Ostad

Module 15 | Assignment

Full Name: **Sakir Ahamed Bissas**

Phone: **01995542277**

Date of Submission: **29 May 2023**

http://ost-ass-mod15.laralocal

I used this virtual host from my localhost

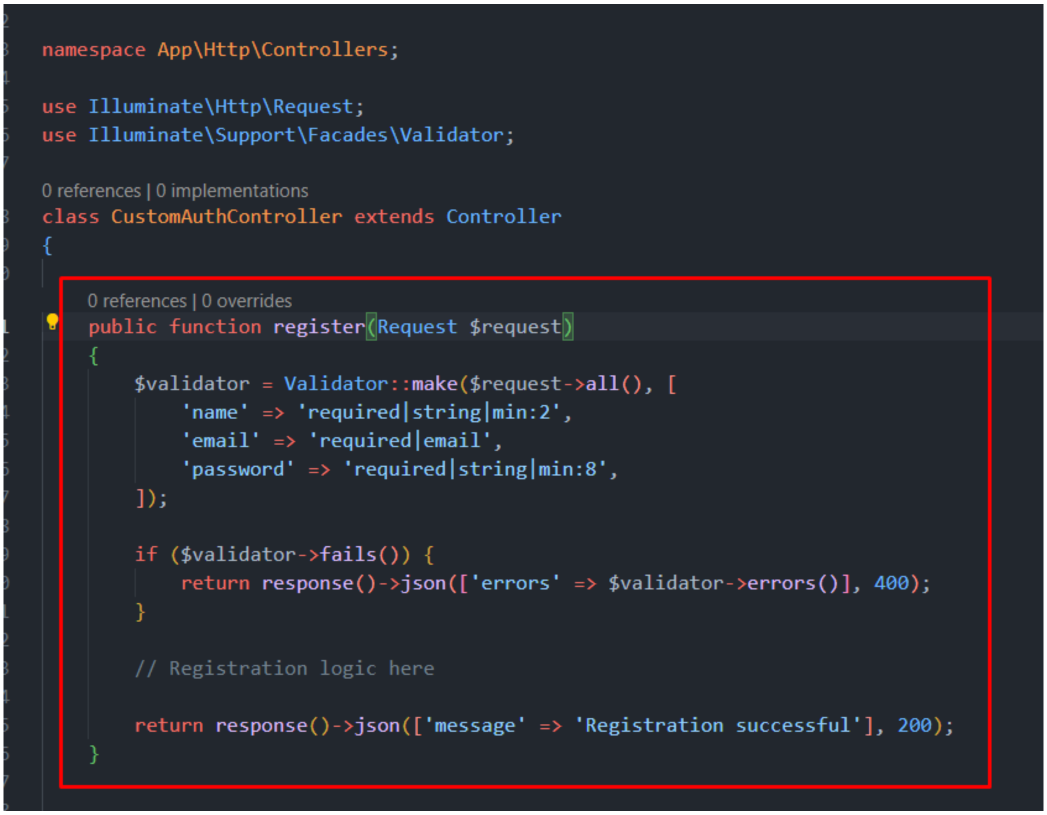
Task: (1) Request Validation

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.

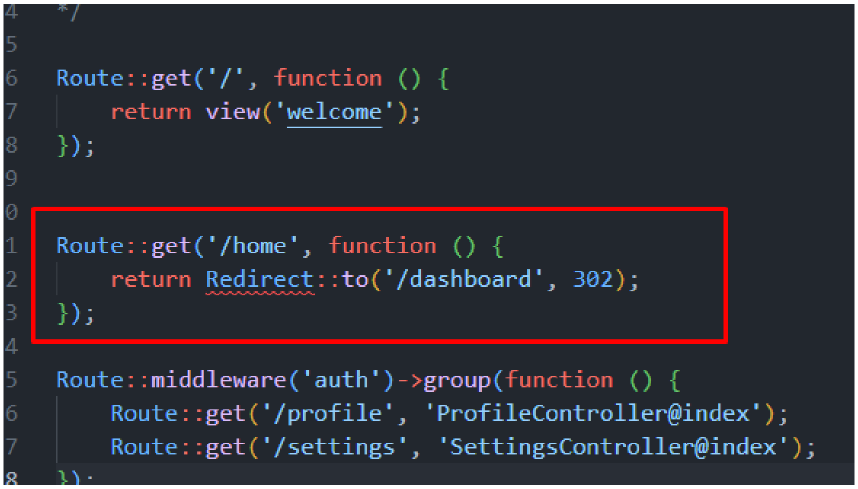
password: required, string, minimum length 8.



```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class CustomAuthController extends Controller
{

    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required|string|min:2',
            'email' => 'required|email',
            'password' => 'required|string|min:8',
        ]);

        if ($validator->fails()) {
            return response()->json(['errors' => $validator->errors()], 400);
        }

        // Registration logic here

        return response()->json(['message' => 'Registration successful'], 200);
    }
```

Task (2) Request Redirect

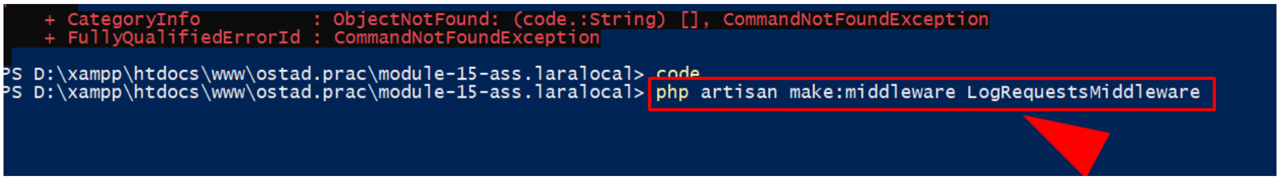
Create a route /home that redirects to /dashboard using a 302 redirect.



