








Layout file located in the components folder:

Name	Date modified	Type	Size
Components	9/15/2024 12:14 AM	File folder	
 contact.blade.php	9/15/2024 12:14 AM	PHP Source File	2 KB
 gallery.blade.php	9/15/2024 12:14 AM	PHP Source File	6 KB
 login.blade.php	9/15/2024 12:20 AM	PHP Source File	3 KB
 register.blade.php	9/15/2024 12:20 AM	PHP Source File	4 KB
 review.blade.php	9/15/2024 12:14 AM	PHP Source File	5 KB
 user.blade.php	9/15/2024 12:14 AM	PHP Source File	1 KB
 welcome.blade.php	9/15/2024 12:14 AM	PHP Source File	1 KB

Our layout file:

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta name="csrf-token" content="{{ csrf_token() }}">
8     <title>{{ $title ?? 'Jet's Gaming Hub' }}</title>
9     <link href="{{ asset('css/app.css') }}" rel="stylesheet">
10     @stack('styles')
11 </head>
12
13 <body class="@yield('body-class')">
14 <header>
15     <div class="header-container">
16         <h1>Jet's Gaming Hub</h1>
17         <nav>
18             @if(session('username') && session('username') != 'Guest')
19                 <span>Welcome, {{ session('username') }}</span>
20                 <a href="{{ url('/logout') }}">Logout</a>
21             @elseif(!request()->is('/'))
22                 <span>Welcome, Guest</span>
23             @endif
24             <a href="{{ url('/') }}">Homepage</a>
25             <a href="{{ url('/gallery') }}">Gallery</a>
26             <a href="{{ url('/review') }}">Reviews</a>
27             <a href="{{ url('/contact') }}">Contact</a>
28         </nav>
29     </div>
30 </header>
31
32 <main class="@yield('main-class')">
33     @yield('content')
34 </main>
35
36 <footer>
37     <div class="footer-container">
38         <p>&copy; 2024 Jet's Gaming Hub. All rights reserved.</p>
39     </div>
40 </footer>
41
42 @stack('scripts')
43 </body>
44 </html>
45
```

Implementation of layout file in our content.blade.php

```
1 @extends('Components.Layout')
2
3 @php
4     $title = 'Contact Us';
5 @endphp
6
7 @push('styles')
8     <link href="{{ asset('css/contact.css') }}" rel="stylesheet">
9 @endpush
10
11 @section('body-class', 'contact')
12
13 @section('content')
14     <div class="min-h-screen flex items-center justify-center bg-gradient-to-b from-transparent via-white to-white dark:via-zinc-900 dark:to-zinc-900">
15         <div class="form-container">
16             <h1>Contact Us</h1>
17             <form action="">
18                 <div class="form-group">
19                     <label for="name" class="form-label">Name:</label>
20                     <input type="text" name="name" id="name" class="form-input" required>
21                 </div>
22                 <div class="form-group">
23                     <label for="email" class="form-label">Email Address:</label>
24                     <input type="email" name="email" id="email" class="form-input" required>
25                     <p id="emailHelp" class="form-text">We'll never share your email with anyone else.</p>
26                 </div>
27                 <div class="form-group">
28                     <label for="message" class="form-label">Message:</label>
29                     <textarea name="message" id="message" class="textarea" required></textarea>
30                 </div>
31                 <div class="form-actions">
32                     <button type="submit" class="form-button">Submit</button>
33                 </div>
34             </form>
35         </div>
36     </div>
37 @endsection
```

The purpose of the layout file is to easily implement a template, for example the navbar design. We can now avoid the use of copy pasting navbar from each blade file where instead we can now make it a template and with a simple line of code it will automatically implement the layout file or design which in our case, is the navbar. It also makes the code cleaner by preventing unnecessary lines like I said; copy pasting the navbar before.

How each view file extends the layout and inserts specific content. This makes designing a whole lot easier, and making the code a whole lot cleaner. Since with this, you can just divide the page into sections together with their own specific design.

Routing:

```
C:\Users\User > Desktop > ec > Group2 > Group2 > routes > web.php
```

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use Illuminate\Support\Facades\Session;
5
6 // Route to display the login page
7 Route::get('/', function () {
8     if (Session::has('username') && Session::get('username') != 'Guest') {
9         return redirect('/user');
10    }
11    return view('login');
12 });
13
14 // Route to display the registration page
15 Route::get('/register', function () {
16     return view('register');
17 });
18
19 // Route to display the gallery page
20 Route::get('/gallery', function () {
21     return view('gallery');
22 });
23
24 // Route to display the review page
25 Route::get('/review', function () {
26     return view('review');
27 });
28
29 // Route to display the contact page
30 Route::get('/contact', function () {
31     return view('contact');
32 });
33
34 // Route to handle user login or continue as guest
35 Route::get('/user', function () {
36     // Get the username from the request or default to 'Guest'
37     $username = request()->input('username', 'Guest');
38
39     // Validate username to ensure it only contains alphabetic characters
40     if (!preg_match('/^[A-Za-z]+$/', $username)) {
41         $username = 'Guest';
42     }
43
44     // Store the username in session
45     Session::put('username', $username);
46
47     // Return the user view with the username
48     return view('user', ['username' => $username]);
49 });
50
51 // Route to handle user login or registration submission
52 Route::post('/user', function () {
53     // Get the username from the request or default to 'Guest'
54     $username = request()->input('username', 'Guest');
55
56     // Validate username to ensure it only contains alphabetic characters
57     if (!preg_match('/^[A-Za-z]+$/', $username)) {
58         $username = 'Guest';
59     }
60
61     // Store the username in session
62     Session::put('username', $username);
63
64     // Return the user view with the username
65     return view('user', ['username' => $username]);
66 });
67
68 // Validate username to ensure it only contains alphabetic characters
69 if (!preg_match('/^[A-Za-z]+$/', $username)) {
70     $username = 'Guest';
71 }
72
73 // Store the username in session
74 Session::put('username', $username);
75
76 // Return the user view with the username
77 return view('user', ['username' => $username]);
78 });
79
80 // Route to handle user logout
81 Route::get('/logout', function () {
82     // Clear the username from session
83     Session::forget('username');
84
85     // Redirect to the homepage or login page
86     return redirect('/');
87 });
```

In the routing, we have the log in and log out function using sessions. We also added the register view wherein the user can register for the website but it's still a Work In Progress as we are still exploring SQLite and how it works, hopefully we can implement it after trying to understand how to connect it to make the registration and login functional. Also, in the future we will also add a review or feedback on the game for our website, since it's a gaming site wherein you can see reviews from other users and their rating.

For me, the challenges I faced are the routing, like last activity I had a little bit confusion when passing the name parameter and having a trouble displaying the username. Like that one time It will display the user but when I click gallery or reviews, the user will become "Welcome, Gallery" even though on the url parameter it displayed the correct username.

Based on what I gave gathered, the difference between `{{ $slot }}` and `@yield`, is that `{{ $slot }}` displays the variables directly, wherein a `@yield` can be used in a section to act as a placeholder for contents