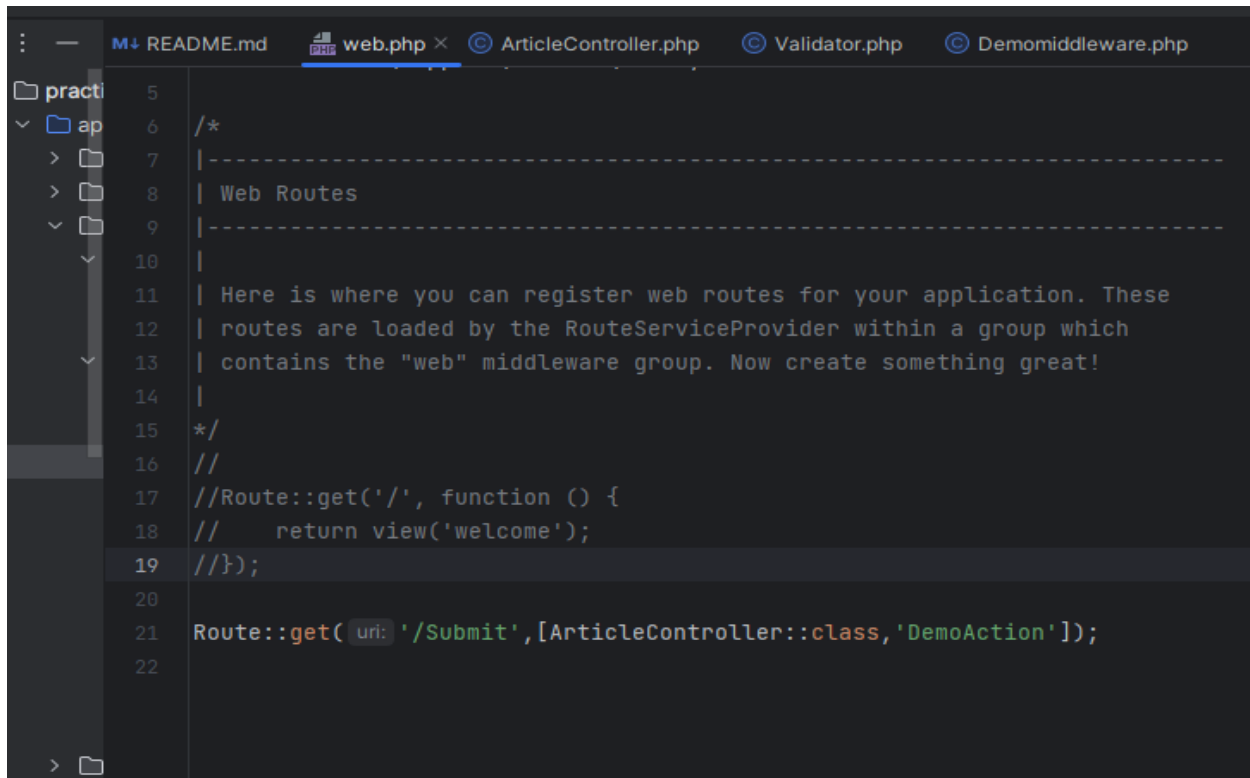
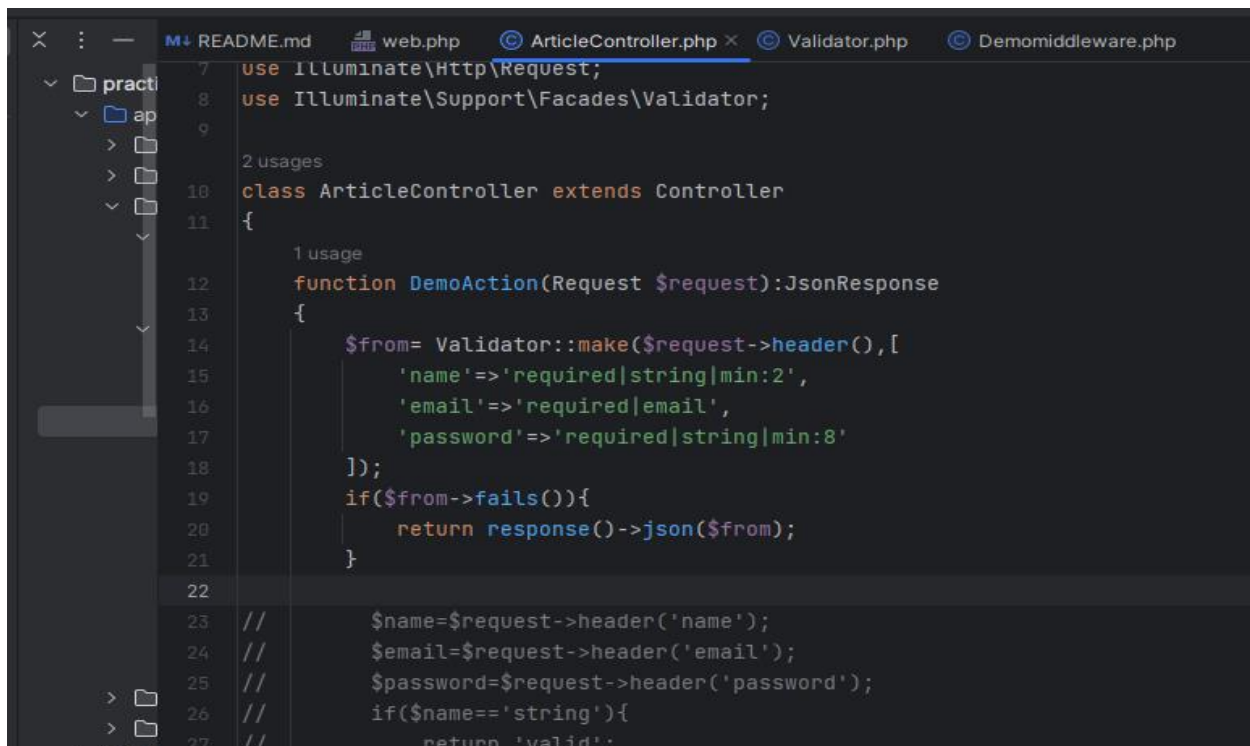