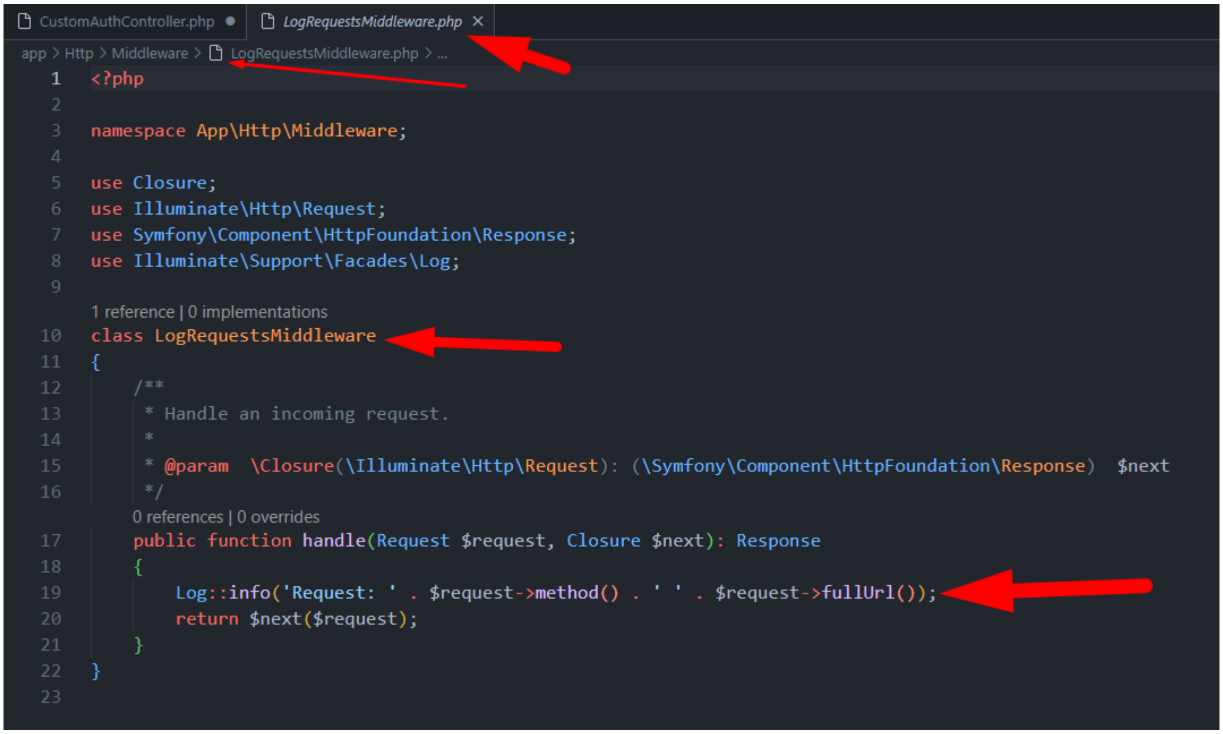
```
 */

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


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

Route::middleware('auth')->group(function () {
    Route::get('/profile', 'ProfileController@index');
    Route::get('/settings', 'SettingsController@index');
});
```

Task (3): Global Middleware

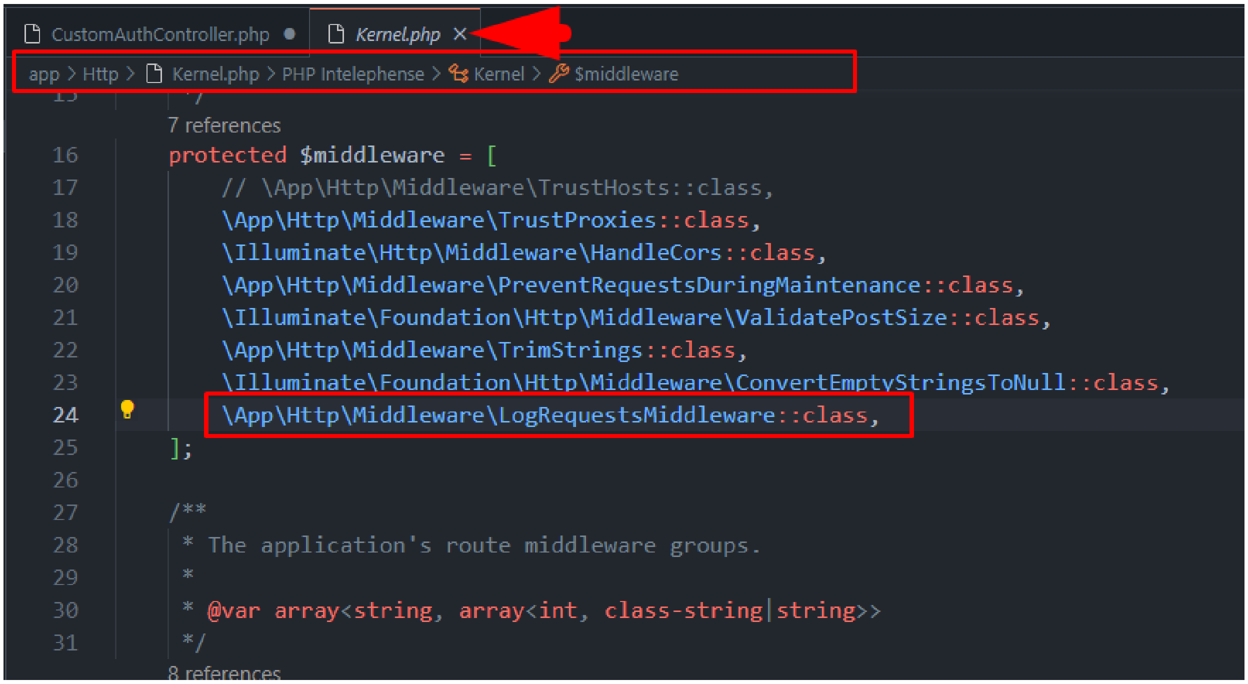
Create a global middleware that logs the request method and URL for every incoming request. Log the information to the Laravel log file.



```
    + CategoryInfo          : ObjectNotFound: (code.:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

PS D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal> code
PS D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal> php artisan make:middleware LogRequestsMiddleware
```


CustomAuthController.php
LogRequestsMiddleware.php
app > Http > Middleware > LogRequestsMiddleware.php > ...
1 reference | 0 implementations
0 references | 0 overrides

