

## NARRATIVE REPORT: LARAVEL PORTFOLIO WEBSITE

### Objective:

The primary goal of this project was to create a Laravel website that showcases my understanding of key Laravel concepts, including routes, views, layouts, and middleware. This portfolio site serves as a repository of all my lab activities and demonstrates the implementation of core web development practices in Laravel.

### Project Overview:

#### 1. Routes:

- I set up routes for each lab activity, including a parameterized route `/Projects/{Lab}` to load lab-specific content dynamically.

#### 2. Views:

- Each lab has its own view displaying a title, description, and summary. A master layout is used to keep the design consistent across all pages, ensuring a unified look.

#### 3. Layout:

- The website uses a master layout that contains shared components like the header, footer, and navigation. This makes sure every page has the same style and structure. However, I chose not to include the header and footer in some views where they were unnecessary.

#### 4. Middleware:

- Two middleware were created:
  - `CheckAge` verifies the user's age and sets the verification status based on whether the user is at least 21 or below.
  - `LogRequests` logs user details, including the request time, HTTP method, URL, name, IP address, age, and verification status to a `log.txt` file.

#### 5. Content:

- The portfolio includes all my completed labs, with brief explanations and summary for each, giving an overview of what was done.

# PORTFOLIO DOCUMENTATION

## ROUTES

(Explanation in the comments)

web.php M X

routes > web.php

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use Illuminate\Support\Facades\Session;
5  use App\Http\Middleware\CheckAge; // Import CheckAge middleware
6  use App\Http\Middleware\LogRequests; // Import LogRequests middleware
7
8  // Route for Landing Page
9  Route::get(uri: '/', action: function (): Factory|View {
10 |     return view(view: 'LandingPage');
11 | });
12
13 // Route to handle the form submission
14 Route::post(uri: '/Welcome', action: function (): Redirector|RedirectResponse {
15 |     // Get the name from the request or default to 'Guest'
16 |     $name = request()->input(key: 'name', default: 'Guest');
17
18 |     // Validate name to ensure it only contains alphabetic characters
19 |     if (!preg_match(pattern: '/^[A-Za-z]+$/', subject: $name)) {
20 |         $name = 'Guest';
21 |     }
22
23 |     // Store the name in session
24 |     session(key: ['name' => $name]);
25
26 |     // Redirect to Welcome page
27 |     return redirect(to: '/Welcome');
28 | });
29
30 // GET Route for Welcome page (after form submission)
31 Route::get(uri: '/Welcome', action: function (): Factory|View {
32 |     // Retrieve the name from the session
33 |     $name = session(key: 'name', default: 'Guest');
34
35 |     return view(view: 'Welcome', data: ['name' => $name]);
36 | });
```

```

38 // Other routes
39 ✓ Route::get(uri: '/Home', action: function (): Factory|View {
40 |     return view(view: 'Home');
41 | });
42
43 ✓ Route::get(uri: '/About', action: function (): Factory|View {
44 |     return view(view: 'About');
45 | });
46
47 ✓ Route::get(uri: '/Projects', action: function (): Factory|View {
48 |     return view(view: 'Projects');
49 | });
50 💡
51 | // Parameterized route to render lab views based on the lab name
52 ✓ Route::get(uri: '/Projects/{Lab}', action: function ($Lab): Factory|View {
53 |     return view(view: $Lab);
54 | });
55
56 ✓ Route::get(uri: '/Contact', action: function (): Factory|View {
57 |     return view(view: 'Contact');
58 | });
59
60 ✓ Route::get(uri: '/Verify', action: function (): Factory|View {
61 |     return view(view: 'Verify');
62 | });
63
64 // Route to handle "Continue as Guest" link
65 ✓ Route::get(uri: '/Guest', action: function (): Redirector|RedirectResponse {
66 |     // Store 'Guest' in session
67 |     session(key: ['name' => 'Guest']);
68 |
69 |     // Redirect to Welcome page
70 |     return redirect(to: '/Welcome');
71 | });

```

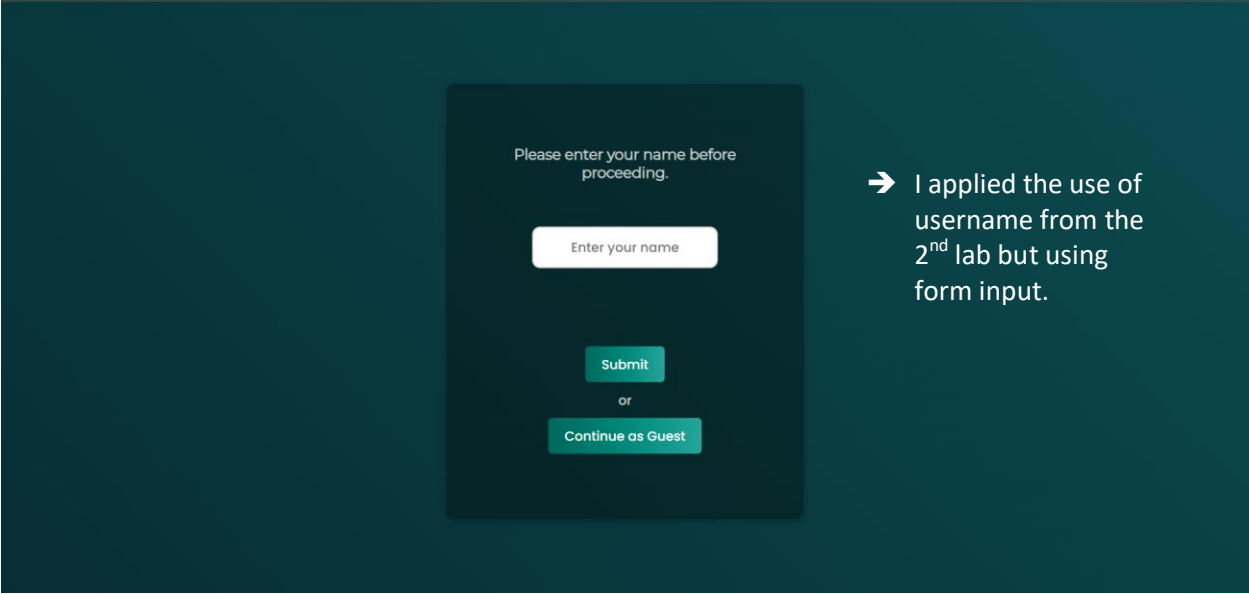
```

74 // Route to handle when access is denied
75 Route::get(uri: '/Access-denied', action: function (): Factory|View {
76 |     // Clear the age and verification_stat from the session
77 |     Session::forget(keys: 'age');
78 |     Session::forget(keys: 'verification_stat');
79 |
80 |     return view(view: 'Access-denied');
81 | });
82
83 // Route to handle the form submission of age
84 Route::post(uri: '/verify-age', action: function (): Redirector|RedirectResponse {
85 |     // Get the 'age' from the request input
86 |     $age = request()->input(key: 'age');
87 |
88 |     // Store the age in session
89 |     Session::put(key: 'age', value: $age);
90 |
91 |     return redirect(to: '/Projects');
92 | });
93
94 // Apply CheckAge with a minimum age of 21 for Projects
95 Route::middleware(middleware: [CheckAge::class . ':21', LogRequests::class])->group(callback: function (): void {
96 |     Route::get(uri: '/Projects', action: function (): Factory|View {
97 |
98 |         $name = Session::get(key: 'name', default: 'Guest');
99 |         $age = Session::get(key: 'age');
100 |         $verification_stat = Session::get(key: 'verification_stat');
101 |
102 |         return view(view: 'Projects', data: ['name' => $name, 'age' => $age, 'verification_stat' => $verification_stat]);
103 |     });
104 | });

