td>
</tr>
@endforeach
</tbody>
</table>
</div>
</body>
```

```
</html>
```

Screenshot of code:



```
resources > views > Stud_view.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <title>View Student Records</title>
5  <meta charset="utf-8">
6  <meta name="viewport" content="width=device-width, initial-scale=1">
7  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
8  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js">
10 </script>
11 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">
12 </script>
13 </head>
14 <body>
15 <div class="container">
16 <h2 class="text-center">View Student Records</h2>
17 <table class="table table-bordered table-striped">
18 <thead>
19 <tr>
20 <th>ID</th>
21 <th>First Name</th>
22 <th>Last Name</th>
23 <th>City Name</th>
24 <th>Email</th>
25 </thead>
26 <tbody>
27 <tr>
28 <td>{{ $user->id }}</td>
29 <td>{{ $user->first_name }}</td>
30 <td>{{ $user->last_name }}</td>
31 <td>{{ $user->city_name }}</td>
32 <td>{{ $user->email }}</td>
33 </tr>
34 @foreach ($users as $user)
35 <tr>
36 <td>{{ $user->id }}</td>
37 <td>{{ $user->first_name }}</td>
38 <td>{{ $user->last_name }}</td>
39 <td>{{ $user->city_name }}</td>
40 <td>{{ $user->email }}</td>
41 </tr>
42 @endforeach
43 </tbody>
44 </table>
45 </div>
```



```
resources > views > Stud_view.blade.php
22
23
24 <th>First Name</th>
25 <th>Last Name</th>
26 <th>City Name</th>
27 <th>Email</th>
28 </tr>
29 </thead>
30 <tbody>
31 @foreach ($users as $user)
32 <tr>
33 <td>{{ $user->id }}</td>
34 <td>{{ $user->first_name }}</td>
35 <td>{{ $user->last_name }}</td>
36 <td>{{ $user->city_name }}</td>
37 <td>{{ $user->email }}</td>
38 </tr>
39 @endforeach
40 </tbody>
41 </table>
42 </div>
43 </body>
44 </html>
45 |
```

Step 7: Create Route C:\xampp\htdocs\form-data\routes\web.php

In my example, I will create manually my routes. Just open the **"routes/web.php"** file and add the following routes.

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\StudInsertController;
use App\Http\Controllers\StudViewController;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application.
| These
|
| routes are loaded by the RouteServiceProvider within a group
| which
|
| contains the "web" middleware group. Now create something
| great!
|
*/
Route::get('/', function () {
    return view('welcome');
});
```

```

Route::group([],function(){
    Route::get('/insertform','App\Http\Controllers\StudInsertController
    @insertform');

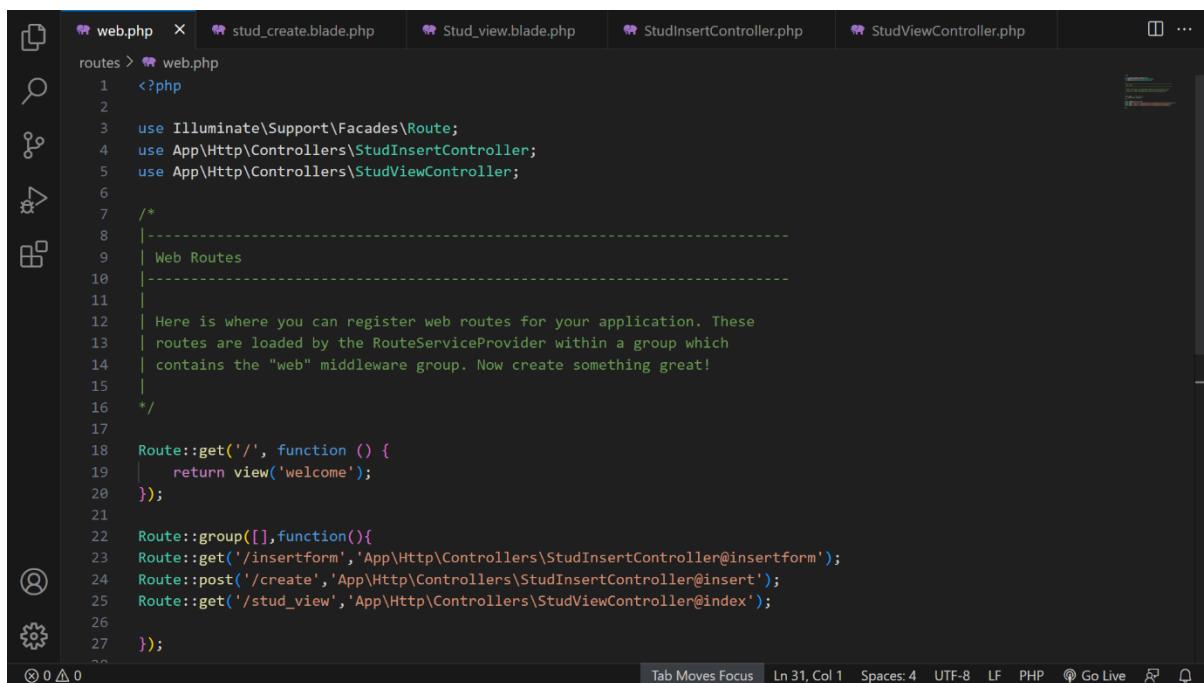
    Route::post('/create','App\Http\Controllers\StudInsertController@ins
    ert');

    Route::get('/stud_view','App\Http\Controllers\StudViewController@i
    ndex');

});


```

Screenshot of code Web.php:



```

routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\StudInsertController;
5  use App\Http\Controllers\StudViewController;
6
7  /*
8  | -----
9  | Web Routes
10 |
11 |
12 | Here is where you can register web routes for your application. These
13 | routes are loaded by the RouteServiceProvider within a group which
14 | contains the "web" middleware group. Now create something great!
15 |
16 */
17
18 Route::get('/', function () {
19     return view('welcome');
20 });
21
22 Route::group([],function(){
23     Route::get('/insertform','App\Http\Controllers\StudInsertController@insertform');
24     Route::post('/create','App\Http\Controllers\StudInsertController@insert');
25     Route::get('/stud_view','App\Http\Controllers\StudViewController@index');
26 });
27

```

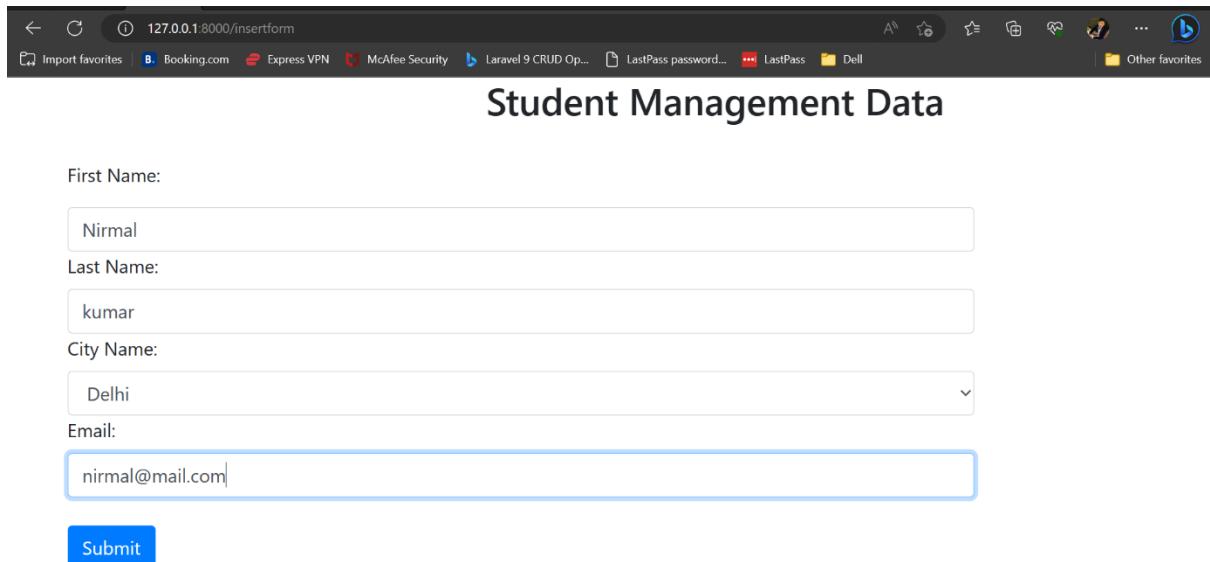
Step 8:
Write code :
 Php artisan serve

127.0.0.1:8000/createform

Screenshot

Output:

Fill the form



127.0.0.1:8000/insertform

Student Management Data

First Name:

Last Name:

City Name:

Email:

Submit

Submit data in database

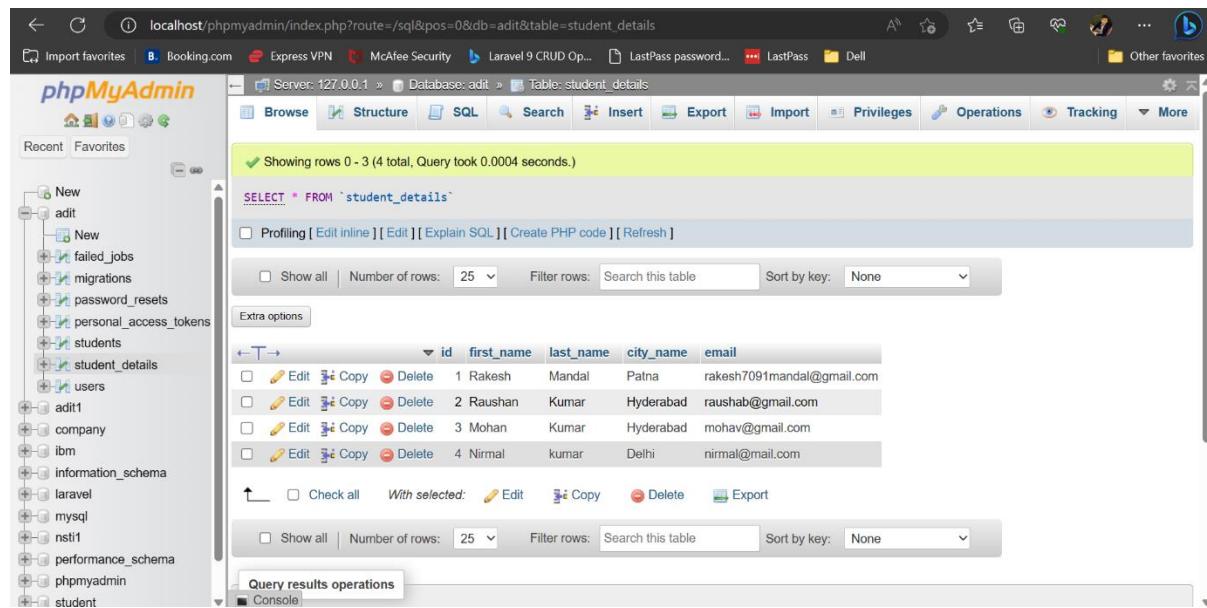
Record inserted successfully.

[Click Here](#) To Go Back.

[Click Here](#) To View Data.

127.0.0.1:8000/insertform

Show data in database



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=adit&table=student_details

phpMyAdmin

Recent Favorites

Server: 127.0.0.1 » Database: adit » Table: student_details

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking More

Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

SELECT * FROM `student_details`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	id	first_name	last_name	city_name	email
<input type="checkbox"/>	1	Rakesh	Mandal	Patna	rakesh7091mandal@gmail.com
<input type="checkbox"/>	2	Raushan	Kumar	Hyderabad	raushab@gmail.com
<input type="checkbox"/>	3	Mohan	Kumar	Hyderabad	mohav@gmail.com
<input type="checkbox"/>	4	Nirmal	kumar	Delhi	nirmal@mail.com

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

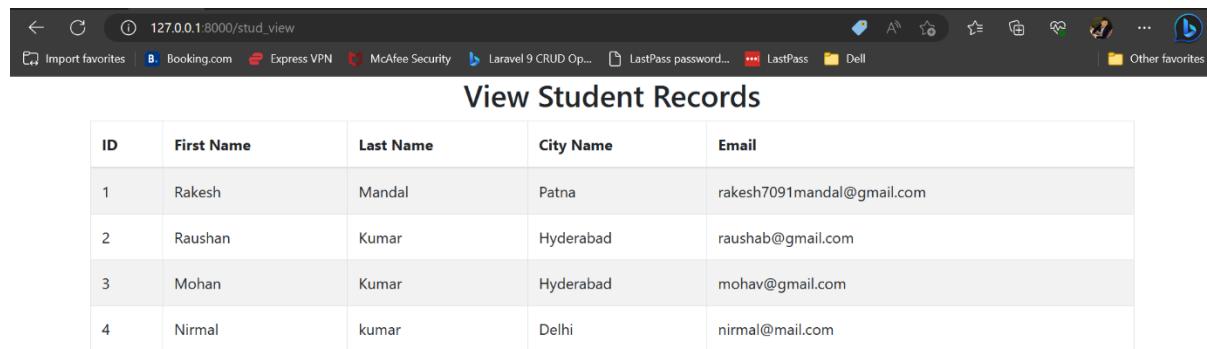
Query results operations

Console

The screenshot shows the phpMyAdmin interface for the 'adit' database. The 'student_details' table is selected. The table has 4 rows with the following data:

ID	first_name	last_name	city_name	email
1	Rakesh	Mandal	Patna	rakesh7091mandal@gmail.com
2	Raushan	Kumar	Hyderabad	raushab@gmail.com
3	Mohan	Kumar	Hyderabad	mohav@gmail.com
4	Nirmal	kumar	Delhi	nirmal@mail.com

View data



127.0.0.1:8000/stud_view

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

View Student Records

ID	First Name	Last Name	City Name	Email
1	Rakesh	Mandal	Patna	rakesh7091mandal@gmail.com
2	Raushan	Kumar	Hyderabad	raushab@gmail.com
3	Mohan	Kumar	Hyderabad	mohav@gmail.com
4	Nirmal	kumar	Delhi	nirmal@mail.com

6 Aim: - Demonstrate all the CRUD operation using Database form controllers.

Learning outcome: - Able to understand CRUD operation in Laravel Framework.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:

Open your Command prompt and Please enter a Laravel command
`composer create-project --prefer-dist laravel/laravel crud-opration`

After you download it and put it in your `htdocs` folder if you're using `xampp`. Then run the following command:

```
composer update
```

Once done above we need to install the **Laravel Collective Package**, run the following command below:

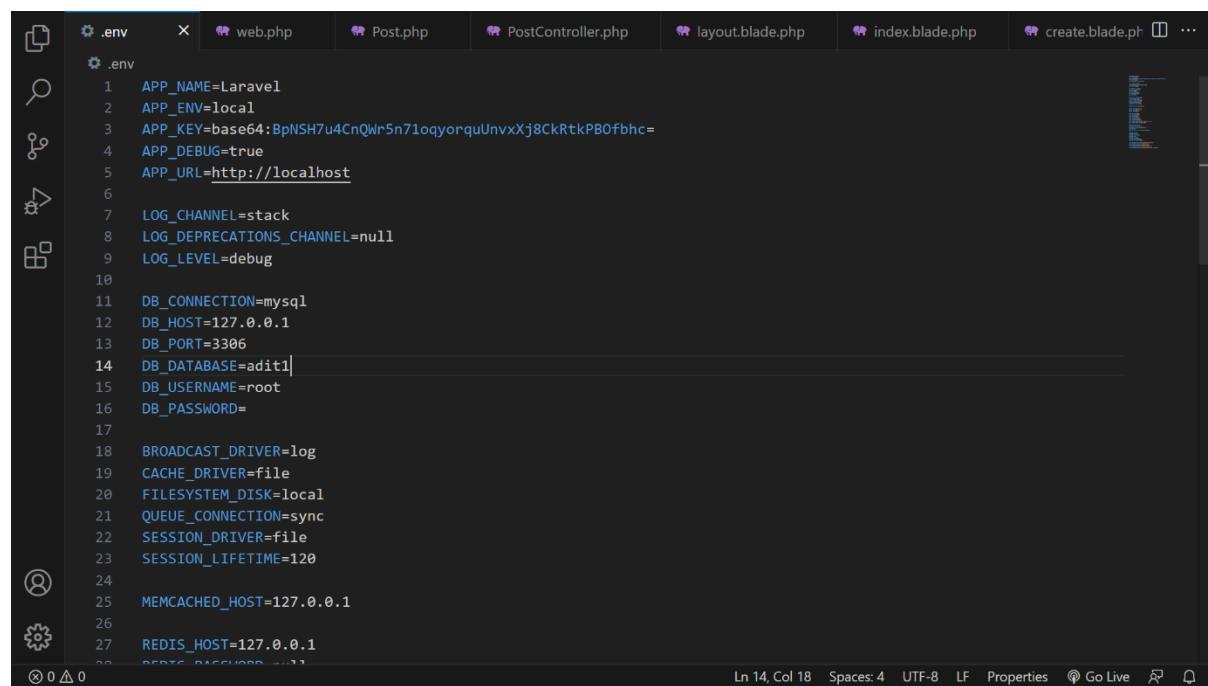
```
composer require laravelcollective/html
```

Step 2: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:BpNSH7u4CnQWr5n71oqyorquUnvxXj8CkRtkPB0fbhc=
APP_DEBUG=true
APP_URL=http://localhost
LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=adit1
DB_USERNAME=root
DB_PASSWORD=
BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
MEMCACHED_HOST=127.0.0.1
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
```

Step 3: Migration Setup

Let's create a migration for our Laravel crud operation. If you are using a fresh Laravel project we need to customize the

user's table because I added the username field in my authentication. Here are the complete migrations below:

Code:

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreatePostsTable extends Migration {

    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name')->nullable();
            $table->longText('detail')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *

```

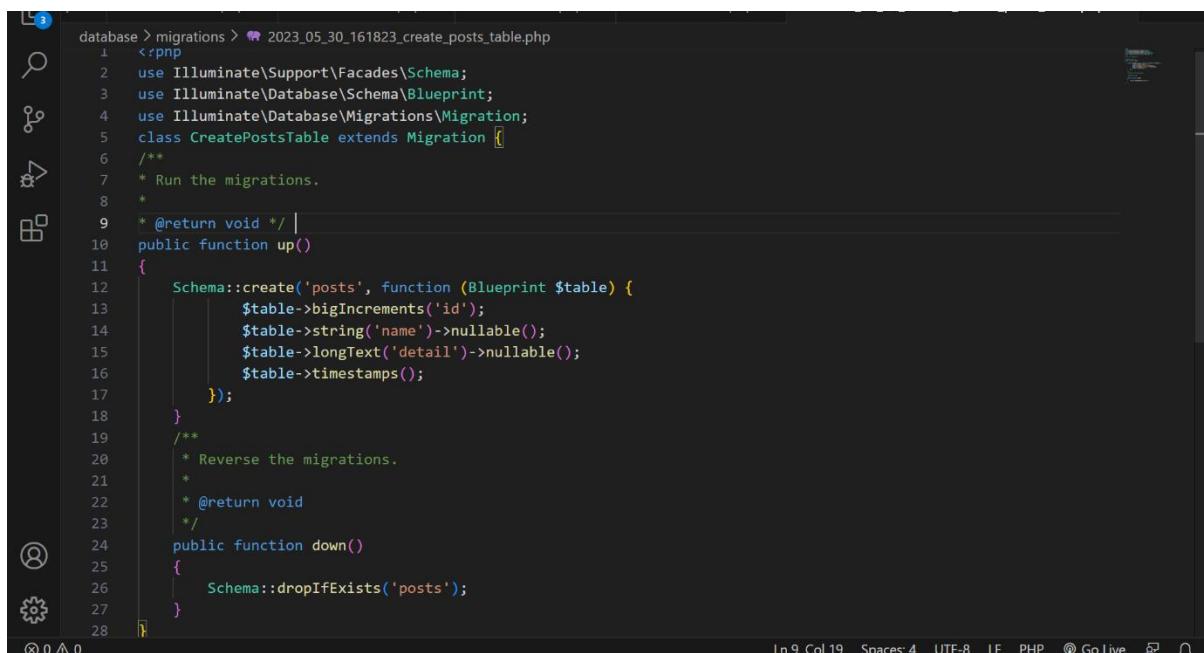
```

* @return void
*/
public function down()
{
    Schema::dropIfExists('posts');
}

}

```

Screenshot of code:



```

database > migrations > 2023_05_30_161823_create_posts_table.php
1  <?php
2  use Illuminate\Support\Facades\Schema;
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Database\Migrations\Migration;
5  class CreatePostsTable extends Migration
6  {
7      /**
8      * Run the migrations.
9      */
10     * @return void
11     */
12     public function up()
13     {
14         Schema::create('posts', function (Blueprint $table) {
15             $table->bigIncrements('id');
16             $table->string('name')->nullable();
17             $table->longText('detail')->nullable();
18             $table->timestamps();
19         });
20     }
21     /**
22     * Reverse the migrations.
23     */
24     * @return void
25     */
26     public function down()
27     {
28         Schema::dropIfExists('posts');
29     }
30 }

```

Now let's run the following command below:

```
php artisan migrate
```

Step 4: Laravel CRUD Operation Routes

In my example, I will create manually my CRUD routes. Just open the **"routes/web.php"** file and add the following routes.

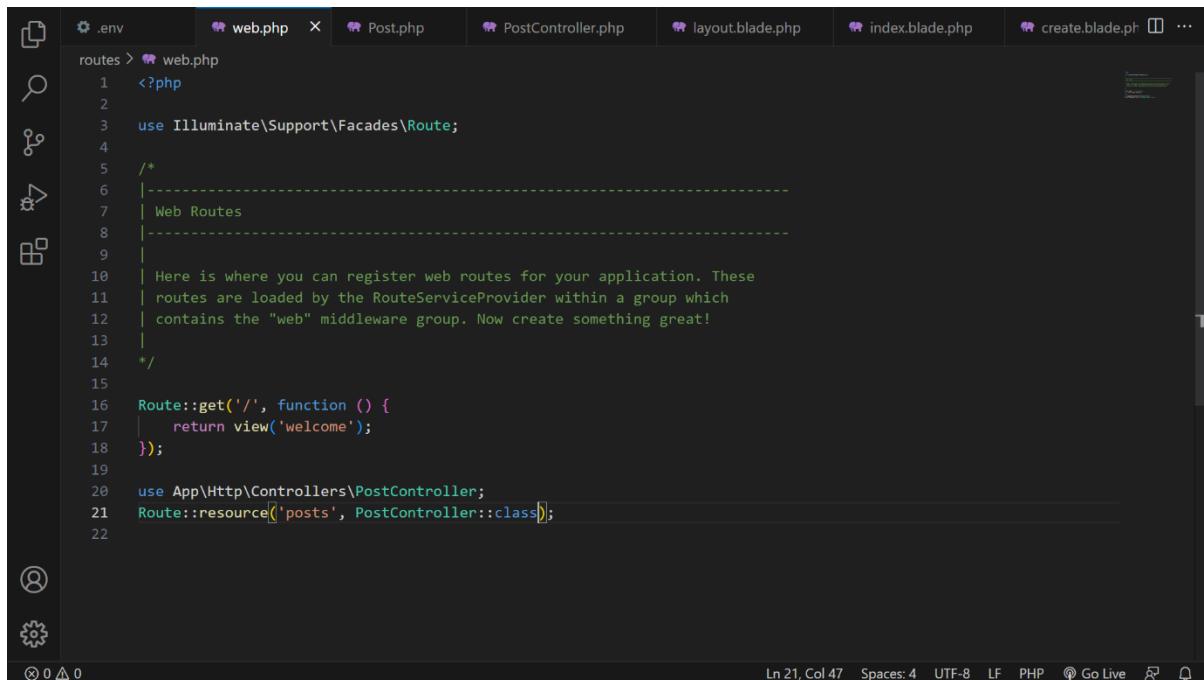
```
<?php
```

```
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application.
| These
|
| routes are loaded by the RouteServiceProvider within a group
| which
|
| contains the "web" middleware group. Now create something
| great!
|
*/
Route::get('/', function () {
    return view('welcome');
});

use App\Http\Controllers\PostController;
Route::resource('posts', PostController::class);
?>
```

Screenshot of code Web.php:



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  | -----
7  | Web Routes
8  | -----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 use App\Http\Controllers\PostController;
21 Route::resource(['posts', PostController::class]);
```

Ln 21, Col 47 Spaces: 4 UTF-8 LF PHP ⌂ Go Live ⌂

Step 5: Create Controllers for our Laravel CRUD Example

Now let's create our UserController just run the following command below:

```
php artisan make:controller UsersController
```

After running the command above you will find the generated controller to your **"App\Http\Controllers"**. Now let's add our methods for our Laravel crud operation example. Take note that we are using users to do this with a simple example.

In your **UsersController** you will have these methods:

- 1. index()**
- 2. create()**
- 3. store()**
- 4. show()**
- 5. edit()**

6. update()

7. destroy()

PostController.php

Code:

```
<?php

namespace App\Http\Controllers;

use App\Models\Post;

use Illuminate\Http\Request;

class PostController extends Controller

{

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */

    public function index()

    {

        $posts = Post::latest()->paginate(5);

        return view('posts.index',compact('posts'))->with('i',

        (request()->input('page', 1) - 1) * 5);

    }

}
```

```
/**  
 * Show the form for creating a new resource.  
 *  
 * @return \Illuminate\Http\Response  
 */  
  
public function create()  
{  
    return view('posts.create');  
}  
  
/**  
 * Store a newly created resource in storage.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @return \Illuminate\Http\Response  
 */  
  
public function store(Request $request)  
{  
    $request->validate([  
        'name' => 'required',  
        'detail' => 'required',  
    ]);  
}
```

```
    ]);

Post::create($request->all());

return redirect()->route('posts.index')->with('success','Post
created successfully.);

}

/**

 * Display the specified resource.

 *

 * @param \App\Models\Post $post

 * @return \Illuminate\Http\Response

 */

public function show(Post $post)

{

    return view('posts.show',compact('post'));

}

/**

 * Show the form for editing the specified resource.

 *

 * @param \App\Models\Post $post

 * @return \Illuminate\Http\Response


```

```
*/  
  
public function edit(Post $post)  
{  
    return view('posts.edit',compact('post'));  
}  
  
/**  
 * Update the specified resource in storage.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @param \App\Models\Post $post  
 * @return \Illuminate\Http\Response  
 */  
  
public function update(Request $request, Post $post)  
{  
    $request->validate([  
        'name' => 'required',  
        'detail' => 'required',  
    ]);  
    $post->update($request->all());  
}
```

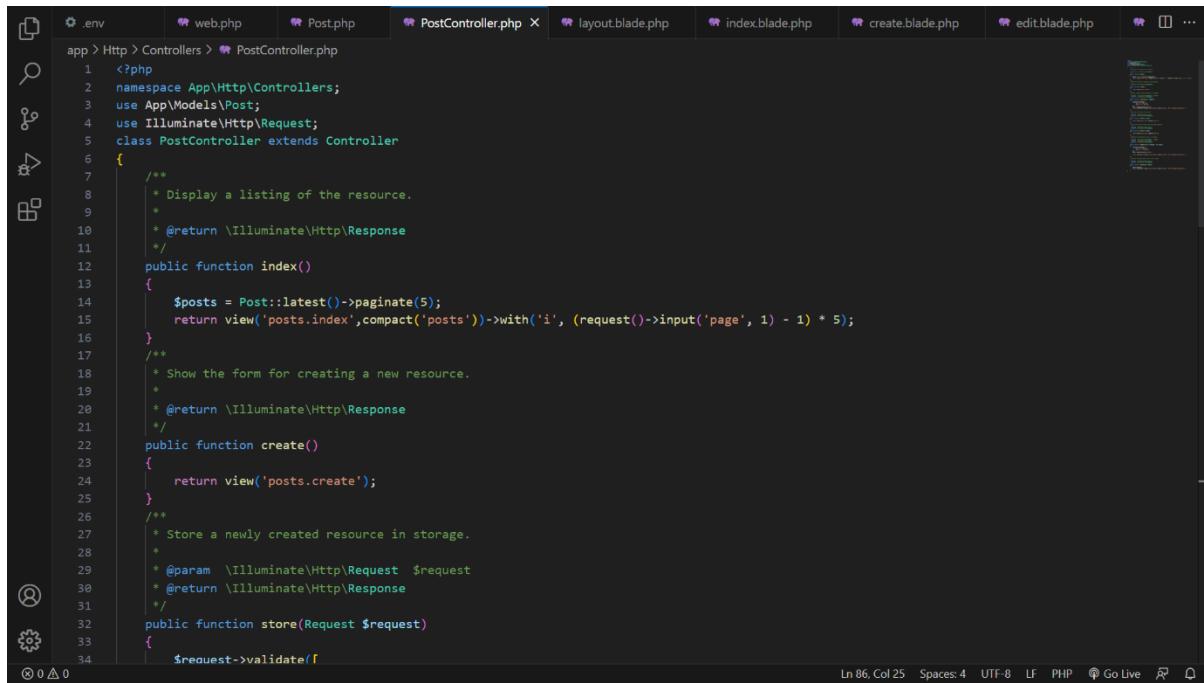
```
    return redirect()->route('posts.index')->with('success', 'Post
updated successfully');

}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Post $post
 * @return \Illuminate\Http\Response
 */
public function destroy(Post $post)
{
    $post->delete();

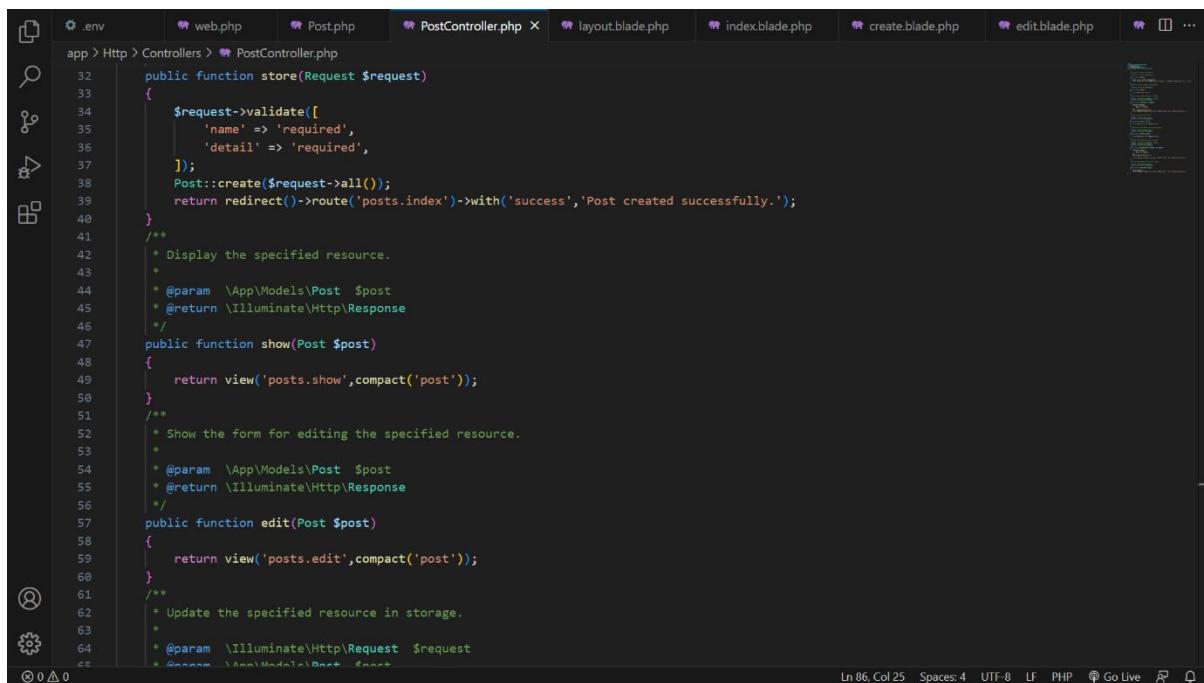
    return redirect()->route('posts.index')->with('success', 'Post
deleted successfully');
}
```

Screenshot of code:



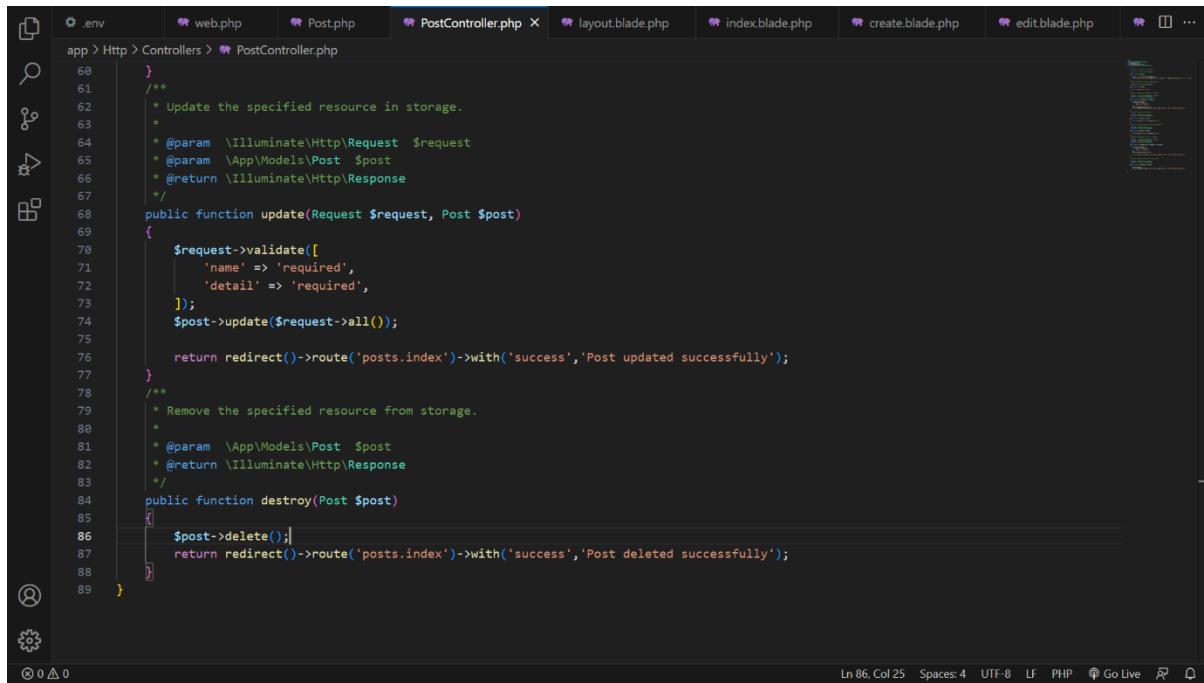
```
1  <?php
2  namespace App\Http\Controllers;
3  use App\Models\Post;
4  use Illuminate\Http\Request;
5  class PostController extends Controller
6  {
7      /**
8      * Display a listing of the resource.
9      *
10     * @return \Illuminate\Http\Response
11    */
12    public function index()
13    {
14        $posts = Post::latest()->paginate(5);
15        return view('posts.index',compact('posts'))->with('i', (request()->input('page', 1) - 1) * 5);
16    }
17    /**
18     * Show the form for creating a new resource.
19     *
20     * @return \Illuminate\Http\Response
21    */
22    public function create()
23    {
24        return view('posts.create');
25    }
26    /**
27     * Store a newly created resource in storage.
28     *
29     * @param \Illuminate\Http\Request $request
30     * @return \Illuminate\Http\Response
31     */
32    public function store(Request $request)
33    {
34        $request->validate([
35            'name' => 'required',
36            'detail' => 'required',
37        ]);
38        Post::create($request->all());
39        return redirect()->route('posts.index')->with('success','Post created successfully.');
40    }
41    /**
42     * Display the specified resource.
43     *
44     * @param \App\Models\Post $post
45     * @return \Illuminate\Http\Response
46     */
47    public function show(Post $post)
48    {
49        return view('posts.show',compact('post'));
50    }
51    /**
52     * Show the form for editing the specified resource.
53     *
54     * @param \App\Models\Post $post
55     * @return \Illuminate\Http\Response
56     */
57    public function edit(Post $post)
58    {
59        return view('posts.edit',compact('post'));
60    }
61    /**
62     * Update the specified resource in storage.
63     *
64     * @param \Illuminate\Http\Request $request
65     * @param \App\Models\Post $post
66     */
67    public function update(Request $request, Post $post)
68    {
69        $post->update($request->all());
70        return redirect()->route('posts.show', $post->id);
71    }
72}
```

Ln 86, Col 25 Spaces: 4 UTF-8 LF PHP ⚙ Go Live ⚙



```
32    public function store(Request $request)
33    {
34        $request->validate([
35            'name' => 'required',
36            'detail' => 'required',
37        ]);
38        Post::create($request->all());
39        return redirect()->route('posts.index')->with('success','Post created successfully.');
40    }
41    /**
42     * Display the specified resource.
43     *
44     * @param \App\Models\Post $post
45     * @return \Illuminate\Http\Response
46     */
47    public function show(Post $post)
48    {
49        return view('posts.show',compact('post'));
50    }
51    /**
52     * Show the form for editing the specified resource.
53     *
54     * @param \App\Models\Post $post
55     * @return \Illuminate\Http\Response
56     */
57    public function edit(Post $post)
58    {
59        return view('posts.edit',compact('post'));
60    }
61    /**
62     * Update the specified resource in storage.
63     *
64     * @param \Illuminate\Http\Request $request
65     * @param \App\Models\Post $post
66     */
67    public function update(Request $request, Post $post)
68    {
69        $post->update($request->all());
70        return redirect()->route('posts.show', $post->id);
71    }
72}
```

Ln 86, Col 25 Spaces: 4 UTF-8 LF PHP ⚙ Go Live ⚙



```
60     }
61     /**
62      * Update the specified resource in storage.
63      *
64      * @param \Illuminate\Http\Request $request
65      * @param \App\Models\Post $post
66      * @return \Illuminate\Http\Response
67      */
68     public function update(Request $request, Post $post)
69     {
70         $request->validate([
71             'name' => 'required',
72             'detail' => 'required',
73         ]);
74         $post->update($request->all());
75
76         return redirect()->route('posts.index')->with('success','Post updated successfully');
77     }
78     /**
79      * Remove the specified resource from storage.
80      *
81      * @param \App\Models\Post $post
82      * @return \Illuminate\Http\Response
83      */
84     public function destroy(Post $post)
85     {
86         $post->delete();
87         return redirect()->route('posts.index')->with('success','Post deleted successfully');
88     }
89 }
```

6. Setup Post Model

Post.php

Code:

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Post extends Model
```

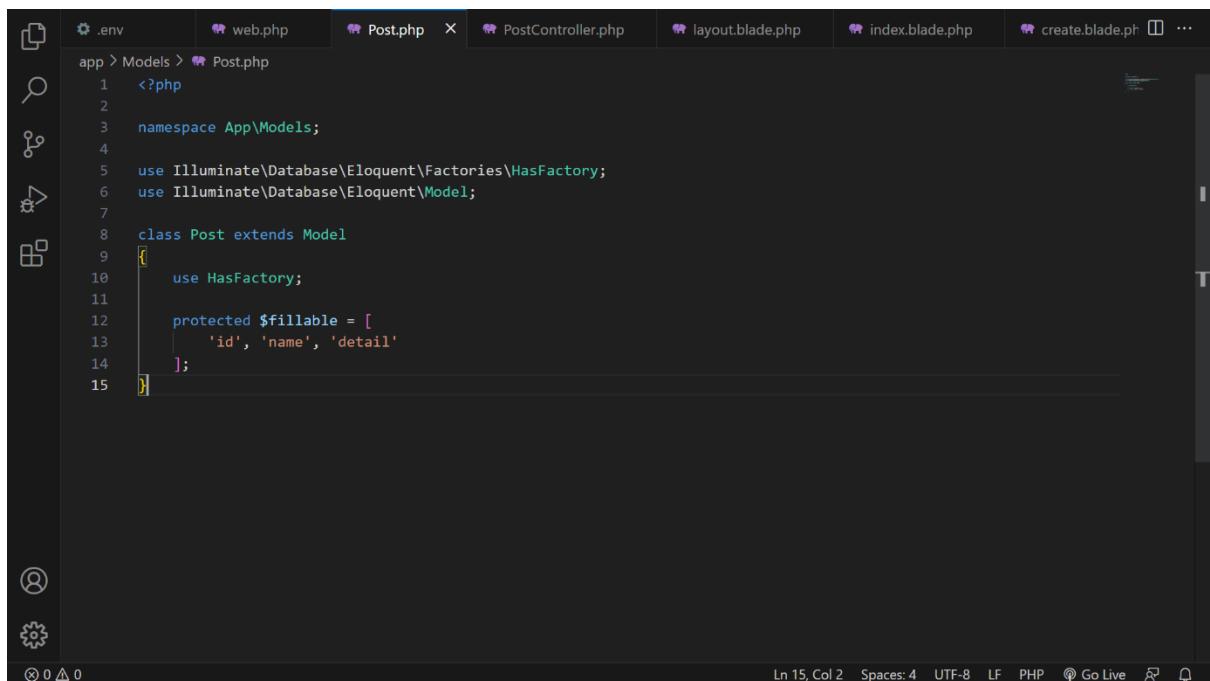
```
{
```

```
    use HasFactory;
```

```
    protected $fillable = [
```

```
'id', 'name', 'detail'  
];  
}
```

Screenshot of code:



```
app > Models > Post.php  
1  <?php  
2  
3  namespace App\Models;  
4  
5  use Illuminate\Database\Eloquent\Factories\HasFactory;  
6  use Illuminate\Database\Eloquent\Model;  
7  
8  class Post extends Model  
9  {  
10     use HasFactory;  
11  
12     protected $fillable = [  
13         'id', 'name', 'detail'  
14     ];  
15 }
```

Step 7: Add Blade Files for Our Laravel CRUD

In this last step, we will add our layouts and blades in order to display our Laravel CRUD examples.