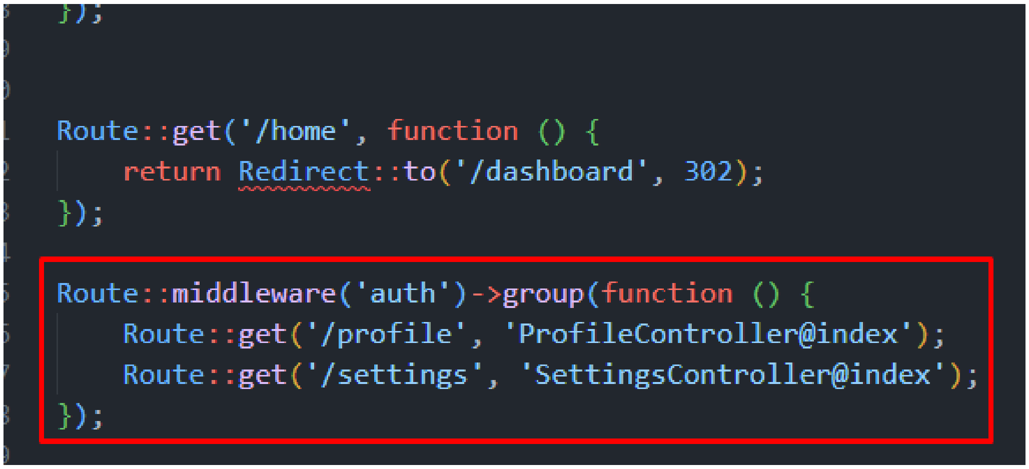
```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
use Illuminate\Support\Facades\Log;

class LogRequestsMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        Log::info('Request: ' . $request->method() . ' ' . $request->fullUrl());
        return $next($request);
    }
}
```


CustomAuthController.php
Kernel.php
app > Http > Kernel.php > PHP Intelephense > Kernel > $middleware
7 references
8 references

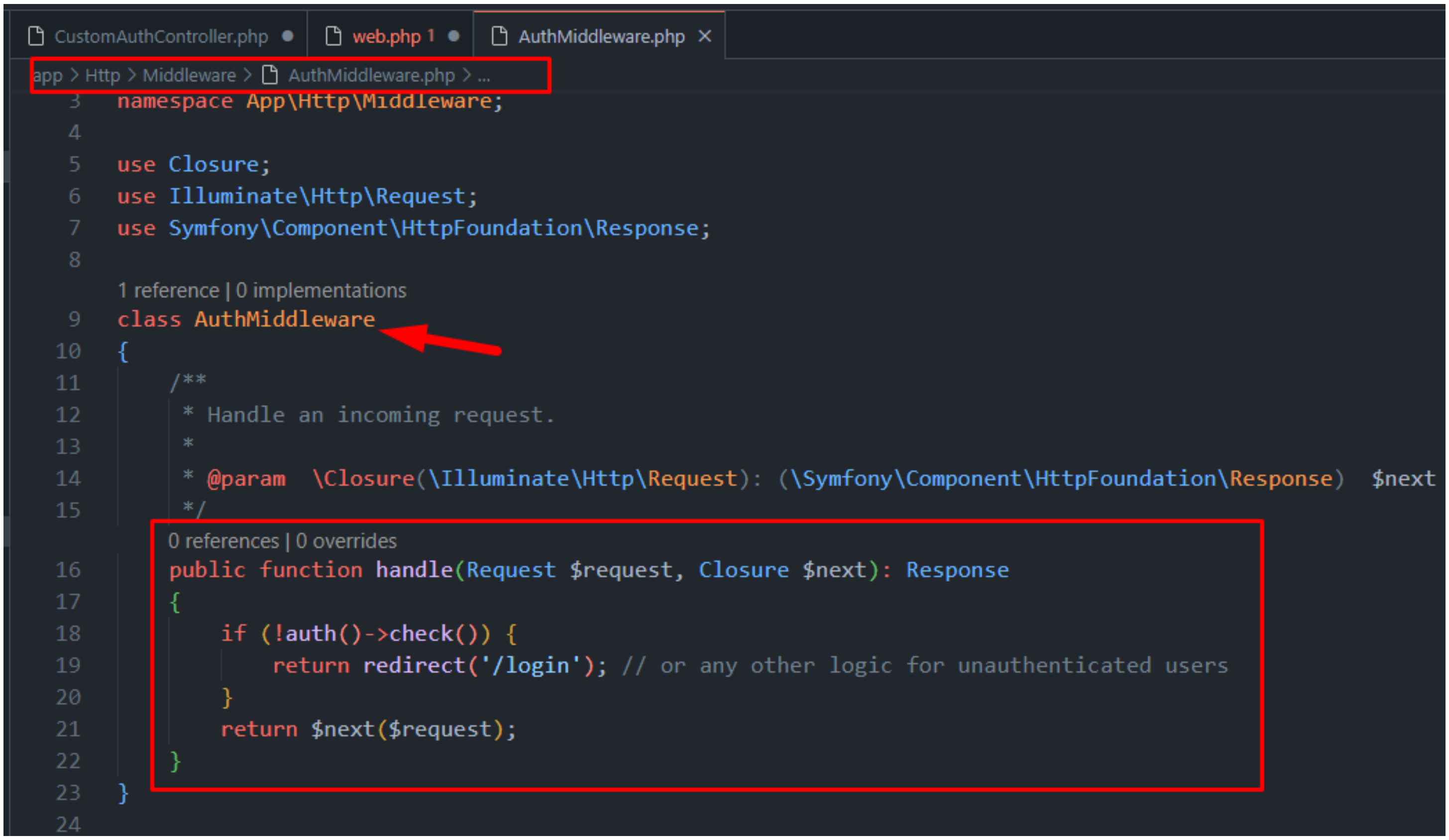
```php
    protected $middleware = [
        // \App\Http\Middleware\TrustHosts::class,
        \App\Http\Middleware\TrustProxies::class,
        \Illuminate\Http\Middleware\HandleCors::class,
        \App\Http\Middleware\PreventRequestsDuringMaintenance::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\LogRequestsMiddleware::class,
    ];

    /**
     * The application's route middleware groups.
     *
     * @var array<string, array<int, class-string|string>>
     */
```

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.



```php
});


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

Route::middleware('auth')->group(function () {
    Route::get('/profile', 'ProfileController@index');
    Route::get('/settings', 'SettingsController@index');
});
```