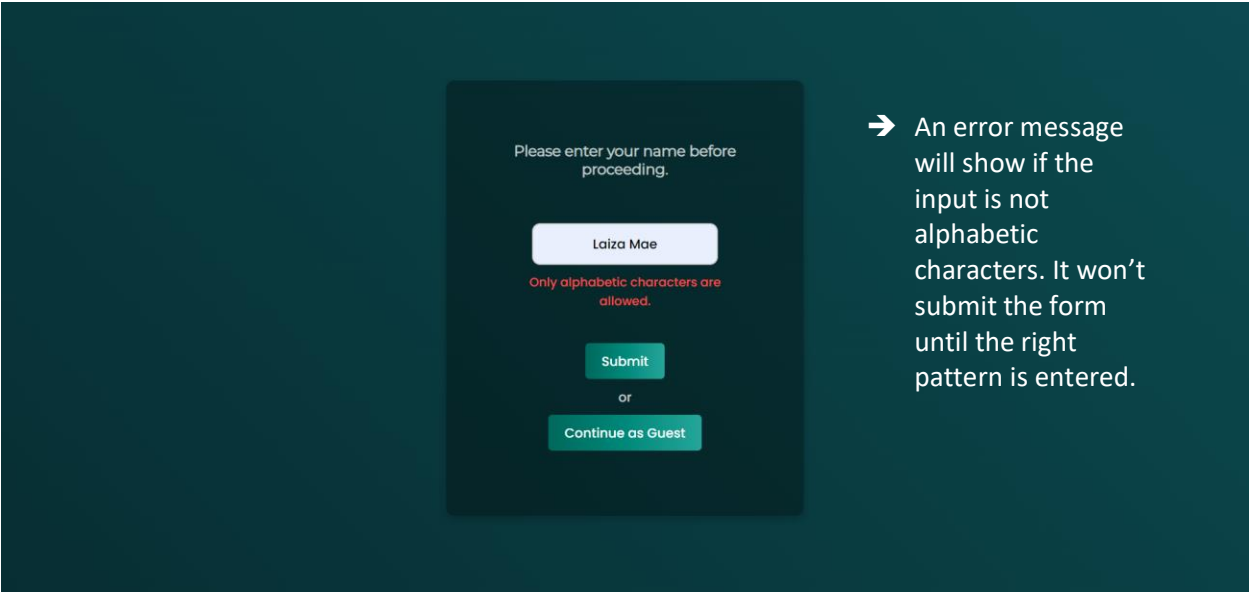
```

VIEWS

Landing Page

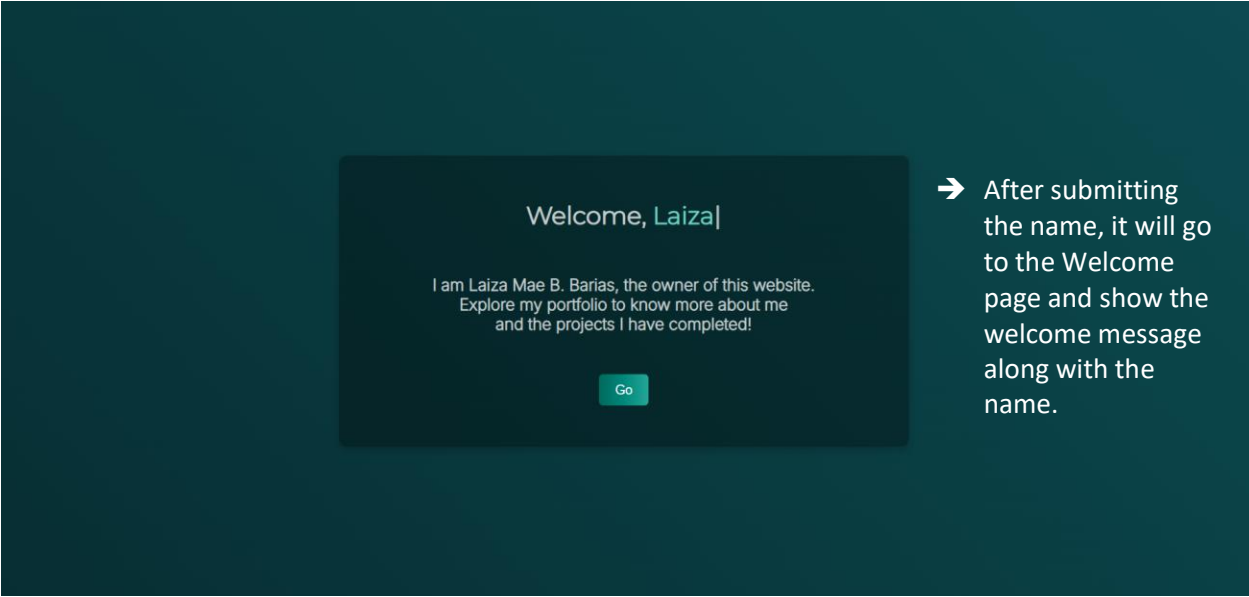


➔ I applied the use of username from the 2<sup>nd</sup> lab but using form input.

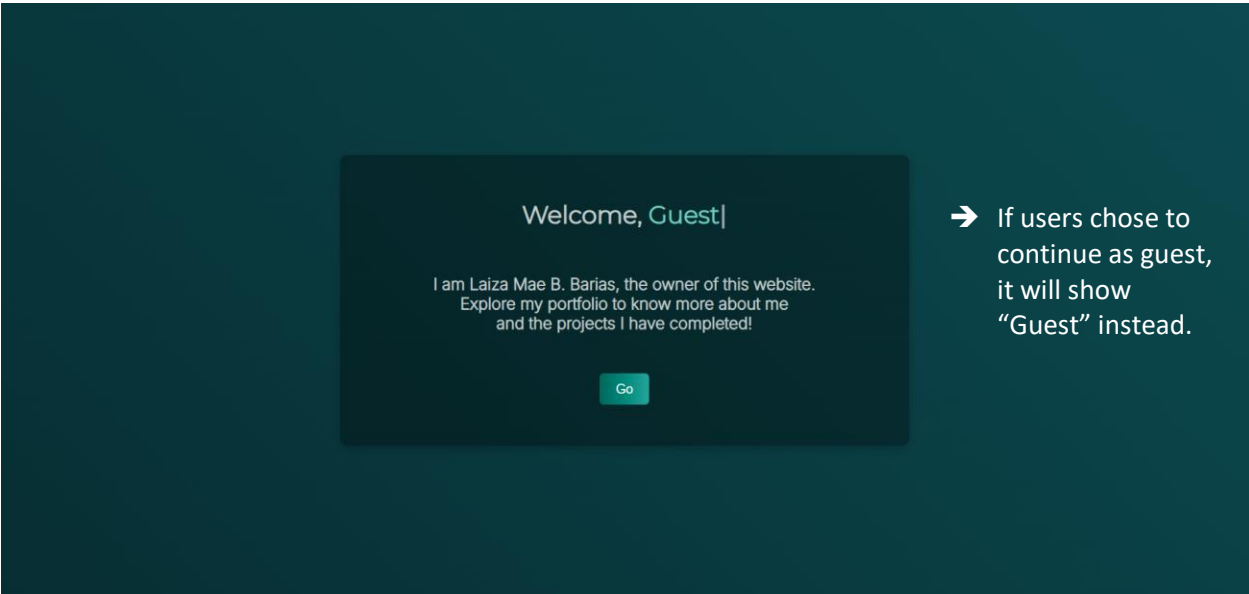


➔ An error message will show if the input is not alphabetic characters. It won't submit the form until the right pattern is entered.

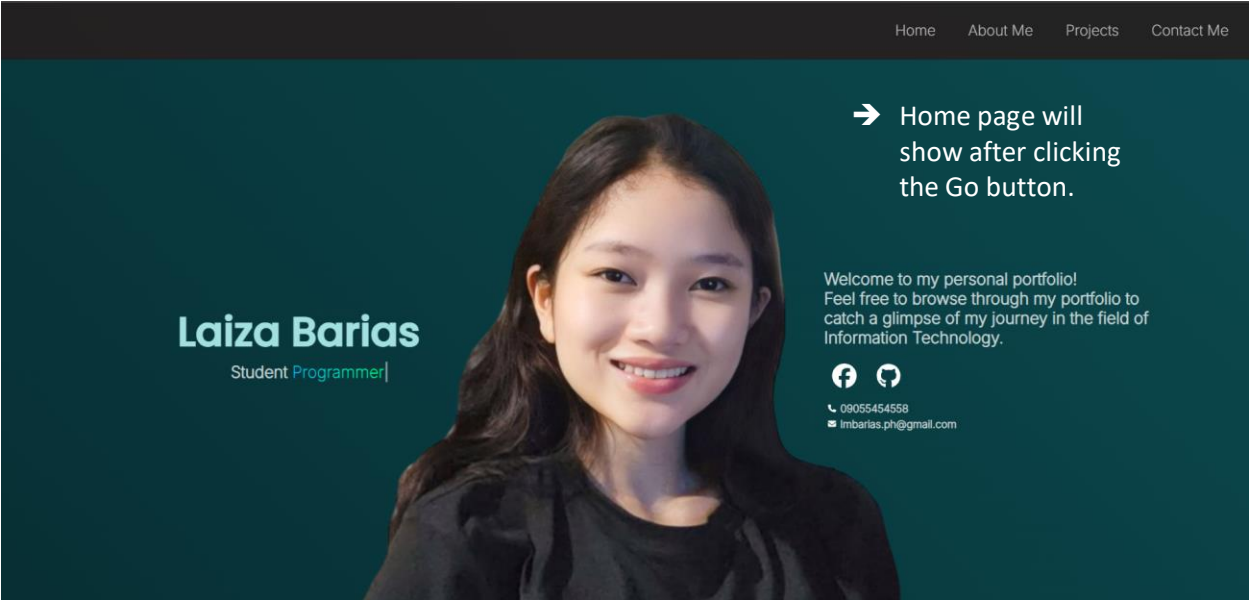
Welcome Page



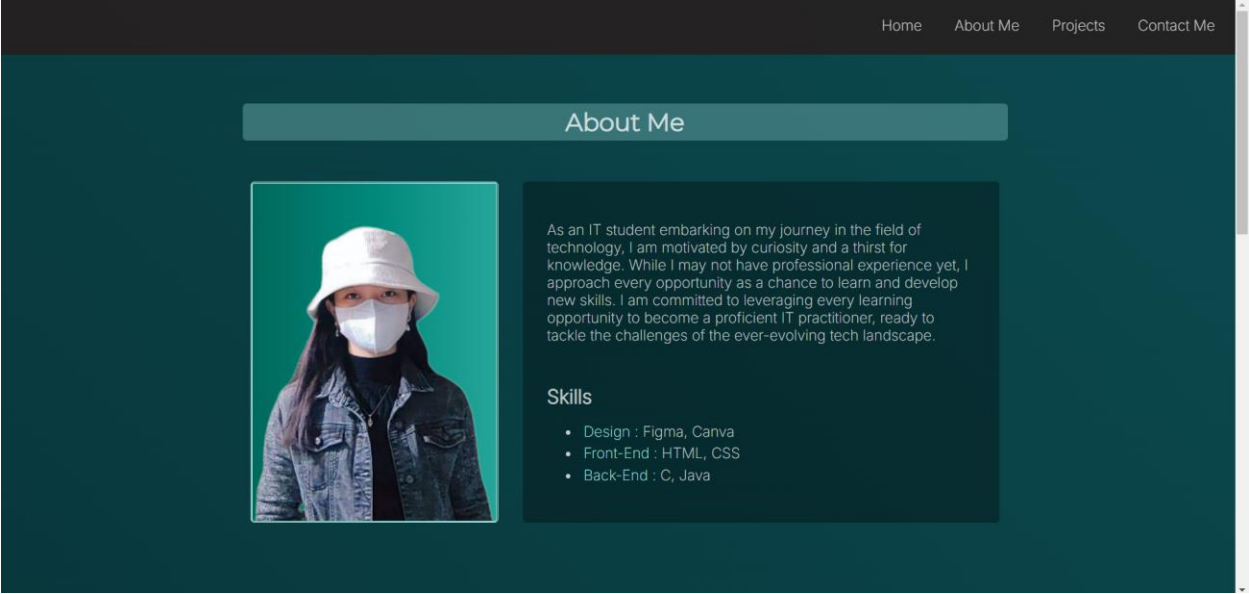
➔ After submitting the name, it will go to the Welcome page and show the welcome message along with the name.



Home Page



About Me Page



Education		
Grade	School	Year Studied
Kindergarten	Tabaco Creative Learning Center	2008-2010
Elementary	Tabaco South Central Elementary School	2010-2016
High School	Tabaco National High School	2016-2022
College	Bicol University	2022-present

### Hobbies

#### Reading Novels and Manhwas/Mangas

I have been a Wattpader ever since I was 11 years old. I have read a lot of stories, from action, to historical, romance, sci-fi, etc. It's amazing how it seems like I am experiencing the story first-hand when I'm reading. Then, when I entered high school, I discovered manhwas, manhwas and mangas. And, of course, I became addicted. I've come to love both written and visual stories. Until now, I am reading whenever I have free time.

[See more](#)

#### Watching Anime




There's something special about anime shows that movies and dramas don't have.

[See more](#)




### Reading Novels and Manhwas/Mangas

I have been a Wattpader ever since I was 11 years old. I have read a lot of stories, from action, to historical, romance, sci-fi, etc. It's amazing how it seems like I am experiencing the story first-hand when I'm reading. Then, when I entered high school, I discovered manhwas, manhwas and mangas. And, of course, I became addicted. I've come to love both written and visual stories. Until now, I am reading whenever I have free time.

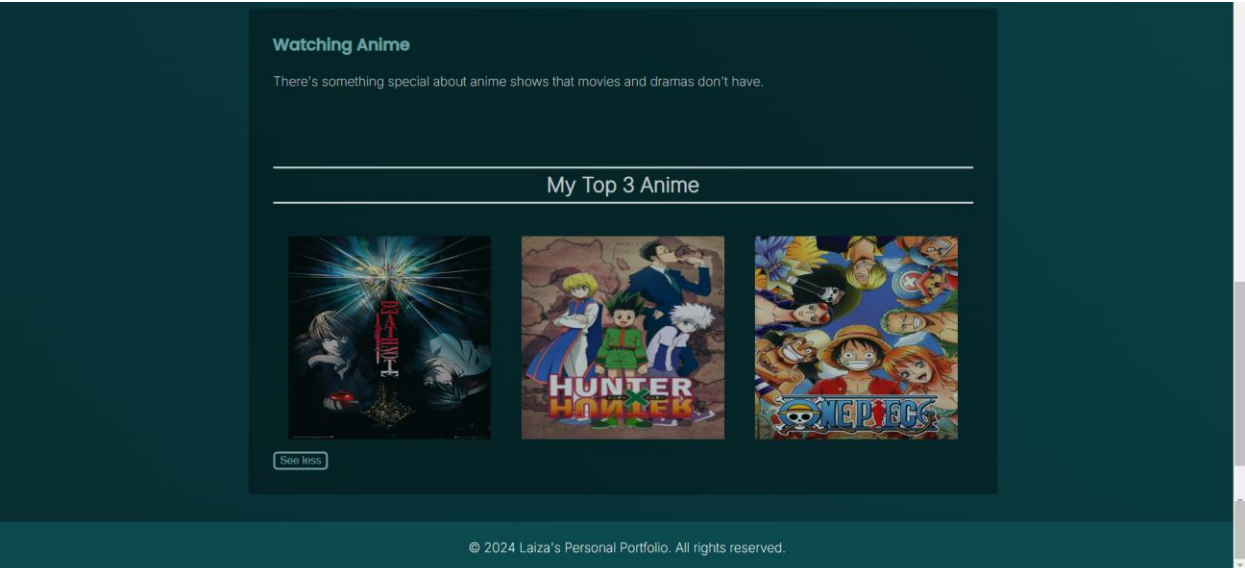
