

Module 15 Assignment

Syed Amir Hamza

Task 1: Request Validation

Implement request validation for a registration form that contains the following fields: name, email, and password. Validate the following rules:

name: required, string, minimum length 2.

email: required, valid email format.

password: required, string, minimum length 8.

Answer

```
public function ValidateAndRegister(Request $request)
{
    // dd($request->all());

    $request->validate([
        [
            'name'=>'required|string|min:2',
            'email'=>'required|email',
            'password'=>'required|string|min:8'
        ]
    ]);
    return back()->withSuccess('Register Success !!');
}
```

Task 2: Request Redirect

Create a route /home that redirects to /dashboard using a 302 redirect.

Answer

```
Route::get('/home', function () {
    return redirect('/dashboard', 302);
});
```

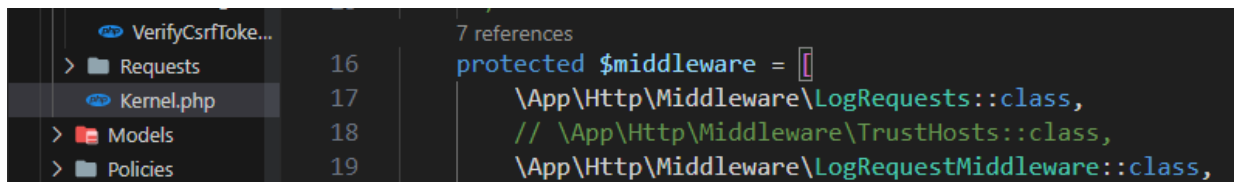
Task 3: Global Middleware

Create a global middleware that logs the request method and URL for every incoming request. Log the information to the Laravel log file.

Answer



```
0 references | 0 overrides
public function handle(Request $request, Closure $next): Response
{
    Log::info('Request Method: '.$request->method());
    Log::info('Request URL: '.$request->fullUrl());
    return $next($request);
}
```



```
7 references
protected $middleware = [
    \App\Http\Middleware\LogRequests::class,
    // \App\Http\Middleware\TrustHosts::class,
    \App\Http\Middleware\LogRequestMiddleware::class,
```

Task 4: Route Middleware

Create a route group for authenticated users only. This group should include routes for /profile and /settings. Apply a middleware called AuthMiddleware to the route group to ensure only authenticated users can access these routes.

Answer



```
53
54
55
56
57
58
59
60
61
62
Route::middleware(['auth'])->group(function () {
    Route::get('/profile', function () {
        // Your profile logic here...
    });

    Route::get('/settings', function () {
        // Your settings logic here...
    });
});
```

Task 5: Controller

Create a controller called ProductController that handles CRUD operations for a resource called Product. Implement the following methods:

index(): Display a list of all products.

create(): Display the form to create a new product.

store(): Store a newly created product.

edit(\$id): Display the form to edit an existing product.

update(\$id): Update the specified product.

destroy(\$id): Delete the specified product.

Answer

```
71 console.php
72 web.php
73 storage
74 tests
75 vendor
76 editorsconfig

74 Route::resource("/products", ProductController::class);
75
```

```
File Edit Selection View Go Run Terminal Help
EXPLORER
ASS-15-APP
  app
    Console
    Exceptions
    Http
      Controllers
        ContactContro...
        Controller.php
        PostController...
        ProductContro...
        RegisterContr...
        Middleware
        Authenticate.p...
        EncryptCookie...
        LogRequestMi...
        LogRequests.p...
        PreventReques...
        RedirectIfAuth...
        TrimStrings.php
        TrustHosts.php
        TrustProxies.php
        ValidateSignat...
        VerifyCsrfToke...
      Requests
      Kernel.php
      Models
      Policies
      Providers
      bootstrap
      config
      database
      public
      resources
      css
      js
      views
      background
      component
        contactform.bl...
        deshbboardher...
        homehero.bla...
      OUTLINE
      TIMELINE

app > Http > Controllers > ProductController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Product;
6 use App\Http\Requests\StoreProductRequest;
7 use App\Http\Requests\UpdateProductRequest;
8
9 // class ProductController extends Controller
10 > // {
11 // }
12
13 2 references | 0 implementations
14 class ProductController extends Controller
15 {
16
17     0 references | 0 overrides
18     public function index()
19     {
20         $products = Product::all();
21         return view('products.index', compact('products'));
22     }
23
24     0 references | 0 overrides
25     public function create()
26     {
27         return view('products.create');
28     }
29
30     0 references | 0 overrides
31     public function store(StoreProductRequest $request)
32     {
33         $product = new Product;
34         // assuming the request contains 'name', 'description' and 'price'
35         $product->name = $request->name;
36         $product->description = $request->description;
37         $product->price = $request->price;
38         $product->save();
39         return redirect()->route('products.index');
40     }
41
42     0 references | 0 overrides
43     public function edit(Product $product, $id)
44     {
45         $product = Product::find($id);
46         return view('products.edit', compact('product'));
47     }
48
49     0 references | 0 overrides
50     public function update(UpdateProductRequest $request, Product $product, $id)
51     {
52         $product = Product::find($id);
53         $product->name = $request->name;
54         $product->description = $request->description;
55         $product->price = $request->price;
56         $product->save();
57         return redirect()->route('products.index');
58     }
59
60     0 references | 0 overrides
61     public function destroy(Product $product, $id)
62     {
63         $product = Product::find($id);
64         $product->delete();
65         return redirect()->route('products.index');
66     }
67 }
```