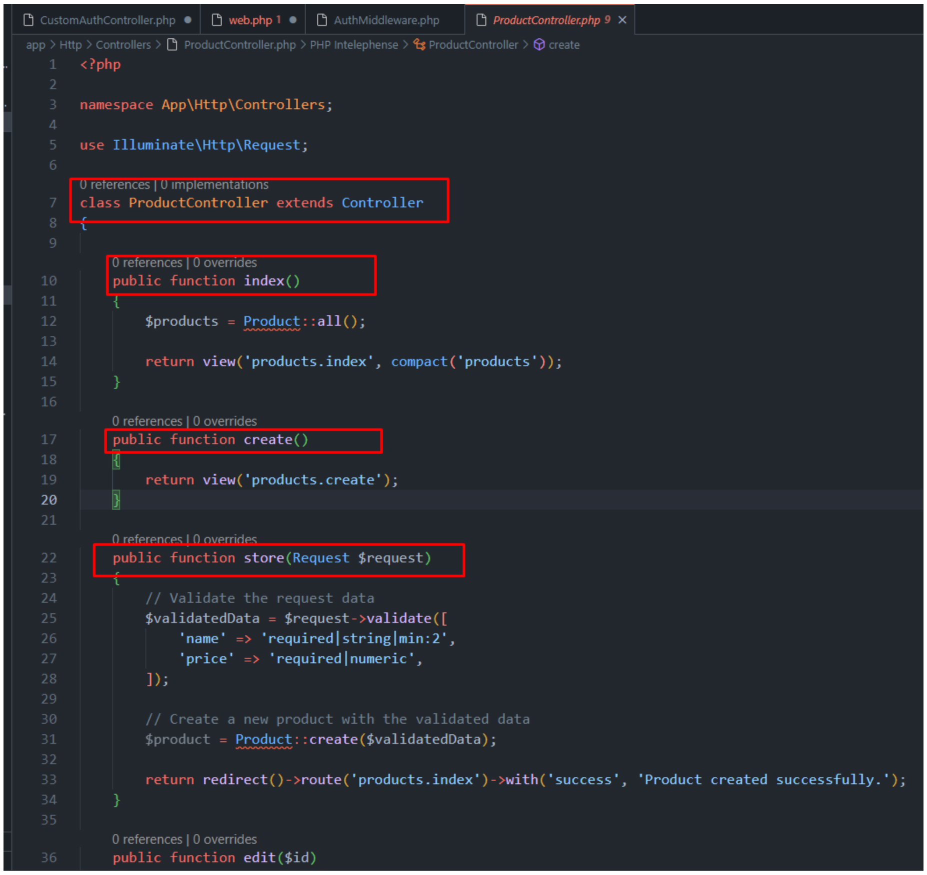


CustomAuthController.php
web.php 1
AuthMiddleware.php ×
app > Http > Middleware > AuthMiddleware.php > ...
1 reference | 0 implementations
0 references | 0 overrides

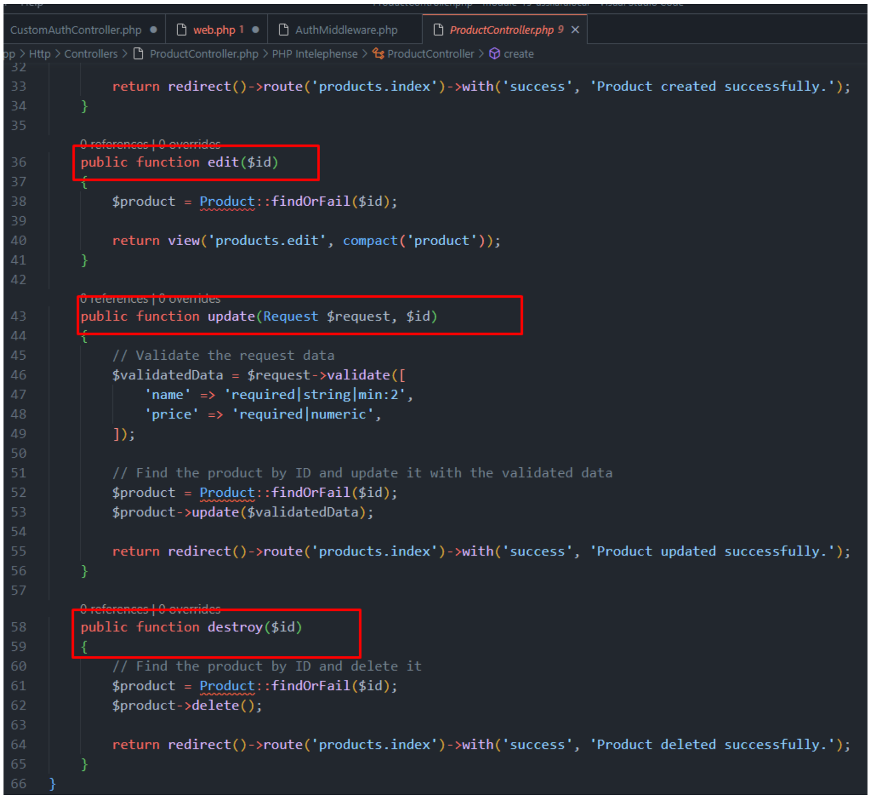
```php
namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class AuthMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        if (!auth()->check()) {
            return redirect('/login'); // or any other logic for unauthenticated users
        }
        return $next($request);
    }
}
```

Task (5): Product Controller

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


CustomAuthController.php
web.php 1
AuthMiddleware.php
ProductController.php 9
app > Http > Controllers > ProductController.php > PHP Intelephense > ProductController > create
0 references | 0 implementations
0 references | 0 overrides