### My Top 3 Wattpad Stories



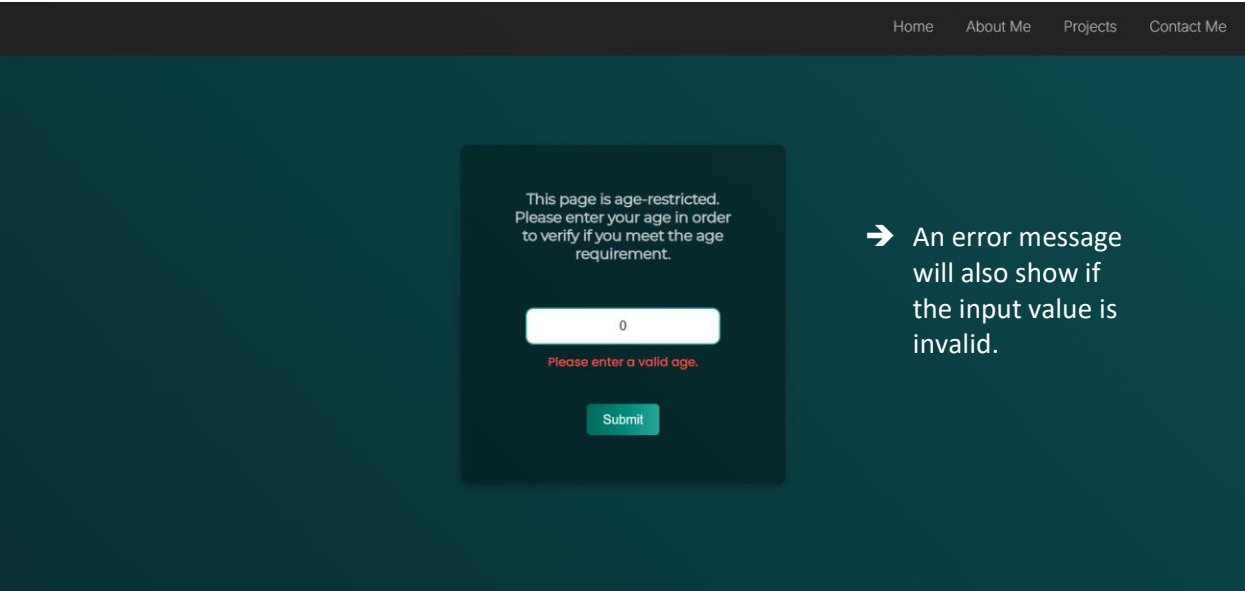
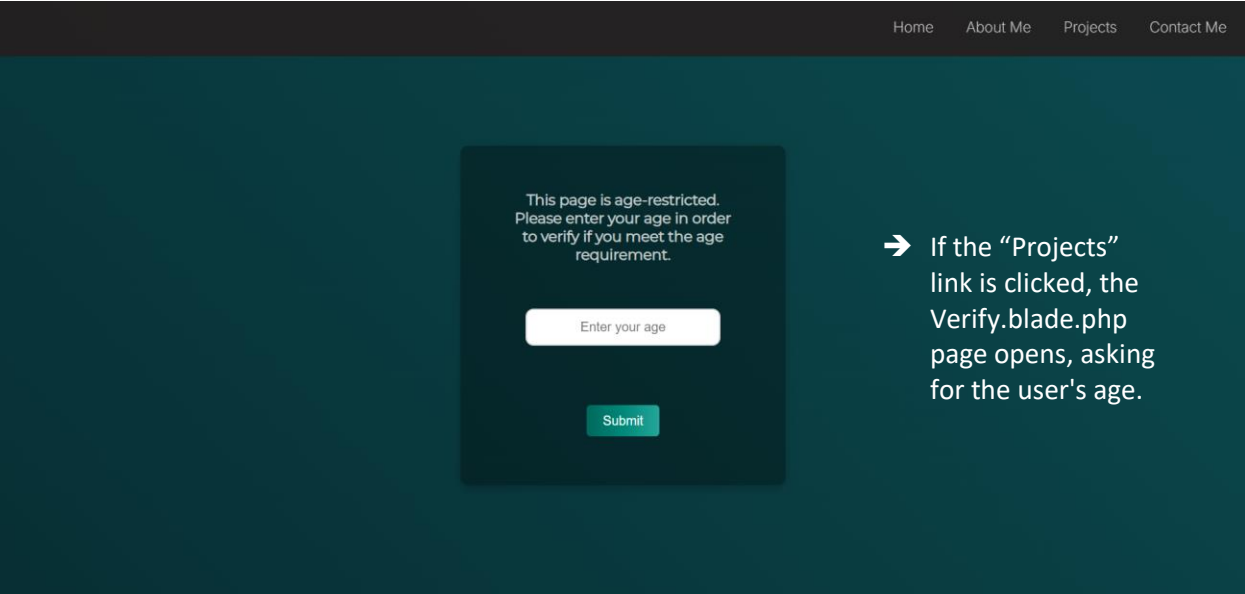
### My Top 3 Manhwas



[See less](#)

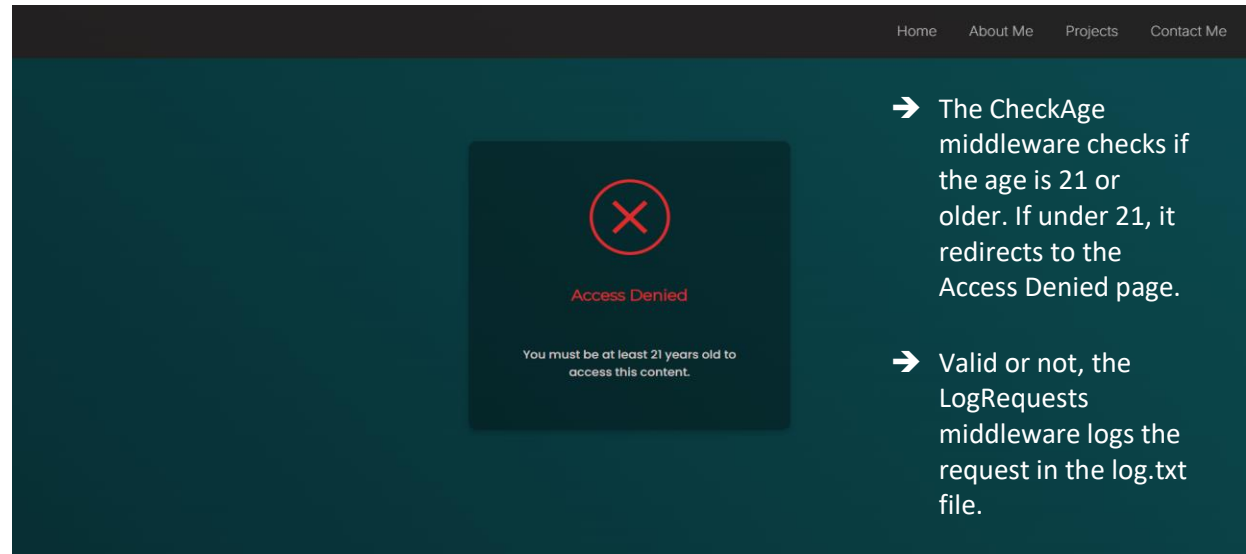


Verify

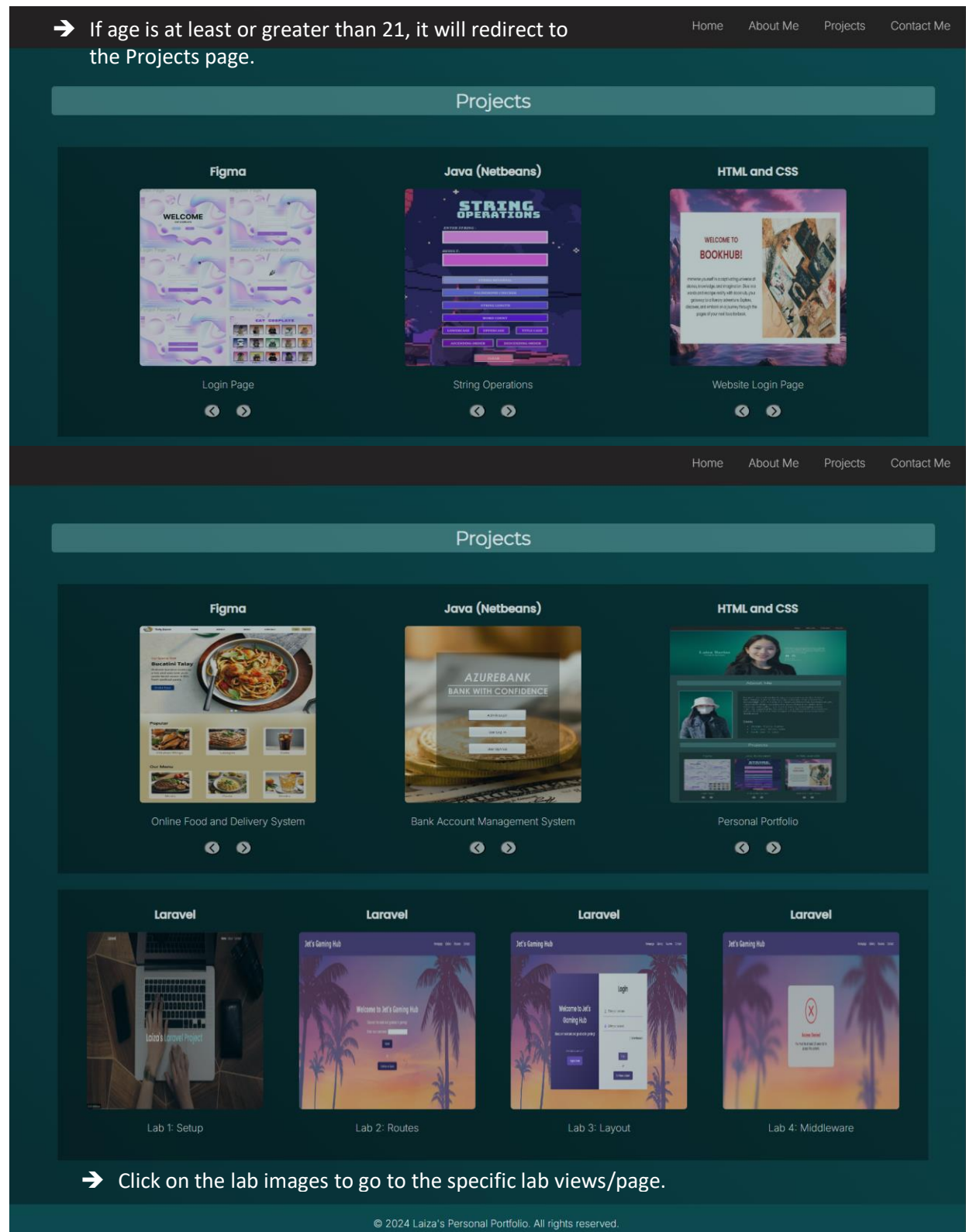




**Access Denied**



## Projects






Lab1

HomeAbout MeProjectsContact Me

Lab 1: Setup



Description

In this lab, the instructions included installing Laravel and its dependencies, creating a new Laravel project, setting up the database connection, and creating basic views and routes. Additionally, the process and the project's structure were to be documented.

Summary

In this lab, I learned how to create my first Laravel project, FirstProject, by installing Laravel via Composer and configuring the environment settings. I set up a Git repository called WebDev to manage my project, where I initialized the repository, tracked changes, and uploaded files to GitHub, emphasizing the importance of version control. Additionally, I developed three views—Home, About, and Content—using Blade templates to render dynamic HTML. Finally, I defined routes in the routes/web.php file, linking user requests to specific views, which taught me how routing manages application flow and user interactions.

→ View PDF Documentation

Next

➔ Click on the View PDF Documentation link to view the documentation of the lab.

© 2024 Laiza's Personal Portfolio. All rights reserved.

Lab1 Documentation

BARIAS\_LAB1\_DOCUMENTATION.pdf1 / 7100%+

1

2

3

2

3

WEB DEVELOPMENT

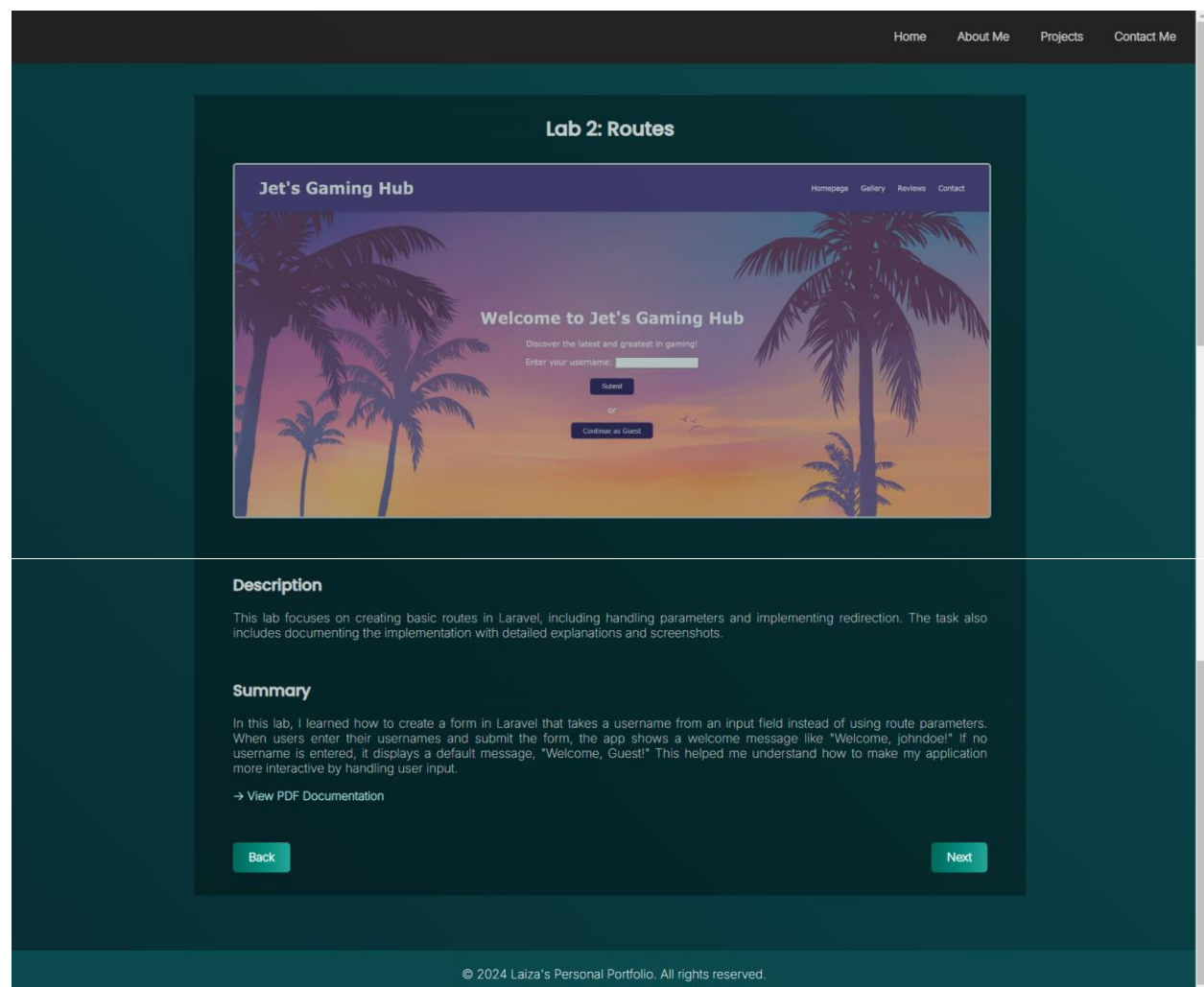
BARIAS, LAIZA MAE B. BSIT-3C08-26-2024

LAB 1: Identify and document the purpose of key directories and files in the laravel project.

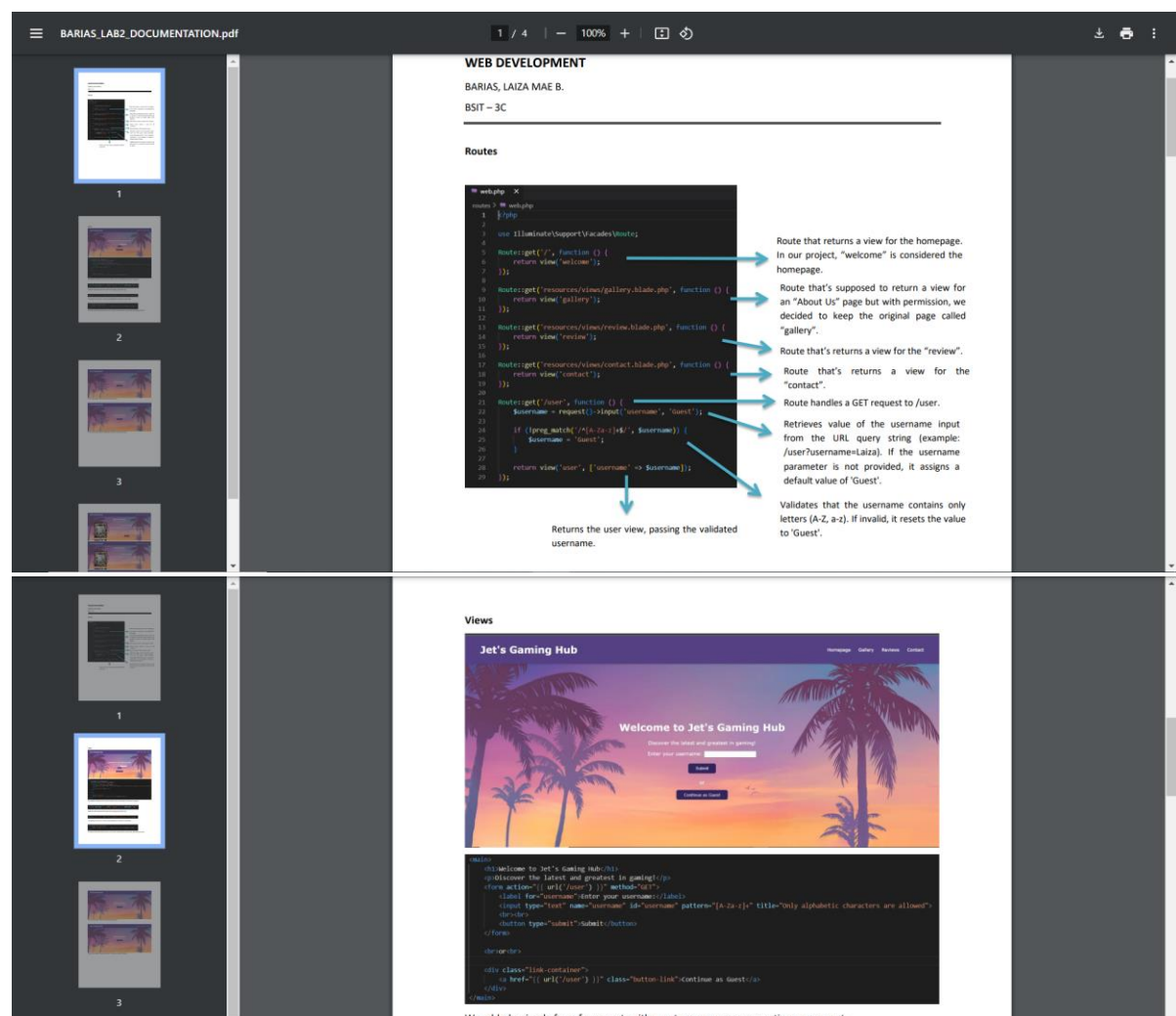
KEY DIRECTORIES:

- app
  - Contains the core application code. This includes the models, controllers, and other classes that make up the application.
- bootstrap
  - Contains files for bootstrapping the Laravel framework.
    - app.php
    - providers.php
- Sub-directory:
  - cache
