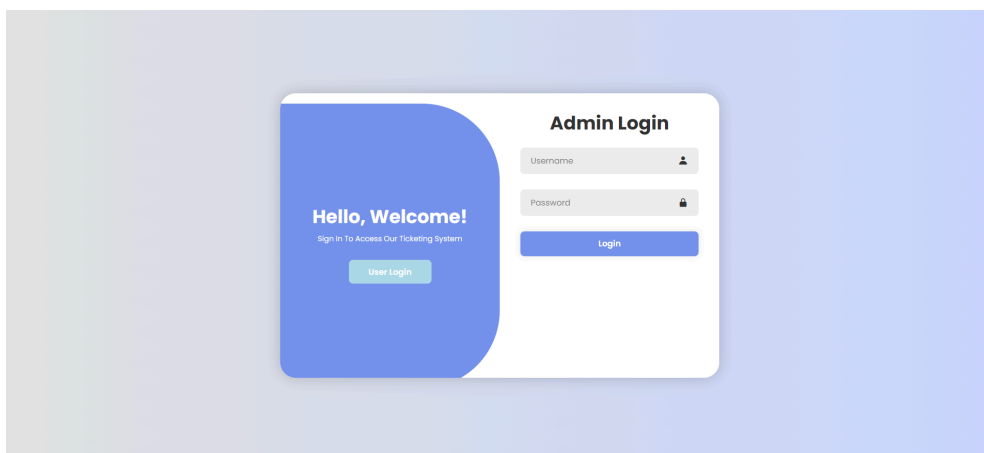
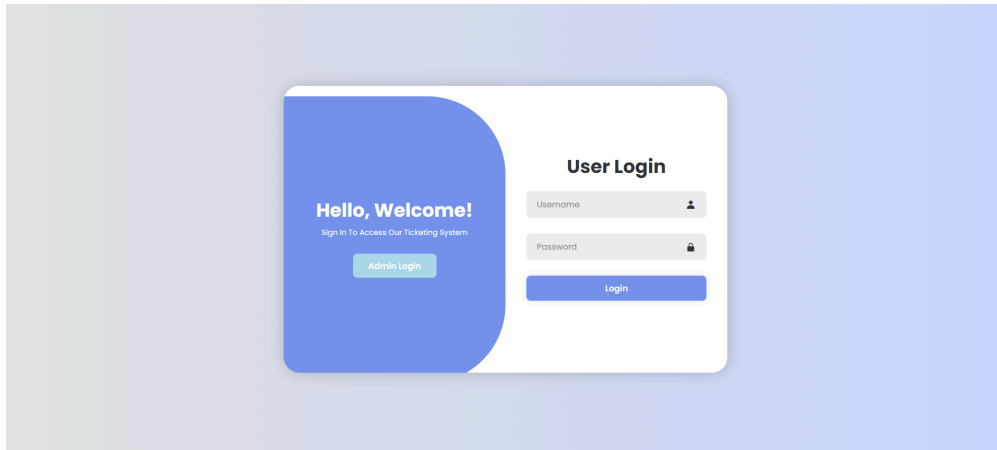


Controller Documentation

1. Login for customer as user and admin



A simple login user interface

Source codes for login.blade.php

```
<DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Login</title>
    <link rel="stylesheet" href="{{ asset('css/login.css') }}">
    <link href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <!-- User Login Form -->
        <div class="form-box login" id="user-login">
            <form action="{{ route('login') }}" method="POST">
                @csrf
                <h1>User Login</h1>
                <div class="input-box">
                    <input type="text" name="username" placeholder="Username" required>
                    <i class="bx bx-user"></i>
                </div>
                <div class="input-box">
                    <input type="password" name="password" placeholder="Password" required>
                    <i class="bx bx-lock-alt"></i>
                </div>
                <button type="submit" class="btn">Login</button>
            </form>
        </div>

        <!-- Admin Login Form (Initially Hidden) -->
        <div class="form-box login" id="admin-login" style="display: none;">
            <form action="{{ route('admin.login') }}" method="POST">
                @csrf
                <h1>Admin Login</h1>
                <div class="input-box">
                    <input type="text" name="username" placeholder="Username" required>
                    <i class="bx bx-user"></i>
                </div>
                <div class="input-box">
                    <input type="password" name="password" placeholder="Password" required>
                    <i class="bx bx-lock-alt"></i>
                </div>
                <button type="submit" class="btn">Login</button>
            </form>
        </div>
    </div>

    <!-- Toggle Button to Switch Between User and Admin Login -->
    <div class="toggle-box">
        <div class="toggle-panel toggle-left">
            <h1>Hello, Welcome</h1>
            <p>Sign In To Access Our Ticketing System</p>
            <button class="btn register-btn" onclick="toggleLogin()">Admin Login</button>
        </div>
    </div>

    <script>
        // Function to switch between User and Admin login
        function toggleLogin() {
            const userLogin = document.getElementById('user-login');
            const adminLogin = document.getElementById('admin-login');
            const toggleButton = document.querySelector('.register-btn');
            const title = document.querySelector('title');

            // Toggle the login forms
            if (userLogin.style.display === 'none') {
                userLogin.style.display = 'block';
                adminLogin.style.display = 'none';
                title.textContent = 'User Login';
                toggleButton.textContent = 'Admin Login';
            } else {
                userLogin.style.display = 'none';
                adminLogin.style.display = 'block';
                title.textContent = 'Admin Login';
                toggleButton.textContent = 'User Login';
            }
        }
    </script>
</body>
</html>
```

- The source code integrated both user and admin login on one page using javascript

2. User dashboard (customer)

WELCOME USER

Create Ticket

View Ticket

Logout

Search...