Views/posts/create.blade.php

```
@extends('posts.layout')  
  
@section('content')  
<div class="row">  
    <div class="col-lg-12 margin-tb">  
        <div class="pull-left">  
            <h2>Add New Post</h2>
```

```
</div>

<div class="pull-right">

    <a class="btn btn-primary" href="{{ route('posts.index') }}">
Back</a>

</div>

</div>

</div>

@if ($errors->any())
<div class="alert alert-danger">
<strong>Error!</strong> <br>
<ul>
    @foreach ($errors->all() as $error)
        <li>{{ $error }}</li>
    @endforeach
</ul>
</div>
@endif

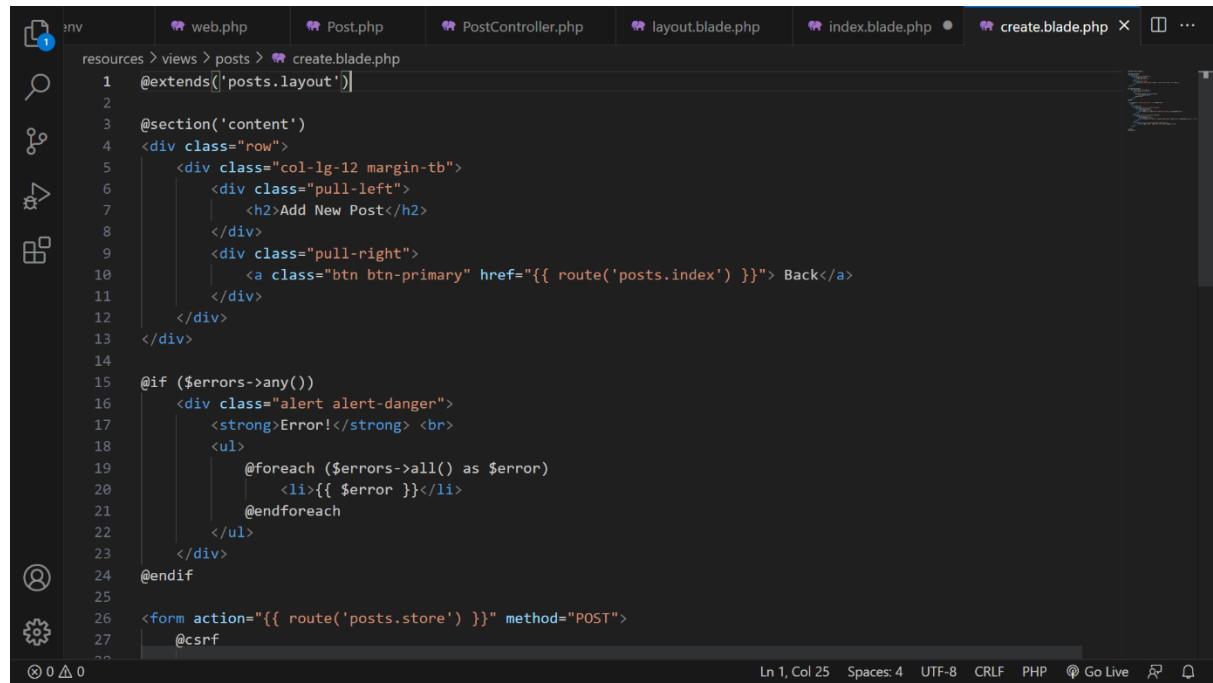
<form action="{{ route('posts.store') }}" method="POST">
@csrf

<div class="row">
<div class="col-xs-12 col-sm-12 col-md-12">
```

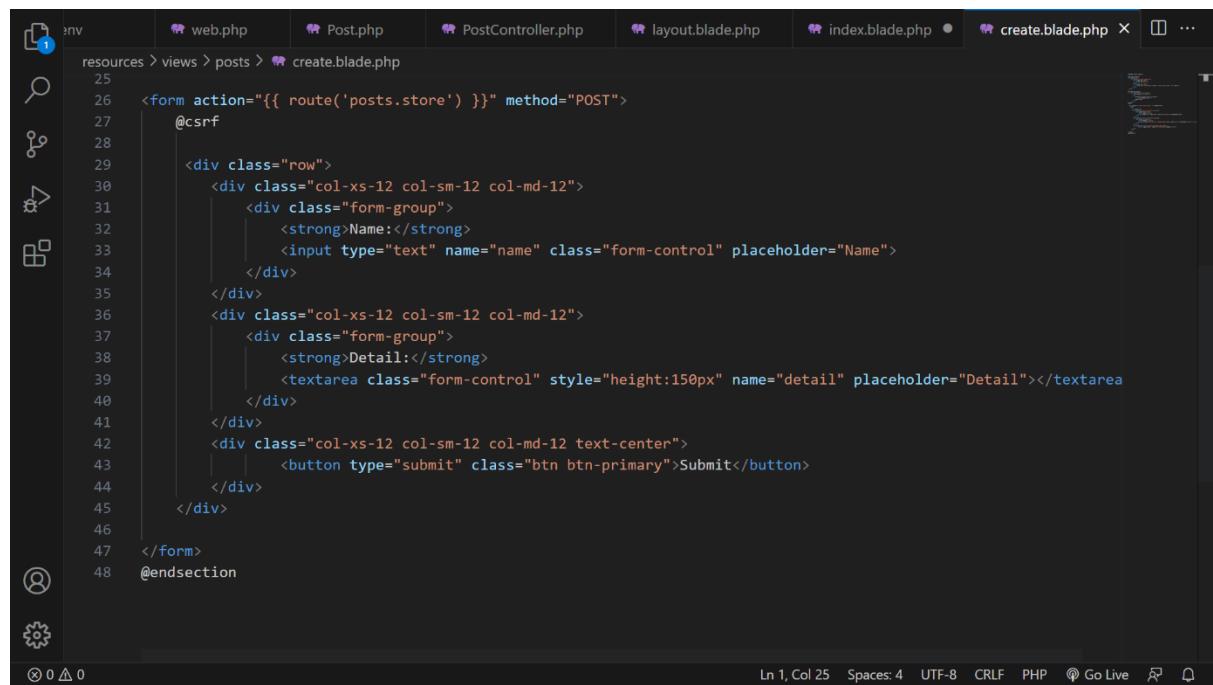
```
<div class="form-group">
    <strong>Name:</strong>
    <input type="text" name="name" class="form-control"
placeholder="Name">
</div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
        <strong>Detail:</strong>
        <textarea class="form-control" style="height:150px"
name="detail" placeholder="Detail"></textarea>
    </div>
</div>
<div class="col-xs-12 col-sm-12 col-md-12 text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</div>

</form>
@endsection
```

Screenshot of code:



```
resources > views > posts > create.blade.php
1  @extends(['posts.layout'])
2
3  @section('content')
4  <div class="row">
5      <div class="col-lg-12 margin-tb">
6          <div class="pull-left">
7              <h2>Add New Post</h2>
8          </div>
9          <div class="pull-right">
10             <a class="btn btn-primary" href="{{ route('posts.index') }}"> Back</a>
11         </div>
12     </div>
13  </div>
14
15  @if ($errors->any())
16      <div class="alert alert-danger">
17          <strong>Error!</strong> <br>
18          <ul>
19              @foreach ($errors->all() as $error)
20                  <li>{{ $error }}</li>
21              @endforeach
22          </ul>
23      </div>
24  @endif
25
26  <form action="{{ route('posts.store') }}" method="POST">
27      @csrf
28
29      <div class="row">
30          <div class="col-xs-12 col-sm-12 col-md-12">
31              <div class="form-group">
32                  <strong>Name:</strong>
33                  <input type="text" name="name" class="form-control" placeholder="Name">
34              </div>
35          </div>
36          <div class="col-xs-12 col-sm-12 col-md-12">
37              <div class="form-group">
38                  <strong>Detail:</strong>
39                  <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
40              </div>
41          </div>
42          <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43              <button type="submit" class="btn btn-primary">Submit</button>
44          </div>
45      </div>
46  </form>
47  @endsection
48
```



```
resources > views > posts > create.blade.php
25
26  <form action="{{ route('posts.store') }}" method="POST">
27      @csrf
28
29      <div class="row">
30          <div class="col-xs-12 col-sm-12 col-md-12">
31              <div class="form-group">
32                  <strong>Name:</strong>
33                  <input type="text" name="name" class="form-control" placeholder="Name">
34              </div>
35          </div>
36          <div class="col-xs-12 col-sm-12 col-md-12">
37              <div class="form-group">
38                  <strong>Detail:</strong>
39                  <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
40              </div>
41          </div>
42          <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43              <button type="submit" class="btn btn-primary">Submit</button>
44          </div>
45      </div>
46
47  @endsection
48
```

Views/posts/edit.blade.php

```
@extends('posts.layout')
```

```
@section('content')
```

```
<div class="row">
```

```
    <div class="col-lg-12 margin-tb">
```

```
<div class="pull-left">
  <h2>Edit Post</h2>
</div>
<div class="pull-right">
  <a class="btn btn-primary" href="{{ route('posts.index') }}">
    Back</a>
</div>
</div>

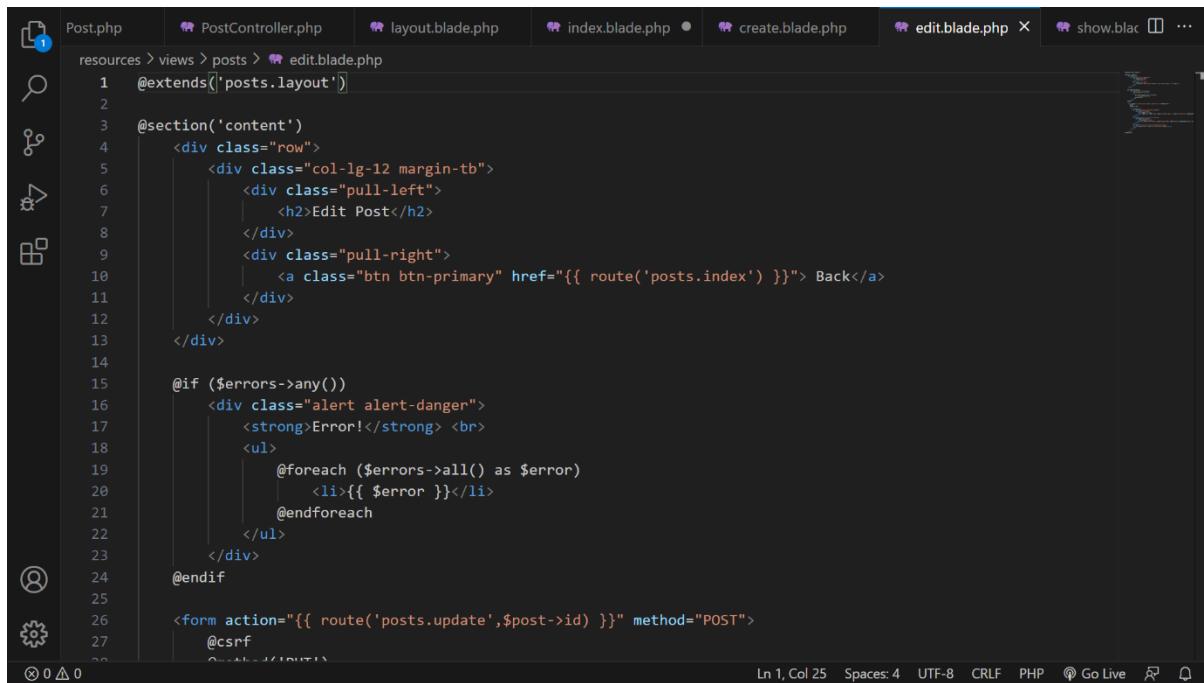
@if ($errors->any())
  <div class="alert alert-danger">
    <strong>Error!</strong> <br>
    <ul>
      @foreach ($errors->all() as $error)
        <li>{{ $error }}</li>
      @endforeach
    </ul>
  </div>
@endif

<form action="{{ route('posts.update',$post->id) }}"
method="POST">
  @csrf
```

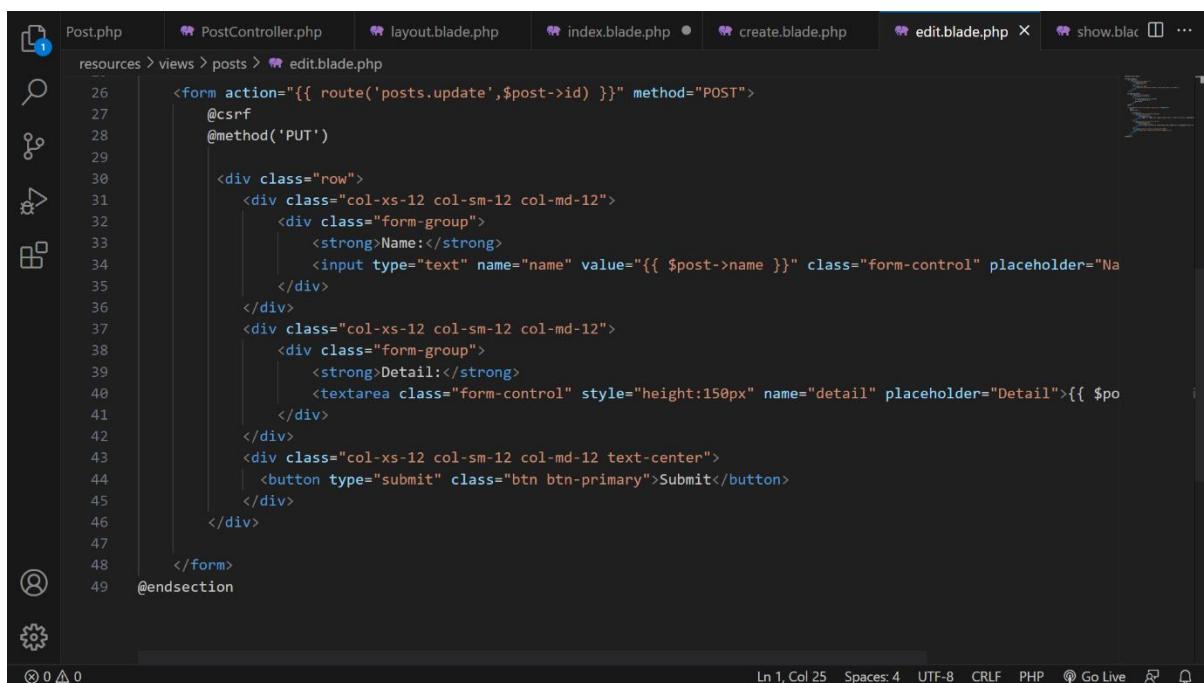
```
@method('PUT')

<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            <input type="text" name="name" value="{{ $post->name
}}>" class="form-control" placeholder="Name">
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Detail:</strong>
            <textarea class="form-control" style="height:150px"
name="detail" placeholder="Detail">{{ $post->detail }}</textarea>
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
        <button type="submit" class="btn btn-
primary">Submit</button>
    </div>
</div>
</form>
@endsection
```

Screenshot of code:



```
Post.php PostController.php layout.blade.php index.blade.php create.blade.php edit.blade.php show.blade.php ...
resources > views > posts > edit.blade.php
1 @extends('posts.layout')
2
3 @section('content')
4     <div class="row">
5         <div class="col-lg-12 margin-tb">
6             <div class="pull-left">
7                 <h2>Edit Post</h2>
8             </div>
9             <div class="pull-right">
10                 <a class="btn btn-primary" href="{{ route('posts.index') }}"> Back </a>
11             </div>
12         </div>
13     </div>
14
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Error!</strong> <br>
18             <ul>
19                 @foreach ($errors->all() as $error)
20                     <li>{{ $error }}</li>
21                 @endforeach
22             </ul>
23         </div>
24     @endif
25
26     <form action="{{ route('posts.update', $post->id) }}" method="POST">
27         @csrf
28         ...
29     </form>
30
31     @endif
32
33     @endif
34
35     @endif
36
37     @endif
38
39     @endif
40
41     @endif
42
43     @endif
44
45     @endif
46
47     @endif
48
49     @endsection
```



```
Post.php PostController.php layout.blade.php index.blade.php create.blade.php edit.blade.php show.blade.php ...
resources > views > posts > index.blade.php
1 <div class="row">
2     <div class="col-xs-12 col-sm-12 col-md-12">
3         <div class="form-group">
4             <strong>Name:</strong>
5             <input type="text" name="name" value="{{ $post->name }}" class="form-control" placeholder="Name" required="required" data-validation="required" data-validation-error="Name is required." data-validation-delay="0" data-validation-type="text">
6         </div>
7         <div class="form-group">
8             <strong>Detail:</strong>
9             <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail">{{ $post->detail }}
10        </div>
11    </div>
12    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
13        <button type="submit" class="btn btn-primary">Submit</button>
14    </div>
15
16 </div>
17
18 </div>
19
20 </div>
21
22 </div>
23
24 </div>
25
26 </div>
27
28 </div>
29
30 </div>
31
32 </div>
33
34 </div>
35
36 </div>
37
38 </div>
39
40 </div>
41
42 </div>
43
44 </div>
45
46 </div>
47
48 </div>
49
50 </div>
51
52 </div>
53
54 </div>
55
56 </div>
57
58 </div>
59
60 </div>
61
62 </div>
63
64 </div>
65
66 </div>
67
68 </div>
69
70 </div>
71
72 </div>
73
74 </div>
75
76 </div>
77
78 </div>
79
80 </div>
81
82 </div>
83
84 </div>
85
86 </div>
87
88 </div>
89
90 </div>
91
92 </div>
93
94 </div>
95
96 </div>
97
98 </div>
99
100 </div>
101
102 </div>
103
104 </div>
105
106 </div>
107
108 </div>
109
110 </div>
111
112 </div>
113
114 </div>
115
116 </div>
117
118 </div>
119
120 </div>
121
122 </div>
123
124 </div>
125
126 </div>
127
128 </div>
129
130 </div>
131
132 </div>
133
134 </div>
135
136 </div>
137
138 </div>
139
140 </div>
141
142 </div>
143
144 </div>
145
146 </div>
147
148 </div>
149
150 </div>
151
152 </div>
153
154 </div>
155
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
8010 </div>
8011
8012 </div>
8013
8014 </div>
8015
8016 </div>
8017
8018 </div>
8019
8020 </div>
8021
8022 </div>
8023
8024 </div>
8025
8026 </div>
8027
8028 </div>
8029
8030 </div>
8031
8032 </div>
8033
8034 </div>
8035
8036 </div>
8037
8038 </div>
8039
8040 </div>
8041
8042 </div>
8043
8044 </div>
8045
8046 </div>
8047
8048 </div>
8049
8050 </div>
8051
8052 </div>
8053
8054 </div>
8055
8056 </div>
8057
8058 </div>
8059
8060 </div>
8061
8062 </div>
8063
8064 </div>
8065
8066 </div>
8067
8068 </div>
8069
8070 </div>
8071
8072 </div>
8073
8074 </div>
8075
8076 </div>
8077
8078 </div>
8079
8080 </div>
8081
8082 </div>
8083
8084 </div>
8085
8086 </div>
8087
8088 </div>
8089
8090 </div>
8091
8092 </div>
8093
8094 </div>
8095
8096 </div>
8097
8098 </div>
8099
80100 </div>
80101
80102 </div>
80103
80104 </div>
80105
80106 </div>
80107
80108 </div>
80109
80110 </div>
80111
80112 </div>
80113
80114 </div>
80115
80116 </div>
80117
80118 </div>
80119
80120 </div>
80121
80122 </div>
80123
80124 </div>
80125
80126 </div>
80127
80128 </div>
80129
80130 </div>
80131
80132 </div>
80133
80134 </div>
80135
80136 </div>
80137
80138 </div>
80139
80140 </div>
80141
80142 </div>
80143
80144 </div>
80145
80146 </div>
80147
80148 </div>
80149
80150 </div>
80151
80152 </div>
80153
80154 </div>
80155
80156 </div>
80157
80158 </div>
80159
80160 </div>
80161
80162 </div>
80163
80164 </div>
80165
80166 </div>
80167
80168 </div>
80169
80170 </div>
80171
80172 </div>
80173
80174 </div>
80175
80176 </div>
80177
80178 </div>
80179
80180 </div>
80181
80182 </div>
80183
80184 </div>
80185
80186 </div>
80187
80188 </div>
80189
80190 </div>
80191
80192 </div>
80193
80194 </div>
80195
80196 </div>
80197
80198 </div>
80199
80200 </div>
80201
80202 </div>
80203
80204 </div>
80205
80206 </div>
80207
80208 </div>
80209
80210 </div>
80211
80212 </div>
80213
80214 </div>
80215
80216 </div>
80217
80218 </div>
80219
80220 </div>
80221
80222 </div>
80223
80224 </div>
80225
80226 </div>
80227
80228 </div>
80229
80230 </div>
80231
80232 </div>
80233
80234 </div>
80235
80236 </div>
80237
80238 </div>
80239
80240 </div>
80241
80242 </div>
80243
80244 </div>
80245
80246 </div>
80247
80248 </div>
80249
80250 </div>
80251
80252 </div>
80253
80254 </div>
80255
80256 </div>
80257
80258 </div>
80259
80260 </div>
80261
80262 </div>
80263
80264 </div>
80265
80266 </div>
80267
80268 </div>
80269
80270 </div>
80271
80272 </div>
80273
80274 </div>
80275
80276 </div>
80277
80278 </div>
80279
80280 </div>
80281
80282 </div>
80283
80284 </div>
80285
80286 </div>
80287
80288 </div>
80289
80290 </div>
80291
80292 </div>
80293
80294 </div>
80295
80296 </div>
80297
80298 </div>
80299
80300 </div>
80301
80302 </div>
80303
80304 </div>
80305
80306 </div>
80307
80308 </div>
80309
80310 </div>
80311
80312 </div>
80313
80314 </div>
80315
80316 </div>
80317
80318 </div>
80319
80320 </div>
80321
80322 </div>
80323
80324 </div>
80325
80326 </div>
80327
80328 </div>
80329
80330 </div>
80331
80332 </div>
80333
80334 </div>
80335
80336 </div>
80337
80338 </div>
80339
80340 </div>
80341
80342 </div>
80343
80344 </div>
80345
80346 </div>
80347
80348 </div>
80349
80350 </div>
80351
80352 </div>
80353
80354 </div>
80355
80356 </div>
80357
80358 </div>
80359
80360 </div>
80361
80362 </div>
80363
80364 </div>
80365
80366 </div>
80367
80368 </div>
80369
80370 </div>
80371
80372 </div>
80373
80374 </div>
80375
80376 </div>
80377
80378 </div>
80379
80380 </div>
80381
80382 </div>
80383
80384 </div>
80385
80386 </div>
80387
80388 </div>
80389
80390 </div>
80391
80392 </div>
80393
80394 </div>
80395
80396 </div>
80397
80398 </div>
80399
80400 </div>
80401
80402 </div>
80403
80404 </div>
80405
80406 </div>
80407
80408 </div>
80409
80410 </div>
80411
80412 </div>
80413
80414 </div>
80415
80416 </div>
80417
80418 </div>
80419
80420 </div>
80421
80422 </div>
80423
80424 </div>
80425
80426 </div>
80427
80428 </div>
80429
80430 </div>
80431
80432 </div>
80433
80434 </div>
80435
80436 </div>
80437
80438 </div>
80439
80440 </div>
80441
80442 </div>
80443
80444 </div>
80445
80446 </div>
80447
80448 </div>
80449
80450 </div>
80451
80452 </div>
80453
80454 </div>
80455
80456 </div>
80457
80458 </div>
80459
80460 </div>
80461
80462 </div>
80463
80464 </div>
80465
80466 </div>
80467
80468 </div>
80469
80470 </div>
80471
80472 </div>
80473
80474 </div>
80475
80476 </div>
80477
80478 </div>
80479
80480 </div>
80481
80482 </div>
80483
80484 </div>
80485
80486 </div>
80487
80488 </div>
80489
80490 </div>
80491
80492 </div>
80493
80494 </div>
80495
80496 </div>
80497
80498 </div>
80499
80500 </div>
80501
80502 </div>
80503
80504 </div>
80505
80506 </div>
80507
80508 </div>
80509
80510 </div>
80511
80512 </div>
80513
80514 </div>
80515
80516 </div>
80517
80518 </div>
80519
80520 </div>
80521
80522 </div>
80523
80524 </div>
80525
80526 </div>
80527
80528 </div>
80529
80530 </div>
80531
80532 </div>
80533
80534 </div>
80535
80536 </div>
80537
80538 </div>
80539
80540 </div>
80541
80542 </div>
80543
80544 </div>
80545
80546 </div>
80547
80548 </div>
80549
80550 </div>
80551
80552 </div>
80553
80554 </div>
80555
80556 </div>
80557
80558 </div>
80559
80560 </div>
80561
80562 </div>
80563
80564 </div>
80565
80566 </div>
80567
80568 </div>
80569
80570 </div>
80571
80572 </div>
80573
80574 </div>
80575
80576 </div>
80577
80578 </div>
80579
80580 </div>
80581
80582 </div>
80583
80584 </div>
80585
80586 </div>
80587
80588 </div>
80589
80590 </div>
80591
80592 </div>
80593
80594 </div>
80595
80596 </div>
80597
80598 </div>
80599
80600 </div>
80601
80602 </div>
80603
80604 </div>
80605
80606 </div>
80607
80608 </div>
80609
80610 </div>
80611
80612 </div>
80613
80
```

```
<div class="row">
  <div class="col-lg-12 margin-tb">
    <div class="pull-left">
      <h1>CRUD Operation In Laravel</h1>
    </div>
    <div class="pull-right">
      <a class="btn btn-success" href="{{ route('posts.create') }}">
        Create New Post</a>
    </div>
  </div>
</div>

@if ($message = Session::get('success'))
  <div class="alert alert-success">
    <p>{{ $message }}</p>
  </div>
@endif

<table class="table table-bordered">
  <tr>
    <th>No</th>
    <th>Name</th>
    <th>Details</th>
    <th width="280px">Action</th>
  
```

```
</tr>

@foreach ($posts as $post)

<tr>

    <td>{{ ++$i }}</td>

    <td>{{ $post->name }}</td>

    <td>{{ $post->detail }}</td>

    <td>

        <form action="{{ route('posts.destroy',$post->id) }}"
method="POST">

            <a class="btn btn-info" href="{{ route('posts.show',$post->id) }}>Show</a>

            <a class="btn btn-primary" href="{{ route('posts.edit',$post->id) }}>Edit</a>

            @csrf

            @method('DELETE')

        <button type="submit" class="btn btn-danger">Delete</button>

    </form>

    </td>

</tr>

@endforeach
```

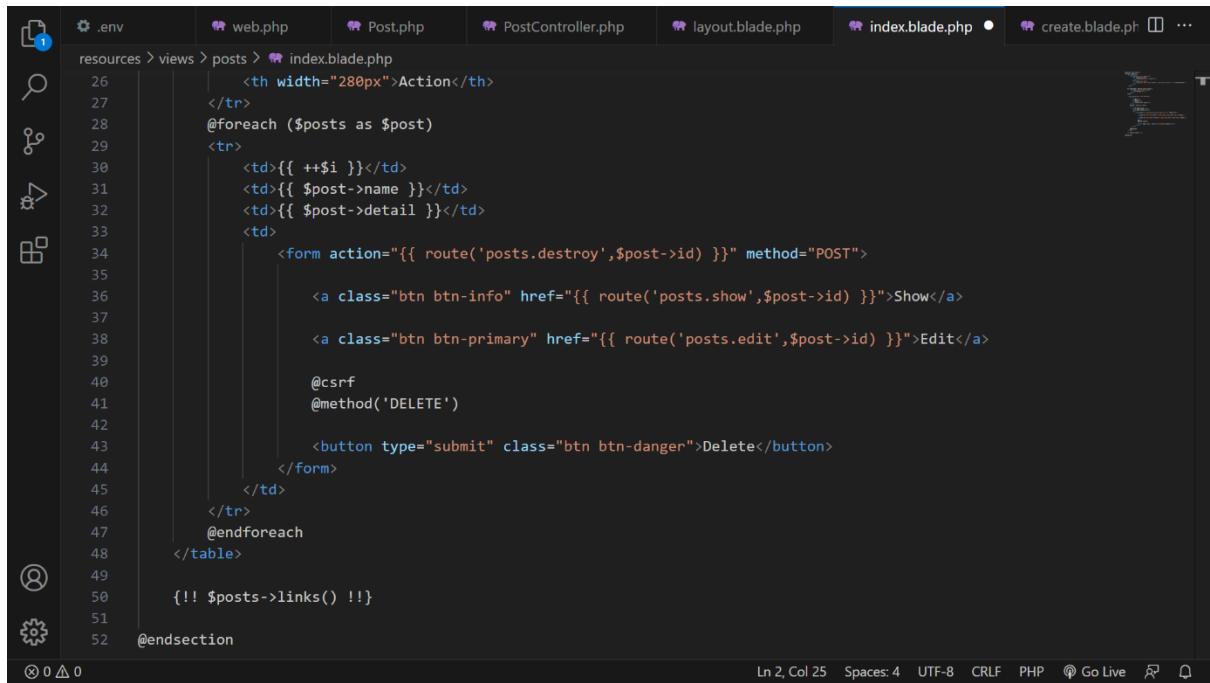
</table>

{!! \$posts->links() !!}

@endsection

Screenshot of code:

```
resources > views > posts > index.blade.php
  1
  2  @extends('posts.layout')
  3  @section('content')
  4      <div class="row">
  5          <div class="col-lg-12 margin-tb">
  6              <div class="pull-left">
  7                  <h1>CRUD Operation In Laravel</h1>
  8              </div>
  9              <div class="pull-right">
 10                  <a class="btn btn-success" href="{{ route('posts.create') }}> Create New Post</a>
 11              </div>
 12          </div>
 13      </div>
 14
 15      @if ($message = Session::get('success'))
 16          <div class="alert alert-success">
 17              <p>{{ $message }}</p>
 18          </div>
 19      @endif
 20
 21      <table class="table table-bordered">
 22          <tr>
 23              <th>No</th>
 24              <th>Name</th>
 25              <th>Details</th>
 26              <th width="280px">Action</th>
 27          </tr>
 28      </table>
```



```
resources > views > posts > index.blade.php
26     <th width="280px">Action</th>
27     </tr>
28     @foreach ($posts as $post)
29     <tr>
30         <td>{{ ++$i }}</td>
31         <td>{{ $post->name }}</td>
32         <td>{{ $post->detail }}</td>
33         <td>
34             <form action="{{ route('posts.destroy', $post->id) }}" method="POST">
35                 <a class="btn btn-info" href="{{ route('posts.show', $post->id) }}>Show</a>
36                 <a class="btn btn-primary" href="{{ route('posts.edit', $post->id) }}>Edit</a>
37                 @csrf
38                 @method('DELETE')
39                 <button type="submit" class="btn btn-danger">Delete</button>
40             </form>
41         </td>
42     </tr>
43     @endforeach
44     </table>
45
46     {!! $posts->links() !!}
47
48     @endsection
49
50
51
52
```

Views/posts/layout.blade.php

```
<!DOCTYPE html>

<html>

<head>

    <title>Laravel 9 CRUD Operation Step By Step - Techsolutionstuff</title>

    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">

</head>

<body>

<div class="container" style="margin-top: 15px;">

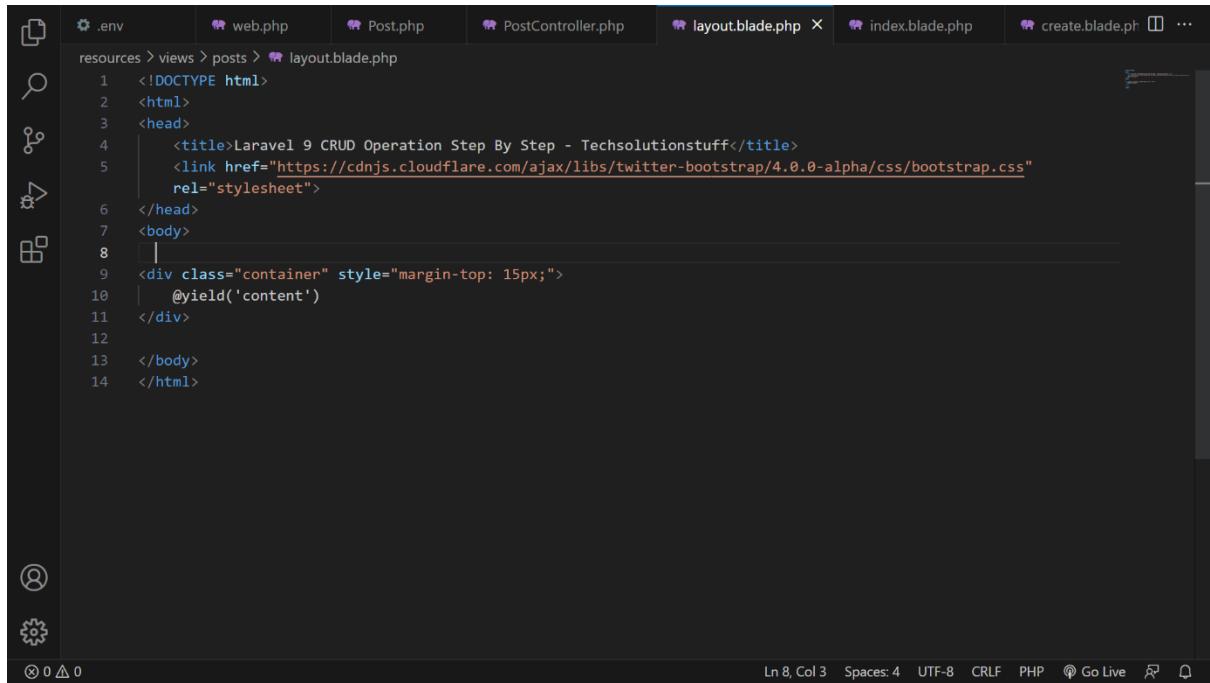
    @yield('content')

</div>
```

```
</body>
```

```
</html>
```