Task 6: Single Action Controller

Create a single action controller called `ContactController` that handles a contact form submission. Implement the `__invoke()` method to process the form submission and send an email to a predefined address with the submitted data.

Answer

The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer on the left shows a project structure with folders like 'Middleware', 'Requests', 'Models', 'Policies', 'Providers', 'bootstrap', 'config', 'database', 'public', 'resources', 'css', 'js', 'views', 'background', and 'component'. The code editor on the right shows a PHP class definition for 'ContactController' extending 'Controller'. The class has a single method 'invoke' that handles an incoming request, validates the request data, sends an email, and redirects back to the form page with a success message.

```

class ContactController extends Controller
{
    // Task 6: Single Action Controller

    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        // Validate the request data
        $data = $request->validate([
            'name' => 'required',
            'email' => 'required|email',
            'message' => 'required|min:20',
        ]);

        // Send an email
        Mail::raw($data['message'], function ($message) use ($data) {
            $message->from($data['email'], $data['name']);
            $message->to('predefined@example.com');
        });

        // Redirect back to the form page with a success message
        return back()->withSuccess('success', 'Thank you for your message. We will get back to you soon!');
    }
}

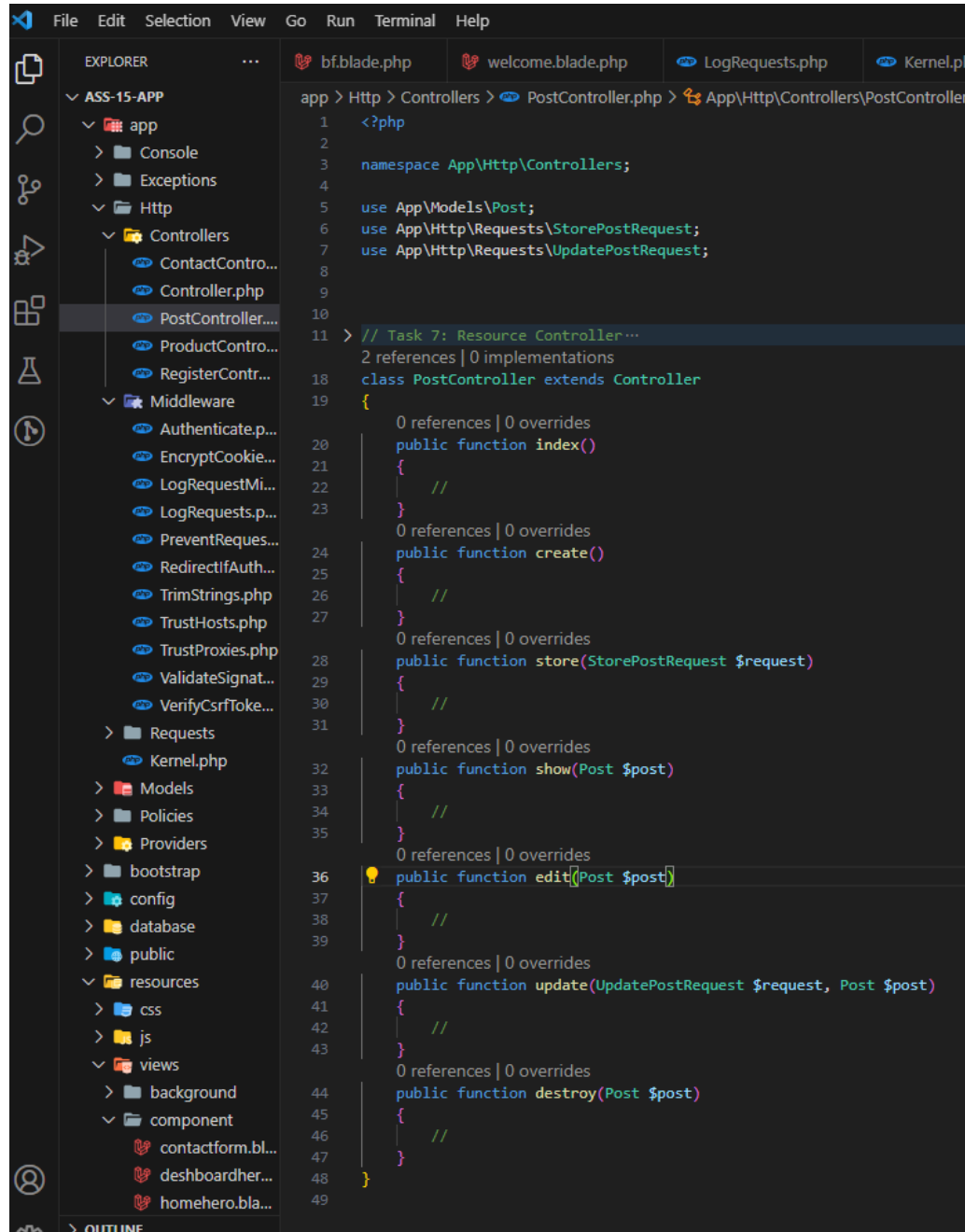
```

