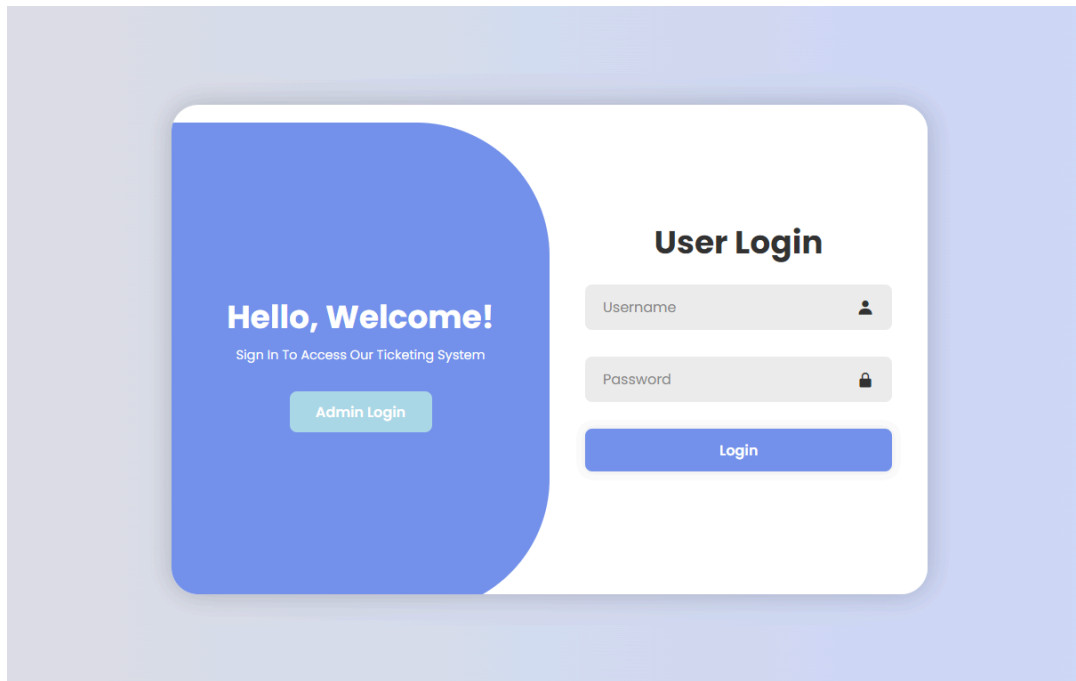
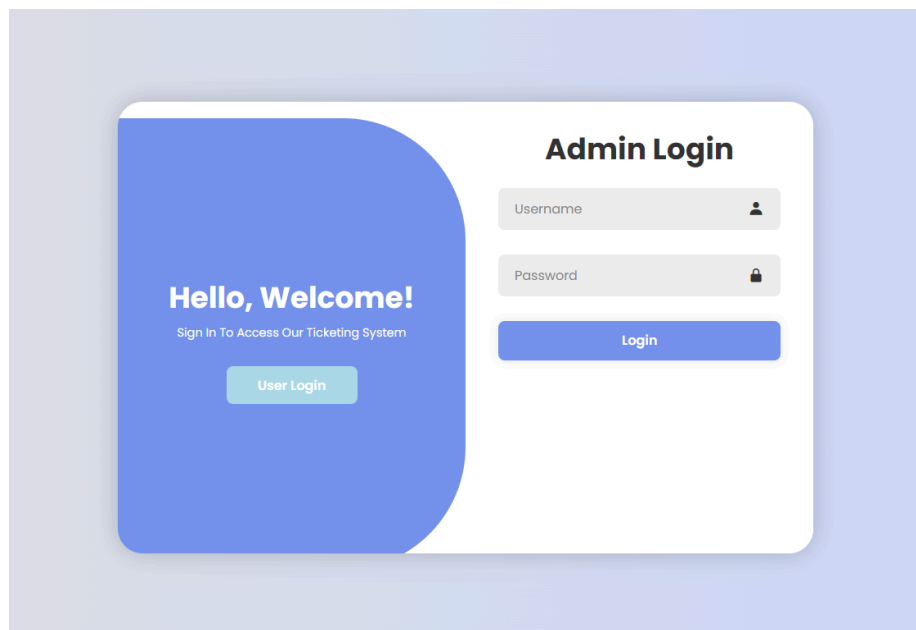


## 1. Log In



- A basic log in UI, where you can login here as a normal user (customer) or as an Admin