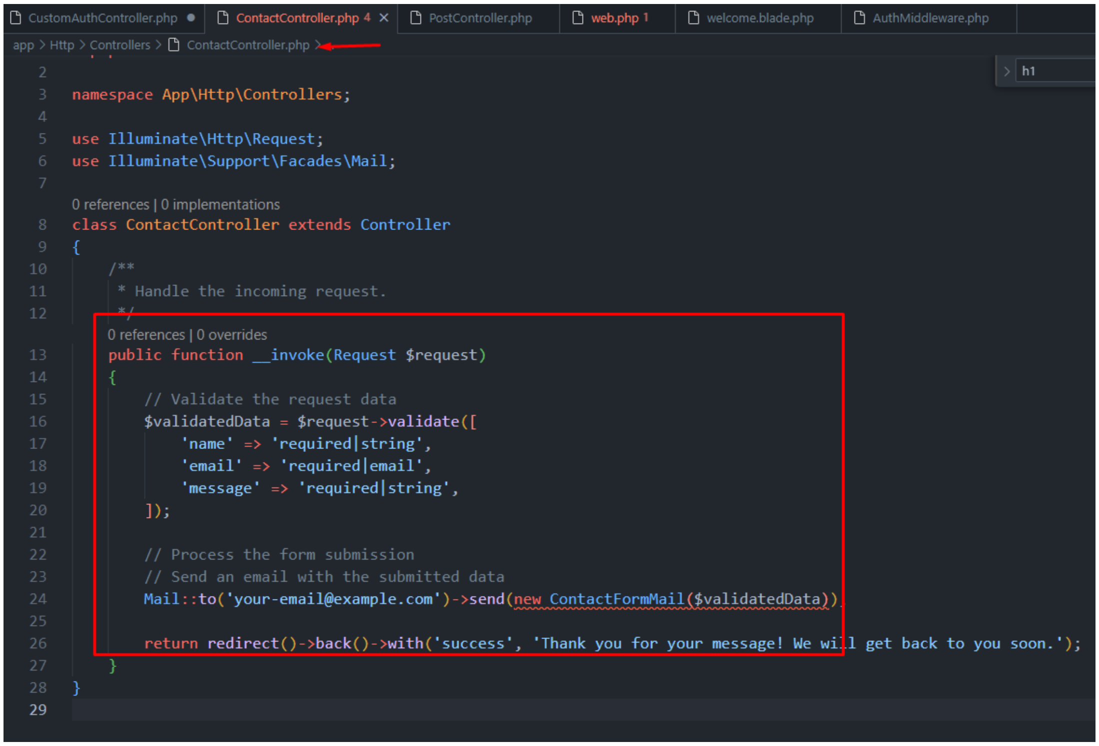
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProductController extends Controller
{

    public function index()
    {
        $products = Product::all();

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

    public function create()
    {
        return view('products.create');
    }

    public function store(Request $request)
    {
        // Validate the request data
        $validatedData = $request->validate([
            'name' => 'required|string|min:2',
            'price' => 'required|numeric',
        ]);

        // Create a new product with the validated data
        $product = Product::create($validatedData);

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

    public function edit($id)
```


CustomAuthController.php
web.php 1
AuthMiddleware.php
ProductController.php 9
pp > Http > Controllers > ProductController.php > PHP Intelephense > ProductController > create
0 references | 0 overrides

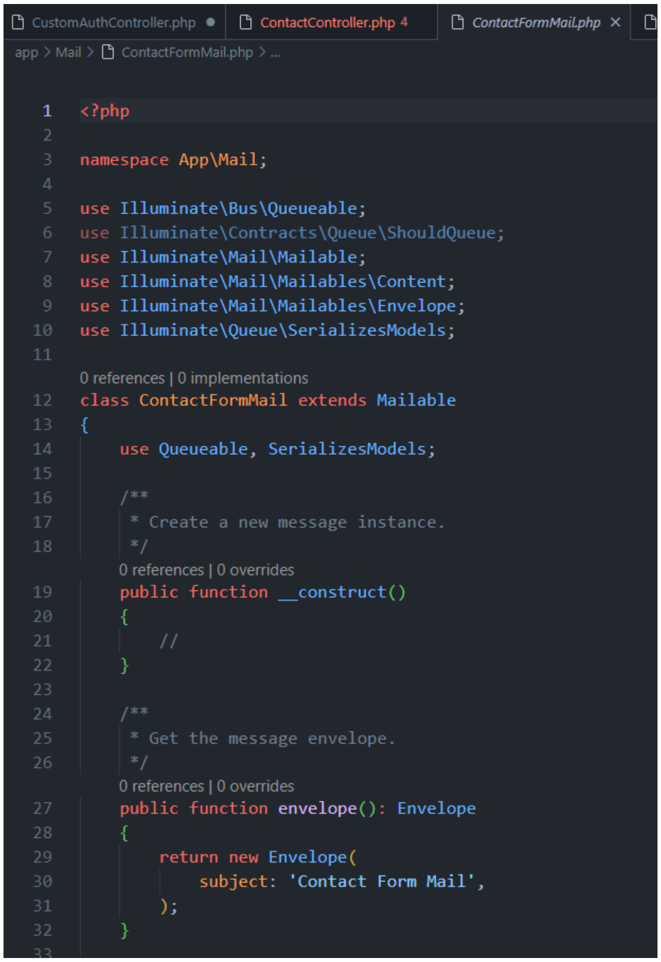
```php
        return redirect()->route('products.index')->with('success', 'Product created successfully.');
    }

    public function edit($id)
    {
        $product = Product::findOrFail($id);

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

    public function update(Request $request, $id)
    {
        // Validate the request data
        $validatedData = $request->validate([
            'name' => 'required|string|min:2',
            'price' => 'required|numeric',
        ]);

        // Find the product by ID and update it with the validated data
        $product = Product::findOrFail($id);
        $product->update($validatedData);

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

    public function destroy($id)
    {
        // Find the product by ID and delete it
        $product = Product::findOrFail($id);
        $product->delete();

