

Assignment 15

Task1:

```
use Illuminate\Support\Facades\Validator;

public function store(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);
    }
}
```

Task2:

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

Task3:

```
<?php

namespace App\Http\Middleware;

use Closure;
```

```

use Illuminate\Support\Facades\Log;

class LogRequestMiddleware
{
    public function handle($request, Closure $next)
    {
        Log::info('Request: ' . $request->method() . ' ' . $request->url());

        return $next($request);
    }
}

protected $middleware = [
    // ...
    \App\Http\Middleware\LogRequestMiddleware::class,
];

```

Task4:

```

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

```

Task 5: Controller

```
php artisan make:controller ProductController --resource
```

```
<?php
```

```
namespace App\Http\Controllers;

use App\Models\Product;
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)
    {
        $request->validate([
            'name' => 'required',
            'price' => 'required|numeric',
        ]);

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

        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)
    {
        $request->validate([
            'name' => 'required',
            'price' => 'required|numeric',
        ]);

        $product = Product::findOrFail($id);
        $product->update($request->all());

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

    public function destroy($id)
    {
        $product = Product::findOrFail($id);
        $product->delete();

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

```

Task 6: Single Action Controller

```
php artisan make:controller ContactController --invokable
```

```
<?php
```

```

namespace App\Http\Controllers;

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

class ContactController extends Controller
{
    public function __invoke(Request $request)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required|email',
            'message' => 'required',
        ]);

        $data = [
            'name' => $request->input('name'),
            'email' => $request->input('email'),
            'message' => $request->input('message'),
        ];

        $toEmail = 'your-email@example.com';

        Mail::to($toEmail)->send(new \App\Mail\ContactFormMail($data));

        return redirect()->back()->with('success', 'Thank you for your message. We will be in touch soon.');
    }
}

```

Task 7: Resource Controller

```
php artisan make:controller PostController --resource
```

```
<?php

namespace App\Http\Controllers;

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

class PostController extends Controller
{
    public function index()
    {
        $posts = Post::all();
        return view('posts.index', compact('posts'));
    }

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

    public function store(Request $request)
    {
        $request->validate([
            'title' => 'required',
            'content' => 'required',
        ]);

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

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

    public function show($id)
    {
```

```
$post = Post::findOrFail($id);
return view('posts.show', compact('post'));
}

public function edit($id)
{
    $post = Post::findOrFail($id);
    return view('posts.edit', compact('post'));
}

public function update(Request $request, $id)
{
    $request->validate([
        'title' => 'required',
        'content' => 'required',
    ]);

    $post = Post::findOrFail($id);
    $post->update($request->all());

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

public function destroy($id)
{
    $post = Post::findOrFail($id);
    $post->delete();

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

Task 8: Blade Template Engine

1. Open your Laravel project and navigate to the **resources/views** directory.
2. Create a new file called **welcome.blade.php** and open it.
3. Add the following code to the **welcome.blade.php** file:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Welcome to Laravel</title>
  </head>
  <body>
    <nav>

    </nav>
    <section>
      <h1>Welcome to Laravel!</h1>
    </section>
  </body>
</html>
```