BARIAS, LAIZA MAE B.
BSIT – 3C

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 atleast 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

routes > web.php

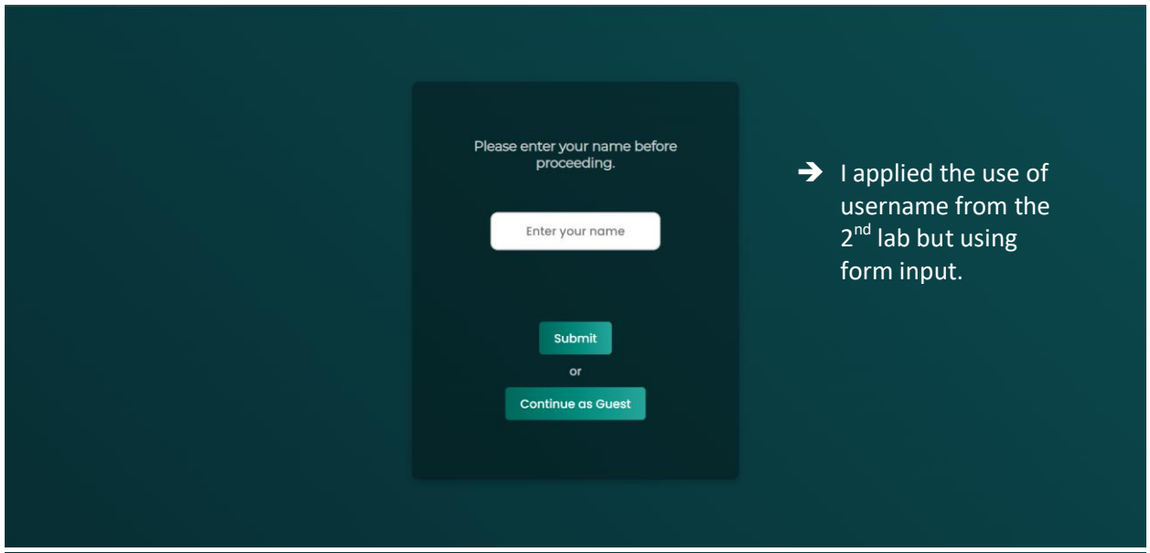
```php
<?php

use Illuminate\Support\Facades\Route;
use Illuminate\Support\Facades\Session;
use App\Http\Middleware\CheckAge; // Import CheckAge middleware
use App\Http\Middleware\LogRequests; // Import LogRequests middleware

// Route for Landing Page
Route::get(uri: '/', action: function (): Factory|View {
    return view(view: 'LandingPage');
});

// Route to handle the form submission
Route::post(uri: '/Welcome', action: function (): Redirector|RedirectResponse {
    // Get the name from the request or default to 'Guest'
    $name = request()->input(key: 'name', default: 'Guest');

    // Validate name to ensure it only contains alphabetic characters
    if (!preg_match(pattern: '/^[A-Za-z]+$/', subject: $name)) {
        $name = 'Guest';
    }

    // Store the name in session
    session(key: ['name' => $name]);

    // Redirect to Welcome page
    return redirect(to: '/Welcome');
});

// GET Route for Welcome page (after form submission)
Route::get(uri: '/Welcome', action: function (): Factory|View {
    // Retrieve the name from the session
    $name = session(key: 'name', default: 'Guest');

    return view(view: 'Welcome', data: ['name' => $name]);
});
```

```php
// Other routes
Route::get(uri: '/Home', action: function (): Factory|View {
    return view(view: 'Home');
});

Route::get(uri: '/About', action: function (): Factory|View {
    return view(view: 'About');
});

Route::get(uri: '/Projects', action: function (): Factory|View {
    return view(view: 'Projects');
});

// Parameterized route to render lab views based on the lab name
Route::get(uri: '/Projects/{Lab}', action: function ($Lab): Factory|View {
    return view(view: $Lab);
});

Route::get(uri: '/Contact', action: function (): Factory|View {
    return view(view: 'Contact');
});

Route::get(uri: '/Verify', action: function (): Factory|View {
    return view(view: 'Verify');
});

// Route to handle "Continue as Guest" link
Route::get(uri: '/Guest', action: function (): Redirector|RedirectResponse {
    // Store 'Guest' in session
    session(key: ['name' => 'Guest']);

    // Redirect to Welcome page
    return redirect(to: '/Welcome');
});
```

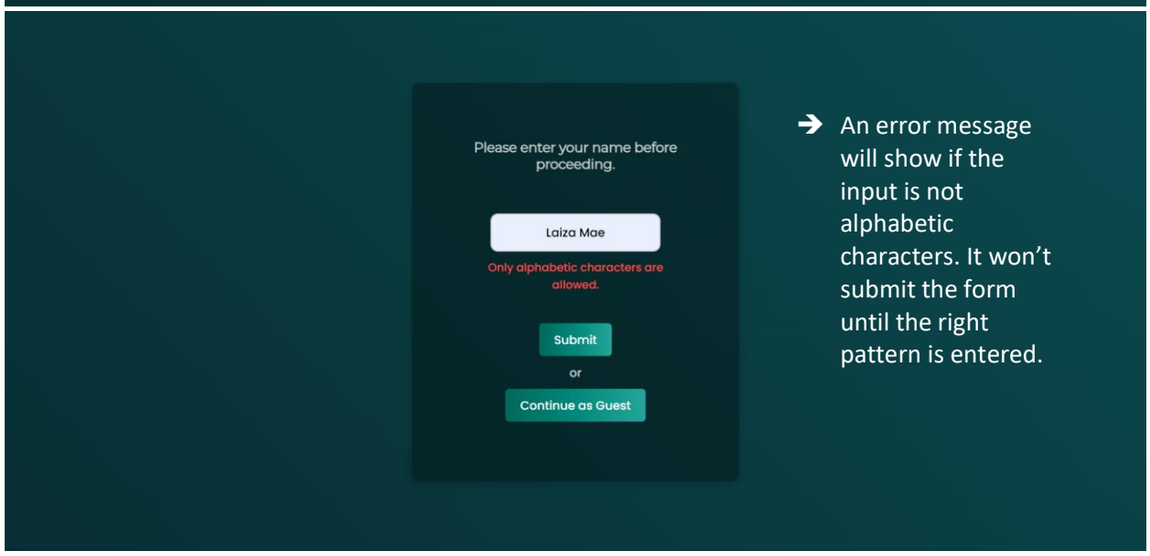
```php
// Route to handle when access is denied
Route::get(uri: '/Access-denied', action: function (): Factory|View {
    // Clear the age and verification_stat from the session
    Session::forget(keys: 'age');
    Session::forget(keys: 'verification_stat');

    return view(view: 'Access-denied');
});

// Route to handle the form submission of age
Route::post(uri: '/verify-age', action: function (): Redirector|RedirectResponse {
    // Get the 'age' from the request input
    $age = request()->input(key: 'age');

    // Store the age in session
    Session::put(key: 'age', value: $age);

    return redirect(to: '/Projects');
});

// Apply CheckAge with a minimum age of 21 for Projects
Route::middleware(middleware: [CheckAge::class . ':21', LogRequests::class])->group(callback: function (): void {
    Route::get(uri: '/Projects', action: function (): Factory|View {

        $name = Session::get(key: 'name', default: 'Guest');
        $age = Session::get(key: 'age');
        $verification_stat = Session::get(key: 'verification_stat');

        return view(view: 'Projects', data: ['name' => $name, 'age' => $age, 'verification_stat' => $verification_stat]);
    });
});
```