    - Used to store framework-generated files that improve performance by reducing the need to recompile or regenerate them every time the application runs.

## Lab2



## Lab2 Documentation



Lab3

HomeAbout MeProjectsContact Me

Lab 3: Layout

Jet's Gaming Hub

HomepageGalleryReviewsContact

Welcome to Jet's Gaming Hub

Discover the latest and greatest in gaming!

Don't have an account?

Register Here

Login

Enter your username

Enter your password

☐ Show Password

Login

or

Continue as Guest

Description

In this lab, the instruction was to create a layout file to define a common structure for the views and to create multiple Blade view files that extend this layout.

Summary

In this lab, I learned how to use a layout file in Laravel to create a consistent design for my application. I made a "Components" folder in the resources/views directory and created a Layout.blade.php file that includes the basic HTML structure, such as headers, footers, and navigation. By extending this layout in three different Blade files, I was able to add specific content for each page while keeping the overall look the same. This taught me how layout files help save time and make my code cleaner.

→ View PDF Documentation

Back

Next

© 2024 Laiza's Personal Portfolio. All rights reserved.

Lab3 Documentation

BARIAS\_LAB3\_DOCUMENTATION.pdf

1 / 17 | 100% +

1

2

3

1

2

BARIAS, LAIZA MAE B.  
BSIT – 3C

Web Development Lab: Layout

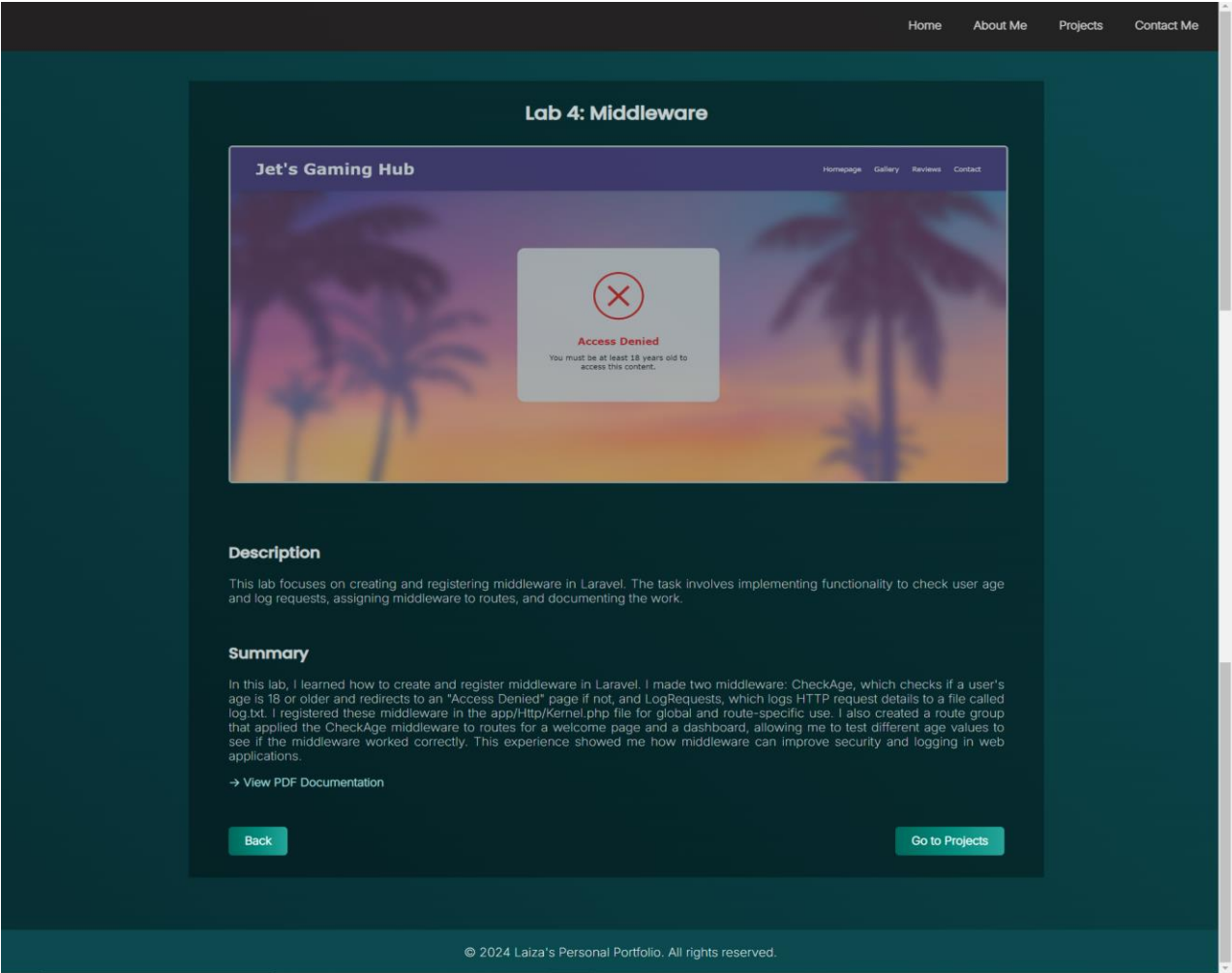
The layout file is a master Blade template that provides a common structure for the website's pages. It usually contains elements that are consistent across multiple views, such as the header, footer, and navigation bar. This approach helps maintain a uniform look and feel throughout the website and avoids repetitive code.

Each view file that extends the layout uses @extends to inherit the structure and @section to insert specific content into the placeholders defined in the layout file.

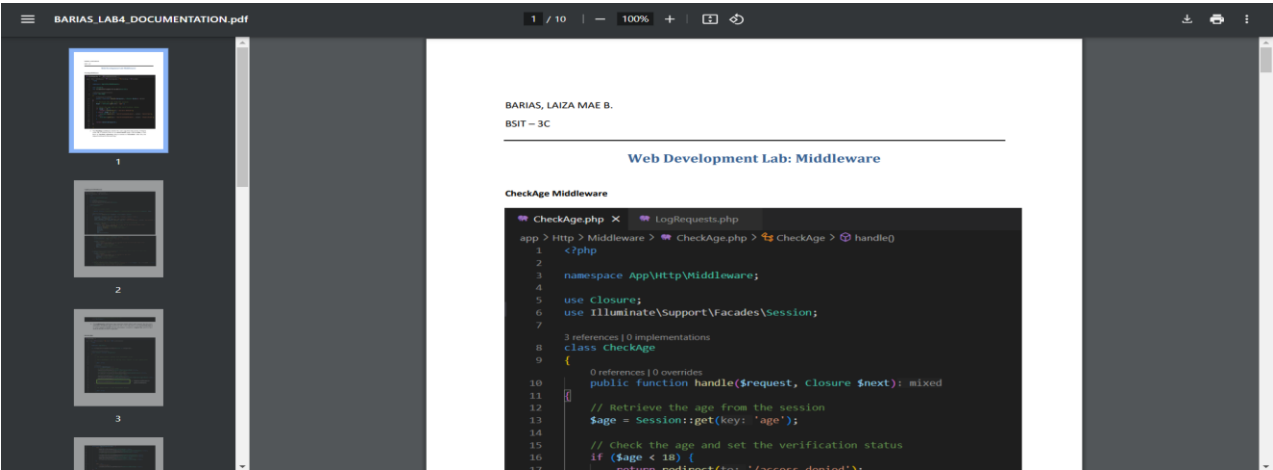
layout.blade.php

