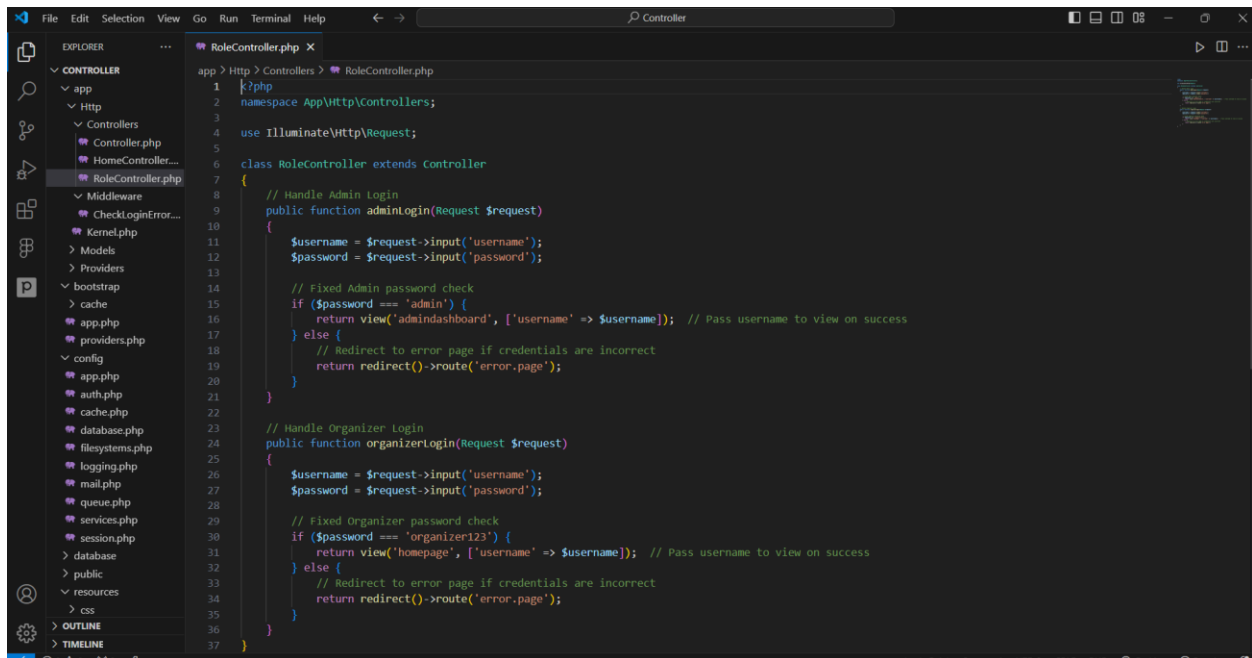
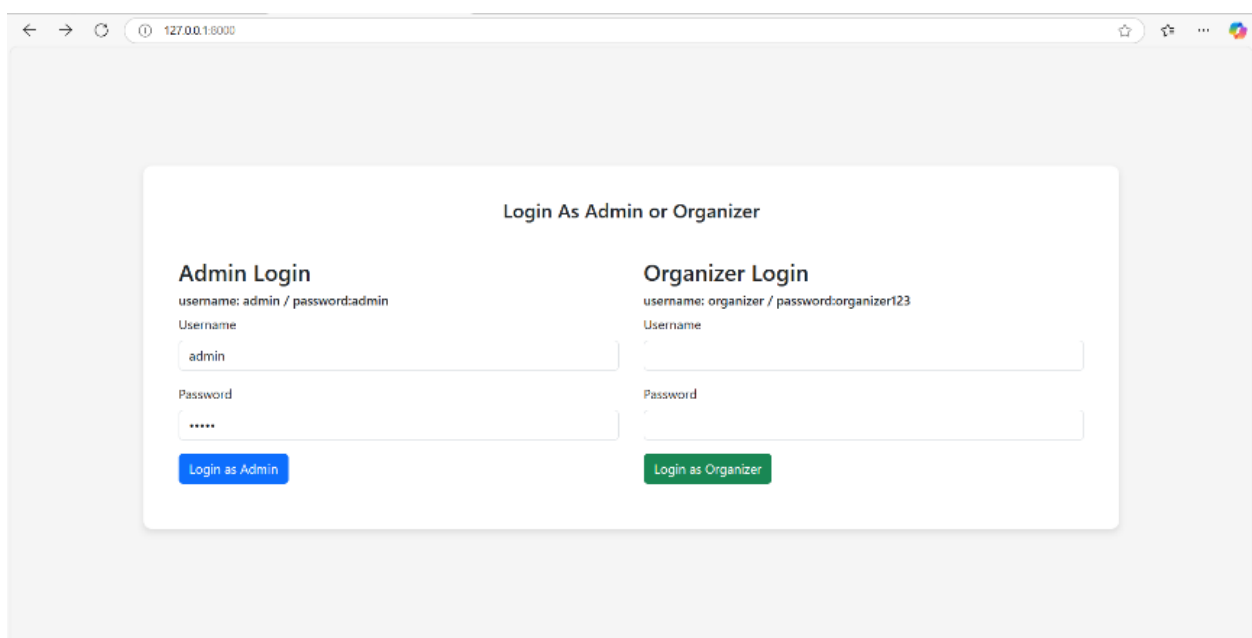