VIEWS

Landing Page

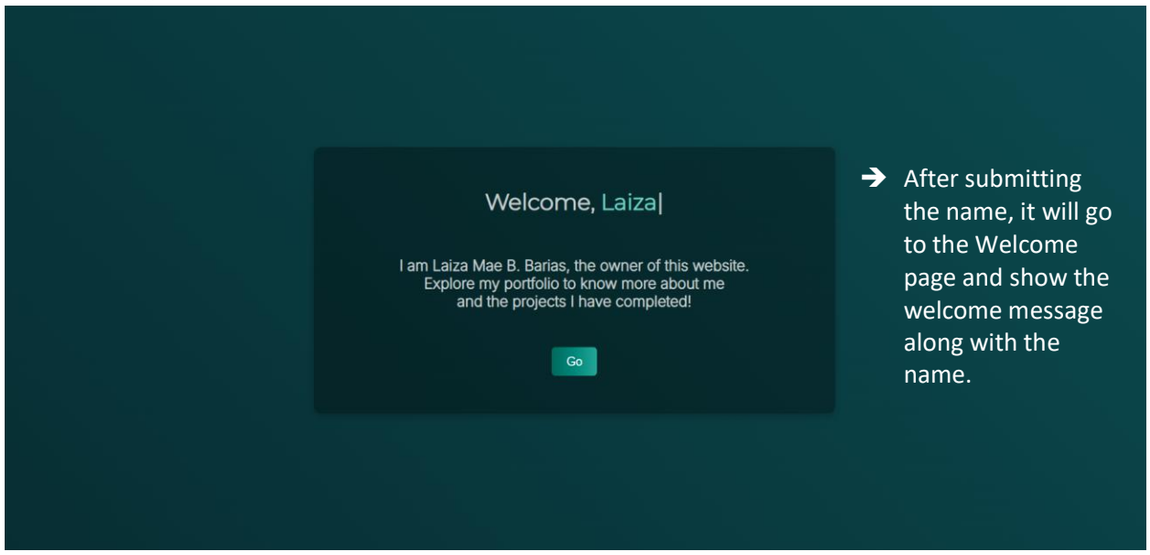


➔ I applied the use of username from the 2nd lab but using form input.

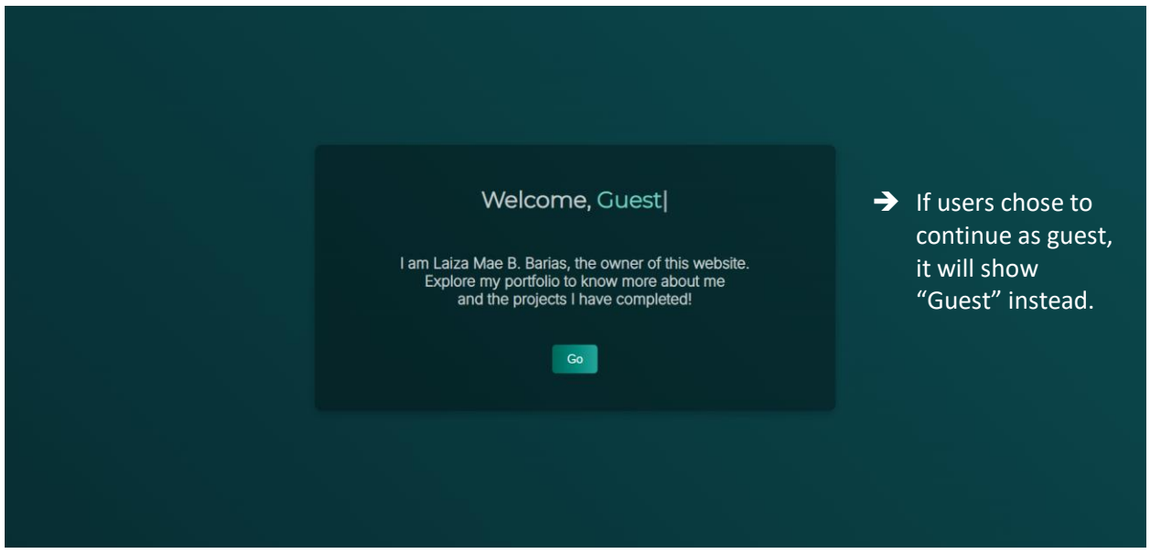


➔ An error message will show if the input is not alphabetic characters. It won't submit the form until the right pattern is entered.

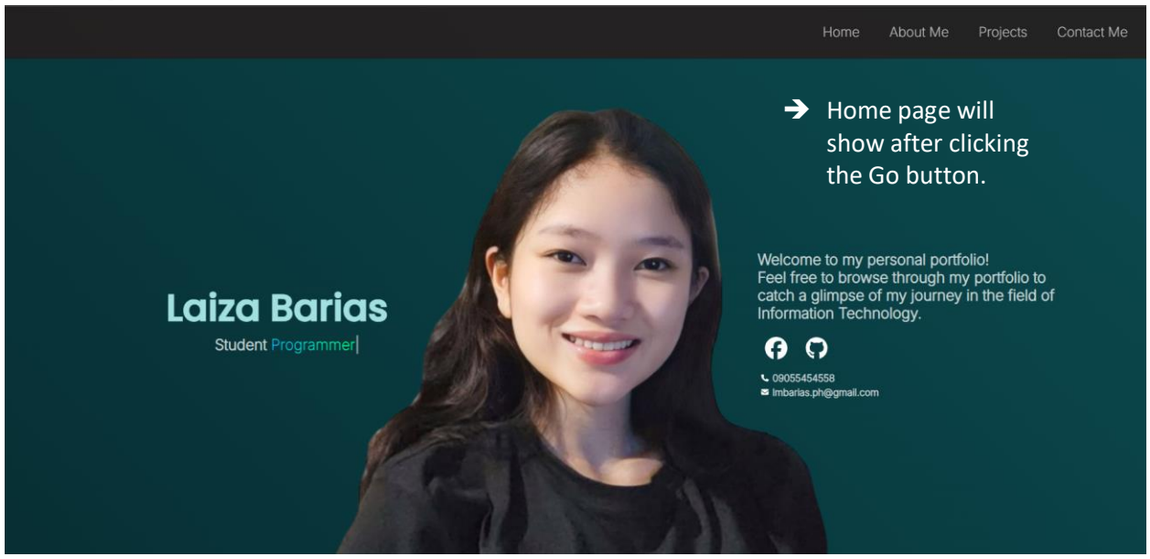
Welcome Page



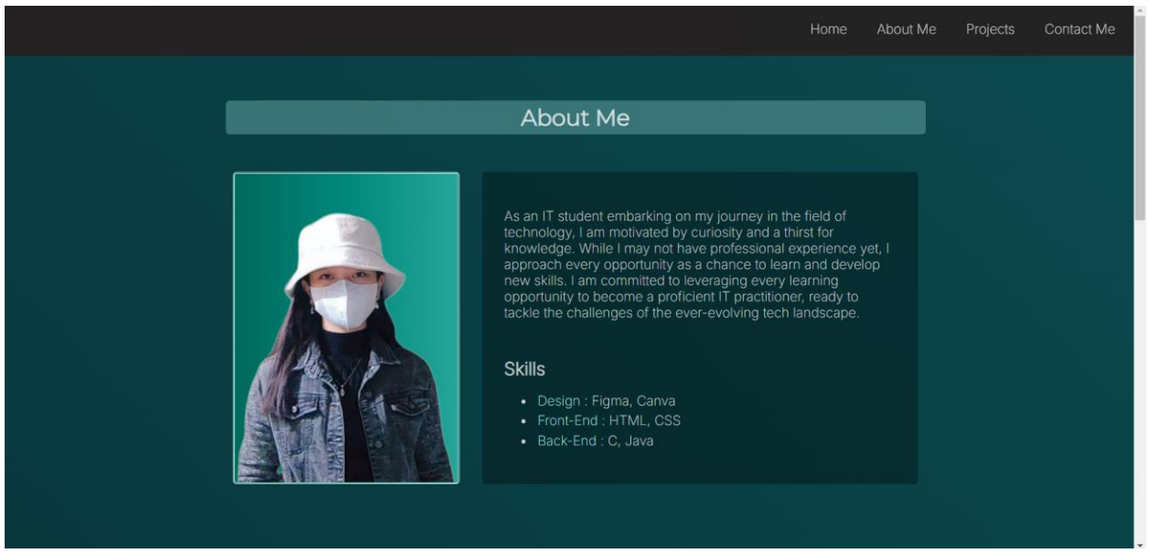
➔ After submitting the name, it will go to the Welcome page and show the welcome message along with the name.