1. Implement request validation for a registration form that contains the following fields: name, email, and password. Validate the following rules:
name: required, string, minimum length 2.
email: required, valid email format.
password: required, string, minimum length 8.

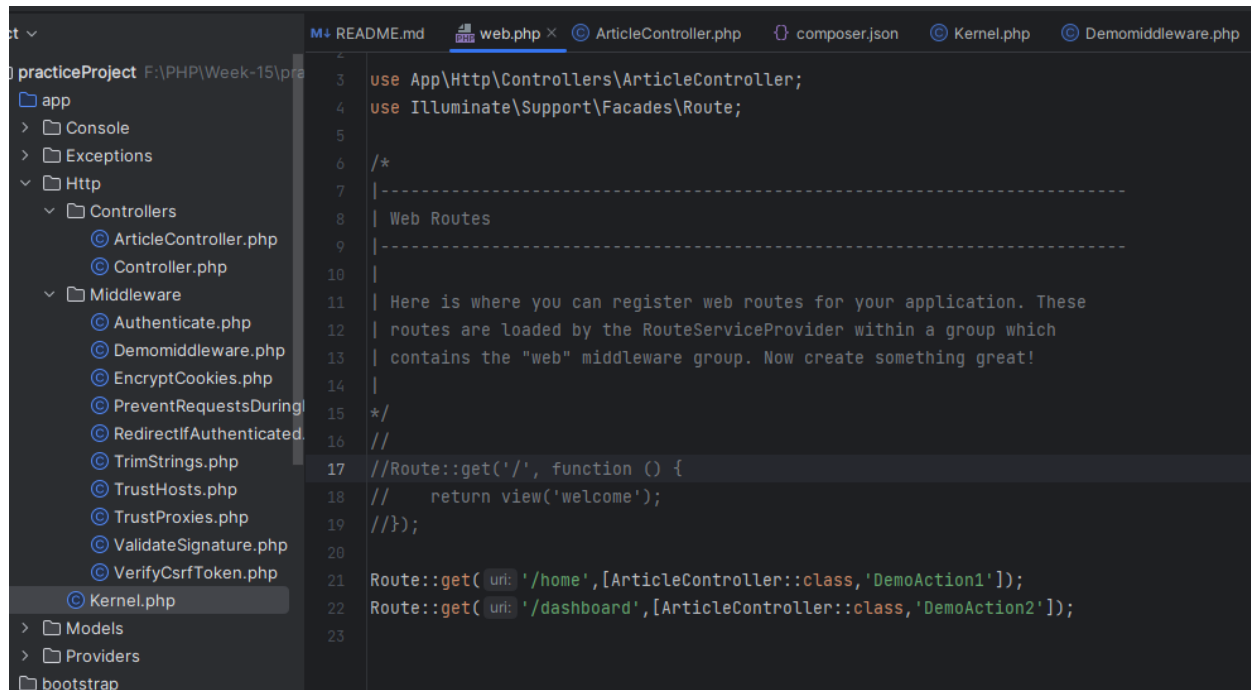


```
5
6 /*
7 |-----
8 | Web Routes
9 |-----
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 */
16 //
17 //Route::get('/', function () {
18 //    return view('welcome');
19 //});
20
21 Route::get(uri: '/Submit',[ArticleController::class,'DemoAction']);
22
```