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

Task (6): Single Action Controller

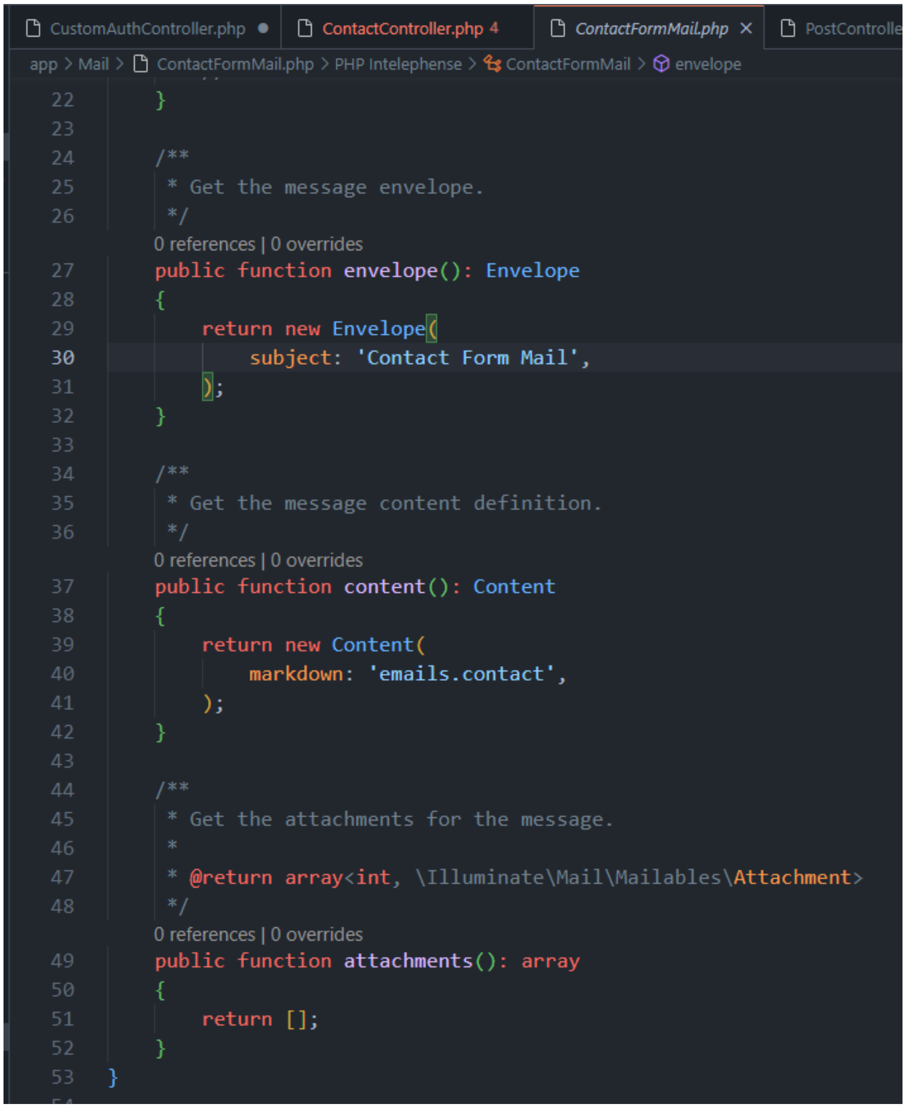
Create a single action controller called ContactController that handles a contact form submission. Implement the __invoke() method to process the form submission and send an email to a predefined address with the submitted data.



```php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Mail;

class ContactController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        // Validate the request data
        $validatedData = $request->validate([
            'name' => 'required|string',
            'email' => 'required|email',
            'message' => 'required|string',
        ]);

        // Process the form submission
        // Send an email with the submitted data
        Mail::to('your-email@example.com')->send(new ContactFormMail($validatedData));

        return redirect()->back()->with('success', 'Thank you for your message! We will get back to you soon.');
    }
}

```


CustomAuthController.php
ContactController.php 4
ContactFormMail.php
app > Mail > ContactFormMail.php > ...
0 references | 0 implementations
0 references | 0 overrides
0 references | 0 overrides

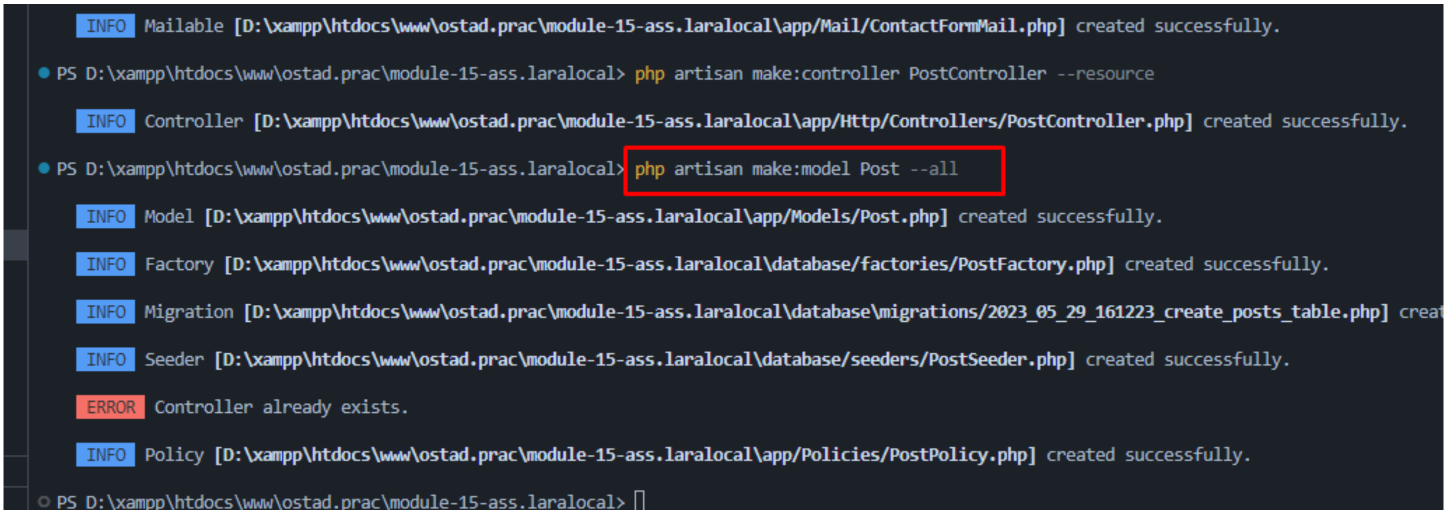
```php
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Mail\Mailables\Content;
use Illuminate\Mail\Mailables\Envelope;
use Illuminate\Queue\SerializesModels;

class ContactFormMail extends Mailable
{
    use Queueable, SerializesModels;

    /**
     * Create a new message instance.
     */
    public function __construct()
    {
        //
    }

    /**
     * Get the message envelope.
     */
    public function envelope(): Envelope
    {
        return new Envelope(
            subject: 'Contact Form Mail',
        );
    }
```


CustomAuthController.php
ContactController.php 4
ContactFormMail.php
PostControlle
app > Mail > ContactFormMail.php > PHP Intelephense > ContactFormMail > envelope