Screenshot of code:



```
resources > views > posts > layout.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Laravel 9 CRUD Operation Step By Step - Techsolutionstuff</title>
5  |   <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">
6  </head>
7  <body>
8  |
9  <div class="container" style="margin-top: 15px;">
10 |   @yield('content')
11 </div>
12
13 </body>
14 </html>
```

Views/posts/show.blade.php

```
@extends('posts.layout')
```

```
@section('content')
```

```
 <div class="row">
```

```
   <div class="col-lg-12 margin-tb">
```

```
     <div class="pull-left">
```

```
       <h2> Show Post</h2>
```

```
     </div>
```

```
     <div class="pull-right">
```

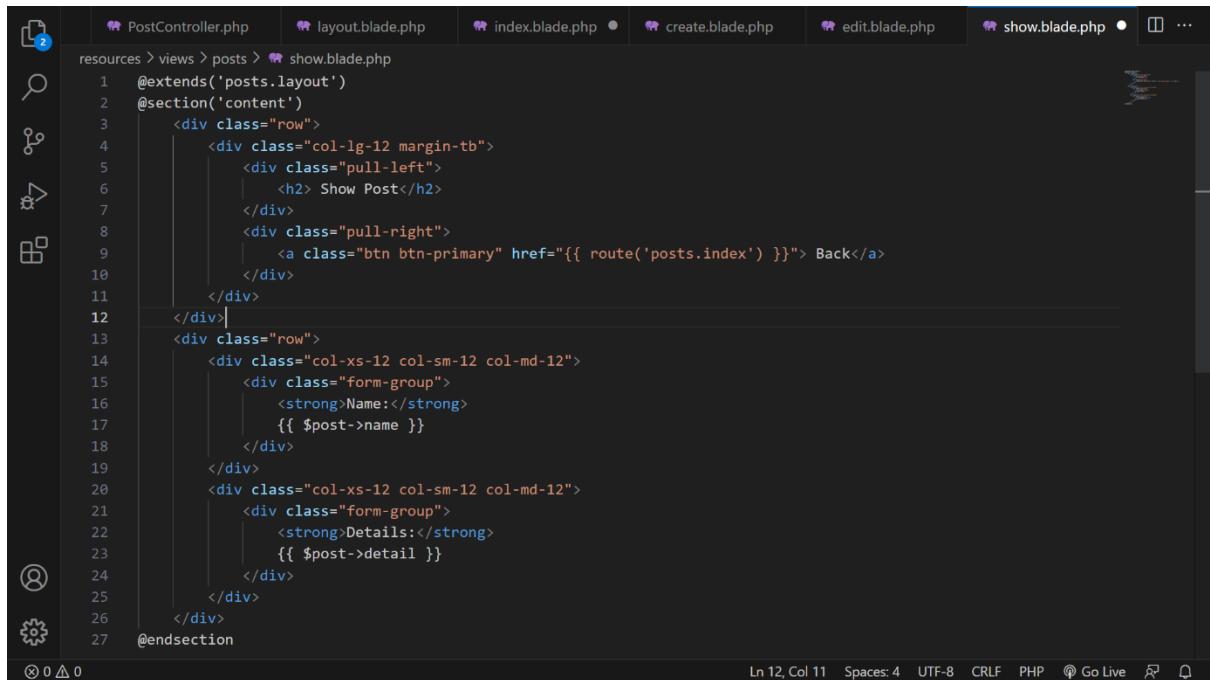
```
       <a class="btn btn-primary" href="{{ route('posts.index') }}>
```

```
Back</a>
```

```
     </div>
```

```
</div>
</div>
<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Name:</strong>
      {{ $post->name }}
    </div>
  </div>
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Details:</strong>
      {{ $post->detail }}
    </div>
  </div>
</div>
@endsection
```

Screenshot of code:



```

resources > views > posts > show.blade.php
1  @extends('posts.layout')
2  @section('content')
3  <div class="row">
4      <div class="col-lg-12 margin-tb">
5          <div class="pull-left">
6              <h2> Show Post</h2>
7          </div>
8          <div class="pull-right">
9              <a class="btn btn-primary" href="{{ route('posts.index') }}> Back</a>
10         </div>
11     </div>
12  </div>
13  <div class="row">
14      <div class="col-xs-12 col-sm-12 col-md-12">
15          <div class="form-group">
16              <strong>Name:</strong>
17              {{ $post->name }}
18          </div>
19      </div>
20      <div class="col-xs-12 col-sm-12 col-md-12">
21          <div class="form-group">
22              <strong>Details:</strong>
23              {{ $post->detail }}
24          </div>
25      </div>
26  </div>
27  @endsection

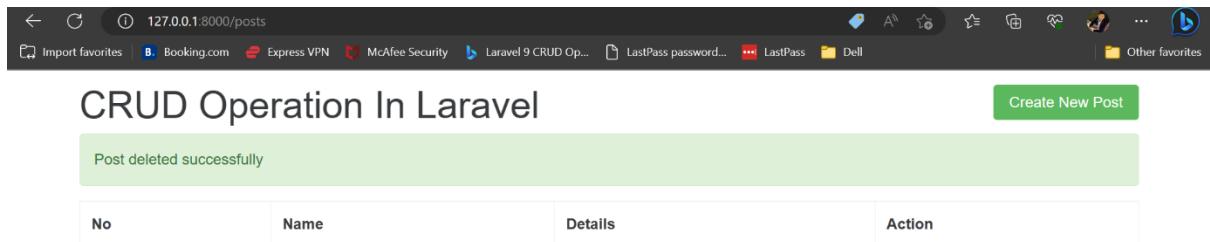
```

Ln 12, Col 11 Spaces: 4 UTF-8 CRLF PHP ⌂ Go Live ⌂ ⌂

127.0.0.1:8000/posts

Screenshot

Output:



127.0.0.1:8000/posts

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

CRUD Operation In Laravel

Create New Post

Post deleted successfully

No	Name	Details	Action
----	------	---------	--------

Create New Post

← ⏪ ⓘ 127.0.0.1:8000/posts/create

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

Add New Post

Name:
Rakesh Kumar Mandal

Detail:
Sitamarhi, Bihar, 843330

Submit

Created

← ⏪ ⓘ 127.0.0.1:8000/posts

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

CRUD Operation In Laravel

Create New Post

Post created successfully.

No	Name	Details	Action
1	Rakesh Kumar Mandal	Sitamarhi, Bihar, 843330	Show Edit Delete

Show in database

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=adit1&table=posts

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... phpMyAdmin

Server: 127.0.0.1 » Database: adit1 » Table: posts

Browse Structure SQL Search Insert Export

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

SELECT * FROM `posts`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

Extra options

	id	name	detail
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Rakesh Kumar Mandal	Sitamarhi, Bihar, 843330

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

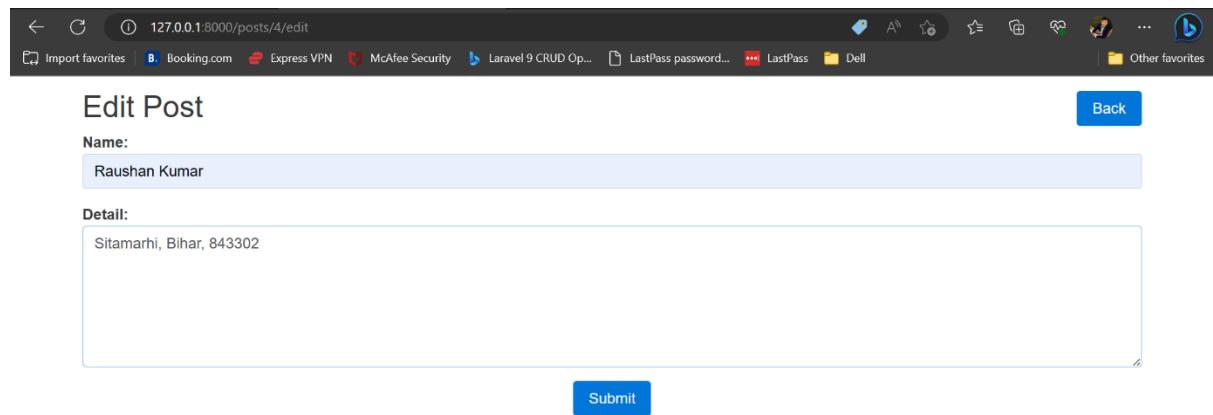
Print Copy to clipboard Export Display chart Create view

Console mark this SQL query

New adit adit1 New failed_jobs migrations password_resets personal_access_tokens posts users company ibm information_schema laravel mysql nsti1 performance_schema phpmyadmin student test

Show Post

Edit Post



← ⌂ ⓘ 127.0.0.1:8000/posts/4/edit

Import favorites | B Booking.com | Express VPN | McAfee Security | Laravel 9 CRUD Op... | LastPass password... | LastPass | Dell | Other favorites

Back

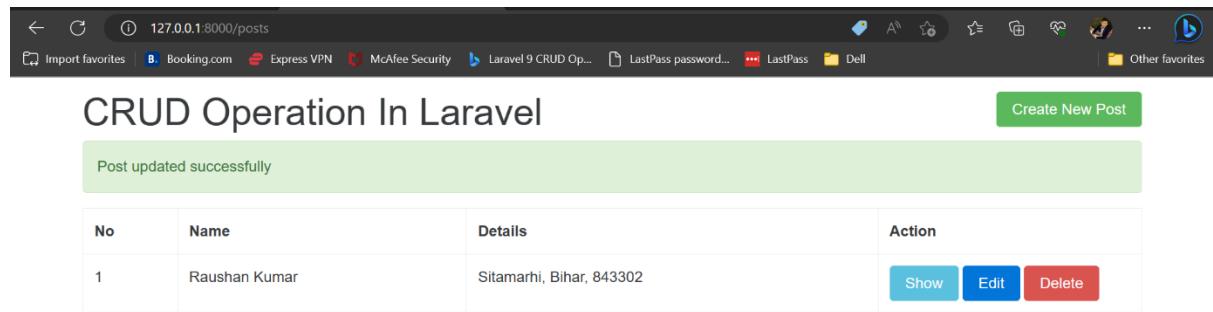
Edit Post

Name:

Detail:

Submit

Delete



← ⌂ ⓘ 127.0.0.1:8000/posts

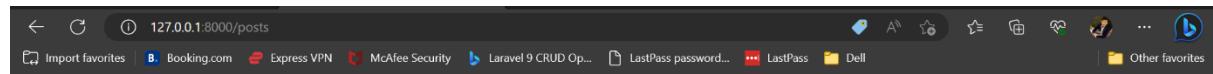
Import favorites | B Booking.com | Express VPN | McAfee Security | Laravel 9 CRUD Op... | LastPass password... | LastPass | Dell | Other favorites

Create New Post

Post updated successfully

No	Name	Details	Action
1	Raushan Kumar	Sitamarhi, Bihar, 843302	Show Edit Delete

Deleted data



127.0.0.1:8000/posts

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

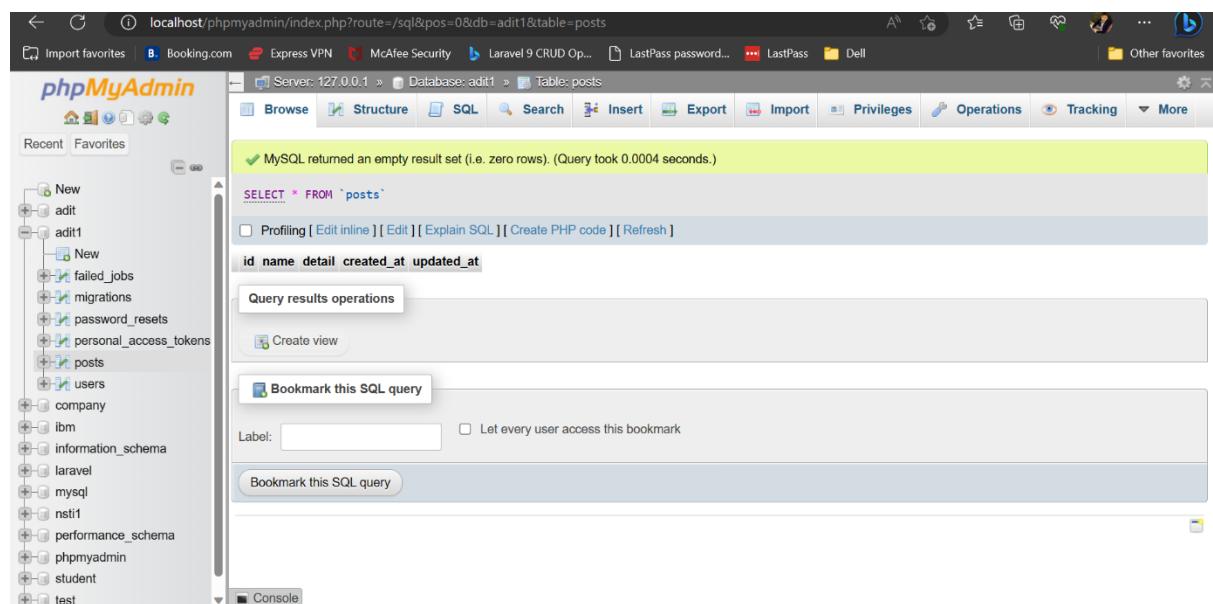
CRUD Operation In Laravel

Create New Post

Post deleted successfully

No	Name	Details	Action

Show in database



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=adit1&table=posts

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

phpMyAdmin

Server: 127.0.0.1 » Database: adit1 » Table: posts

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking More

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
SELECT * FROM `posts`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	id	name	detail	created_at	updated_at

Query results operations

Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

Console

7 **Aim:** - Create a Model and Demonstrate all CRUD operations.

Learning outcome: - Able to understand CRUD operation use of model.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:-

Open your Command prompt and Please enter a Laravel command

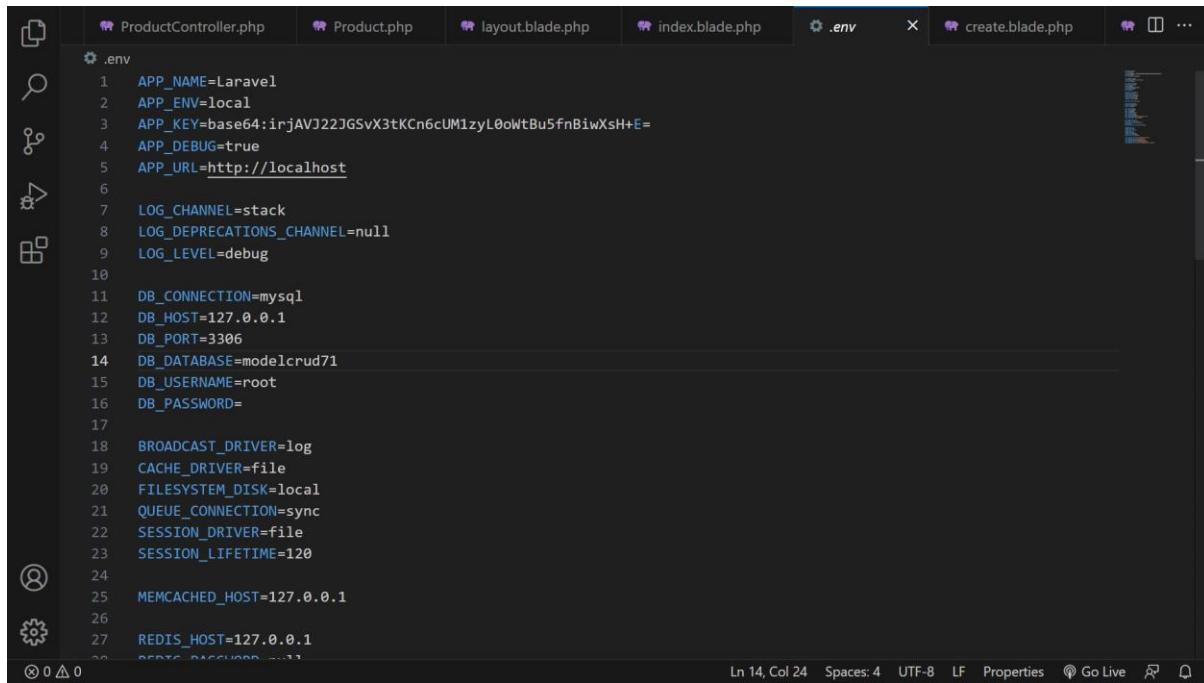
```
composer create-project --prefer-dist
laravel/Laravel modelcrud71
```

Step 2: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:irjAVJ22JG5vX3tKCn6cUM1zyL0oWtBu5fnBiwXsH+E=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=modelcrud71
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=
```

Step 3: Migration Setup

Let's create a migration for our Laravel crud operation. If you are using a fresh Laravel project we need to customize the user's table because I added the username field in my authentication. Here are the complete migrations below:

Code:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
    
```

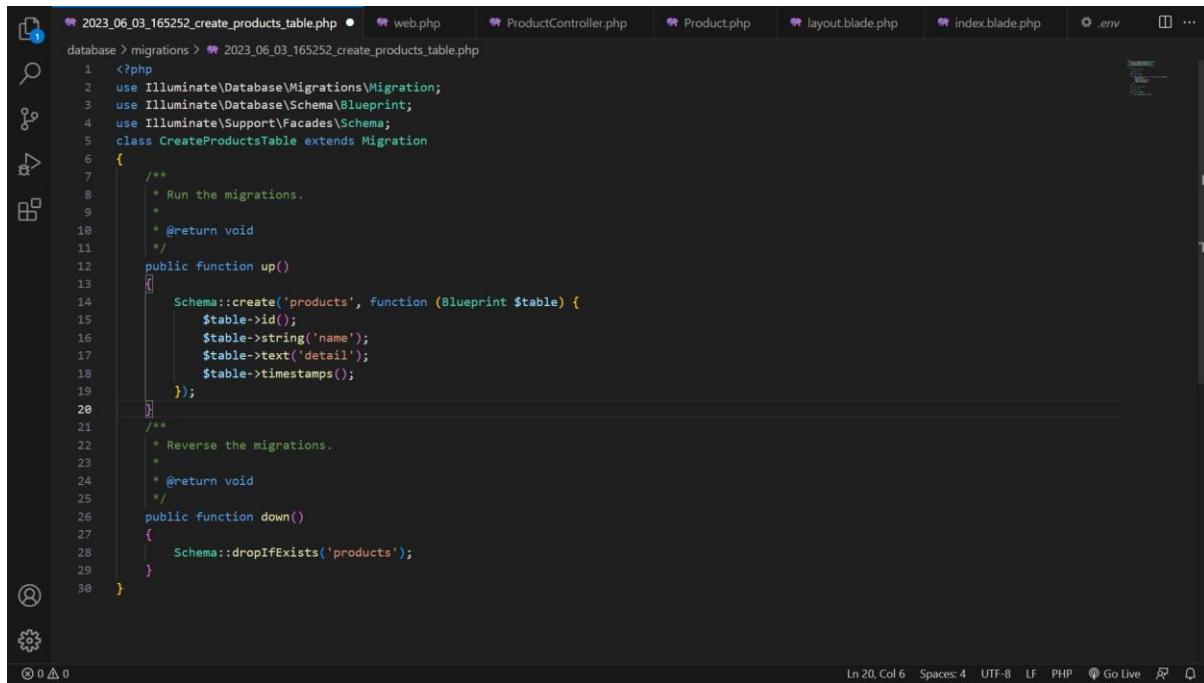
```

*
* @return void
*/
public function up()
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->text('detail');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
*/
public function down()
{
    Schema::dropIfExists('products');
}
}

```

Screenshot of code:



```
2023_06_03_165252_create_products_table.php ● 2023_06_03_165252_create_products_table.php
database > migrations > 2023_06_03_165252_create_products_table.php
1  <?php
2  use Illuminate\Database\Migrations\Migration;
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Support\Facades\Schema;
5  class CreateProductsTable extends Migration
6  {
7      /**
8      * Run the migrations.
9      *
10     * @return void
11    */
12    public function up()
13    {
14        Schema::create('products', function (Blueprint $table) {
15            $table->id();
16            $table->string('name');
17            $table->text('detail');
18            $table->timestamps();
19        });
20    }
21    /**
22     * Reverse the migrations.
23     *
24     * @return void
25    */
26    public function down()
27    {
28        Schema::dropIfExists('products');
29    }
30 }

```

Ln 20, Col 6 Spaces: 4 UTF-8 LF PHP Go Live ⌂ ⌂

Now let's run the following command below:

```
php artisan migrate
```

Step 4: Operation Routes Add Resource Route

In my example, I will create manually my CRUD routes. Just open the **"routes/web.php"** file and add the following routes.

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\ProductController;
```

```
/*
```

```
|-----
```

| Web Routes

|-----

|

| Here is where you can register web routes for your application.
These

| routes are loaded by the RouteServiceProvider within a group
which

| contains the "web" middleware group. Now create something
great!

|

*/

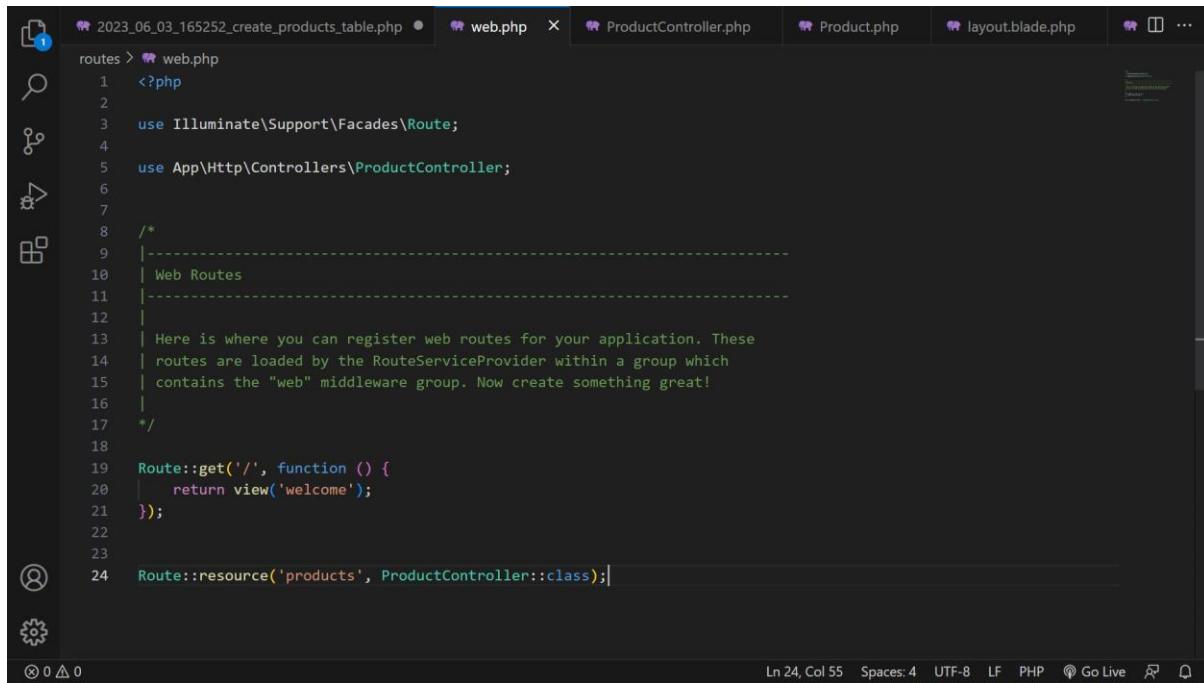
```
Route::get('/', function () {
```

```
    return view('welcome');
```

```
});
```

```
Route::resource('products', ProductController::class);
```

Screenshot of code Web.php:



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\ProductController;
6
7
8  /*
9  |
10 | Web Routes
11 |
12 |
13 | Here is where you can register web routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | contains the "web" middleware group. Now create something great!
16 |
17 */
18
19 Route::get('/', function () {
20     return view('welcome');
21 });
22
23
24 Route::resource('products', ProductController::class);
```

Ln 24, Col 55 Spaces: 4 UTF-8 LF PHP ⌂ Go Live ⌂ ⌂

Step 5: Create Controllers and Model

Now let's create our UserController just run the following command below:

```
php artisan make:controller ProductController --resource --model=Product
```

After running the command above you will find the generated controller to your **"App/Http/Controllers"**. Now let's add our methods for our Laravel crud operation example. Take note that we are using users to do this with a simple example.

In your `ProductController` you will have these methods:

- 1. index()**
- 2. create()**
- 3. store()**
- 4. show()**
- 5. edit()**
- 6. update()**
- 7. destroy()**

ProductController.php

Code:

```
<?php

namespace App\Http\Controllers;

use App\Models\Product;

use Illuminate\Http\Request;

class ProductController extends Controller

{

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */

    public function index()

    {

        $products = Product::latest()->paginate(5);

        return view('products.index',compact('products'))

            ->with('i', (request()->input('page', 1) - 1) * 5);

    }

    /**
     * Show the form for creating a new resource.
     *

```

```
* @return \Illuminate\Http\Response
*/
public function create()
{
    return view('products.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);
}
```

Product::create(\$request->all());

return redirect()->route('products.index')

```
        ->with('success','Product created successfully.');
```

```
}
```

```
/**
```

```
 * Display the specified resource.
```

```
 *
```

```
 * @param \App\Product $product
```

```
 * @return \Illuminate\Http\Response
```

```
 */
```

```
public function show(Product $product)
```

```
{
```

```
    return view('products.show',compact('product'));
```

```
}
```

```
/**
```

```
 * Show the form for editing the specified resource.
```

```
 *
```

```
 * @param \App\Product $product
```

```
 * @return \Illuminate\Http\Response
```

```
 */
```

```
public function edit(Product $product)
```

```
{
```

```
        return view('products.edit',compact('product'));

    }

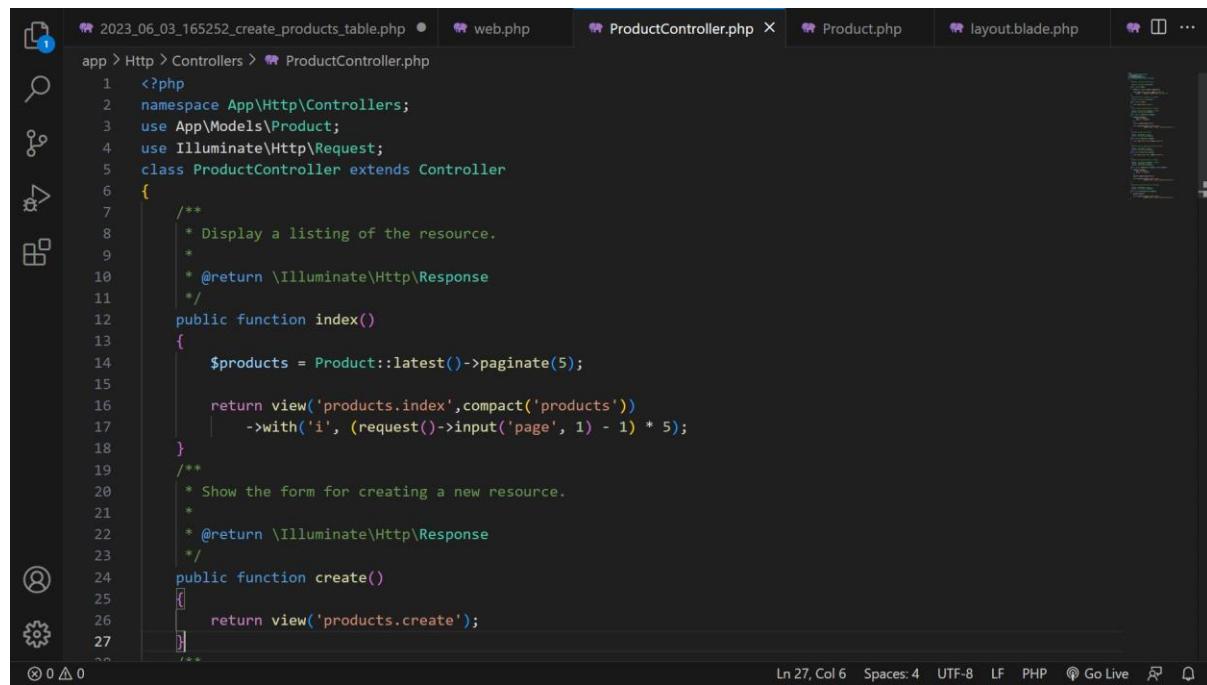
/** 
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Product $product)
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);

    $product->update($request->all());

    return redirect()->route('products.index')
        ->with('success','Product updated successfully');
}
```

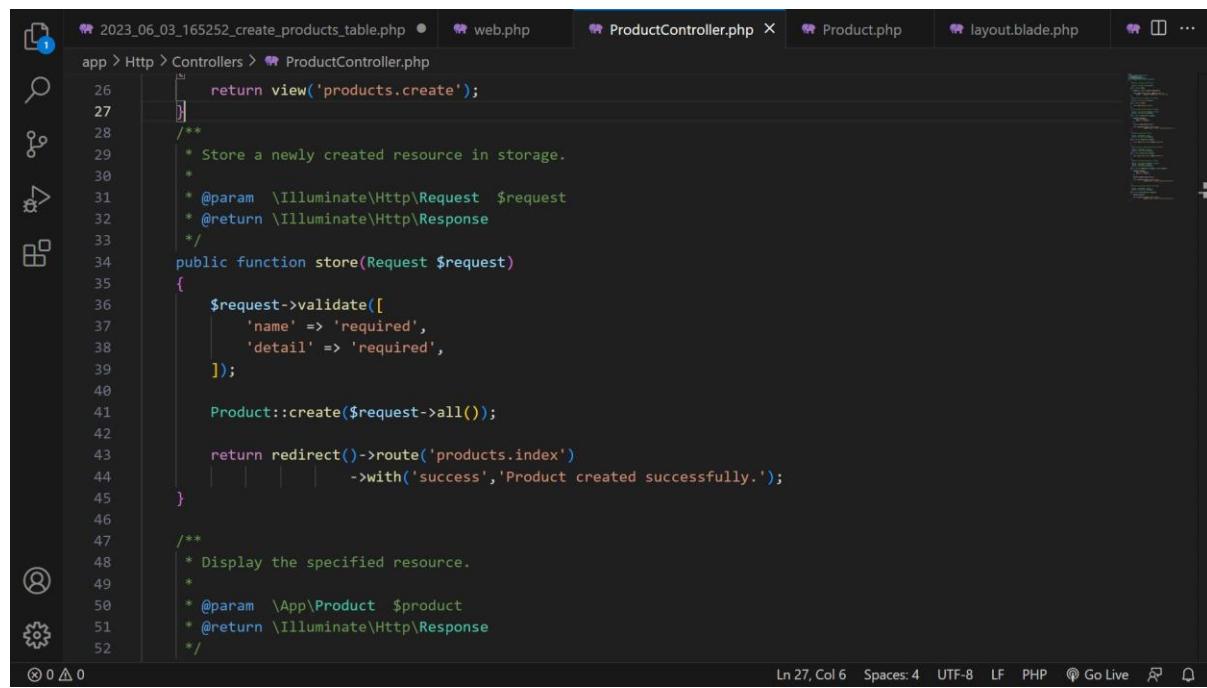
```
/**  
 * Remove the specified resource from storage.  
 *  
 * @param \App\Product $product  
 * @return \Illuminate\Http\Response  
 */  
public function destroy(Product $product)  
{  
    $product->delete();  
  
    return redirect()->route('products.index')  
        ->with('success','Product deleted successfully');  
}  
}
```

Screenshot of code:



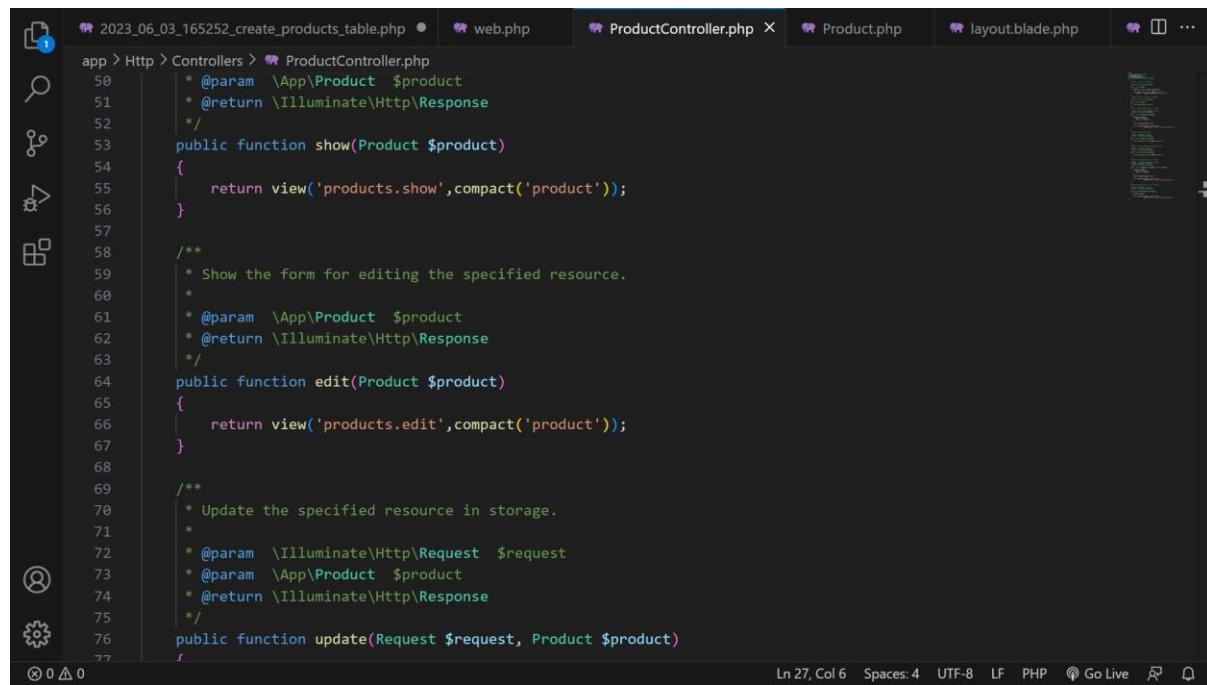
```
app > Http > Controllers > ProductController.php
1  <?php
2  namespace App\Http\Controllers;
3  use App\Models\Product;
4  use Illuminate\Http\Request;
5  class ProductController extends Controller
6  {
7      /**
8      * Display a listing of the resource.
9      *
10     * @return \Illuminate\Http\Response
11    */
12    public function index()
13    {
14        $products = Product::latest()->paginate(5);
15
16        return view('products.index',compact('products'))
17            ->with('i', (request()->input('page', 1) - 1) * 5);
18    }
19    /**
20     * Show the form for creating a new resource.
21     *
22     * @return \Illuminate\Http\Response
23    */
24    public function create()
25    {
26        return view('products.create');
27    }
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
```

Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂



```
app > Http > Controllers > ProductController.php
26    return view('products.create');
27
28    /**
29     * Store a newly created resource in storage.
30     *
31     * @param \Illuminate\Http\Request $request
32     * @return \Illuminate\Http\Response
33     */
34    public function store(Request $request)
35    {
36        $request->validate([
37            'name' => 'required',
38            'detail' => 'required',
39        ]);
40
41        Product::create($request->all());
42
43        return redirect()->route('products.index')
44            ->with('success','Product created successfully.');
45    }
46
47    /**
48     * Display the specified resource.
49     *
50     * @param \App\Product $product
51     * @return \Illuminate\Http\Response
52     */
53
```

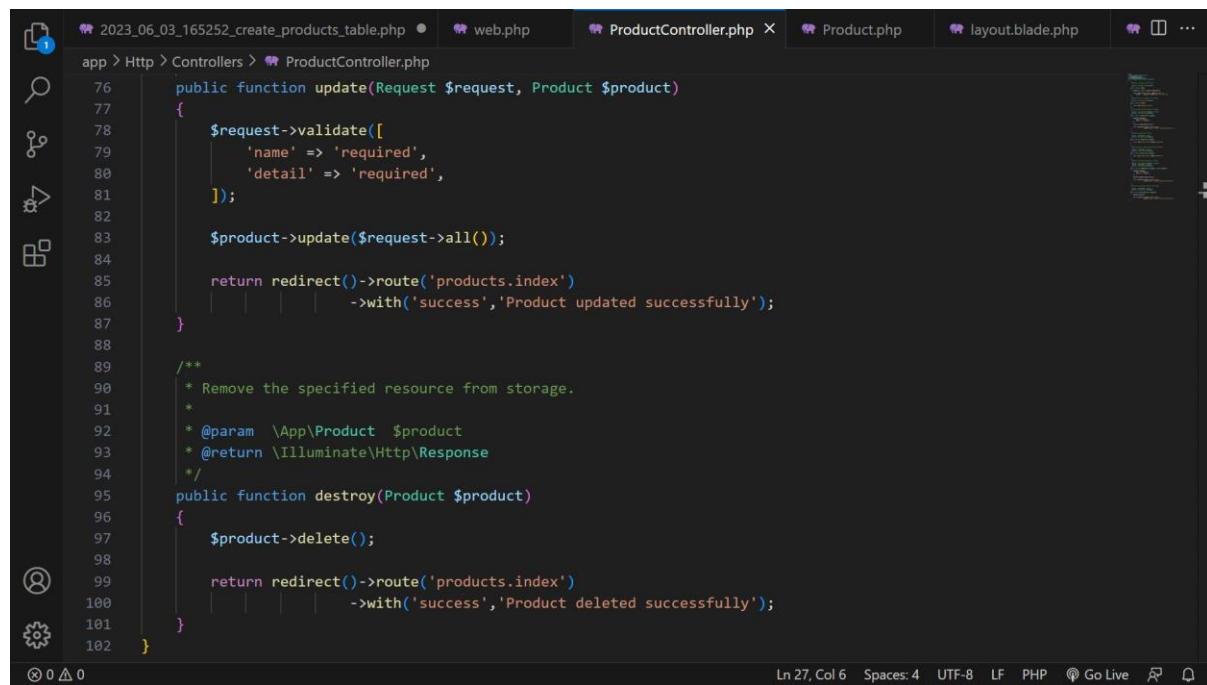
Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂



app > Http > Controllers > ProductController.php

```
50  * @param \App\Product $product
51  * @return \Illuminate\Http\Response
52  */
53  public function show(Product $product)
54  {
55      return view('products.show',compact('product'));
56  }
57
58  /**
59  * Show the form for editing the specified resource.
60  *
61  * @param \App\Product $product
62  * @return \Illuminate\Http\Response
63  */
64  public function edit(Product $product)
65  {
66      return view('products.edit',compact('product'));
67  }
68
69  /**
70  * Update the specified resource in storage.
71  *
72  * @param \Illuminate\Http\Request $request
73  * @param \App\Product $product
74  * @return \Illuminate\Http\Response
75  */
76  public function update(Request $request, Product $product)
77  {
78      $request->validate([
79          'name' => 'required',
80          'detail' => 'required',
81      ]);
82
83      $product->update($request->all());
84
85      return redirect()->route('products.index')
86          ->with('success','Product updated successfully');
87  }
88
89  /**
90  * Remove the specified resource from storage.
91  *
92  * @param \App\Product $product
93  * @return \Illuminate\Http\Response
94  */
95  public function destroy(Product $product)
96  {
97      $product->delete();
98
99      return redirect()->route('products.index')
100         ->with('success','Product deleted successfully');
101 }
```

Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂



app > Http > Controllers > ProductController.php

```
76  public function update(Request $request, Product $product)
77  {
78      $request->validate([
79          'name' => 'required',
80          'detail' => 'required',
81      ]);
82
83      $product->update($request->all());
84
85      return redirect()->route('products.index')
86          ->with('success','Product updated successfully');
87  }
88
89  /**
90  * Remove the specified resource from storage.
91  *
92  * @param \App\Product $product
93  * @return \Illuminate\Http\Response
94  */
95  public function destroy(Product $product)
96  {
97      $product->delete();
98
99      return redirect()->route('products.index')
100         ->with('success','Product deleted successfully');
101 }
```

Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂

Product.php

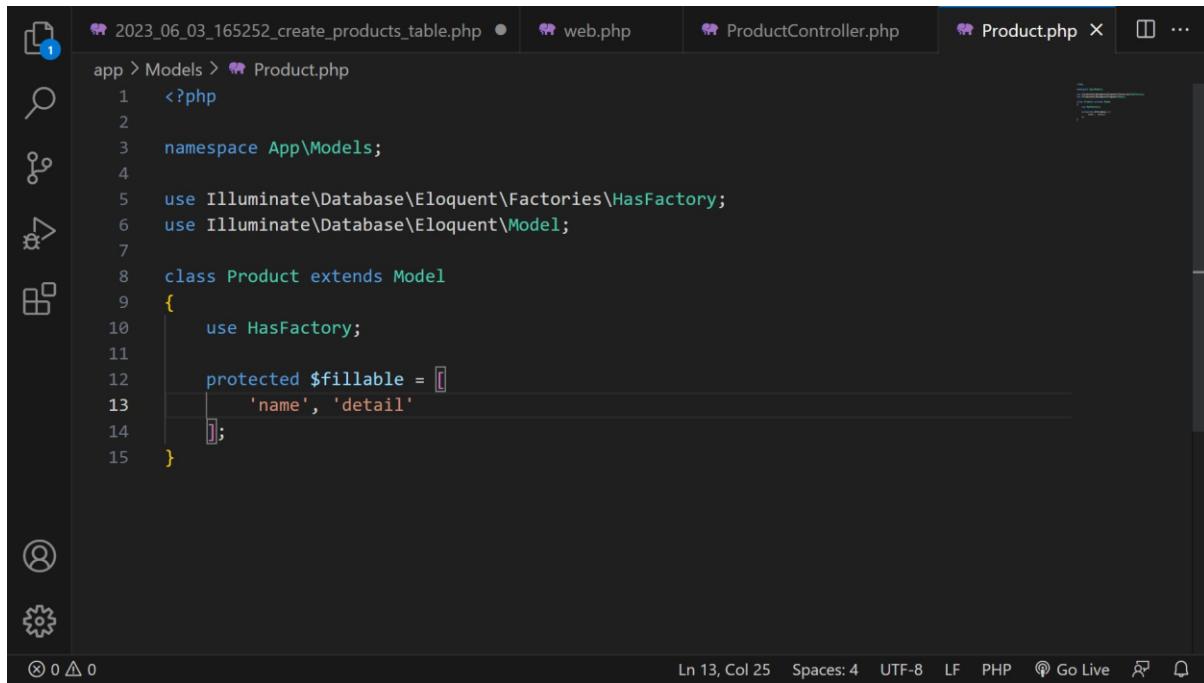
Code:

```
<?php
```

namespace App\Models;

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Product extends Model  
{  
    use HasFactory;  
  
    protected $fillable = [  
        'name', 'detail'  
    ];  
}
```

Screenshot of code:



```
2023_06_03_165252_create_products_table.php ● web.php ProductController.php Product.php X ...  
app > Models > Product.php  
1 <?php  
2  
3 namespace App\Models;  
4  
5 use Illuminate\Database\Eloquent\Factories\HasFactory;  
6 use Illuminate\Database\Eloquent\Model;  
7  
8 class Product extends Model  
9 {  
10     use HasFactory;  
11  
12     protected $fillable = [  
13         'name', 'detail'  
14     ];  
15 }  
  
Ln 13, Col 25 Spaces: 4 UTF-8 LF PHP ⚡ Go Live 🔍
```

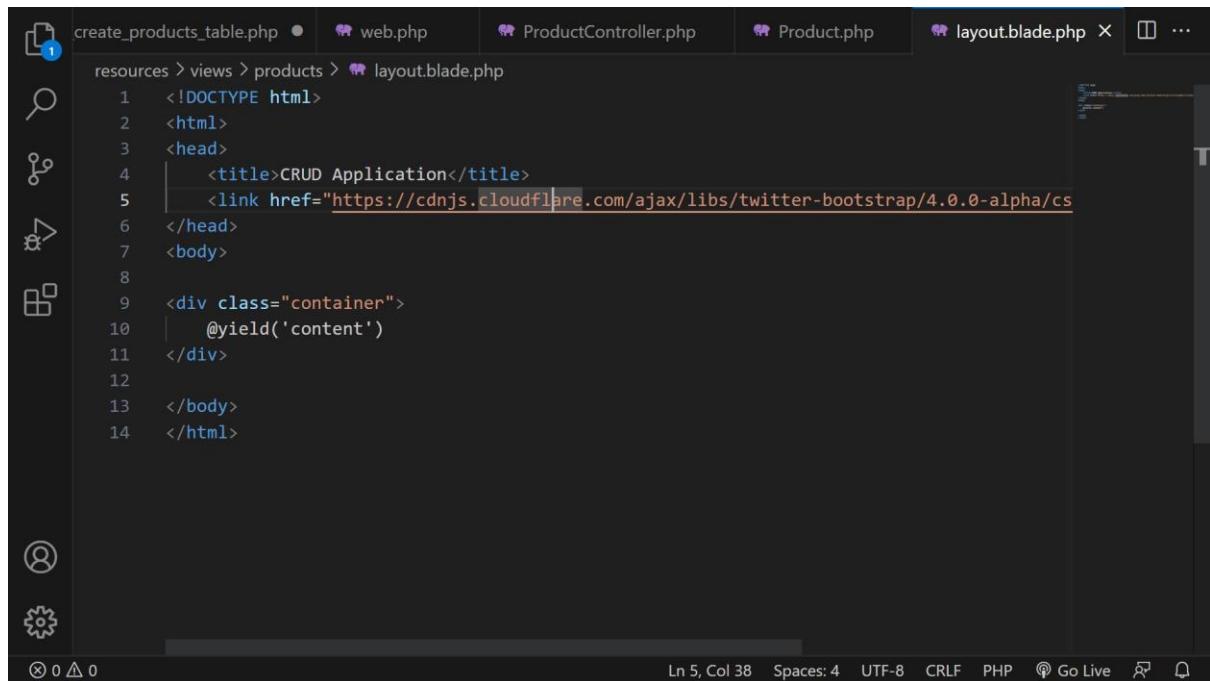
6. Create all blade.php file and write code

Layout.blade.php Code:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
    <title>CRUD Application</title>  
  
    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">  
  
</head>  
  
<body>  
  
  
<div class="container">  
  
    @yield('content')  
  
</div>
```

```
</body>  
</html>
```

Screenshot of code:



```
resources > views > products > layout.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>CRUD Application</title>
5  |   <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/cs
6  </head>
7  <body>
8
9  <div class="container">
10 |   @yield('content')
11 </div>
12
13 </body>
14 </html>
```

index.blade.php Code:

```
@extends('products.layout')  
  
@section('content')  
    <div class="row">  
        <div class="col-lg-12 margin-tb">  
            <div class="pull-left">  
                <h2>CRUD Application</h2>  
            </div>  
            <div class="pull-right">
```

```
        <a class="btn btn-success" href="{{ route('products.create') }}> Create New Product</a>
    </div>
</div>
</div>
```

```
@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif
```

```
<table class="table table-bordered">
    <tr>
        <th>No</th>
        <th>Name</th>
        <th>Details</th>
        <th width="280px">Action</th>
    </tr>
    @foreach ($products as $product)
        <tr>
            <td>{{ ++$i }}</td>
            <td>{{ $product->name }}</td>
            <td>{{ $product->detail }}</td>
```

```

<td>

    <form action="{{ route('products.destroy', $product->id) }}"
method="POST">

        <a class="btn btn-info" href="{{
route('products.show', $product->id) }}>Show</a>

        <a class="btn btn-primary" href="{{
route('products.edit', $product->id) }}>Edit</a>

        @csrf
        @method('DELETE')

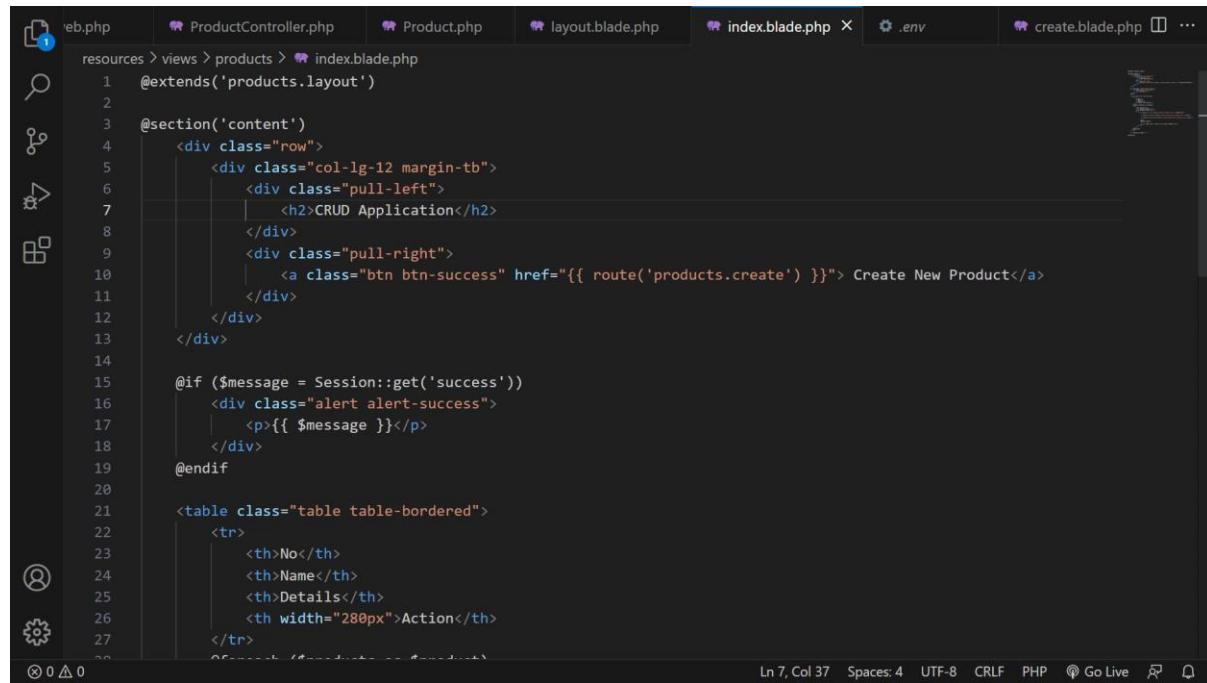
        <button type="submit" class="btn btn-
danger">Delete</button>
    </form>
</td>
</tr>
@endforeach
</table>

{!! $products->links() !!}

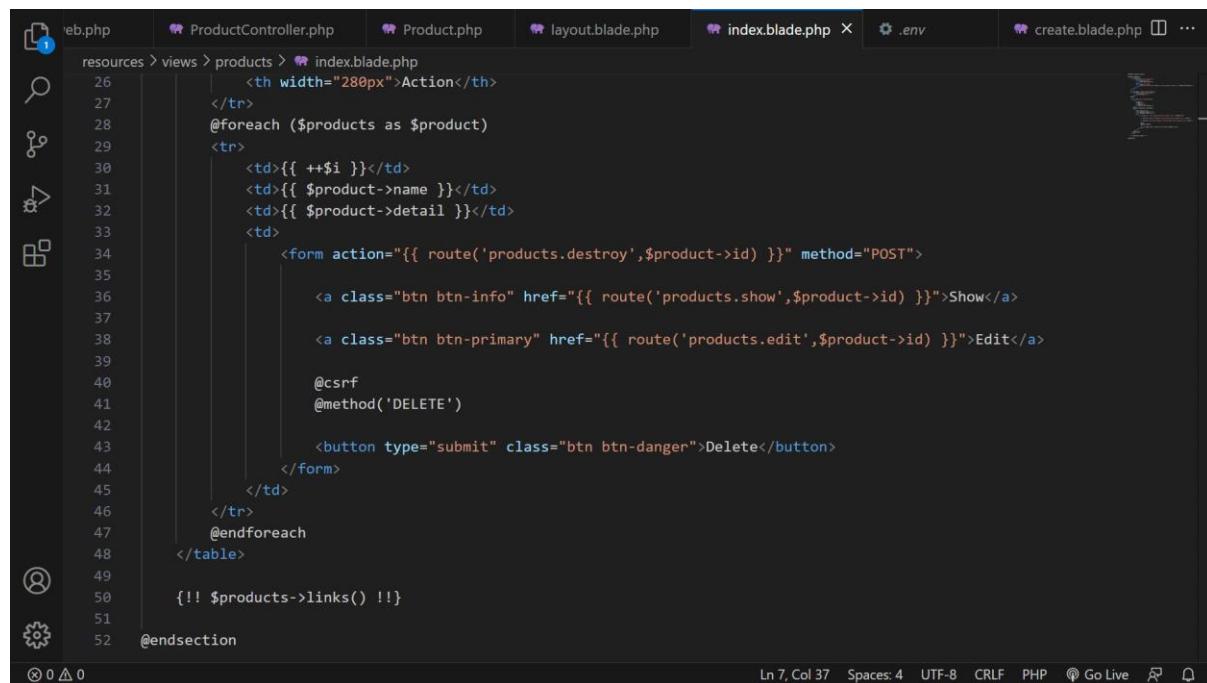
```

@endsection

Screenshot of code:



```
resources > views > products > index.blade.php
1  @extends('products.layout')
2
3  @section('content')
4      <div class="row">
5          <div class="col-lg-12 margin-tb">
6              <div class="pull-left">
7                  <h2>CRUD Application</h2>
8              </div>
9              <div class="pull-right">
10                 <a class="btn btn-success" href="{{ route('products.create') }}> Create New Product</a>
11             </div>
12         </div>
13     </div>
14
15     @if ($message = Session::get('success'))
16         <div class="alert alert-success">
17             <p>{{ $message }}</p>
18         </div>
19     @endif
20
21     <table class="table table-bordered">
22         <tr>
23             <th>No</th>
24             <th>Name</th>
25             <th>Details</th>
26             <th width="280px">Action</th>
27         </tr>
28     </table>
29
30     @endif
31
32     @endsection
33
34     @if ($products)
35         <table class="table table-bordered">
36             <thead>
37                 <tr>
38                     <th width="280px">Action</th>
39                 </tr>
40             </thead>
41             <tbody>
42                 @foreach ($products as $product)
43                     <tr>
44                         <td>{{ ++$i }}</td>
45                         <td>{{ $product->name }}</td>
46                         <td>{{ $product->detail }}</td>
47                         <td>
48                             <form action="{{ route('products.destroy', $product->id) }}" method="POST">
49                                 @csrf
50                                 @method('DELETE')
51                                 <button type="submit" class="btn btn-danger">Delete</button>
52                             </form>
53                         </td>
54                     </tr>
55                 @endforeach
56             </tbody>
57         </table>
58     @endif
59
60     @endif
61
62     @endif
63
64     @endif
65
66     @endif
67
68     @endif
69
70     @endif
71
72     @endif
73
74     @endif
75
76     @endif
77
78     @endif
79
80     @endif
81
82     @endif
83
84     @endif
85
86     @endif
87
88     @endif
89
90     @endif
91
92     @endif
93
94     @endif
95
96     @endif
97
98     @endif
99
100    @endif
101
102    @endif
103
104    @endif
105
106    @endif
107
108    @endif
109
110    @endif
111
112    @endif
113
114    @endif
115
116    @endif
117
118    @endif
119
120    @endif
121
122    @endif
123
124    @endif
125
126    @endif
127
128    @endif
129
130    @endif
131
132    @endif
133
134    @endif
135
136    @endif
137
138    @endif
139
140    @endif
141
142    @endif
143
144    @endif
145
146    @endif
147
148    @endif
149
150    @endif
151
152    @endif
153
154    @endif
155
156    @endif
157
158    @endif
159
160    @endif
161
162    @endif
163
164    @endif
165
166    @endif
167
168    @endif
169
170    @endif
171
172    @endif
173
174    @endif
175
176    @endif
177
178    @endif
179
180    @endif
181
182    @endif
183
184    @endif
185
186    @endif
187
188    @endif
189
190    @endif
191
192    @endif
193
194    @endif
195
196    @endif
197
198    @endif
199
200    @endif
201
202    @endif
203
204    @endif
205
206    @endif
207
208    @endif
209
210    @endif
211
212    @endif
213
214    @endif
215
216    @endif
217
218    @endif
219
220    @endif
221
222    @endif
223
224    @endif
225
226    @endif
227
228    @endif
229
230    @endif
231
232    @endif
233
234    @endif
235
236    @endif
237
238    @endif
239
240    @endif
241
242    @endif
243
244    @endif
245
246    @endif
247
248    @endif
249
250    @endif
251
252    @endif
253
254    @endif
255
256    @endif
257
258    @endif
259
260    @endif
261
262    @endif
263
264    @endif
265
266    @endif
267
268    @endif
269
270    @endif
271
272    @endif
273
274    @endif
275
276    @endif
277
278    @endif
279
280    @endif
281
282    @endif
283
284    @endif
285
286    @endif
287
288    @endif
289
290    @endif
291
292    @endif
293
294    @endif
295
296    @endif
297
298    @endif
299
299
```



```
resources > views > products > index.blade.php
26     <th width="280px">Action</th>
27     </tr>
28     @foreach ($products as $product)
29         <tr>
30             <td>{{ ++$i }}</td>
31             <td>{{ $product->name }}</td>
32             <td>{{ $product->detail }}</td>
33             <td>
34                 <form action="{{ route('products.destroy', $product->id) }}" method="POST">
35                     @csrf
36                     @method('DELETE')
37                     <button type="submit" class="btn btn-danger">Delete</button>
38                 </form>
39             </td>
40         </tr>
41     @endforeach
42     </table>
43
44     @endif $products->links() !!
45
46     @endsection
47
48     @endif
49
50     @endif
51
52     @endif
53
54     @endif
55
56     @endif
57
58     @endif
59
59
```

create.blade.php Code:

```
@extends('products.layout')
```

```
@section('content')
```

```
<div class="row">
  <div class="col-lg-12 margin-tb">
    <div class="pull-left">
      <h2>Add New Product</h2>
    </div>
    <div class="pull-right">
      <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
    </div>
  </div>
</div>

@if ($errors->any())
  <div class="alert alert-danger">
    <strong>Whoops!</strong> There were some problems with
    your input.<br><br>
    <ul>
      @foreach ($errors->all() as $error)
        <li>{{ $error }}</li>
      @endforeach
    </ul>
  </div>
@endif
```

```
<form action="{{ route('products.store') }}" method="POST">

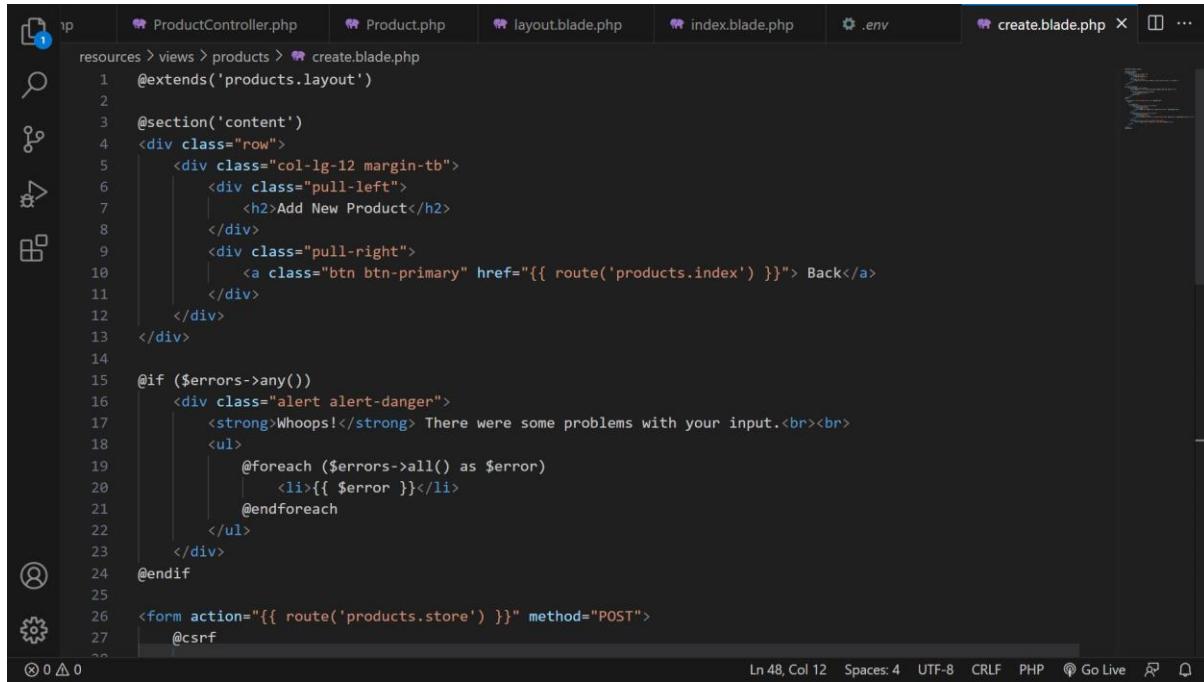
    @csrf

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Name:</strong>
                <input type="text" name="name" class="form-control" placeholder="Name">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Detail:</strong>
                <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>

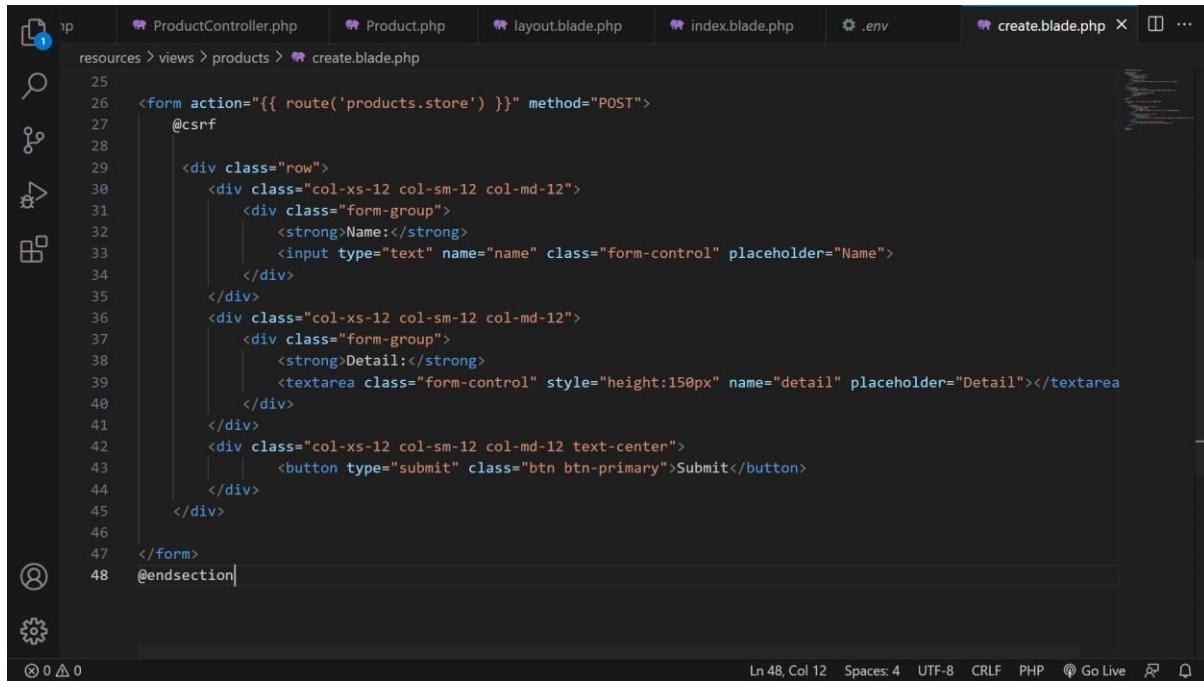
</form>
```

```
@endsection
```

Screenshot of code:



```
resources > views > products > create.blade.php
1  @extends('products.layout')
2
3  @section('content')
4      <div class="row">
5          <div class="col-lg-12 margin-tb">
6              <div class="pull-left">
7                  <h2>Add New Product</h2>
8              </div>
9              <div class="pull-right">
10                 <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
11             </div>
12         </div>
13     </div>
14
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Whoops!</strong> There were some problems with your input.<br><br>
18             <ul>
19                 @foreach ($errors->all() as $error)
20                     <li>{{ $error }}</li>
21                 @endforeach
22             </ul>
23         </div>
24     @endif
25
26     <form action="{{ route('products.store') }}" method="POST">
27         @csrf
```



```
resources > views > products > create.blade.php
25
26     <form action="{{ route('products.store') }}" method="POST">
27         @csrf
28
29         <div class="row">
30             <div class="col-xs-12 col-sm-12 col-md-12">
31                 <div class="form-group">
32                     <strong>Name:</strong>
33                     <input type="text" name="name" class="form-control" placeholder="Name">
34                 </div>
35             </div>
36             <div class="col-xs-12 col-sm-12 col-md-12">
37                 <div class="form-group">
38                     <strong>Detail:</strong>
39                     <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
40                 </div>
41             </div>
42             <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43                 <button type="submit" class="btn btn-primary">Submit</button>
44             </div>
45         </div>
46
47     </form>
48     @endsection
```

edit.blade.php Code:

```
@extends('products.layout')
```

```
@section('content')

<div class="row">

    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Edit Product</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
        </div>
    </div>
</div>

@if ($errors->any())
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with
        your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>

```

```
@endif

<form action="{{ route('products.update',$product->id) }}"
method="POST">

    @csrf

    @method('PUT')


<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            <input type="text" name="name" value="{{ $product->name }}" class="form-control" placeholder="Name">
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Detail:</strong>
            <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail">{{ $product->detail }}</textarea>
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
```

```
        <button type="submit" class="btn btn-primary">Submit</button>
```

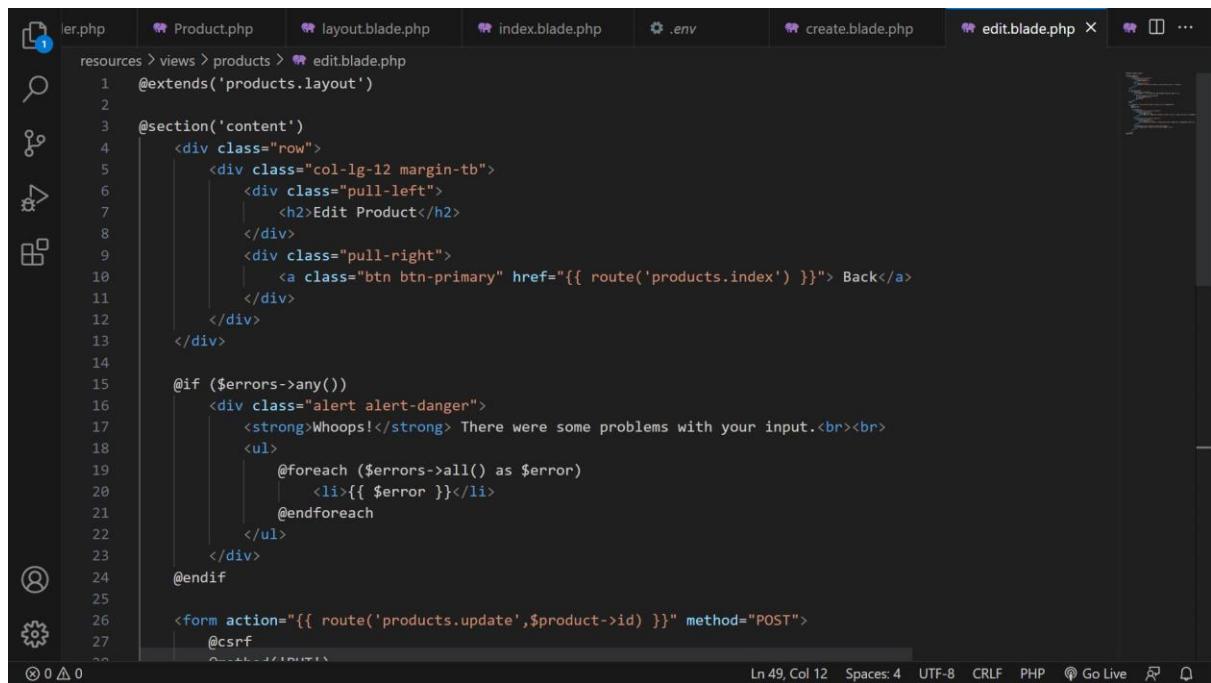
```
</div>
```

```
</div>
```

```
</form>
```

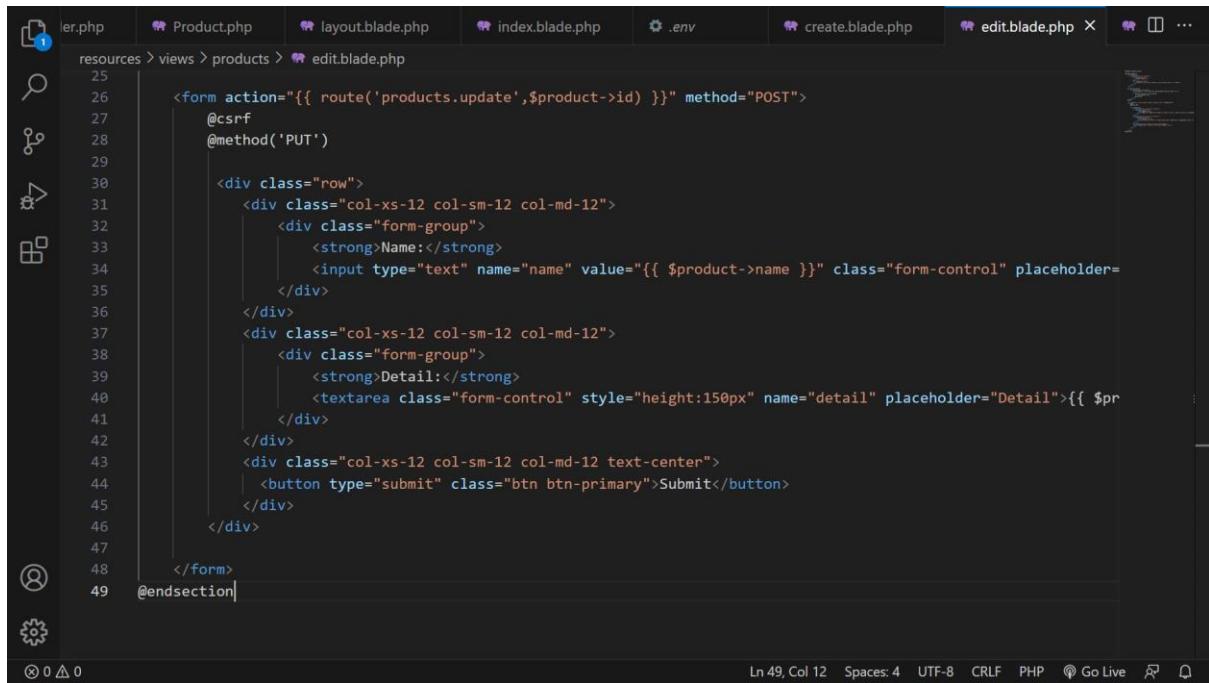
```
@endsection
```

Screenshot of code:



The screenshot shows a code editor with a dark theme, displaying a Laravel blade template for editing a product. The file is named `edit.blade.php` and is located in the `resources/views/products` directory. The code includes sections for content, error handling, and a form for updating the product.

```
ler.php Product.php layout.blade.php index.blade.php .env create.blade.php edit.blade.php ...  
resources > views > products > edit.blade.php  
1  @extends('products.layout')  
2  
3  @section('content')  
4      <div class="row">  
5          <div class="col-lg-12 margin-tb">  
6              <div class="pull-left">  
7                  <h2>Edit Product</h2>  
8              </div>  
9              <div class="pull-right">  
10                 <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>  
11             </div>  
12         </div>  
13     </div>  
14  
15     @if ($errors->any())  
16         <div class="alert alert-danger">  
17             <strong>Whoops!</strong> There were some problems with your input.<br><br>  
18             <ul>  
19                 @foreach ($errors->all() as $error)  
20                     <li>{{ $error }}</li>  
21                 @endforeach  
22             </ul>  
23         </div>  
24     @endif  
25  
26     <form action="{{ route('products.update', $product->id) }}" method="POST">  
27         @csrf
```



```
resources > views > products > edit.blade.php
25  <form action="{{ route('products.update', $product->id) }}" method="POST">
26      @csrf
27      @method('PUT')
28
29      <div class="row">
30          <div class="col-xs-12 col-sm-12 col-md-12">
31              <div class="form-group">
32                  <strong>Name:</strong>
33                  <input type="text" name="name" value="{{ $product->name }}" class="form-control" placeholder="Name">
34              </div>
35          </div>
36          <div class="col-xs-12 col-sm-12 col-md-12">
37              <div class="form-group">
38                  <strong>Detail:</strong>
39                  <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail">{{ $product->detail }}</textarea>
40              </div>
41          </div>
42          <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43              <button type="submit" class="btn btn-primary">Submit</button>
44          </div>
45      </div>
46  </form>
47
48  @endsection|
```

show.blade.php Code:

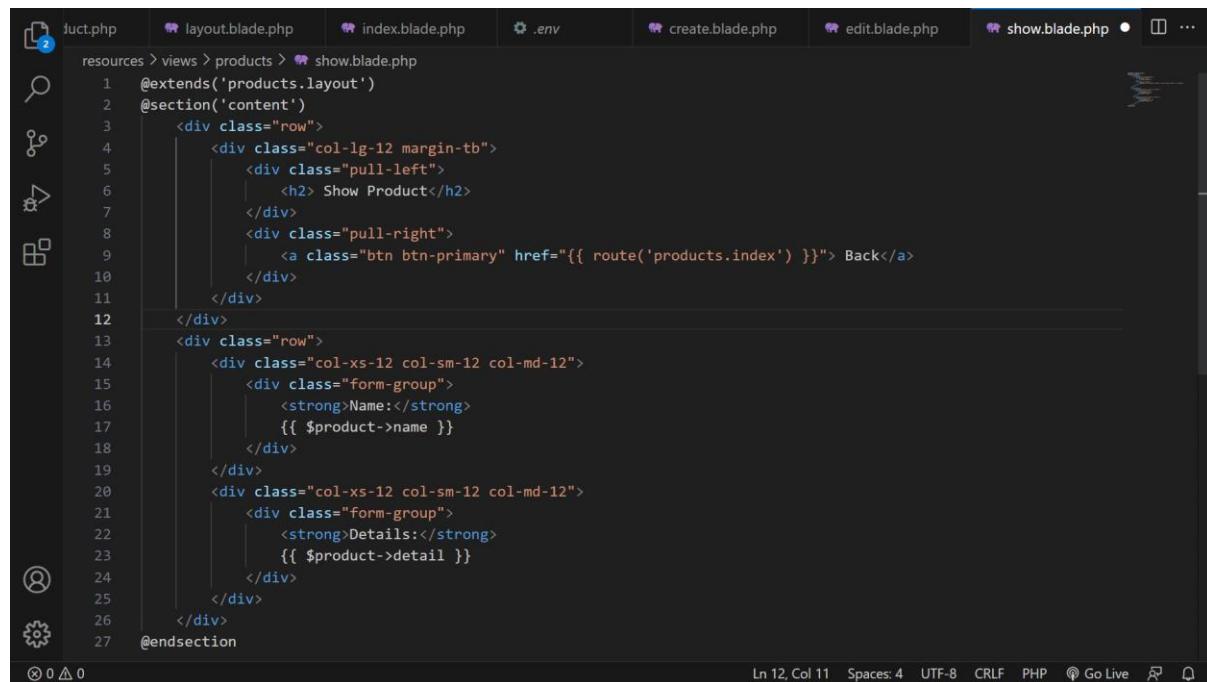
```
@extends('products.layout')

@section('content')

<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2> Show Product</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
        </div>
    </div>
</div>
</div>
```

```
<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Name:</strong>
      {{ $product->name }}
    </div>
  </div>
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Details:</strong>
      {{ $product->detail }}
    </div>
  </div>
</div>
@endsection
```

Screenshot of code:



The screenshot shows a code editor with a dark theme. The file is 'show.blade.php' located in the 'products' directory under 'views'. The code is a Blade template:

```
1  @extends('products.layout')
2  @section('content')
3      <div class="row">
4          <div class="col-lg-12 margin-tb">
5              <div class="pull-left">
6                  <h2> Show Product</h2>
7              </div>
8              <div class="pull-right">
9                  <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
10             </div>
11         </div>
12     </div>
13     <div class="row">
14         <div class="col-xs-12 col-sm-12 col-md-12">
15             <div class="form-group">
16                 <strong>Name:</strong>
17                 {{ $product->name }}
18             </div>
19         </div>
20         <div class="col-xs-12 col-sm-12 col-md-12">
21             <div class="form-group">
22                 <strong>Details:</strong>
23                 {{ $product->detail }}
24             </div>
25         </div>
26     </div>
27     @endsection
```

The status bar at the bottom shows: Ln 12, Col 11, Spaces: 4, UTF-8, CRLF, PHP, Go Live, and a refresh icon.