## CONTROLLER

**RoleController** manages the login request for the two user roles: **admin** and **organizer**. It contains two main methods the **adminLogin** and **organizerLogin**. The **adminLogin** method retrieves the **username** and **password** from the request. If the password matches the fixed value “**admin**”, the user is redirected to the **admindashboard** view, with username passed to the view, but if the password is incorrect, it redirects to an error page.



```
1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5
6 class RoleController extends Controller
7 {
8     // Handle Admin Login
9     public function adminLogin(Request $request)
10     {
11         $username = $request->input('username');
12         $password = $request->input('password');
13
14         // Fixed Admin password check
15         if ($password === 'admin') {
16             return view('admindashboard', ['username' => $username]); // Pass username to view on success
17         } else {
18             // Redirect to error page if credentials are incorrect
19             return redirect()->route('error.page');
20         }
21     }
22
23     // Handle Organizer Login
24     public function organizerLogin(Request $request)
25     {
26         $username = $request->input('username');
27         $password = $request->input('password');
28
29         // Fixed Organizer password check
30         if ($password === 'organizer123') {
31             return view('homepage', ['username' => $username]); // Pass username to view on success
32         } else {
33             // Redirect to error page if credentials are incorrect
34             return redirect()->route('error.page');
35         }
36     }
37 }
```



