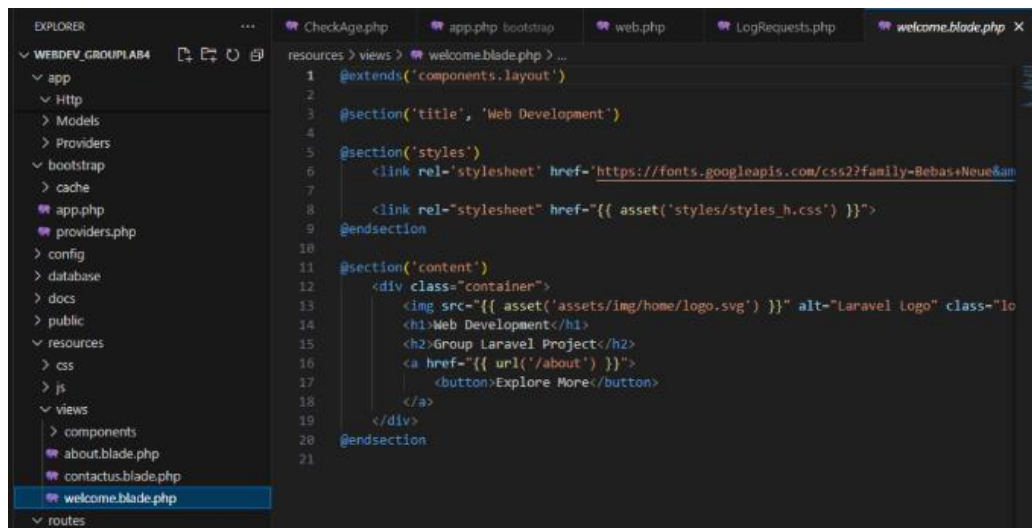


Documentation on Group Laboratory #4

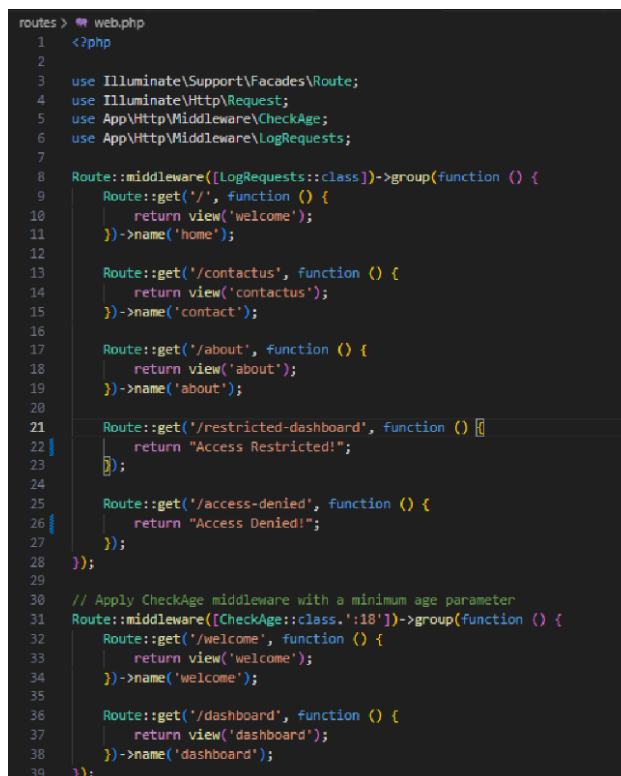
BLADE TEMPLATE FILES

Only the **welcome.blade.php** since the other two remaining blade file doesn't have any changes.



```
1 @extends('components.layout')
2
3 @section('title', 'Web Development')
4
5 @section('styles')
6     <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Bebas+Neue&am
7
8     <link rel="stylesheet" href="{{ asset('styles/styles_h.css') }}">
9 @endsection
10
11 @section('content')
12     <div class="container">
13         Web Development</h1>
15         <h2>Group Laravel Project</h2>
16         <a href="{{ url('/about') }}">
17             <button>Explore More</button>
18         </a>
19     </div>
20 @endsection
21
```