Create a New Ticket

Ticket Title:
Enter the ticket title

Description:
Describe the issue

Priority:
Low

Category:
Technical

Create Ticket

Source code for 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     </form>
47 </div>
48 @endsection

```

3. Admin dashboard

Source code for view-ticke.bladet.php:

```
1 @extends('layouts.app')
2
3 @section('content')
4
5     <div class="container">
6         <h1>View Tickets</h1>
7         <p>This is the page where you can view all tickets.</p>
8         <!-- resources/views/view-ticket.blade.php -->
9
10    <h1>Admin - View Tickets</h1>
11
12    @foreach ($tickets as $ticket)
13        <div>
14            <h3>Ticket #{{ $ticket['id'] }}: {{ $ticket['title'] }}</h3>
15            <p>Status: {{ $ticket['status'] }}</p>
16            <p>Description: {{ $ticket['description'] }}</p>
17            <hr>
18        </div>
19    @endforeach
20
21    </div>
22 @endsection
23
```

4. Layout file (app.blade.php)

Source code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Ticket System</title>
7     <link rel="stylesheet" href="{{ asset('css/app.css') }}">
8 </head>
9 <body>
10     <div class="wrapper">
11         <!-- Sidebar -->
12         <div class="sidebar">
13             <h2>{{ session('user_type') == 'admin' ? 'WELCOME ADMIN' : 'WELCOME USER' }}</h2>
14             <ul>
15                 @if(session('user_type') == 'admin')
16                     <li><a href="{{ route('view-ticket') }}">View Ticket</a></li>
17                 @else
18                     <li><a href="{{ route('create-ticket') }}">Create Ticket</a></li>
19                     <li><a href="{{ route('view-ticket') }}">View Ticket</a></li>
20                 @endif
21             </ul>
22             <form action="{{ route('logout') }}" method="POST">
23                 @csrf
24                 <button type="submit" class="logout-btn" style="
25                     background-color: #e74c3c !important;
26                     color: white !important;
27                     width: 100% !important;
28                     padding: 12px !important;
29                     font-size: 16px !important;
30                     text-align: center !important;
31                     border: none !important;
32                     border-radius: 8px !important;
33                     cursor: pointer !important;
34                 ">Logout</button>
35             </form>
36         </div>
37
38         <!-- Content Area -->
39         <div class="content">
40             <!-- Full-Width Search Bar -->
41             <div class="search-container">
42                 <input type="text" placeholder="Search..." class="search-bar">
43                 <button class="search-button">
44                     
45                 </button>
46             </div>
47
48             <!-- Full-Width Form Container -->
49             <div class="form-container">
50                 @yield('content')
51             </div>
52         </div>
53     </div>
54 </body>
55 </html>
```

5. Routes (web.php)

```
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');
```

- Here we can see that before logging in, it must pass into the middleware only allowing the user role if logged in as a customer
- Only allows admin role to be logged into the admin dashboard

6. Controllers

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

Authcontroller.php

- Handles the login credentials, where the credentials are fixed as of now. For admin the login credentials are admin & adminpassword : For user or the customer the login credentials are user & userpassword.

Ticketcontroller.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class TicketController extends Controller
{
    // Show the ticket creation form
    public function create()
    {
        return view('create-ticket'); // Make sure this view exists
    }

    // Store the ticket - restricted to users
    public function store(Request $request)
    {
        // Logic for storing the ticket goes here

        return redirect()->route('view-ticket')->with('success', 'Ticket created successfully!');
    }

    // View tickets - restricted to admins
    public function view()
    {
        // Dummy data to simulate tickets
        $tickets = [
            [
                'id' => 1,
                'title' => 'Sample Ticket 1',
                'status' => 'open',
                'description' => 'Description for sample ticket 1',
            ],
            [
                'id' => 2,
                'title' => 'Sample Ticket 2',
                'status' => 'closed',
                'description' => 'Description for sample ticket 2',
            ],
            [
                'id' => 3,
                'title' => 'Sample Ticket 3',
                'status' => 'in-progress',
                'description' => 'Description for sample ticket 3',
            ],
        ];

        return view('view-ticket', compact('tickets'));
    }
}
```

- Handles ticket storage and dummy data

7. Middleware

checkuserole.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
25         return $next($request);
26     }
27 }
28
```

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
```