## 2. Controllers

### - AuthController.php

```
AuthController.php X
C:\Users\User\Desktop\webdev proj > Controller > app > Http > Controllers > AuthController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5
6  use Illuminate\Http\Request;
7
8  class AuthController extends Controller
9  {
10     // Show the login form
11     public function showLoginForm()
12     {
13         return view('login');
14     }
15
16     // Handle User login
17     public function loginUser(Request $request)
18     {
19         $credentials = $request->only('username', 'password');
20
21         if ($credentials['username'] === 'user' && $credentials['password'] === 'userpassword') {
22             session(['user_type' => 'user']);
23             return redirect()->route('create-ticket');
24         }
25
26         return redirect()->back()->with('error', 'Invalid username or password');
27     }
28
29     // Handle Admin login
30     public function loginAdmin(Request $request)
31     {
32         $credentials = $request->only('username', 'password');
33
34         if ($credentials['username'] === 'admin' && $credentials['password'] === 'adminpassword') {
35             session(['user_type' => 'admin']);
36             return redirect()->route('view-ticket');
37         }
38
39         return redirect()->back()->with('error', 'Invalid username or password');
40     }
41
42     // Handle logout
43     public function logout()
44     {
45         session()->forget('user_type');
46         return redirect()->route('login'); // Corrected line
47     }
48
49 }
```

- Here we used 3 functions, loginUser, loginAdmin, and logout. It shows a simple login function where it will only accept a fixed credential, in this case we used: **For User (customer) user & userpassword, For Admin admin & adminpassword.**

- TicketController.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class TicketController extends Controller
8  {
9      // Show the ticket creation form
10     public function create()
11     {
12         return view('create-ticket'); // Make sure this view exists
13     }
14
15     // Store the ticket - restricted to users
16     public function store(Request $request)
17     {
18         // Logic for storing the ticket goes here
19
20         return redirect()->route('view-ticket')->with('success', 'Ticket created successfully!');
21     }
22
23     // View tickets - restricted to admins
24     public function view()
25     {
26         // Dummy data to simulate tickets
27         $tickets = [
28             [
29                 'id' => 1,
30                 'title' => 'Sample Ticket 1',
31                 'status' => 'open',
32                 'description' => 'Description for sample ticket 1',
33             ],
34             [
35                 'id' => 2,
36                 'title' => 'Sample Ticket 2',
37                 'status' => 'closed',
38                 'description' => 'Description for sample ticket 2',
39             ],
40             [
41                 'id' => 3,
42                 'title' => 'Sample Ticket 3',
43                 'status' => 'in-progress',
44                 'description' => 'Description for sample ticket 3',
45             ],
46         ];
47
48         return view('view-ticket', compact('tickets'));
49     }
50 }
51
```

- Handles the ticket submission and storage
- Restricts the view tickets only to admins
- Shows the ticket creation form to users
- Experimental data can be viewed only to admins

### 3. Middleware

#### checkusertype.php

```
C:\Users\User\Desktop\webdev proj > Controller > app > Http > Middleware > CheckUserType.php
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7
8  class CheckUserType
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param \Illuminate\Http\Request $request
14      * @param \Closure $next
15      * @param string $type
16      * @return mixed
17      */
18     public function handle(Request $request, Closure $next, $type)
19     {
20         if (session('user_type') !== $type) {
21             return redirect()->route('login');
22         }
23         return $next($request);
24     }
25 }
26
```

- Middleware that checks if the authenticated user before proceeding to routes  
checkuserrole.php

```
C:\Users\User\Desktop\webdev proj > Controller > app > Http > Middleware > CheckUserRole.php
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7
8  class CheckUserRole
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param \Illuminate\Http\Request $request
14      * @param \Closure $next
15      * @param string $role
16      * @return mixed
17      */
18     public function handle(Request $request, Closure $next, $role)
19     {
20         // Check if the session has a user type and if it matches the required role
21         if ($request->session()->get('user_type') !== $role) {
22             return redirect()->route('login')->withErrors(['Unauthorized access']);
23         }
24         return $next($request);
25     }
26 }
27
```

## 4. Source codes (CSS Files)

### app.css