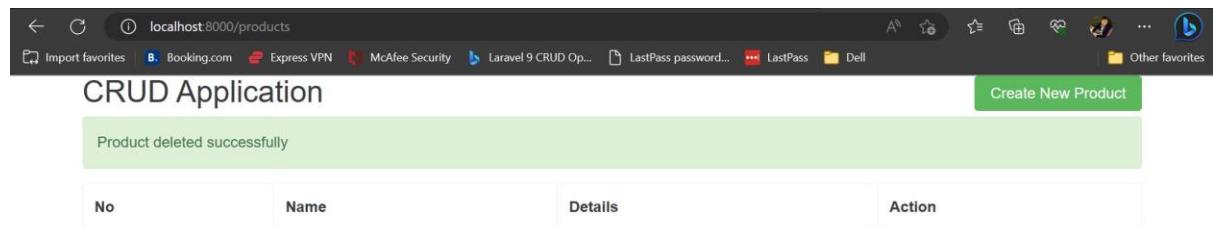
**Step 7: Then final step to run the last command
“php artisan serve”**

Step 8: Then go to browser and this code

127.0.0.1:8000/products

Screenshot

Output:



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

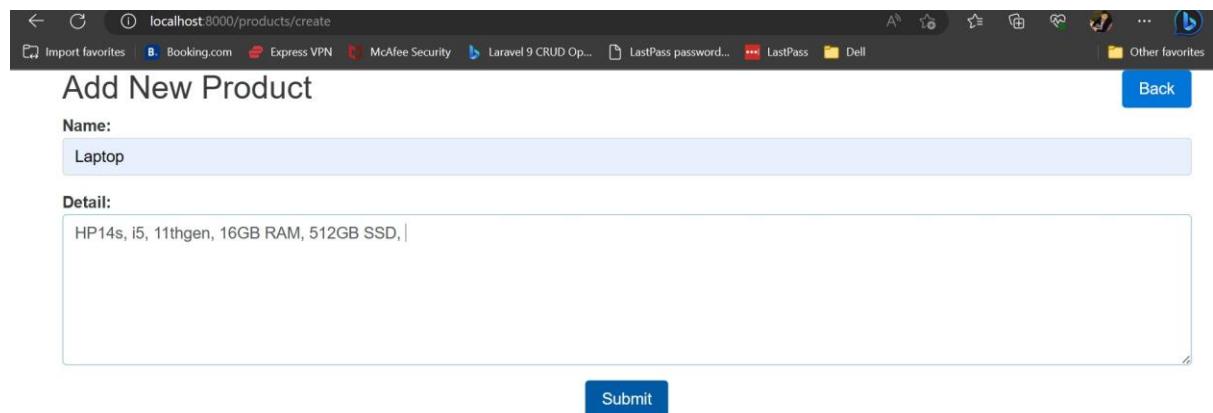
CRUD Application

Create New Product

Product deleted successfully

No	Name	Details	Action

Create Product Name and Details



localhost:8000/products/create

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

Add New Product

Back

Name: Laptop

Detail: HP14s, i5, 11thgen, 16GB RAM, 512GB SSD, |

Submit

Created

The screenshot shows a browser window with the URL `localhost:8000/products`. The title bar says "CRUD Application". A green success message box says "Product created successfully.". Below is a table with one row of data:

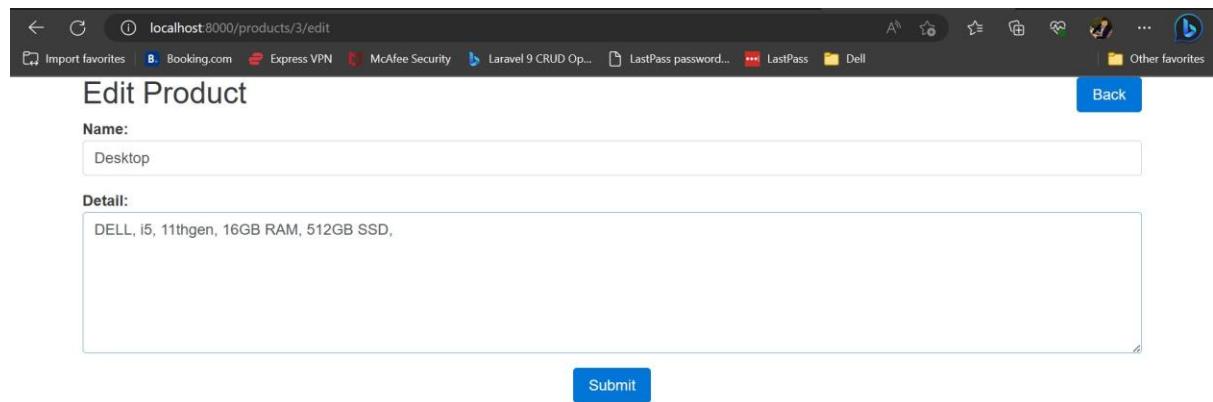
No	Name	Details	Action
1	Laptop	HP14s, i5, 11thgen, 16GB RAM, 512GB SSD,	Show Edit Delete

Show Product

The screenshot shows a browser window with the URL `localhost:8000/products/3`. The title bar says "Show Product". The page displays the product details for the item with ID 3:

Name: Laptop
Details: HP14s, i5, 11thgen, 16GB RAM, 512GB SSD,

Edit Product



localhost:8000/products/3/edit

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

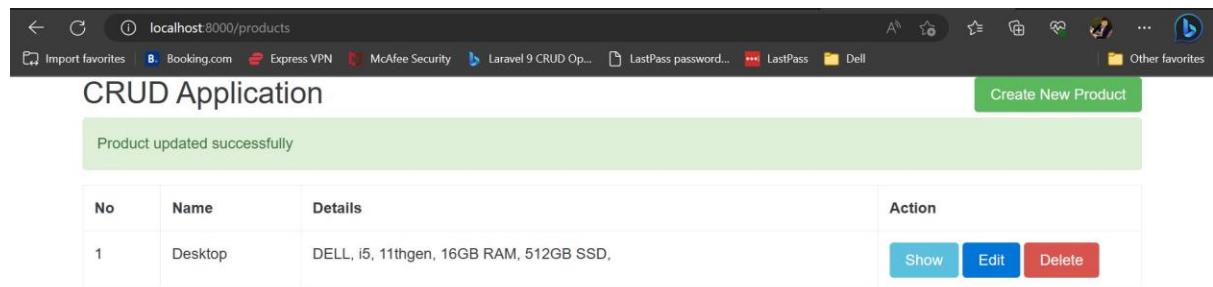
Edit Product

Name: Desktop

Detail: DELL, i5, 11thgen, 16GB RAM, 512GB SSD,

Submit

Edit successfully



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

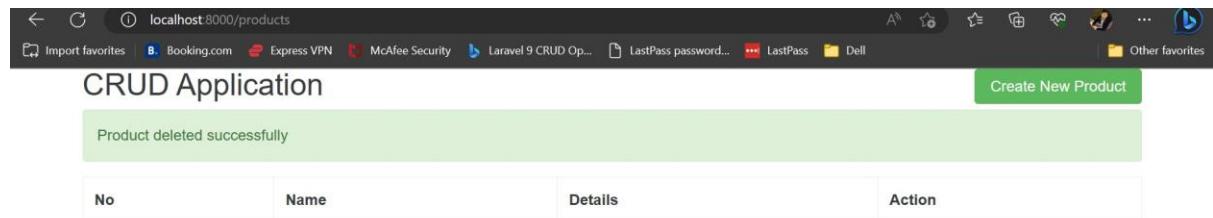
CRUD Application

Create New Product

Product updated successfully

No	Name	Details	Action
1	Desktop	DELL, i5, 11thgen, 16GB RAM, 512GB SSD,	Show Edit Delete

Delete



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

CRUD Application

Create New Product

Product deleted successfully

No	Name	Details	Action
----	------	---------	--------

8 Aim: - Create a complete end-to-end solution which demonstrate session and database connectivity.

Learning outcome: - Able to understand end-to-end solution in Laravel framework.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:-

Open your Command prompt and Please enter a Laravel command

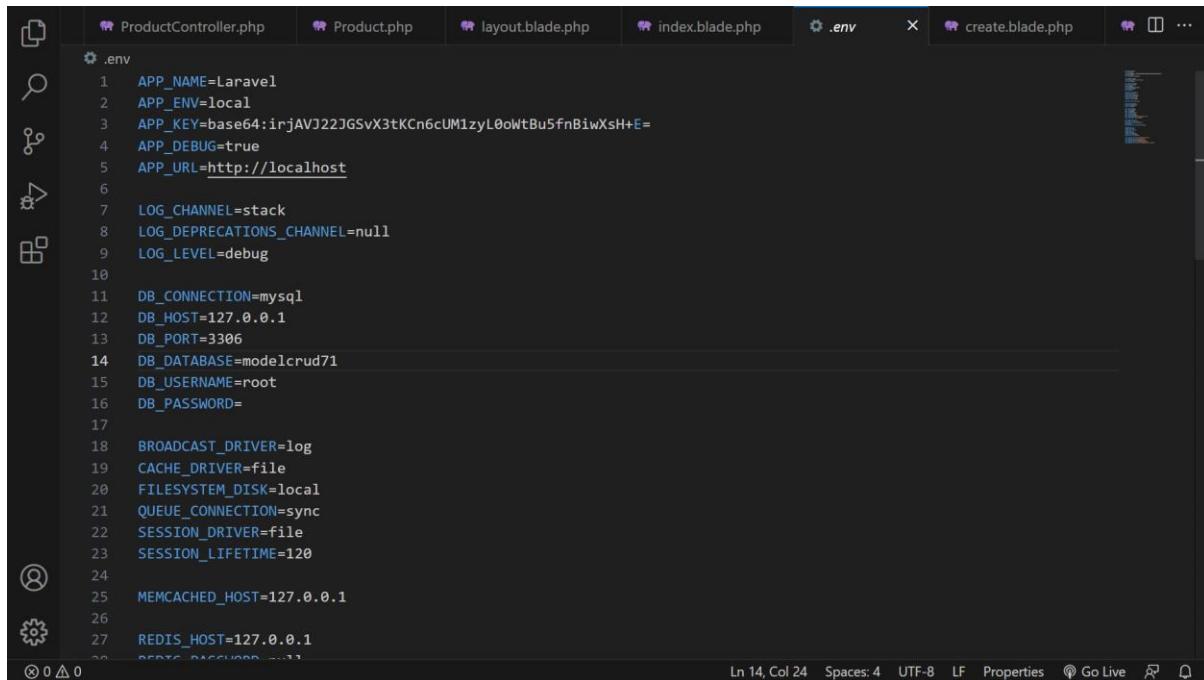
```
composer create-project --prefer-dist  
laravel/Laravel modelcrud71
```

Step 2: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:irjAVJ22JG5vX3tKCn6cUM1zyL0oWtBu5fnBiwXsH+E=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=modelcrud71
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=
```

Step 3: Migration Setup

Let's create a migration for our Laravel crud operation. If you are using a fresh Laravel project we need to customize the user's table because I added the username field in my authentication. Here are the complete migrations below:

Code:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
    
```

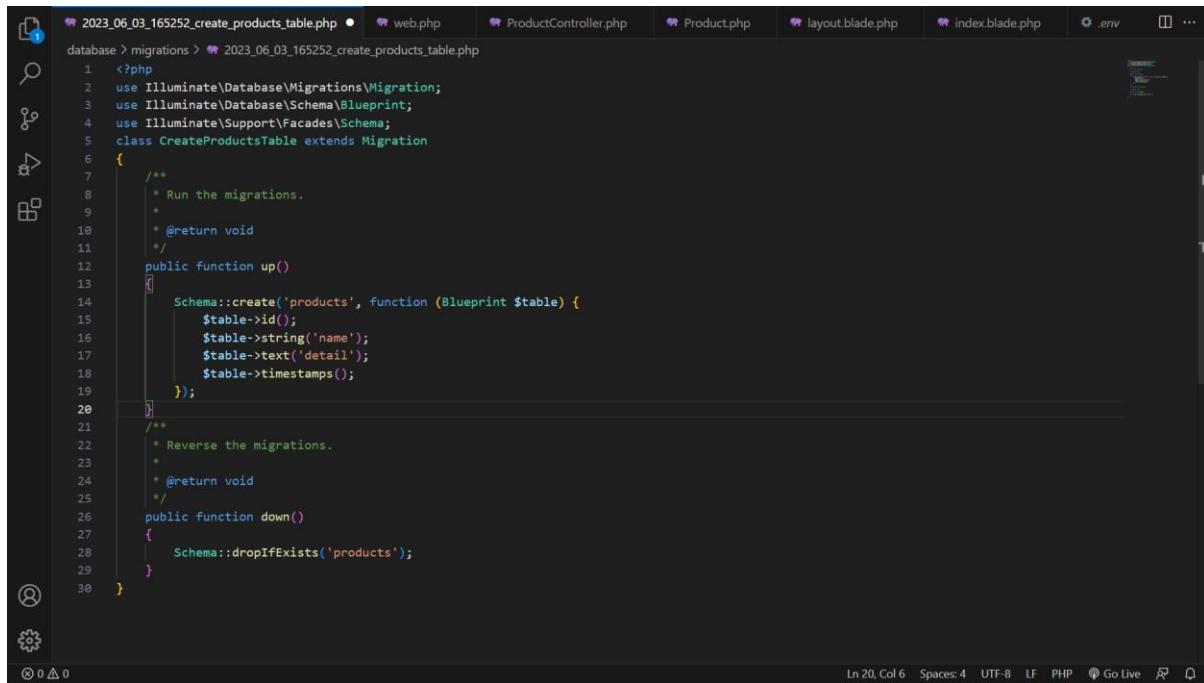
```

*
* @return void
*/
public function up()
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->text('detail');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
*/
public function down()
{
    Schema::dropIfExists('products');
}
}

```

Screenshot of code:



```
2023_06_03_165252_create_products_table.php ● 2023_06_03_165252_create_products_table.php
database > migrations > 2023_06_03_165252_create_products_table.php
1  <?php
2  use Illuminate\Database\Migrations\Migration;
3  use Illuminate\Database\Schema\Blueprint;
4  use Illuminate\Support\Facades\Schema;
5  class CreateProductsTable extends Migration
6  {
7      /**
8      * Run the migrations.
9      *
10     * @return void
11    */
12    public function up()
13    {
14        Schema::create('products', function (Blueprint $table) {
15            $table->id();
16            $table->string('name');
17            $table->text('detail');
18            $table->timestamps();
19        });
20    }
21    /**
22     * Reverse the migrations.
23     *
24     * @return void
25    */
26    public function down()
27    {
28        Schema::dropIfExists('products');
29    }
30 }

```

Ln 20, Col 6 Spaces: 4 UTF-8 LF PHP Go Live ⌂ ⌂

Now let's run the following command below:

```
php artisan migrate
```

Step 4: Operation Routes Add Resource Route

In my example, I will create manually my CRUD routes. Just open the **"routes/web.php"** file and add the following routes.

```
<?php
```

```
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\ProductController;
```

```
/*
```

```
|-----
```

| Web Routes

|-----

|

| Here is where you can register web routes for your application.
These

| routes are loaded by the RouteServiceProvider within a group
which

| contains the "web" middleware group. Now create something
great!

|

*/

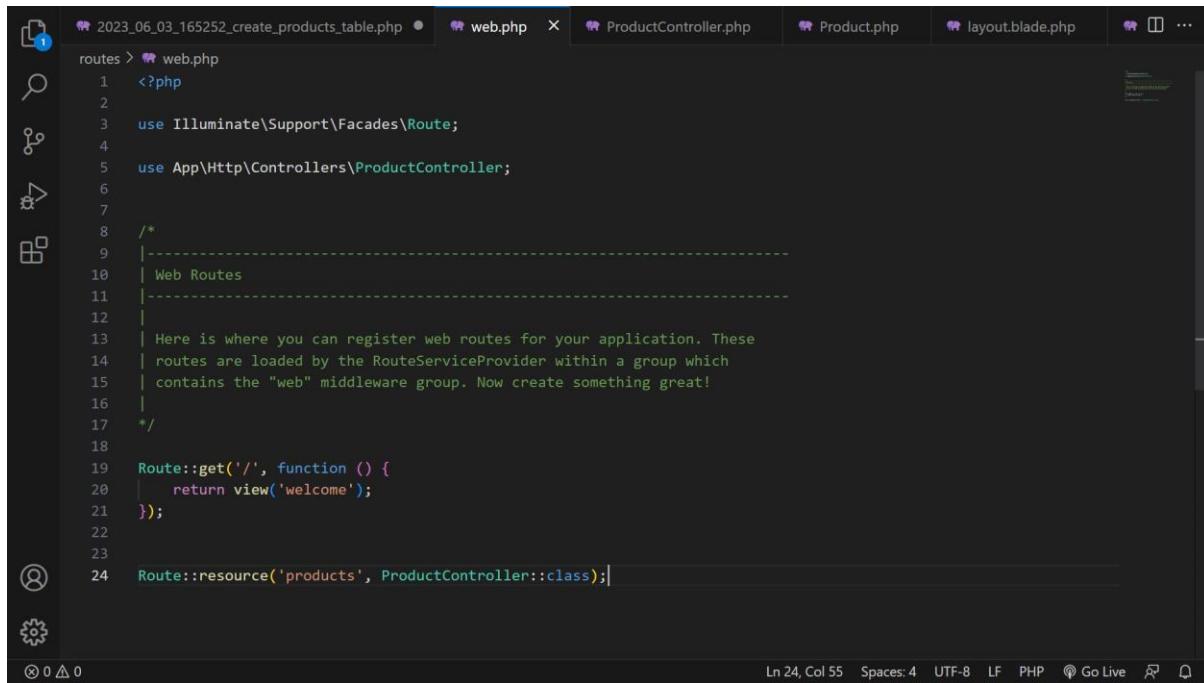
```
Route::get('/', function () {
```

```
    return view('welcome');
```

```
});
```

```
Route::resource('products', ProductController::class);
```

Screenshot of code Web.php:



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\ProductController;
6
7
8  /*
9  |
10 | Web Routes
11 |
12 |
13 | Here is where you can register web routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | contains the "web" middleware group. Now create something great!
16 |
17 */
18
19 Route::get('/', function () {
20     return view('welcome');
21 });
22
23
24 Route::resource('products', ProductController::class);
```

Ln 24, Col 55 Spaces: 4 UTF-8 LF PHP ⌂ Go Live ⌂ ⌂

Step 5: Create Controllers and Model

Now let's create our UserController just run the following command below:

```
php artisan make:controller ProductController --resource --model=Product
```

After running the command above you will find the generated controller to your **"App/Http/Controllers"**. Now let's add our methods for our Laravel crud operation example. Take note that we are using users to do this with a simple example.

In your **ProductController** you will have these methods:

- 1. index()**
- 2. create()**
- 3. store()**
- 4. show()**
- 5. edit()**
- 6. update()**
- 7. destroy()**

ProductController.php

Code:

```
<?php

namespace App\Http\Controllers;

use App\Models\Product;

use Illuminate\Http\Request;

class ProductController extends Controller

{

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */

    public function index()

    {

        $products = Product::latest()->paginate(5);

        return view('products.index',compact('products'))

            ->with('i', (request()->input('page', 1) - 1) * 5);

    }

    /**
     * Show the form for creating a new resource.
     *

```

```
* @return \Illuminate\Http\Response
*/
public function create()
{
    return view('products.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);
}
```

Product::create(\$request->all());

return redirect()->route('products.index')

```
        ->with('success','Product created successfully.');
```

```
}
```

```
/**
```

```
 * Display the specified resource.
```

```
 *
```

```
 * @param \App\Product $product
```

```
 * @return \Illuminate\Http\Response
```

```
 */
```

```
public function show(Product $product)
```

```
{
```

```
    return view('products.show',compact('product'));
```

```
}
```

```
/**
```

```
 * Show the form for editing the specified resource.
```

```
 *
```

```
 * @param \App\Product $product
```

```
 * @return \Illuminate\Http\Response
```

```
 */
```

```
public function edit(Product $product)
```

```
{
```

```
        return view('products.edit',compact('product'));

    }

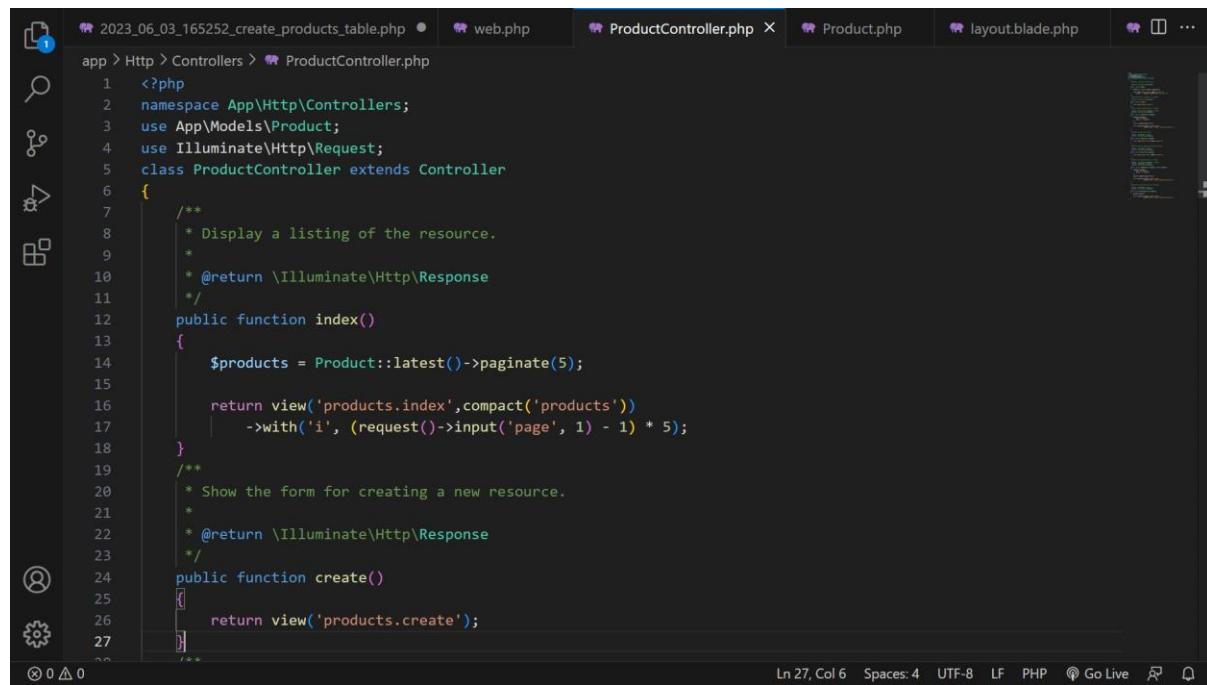
/** 
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Product $product
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, Product $product)
{
    $request->validate([
        'name' => 'required',
        'detail' => 'required',
    ]);

    $product->update($request->all());

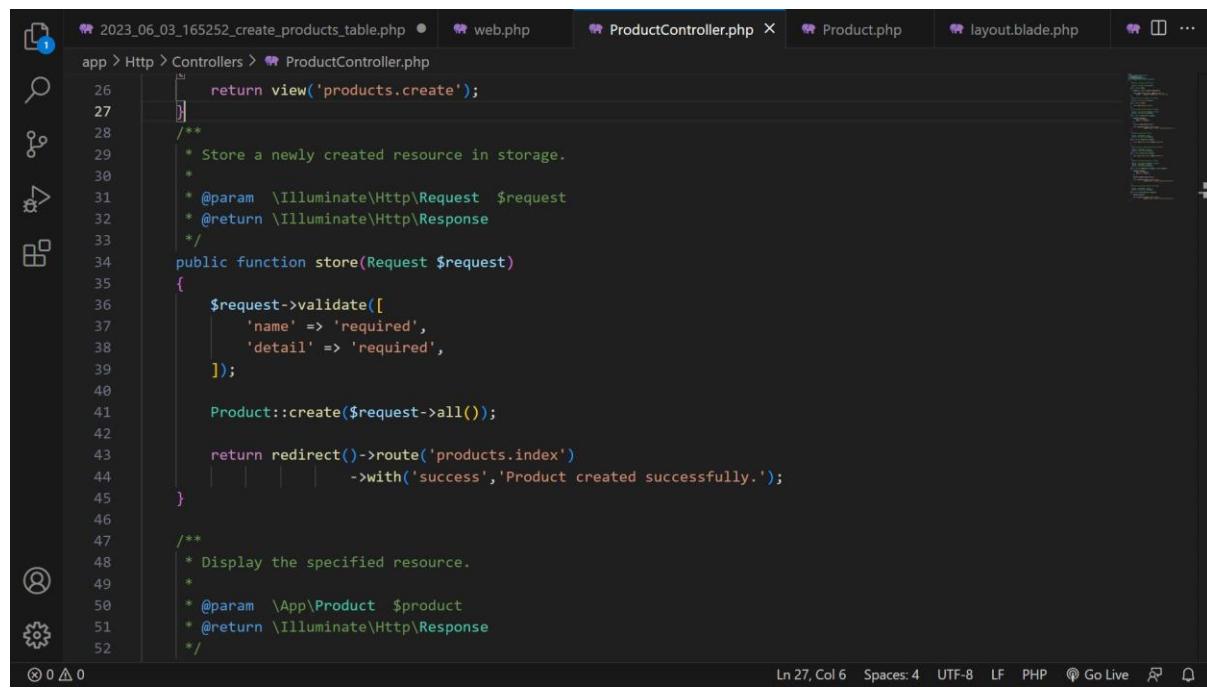
    return redirect()->route('products.index')
        ->with('success','Product updated successfully');
}
```

```
/**  
 * Remove the specified resource from storage.  
 *  
 * @param \App\Product $product  
 * @return \Illuminate\Http\Response  
 */  
public function destroy(Product $product)  
{  
    $product->delete();  
  
    return redirect()->route('products.index')  
        ->with('success','Product deleted successfully');  
}  
}
```

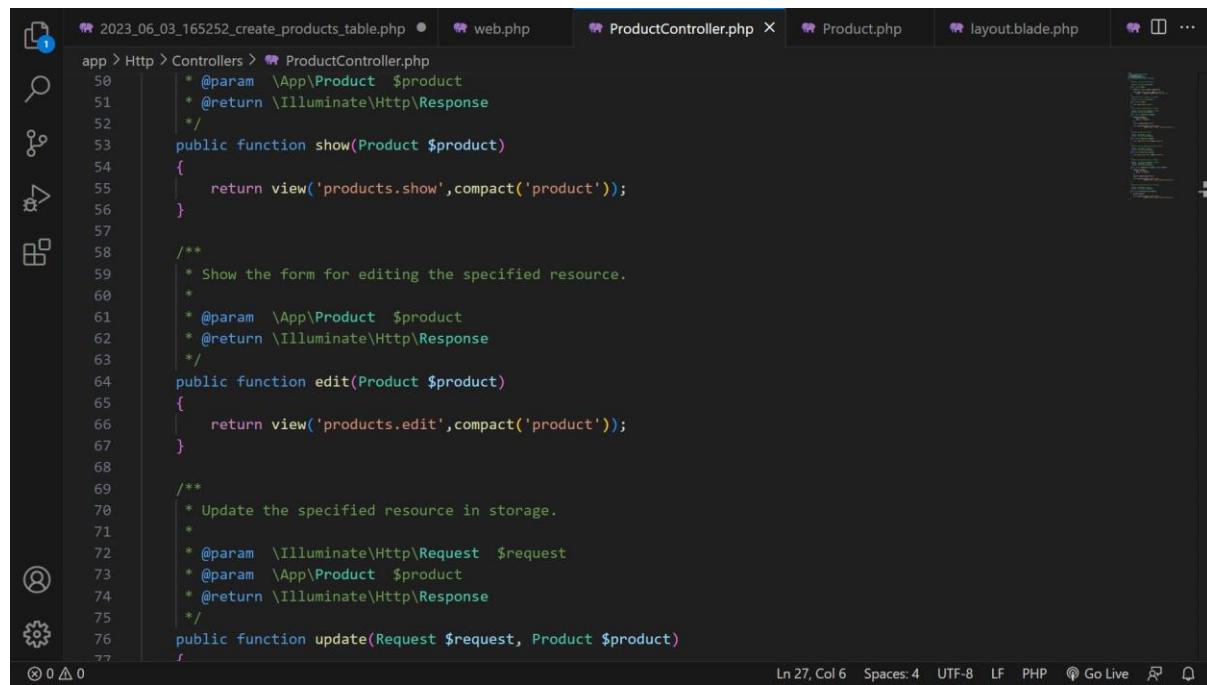
Screenshot of code:



```
2023_06_03_165252_create_products_table.php ● web.php ✎ ProductController.php ✎ Product.php ✎ layout.blade.php ✎ ...  
app > Http > Controllers > ProductController.php  
1 <?php  
2 namespace App\Http\Controllers;  
3 use App\Models\Product;  
4 use Illuminate\Http\Request;  
5 class ProductController extends Controller  
6 {  
7     /**  
8      * Display a listing of the resource.  
9      *  
10     * @return \Illuminate\Http\Response  
11     */  
12    public function index()  
13    {  
14        $products = Product::latest()->paginate(5);  
15  
16        return view('products.index',compact('products'))  
17            ->with('i', (request()->input('page', 1) - 1) * 5);  
18    }  
19    /**  
20     * Show the form for creating a new resource.  
21     *  
22     * @return \Illuminate\Http\Response  
23     */  
24    public function create()  
25    {  
26        return view('products.create');  
27    }  
28 }  
Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂
```



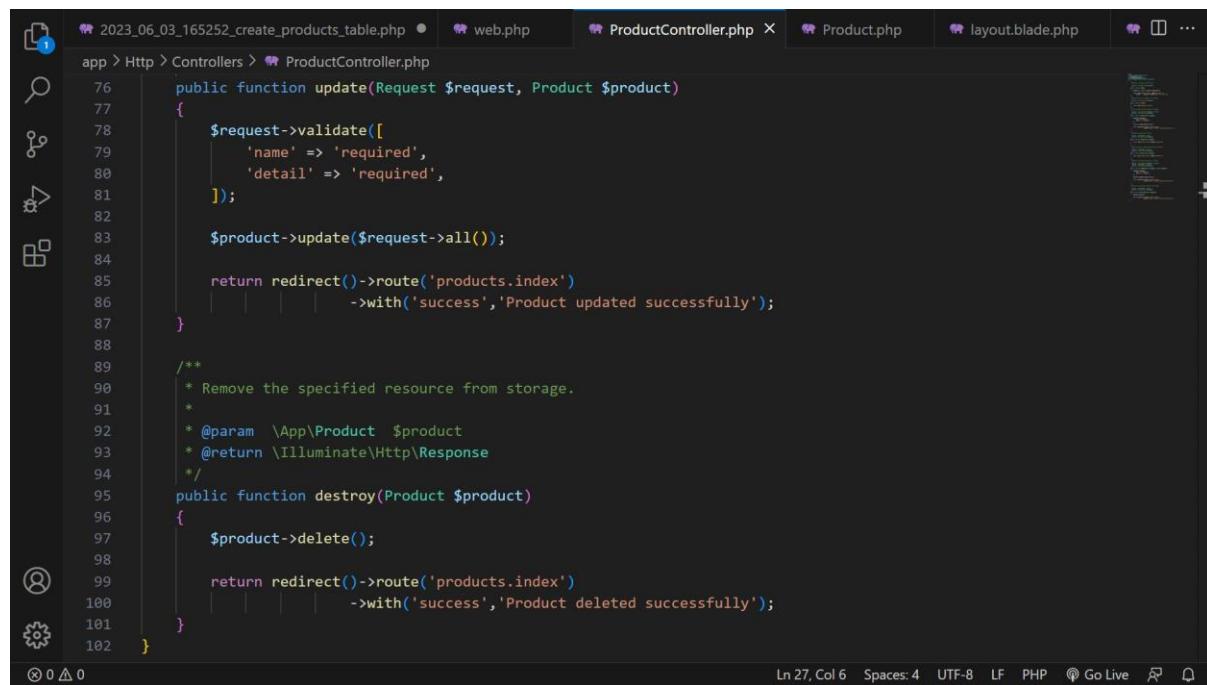
```
2023_06_03_165252_create_products_table.php ● web.php ✎ ProductController.php ✎ Product.php ✎ layout.blade.php ✎ ...  
app > Http > Controllers > ProductController.php  
26     return view('products.create');  
27 }  
28 /**  
29 * Store a newly created resource in storage.  
30 *  
31 * @param \Illuminate\Http\Request $request  
32 * @return \Illuminate\Http\Response  
33 */  
34 public function store(Request $request)  
35 {  
36     $request->validate([  
37         'name' => 'required',  
38         'detail' => 'required',  
39     ]);  
40  
41     Product::create($request->all());  
42  
43     return redirect()->route('products.index')  
44         ->with('success','Product created successfully.');//  
45 }  
46  
47 /**  
48 * Display the specified resource.  
49 *  
50 * @param \App\Product $product  
51 * @return \Illuminate\Http\Response  
52 */  
53 }  
Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂
```



app > Http > Controllers > ProductController.php

```
50  * @param \App\Product $product
51  * @return \Illuminate\Http\Response
52  */
53  public function show(Product $product)
54  {
55      return view('products.show',compact('product'));
56  }
57
58  /**
59  * Show the form for editing the specified resource.
60  *
61  * @param \App\Product $product
62  * @return \Illuminate\Http\Response
63  */
64  public function edit(Product $product)
65  {
66      return view('products.edit',compact('product'));
67  }
68
69  /**
70  * Update the specified resource in storage.
71  *
72  * @param \Illuminate\Http\Request $request
73  * @param \App\Product $product
74  * @return \Illuminate\Http\Response
75  */
76  public function update(Request $request, Product $product)
77  {
78      $request->validate([
79          'name' => 'required',
80          'detail' => 'required',
81      ]);
82
83      $product->update($request->all());
84
85      return redirect()->route('products.index')
86          ->with('success','Product updated successfully');
87  }
88
89  /**
90  * Remove the specified resource from storage.
91  *
92  * @param \App\Product $product
93  * @return \Illuminate\Http\Response
94  */
95  public function destroy(Product $product)
96  {
97      $product->delete();
98
99      return redirect()->route('products.index')
100         ->with('success','Product deleted successfully');
101 }
```

Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂



app > Http > Controllers > ProductController.php

```
76  public function update(Request $request, Product $product)
77  {
78      $request->validate([
79          'name' => 'required',
80          'detail' => 'required',
81      ]);
82
83      $product->update($request->all());
84
85      return redirect()->route('products.index')
86          ->with('success','Product updated successfully');
87  }
88
89  /**
90  * Remove the specified resource from storage.
91  *
92  * @param \App\Product $product
93  * @return \Illuminate\Http\Response
94  */
95  public function destroy(Product $product)
96  {
97      $product->delete();
98
99      return redirect()->route('products.index')
100         ->with('success','Product deleted successfully');
101 }
```

Ln 27, Col 6 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⌂ ⌂

Product.php

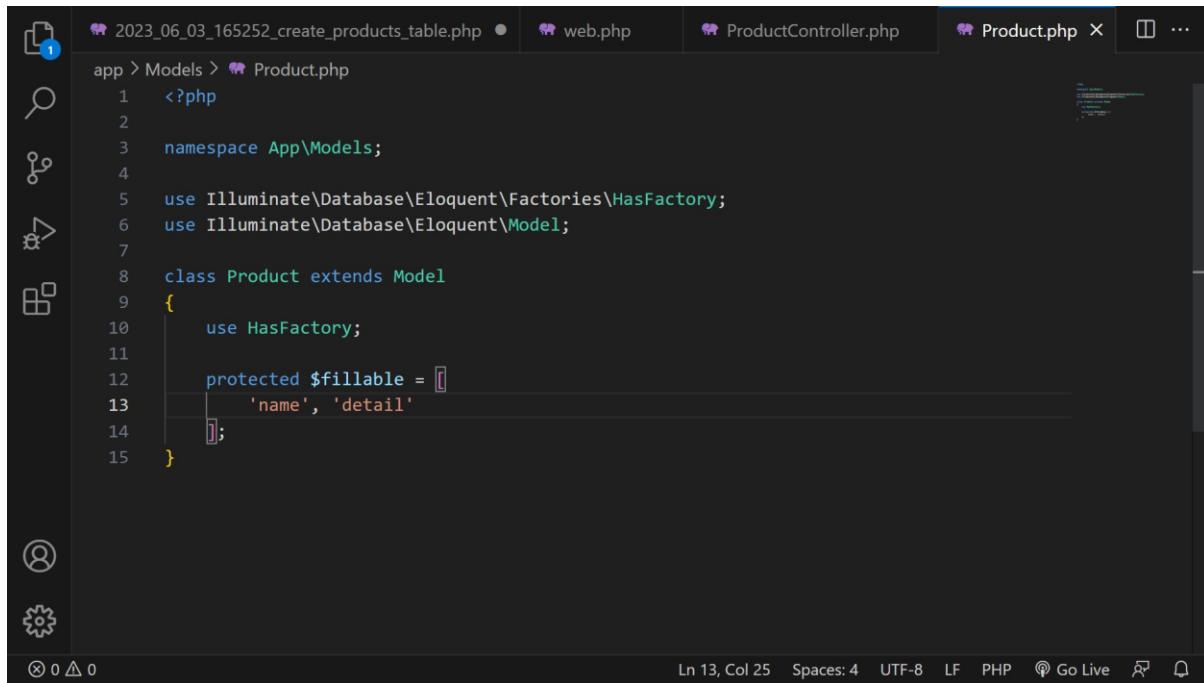
Code:

```
<?php
```

namespace App\Models;

```
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
  
class Product extends Model  
{  
    use HasFactory;  
  
    protected $fillable = [  
        'name', 'detail'  
    ];  
}
```

Screenshot of code:



```
2023_06_03_165252_create_products_table.php ● web.php ProductController.php Product.php X ...  
app > Models > Product.php  
1 <?php  
2  
3 namespace App\Models;  
4  
5 use Illuminate\Database\Eloquent\Factories\HasFactory;  
6 use Illuminate\Database\Eloquent\Model;  
7  
8 class Product extends Model  
9 {  
10     use HasFactory;  
11  
12     protected $fillable = [  
13         'name', 'detail'  
14     ];  
15 }  
  
Ln 13, Col 25 Spaces: 4 UTF-8 LF PHP ⚡ Go Live 🔍
```

6. Create all blade.php file and write code

Layout.blade.php Code:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
    <title>CRUD Application</title>  
  
    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/css/bootstrap.css" rel="stylesheet">
```

```
</head>
```

```
<body>
```

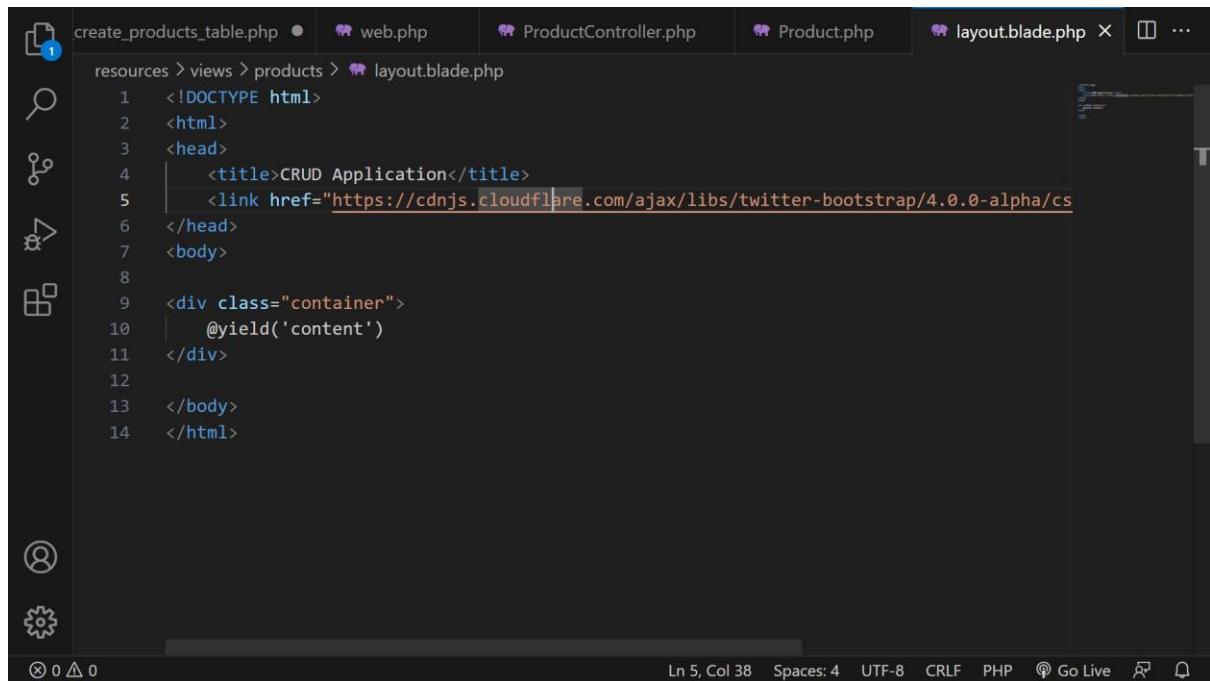
```
<div class="container">
```

```
    @yield('content')
```

```
</div>
```

```
</body>  
</html>
```

Screenshot of code:



```
resources > views > products > layout.blade.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>CRUD Application</title>
5  |   <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0-alpha/cs
6  </head>
7  <body>
8
9  <div class="container">
10 |   @yield('content')
11 </div>
12
13 </body>
14 </html>
```

index.blade.php Code:

```
@extends('products.layout')  
  
@section('content')  
    <div class="row">  
        <div class="col-lg-12 margin-tb">  
            <div class="pull-left">  
                <h2>CRUD Application</h2>  
            </div>  
            <div class="pull-right">
```

```
        <a class="btn btn-success" href="{{ route('products.create') }}> Create New Product</a>
    </div>
</div>
</div>
```

```
@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif
```

```
<table class="table table-bordered">
    <tr>
        <th>No</th>
        <th>Name</th>
        <th>Details</th>
        <th width="280px">Action</th>
    </tr>
    @foreach ($products as $product)
        <tr>
            <td>{{ ++$i }}</td>
            <td>{{ $product->name }}</td>
            <td>{{ $product->detail }}</td>
```

```

<td>

    <form action="{{ route('products.destroy', $product->id) }}"
method="POST">

        <a class="btn btn-info" href="{{
route('products.show', $product->id) }}>Show</a>

        <a class="btn btn-primary" href="{{
route('products.edit', $product->id) }}>Edit</a>

        @csrf
        @method('DELETE')

        <button type="submit" class="btn btn-
danger">Delete</button>
    </form>
</td>
</tr>
@endforeach
</table>

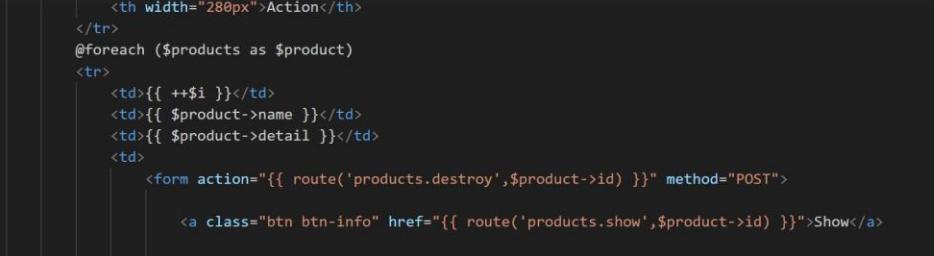
{!! $products->links() !!}

```

@endsection

Screenshot of code:

```
resources > views > products > index.blade.php
  1  @extends('products.layout')
  2
  3  @section('content')
  4      <div class="row">
  5          <div class="col-lg-12 margin-tb">
  6              <div class="pull-left">
  7                  |  <h2>CRUD Application</h2>
  8              </div>
  9              <div class="pull-right">
 10                  |  <a class="btn btn-success" href="{{ route('products.create') }}> Create New Product</a>
 11              </div>
 12          </div>
 13      </div>
 14
 15      @if ($message = Session::get('success'))
 16          <div class="alert alert-success">
 17              |  <p>{{ $message }}</p>
 18          </div>
 19      @endif
 20
 21      <table class="table table-bordered">
 22          <tr>
 23              <th>No</th>
 24              <th>Name</th>
 25              <th>Details</th>
 26              <th width="280px">Action</th>
 27          </tr>
 28      </table>
```



```
web.php ProductController.php Product.php layout.blade.php index.blade.php .env create.blade.php ...  
resources > views > products > index.blade.php  
26 | <th width="280px">Action</th>  
27 | </tr>  
28 | @foreach ($products as $product)  
29 | <tr>  
30 | <td>{{ ++$i }}</td>  
31 | <td>{{ $product->name }}</td>  
32 | <td>{{ $product->detail }}</td>  
33 | <td>  
34 | <form action="{{ route('products.destroy', $product->id) }}" method="POST">  
35 | <a class="btn btn-info" href="{{ route('products.show', $product->id) }}>Show</a>  
36 | <a class="btn btn-primary" href="{{ route('products.edit', $product->id) }}>Edit</a>  
37 | @csrf  
38 | @method('DELETE')  
39 | <button type="submit" class="btn btn-danger">Delete</button>  
40 | </form>  
41 | </td>  
42 | </tr>  
43 | @endforeach  
44 | </table>  
45 |  
46 |  
47 |  
48 |  
49 |  
50 | {!! $products->links() !!}  
51 |  
52 | @endsection  
Ln 7, Col 37 Spaces: 4 UTF-8 CRLF PHP Go Live
```

create.blade.php Code:

```
@extends('products.layout')
```

```
@section('content')
```

```
<div class="row">
  <div class="col-lg-12 margin-tb">
    <div class="pull-left">
      <h2>Add New Product</h2>
    </div>
    <div class="pull-right">
      <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
    </div>
  </div>
</div>

@if ($errors->any())
  <div class="alert alert-danger">
    <strong>Whoops!</strong> There were some problems with
    your input.<br><br>
    <ul>
      @foreach ($errors->all() as $error)
        <li>{{ $error }}</li>
      @endforeach
    </ul>
  </div>
@endif
```

```
<form action="{{ route('products.store') }}" method="POST">

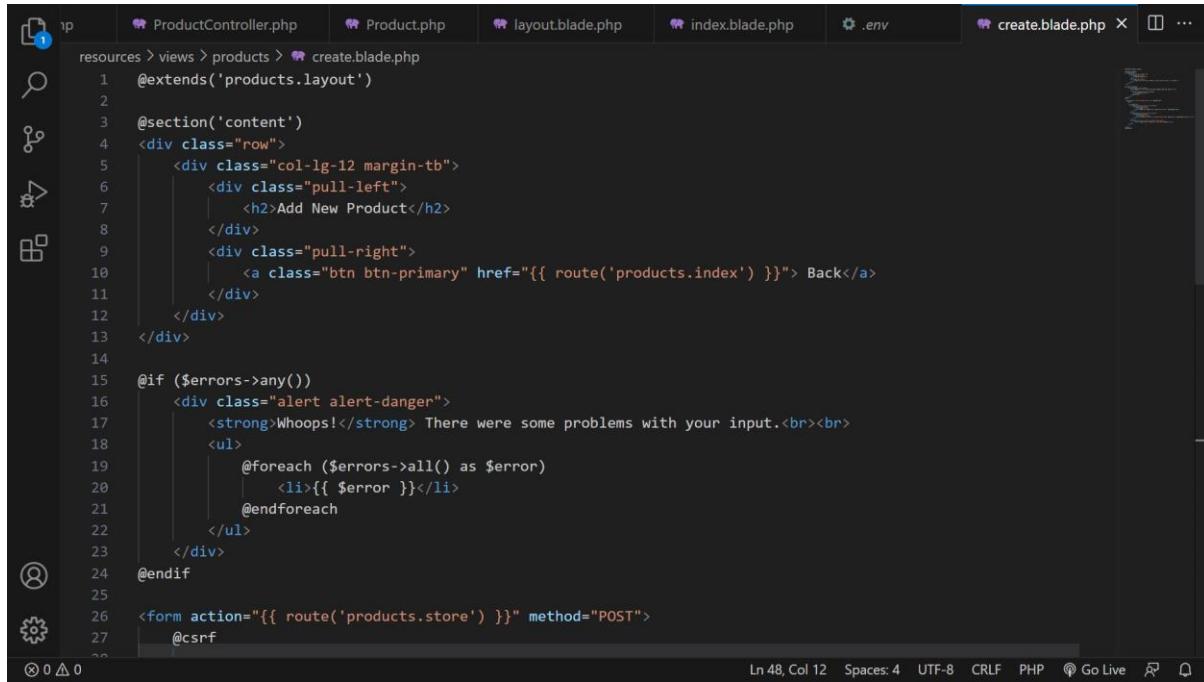
    @csrf

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Name:</strong>
                <input type="text" name="name" class="form-control" placeholder="Name">
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Detail:</strong>
                <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
            <button type="submit" class="btn btn-primary">Submit</button>
        </div>
    </div>

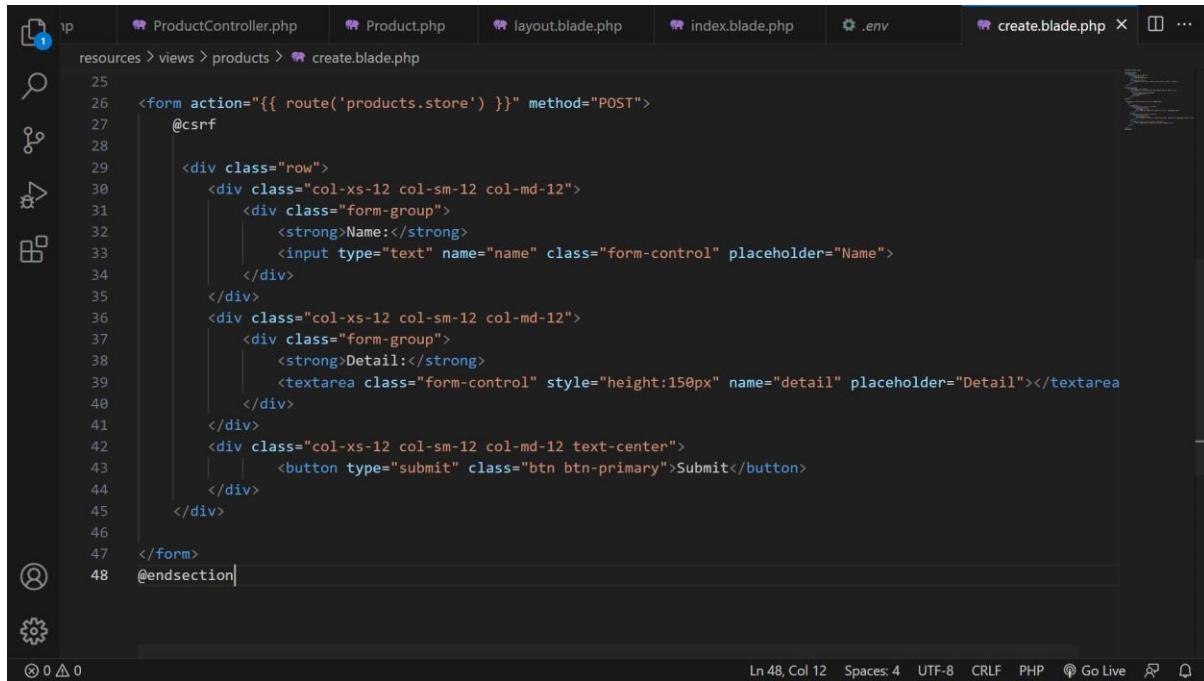
</form>
```

```
@endsection
```

Screenshot of code:



```
resources > views > products > create.blade.php
1  @extends('products.layout')
2
3  @section('content')
4      <div class="row">
5          <div class="col-lg-12 margin-tb">
6              <div class="pull-left">
7                  <h2>Add New Product</h2>
8              </div>
9              <div class="pull-right">
10                 <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
11             </div>
12         </div>
13     </div>
14
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Whoops!</strong> There were some problems with your input.<br><br>
18             <ul>
19                 @foreach ($errors->all() as $error)
20                     <li>{{ $error }}</li>
21                 @endforeach
22             </ul>
23         </div>
24     @endif
25
26     <form action="{{ route('products.store') }}" method="POST">
27         @csrf
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```



```
resources > views > products > create.blade.php
25
26     <form action="{{ route('products.store') }}" method="POST">
27         @csrf
28
29         <div class="row">
30             <div class="col-xs-12 col-sm-12 col-md-12">
31                 <div class="form-group">
32                     <strong>Name:</strong>
33                     <input type="text" name="name" class="form-control" placeholder="Name">
34                 </div>
35             </div>
36             <div class="col-xs-12 col-sm-12 col-md-12">
37                 <div class="form-group">
38                     <strong>Detail:</strong>
39                     <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail"></textarea>
40                 </div>
41             </div>
42             <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43                 <button type="submit" class="btn btn-primary">Submit</button>
44             </div>
45         </div>
46
47     </form>
48     @endsection
```

edit.blade.php Code:

```
@extends('products.layout')
```

```
@section('content')

<div class="row">

    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2>Edit Product</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
        </div>
    </div>
</div>

@if ($errors->any())
    <div class="alert alert-danger">
        <strong>Whoops!</strong> There were some problems with
        your input.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>

```

```
@endif

<form action="{{ route('products.update',$product->id) }}"
method="POST">

    @csrf

    @method('PUT')


<div class="row">
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Name:</strong>
            <input type="text" name="name" value="{{ $product->name }}" class="form-control" placeholder="Name">
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12">
        <div class="form-group">
            <strong>Detail:</strong>
            <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail">{{ $product->detail }}</textarea>
        </div>
    </div>
    <div class="col-xs-12 col-sm-12 col-md-12 text-center">
```

```
        <button type="submit" class="btn btn-primary">Submit</button>
```

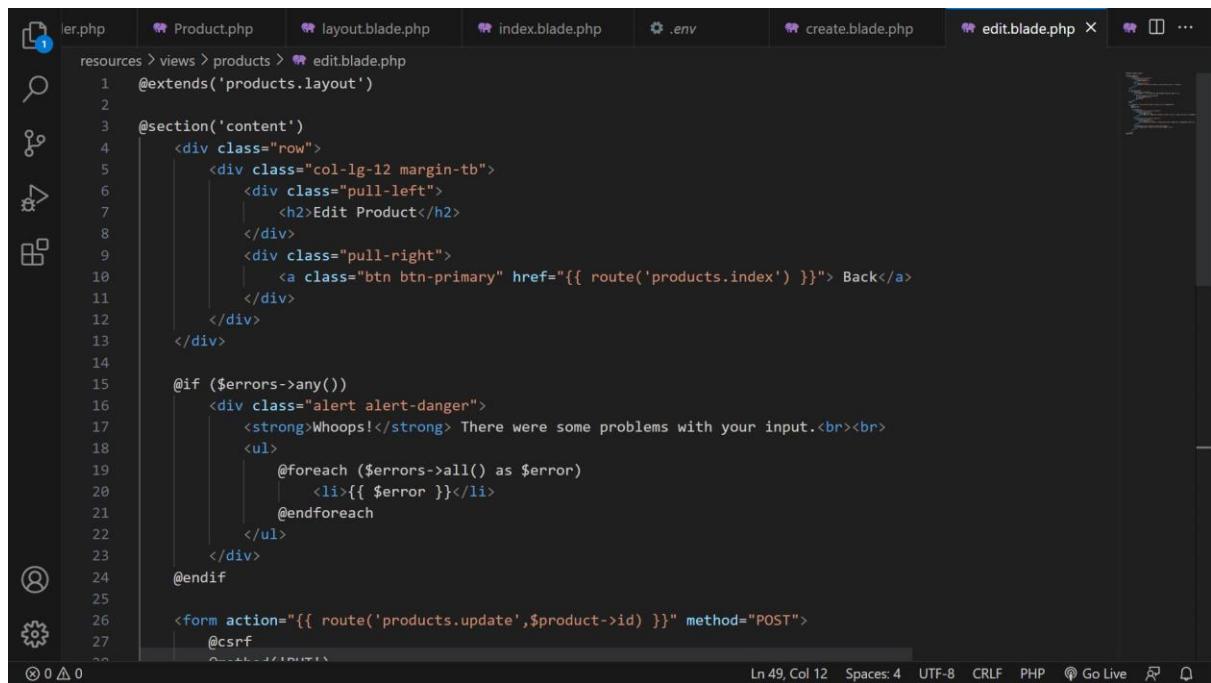
```
</div>
```

```
</div>
```

```
</form>
```

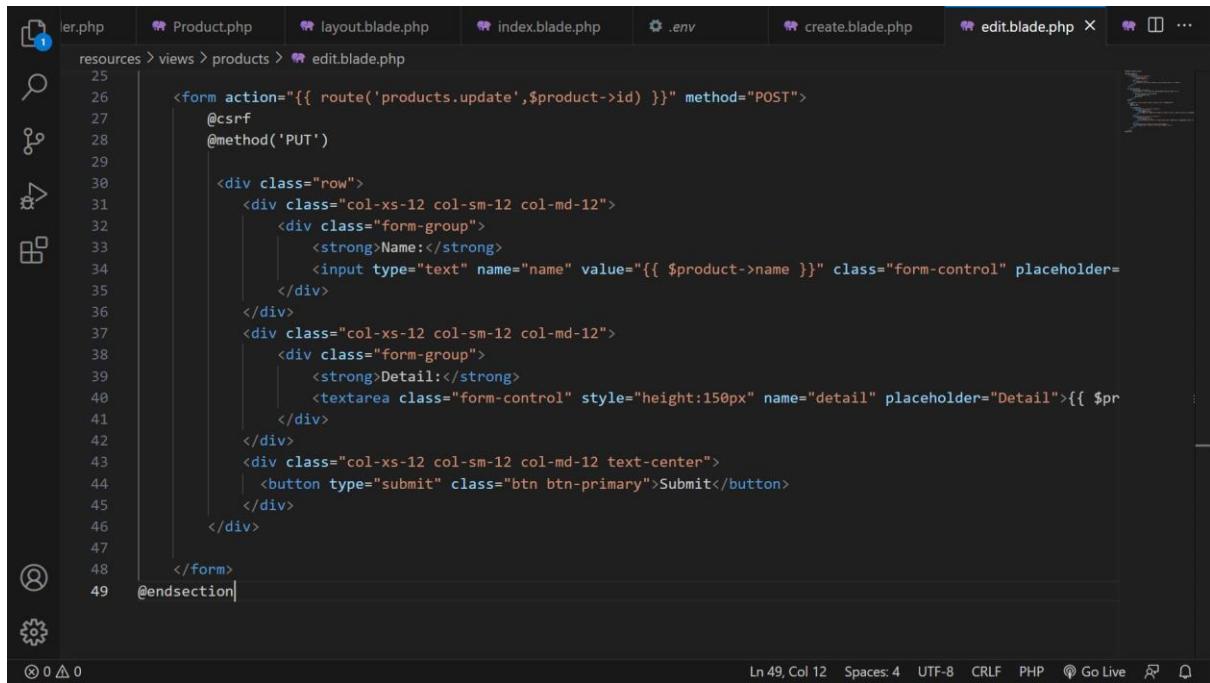
```
@endsection
```

Screenshot of code:



The screenshot shows a code editor with a dark theme, displaying a Laravel blade template file named `edit.blade.php`. The file is located in the `resources > views > products > edit.blade.php` directory. The code includes sections for content, error handling, and a form for product update.

```
resources > views > products > edit.blade.php
1  @extends('products.layout')
2
3  @section('content')
4      <div class="row">
5          <div class="col-lg-12 margin-tb">
6              <div class="pull-left">
7                  <h2>Edit Product</h2>
8              </div>
9              <div class="pull-right">
10                 <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
11             </div>
12         </div>
13     </div>
14
15     @if ($errors->any())
16         <div class="alert alert-danger">
17             <strong>Whoops!</strong> There were some problems with your input.<br><br>
18             <ul>
19                 @foreach ($errors->all() as $error)
20                     <li>{{ $error }}</li>
21                 @endforeach
22             </ul>
23         </div>
24     @endif
25
26     <form action="{{ route('products.update', $product->id) }}" method="POST">
27         @csrf
28         <div class="form-group">
```



```
resources > views > products > edit.blade.php
25  <form action="{{ route('products.update', $product->id) }}" method="POST">
26      @csrf
27      @method('PUT')
28
29      <div class="row">
30          <div class="col-xs-12 col-sm-12 col-md-12">
31              <div class="form-group">
32                  <strong>Name:</strong>
33                  <input type="text" name="name" value="{{ $product->name }}" class="form-control" placeholder="Name">
34              </div>
35          </div>
36          <div class="col-xs-12 col-sm-12 col-md-12">
37              <div class="form-group">
38                  <strong>Detail:</strong>
39                  <textarea class="form-control" style="height:150px" name="detail" placeholder="Detail">{{ $product->detail }}</textarea>
40              </div>
41          </div>
42          <div class="col-xs-12 col-sm-12 col-md-12 text-center">
43              <button type="submit" class="btn btn-primary">Submit</button>
44          </div>
45      </div>
46  </form>
47
48  @endsection|
```

show.blade.php Code:

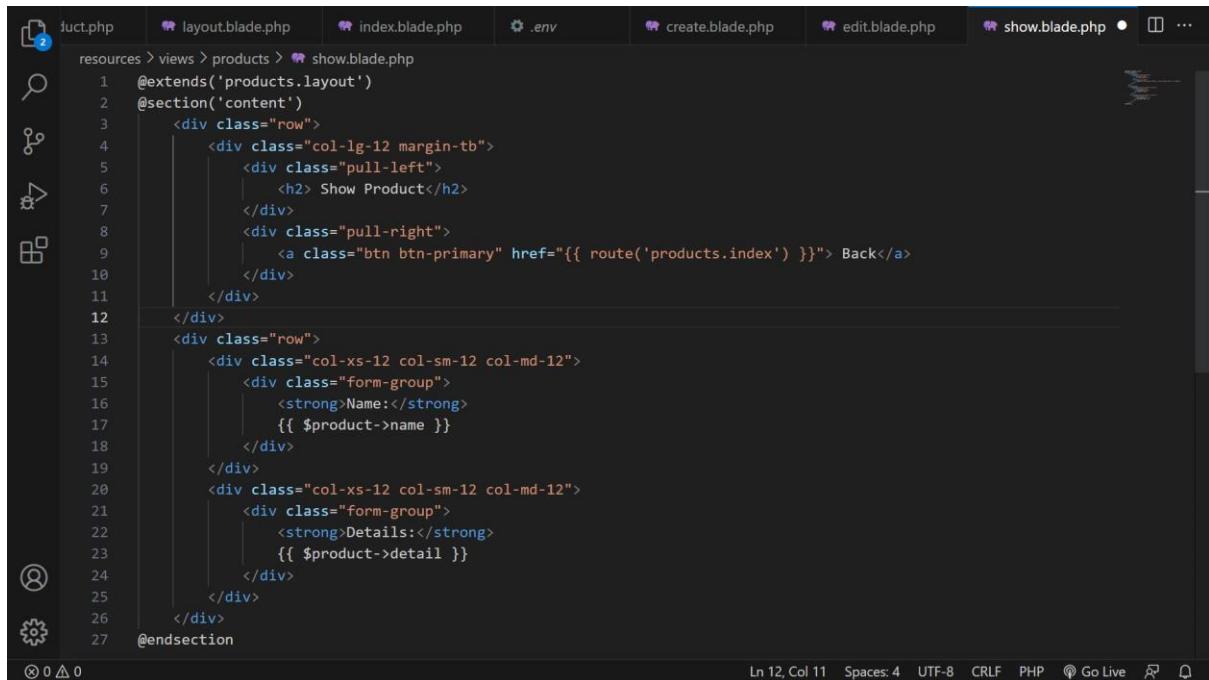
```
@extends('products.layout')

@section('content')

<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2> Show Product</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>
        </div>
    </div>
</div>
```

```
<div class="row">
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Name:</strong>
      {{ $product->name }}
    </div>
  </div>
  <div class="col-xs-12 col-sm-12 col-md-12">
    <div class="form-group">
      <strong>Details:</strong>
      {{ $product->detail }}
    </div>
  </div>
</div>
@endsection
```

Screenshot of code:



```
layout.php layout.blade.php index.blade.php .env create.blade.php edit.blade.php show.blade.php ...  
resources > views > products > show.blade.php  
1  @extends('products.layout')  
2  @section('content')  
3      <div class="row">  
4          <div class="col-lg-12 margin-tb">  
5              <div class="pull-left">  
6                  <h2> Show Product</h2>  
7              </div>  
8              <div class="pull-right">  
9                  <a class="btn btn-primary" href="{{ route('products.index') }}> Back</a>  
10             </div>  
11         </div>  
12     </div>  
13     <div class="row">  
14         <div class="col-xs-12 col-sm-12 col-md-12">  
15             <div class="form-group">  
16                 <strong>Name:</strong>  
17                 {{ $product->name }}  
18             </div>  
19         </div>  
20         <div class="col-xs-12 col-sm-12 col-md-12">  
21             <div class="form-group">  
22                 <strong>Details:</strong>  
23                 {{ $product->detail }}  
24             </div>  
25         </div>  
26     </div>  
27     @endsection  
Ln 12, Col 11  Spaces: 4  UTF-8  CRLF  PHP  ⌂ Go Live  ⌂  ⌂
```

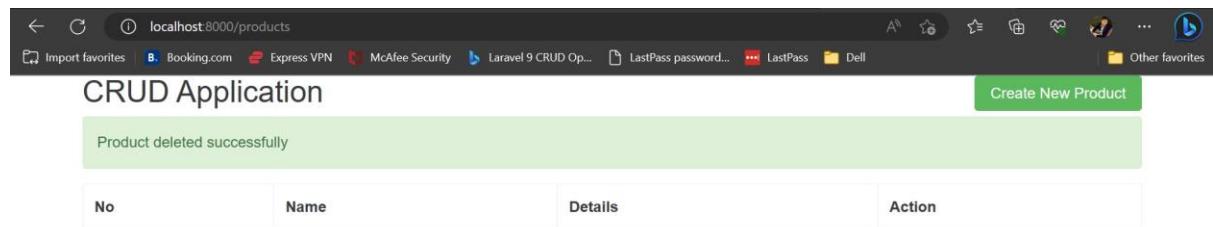
**Step 7: Then final step to run the last command
“php artisan serve”**

Step 8: Then go to browser and this code

127.0.0.1:8000/products

Screenshot

Output:



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

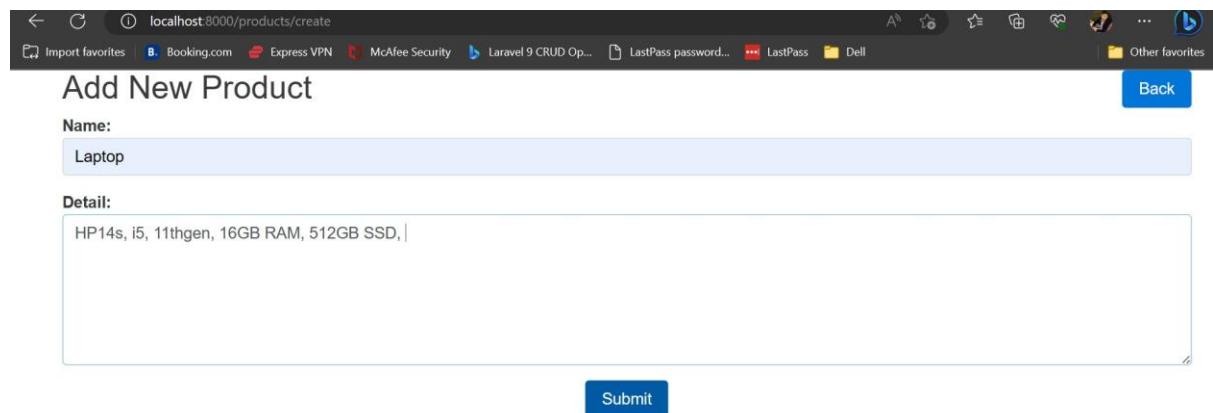
CRUD Application

Create New Product

Product deleted successfully

No	Name	Details	Action

Create Product Name and Details



localhost:8000/products/create

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

Add New Product

Back

Name: Laptop

Detail: HP14s, i5, 11thgen, 16GB RAM, 512GB SSD, |

Submit

Created

The screenshot shows a browser window with the URL `localhost:8000/products`. The title bar says "CRUD Application". A green success message box says "Product created successfully.". Below is a table with one row of data:

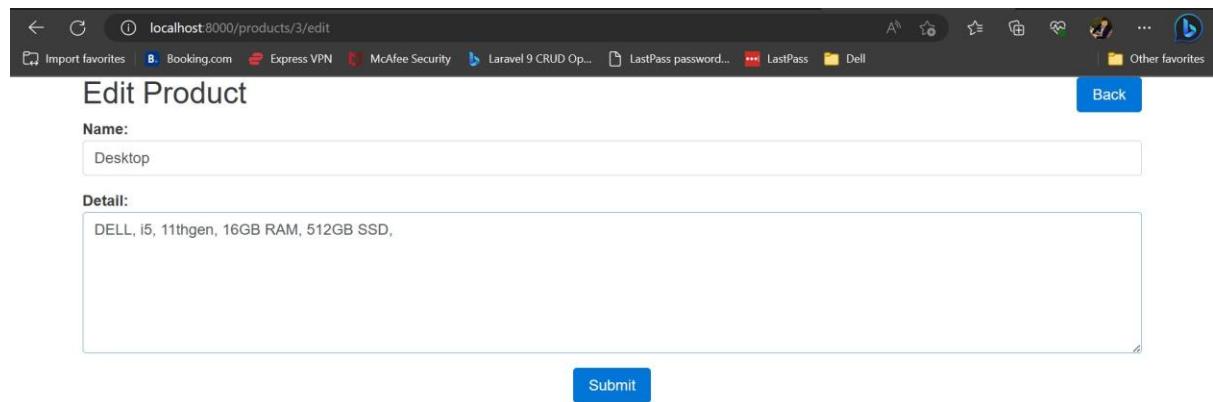
No	Name	Details	Action
1	Laptop	HP14s, i5, 11thgen, 16GB RAM, 512GB SSD,	Show Edit Delete

Show Product

The screenshot shows a browser window with the URL `localhost:8000/products/3`. The title bar says "Show Product". The page displays the product details for the product with ID 3:

Name: Laptop
Details: HP14s, i5, 11thgen, 16GB RAM, 512GB SSD,

Edit Product



localhost:8000/products/3/edit

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

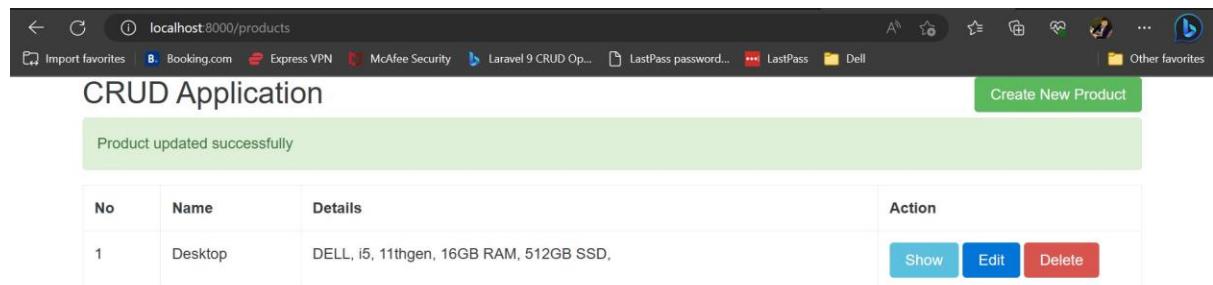
Edit Product

Name: Desktop

Detail: DELL, i5, 11thgen, 16GB RAM, 512GB SSD,

Submit

Edit successfully



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

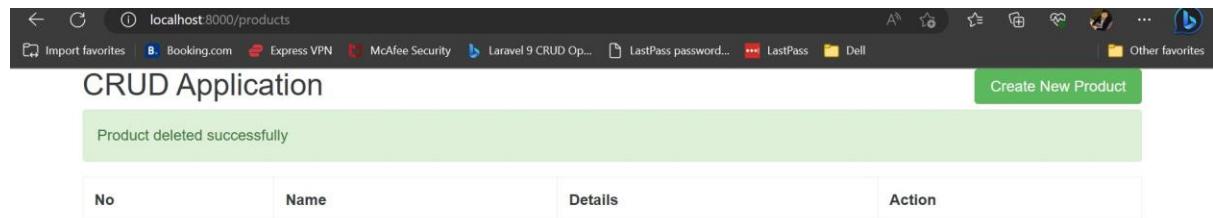
CRUD Application

Create New Product

Product updated successfully

No	Name	Details	Action
1	Desktop	DELL, i5, 11thgen, 16GB RAM, 512GB SSD,	Show Edit Delete

Delete



localhost:8000/products

Import favorites Booking.com Express VPN McAfee Security Laravel 9 CRUD Op... LastPass password... LastPass Dell Other favorites

CRUD Application

Create New Product

Product deleted successfully

No	Name	Details	Action
----	------	---------	--------

9 **Aim:** - Demonstrate csv file upload in Laravel, extract its data and insert into database.

Learning outcome: - Able to understand Laravel Framework in upload csv file in database.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:-

Open your Command prompt and Please enter a Laravel command

“Composer create-project --prefer-dist laravel/laravel cscproject”

Step 2: Create Database , Table and column

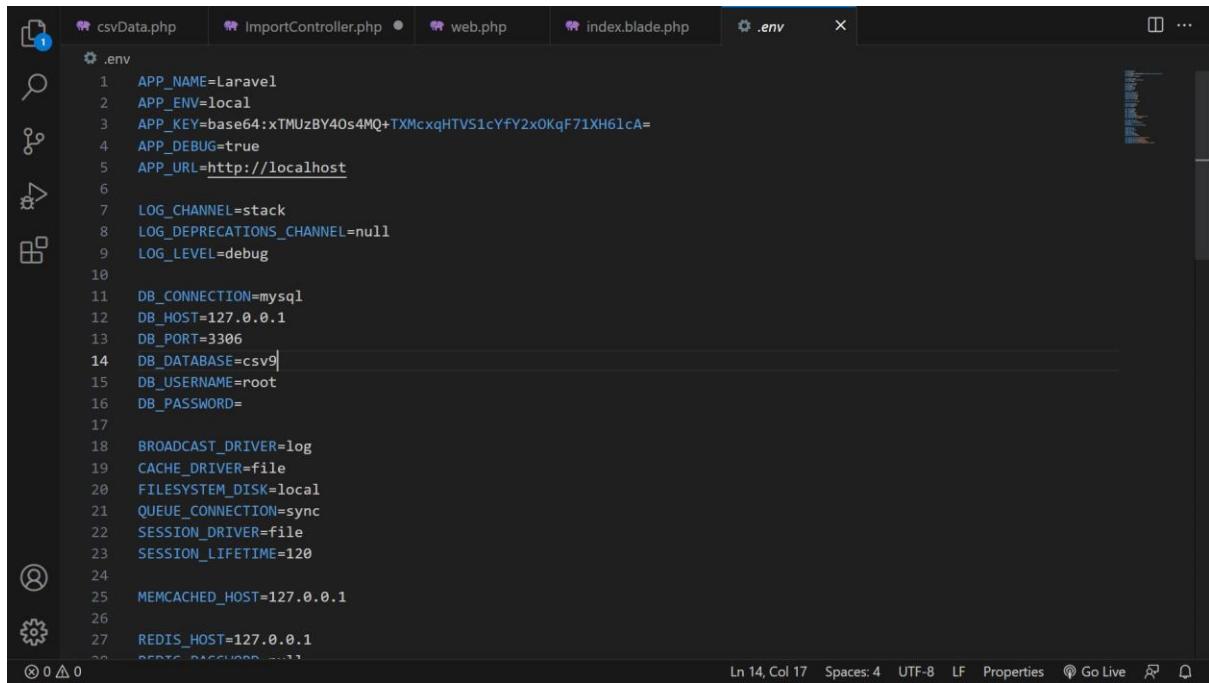
```
CREATE TABLE `csv_data` (`id` int(11) NOT NULL PRIMARY KEY  
AUTO_INCREMENT, `first_name` varchar(80) NOT NULL, `last_name`  
varchar(80) NOT NULL, `gender` varchar(20) NOT NULL, `email`  
varchar(80) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Step 3: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:xTMUzBY4Os4MQ+TMMcxqHTVS1cYfY2x0KqF71XH61cA=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=csv9
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=
```

Step 4: Migration Setup

Now, Open command prompt and run the following command to create the table into your added database:

Code:

“Php artisan migrate”

Step 4: Create one file Model name is csvData

In this step, you need to open new terminal and write code

Code:

“php artisan make: model csvData”

And open file write code

Code:

```
<?php

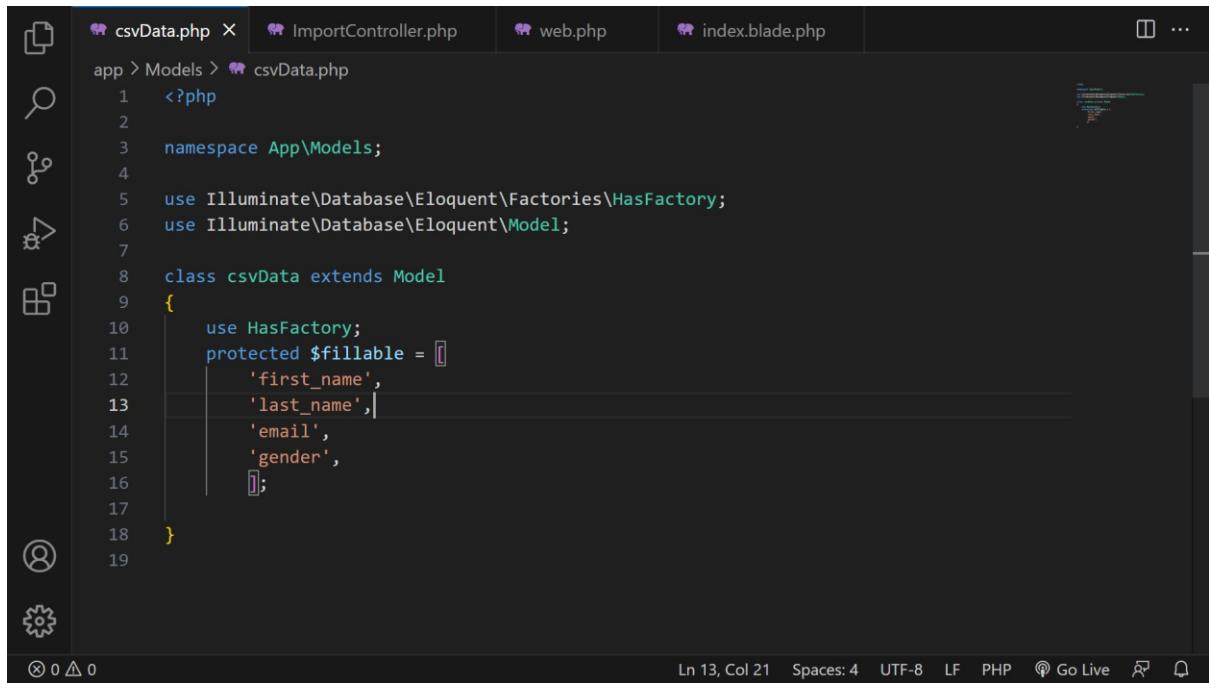
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class csvData extends Model
{
    use HasFactory;
    protected $fillable = [
        'first_name',
        'last_name',
        'email',
        'gender',
    ];
}

}
```

Screenshot of code:



```
csvData.php X ImportController.php web.php index.blade.php ...  
app > Models > csvData.php  
1  <?php  
2  
3  namespace App\Models;  
4  
5  use Illuminate\Database\Eloquent\Factories\HasFactory;  
6  use Illuminate\Database\Eloquent\Model;  
7  
8  class csvData extends Model  
9  {  
10     use HasFactory;  
11     protected $fillable = [  
12         'first_name',  
13         'last_name',  
14         'email',  
15         'gender',  
16     ];  
17 }  
18 }  
19  
Ln 13, Col 21  Spaces: 4  UTF-8  LF  PHP  ⚡ Go Live  ⚡  ⚡
```

Step 5: Create a Controller name **ImportController**

Code: “`php artisan make:controller ImportController`”

After write code in file

Code:

```
<?php
```

```
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
use App\Models\csvData;  
use Illuminate\Support\Facades\Session;  
  
class ImportController extends Controller
```

```
{  
    public function index(){  
        return view('index');  
    }  
    public function uploadCsv(Request $request){  
        if ($request->input('submit') != null ){  
            $file = $request->file('file');  
            // File Details  
            $filename = $file->getClientOriginalName();  
            $extension = $file->getClientOriginalExtension();  
            $tempPath = $file->getRealPath();  
            $fileSize = $file->getSize();  
            $mimeType = $file->getMimeType();  
            // Valid File Extensions  
            $valid_extension = array("csv");  
            // 2MB in Bytes  
            $maxFileSize = 2097152;  
  
            // Check file extension  
            if(in_array(strtolower($extension),$valid_extension)){  
                // Check file size  
                if($fileSize <= $maxFileSize){  
                    // File upload location  
                    $location = 'uploads';  
                }  
            }  
        }  
    }  
}
```

```
// Upload file
$file->move($location,$filename);

// Import CSV to Database
$filepath = public_path($location."/".$filename);

// Reading file
$file = fopen($filepath,"r");

$dataRow = true;

while (($data = fgetcsv($file, 4000, ",")) !== FALSE) {
    if (!$dataRow) {
        CsvData::create([
            "first_name" => $data['1'],
            "last_name" => $data['2'],
            "email" => $data['3'],
            "gender" => $data['4']
        ]);
    }
    $dataRow = false;
    Session::flash('message','Import Successful.');
}

fclose($file);

return redirect(url('/'));

}

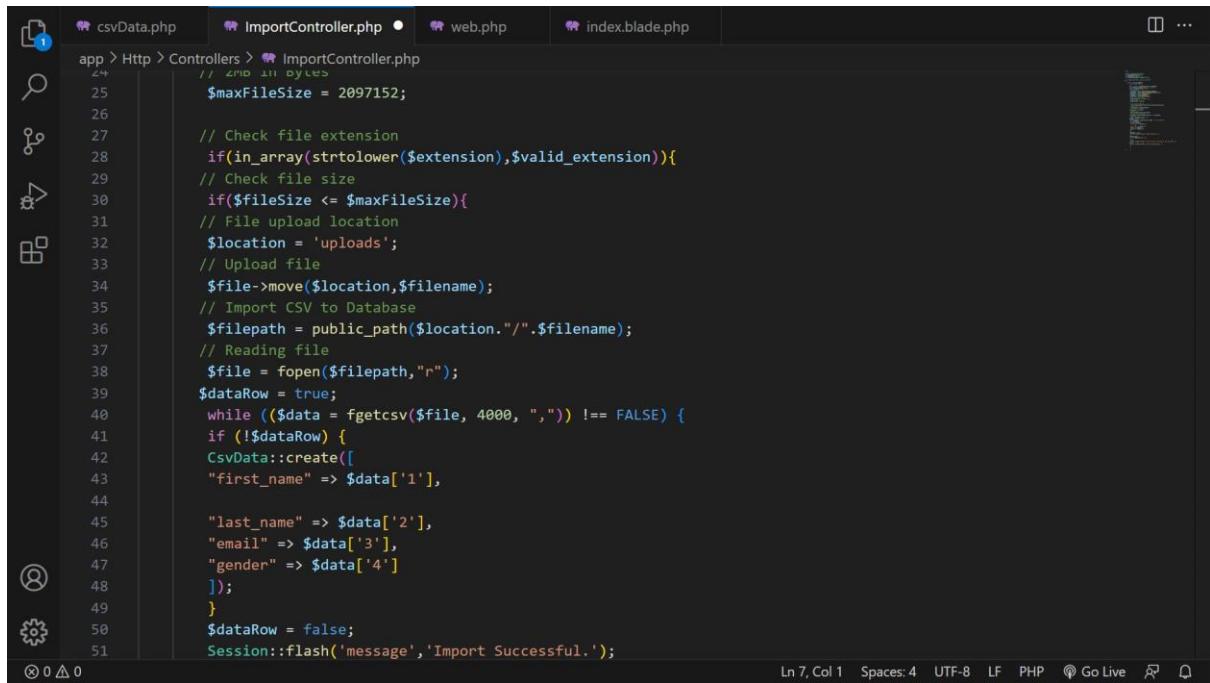
else{
```

```
Session::flash('message','File too large. File must be less than  
2MB.');?>}  
}  
Session::flash('message','Invalid File Extension.');?>}  
}  
}  
}  
}
```

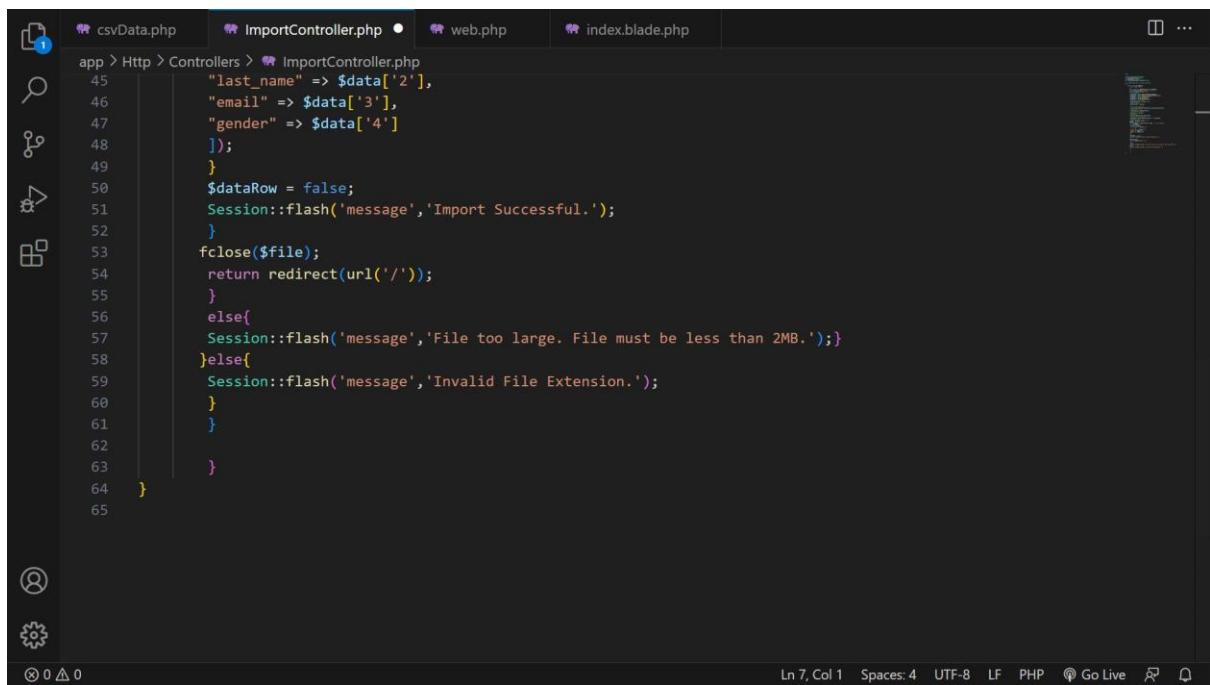
Screenshot of code:



```
app > Http > Controllers > ImportController.php
1  <?php
2
3  namespace App\Http\Controllers;
4  use Illuminate\Http\Request;
5  use App\Models\csvData;
6  use Illuminate\Support\Facades\Session;
7
8  class ImportController extends Controller
9  {
10     public function index(){
11         return view('index');
12     }
13     public function uploadCsv(Request $request){
14         if ($request->input('submit') != null ){
15             $file = $request->file('file');
16             // File Details
17             $filename = $file->getClientOriginalName();
18             $extension = $file->getClientOriginalExtension();
19             $tempPath = $file->getRealPath();
20             $fileSize = $file->getSize();
21             $mimeType = $file->getMimeType();
22             // Valid File Extensions
23             $valid_extension = array("csv");
24             // 2MB in Bytes
25             $maxFileSize = 2097152;
26
27             // Check file extension
28             if(in_array($extension, $valid_extension)) {
29                 Session::flash('success', 'File uploaded successfully');
30                 return redirect('index');
31             } else {
32                 Session::flash('error', 'Only CSV files are allowed');
33                 return redirect('index');
34             }
35         }
36     }
37 }
```



```
app > Http > Controllers > ImportController.php
25     // Limit in bytes
26     $maxFileSize = 2097152;
27
28     // Check file extension
29     if(in_array(strtolower($extension), $valid_extension)){
30         // Check file size
31         if($fileSize <= $maxFileSize){
32             // File upload location
33             $location = 'uploads';
34             // Upload file
35             $file->move($location, $filename);
36             // Import CSV to Database
37             $filepath = public_path($location."/".$filename);
38             // Reading file
39             $file = fopen($filepath, "r");
40             $dataRow = true;
41             while (( $data = fgetcsv($file, 4000, ",")) != FALSE) {
42                 if (!$dataRow) {
43                     CsvData::create([
44                         "first_name" => $data['1'],
45                         "last_name" => $data['2'],
46                         "email" => $data['3'],
47                         "gender" => $data['4']
48                     ]);
49                 }
50             }
51             $dataRow = false;
52             Session::flash('message', 'Import Successful.');
53         }
54     }
55
56     fclose($file);
57     return redirect(url('/'));
58 }
59 else{
60     Session::flash('message', 'File too large. File must be less than 2MB.');
61 }
62
63 }
64 }
```



```
55
56     fclose($file);
57     return redirect(url('/'));
58 }
59 else{
60     Session::flash('message', 'File too large. File must be less than 2MB.');
61 }
62
63 }
64 }
```

Step 6: Go to Routes

And write code

Code:

```
<?php
```

```
use Illuminate\Support\Facades\Route;

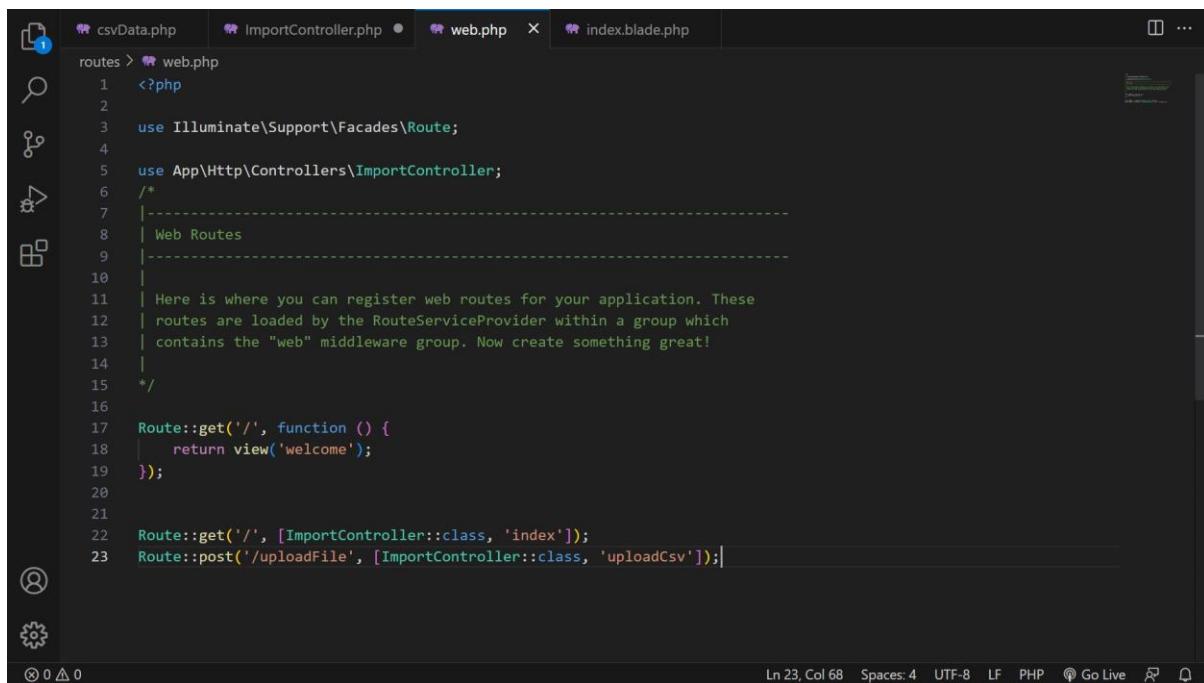
use App\Http\Controllers\ImportController;

/*
|--------------------------------------------------------------------------
|
| Web Routes
|
| Here is where you can register web routes for your application.
| These
|
| routes are loaded by the RouteServiceProvider within a group
| which
|
| contains the "web" middleware group. Now create something
| great!
|
*/
Route::get('/', function () {
    return view('welcome');
});
```

```
Route::get('/', [ImportController::class, 'index']);

Route::post('/uploadFile', [ImportController::class, 'uploadCsv']);
```

Screenshot of code:



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\ImportController;
6  /*
7  |-----|
8  | Web Routes
9  |
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 */
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
21
22 Route::get('/', [ImportController::class, 'index']);
23 Route::post('/uploadFile', [ImportController::class, 'uploadCsv']);
```

Step 7: Create View page index.blade.php

Write the Code in file:

```
<!doctype html>

<html>

<head>

<title>Import CSV Data to MySQL database with
Laravel</title>

</head>

<body>

<!-- Message -->

@if (Session::has('message'))
```

```

<p>{{ Session::get('message') }}</p>

@endif

<!-- Form -->

<form method='post' action='/uploadFile'
enctype='multipart/form-data'>

@csrf

<input type='file' name='file'>

<input type='submit' name='submit' value='Import'>

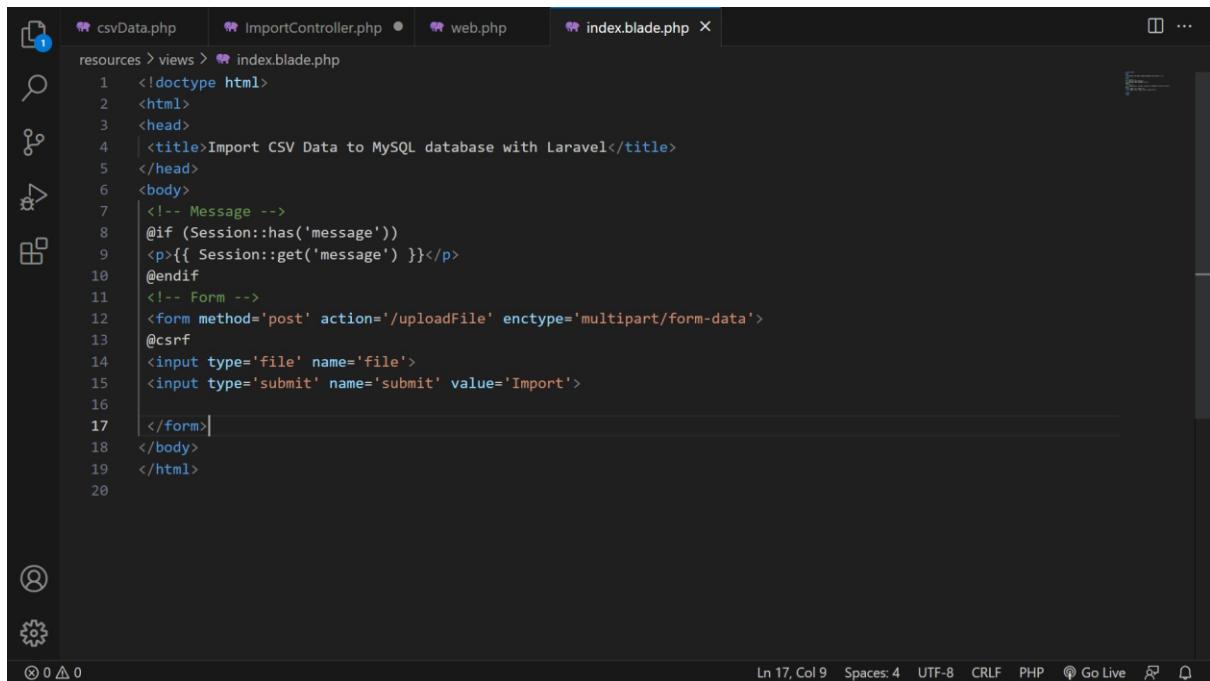
</form>

</body>

</html>

```

Screenshot of code:



```

resources > views > index.blade.php
<!DOCTYPE html>
<html>
<head>
<title>Import CSV Data to MySQL database with Laravel</title>
</head>
<body>
<!-- Message -->
@if (Session::has('message'))
<p>{{ Session::get('message') }}</p>
@endif
<!-- Form -->
<form method='post' action='/uploadFile' enctype='multipart/form-data'>
@csrf
<input type='file' name='file'>
<input type='submit' name='submit' value='Import'>
</form>
</body>
</html>

```

Step 8: Run Development Server

Then write the code for run the server

“php artisan serve”

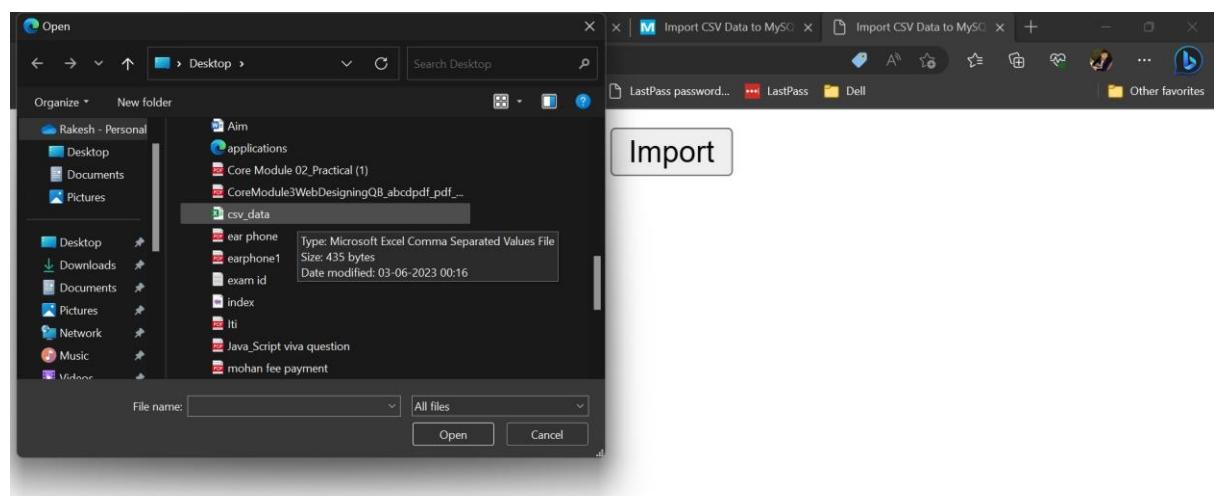
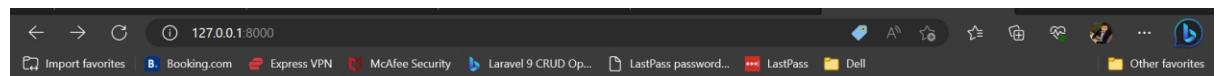
Step 9: Create csv file in excel sheet and enter data

Step 10: Then Go to browser and search

“<http://localhost:8000>”

Screenshot of Output:

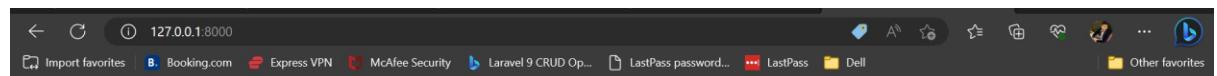
Choose file





Choose File csv_data.csv

Import



Import Successful.

Choose File No file chosen

Import

Show in Database

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=csv9&table=csv_data

The screenshot shows the phpMyAdmin interface with the following details:

- Database Structure:** The left sidebar shows the database structure with the 'csv9' database selected. It contains the following tables:
 - adit
 - adit1
 - appdata
 - auth
 - auth2
 - company
 - csv_data
 - users
 - ibm
 - information_schema
 - laravel
 - live
 - live2
 - model
- Table Data:** The 'csv_data' table is selected in the main area. The table structure is as follows:

	id	first_name	last_name	email	gender	up
<input type="checkbox"/>	1	Rakesh	Kumar	Male	rakesh@gmail.com	20
<input type="checkbox"/>	2	Raushan	Kumar	Male	raushan@gmail.com	20
<input type="checkbox"/>	3	Amit	Kumar	Male	amit@gmail.com	20
<input type="checkbox"/>	4	Priyanaka	Gupta	Female	priyanka@gmail.com	20
<input type="checkbox"/>	5	Dharam	Baghel	Male	dharam@gmail.com	20
<input type="checkbox"/>	6	Dinesh	Ajmeera	Male	dinesh@gmail.com	20
<input type="checkbox"/>	7	Rahul	Soyankar	Male	rahul@gmail.com	20
<input type="checkbox"/>	8	Shivam	Vishawkrma	Male	shivam@gmail.com	20
<input type="checkbox"/>	9	Mohan	Kumar	Male	mohan@gmail.com	20
<input type="checkbox"/>	10	Virendra	Kumar	Male	virendra@gmail.com	20

11 Aim: - Create a login and register Forms using Live wire.

Learning outcome: - Able to understand Laravel Framework in Live wire.

Requirement of Hardware & Software: -

- 1 Working PC.
- 2 Windows Operating System.
- 3 Xampp server and also PHP.
- 4 Composer, Laravel 8 and any browser.

Procedure: -

Step1:-

Open your Command prompt and Please enter a Laravel command
“Composer create-project --prefer-dist laravel/laravel livewirelogin”

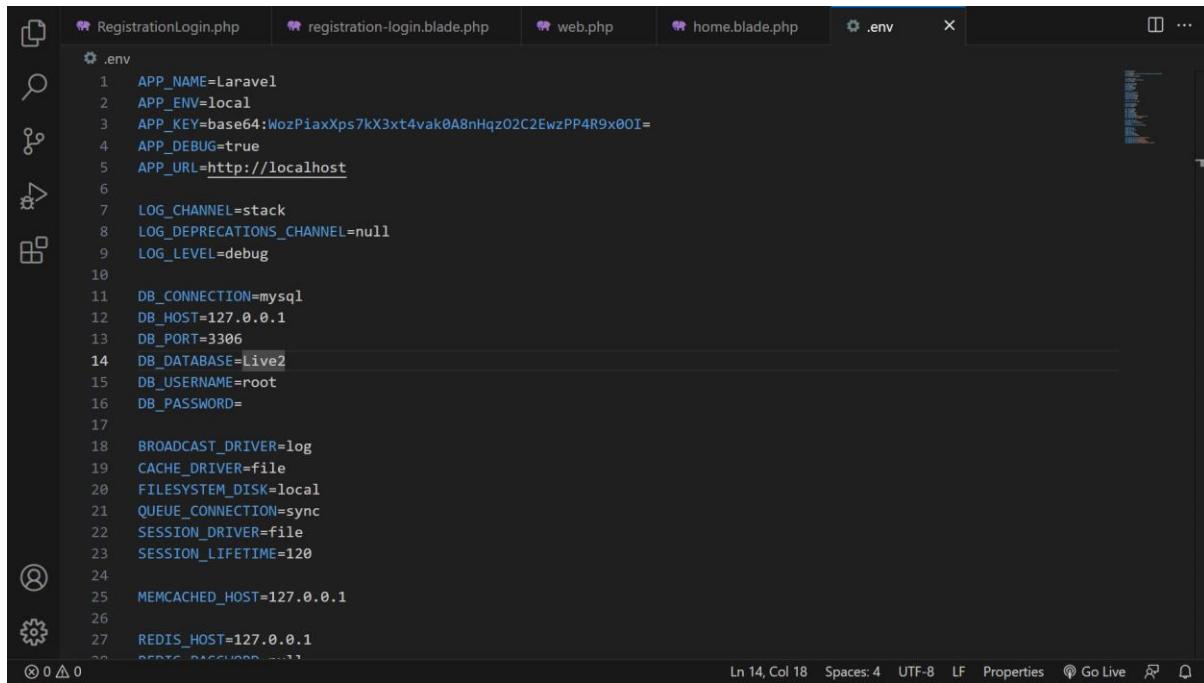


Step 2: Database Configuration

If your **Laravel project** is fresh then you need to update your database credentials. Just open the .env file in your Laravel project.

.env

Screenshot of code:



```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:WozPiaxXps7kX3xt4vak0A8nHqz02C2EwzPP4R9x0OI=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=live2
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=
```

Step 3: Migration Setup

Now, Open command prompt and run the following command to create the table into your added database:

Code:

“`Php artisan migrate`”

Step 4: Install Livewire to Create Components

In this step, you need to install livewire to your Laravel project using the following command:

“`Composer require livewire/livewire`”

Step 5: Create Livewire Components for Login and Registration

In this step, create the livewire components for creating a Laravel livewire registration and login component using the

following command. So Open your cmd and run the following commands:

“php artisan make:livewire registration-login”

Step 6: Create some file

app/Http/Livewire/RegistrationLogin.php

resources/views/livewire/registration-login.blade.php

Now, Navigate to app/Http/Livewire folder and open RegistrationLogin.php file. Then add the following code into your RegistrationLogin.php file:

Code of RegistrationLongin.php :

```
<?php
namespace App\Http\Livewire;
use Livewire\Component;
use Hash;
use App\Models\User;
class RegistrationLogin extends Component
{
    public $users, $email, $password, $name;
    public $registerForm = false;
    public function render()
    {
        return view('livewire.registration-login');
    }
}
```

```
private function resetInputFields(){
    $this->name = "";
    $this->email = "";
    $this->password = "";
}

public function login()
{
    $validatedDate = $this->validate([
        'email' => 'required|email',
        'password' => 'required',
    ]);
}

if(\Auth::attempt(array('email' => $this->email, 'password' => $this->password))){
    session()->flash('message', "You have been successfully login.");
} else{
    session()->flash('error', 'email and password are wrong.');
}

public function register()
{
    $this->registerForm = !$this->registerForm;
}
```

```
public function registerStore()
{
    $v = $this->validate([
        'name' => 'required',
        'email' => 'required|email',
        'password' => 'required',
    ]);

    $this->password = Hash::make($this->password);

    $data = [ 'name' => $this->name,
        'email' => $this->email,
        'password' => $this->password
    ];

    User::create($data);

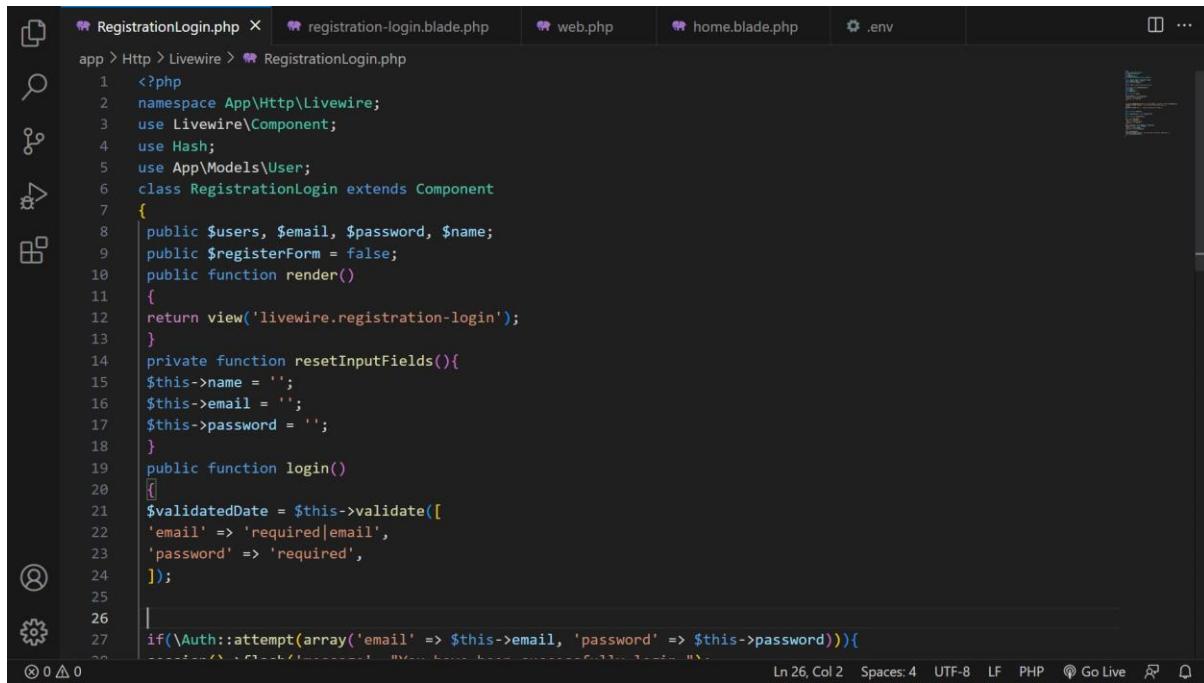
    session()->flash('message', 'You have been successfully registered.');

    $this->resetInputFields();

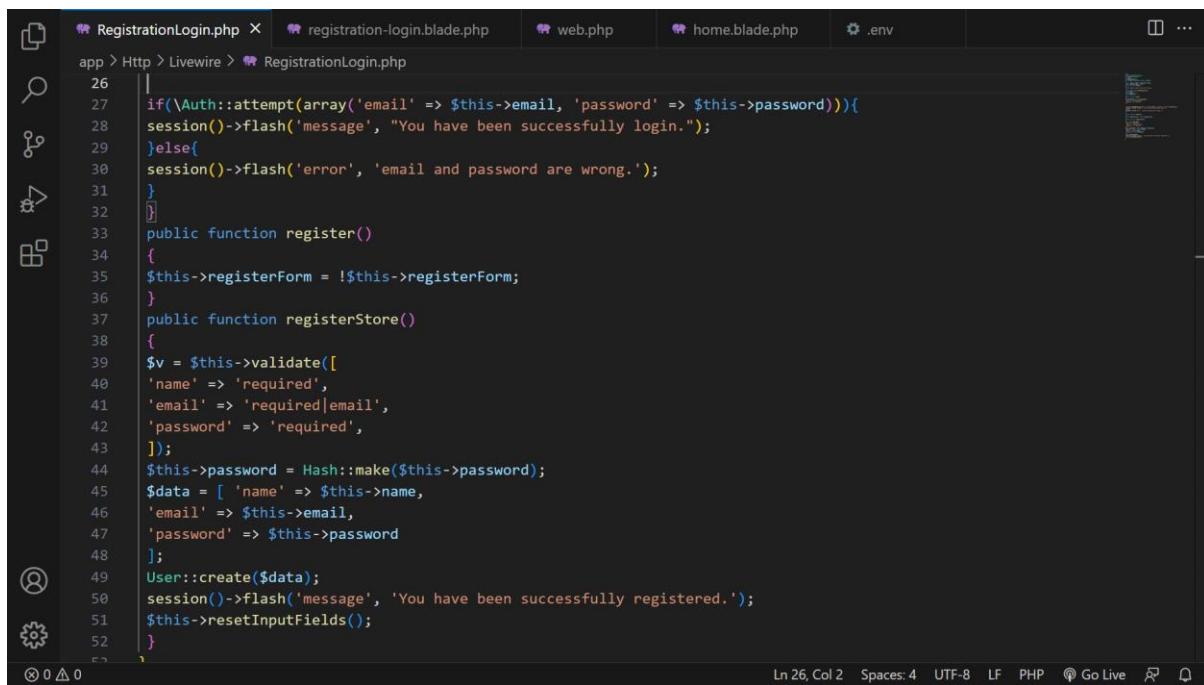
}

}
```