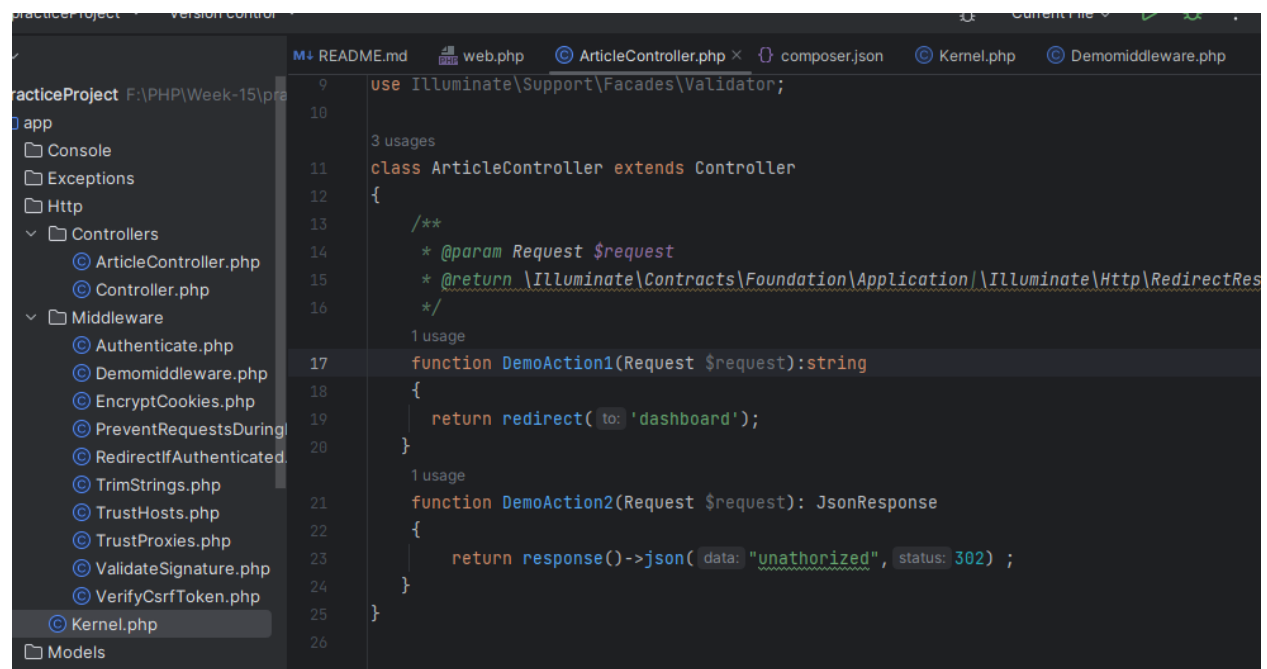


```
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Validator;
9
10 2 usages
11 class ArticleController extends Controller
12 {
13     1 usage
14     function DemoAction(Request $request):JsonResponse
15     {
16         $from= Validator::make($request->header(),[
17             'name'=>'required|string|min:2',
18             'email'=>'required|email',
19             'password'=>'required|string|min:8'
20         ]);
21         if($from->fails()){
22             return response()->json($from);
23         }
24
25         // $name=$request->header('name');
26         // $email=$request->header('email');
27         // $password=$request->header('password');
28         // if($name=='string'){
29         //     return 'valid';
30         // }
31     }
32 }
```

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

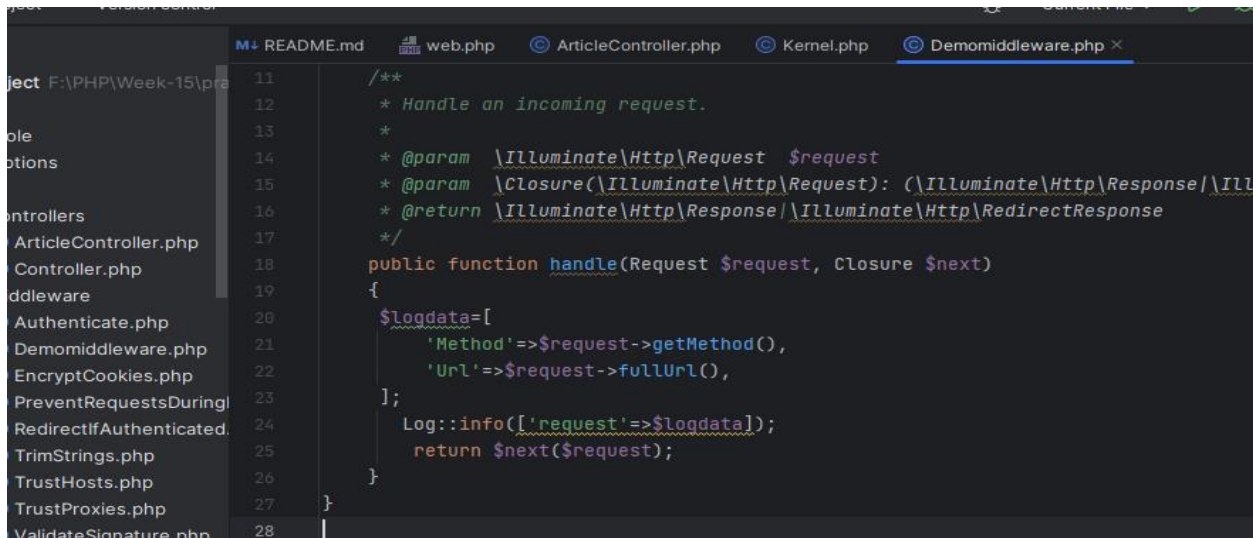


```
1 use App\Http\Controllers\ArticleController;
2 use Illuminate\Support\Facades\Route;
3
4 /*
5 |-----
6 | Web Routes
7 |-----
8 |
9 | Here is where you can register web routes for your application. These
10 | routes are loaded by the RouteServiceProvider within a group which
11 | contains the "web" middleware group. Now create something great!
12 |
13 */
14 //
15 //Route::get('/', function () {
16 //    return view('welcome');
17 //});
18
19 Route::get(uri: '/home', [ArticleController::class, 'DemoAction1']);
20 Route::get(uri: '/dashboard', [ArticleController::class, 'DemoAction2']);
```

