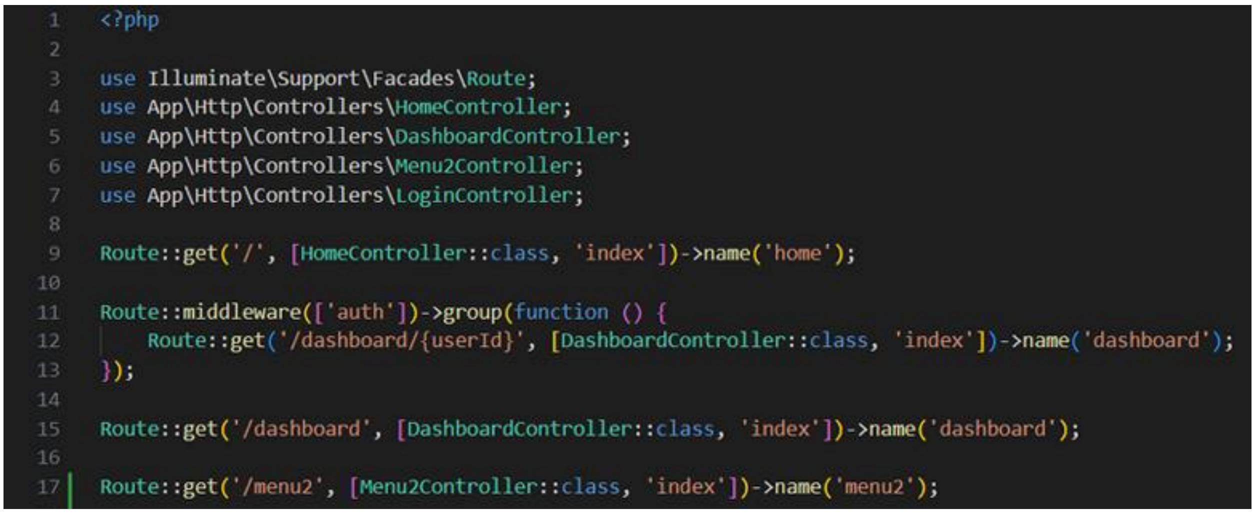
PRECIOUS GWYNTH A. MALLO	**WEBDEV**
BSIT 3C	**LAB 6**

POPULATING PAGES

PART 1: CREATE VIEWS USING BLADES TEMPLATES

- To create the DashboardController and Menu2Controller in Laravel, run specific terminal commands. These controllers handle the logic for the dashboard and menu of our app.
- The code links URLs to specific controller methods. For instance, visiting the home page (/) calls the HomeController's index method to show the homepage. Similarly, visiting /dashboard calls the DashboardController's index method to show the dashboard, and /menu2 calls the Menu2Controller's index method to show the menu.