```
resource > views > Components > Layout.blade.php
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(path: 'css/app.css') }}" rel="stylesheet">
10  @stack('styles')
11 </head>
12
13 <body class="@yield(section: 'body-class')">
14 <header>
15
16   @stack('styles')
17 </header>
18
19 <body class="@yield(section: 'body-class')">
20 <header>
21   <div class="header-container">
22     <h1>Jet's Gaming Hub</h1>
23     <nav>
24       @if(session(key: 'username') && session(key: 'username') !== 'Guest')
25         <span>Welcome, {{ session(key: 'username') }}</span>
26         <a href="{{ url(path: '/logout') }}">Logout</a>
27       @elseif(request()-is(patterns: '/'))
28         <span>Welcome, Guest</span>
29       @endif
30       <a href="{{ url(path: '/') }}">Homepage</a>
31       <a href="{{ url(path: '/gallery') }}">Gallery</a>
32       <a href="{{ url(path: '/review') }}">Reviews</a>
33       <a href="{{ url(path: '/contact') }}">Contact</a>
34     </nav>
35   </div>
36 </header>
37
38 <main class="@yield(section: 'main-class')">
39   @yield(section: 'content')
40 </main>
41
42 <footer>
43   <div class="footer-container">
44     <p>©copy; 2024 Jet's Gaming Hub. All rights reserved.</p>
45   </div>
46 </footer>
47
48 @stack('scripts')
49 </body>
50 </html>
```

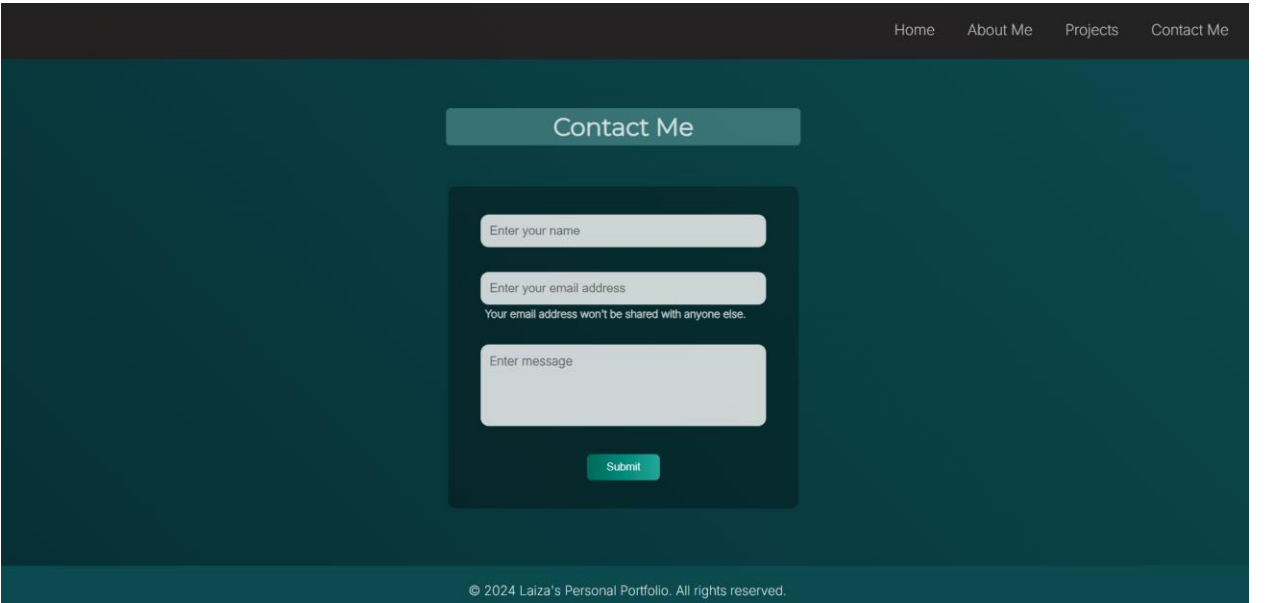
Lab4



Lab4 Documentation



Contact Me





LAYOUT

Layout.blade.php

resources > views > Components > Layout.blade.php

Summart

