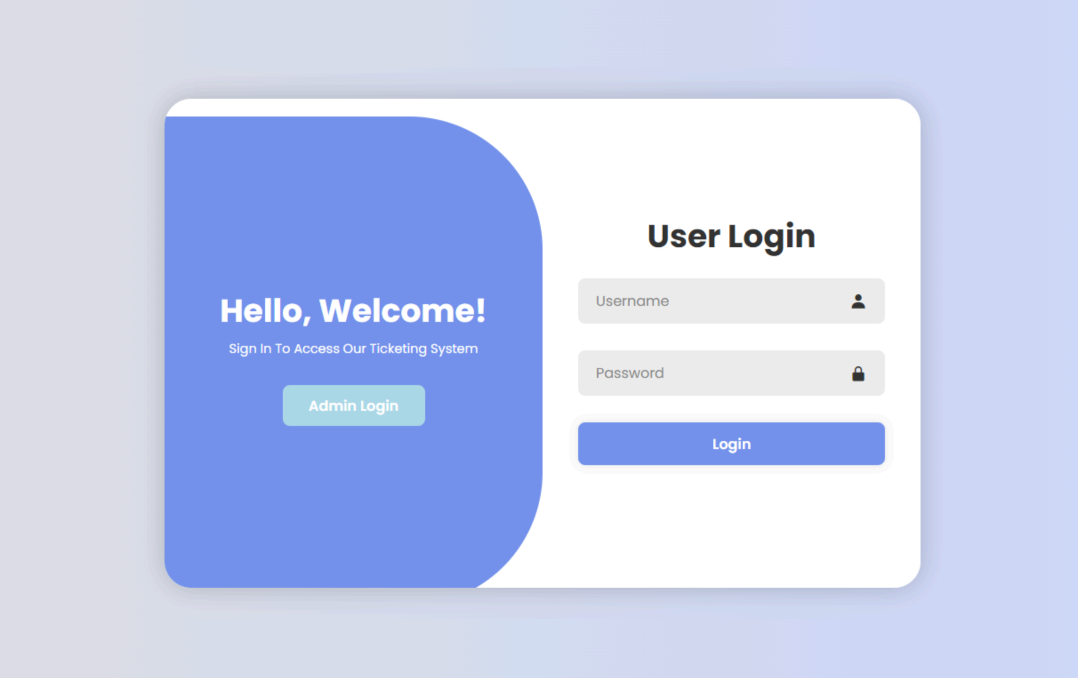
Andreev Albert D. Ataiza WebDev BSIT-3C

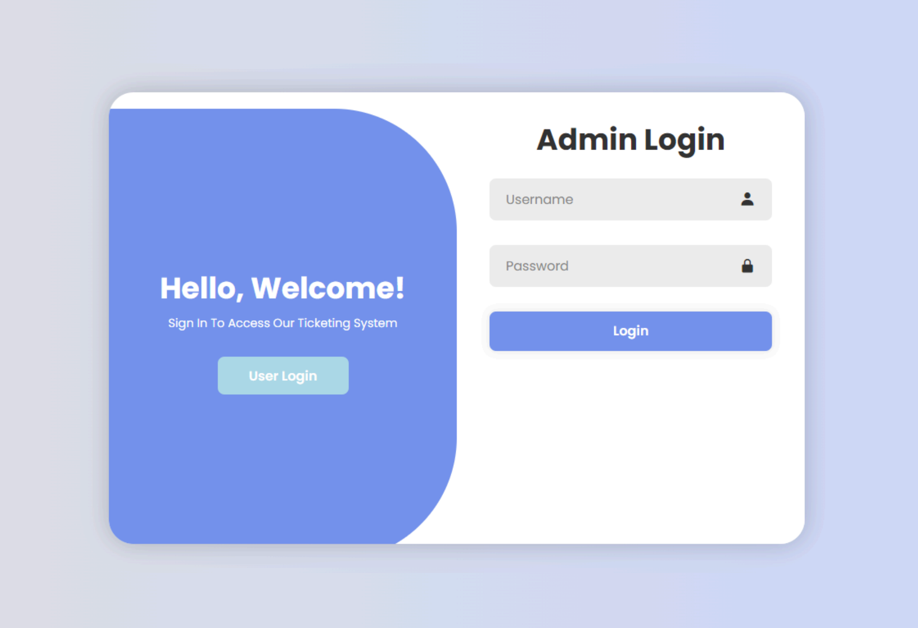
1. Log In


User Login

Hello, Welcome!