Home Page



About Me Page



Education

Grade	School	Year Studied
Kindergarten	Tabaco Creative Learning Center	2008-2010
Elementary	Tabaco South Central Elementary Schoool	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, manhuas 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

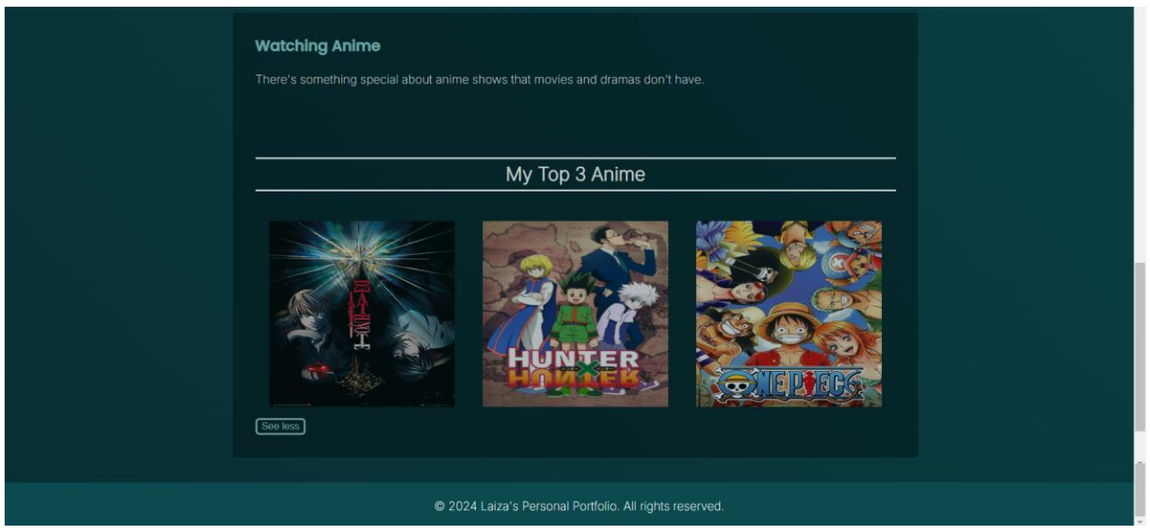
Reading Novels and Manhwas/Mangas

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, manhuas 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.

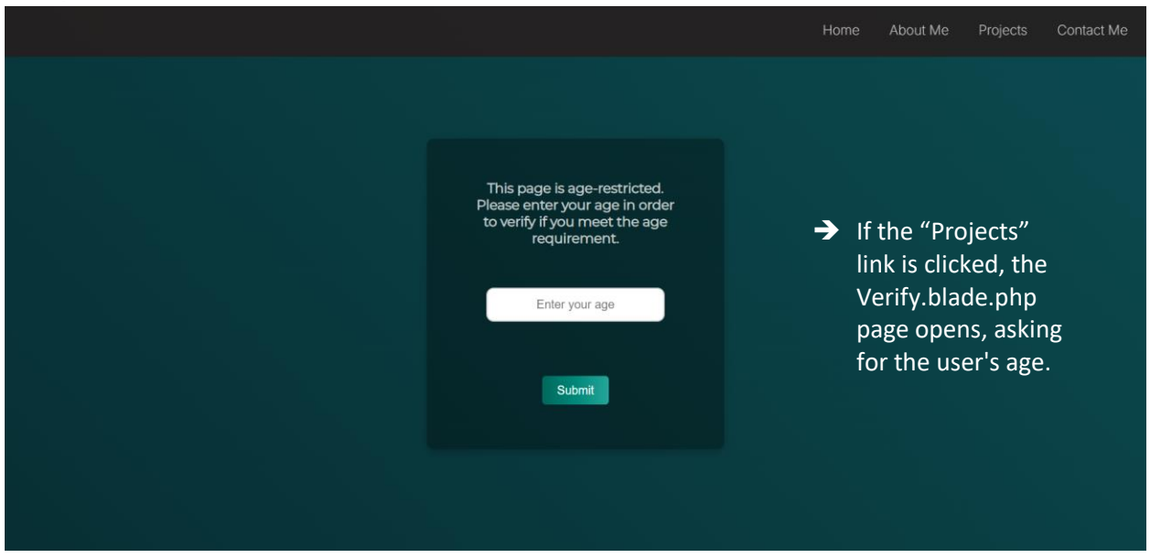
My Top 3 Wattpad Stories

My Top 3 Manhwas

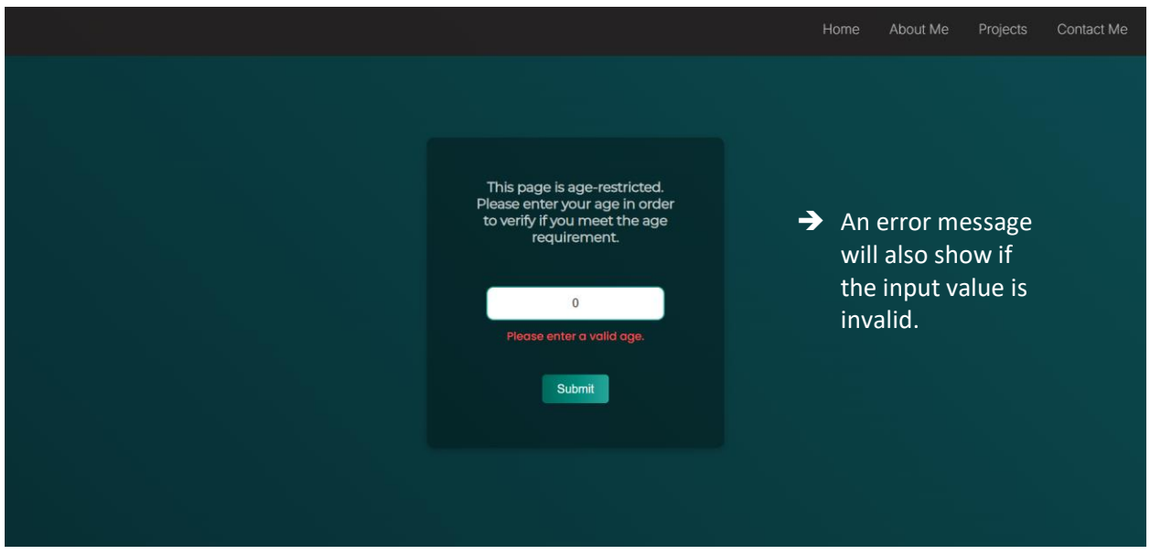
See less



Verify

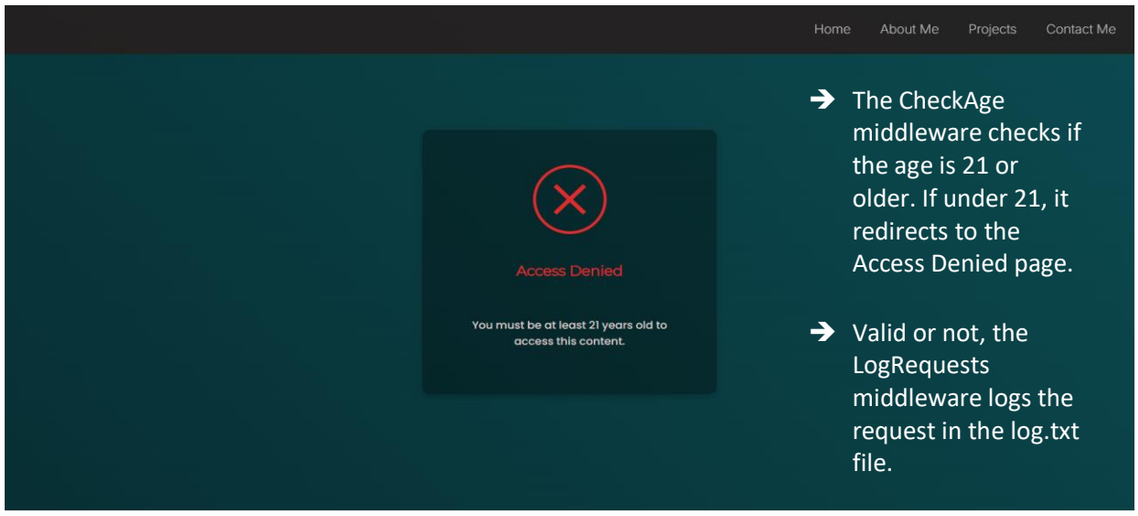


➔ If the “Projects” link is clicked, the Verify.blade.php page opens, asking for the user's age.



