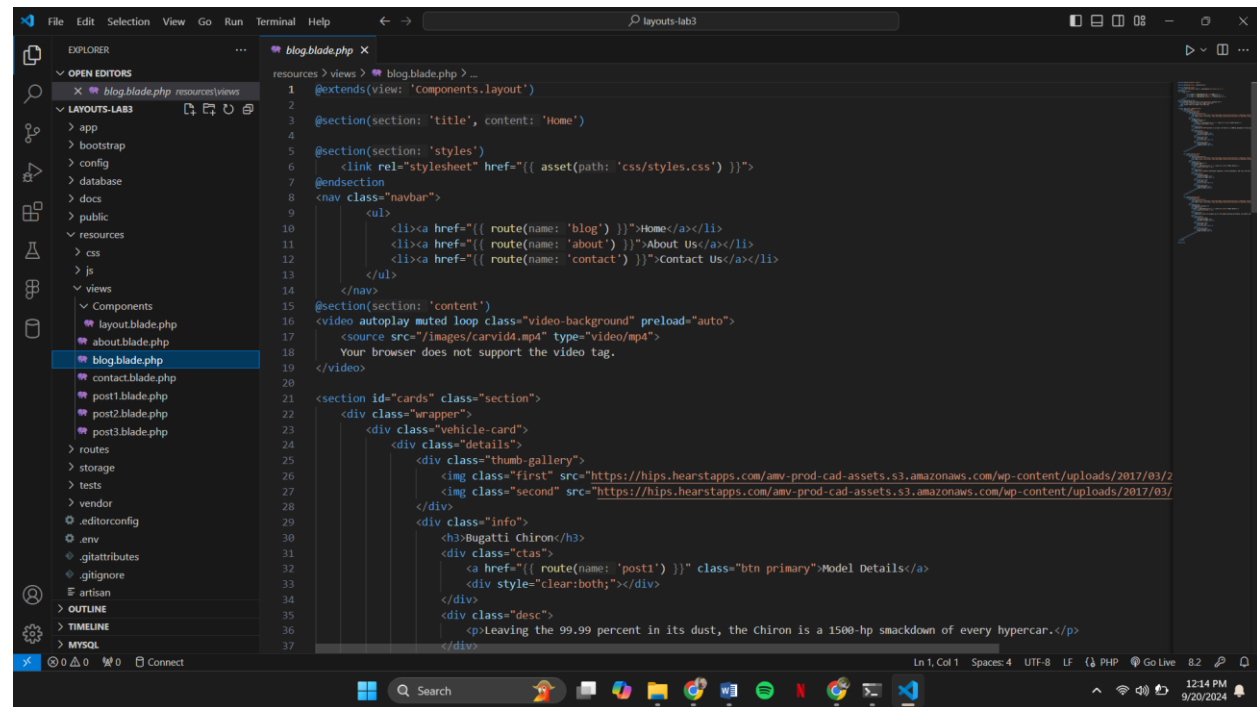
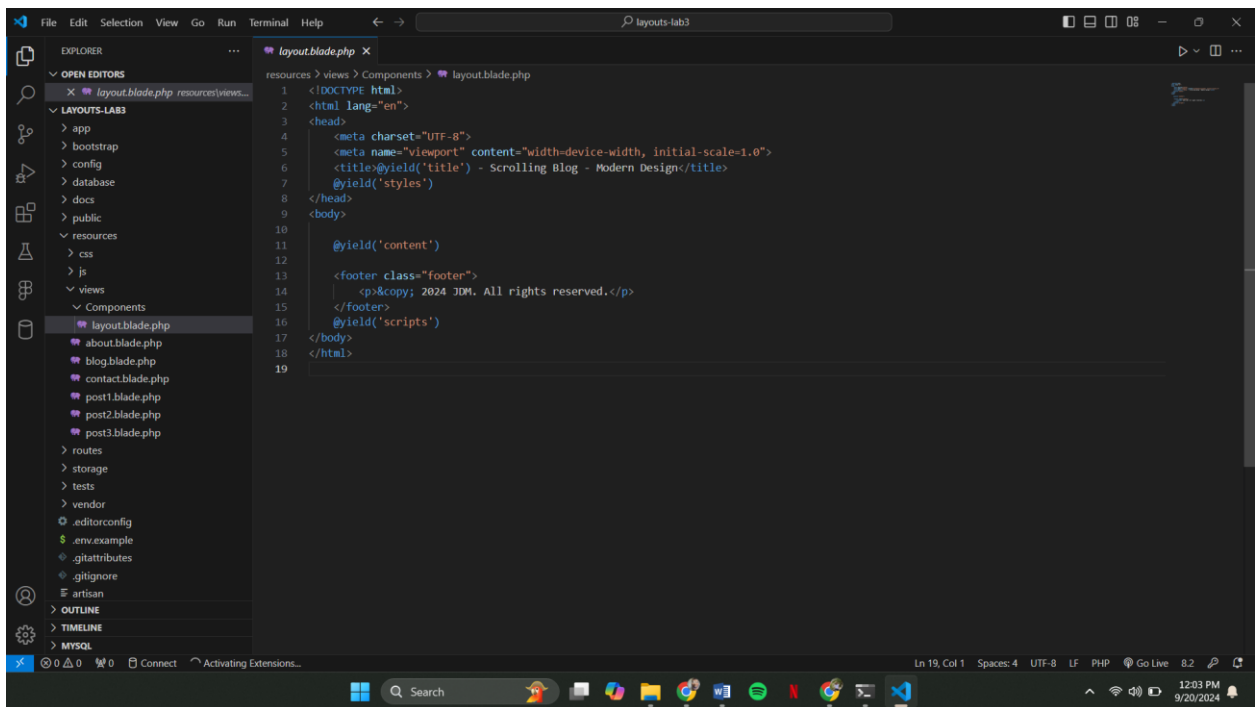