```
C:\Users\User> cd Desktop > webdevproj > Controller > public > css > appcss > ...  
1 /* General Styling */  
2 body, html {  
3     margin: 0;  
4     padding: 0;  
5     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
6     color: #333333;  
7     background-color: #f4f7fa;  
8 }  
9  
10  
11 .button {  
12     appearance: none;  
13     background: none;  
14     border: none;  
15     font: inherit;  
16     color: inherit;  
17     padding: 0;  
18     margin: 0;  
19     box-shadow: none;  
20 }  
21  
22 .wrapper {  
23     display: flex;  
24     height: 100vh;  
25 }  
26  
27 /* Sidebar Styling */  
28 .sidebar {  
29     width: 250px;  
30     background-color: #34495e;  
31     color: #fff;  
32     padding-top: 20px;  
33     display: flex;  
34     flex-direction: column;  
35     align-items: center;  
36     box-shadow: 2px 0 5px #c3c6c8(0, 0, 0, 0.1);  
37     position: fixed;  
38     height: 100vh;  
39 }  
40  
41 .sidebar h2 {  
42     font-size: 20px;  
43     margin-bottom: 20px;  
44     text-align: center;  
45     font-weight: bold;  
46 }  
47  
48 .sidebar ul {  
49     list-style-type: none;  
50     padding: 0;  
51     width: 100%;  
52 }  
53  
54 .sidebar ul li {  
55     width: 100%;  
56 }
```

### Login.css

```
C:\Users\User> cd Desktop > webdevproj > Controller > public > css > logincss > ...  
1 @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700;800;900;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');  
2  
3 {  
4     margin: 0;  
5     padding: 0;  
6     box-sizing: border-box;  
7     font-family: "Poppins", sans-serif;  
8     text-decoration: none;  
9     list-style: none;  
10 }  
11  
12  
13 body {  
14     display: flex;  
15     justify-content: center;  
16     align-items: center;  
17     min-height: 100vh;  
18     background: linear-gradient(90deg, #34495e, #34495e);  
19 }  
20  
21 .container {  
22     position: relative;  
23     width: 800px;  
24     height: 500px;  
25     background: #fff;  
26     margin: 20px;  
27     border-radius: 20px;  
28     box-shadow: 0 0 5px #c3c6c8(0, 0, 0, .2);  
29     overflow: hidden;  
30 }  
31  
32 .container h1 {  
33     font-size: 36px;  
34     margin: 10px 0;  
35 }  
36  
37 .container p {  
38     font-size: 14.5px;  
39     margin: 15px 0;  
40 }  
41  
42 form { width: 100%; }  
43  
44 .form-box {  
45     position: absolute;  
46     right: 0;  
47     width: 50%;  
48     height: 100%;  
49     background: #fff;  
50     display: flex;  
51     align-items: center;  
52     color: #333333;  
53     text-align: center;  
54     padding: 40px;  
55     z-index: 1;  
56 }
```

## 5. Source codes (view files)

### create-ticket.blade.php

```
<?php User / User / Desktop / webdev pro / Controller / resources / views / create-ticket.blade.php
1 <!-- resources/views/create-ticket.blade.php -->
2 @extends('layouts.app')
3
4 @section('content')
5 <div class="container">
6 <h1>Create a New Ticket</h1>
7
8 <form action="{{ route('store-ticket') }}" method="POST">
9 @csrf
10
11 <!-- Ticket Title -->
12 <div class="form-group">
13 <label for="title">Ticket Title</label>
14 <input type="text" id="title" name="title" class="form-control" placeholder="Enter the ticket title" required>
15 </div>
16
17 <!-- Ticket Description -->
18 <div class="form-group">
19 <label for="description">Description</label>
20 <textarea id="description" name="description" class="form-control" rows="4" placeholder="Describe the issue" required</textarea>
21 </div>
22
23 <!-- Priority Selection -->
24 <div class="form-group">
25 <label for="priority">Priority</label>
26 <select id="priority" name="priority" class="form-control">
27 <option value="low">Low</option>
28 <option value="medium">Medium</option>
29 <option value="high">High</option>
30 </select>
31 </div>
32
33 <!-- Category Selection -->
34 <div class="form-group">
35 <label for="category">Category</label>
36 <select id="category" name="category" class="form-control">
37 <option value="technical">Technical</option>
38 <option value="billing">Billing</option>
39 <option value="account">Account</option>
40 <option value="other">Other</option>
41 </select>
42 </div>
43
44 <!-- Submit Button -->
45 <button type="submit" class="btn btn-primary">Create Ticket</button>
46 </div>
47 </div>
48 @endsection
49
```

### Login.blade.php