```
1 <!DOCTYPE html>
2 <html lang="{{ str_replace(search: '_', replace: '-', subject: 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>{{ $title ?? 'Laiza\'s Personal Portfolio' }}</title>
9     <link href="{{ asset(path: 'css/Home.css') }}" rel="stylesheet">
10    @stack('styles')
11    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
12 </head>
13
14 <body class="@yield(section: 'body-class')">
15     @if($title !== 'Landing Page' && $title !== 'Welcome')
16         <header>
17             <div class="header-container">
18                 <nav class="navbar">
19                     <ul>
20                         <li><a href="{{ url(path: '/Home') }}">Home</a></li>
21                         <li><a href="{{ url(path: '/About') }}">About Me</a></li>
22                         <li><a href="{{ url(path: '/Verify') }}">Projects</a></li>
23                         <li><a href="{{ url(path: '/Contact') }}">Contact Me</a></li>
24                     </ul>
25                 </nav>
26             </div>
27         </header>
28     @endif
```

→ /Verify instead of /Projects since it's age-restricted and will need to verify the age first before they can access the Projects page or not.

```
30 <main>
31     @yield(section: 'content')
32 </main>
33
34 @if($title !== 'Landing Page' && $title !== 'Welcome' && $title !== 'Home' && $title !== 'Contact' && $title !== 'Verify Age' && $title !== 'Access Denied')
35     <footer>
36         <p>&copy; 2024 Laiza's Personal Portfolio. All rights reserved.</p>
37     </footer>
38 @endif
39
40 @stack('scripts')
41
42 <script>
43     // navbar scroll
44     const navEl = document.querySelector('.navbar');
45     window.addEventListener('scroll', () => {
46         if (window.scrollY >= 56) {
47             navEl.classList.add('navbar_scrolled');
48         } else if (window.scrollY < 56) {
49             navEl.classList.remove('navbar_scrolled');
50         }
51     });
52 </script>
53 </body>
54
55
56 </html>
```

→ I didn't include the header and footer in some views since it is unnecessary to include them.

→ I also added js for the navbar, so that when users scroll down the page, they won't need to scroll up again to see the navbar.

## Middleware

### CheckAge Middleware