### Login As Admin or Organizer

#### Admin Login

username: admin / password:admin

Username

Password

Login as Admin

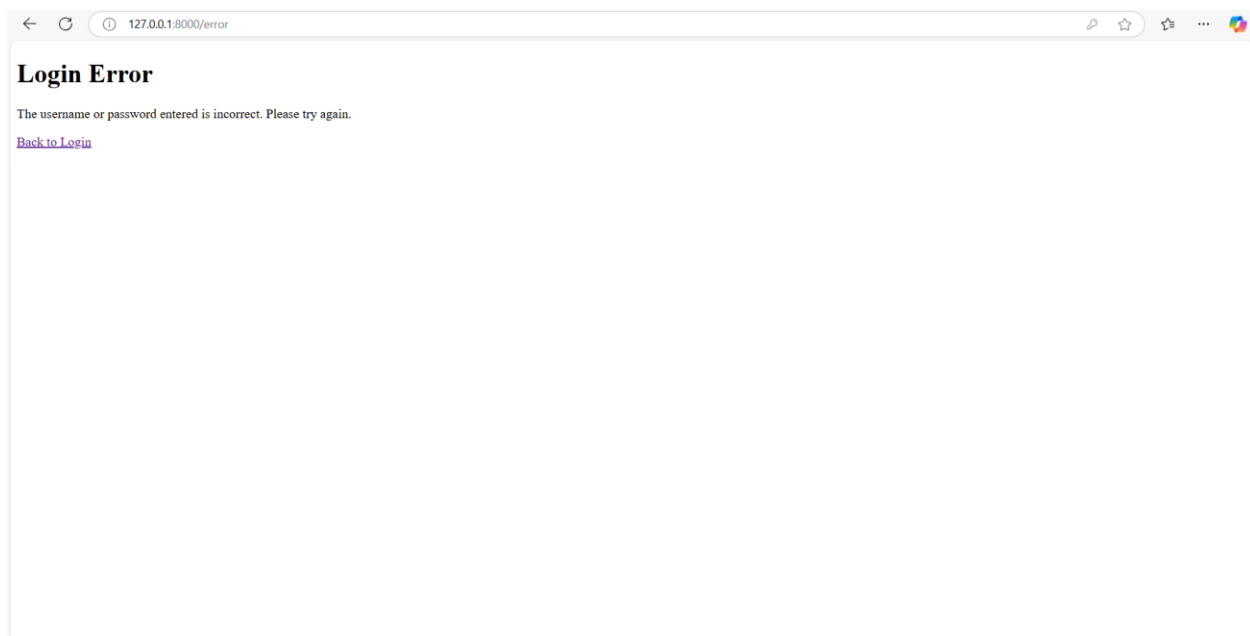
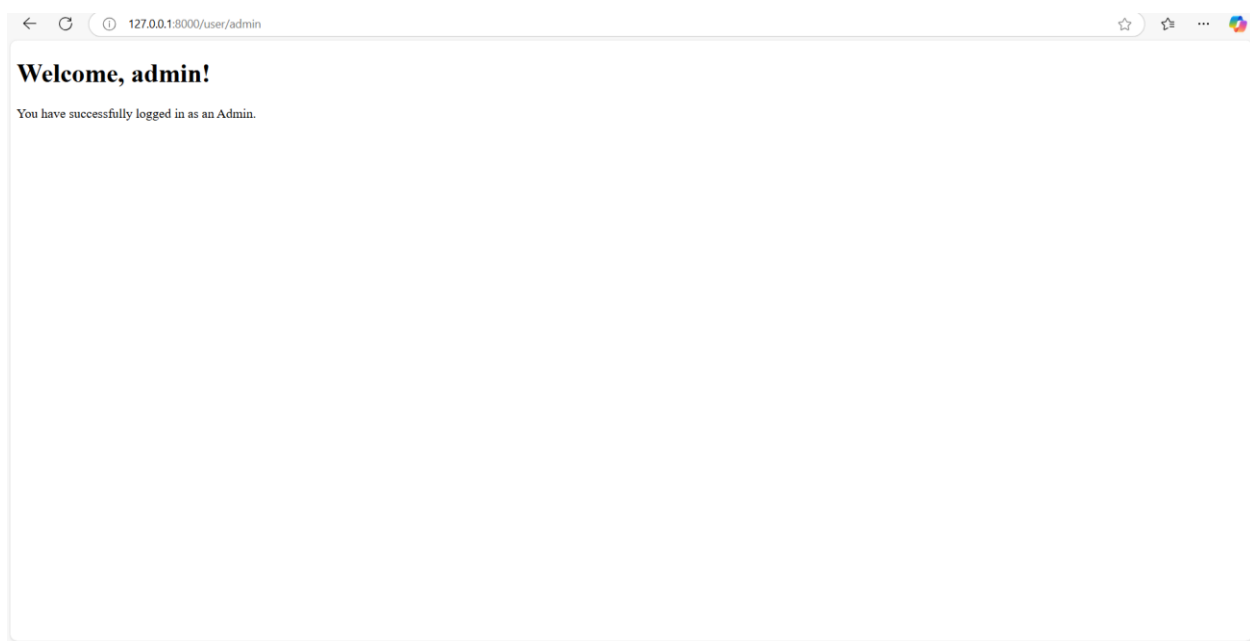
#### Organizer Login