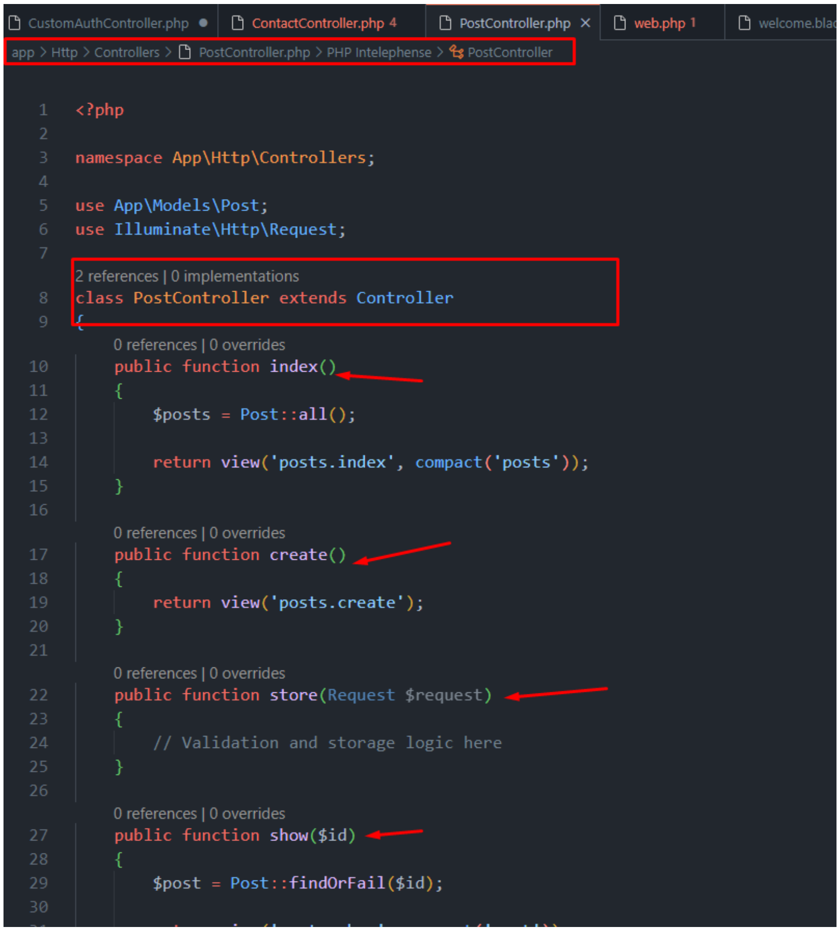
```
    }

    /**
     * Get the message envelope.
     */
    0 references | 0 overrides
    public function envelope(): Envelope
    {
        return new Envelope(
            subject: 'Contact Form Mail',
        );
    }

    /**
     * Get the message content definition.
     */
    0 references | 0 overrides
    public function content(): Content
    {
        return new Content(
            markdown: 'emails.contact',
        );
    }

    /**
     * Get the attachments for the message.
     *
     * @return array<int, \Illuminate\Mail\Mailables\Attachment>
     */
    0 references | 0 overrides
    public function attachments(): array
    {
        return [];
    }
}
```

Task (7): Resource Controller

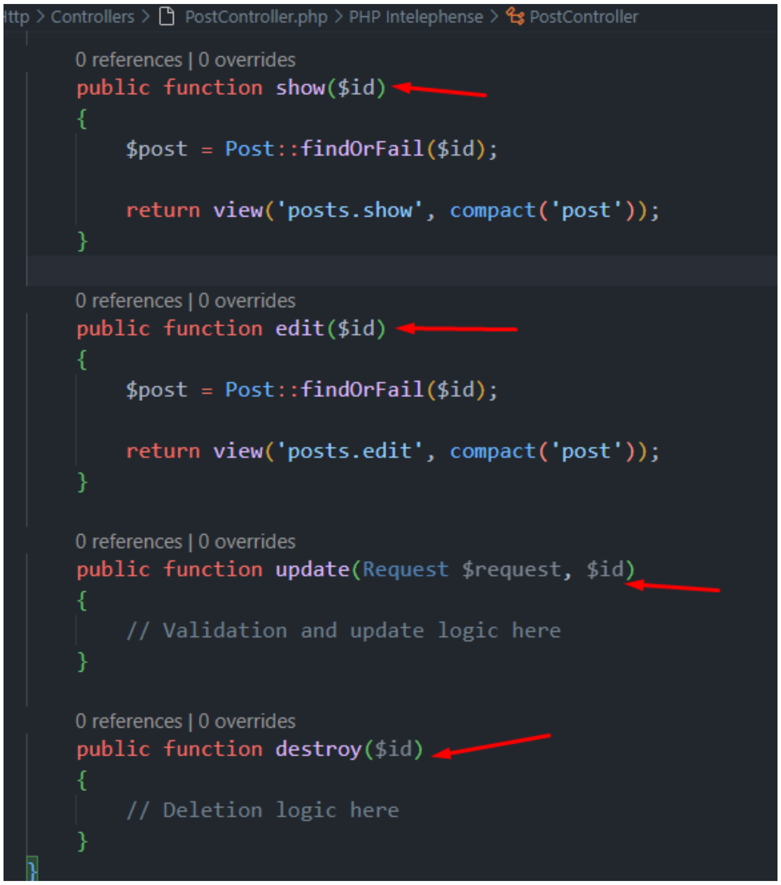
Create a resource controller called PostController that handles CRUD operations for a resource called Post. Ensure that the controller provides the necessary methods for the resourceful routing conventions in Laravel.



```
   INFO  Mailable [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\app/Mail/ContactFormMail.php] created successfully.

PS D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal> php artisan make:controller PostController --resource

   INFO  Controller [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\app/Http/Controllers/PostController.php] created successfully.

PS D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal> php artisan make:model Post --all

   INFO  Model [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\app/Models/Post.php] created successfully.

   INFO  Factory [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\database/factories/PostFactory.php] created successfully.

   INFO  Migration [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\database\migrations/2023_05_29_161223_create_posts_table.php] creat

   INFO  Seeder [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\database/seeders/PostSeeder.php] created successfully.

   ERROR  Controller already exists.

   INFO  Policy [D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal\app/Policies/PostPolicy.php] created successfully.

PS D:\xampp\htdocs\www\ostad.prac\module-15-ass.laralocal>
```



