BICOL UNIVERSITY COLLEGE OF SCIENCE
LEGAZPI, ALBAY
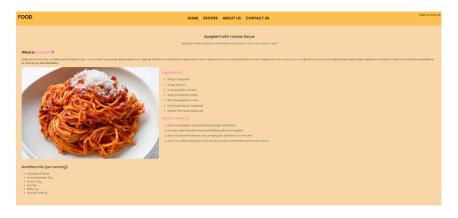
IT Elect 1 – WEB DEVELOPMENT
DOCUMENTATION
(LABORATORY 3)

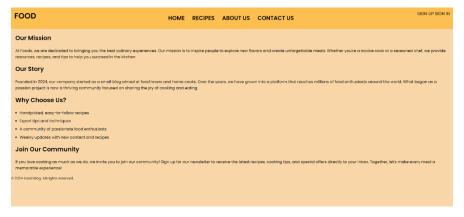**SUBMITTED BY:**
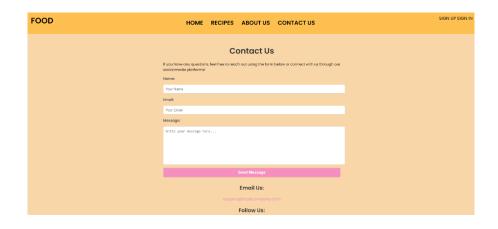Valerie Joyce R. Soreda

HOMEPAGE

RECIPES PAGE

ABOUT US PAGE

CONTACT US PAGE

LAYOUT FILE CODE

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>@yield('title')</title>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.6.0/css/all.min.css" integrity=
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
</head>

<body>
    <!-- Header Section -->
    <header class="header">
        <div class="logo">
            <a href="/">FOOD</a>
        </div>
        <nav>
            <ul>
                <li><a href="/">HOME</a></li>
                <li><a href="/recipes">RECIPES</a></li>
                <li><a href="/about">ABOUT US</a></li>
                <li><a href="/contact">CONTACT US</a></li>
            </ul>
        </nav>
        <div class="auth-links">
            <a href="/signup" class="signup-btn">SIGN UP</a>
            <a href="/signin" class="signin-btn">SIGN IN</a>
        </div>
    </header>
</body>

    <!-- Main Content -->
    <main>
        @yield('content')
    </main>

    <!-- Footer Section -->
    <footer>
        <p>&copy; 2024 Food Blog. All rights reserved.</p>
    </footer>

    <!-- Add JS for layout -->
    <script src="{{ asset('js/app.js') }}"></script>
```

Purpose of the layout file and how it is used: This code serves as a layout template for a food blog web application, featuring a structured header, main content area, and footer. By using Blade templating in Laravel, other views can extend this layout and fill in specific content while maintaining a consistent structure.

```php
routes > web.php
1   <?php
2
3   use Illuminate\Support\Facades\Route;
4
5   Route::get('/', function () {
6       return view('Components.home');
7   })->name('home');
8
9   Route::get('/about', function () {
10      return view('Components.about');
11  })->name('about');
12
13  Route::get('/contact', function () {
14      return view('Components.contact');
15  })->name('contact');
16
17  Route::get('/recipes', function () {
18      return view('Components.recipes');
19  })->name('recipes');
20
```

The routing setup and how it serves the views: The routing setup in `web.php` defines routes using `Route::get()`, mapping specific URLs (like `/about` and `/recipes`) to anonymous functions that return views. Each route is assigned a name, such as `'home'` or `'recipes'`, for easy reference elsewhere in the application. These routes serve views located in the `Components` directory, such as `Components.home` and `Components.about`.

**Explain the difference between {{$slot}} and @yield:** {{$slot}} is used in Blade components to allow dynamic content insertion, while @yield defines a section placeholder in layouts that individual views can fill with their specific content.

One challenge was ensuring the correct views were rendered when navigating to different routes. We resolved this by organizing the view files in the `Components` directory and updating the routes to point to the correct views. Also, the named routes helped simplify redirection and link generation throughout the application.