➔ An error message will also show if the input value is invalid.

Access Denied

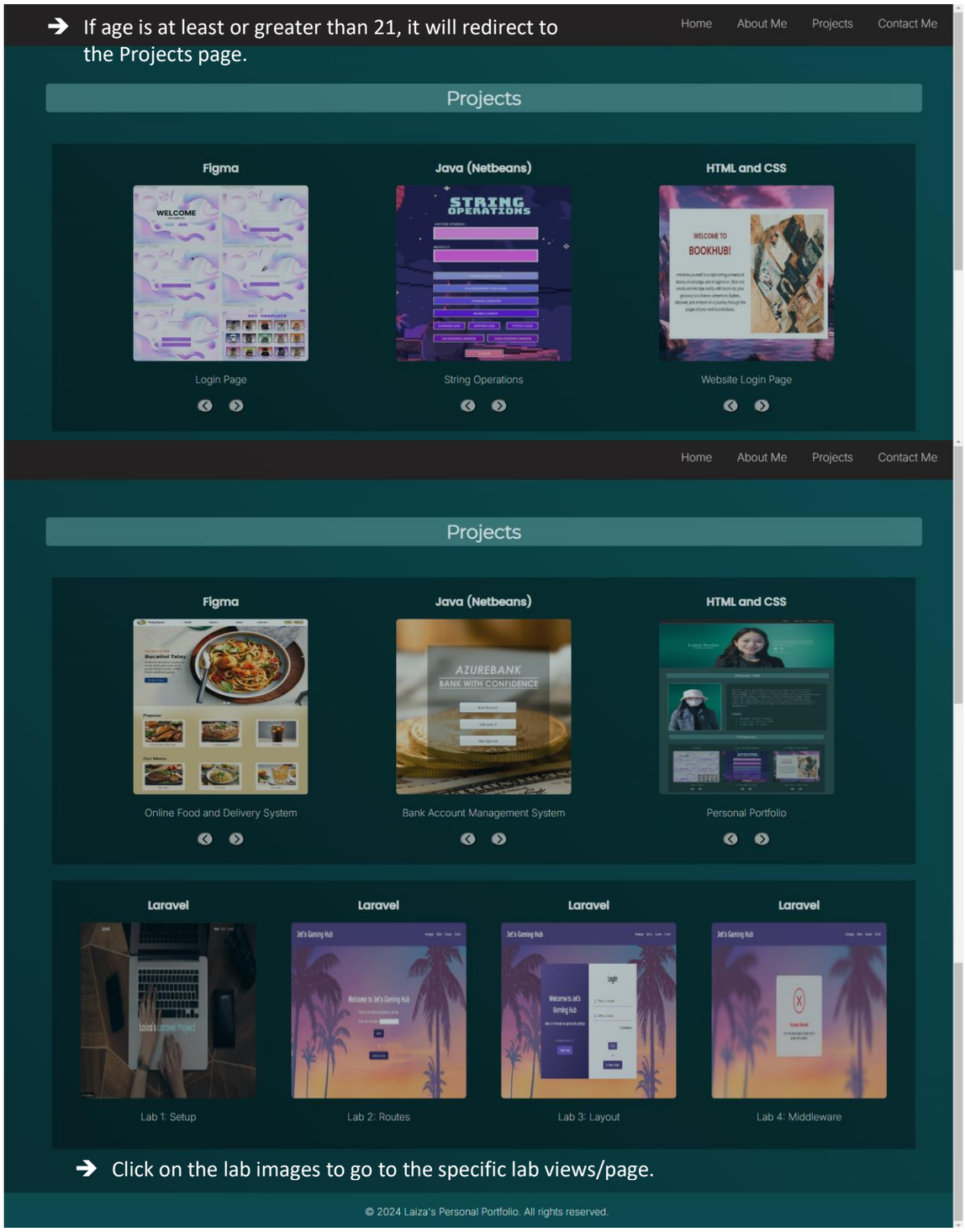


➔ The CheckAge middleware checks if the age is 21 or older. If under 21, it redirects to the Access Denied page.

➔ Valid or not, the LogRequests middleware logs the request in the log.txt file.

Projects

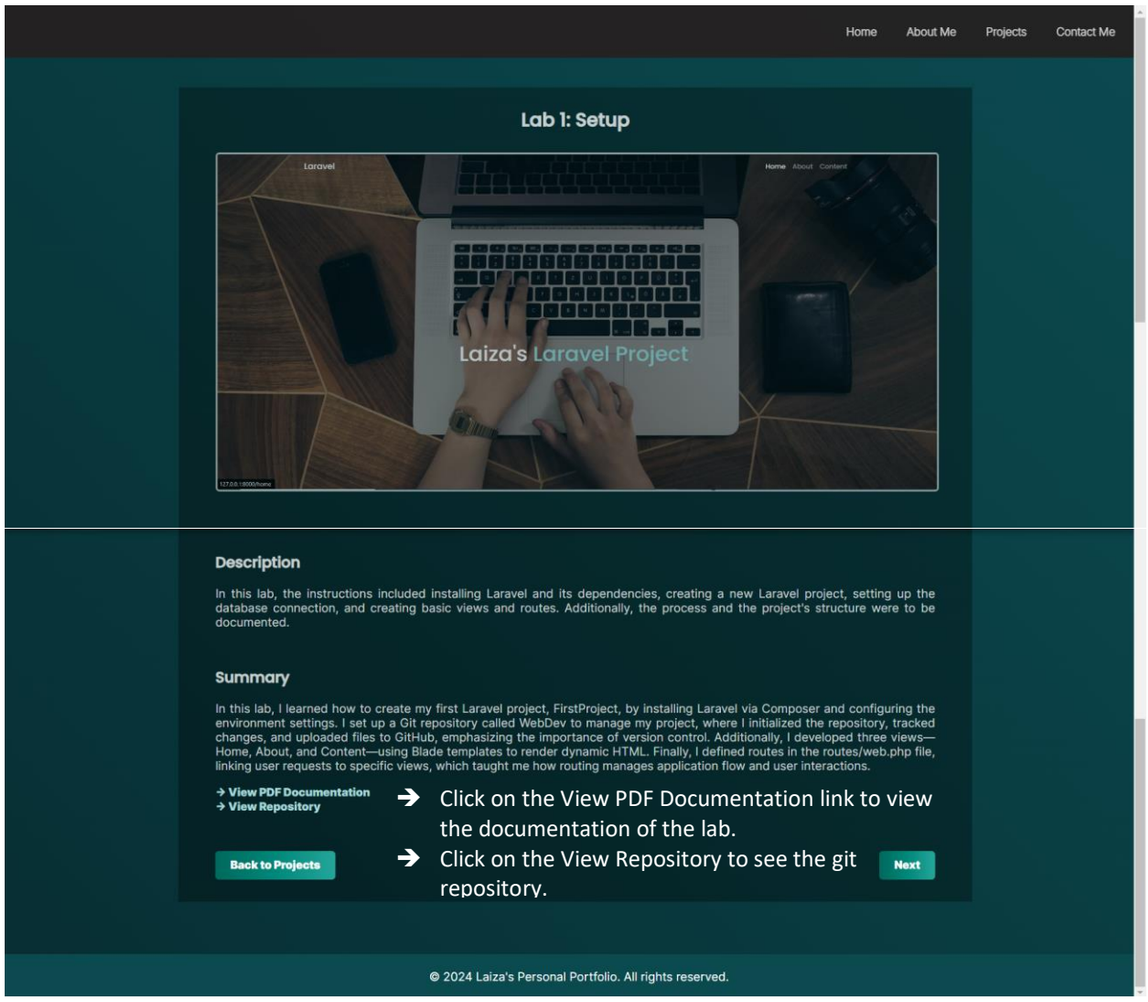
➔ If age is at least or greater than 21, it will redirect to the Projects page.