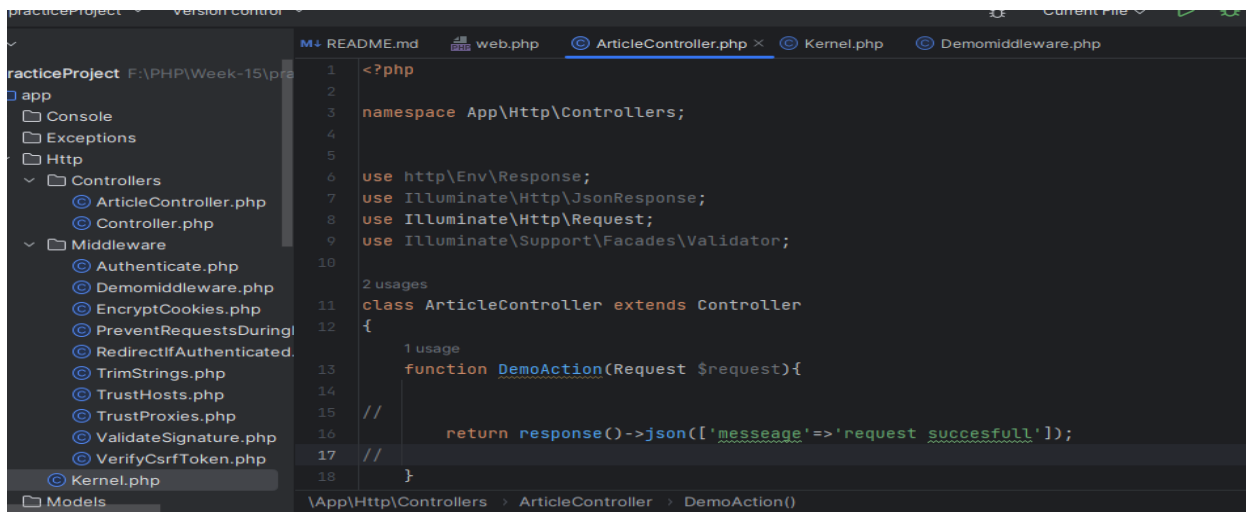


```
1 use Illuminate\Support\Facades\Validator;
2
3 class ArticleController extends Controller
4 {
5     /**
6      * @param Request $request
7      * @return \Illuminate\Contracts\Foundation\Application|\Illuminate\Http\RedirectResponse
8      */
9     function DemoAction1(Request $request): string
10     {
11         return redirect(to: 'dashboard');
12     }
13
14     function DemoAction2(Request $request): JsonResponse
15     {
16         return response()->json(data: "unathorized", status: 302);
17     }
18 }
```

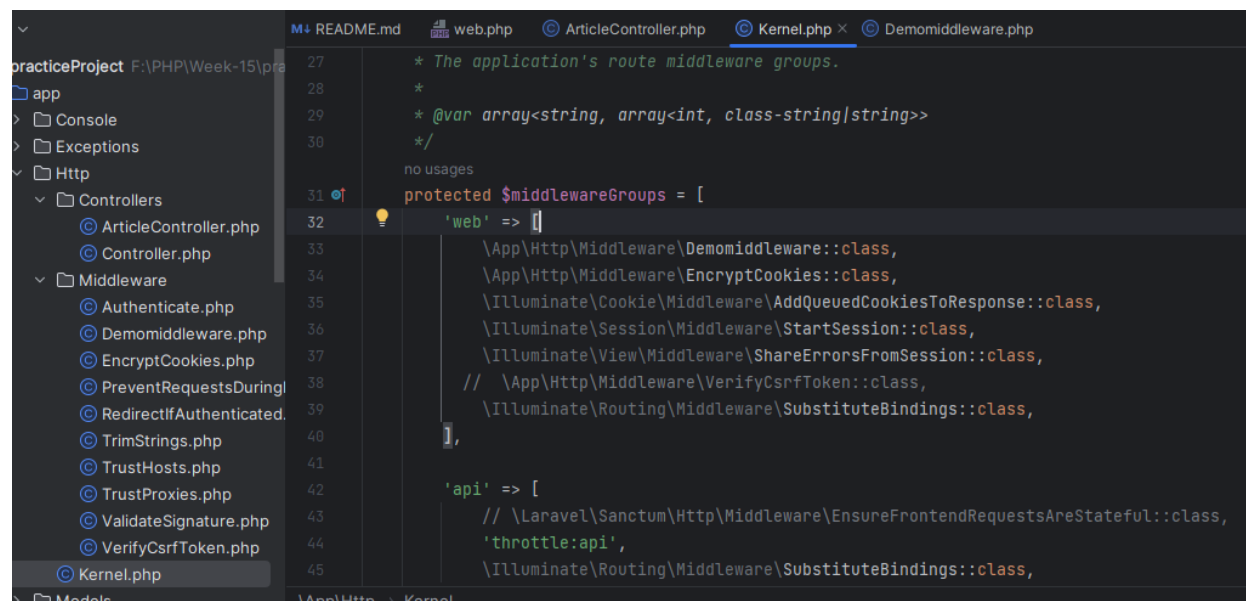
3. Create a global middleware that logs the request method and URL for every incoming request. Log the information to the Laravel log file



```
11  /**
12   * Handle an incoming request.
13   *
14   * @param  \Illuminate\Http\Request $request
15   * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
16   * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
17   */
18  public function handle(Request $request, Closure $next)
19  {
20      $logdata=[
21          'Method'=>$request->getMethod(),
22          'Url'=>$request->fullUrl(),
23      ];
24      Log::info(['request'=>$logdata]);
25      return $next($request);
26  }
27  }
28  }
```



```
1  <?php
2
3  namespace App\Http\Controllers;
4
5
6
7  use http\Env\Response;
8  use Illuminate\Http\JsonResponse;
9  use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Validator;
11
12 2 usages
13 class ArticleController extends Controller
14 {
15     1 usage
16     function DemoAction(Request $request){
17
18         //
19         return response()->json(['message'=>'request successful']);
20     }
21 }
```



```
27  * The application's route middleware groups.
28  *
29  * @var array<string, array<int, class-string|string>>
30  */
31  no usages
32  protected $middlewareGroups = [
33      'web' => [
34          \App\Http\Middleware\Demomiddleware::class,
35          \App\Http\Middleware\EncryptCookies::class,
36          \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
37          \Illuminate\Session\Middleware\StartSession::class,
38          \Illuminate\View\Middleware\ShareErrorsFromSession::class,
39          // \App\Http\Middleware\VerifyCsrfToken::class,
40          \Illuminate\Routing\Middleware\SubstituteBindings::class,
41      ],
42      'api' => [
43          // \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
44          'throttle:api',
45          \Illuminate\Routing\Middleware\SubstituteBindings::class,
```