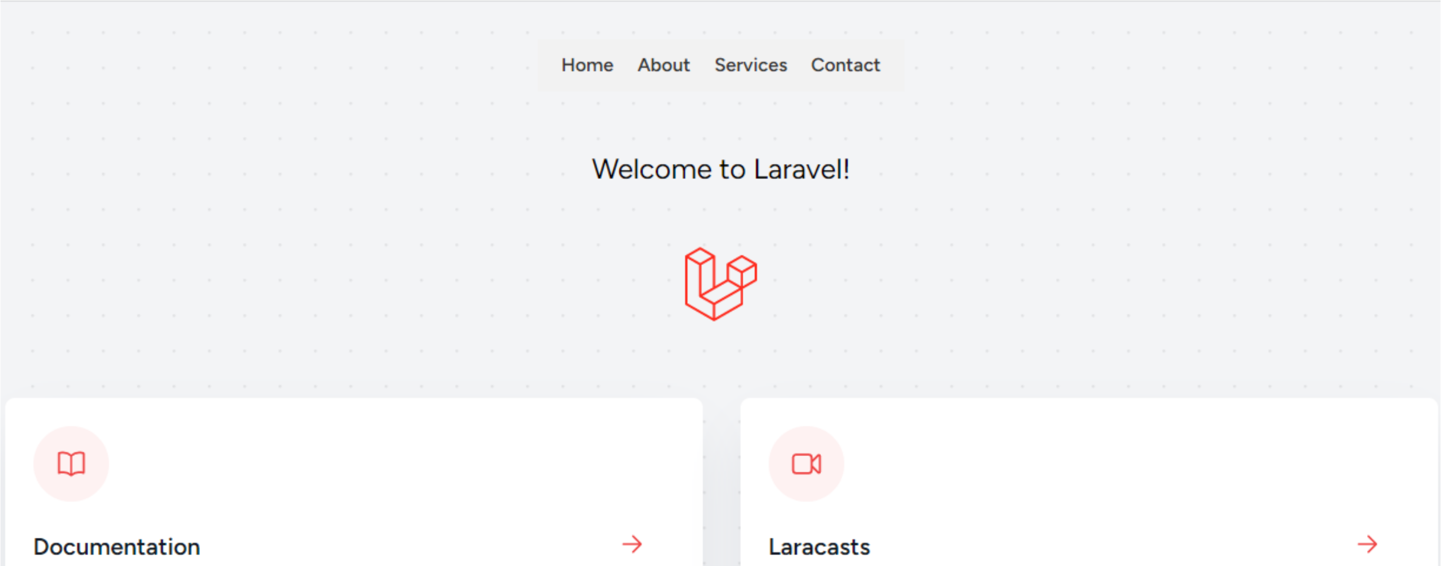
```php
<?php

namespace App\Http\Controllers;

use App\Models\Post;
use Illuminate\Http\Request;

2 references | 0 implementations
class PostController extends Controller
{
    0 references | 0 overrides
    public function index()
    {
        $posts = Post::all();

        return view('posts.index', compact('posts'));
    }

    0 references | 0 overrides
    public function create()
    {
        return view('posts.create');
    }

    0 references | 0 overrides
    public function store(Request $request)
    {
        // Validation and storage logic here
    }

    0 references | 0 overrides
    public function show($id)
    {
        $post = Post::findOrFail($id);

```

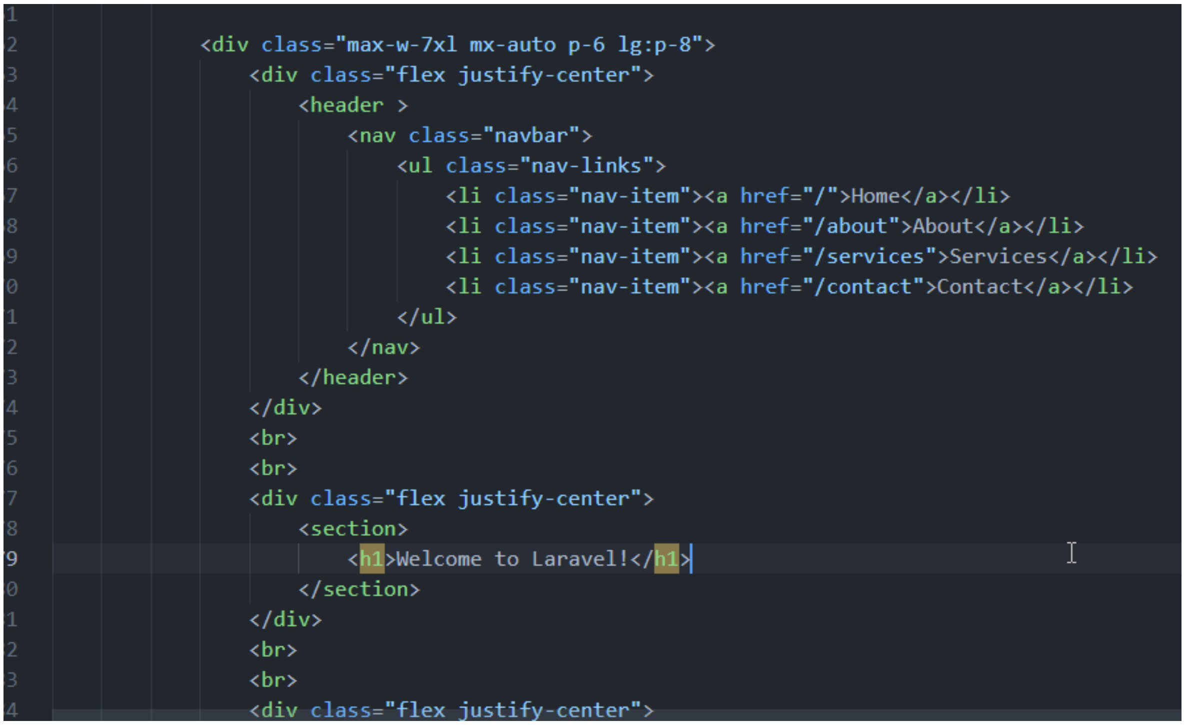

ttp > Controllers > PostController.php > PHP Intelephense > PostController

```php
    0 references | 0 overrides
    public function show($id)
    {
        $post = Post::findOrFail($id);

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

    0 references | 0 overrides
    public function edit($id)
    {
        $post = Post::findOrFail($id);

        return view('posts.edit', compact('post'));
    }

    0 references | 0 overrides
    public function update(Request $request, $id)
    {
        // Validation and update logic here
    }

    0 references | 0 overrides
    public function destroy($id)
    {
        // Deletion logic here
    }
}
```

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!".





```

            <div class="max-w-7xl mx-auto p-6 lg:p-8">
                <div class="flex justify-center">
                    <header >
                        <nav class="navbar">
                            <ul class="nav-links">
                                <li class="nav-item"><a href="/">Home</a></li>
                                <li class="nav-item"><a href="/about">About</a></li>
                                <li class="nav-item"><a href="/services">Services</a></li>
                                <li class="nav-item"><a href="/contact">Contact</a></li>
                            </ul>
                        </nav>
                    </header>
                </div>
                <br>
                <br>
                <div class="flex justify-center">
                    <section>
                        <h1>Welcome to Laravel!</h1>
                    </section>
                </div>
                <br>
                <br>
                <div class="flex justify-center">
```