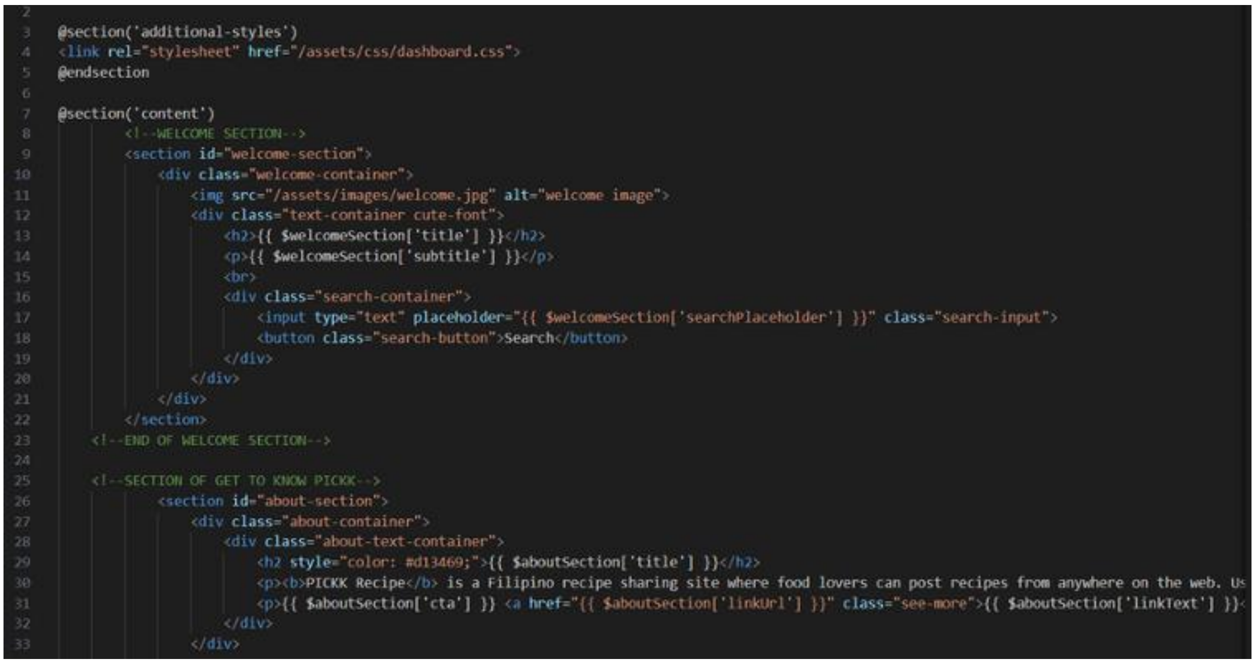
```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\Menu2Controller;
use App\Http\Controllers\LoginController;

Route::get('/', [HomeController::class, 'index'])->name('home');

Route::middleware(['auth'])->group(function () {
    Route::get('/dashboard/{userId}', [DashboardController::class, 'index'])->name('dashboard');
});

Route::get('/dashboard', [DashboardController::class, 'index'])->name('dashboard');

Route::get('/menu2', [Menu2Controller::class, 'index'])->name('menu2');
```

PART 2: CREATE CONTROLLERS THAT LOAD CONTENT FOR EACH PAGE (SIMULATE ACCESS TO A DB)

1. **Load Information:**
    - Load data like posts, ratings, comments, pictures, etc., using arrays.

- Create controllers that simulate loading webpage content as if accessing a database. Instead of a real database, use arrays to represent data (like food categories, recipes, and images). Each category and recipe is stored in an array with key-value pairs (e.g., images, names, descriptions).

- Controllers gather this data and pass it to the views. For example, the Menu2Controller prepares an array with necessary information (like food categories and recipes) and sends it to the view using Laravel's compact() function. The view then displays the content dynamically using Blade syntax.



```

@section('additional-styles')
<link rel="stylesheet" href="/assets/css/dashboard.css">
@endsection

@section('content')
        <!--WELCOME SECTION-->
        <section id="welcome-section">
            <div class="welcome-container">
                <img src="/assets/images/welcome.jpg" alt="welcome image">
                <div class="text-container cute-font">
                    <h2>{{ $welcomeSection['title'] }}</h2>
                    <p>{{ $welcomeSection['subtitle'] }}</p>
                    <br>
                    <div class="search-container">
                        <input type="text" placeholder="{{ $welcomeSection['searchPlaceholder'] }}" class="search-input">
                        <button class="search-button">Search</button>
                    </div>
                </div>
            </div>
        </section>
    <!--END OF WELCOME SECTION-->

    <!--SECTION OF GET TO KNOW PICKK-->
            <section id="about-section">
                <div class="about-container">
                    <div class="about-text-container">
                        <h2 style="color: #d13469;">{{ $aboutSection['title'] }}</h2>
                        <p><b>PICKK Recipe</b> is a Filipino recipe sharing site where food lovers can post recipes from anywhere on the web. Us
                        <p>{{ $aboutSection['cta'] }} <a href="{{ $aboutSection['linkUrl'] }}" class="see-more">{{ $aboutSection['linkText'] }}<
                    </div>
                </div>
```

PART 3:

1. **Controller Logic:**

   - The DashboardController retrieves and processes data for the dashboard page, like user stats and recent activities, and sends it to the view for display.
   - The MenuController handles the logic for the menu page, managing data related to the menu and ensuring it is presented correctly to the user.

2. **Parameter Handling:**

   - Controllers use parameters to customize data. For instance, a user might request a specific dashboard or menu item, and the controller will use parameters to fetch and display the correct data.

3. **Route Assignments:**

   - Route assignments map specific URLs to actions in the controllers. When a user visits a URL, the route ensures the correct controller action is called to display the needed data.