Juliane Rebecca S. Dayandante

BSIT – 3C

Layout file

- We use this for common structure, we just extend this to our other files so that we can avoid code duplication or repeating code.



@extends('Components.layout')

- In this line tells the Blade that the current view (Page 1) extends from a layout file located in the Components folder.

@section('styles'):

- This defines the styles section, which used for inserting CSS or external stylesheets specific to this page.
- Inside this section, a CSS file (post1.css) is linked using {{ asset('css/post1.css') }}. The asset helper generates a URL for the post1.css file located in the public/css directory.

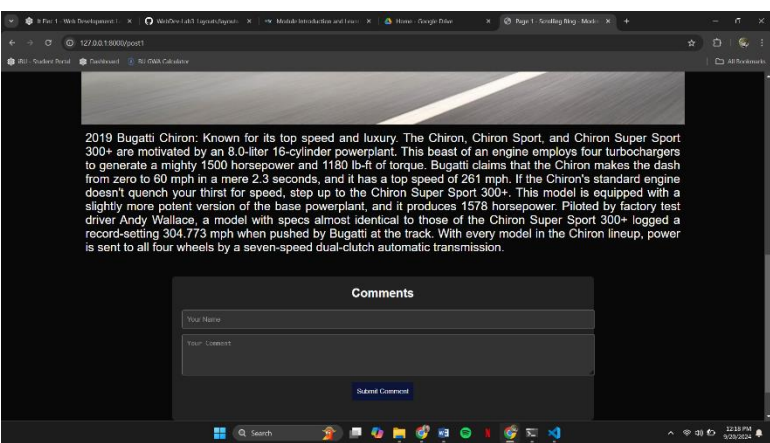
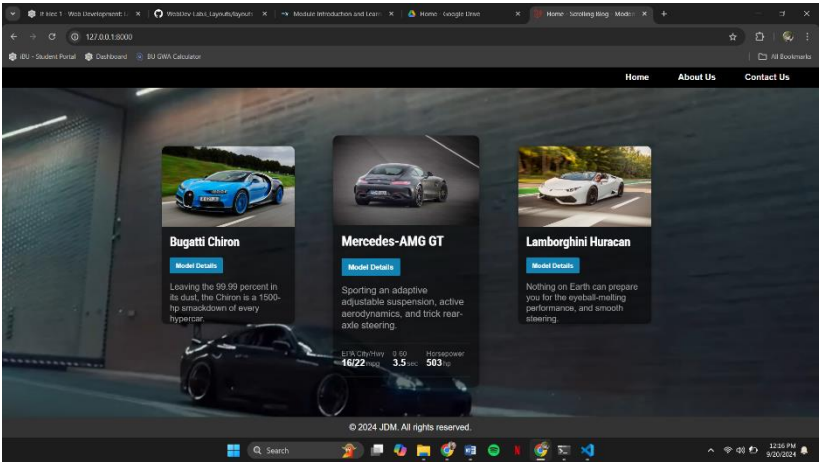
@section('content'):