Screenshot of code:



```
app > Http > Livewire > RegistrationLogin.php
1  <?php
2  namespace App\Http\Livewire;
3  use Livewire\Component;
4  use Hash;
5  use App\Models\User;
6  class RegistrationLogin extends Component
7  {
8      public $users, $email, $password, $name;
9      public $registerForm = false;
10     public function render()
11     {
12         return view('livewire.registration-login');
13     }
14     private function resetInputFields(){
15         $this->name = '';
16         $this->email = '';
17         $this->password = '';
18     }
19     public function login()
20     {
21         $validatedData = $this->validate([
22             'email' => 'required|email',
23             'password' => 'required',
24         ]);
25
26         if(\Auth::attempt(['email' => $this->email, 'password' => $this->password])){
27             session()->flash('message', "You have been successfully login.");
28         }else{
29             session()->flash('error', 'email and password are wrong.');
30         }
31     }
32
33     public function register()
34     {
35         $this->registerForm = !$this->registerForm;
36     }
37     public function registerStore()
38     {
39         $v = $this->validate([
40             'name' => 'required',
41             'email' => 'required|email',
42             'password' => 'required',
43         ]);
44         $this->password = Hash::make($this->password);
45         $data = [ 'name' => $this->name,
46             'email' => $this->email,
47             'password' => $this->password
48         ];
49         User::create($data);
50         session()->flash('message', 'You have been successfully registered.');
51         $this->resetInputFields();
52     }
53 }
```



```
app > Http > Livewire > RegistrationLogin.php
26
27     if(\Auth::attempt(['email' => $this->email, 'password' => $this->password])){
28         session()->flash('message', "You have been successfully login.");
29     }else{
30         session()->flash('error', 'email and password are wrong.');
31     }
32
33     public function register()
34     {
35         $this->registerForm = !$this->registerForm;
36     }
37     public function registerStore()
38     {
39         $v = $this->validate([
40             'name' => 'required',
41             'email' => 'required|email',
42             'password' => 'required',
43         ]);
44         $this->password = Hash::make($this->password);
45         $data = [ 'name' => $this->name,
46             'email' => $this->email,
47             'password' => $this->password
48         ];
49         User::create($data);
50         session()->flash('message', 'You have been successfully registered.');
51         $this->resetInputFields();
52     }
53 }
```

Step 7: Resources/views/livewire/registration-login.blade.php

After that, Navigate to resources/views/livewire folder and open registration-login.blade.php file. Then add the following code into your registration-login.blade.php file:

Code:

```
<div>

<div class="row">

<div class="col-md-12">

@if (session()->has('message'))

<div class="alert alert-success">

{{ session('message') }}

</div>

@endif

@if (session()->has('error'))

<div class="alert alert-danger">

{{ session('error') }}

</div>

@endif

</div>

</div>

@if($registerForm)

<form>

<div class="row">
```

```
<div class="col-md-12">

<div class="form-group">
<label>Name :</label>
<input type="text" wire:model="name" class="form-control">
@error('name') <span class="text-danger error">{{ $message }}</span>@enderror
</div>
</div>

<div class="col-md-12">
<div class="form-group">
<label>Email :</label>
<input type="text" wire:model="email" class="form-control">
@error('email') <span class="text-danger error">{{ $message }}</span>@enderror
</div>
</div>

<div class="col-md-12">
<div class="form-group">
<label>Password :</label>
<input type="password" wire:model="password" class="form-control">
```

```
@error('password') <span class="text-danger error">{{  
$message }}</span>@enderror  
</div>  
</div>  
wire:click.prevent="registerStore">Register</button>  
</div>  
wire:click.prevent="register"><strong>Login</strong></a>  
</div>  
</div>  
</form>  
 @else  
Email :</label>  

```

```
@error('email') <span class="text-danger error">{{ $message }}</span>@enderror
</div>
</div>

<div class="col-md-12">
<div class="form-group">
<label>Password :</label>
<input type="password" wire:model="password" class="form-control">
@error('password') <span class="text-danger error">{{ $message }}</span>@enderror
</div>
</div>

<div class="col-md-12 text-center">
<button class="btn text-white btn-success"
wire:click.prevent="login">Login</button>
</div>

<div class="col-md-12">
Don't have account? <a class="btn btn-primary text-white"
wire:click.prevent="register"><strong>Register
Here</strong></a>
</div>
```

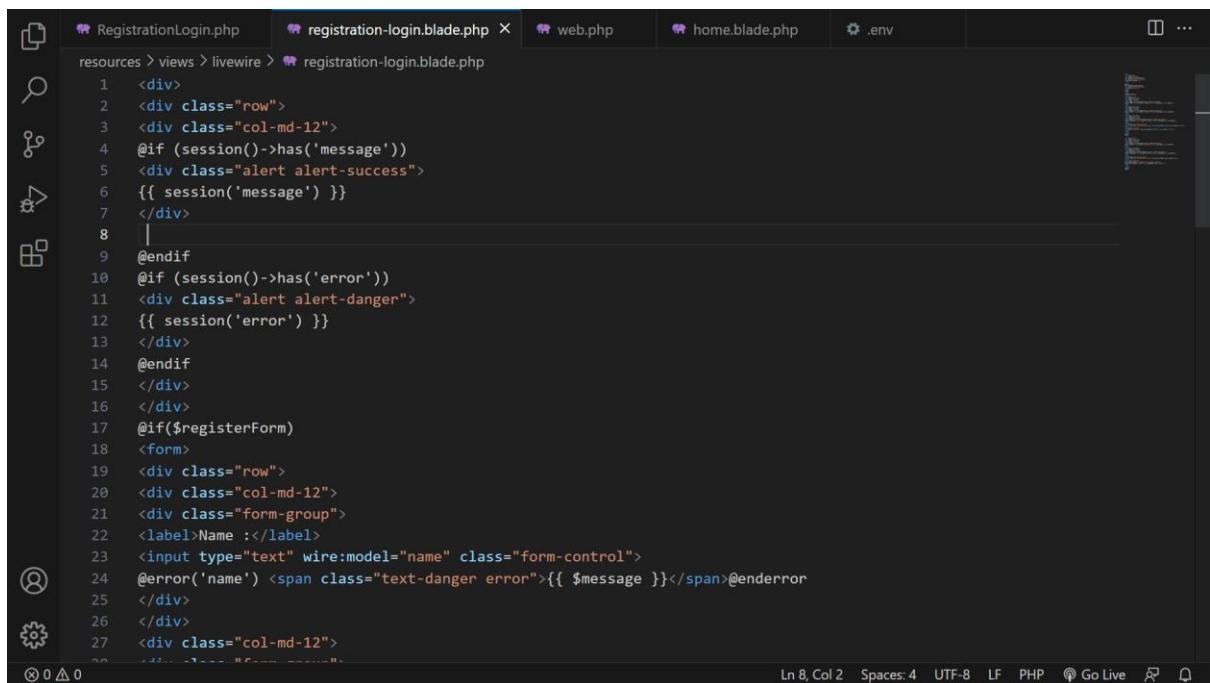
```
</div>

</form>

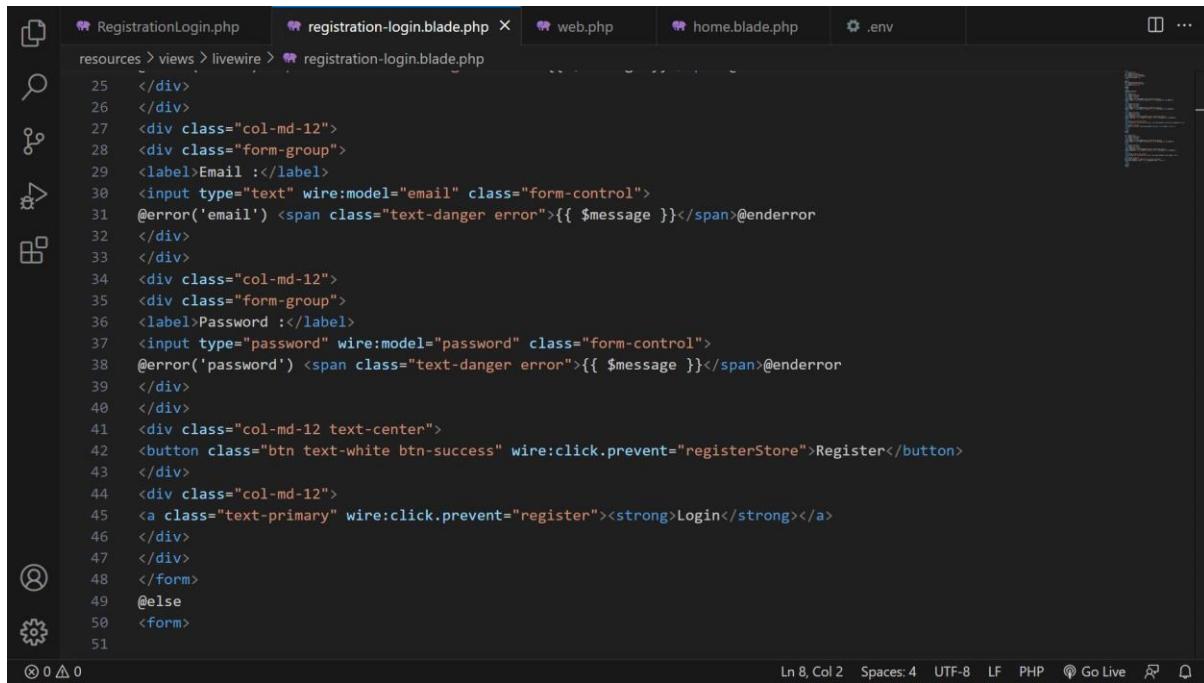
@endif

</div>
```

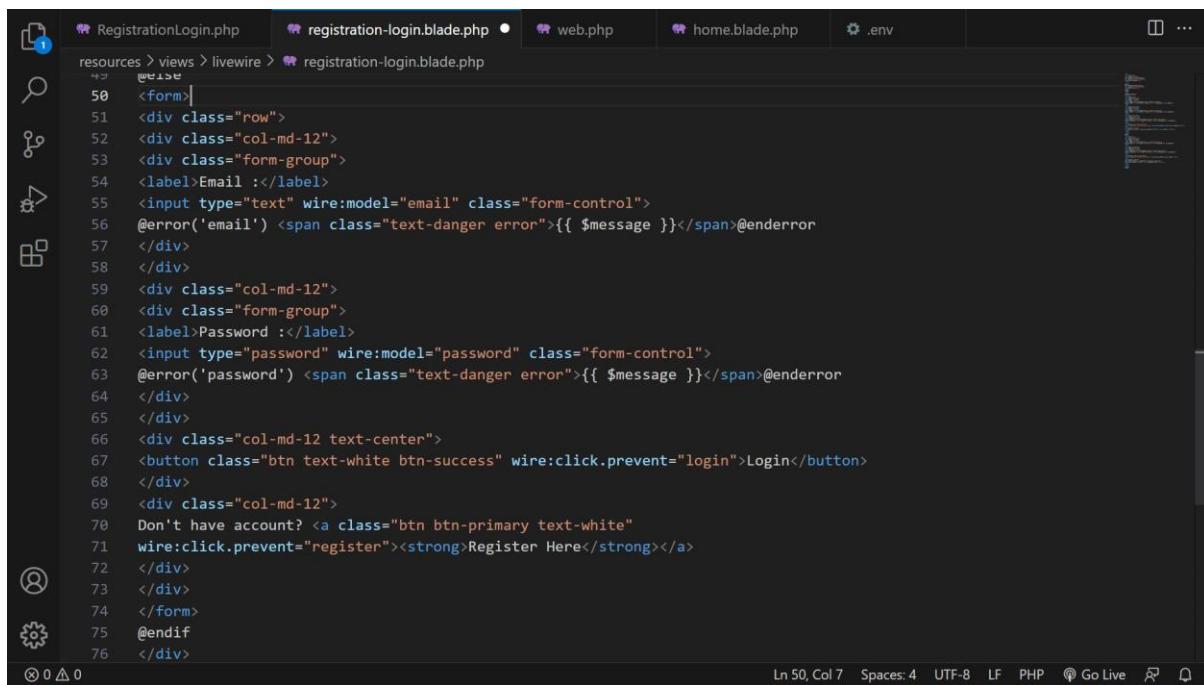
Screenshot of code :



```
resources > views > livewire > registration-login.blade.php
1  <div>
2  <div class="row">
3  <div class="col-md-12">
4  @if (session()->has('message'))
5  <div class="alert alert-success">
6  {{ session('message') }}
7  </div>
8  |
9  @endif
10 @if (session()->has('error'))
11 <div class="alert alert-danger">
12 {{ session('error') }}
13 </div>
14 @endif
15 </div>
16 </div>
17 @if($registerForm)
18 <form>
19 <div class="row">
20 <div class="col-md-12">
21 <div class="form-group">
22 <label>Name :</label>
23 <input type="text" wire:model="name" class="form-control">
24 @error('name') <span class="text-danger error">{{ $message }}</span>@enderror
25 </div>
26 </div>
27 <div class="col-md-12">
```



```
resources > views > livewire > registration-login.blade.php
25  </div>
26  </div>
27  <div class="col-md-12">
28  <div class="form-group">
29  <label>Email :</label>
30  <input type="text" wire:model="email" class="form-control">
31  @error('email') <span class="text-danger error">{{ $message }}</span>enderror
32  </div>
33  </div>
34  <div class="col-md-12">
35  <div class="form-group">
36  <label>Password :</label>
37  <input type="password" wire:model="password" class="form-control">
38  @error('password') <span class="text-danger error">{{ $message }}</span>enderror
39  </div>
40  </div>
41  <div class="col-md-12 text-center">
42  <button class="btn text-white btn-success" wire:click.prevent="registerStore">Register</button>
43  </div>
44  <div class="col-md-12">
45  <a class="text-primary" wire:click.prevent="register"><strong>Login</strong></a>
46  </div>
47  </div>
48  </form>
49  @else
50  <form>
51
52  <div class="row">
53  <div class="col-md-12">
54  <div class="form-group">
55  <label>Email :</label>
56  <input type="text" wire:model="email" class="form-control">
57  @error('email') <span class="text-danger error">{{ $message }}</span>enderror
58  </div>
59  <div class="col-md-12">
60  <div class="form-group">
61  <label>Password :</label>
62  <input type="password" wire:model="password" class="form-control">
63  @error('password') <span class="text-danger error">{{ $message }}</span>enderror
64  </div>
65  </div>
66  <div class="col-md-12 text-center">
67  <button class="btn text-white btn-success" wire:click.prevent="login">Login</button>
68  </div>
69  <div class="col-md-12">
70  Don't have account? <a class="btn btn-primary text-white" href="#">
71  wire:click.prevent="register"><strong>Register Here</strong></a>
72  </div>
73  </div>
74  </form>
75  @endif
76  </div>
```



```
resources > views > livewire > registration-login.blade.php
50  <form>
51
52  <div class="row">
53  <div class="col-md-12">
54  <div class="form-group">
55  <label>Email :</label>
56  <input type="text" wire:model="email" class="form-control">
57  @error('email') <span class="text-danger error">{{ $message }}</span>enderror
58  </div>
59  <div class="col-md-12">
60  <div class="form-group">
61  <label>Password :</label>
62  <input type="password" wire:model="password" class="form-control">
63  @error('password') <span class="text-danger error">{{ $message }}</span>enderror
64  </div>
65  </div>
66  <div class="col-md-12 text-center">
67  <button class="btn text-white btn-success" wire:click.prevent="login">Login</button>
68  </div>
69  <div class="col-md-12">
70  Don't have account? <a class="btn btn-primary text-white" href="#">
71  wire:click.prevent="register"><strong>Register Here</strong></a>
72  </div>
73  </div>
74  </form>
75  @endif
76  </div>
```

Step 8: Create Route

In this step, Navigate to routes folder and open web.php. Then add the following routes into your web.php file:

Code:

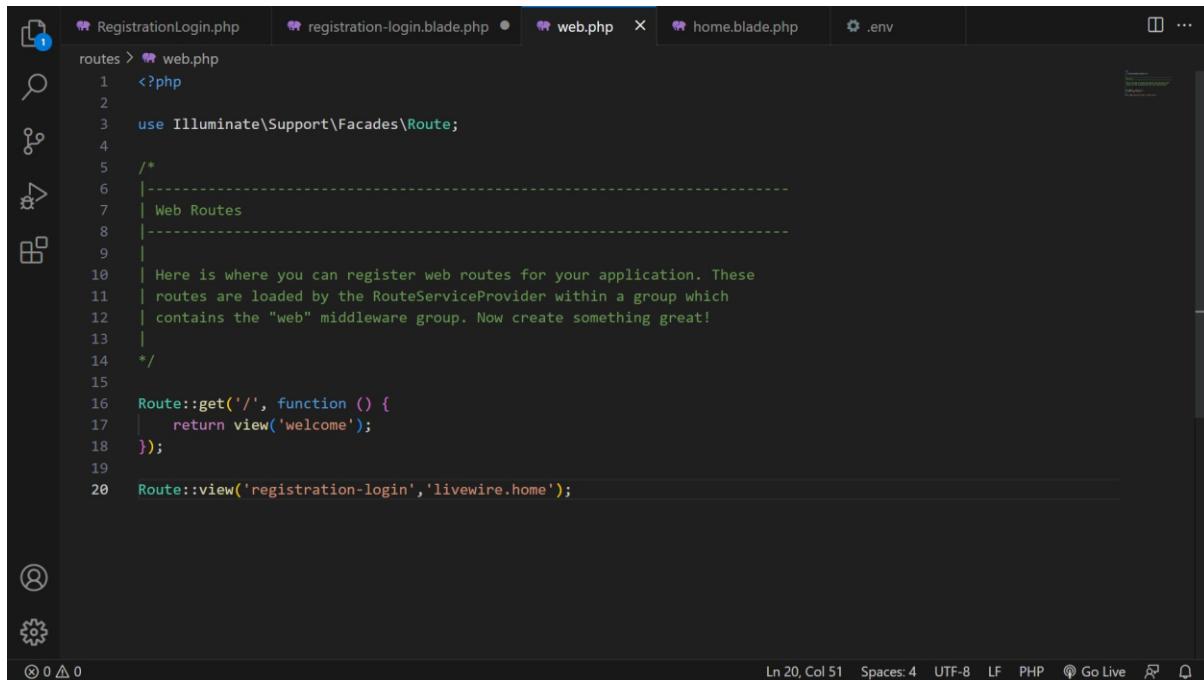
```
<?php

use Illuminate\Support\Facades\Route;

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application.
| These
| routes are loaded by the RouteServiceProvider within a group
| which
| contains the "web" middleware group. Now create something
| great!
|
*/
Route::get('/', function () {
    return view('welcome');
});

Route::view('registration-login','livewire.home');
```

Screenshot of code:



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  | -----
7  | Web Routes
8  | -----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::view('registration-login', 'livewire.home');
```

Step 9: Create View Files

In this step, navigate to resources/views/livewire folder and create one blade view files that name home.blade.php file. Then add the following code into your home.blade.php file

Code:

```
<!DOCTYPE html>

<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Laravel Livewire Registration and Login</title>

<!-- Fonts -->

<link
  href="https://fonts.googleapis.com/css?family=Nunito:200,600"
  rel="stylesheet">

<link rel="stylesheet"
  href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/
  bootstrap.min.css">

<!-- Styles -->

<style>

  html, body {
    background-color: #fff;
    color: #636b6f;
    font-family: 'Nunito', sans-serif;
    font-weight: 200;
    height: 100vh;
    margin: 0;
  }

  .full-height {
    height: 100vh;
  }

  .flex-center {
    align-items: center;
  }

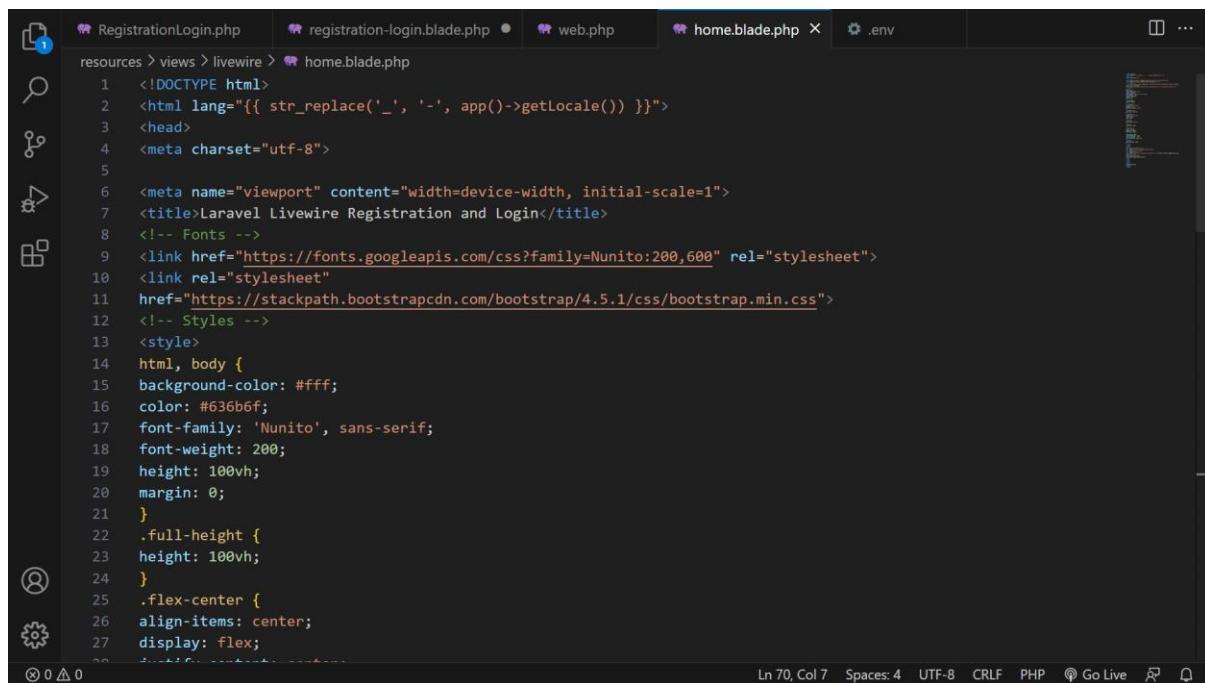
</style>
```

```
display: flex;  
justify-content: center;  
}  
  
.position-ref {  
position: relative;  
}  
  
.top-right {  
position: absolute;  
right: 10px;  
top: 18px;  
}  
  
.content {  
text-align: center;  
}  
  
.title {  
font-size: 84px;  
}  
  
.links > a {  
color: #636b6f;  
padding: 0 25px;  
font-size: 13px;
```

```
font-weight: 600;  
letter-spacing: .1rem;  
text-decoration: none;  
text-transform: uppercase;  
}  
.m-b-md {  
margin-bottom: 30px;  
}  
</style>  
</head>  
<body>  
<div class="container mt-5">  
<div class="row mt-5 justify-content-center">  
<div class="mt-5 col-md-8">  
<div class="card">  
<div class="card-header bg-primary text-white"><h3>Laravel  
Livewire Registration and  
login</h3></div>  
<div class="card-body">  
@livewire('registration-login')  
</div>  
</div>  
</div>
```

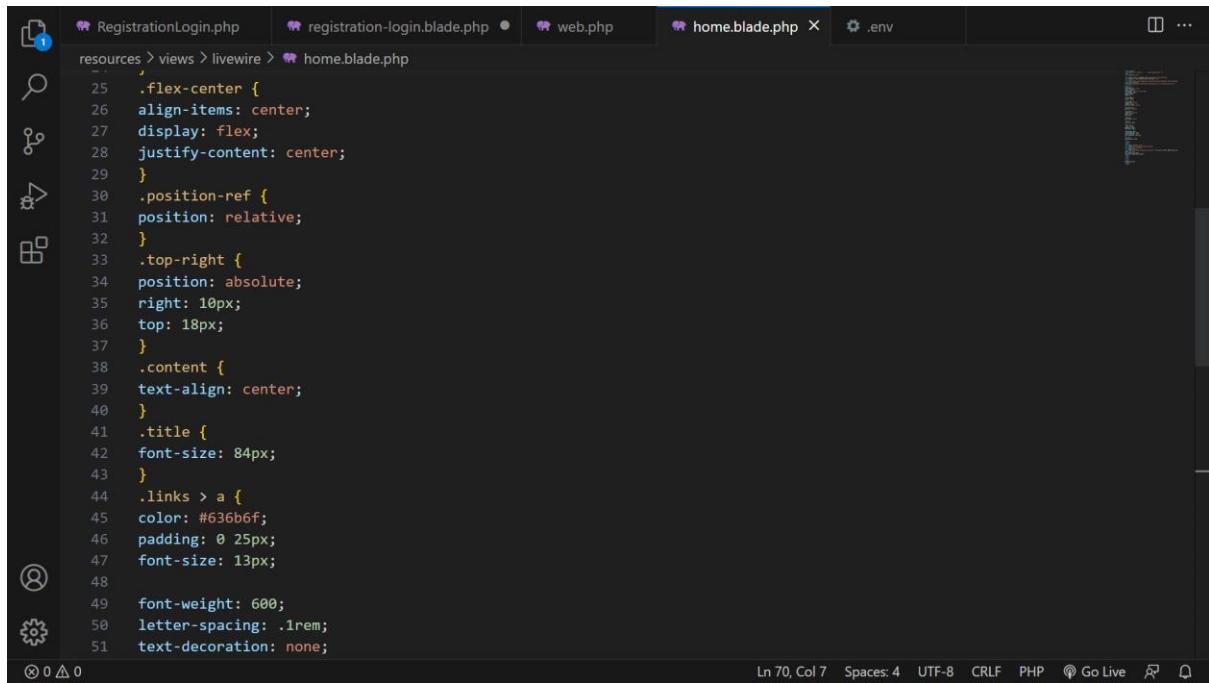
```
</div>  
</div>  
@livewireScripts  
</body>  
</html>
```

Screenshot of code:



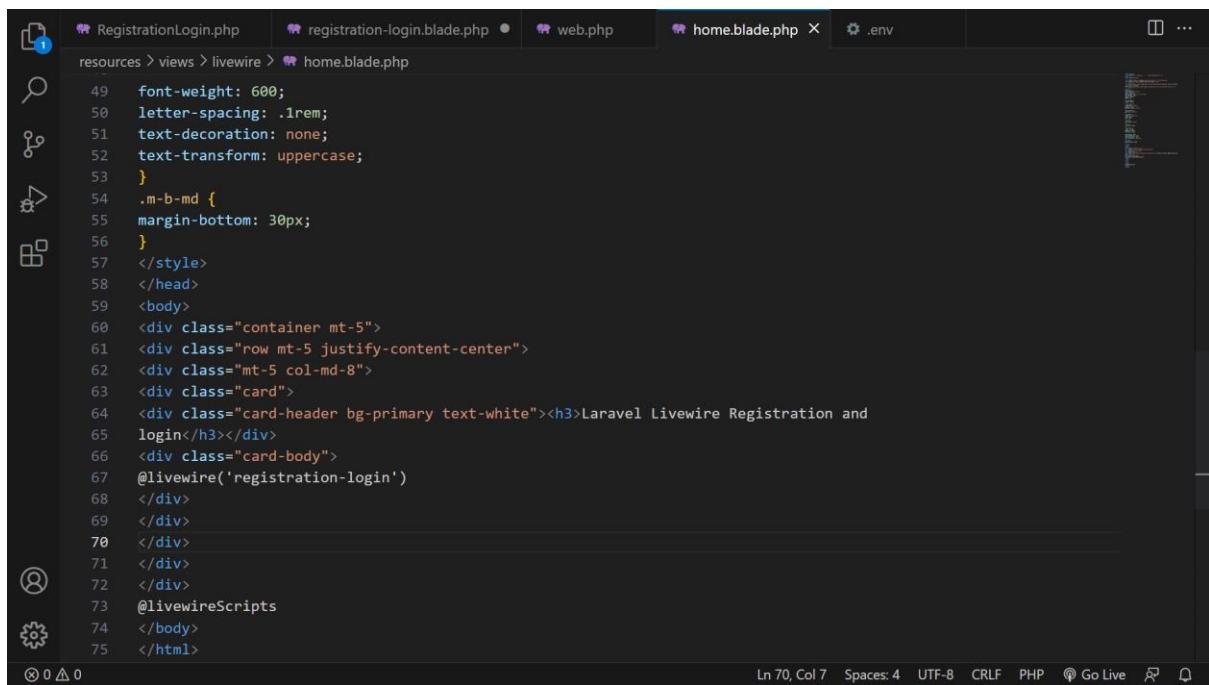
The screenshot shows a code editor with a dark theme. The file being edited is `home.blade.php`, located in the `resources > views > livewire` directory. The code is a Laravel Livewire component for registration and login. It includes a DOCTYPE declaration, a meta tag for the viewport, a title, and a link to a Google Fonts stylesheet. The styles section contains CSS for the `html` and `body` elements, setting the background color to `#fff` and the font family to `'Nunito', sans-serif`. The code editor also shows a sidebar with file navigation and a status bar at the bottom indicating line 70, column 7, and other file details.

```
1  <!DOCTYPE html>  
2  <html lang="{{ str_replace('_', '-', app()->getLocale()) }}>  
3  <head>  
4  <meta charset="utf-8">  
5  
6  <meta name="viewport" content="width=device-width, initial-scale=1">  
7  <title>Laravel Livewire Registration and Login</title>  
8  <!-- Fonts -->  
9  <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">  
10 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css">  
11 <!-- Styles -->  
12 <style>  
13 html, body {  
14 background-color: #fff;  
15 color: #636b6f;  
16 font-family: 'Nunito', sans-serif;  
17 font-weight: 200;  
18 height: 100vh;  
19 margin: 0;  
20 }  
21 .full-height {  
22 height: 100vh;  
23 }  
24 .flex-center {  
25 align-items: center;  
26 display: flex;  
27 justify-content: center;
```



```
resources > views > livewire > home.blade.php
25 .flex-center {
26   align-items: center;
27   display: flex;
28   justify-content: center;
29 }
30 .position-ref {
31   position: relative;
32 }
33 .top-right {
34   position: absolute;
35   right: 10px;
36   top: 18px;
37 }
38 .content {
39   text-align: center;
40 }
41 .title {
42   font-size: 84px;
43 }
44 .links > a {
45   color: #636b6f;
46   padding: 0 25px;
47   font-size: 13px;
48 }
49 font-weight: 600;
50 letter-spacing: .1rem;
51 text-decoration: none;
```

Ln 70, Col 7 Spaces: 4 UTF-8 CRLF PHP ⚡ Go Live ⌂ ⌂



```
resources > views > livewire > home.blade.php
49 font-weight: 600;
50 letter-spacing: .1rem;
51 text-decoration: none;
52 text-transform: uppercase;
53 }
54 .m-b-md {
55   margin-bottom: 30px;
56 }
57 </style>
58 </head>
59 <body>
60 <div class="container mt-5">
61 <div class="row mt-5 justify-content-center">
62 <div class="mt-5 col-md-8">
63 <div class="card">
64 <div class="card-header bg-primary text-white"><h3>Laravel Livewire Registration and
65 login</h3></div>
66 <div class="card-body">
67   @livewire('registration-login')
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73   @livewireScripts
74 </body>
75 </html>
```

Ln 70, Col 7 Spaces: 4 UTF-8 CRLF PHP ⚡ Go Live ⌂ ⌂

Step 10: Run the server

Finally, you need to run the following PHP artisan serve command to start your laravel livewire registration and login app:

Code:

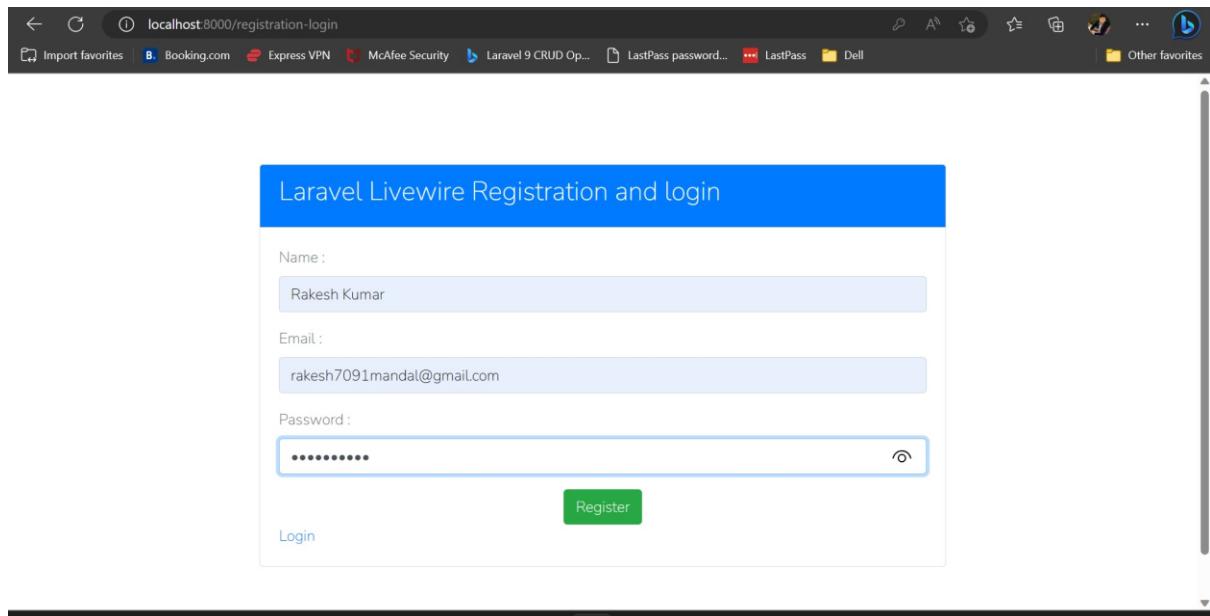
“php artisan serve”

Step 10: Then Go to browser and search

“<http://localhost:8000/registration-login>”

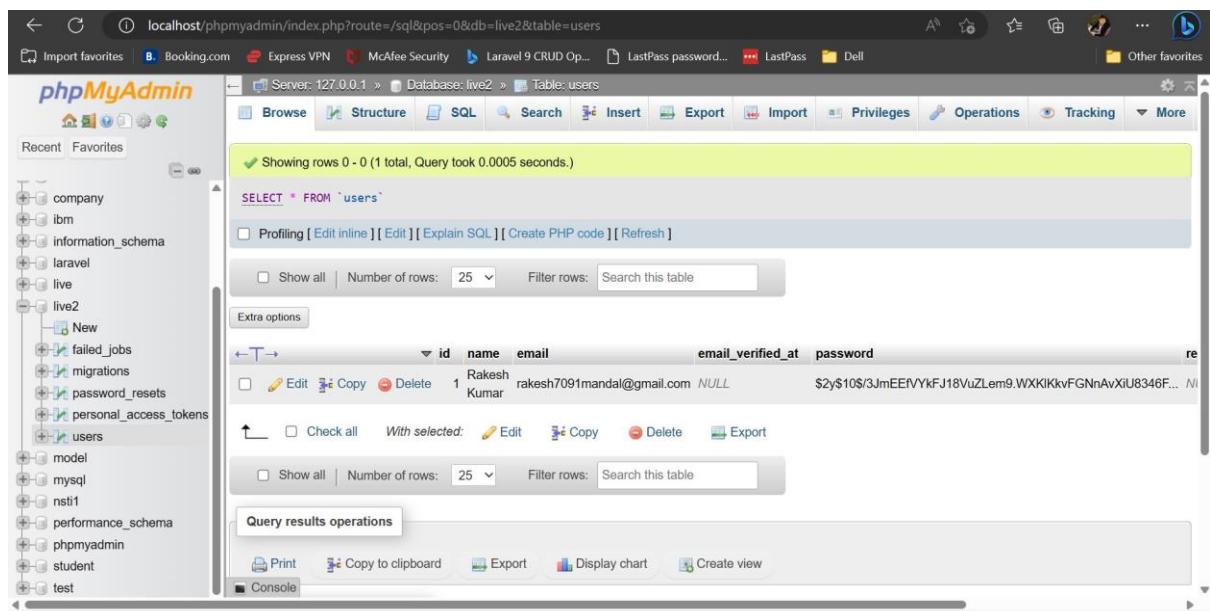
Screenshot of Output:

Registration form



The screenshot shows a browser window with the URL localhost:8000/registration-login. The page title is "Laravel Livewire Registration and login". The form contains three text input fields: "Name" with the value "Rakesh Kumar", "Email" with the value "rakesh7091mandal@gmail.com", and "Password" with the value "*****". Below the form are two buttons: a green "Register" button and a blue "Login" button.

Show in Database



The screenshot shows the phpMyAdmin interface connected to a MySQL database named "live2". The "users" table is selected. The table has columns: id, name, email, email_verified_at, and password. A single row is visible with id 1, name "Rakesh Kumar", email "rakesh7091mandal@gmail.com", and password "\$2y\$10\$/3JmEEfVYkJ18VuZLem9WXKIKkvFGNnAvXiU8346F...". A green status bar at the top of the interface displays the message "Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)".

Longin form

localhost:8000/registration-login

Email: rakesh7091mandal@gmail.com

Password:

Don't have account? [Register Here](#)

[Login](#)

Laravel Livewire Registration and login

Longin Successfully

localhost:8000/registration-login

You have been successfully login.

Email: rakesh7091mandal@gmail.com

Password:

Don't have account? [Register Here](#)

[Login](#)

Laravel Livewire Registration and login