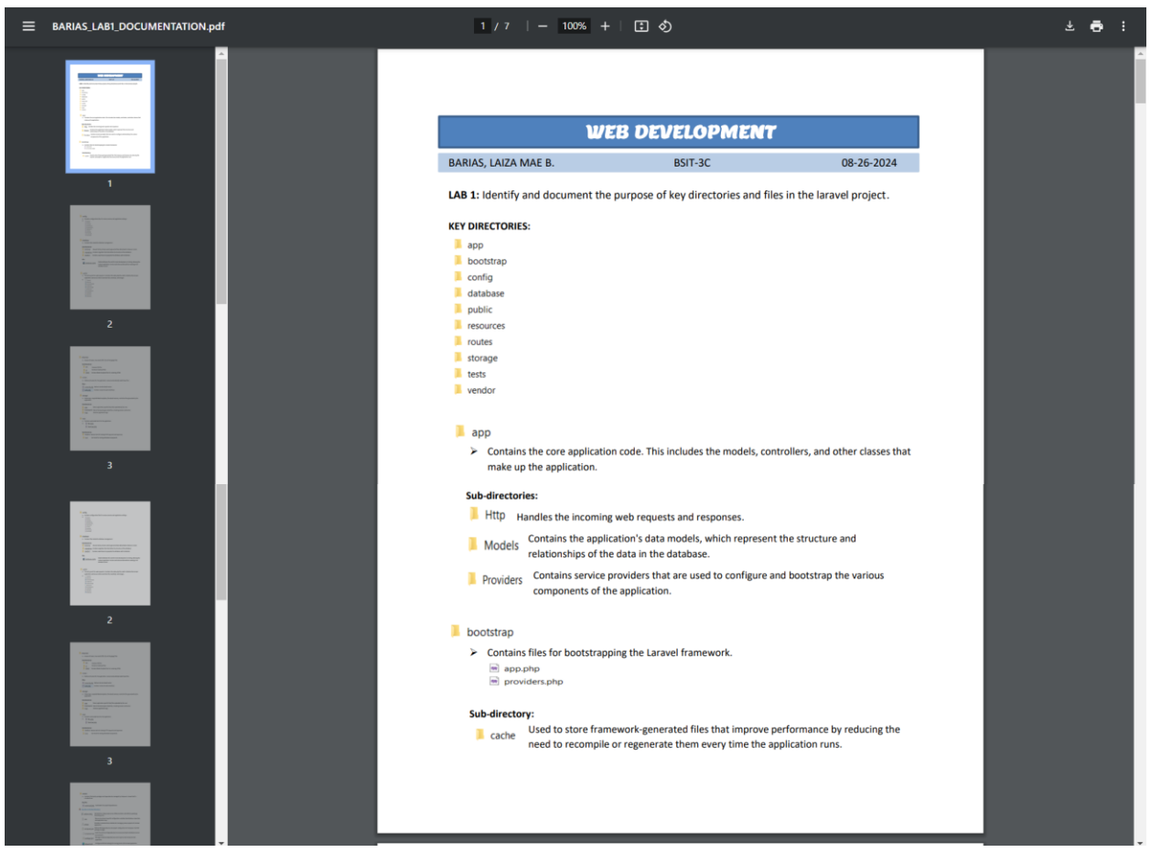
➔ Click on the lab images to go to the specific lab views/page.

