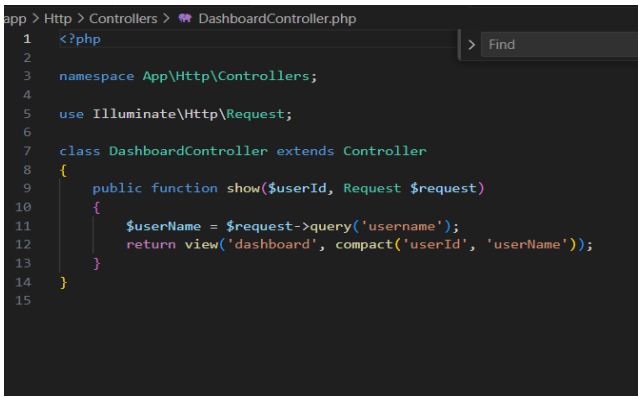
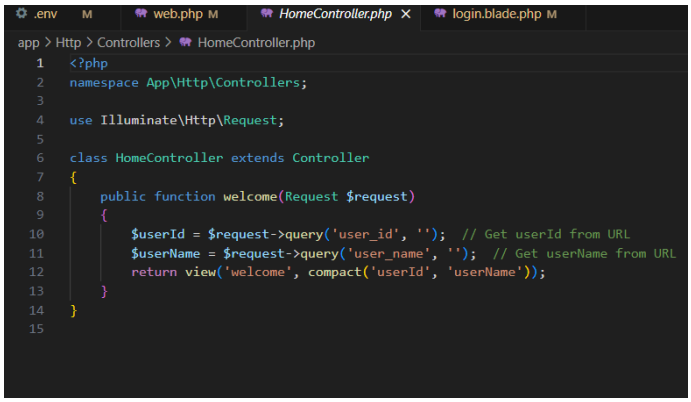
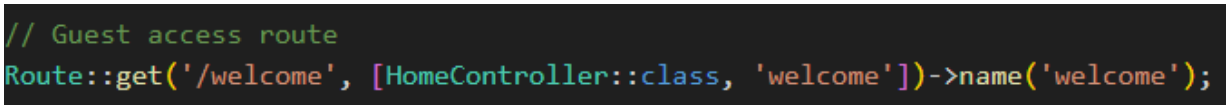
Naje, Niño Miguel L.
BSIT 3C

Controllers Lab Activity

Dashboard and Home Controller:







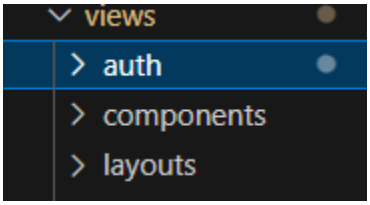
-The DashboardController class manages requests to display user-specific dashboard content. The show method receives a userId as a route parameter and optionally extracts the username from the query parameters in the request. This method returns the dashboard view, providing it with the userId and userName data for rendering personalized content.

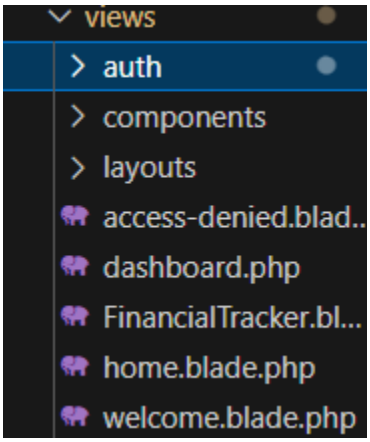
-The HomeController class handles the logic for displaying the welcome page. In the welcome method, user information such as userId and userName is retrieved from the query parameters in the URL. This information is then passed to a view called welcome to be displayed on the page. The compact() function is used to bundle the userId and userName variables and pass them to the view in a concise manner.

-The route Route::get('/welcome', [HomeController::class, 'welcome'])->name('welcome'); defines a path that allows users to access the /welcome page. This route is connected to the welcome method in the HomeController class and is named welcome, making it easy to reference in other parts of the application.