username: organizer / password:organizer123

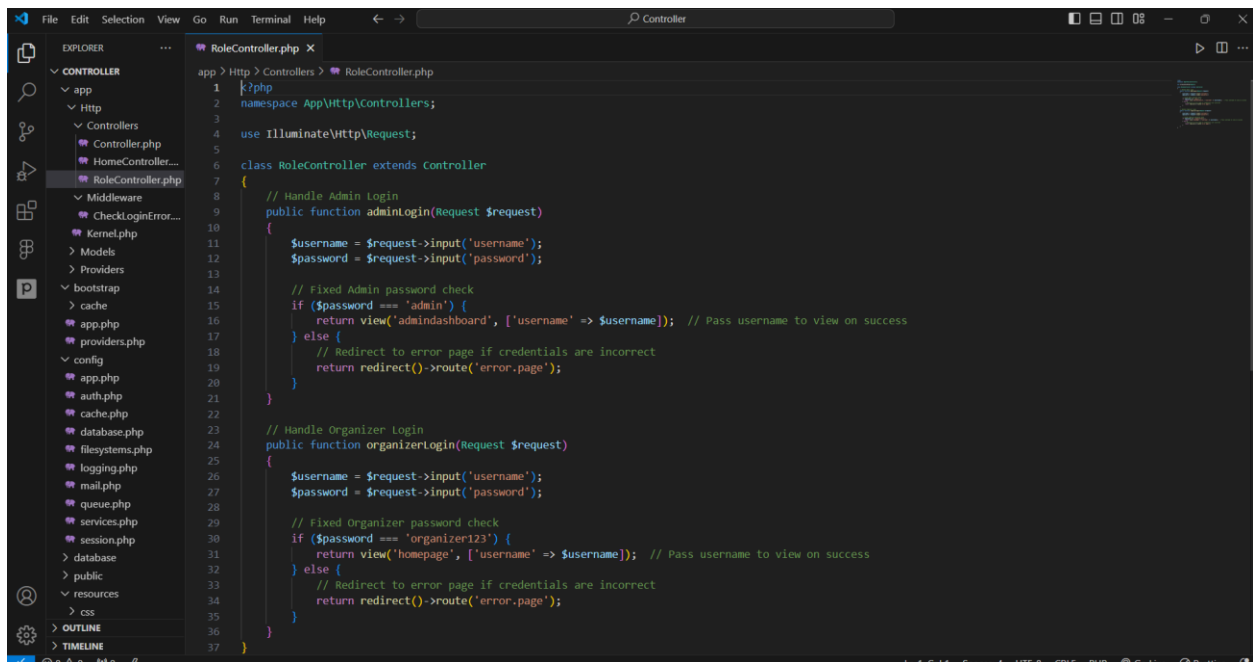
Username

Password

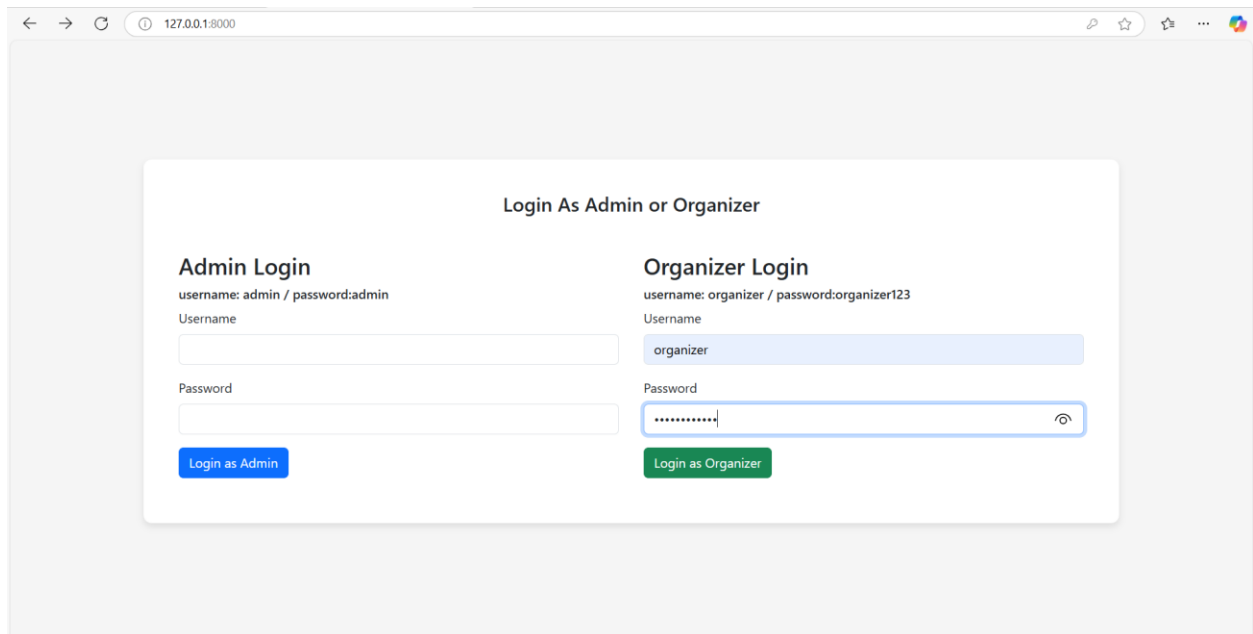
Login as Organizer



The **organizerLogin** method retrieves the **username** and **password** from the request. If the password matches “**organizer123**” it will be redirected to the **homepage** view, with username, but if the password is incorrect, it will be redirected to the error page.



```
1 1?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5
6 class RoleController extends Controller
7 {
8     // Handle Admin Login
9     public function adminLogin(Request $request)
10     {
11         $username = $request->input('username');
12         $password = $request->input('password');
13
14         // Fixed Admin password check
15         if ($password == 'admin') {
16             return view('admindashboard', ['username' => $username]); // Pass username to view on success
17         } else {