```
CheckAge.php X
app > Http > Middleware > CheckAge.php > ...
1 |<?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7 use Symfony\Component\HttpFoundation\Response;
8 use Illuminate\Support\Facades\Session;
9
10 3 references | 0 implementations
11 class CheckAge
12 {
13     /**
14      * Handle an incoming request.
15      *
16      * @param \Illuminate\Http\Request $request
17      * @param \Closure $next
18      * @param int $minAge
19      * @return mixed
20      */
21     0 references | 0 overrides
22     public function handle(Request $request, Closure $next, $minAge): mixed
23     {
24         $age = Session::get(key: 'age');
25
26         if ($age) {
27             if ($age < $minAge) {
28                 Session::put(key: 'verification_stat', value: 'Not Verified');
29             } else{
30                 Session::put(key: 'verification_stat', value: 'Verified');
31             }
32         }
33
34         return $next($request);
35     }
36 }
```

- ➔ Checks if the user's age, stored in the session, meets the specified minimum age requirement (\$minAge)
- ➔ It retrieves the user's age from the session (Session::get('age')).
- ➔ If the age is less than the required minimum (\$minAge), the middleware sets the session's verification\_stat to "Not Verified."
- ➔ If the age is greater than or equal to \$minAge, it sets verification\_stat to "Verified."



## LogRequests Middleware

LogRequests.php X

app > Http > Middleware > LogRequests.php > ...

> Summart