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

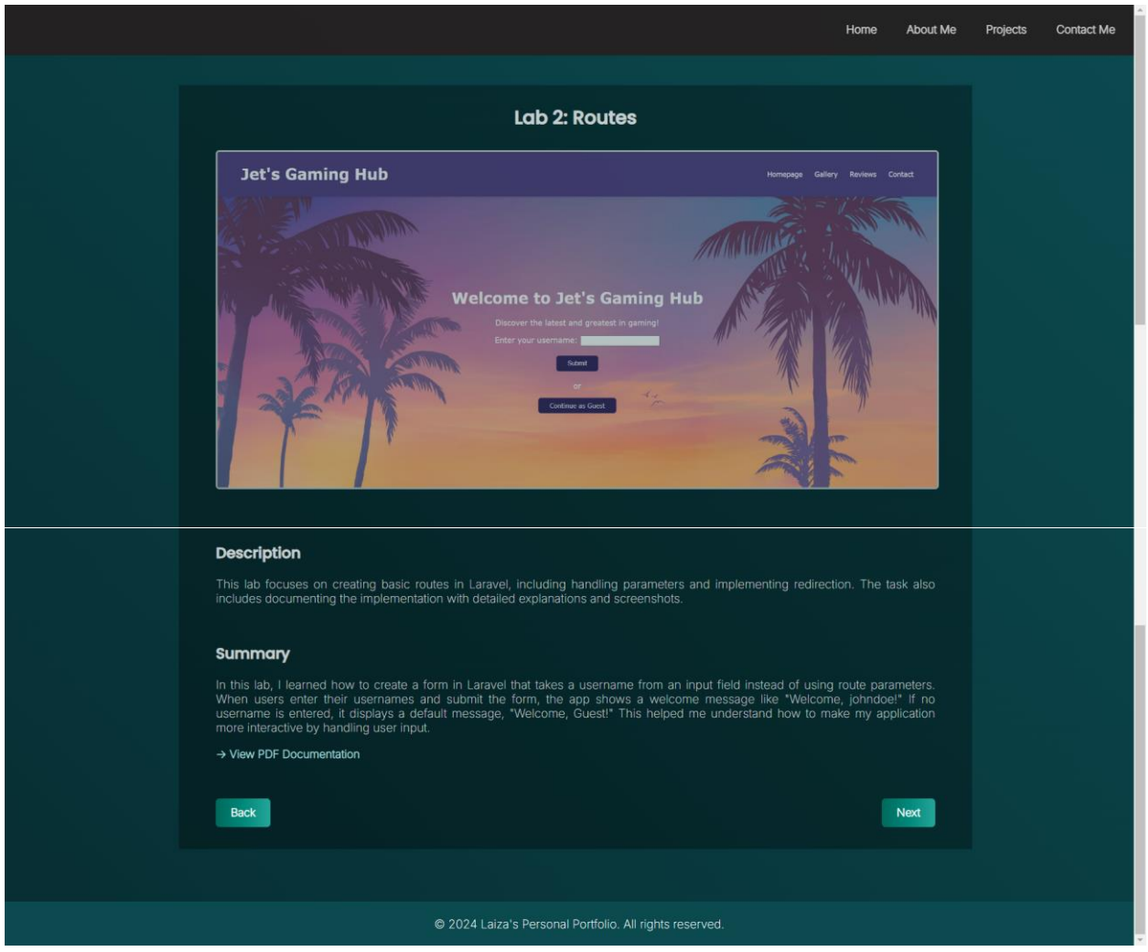
Lab1



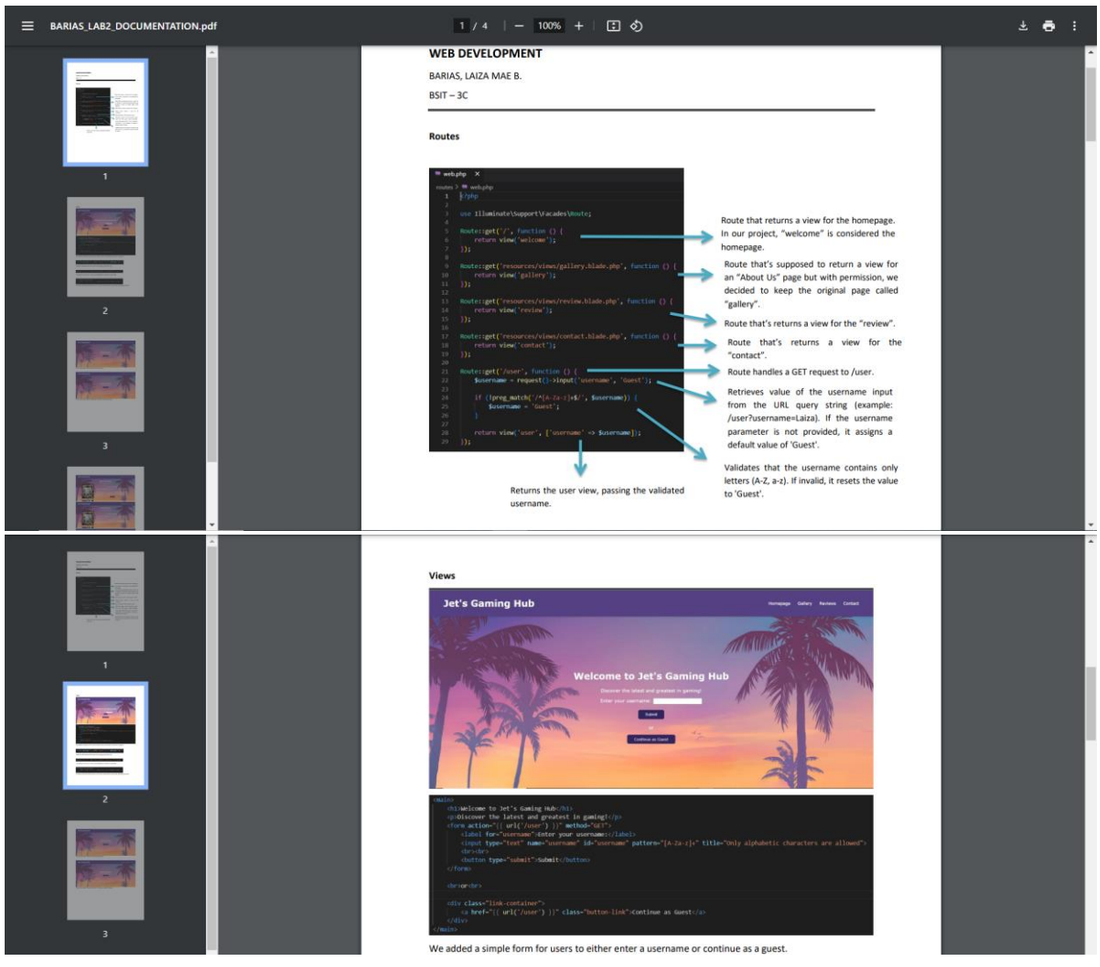
Lab1 Documentation



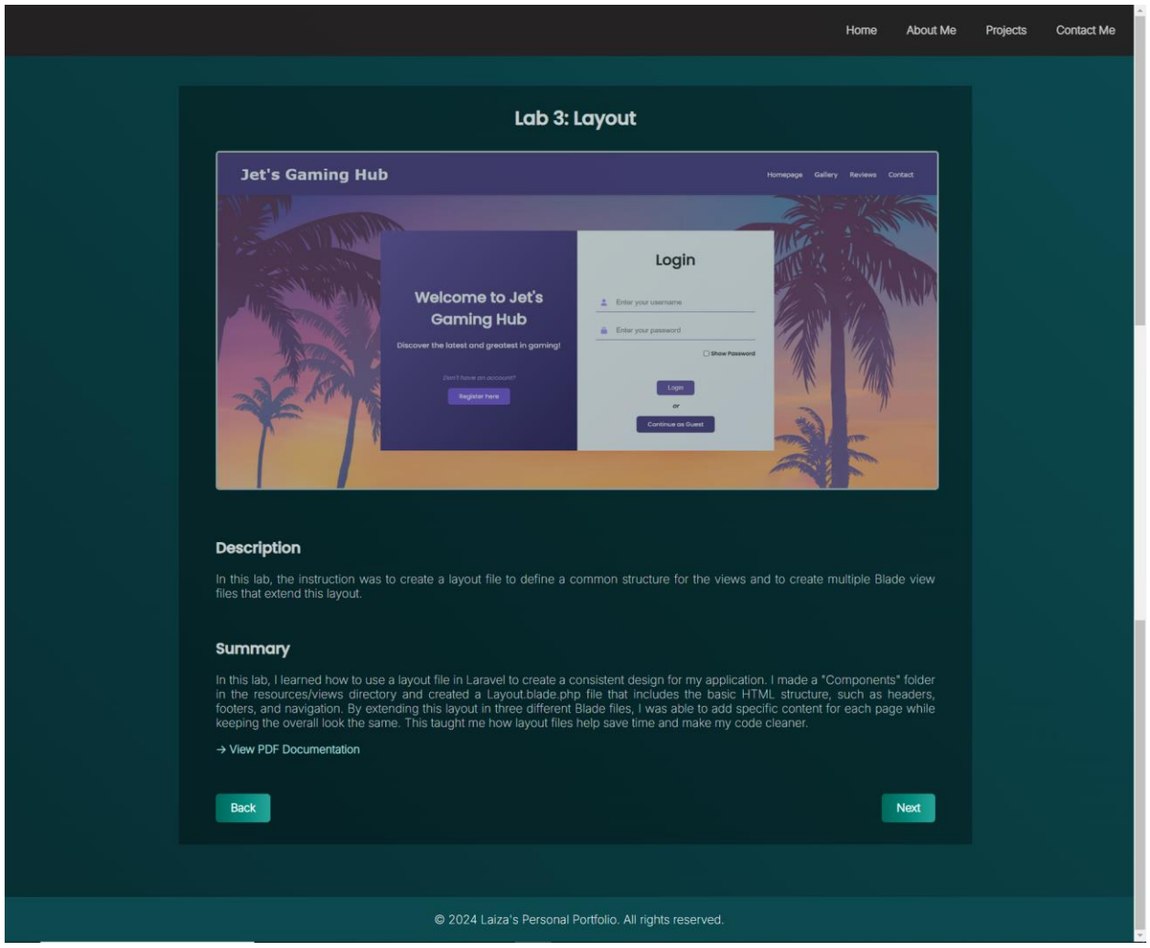
Lab2



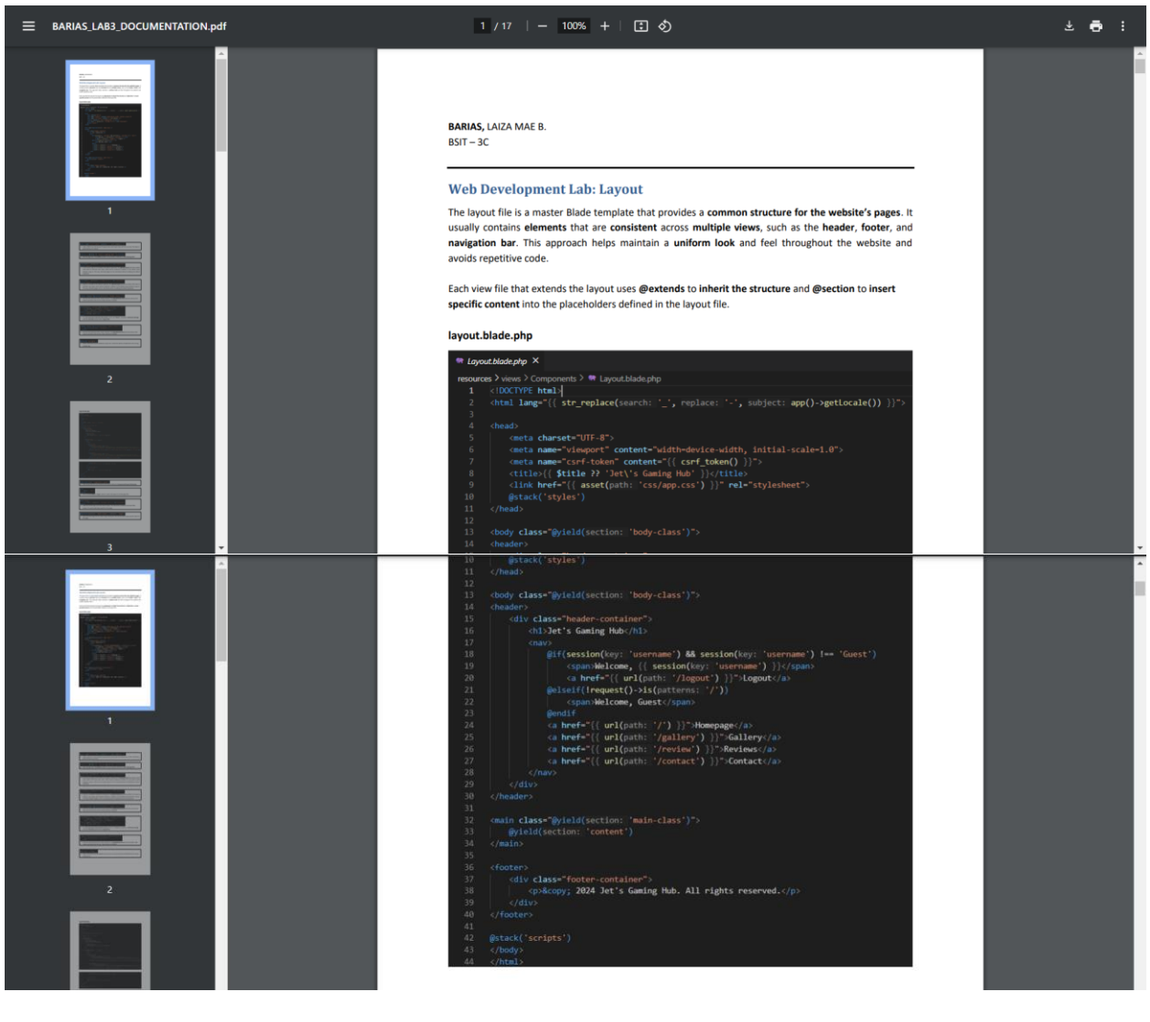
Lab2 Documentation



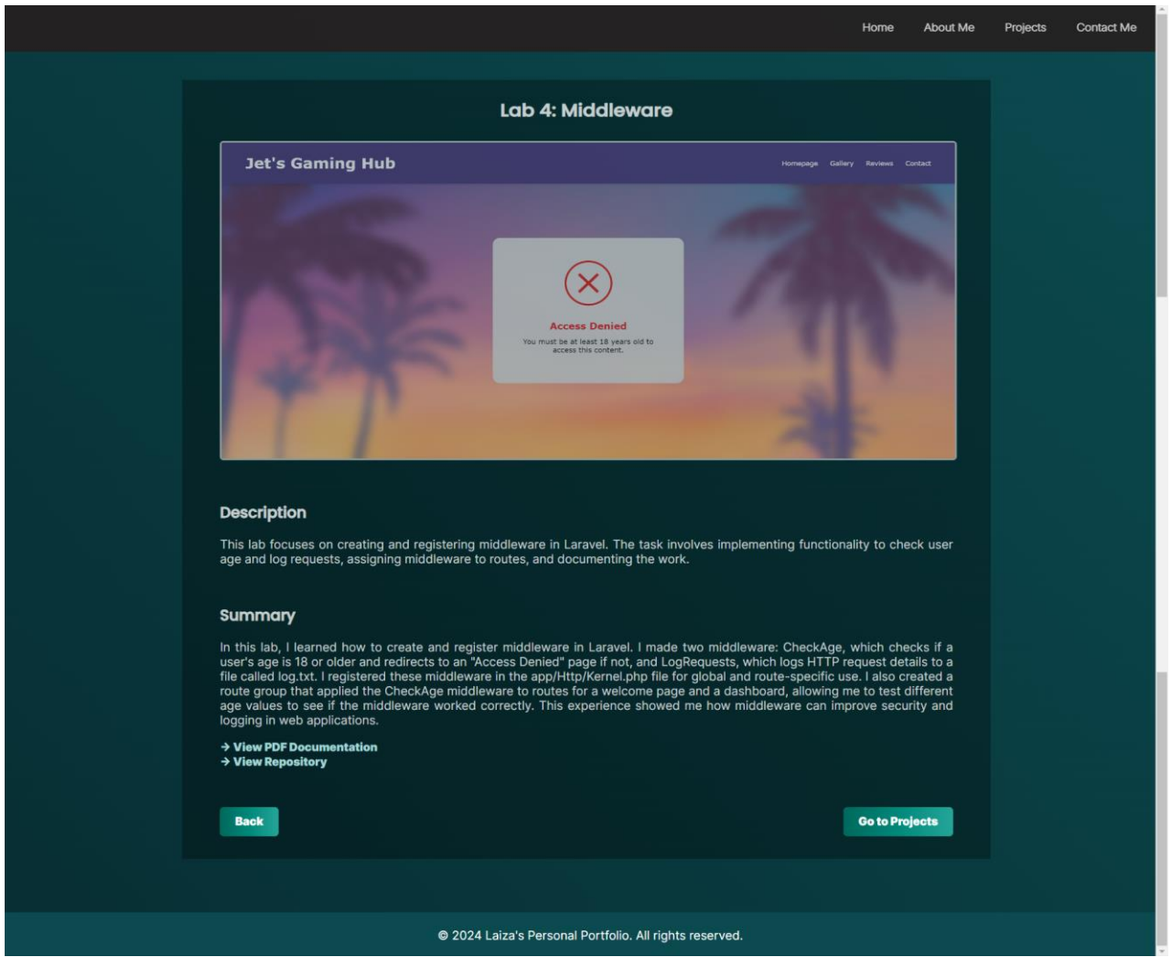
Lab3



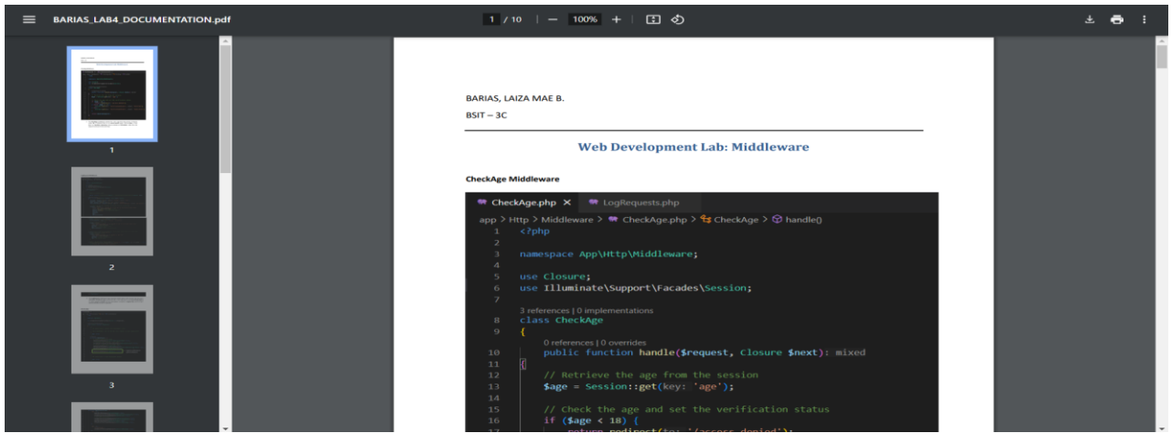
Lab3 Documentation



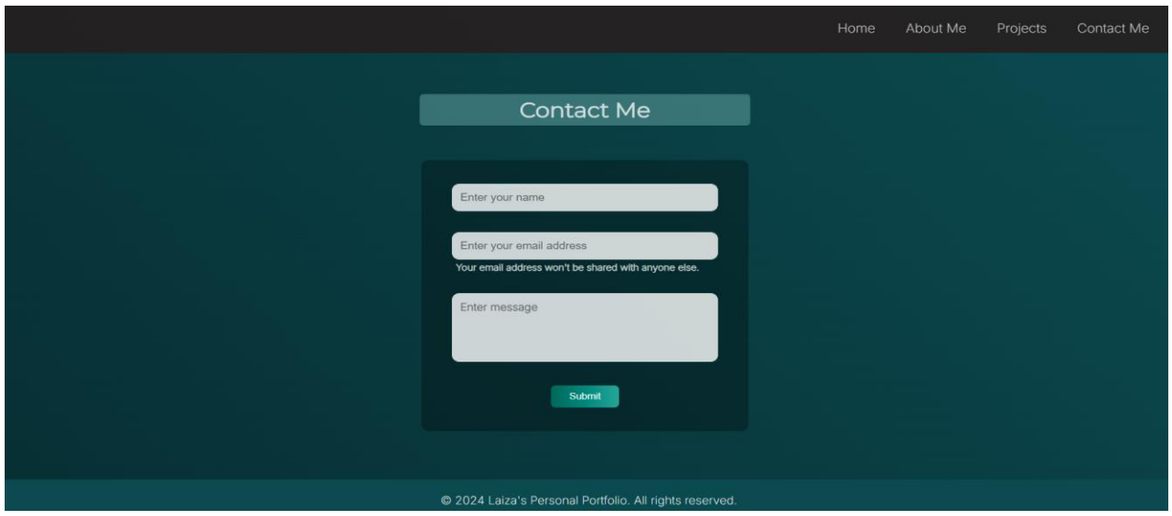
Lab4



Lab4 Documentation



Contact Me



LAYOUT

Layout.blade.php

resources > views > Components > Layout.blade.php

```blade
<!DOCTYPE html>
<html lang="{{ str_replace(search: '_', replace: '-', subject: app()->getLocale()) }}">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="csrf-token" content="{{ csrf_token() }}">
    <title>{{ $title ?? 'Laiza\'s Personal Portfolio' }}</title>
    <link href="{{ asset(path: 'css/Home.css') }}" rel="stylesheet">
    @stack('styles')
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
</head>

<body class="@yield(section: 'body-class')">
    @if($title !== 'Landing Page' && $title !== 'Welcome')
        <header>
            <div class="header-container">
                <nav class="navbar">
                    <ul>
                        <li><a href="{{ url(path: '/Home') }}">Home</a></li>
                        <li><a href="{{ url(path: '/About') }}">About Me</a></li>
                        <li><a href="{{ url(path: '/Verify') }}">Projects</a></li>
                        <li><a href="{{ url(path: '/Contact') }}">Contact Me</a></li>
                    </ul>
                </nav>
            </div>
        </header>