-When a user navigates to /welcome, the welcome method of the HomeController is triggered, gathering and passing user data to the welcome view. If the user accesses /dashboard/{userId}, the show method in DashboardController is executed, and the dashboard page is rendered using the user's specific information.

View:





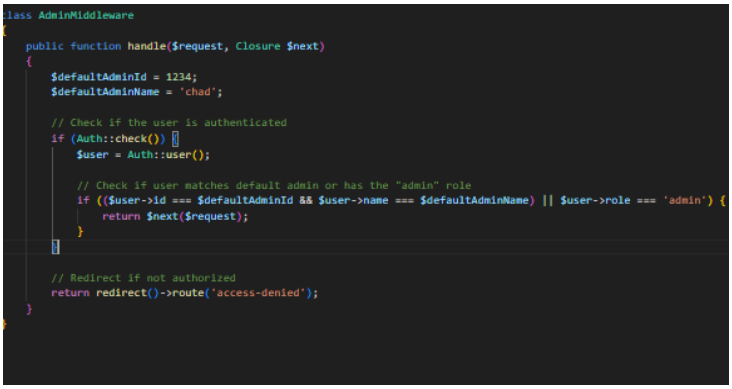
Routes:

```
routes > web.php

// Login routes
Route::get('/login', function () {
    return view('auth.login');
})->name('login');

Route::post('/login', function (Request $request) {
    $userId = $request->input('user_id');
    $userName = $request->input('user_name');

    // Simulate login for demonstration (replace with actual login logic)
    Auth::loginUsingId($userId);

    // Debugging output to verify role
    \Log::info('User ID: ' . Auth::id());
    \Log::info('User Role: ' . Auth::user()->role);  // Logs the user's role

    // Redirect based on role
    if (Auth::user()->role === 'admin') {
        return redirect()->route('dashboard.show', ['userId' => Auth::id()]);
    }

    return redirect()->route('welcome');
})->name('login.submit');

// Guest access route
Route::get('/welcome', [HomeController::class, 'welcome'])->name('welcome');

// Financial Tracker route (accessible without authentication)
Route::get('/FinancialTracker', function () {
    return view('FinancialTracker');
})->name('financial.tracker');

// Access-denied route for unauthorized access
Route::get('/access-denied', function () {
    return view('access-denied');
})->name('access-denied');

// Dashboard route restricted to admins only
Route::get('/dashboard/{userId}', function ($userId) {
    dd('Dashboard route hit', $userId); // Temporarily dump the userId for debugging

    if (Auth::check() && Auth::user()->role === 'admin') {
        return app(DashboardController::class)->show($userId);
    } else {
        return redirect()->route('access-denied');
    }
})->name('dashboard.show');

// Catch-all route for unauthorized access to any undefined paths within the dashboard
Route::get('/dashboard/{any}', function () {
    return redirect()->route('access-denied');
})->where('any', '.*');
```

The **views folder** contains several Blade template files used to display different pages of the web application. These include:

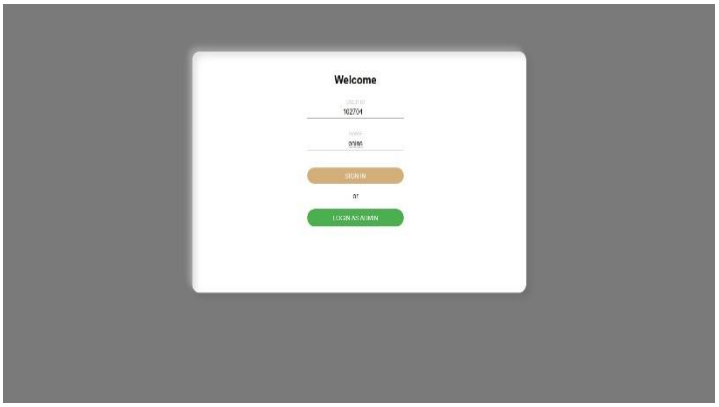
- **auth folder**: Contains authentication-related views, such as the login page.
- **dashboard.blade.php**: The view for the user dashboard page.
- **FinancialTracker.blade.php**: A view dedicated to tracking financial data.
- **home.blade.php**: The view for the main or home page.
- **welcome.blade.php**: The view for the welcome page, often serving as an introductory page for users.