Task 7: Resource Controller

Create a resource controller called PostController that handles CRUD operations for a resource called Post. Ensure that the controller provides the necessary methods for the resourceful routing conventions in Laravel.

Answer

```
console.php 80
web.php      81 Route::resource( "/posts", PostController ::class );
storage      82
tests
```



The screenshot shows an IDE with the following components:

- EXPLORER:** Displays the project structure. The file `PostController.php` is selected under `Http > Controllers`.
- FILE:** Shows the file path: `app > Http > Controllers > PostController.php`.
- EDITOR:** Contains the code for `PostController.php`. The code is as follows:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Post;
6  use App\Http\Requests\StorePostRequest;
7  use App\Http\Requests\UpdatePostRequest;
8
9
10
11 > // Task 7: Resource Controller...
12 2 references | 0 implementations
13 class PostController extends Controller
14 {
15     0 references | 0 overrides
16     public function index()
17     {
18         //
19     }
20     0 references | 0 overrides
21     public function create()
22     {
23         //
24     }
25     0 references | 0 overrides
26     public function store(StorePostRequest $request)
27     {
28         //
29     }
30     0 references | 0 overrides
31     public function show(Post $post)
32     {
33         //
34     }
35     0 references | 0 overrides
36     public function edit(Post $post)
37     {
38         //
39     }
40     0 references | 0 overrides
41     public function update(UpdatePostRequest $request, Post $post)
42     {
43         //
44     }
45     0 references | 0 overrides
46     public function destroy(Post $post)
47     {
48         //
49     }
50 }
```

Task 8: Blade Template Engine


Create a Blade view called `welcome.blade.php` that includes a navigation bar and a section displaying the text "Welcome to Laravel!".


Answer

```
20 Route::get('/', function(){
21     return view('pages.welcome');
22 });
23
```

```
1 <style>
2     .hero {
3         min-height: 80vh;
4         display: flex;
5         justify-content: center;
6         align-items: center;
7         background-color: #f0f0f0;
8         color: #e74c3c;
9         font-size: 2em;
10        text-align: center;
11    }
12 </style>
13
14 <section class="hero">
15     <div>
16         <div>
17             <svg height="75" preserveAspectRatio="xMidYMid" viewBox="0 0 256 264" width="256" xmlns="http://www.w3.org/2000/svg"><path d="M255.855641 59.619717c.094924,3513426.144359,7135979.
18         </div>
19         <h1>Welcome to Laravel</h1>
20     </div>
21 </section>
22 </section>
```

```
10 @extends('Layout.app')
11
12 @section('title', 'Home Page')
13 @section('description', 'this is homepage Description')
14 @section('keyowrds', 'comma, separated, list, of, relevant, keywords')
15
16 @section('content')
17     @include('component.welcomehero')
18 @endsection
19
```

Welcome Home Dashboard Sign up Contact us



Welcome to Laravel

© Ostad | Syed Amir Hamza

THE END