Sign In To Access Our Ticketing System

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

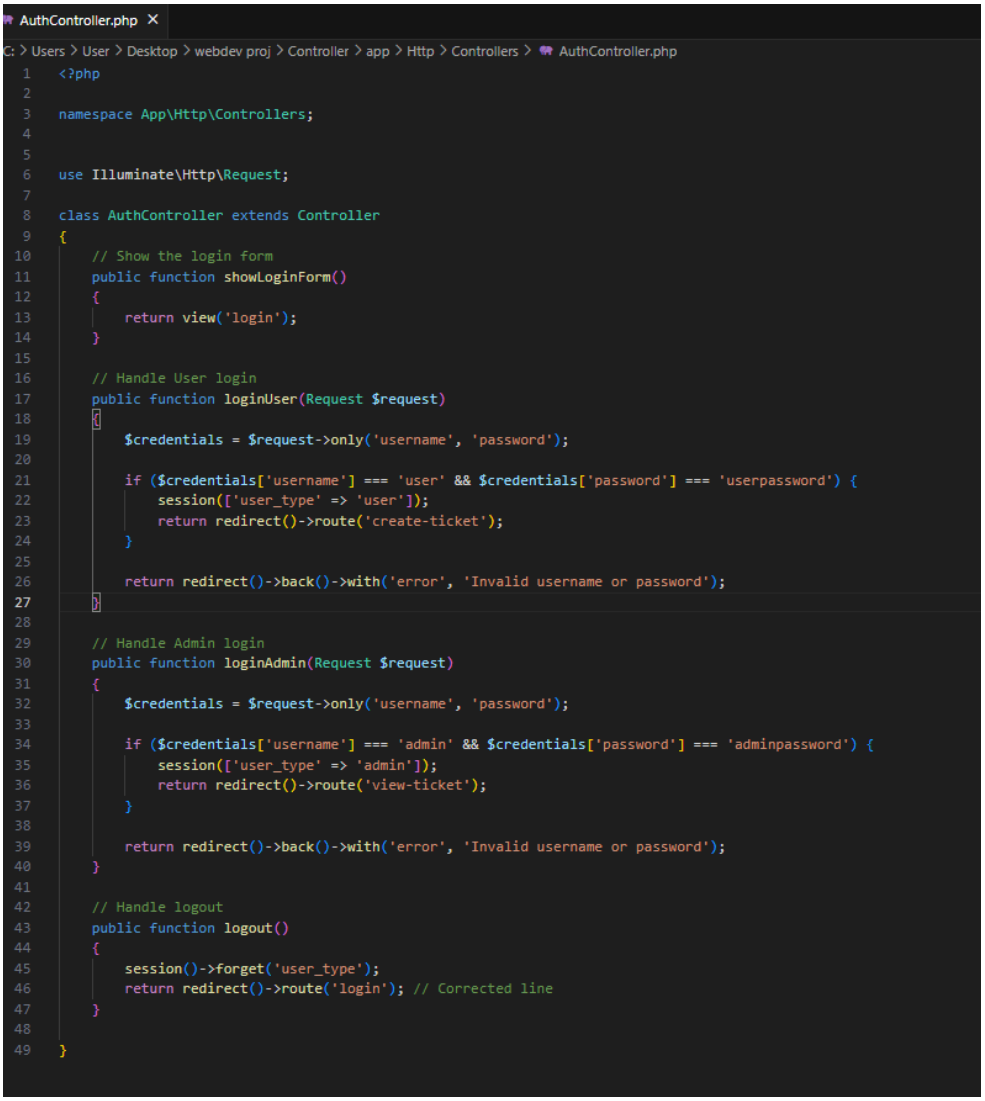

Admin Login

Hello, Welcome!