- **Login Route (/)**: The route Route::get('/', function () {...})->name('login'); handles the root URL and returns the login view. This is typically where users start when visiting the site.
- **Login Submission Route (/login)**: Handles form submissions for logging in and redirects users to the dashboard upon successful login.
- **Welcome Page Route (/welcome)**: The Route::get('/welcome', ...) is for accessing the welcome page, which is publicly available.
- **Financial Tracker Route (/FinancialTracker)**: This route provides access to the financial tracking view without needing user authentication.
- **Access Denied Route (/access-denied)**: Used to display a message when a user tries to access restricted pages without proper authorization

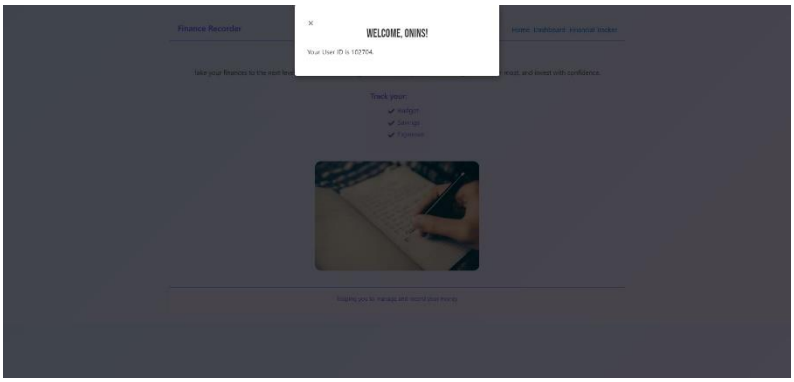


The AdminMiddleware class checks if a user has the appropriate permissions to access certain parts of the application, such as the dashboard. The middleware:

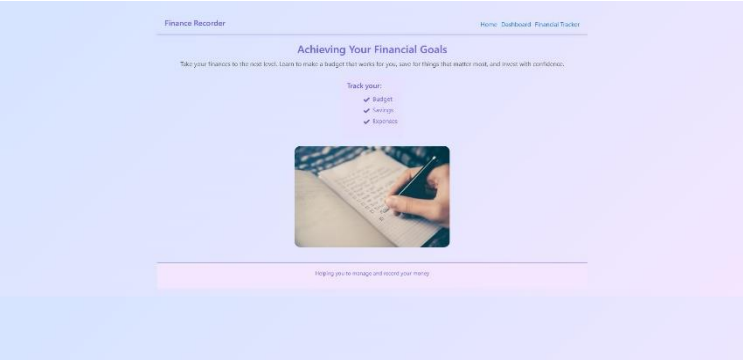
- Confirms if the user is logged in and if their role matches the required one (e.g., admin).
- If the user is not authorized, they are redirected to the access-denied page.
- This middleware ensures that only users with the correct role can access sensitive or restricted areas of the site, enhancing security.



Login page is designed to capture user credentials such as userID and username. It features input fields for users to enter these details and buttons for signing in as a regular user or as an admin. Once the user provides their credentials and clicks "Sign In," the system verifies the information and navigates them to the next step.



After successfully logging in, users are redirected to a **welcome page** that displays a modal greeting them by their name and showing their userID. This modal provides a friendly acknowledgment of their successful login and serves as a brief confirmation.



The **home page** is a main content area that welcomes users to the application. It displays information under a header, such as "Achieving Your Financial Goals," with supportive content encouraging users to take control of their finances.
The page includes links to navigate to other key sections like the **Dashboard** and **Financial Tracker**, ensuring that users can easily find the tools they need.The content highlights the app's key features, such as tracking budgets, savings, and expenses, with a clean layout and visual prompts.