18             // Redirect to error page if credentials are incorrect
19             return redirect()->route('error.page');
20         }
21     }
22
23     // Handle Organizer login
24     public function organizerLogin(Request $request)
25     {
26         $username = $request->input('username');
27         $password = $request->input('password');
28
29         // Fixed Organizer password check
30         if ($password == 'organizer123') {
31             return view('homepage', ['username' => $username]); // Pass username to view on success
32         } else {
33             // Redirect to error page if credentials are incorrect
34             return redirect()->route('error.page');
35         }
36     }
37 }
```



127.0.0.1:8000

### Login As Admin or Organizer

#### Admin Login

username: admin / password:admin

Username

Password

Login as Admin

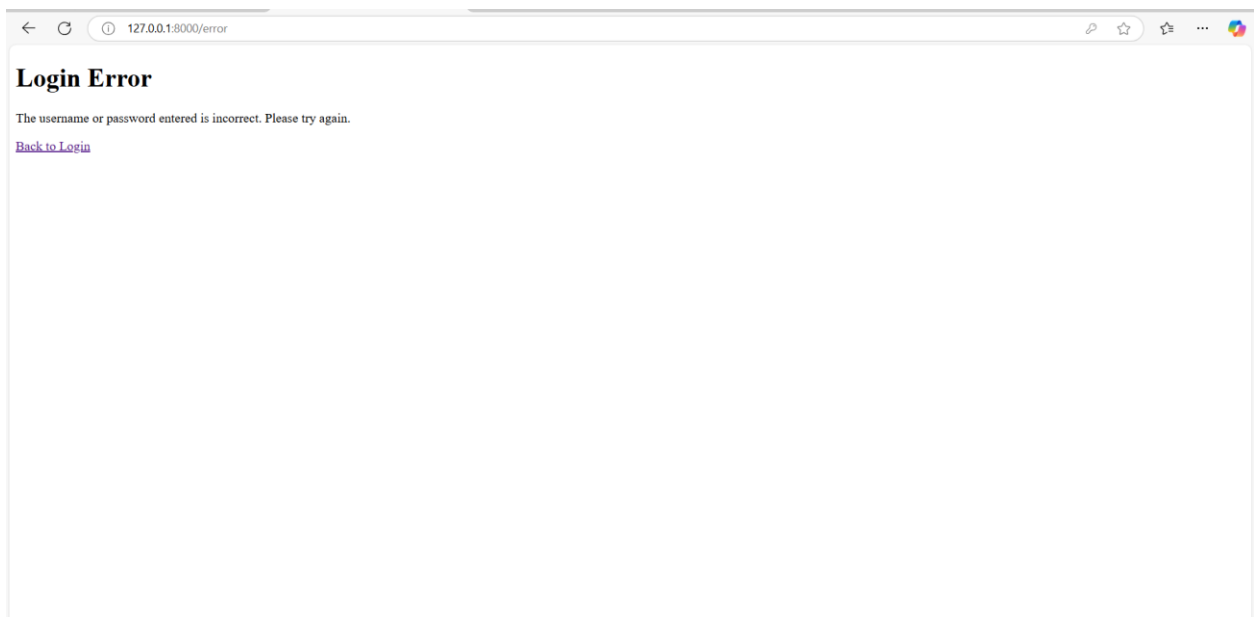
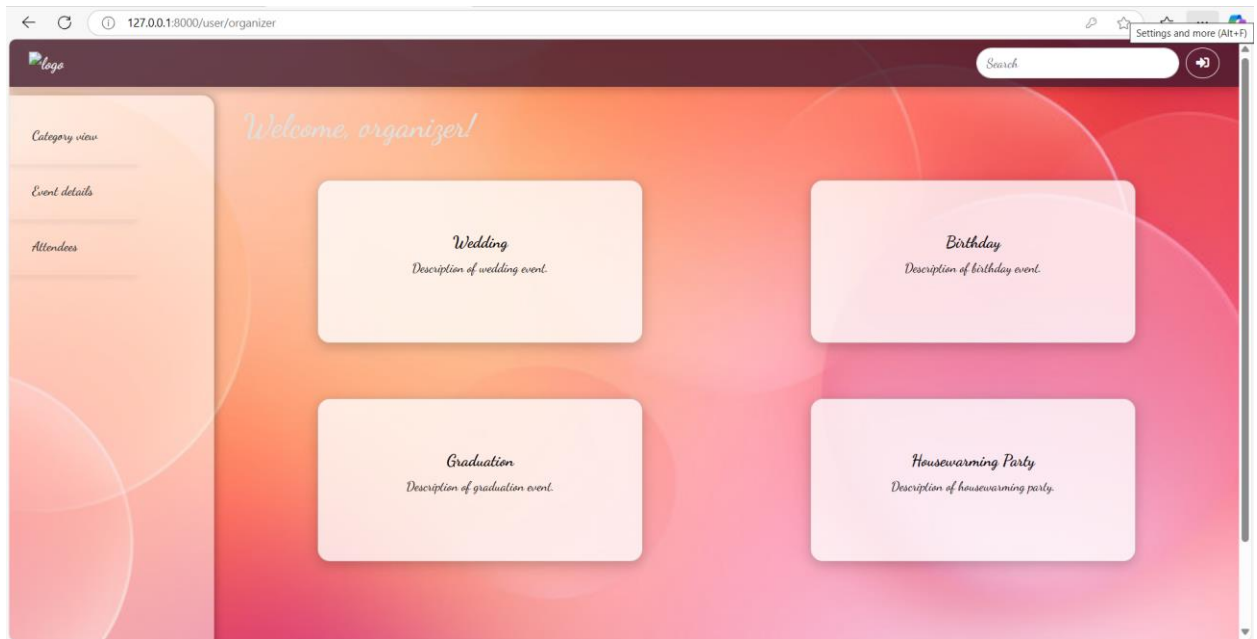
#### Organizer Login