4. 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.

```

53  *
54  * @var array<string, class-string|string>
55  */
56  no usages
57  protected static $routeMiddleware = [
58      'demo' => \App\Http\Middleware\Demomiddleware::class,
59      'auth' => \App\Http\Middleware\Authenticate::class,
60      'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
61      'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
62      'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
63      'can' => \Illuminate\Auth\Middleware\Authorize::class,
64      'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
65      'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
66      'signed' => \App\Http\Middleware\ValidateSignature::class,
67      'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
68      'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
69  ];

```

```

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  Route::middleware(['demo'])->group(function () {
22      Route::get('uri: '/profile', [ArticleController::class, 'DemoAction']);
23      Route::get('uri: '/setting', [ArticleController::class, 'DemoAction']);
24  });
25
26

```

```

10  {
11      /**
12       * Handle an incoming request.
13       *
14       * @param \Illuminate\Http\Request $request
15       * @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)
16       * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
17       */
18      public function handle(Request $request, Closure $next)
19      {
20          $key=$request->header( 'key: 'Api-key');
21          if($key=="xyz123"){
22              return $next($request);
23          }else{
24              return response()->json( data: 'unauthorized', status: 402);
25          }
26      }
27  }
28

```

```

8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\Validator;
10
11 class ArticleController extends Controller
12 {
13     function DemoAction(Request $request){
14
15         //
16         return response()->json(['message'=>'request succesfull']);
17     }
18 }
19
20

```

5. 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.

```

4 use App\Http\Controllers\ProductController;
5 use Illuminate\Support\Facades\Route;
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::resource( name: 'Product', controller: ProductController::class);
23
24

```

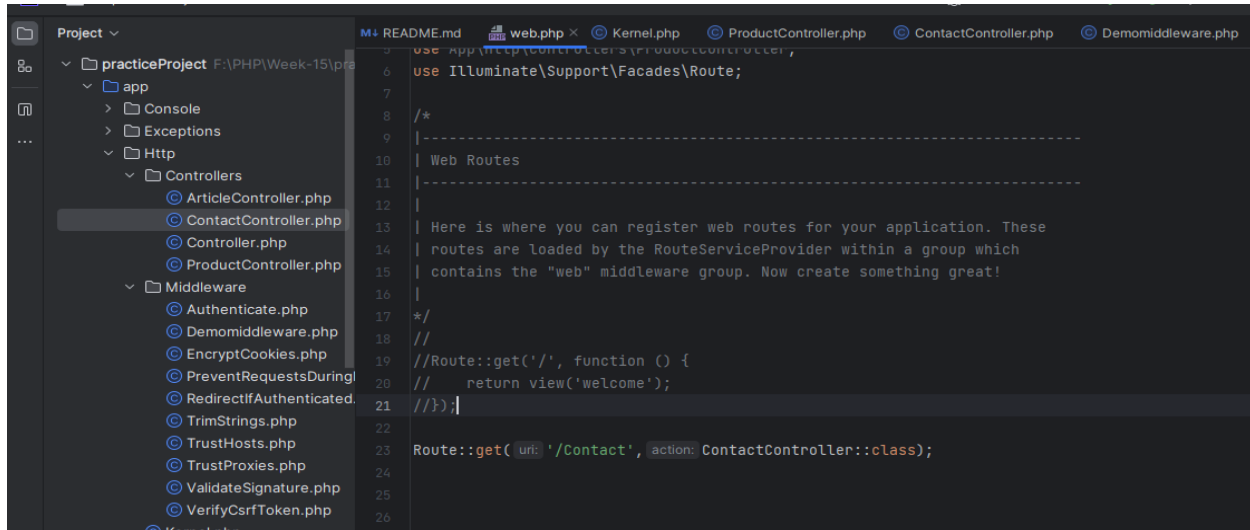
```
7 class ProductController extends Controller
8 {
9
10     no usages
11     public function index()
12     {
13         return "index";
14     }
15     no usages
16     public function create()
17     {
18         return "create";
19     }
20     no usages
21     public function store(Request $request)
22     {
23         return "store";
24     }
25     no usages
26     public function show($id)
27     {
28         return "show";
29     }
30     no usages
31     public function edit($id)
32     {
33         return "edit";
34     }
35 }
```

\App\Http\Controllers > ProductController > destroy()

```
22 public function show($id)
23 {
24     return "show";
25 }
26 no usages
27 public function edit($id)
28 {
29     return "edit";
30 }
31 no usages
32 public function update(Request $request, $id)
33 {
34     return "update";
35 }
36 no usages
37 public function destroy($id)
38 {
39     return "destroy";
40 }
```