```
1  k?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8  use Illuminate\Support\Facades\Session;
9
10
11  3 references | 0 implementations
12  class LogRequests
13  {
14      /**
15       * Handle an incoming request.
16       *
17       * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
18       */
19      0 references | 0 overrides
20      public function handle(Request $request, Closure $next): Response
21      {
22          $name = Session::get(key: 'name', default: 'Guest');
23          $age = Session::get(key: 'age');
24          $verification_stat = Session::get(key: 'verification_stat', default: 'def Verified');
25
26          $ipAddress = $request->ip();
27
28          $logData = sprintf(
29              format: "[%s], %s, %s, Name: %s, IP: %s, Age: %s, Verification Status: %s\n",
30              values: now()->toDateTimeString(), // Time
31              $request->method(), // HTTP method (GET, POST, etc.)
32              $request->fullUrl(), // Full URL of the request
33              $name,
34              $ipAddress,
35              $age,
36              $verification_stat
37          );
38
39          // Attempt to log the request details in log.txt file
40          if (file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND) === false) {
41              // Log an error message if writing to the log file fails
42              \Log::error(message: 'Failed to write to log.txt');
43          }
44
45          if ($verification_stat == 'Not Verified') {
46              return redirect(to: '/Access-denied');
47          }
48
49          return $next($request);
50      }
51  }
```

- ➔ Logs request details and handles access control based on the user's age verification status.
- ➔ It retrieves the user's name, age, and verification\_stat from the session.
- ➔ It logs the HTTP method, full URL of the request, the user's name, IP address, age, and verification status to a log file (log.txt).
- ➔ If the verification\_stat is "Not Verified" (the user doesn't meet the age requirement), the user is redirected to the Access-denied page.

log.txt file

log.txt - Notepad

File Edit Format View Help

[2024-10-15 02:31:15], GET, http://localhost:8000/Projects, Name: Guest, IP: 127.0.0.1, Age: 1, Verification Status: Not Verified

[2024-10-15 02:31:36], GET, http://localhost:8000/Projects, Name: LaizaBarias, IP: 127.0.0.1, Age: 12, Verification Status: Not Verified

[2024-10-15 02:31:40], GET, http://localhost:8000/Projects, Name: Guest, IP: 127.0.0.1, Age: 24, Verification Status: Verified

[2024-10-15 02:32:34], GET, http://localhost:8000/Projects, Name: Barias, IP: 127.0.0.1, Age: 44, Verification Status: Verified

[2024-10-15 02:32:38], GET, http://localhost:8000/Projects, Name: Barias, IP: 127.0.0.1, Age: 7, Verification Status: Not Verified

[2024-10-15 02:33:53], GET, http://localhost:8000/Projects, Name: Barias, IP: 127.0.0.1, Age: 77, Verification Status: Verified

[2024-10-15 02:39:38], GET, http://localhost:8000/Projects, Name: hello, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-15 02:41:42], GET, http://localhost:8000/Projects, Name: hello, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-15 02:48:09], GET, http://localhost:8000/Projects, Name: hello, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-15 03:32:46], GET, http://localhost:8000/Projects, Name: hello, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 01:10:42], GET, http://127.0.0.1:8000/Projects, Name: laiza, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 01:21:12], GET, http://127.0.0.1:8000/Projects, Name: laiza, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 01:21:35], GET, http://127.0.0.1:8000/Projects, Name: laiza, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 01:33:52], GET, http://127.0.0.1:8000/Projects, Name: laiza, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 01:39:25], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:06:13], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 12, Verification Status: Not Verified

[2024-10-18 02:06:19], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:07:39], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:10:06], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:10:18], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:21:16], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:21:38], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:24:42], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:28:03], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:30:56], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 02:35:28], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 03:01:53], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 03:06:57], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 03:07:04], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 04:05:44], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

[2024-10-18 04:12:17], GET, http://127.0.0.1:8000/Projects, Name: asd, IP: 127.0.0.1, Age: 21, Verification Status: Verified

<

Ln 1, Col 1

100%

Unix (LF)

UTF-8

>