username: organizer / password:organizer123

Username

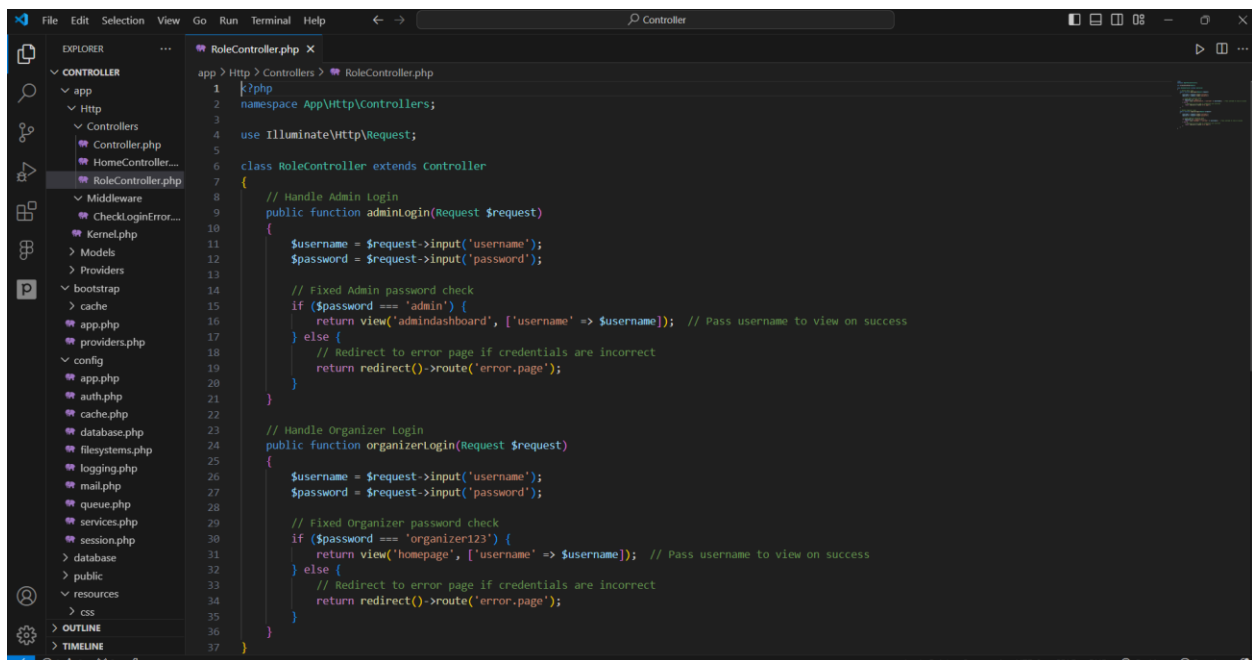
Password

Login as Organizer

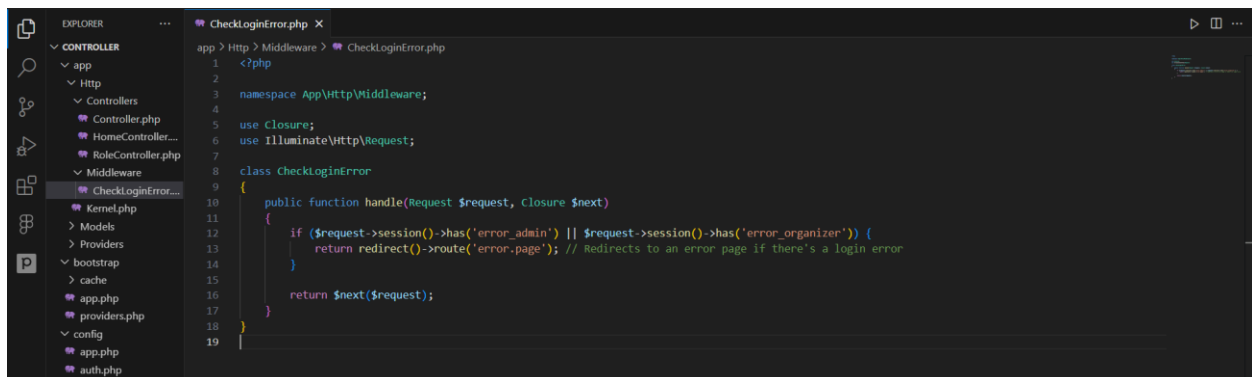


## PARAMETERS

The **RoleController** manages parameters from form submissions, specifically username and password, which are extracted from the Request object for the POST routes **/user/admin** and **/user/organizer**. Within each login method (**adminLogin** and **organizerLogin**), the username and password are retrieved using **\$request->input()**, allowing conditional logic to verify credentials against fixed values. If the password matches, the username is passed to the respective views (**admindashboard** or **homepage**) as an associative array, making the username accessible for display within the view. The **CheckLoginError** middleware also plays a role in parameter handling, focusing on session data to manage access based on login status. It checks for specific session parameters (**error\_admin** or **error\_organizer**) to determine if a login error has occurred. If either of these parameters exists in the session, the middleware redirects the user to the error page; otherwise, it allows the request to proceed.



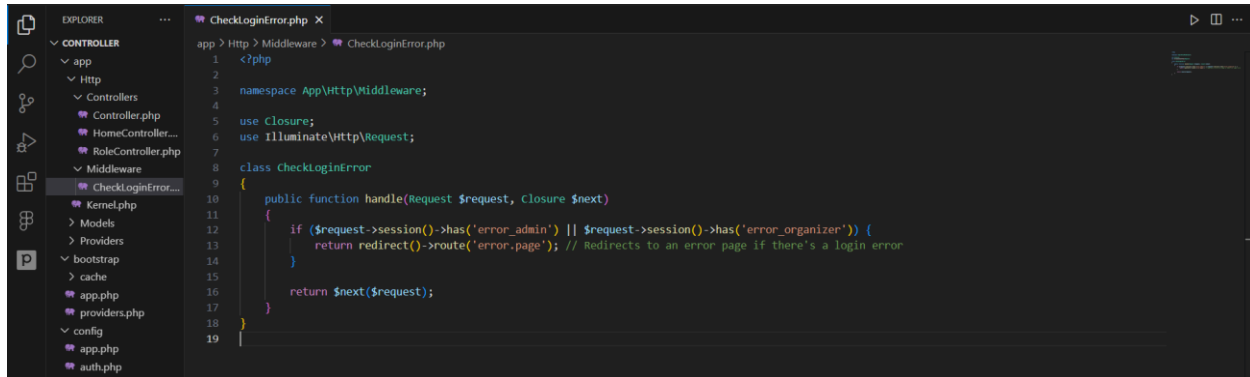
```
1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5
6 class RoleController extends Controller
7 {
8     // Handle Admin Login
9     public function adminLogin(Request $request)
10     {
11         $username = $request->input('username');
12         $password = $request->input('password');
13
14         // Fixed Admin password check
15         if ($password === 'admin') {
16             return view('admindashboard', ['username' => $username]); // Pass username to view on success
17         } else {
18             // Redirect to error page if credentials are incorrect
19             return redirect()->route('error.page');
20         }
21     }
22
23     // Handle Organizer Login
24     public function organizerLogin(Request $request)
25     {
26         $username = $request->input('username');
27         $password = $request->input('password');
28
29         // Fixed Organizer password check
30         if ($password === 'organizer123') {
31             return view('homepage', ['username' => $username]); // Pass username to view on success
32         } else {
33             // Redirect to error page if credentials are incorrect
34             return redirect()->route('error.page');
35         }
36     }
37 }
```



```
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Closure;
6 use Illuminate\Http\Request;
7
8 class CheckLoginError
9 {
10     public function handle(Request $request, Closure $next)
11     {
12         if ($request->session()->has('error_admin') || $request->session()->has('error_organizer')) {
13             return redirect()->route('error.page'); // Redirects to an error page if there's a login error
14         }
15
16         return $next($request);
17     }
18 }
19
```