    @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.

```blade
    <main>
        @yield(section: 'content')
    </main>

    @if($title !== 'Landing Page' && $title !== 'Welcome' && $title !== 'Home' && $title !== 'Contact' && $title !== 'Verify Age' && $title !== 'Access Denied')
        <footer>
            <p>&copy; 2024 Laiza's Personal Portfolio. All rights reserved.</p>
        </footer>
    @endif

    @stack('scripts')

    <script>
        // navbar scroll
        const navEl = document.querySelector('.navbar');
        window.addEventListener('scroll', () => {
            if (window.scrollY >= 56) {
                navEl.classList.add('navbar_scrolled');
            } else if (window.scrollY < 56) {
                navEl.classList.remove('navbar_scrolled');
            }
        });
    </script>
</body>


</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

app > Http > Middleware > CheckAge.php > ...

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
use Illuminate\Support\Facades\Session;

3 references | 0 implementations
class CheckAge
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @param  int  $minAge
     * @return mixed
     */
    0 references | 0 overrides
    public function handle(Request $request, Closure $next, $minAge): mixed
{
    $age = Session::get(key: 'age');

    if ($age) {
        if ($age < $minAge) {
            Session::put(key: 'verification_stat', value: 'Not Verified');
        } else{
            Session::put(key: 'verification_stat', value: 'Verified');
        }
    }

    return $next($request);
}

}
```

- ➔ 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