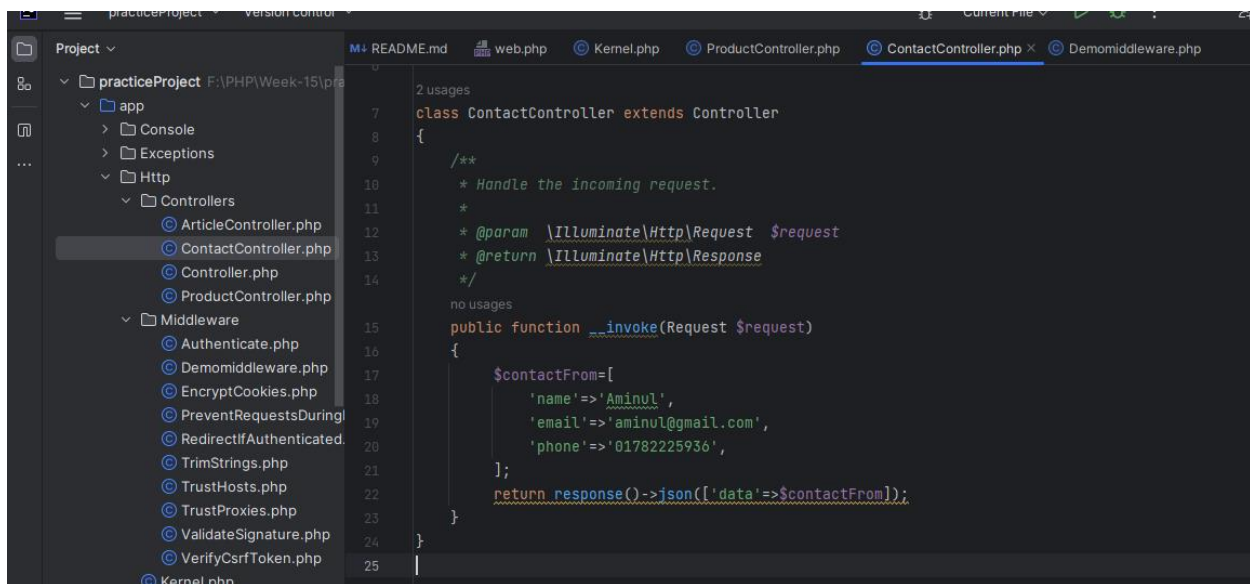
\App\Http\Controllers > ProductController > destroy()

6. 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.



The screenshot shows an IDE with a project named 'practiceProject'. The file explorer on the left shows the directory structure: 'app' > 'Http' > 'Controllers'. The 'ContactController.php' file is selected. The main editor shows the 'web.php' file with the following code:

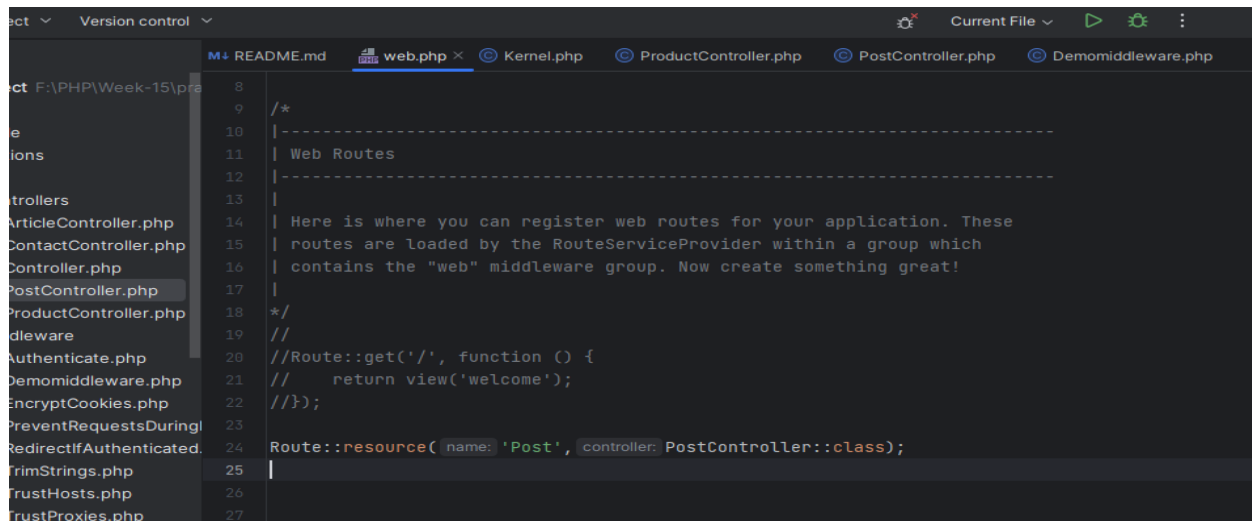
```
1 use Illuminate\Http\Controllers\Controller;
2 use Illuminate\Routing\Router;
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----
7 | Web Routes
8 |-----
9 |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a 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::get('/Contact', function () {
21     //
22 });
```



The screenshot shows the 'ContactController.php' file in the IDE. The file explorer on the left shows the directory structure: 'app' > 'Http' > 'Controllers'. The 'ContactController.php' file is selected. The main editor shows the code for the 'ContactController' class:

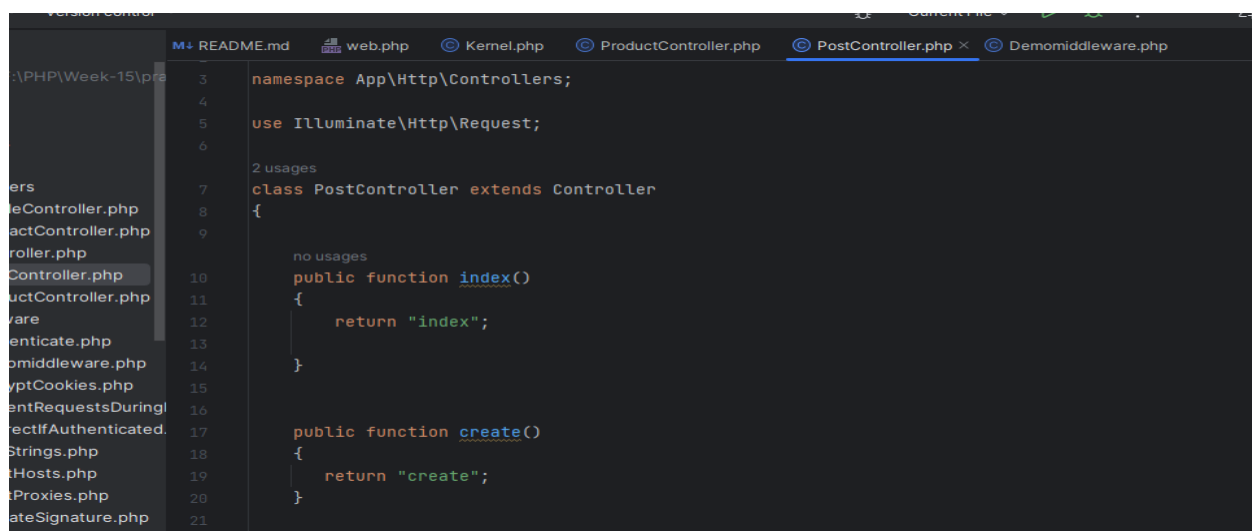
```
1 class ContactController extends Controller
2 {
3     /**
4      * Handle the incoming request.
5      *
6      * @param \Illuminate\Http\Request $request
7      * @return \Illuminate\Http\Response
8      */
9     public function __invoke(Request $request)
10     {
11         $contactFrom=[
12             'name'=>'Aminul',
13             'email'=>'aminul@gmail.com',
14             'phone'=>'01782225936',
15         ];
16         return response()->json(['data'=>$contactFrom]);
17     }
18 }
```

7. 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.



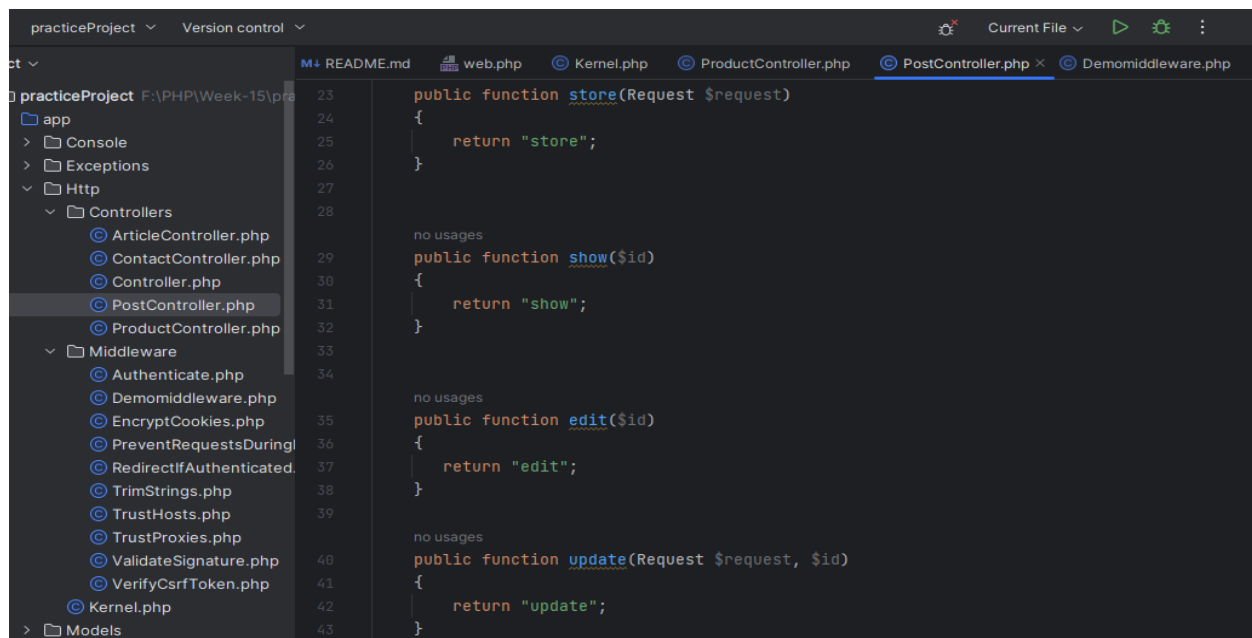
This screenshot shows the `web.php` file in an IDE. The file contains comments and a single route definition. The route is a resource route for 'Post' using the `PostController`.

```
8
9  /*
10 |-----|
11 | Web Routes
12 |-----|
13 |
14 | Here is where you can register web routes for your application. These
15 | routes are loaded by the RouteServiceProvider within a group which
16 | contains the "web" middleware group. Now create something great!
17 |
18 */
19 //
20 //Route::get('/', function () {
21 //    return view('welcome');
22 //});
23
24 Route::resource('name: 'Post', controller: PostController::class);
25
26
27
```



This screenshot shows the `PostController.php` file in an IDE. The controller is in the `App\Http\Controllers` namespace and extends the `Controller` class. It implements the `index()` and `create()` methods.

```
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PostController extends Controller
8 {
9
10     public function index()
11     {
12         return "index";
13     }
14
15     public function create()
16     {
17         return "create";
18     }
19
20
21
```