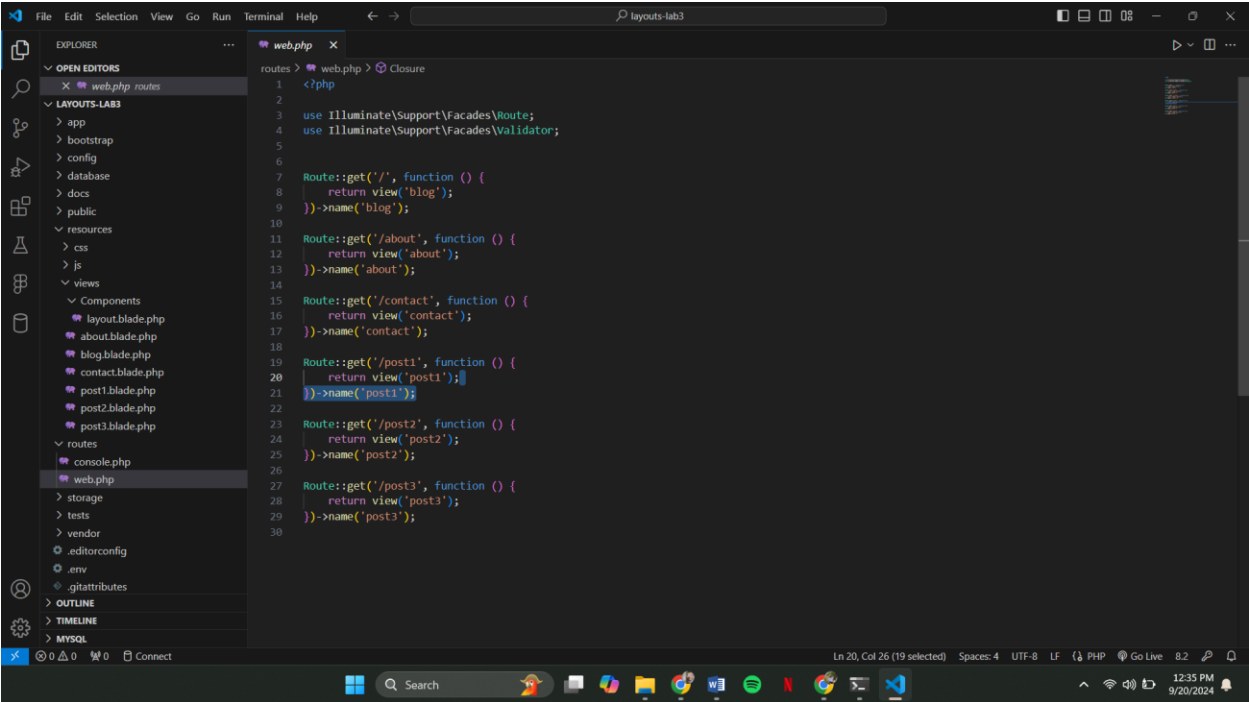
- The content section holds the main structure for this view.
- The layout file has a @yield('content') directive, which will be replaced with the content from this section when the view is rendered.

@section('scripts'):

- This section is for any JavaScript specific to this page.

Web Pages





In this Laravel routing setup, each URL is mapped to a specific view, allowing users to navigate different parts of the website. For example, when someone visits the homepage (/), the system serves the blog.blade.php view, giving them access to the blog content. Similarly, when they access /about, the about.blade.php page is loaded. The contact page at /contact works the same way, displaying the contact.blade.php view. For individual blog posts, routes like /post1, /post2, and /post3 are set up to serve the corresponding post views, making navigation seamless while ensuring each page has its own unique content.

Challenges faced:

As someone focused on the content of the site, one challenge I faced was ensuring that the CSS properly connected to the code, especially in a Laravel setup where it sometimes didn't work as expected. At first, it was frustrating because even though I knew the styles were correct, they weren't showing up. I realized that the issue often came down to how Laravel handles assets, like CSS files, and whether the asset() function was correctly generating the path. To solve this, I ensured that my CSS files were stored in the proper file/folder.

The difference between {{\$slot}} and @yield:

In Laravel Blade, {{ \$slot }} is primarily used with components, allowing dynamic content to be passed into the component from the parent view. On the other hand, @yield is used within layout files to define sections where child views can inject content. Essentially, @yield works with Blade layouts for structuring pages, while {{ \$slot }} is for reusable components, making it more specific to the context of Blade components.