```
<?php User / User / Desktop / webdev pro / Controller / resources / views / login.blade.php
1 <!-- resources/views/login.blade.php -->
2 @extends('layouts.app')
3
4 @section('content')
5 <div class="container">
6 <h1>Login</h1>
7
8 <div class="login-form">
9 <div class="form-group">
10 <input type="text" name="username" placeholder="Username" required>
11 </div>
12 <div class="form-group">
13 <input type="password" name="password" placeholder="Password" required>
14 </div>
15 <button type="submit" class="btn btn-primary">Login</button>
16 </div>
17
18 <div class="admin-login-form">
19 <div class="form-group">
20 <input type="text" name="username" placeholder="Username" required>
21 </div>
22 <div class="form-group">
23 <input type="password" name="password" placeholder="Password" required>
24 </div>
25 <button type="submit" class="btn btn-primary">Admin Login</button>
26 </div>
27 </div>
28 </div>
29 @endsection
30
```

```
<?php User / User / Desktop / webdev pro / Controller / resources / views / login.blade.php
1 <!-- Toggle Button to Switch Between User and Admin Login -->
2 <div class="toggle-box">
3 <div class="toggle-panel toggle-left">
4 <h1>Hello, Welcome!</h1>
5 <p>Sign In To Access Our Ticketing System:</p>
6 <button class="btn register-btn" onclick="toggleLogin()">Admin Login</button>
7 </div>
8 </div>
9
10 <script>
11 // Function to switch between User and Admin login
12 function toggleLogin() {
13 const userLogin = document.getElementById('user-login');
14 const adminLogin = document.getElementById('admin-login');
15 const toggleButton = document.querySelector('.register-btn');
16 const title = document.querySelector('title');
17
18 // Toggle the login forms
19 if (userLogin.style.display === 'none') {
20 userLogin.style.display = 'block';
21 adminLogin.style.display = 'none';
22 title.textContent = 'User Login';
23 toggleButton.textContent = 'Admin Login';
24 } else {
25 userLogin.style.display = 'none';
26 adminLogin.style.display = 'block';
27 title.textContent = 'Admin Login';
28 toggleButton.textContent = 'User Login';
29 }
30 }
31 </script>
32 </div>
33 </div>
34 @endsection
35
```

## view-ticket.php