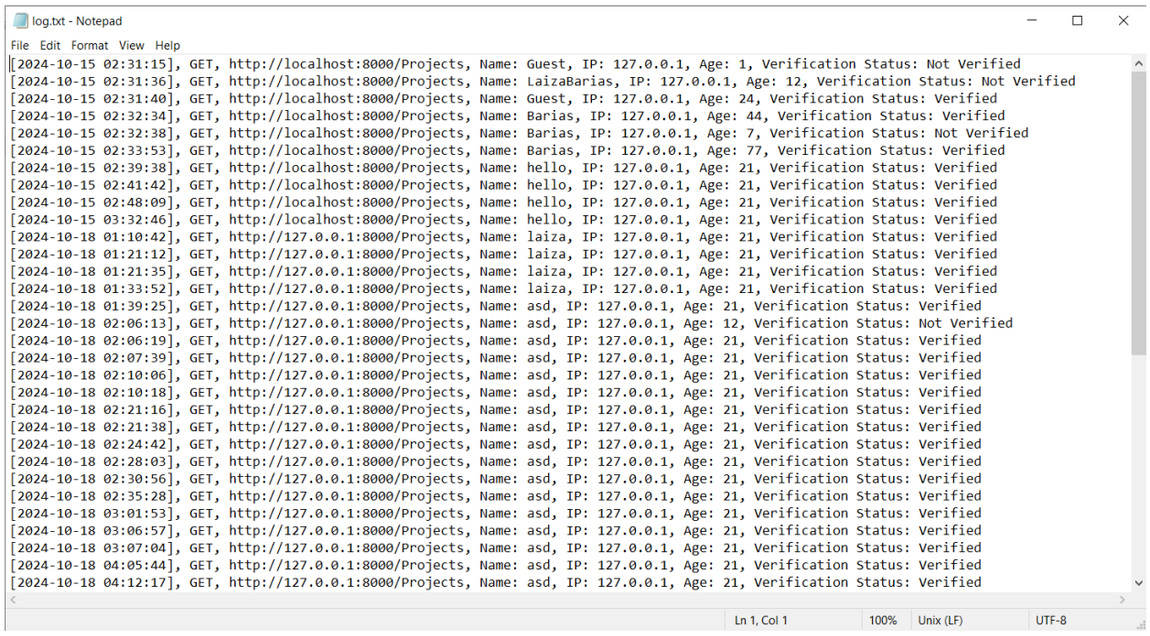
LogRequests.php

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

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
use Illuminate\Support\Facades\Session;


3 references | 0 implementations
class LogRequests
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    0 references | 0 overrides
    public function handle(Request $request, Closure $next): Response
    {
        $name = Session::get(key: 'name', default: 'Guest');
        $age = Session::get(key: 'age');
        $verification_stat = Session::get(key: 'verification_stat', default: 'def Verified');

        $ipAddress = $request->ip();

        $logData = sprintf(
            format: "[%s], %s, %s, Name: %s, IP: %s, Age: %s, Verification Status: %s\n",
            values: now()->toDateTimeString(), // Time
            $request->method(), // HTTP method (GET, POST, etc.)
            $request->fullUrl(), // Full URL of the request
            $name,
            $ipAddress,
            $age,
            $verification_stat
        );

        // Attempt to log the request details in log.txt file
        if (file_put_contents(filename: storage_path(path: 'logs/log.txt'), data: $logData, flags: FILE_APPEND) === false) {
            // Log an error message if writing to the log file fails
            \Log::error(message: 'Failed to write to log.txt');
        }

        if ($verification_stat == 'Not Verified') {
            return redirect(to: '/Access-denied');
        }

        return $next($request);
    }
}
```

- ➔ 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