This screenshot shows the `PostController.php` file in an IDE, displaying the full implementation of the controller. The controller implements the `store()`, `show()`, `edit()`, and `update()` methods.

```
23 public function store(Request $request)
24 {
25     return "store";
26 }
27
28
29 public function show($id)
30 {
31     return "show";
32 }
33
34
35 public function edit($id)
36 {
37     return "edit";
38 }
39
40
41 public function update(Request $request, $id)
42 {
43     return "update";
44 }
45
```



```
cryptCookies.php 44
eventRequestsDuringl 45
directIfAuthenticated,
mStrings.php 46
stHosts.php 47
stProxies.php 48
idateSignature.php 49
rifyCsrfToken.php 50
Lebr 51
```

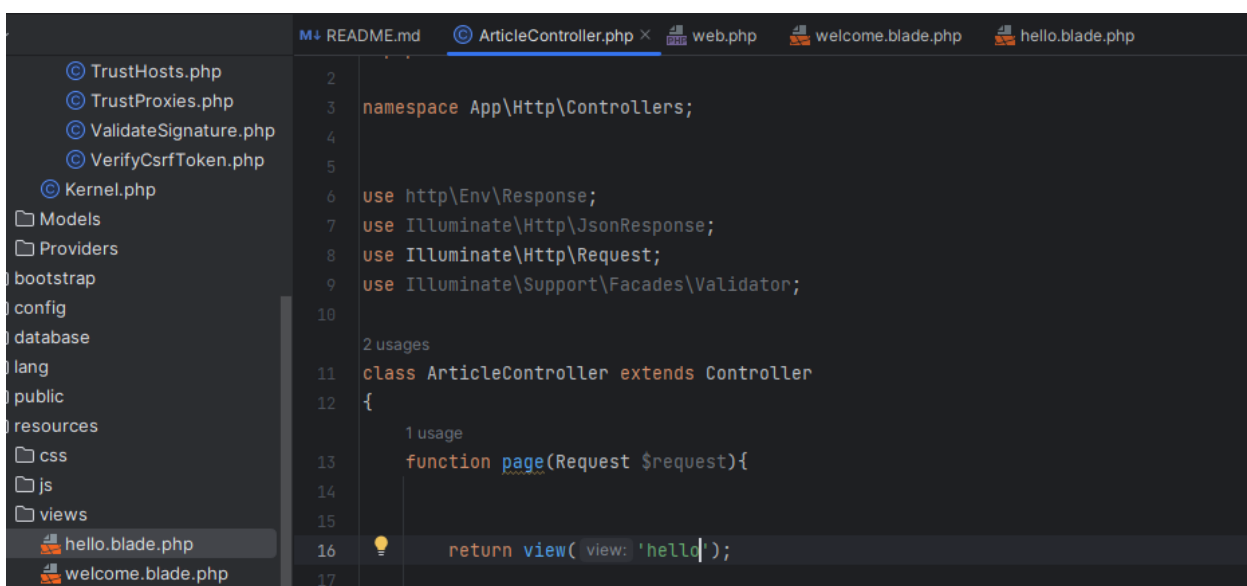
```
no usages
public function destroy($id)
{
    return "destroy";
}
```

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



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like bootstrap, config, database, lang, public, resources, and views. The code editor is open to the web.php file, showing route definitions. The routes are defined using the Route::get method. The first route is for the root URL, which returns the 'welcome' view. The second route is for the '/page' URL, which returns the 'page' view.

```
8
9
10
11 | Web Routes
12 |-----
13 |
14 | Here is where you can register web routes for your application. These
15 | routes are loaded by the RouteServiceProvider within a group which
16 | contains the "web" middleware group. Now create something great!
17 |
18 */
19 //
20 //Route::get('/', function () {
21 //    return view('welcome');
22 //});
23
24 Route::get(uri: '/', [ArticleController::class, 'page']);
25
26
27
```



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like bootstrap, config, database, lang, public, resources, and views. The code editor is open to the ArticleController.php file, showing the namespace and the page method. The page method is defined as a public function that returns the 'hello' view.

```
2
3 namespace App\Http\Controllers;
4
5
6 use http\Env\Response;
7 use Illuminate\Http\JsonResponse;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\Validator;
10
11 2 usages
12 class ArticleController extends Controller
13 {
14     1 usage
15     function page(Request $request){
16         return view( view: 'hello');
17     }
```

The image shows a code editor with a project sidebar on the left and a code editor on the right. The sidebar lists the project structure, including folders like `Models`, `Providers`, `bootstrap`, `config`, `database`, `lang`, `public`, `resources`, `css`, `js`, `views`, and `routes`. The `views` folder is expanded, showing `hello.blade.php` and `welcome.blade.php`. The code editor displays the content of `hello.blade.php`, which is a Blade template. The template includes a `<head>` section with a `<title>` and a `<style>` block. The `<body>` section contains a `<div class="navbar">` with three links: `Home`, `About`, and `Contact`. Below the navbar is a `<div class="content">` containing a `<h1>` with the text "Welcome to Laravel!".

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Welcome to Laravel</title>
5   <style>
6     .navbar {
7       background-color: #f2f2f2;
8       padding: 10px;
9     }
10    .navbar a {
11      color: #333;
12      text-decoration: none;
13      margin-right: 10px;
14    }
15    .content {
16      margin-top: 20px;
17      text-align: center;
18    }
19  </style>
20 </head>
21
22 <body>
23   <div class="navbar">
24     <a href="#">Home</a>
25     <a href="#">About</a>
26     <a href="#">Contact</a>
27   </div>
28   <div class="content">
29     <h1>Welcome to Laravel!</h1>
30   </div>
31 </body>
32 </html>
```