```
C:\Users\User\Desktop\webdev proj > Controller > resources > views > view-ticket.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <h1>View Tickets</h1>
6         <p>This is the page where you can view all tickets.</p>
7     </div>
8 @endsection
9
```

## App.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ticket System</title>
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
</head>
<body>
    <div class="wrapper">
        <!-- Sidebar -->
        <div class="sidebar">
            <h2>{{ session('user_type') == 'admin' ? 'WELCOME ADMIN' : 'WELCOME USER' }}</h2>
            <ul>
                @if(session('user_type') == 'admin')
                    <li><a href="{{ route('view-ticket') }}">View Ticket</a></li>
                @else
                    <li><a href="{{ route('create-ticket') }}">Create Ticket</a></li>
                    <li><a href="{{ route('view-ticket') }}">View Ticket</a></li>
                @endif
            </ul>
            <form action="{{ route('logout') }}" method="POST">
                @csrf
                <button type="submit" class="logout-btn" style="
                    background-color: #e74c3c !important;
                    color: white !important;
                    width: 100% !important;
                    padding: 12px !important;
                    font-size: 16px !important;
                    text-align: center !important;
                    border: none !important;
                    border-radius: 8px !important;
                    cursor: pointer !important;
                ">Logout</button>
            </form>
        </div>

        <!-- Content Area -->
        <div class="content">
            <!-- Full-Width Search Bar -->
            <div class="search-container">
                <input type="text" placeholder="Search..." class="search-bar">
                <button class="search-button">
                    
                </button>
            </div>

            <!-- Full-Width Form Container -->
            <div class="form-container">
                @yield('content')
            </div>
        </div>
    </div>
</body>
</html>
```

```

1  <?php
2
3  use App\Http\Controllers\AuthController;
4  use App\Http\Controllers\TicketController;
5  use Illuminate\Support\Facades\Route;
6
7  // Make the login page the landing page
8  Route::get('/', function () {
9      return redirect()->route('login');
10 });
11
12 // Route to show the create ticket form - restricted to users with 'user' role
13 Route::get('/create-ticket', [TicketController::class, 'create'])
14     ->name('create-ticket')
15     ->middleware('role:user');
16
17 // Route to handle form submission (store the ticket) - restricted to users with 'user' role
18 Route::post('/store-ticket', [TicketController::class, 'store'])
19     ->name('store-ticket')
20     ->middleware('role:user');
21
22 // Login routes
23 Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');
24
25 // Separate POST routes for user and admin logins
26 Route::post('/login', [AuthController::class, 'loginUser'])->name('login.submit');
27 Route::post('/admin-login', [AuthController::class, 'loginAdmin'])->name('admin.login');
28
29 // Logout route
30 Route::post('/logout', [AuthController::class, 'logout'])->name('logout');
31
32 // Admin-specific route to view tickets - restricted to users with 'admin' role
33 Route::get('/view-ticket', [TicketController::class, 'view'])
34     ->name('view-ticket')
35     ->middleware('role:admin');
36

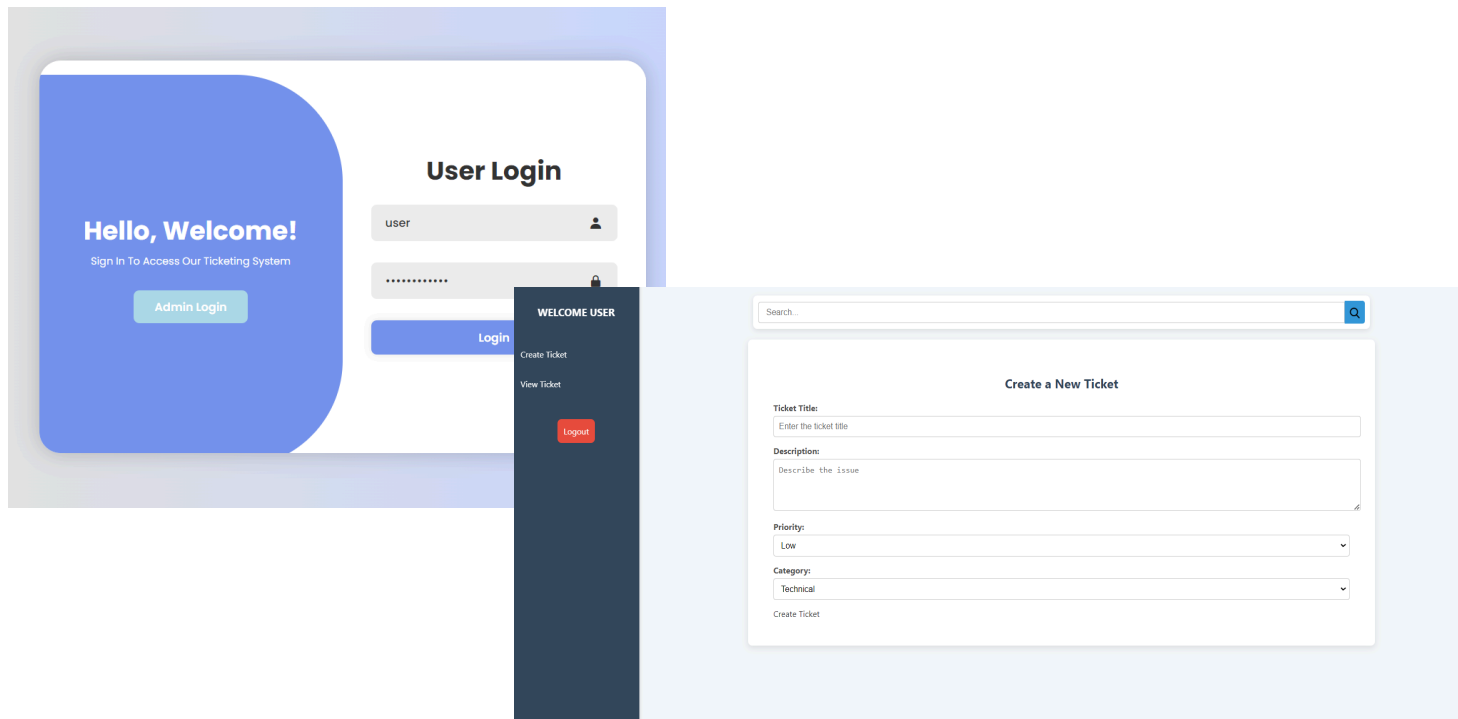
```

- This is the routes, wherein we have 2 login forms on 1 page one of our members suggested and implemented the login for admin and users on 1 page using javascript. And that form functions different to each other, once logging in, it will go through the middleware to check the role type, then go through AuthController.php if you meet the credentials, once it is correct if you are an admin you will proceed to view ticket dashboard, if you are a customer you proceed to fill out form.



## 7. Demo

### User (Customer) Login & Dashboard



### Admin Login & Dashboard