THE ROUTING CONFIGURATION



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use Illuminate\Http\Request;
5 use App\Http\Middleware\CheckAge;
6 use App\Http\Middleware\LogRequests;
7
8 Route::middleware([LogRequests::class])->group(function () {
9     Route::get('/', function () {
10         return view('welcome');
11     })->name('home');
12
13     Route::get('/contactus', function () {
14         return view('contactus');
15     })->name('contact');
16
17     Route::get('/about', function () {
18         return view('about');
19     })->name('about');
20
21     Route::get('/restricted-dashboard', function () {
22         return "Access Restricted!";
23     });
24
25     Route::get('/access-denied', function () {
26         return "Access Denied!";
27     });
28 });
29
30 // Apply CheckAge middleware with a minimum age parameter
31 Route::middleware([CheckAge::class.':18'])->group(function () {
32     Route::get('/welcome', function () {
33         return view('welcome');
34     })->name('welcome');
35
36     Route::get('/dashboard', function () {
37         return view('dashboard');
38     })->name('dashboard');
39 });
```

1. Middleware Application

- The application implements two middleware components, **LogRequests** and **CheckAge**, each serving distinct purposes. The **LogRequests** middleware is assigned to a specific group of routes and is responsible for logging every HTTP request made when users interact with these routes. This helps in tracking user activity and monitoring the system's traffic. On the other hand, the **CheckAge** middleware is designed to enforce an age restriction, where it receives a parameter of 18 to limit access to certain routes. If a user is below the age of 18, they are blocked from accessing these restricted areas, ensuring compliance with age-related access policies.
2. The **routes grouped under the LogRequests middleware** are designed to handle different sections of the application while ensuring that each HTTP request is logged for monitoring purposes.
- **('/')**: This route renders the welcome view, which serves as the home page of the application.
 - **/contactus**: This route displays the contactus view, providing users with the contact page.
 - **/about**: This route loads the about view, offering users information about the purpose or background of the website.
 - **/restricted-dashboard**: When accessed, this route returns a message stating "Access Restricted!", notifying the user that this area is off-limits.
 - **/access-denied**: This route returns a message saying "Access Denied!" This is likely triggered when a minor attempts to access an age-restricted section of the site.

WEBPAGES

- On this page, when the user enters an age of 21 in the URL, the logic applied directs them to the restricted dashboard due to the access restrictions in place.



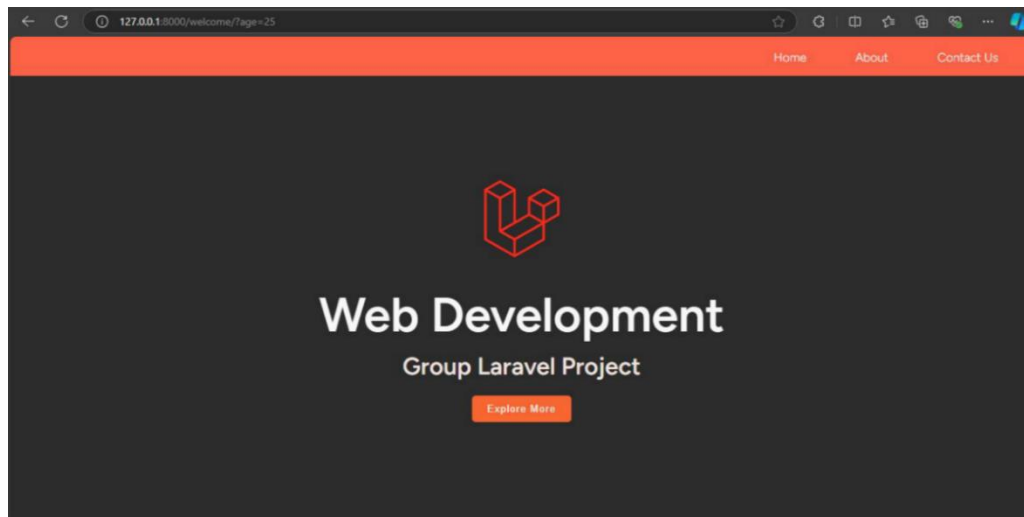
- The welcome page can be freely accessed since the user met the age requirement.



- The page was redirected to **Access Denied** because the user entered an age below 18, as the logic dictates that any user under 18 is denied access to the page.



- On this page, since the age entered in the URL was 25, the user was granted free access to the homepage, as the age logic only restricts access for those under 18 and does not impose any specific restriction for age 21.



GLOBAL MIDDLEWARE

- ***\$app->middleware()***: This function call registers the middleware for the application.
- ***App\Http\Middleware\CheckAge::class***: *It is responsible for checking a user's age before allowing access to certain routes.*
- ***App\Http\Middleware\LogRequests::class***: It is used to log all incoming HTTP requests to track user activity on the routes it applies to.

```
$app->middleware([
    App\Http\Middleware\CheckAge::class,
    App\Http\Middleware\LogRequests::class,
]);
```

```

11 class CheckAge
12 {
13     public function handle(Request $request, Closure $next, $minAge)
14     {
15         // Log the request before redirecting
16         $logData = sprintf(
17             "[%s] %s: %s, Age: %s\n",
18             Carbon::now()->format('Y-m-d H:i:s'),
19             $request->method(),
20             $request->fullUrl(),
21             $request->query('age', 'N/A')
22         );
23
24         file_put_contents(storage_path('logs/log.txt'), $logData, FILE_APPEND);
25
26         if ($request->age < $minAge) {
27             return redirect('access-denied');
28         } elseif ($request->age == 21) {
29             return redirect('restricted-dashboard');
30         }
31
32         return $next($request);
33     }
34 }

```

- The **CheckAge** middleware logs request details, such as the method, URL, and age, into a log file. It checks the user's age, redirecting those under the minimum age to an **"Access Denied"** page and those aged 21 to a **"Restricted Dashboard."** If neither condition is met, the request continues as normal.

```

// Apply CheckAge middleware with a minimum age parameter
Route::middleware([CheckAge::class.':18'])->group(function () {
    Route::get('/welcome', function () {
        return view('welcome');
    })->name('welcome');

    Route::get('/dashboard', function () {
        return view('dashboard');
    })->name('dashboard');
});

```

```

class LogRequests
{
    public function handle(Request $request, Closure $next)
    {
        return $next($request);
    }

    public function terminate(Request $request, $response)
    {
        $logData = sprintf(
            "[%s] %s: %s\n",
            Carbon::now()->format('Y-m-d H:i:s'),
            $request->method(),
            $request->fullUrl()
        );

        file_put_contents(storage_path('logs/log.txt'), $logData, FILE_APPEND);
    }
}

```

- The ***LogRequests*** middleware consists of two methods: ***handle*** and ***terminate***. The ***handle method*** passes the request to the next middleware without making any modifications. The ***terminate method*** logs the request details, including the timestamp, HTTP method, and full URL, by writing this information into a log file after the request is processed.