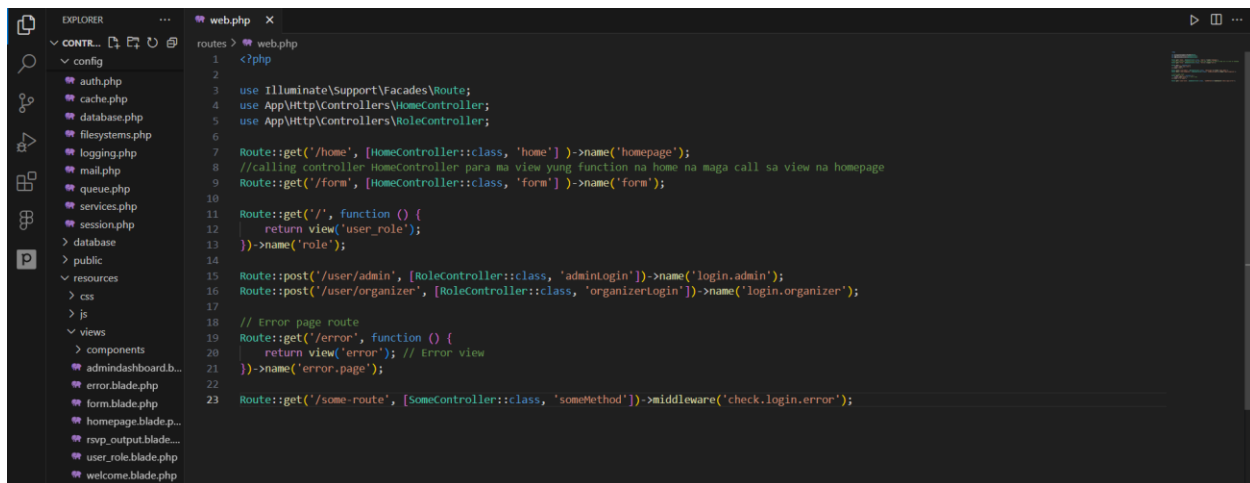
## MIDDLEWARE

The **CheckLoginError** middleware checks if there is an error in the session indicating a failed login attempt for either **admin** or **organizer** role. It looks for specific session keys, `error_admin` or `error_organizer`, which would indicate a login error. If either of these keys exist, the middleware redirects the user to the `error.page` route. If no errors are found, the middleware allows the request to proceed to the next stage.



## ROUTES

The **web.php** file defines the routes for various parts of the application.



The **“/home”** route uses HomeController’s home method, which displays the homepage view while the **“/form”** route also uses HomeController, mapping to a form view.

```
Route::get('/home', [HomeController::class, 'home'] )->name('homepage');  
//calling controller HomeController para ma view yung function na home na maga call sa view na homepage  
Route::get('/form', [HomeController::class, 'form'] )->name('form');
```

The **“/”** route loads the **user\_role** view directly without a controller. The **“/user/admin”** and **“/user/organizer”** routes handle login requests through RoleController's **adminLogin** and **organizerLogin** methods, respectively.

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

Route::post('/user/admin', [RoleController::class, 'adminLogin'])->name('login.admin');
Route::post('/user/organizer', [RoleController::class, 'organizerLogin'])->name('login.organizer');
```

The “/error” route displays an error view if there’s an issue with login. The “/some-route” applies “check.login.error” middleware, which checks the session for login requests through RoleController's adminLogin and organizerLogin methods, respectively

```
// Error page route
Route::get('/error', function () {
    return view('error'); // Error view
})->name('error.page');

Route::get('/some-route', [SomeController::class, 'someMethod'])->middleware('check.login.error');
```