Sign In To Access Our Ticketing System

2. Controllers

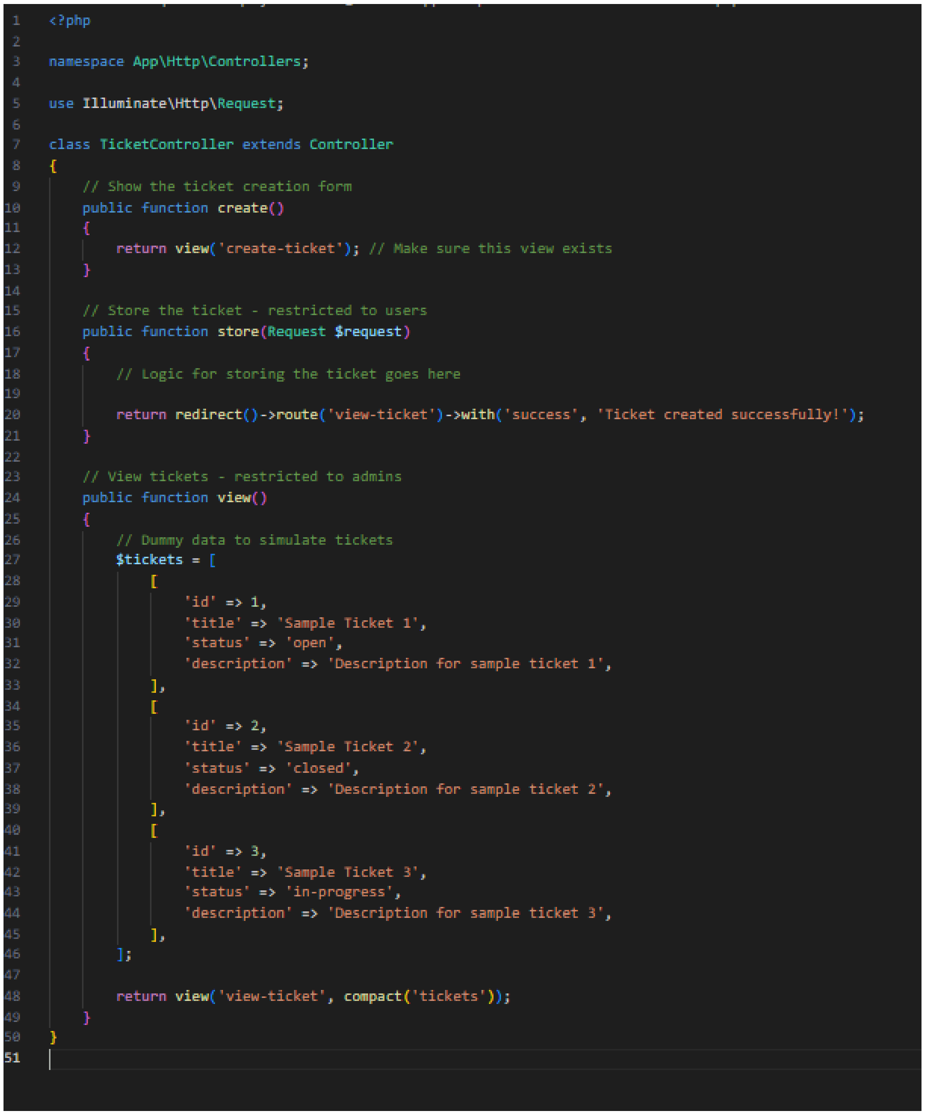
    - AuthController.php


AuthController.php ×
C: > Users > User > Desktop > webdev proj > Controller > app > Http > Controllers > AuthController.php

```php
<?php

namespace App\Http\Controllers;


use Illuminate\Http\Request;

class AuthController extends Controller
{
    // Show the login form
    public function showLoginForm()
    {
        return view('login');
    }

    // Handle User login
    public function loginUser(Request $request)
    {
        $credentials = $request->only('username', 'password');

        if ($credentials['username'] === 'user' && $credentials['password'] === 'userpassword') {
            session(['user_type' => 'user']);
            return redirect()->route('create-ticket');
        }

        return redirect()->back()->with('error', 'Invalid username or password');
    }

    // Handle Admin login
    public function loginAdmin(Request $request)
    {
        $credentials = $request->only('username', 'password');

        if ($credentials['username'] === 'admin' && $credentials['password'] === 'adminpassword') {
            session(['user_type' => 'admin']);
            return redirect()->route('view-ticket');
        }

        return redirect()->back()->with('error', 'Invalid username or password');
    }

    // Handle logout
    public function logout()
    {
        session()->forget('user_type');
        return redirect()->route('login'); // Corrected line
    }

}
```

    - 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

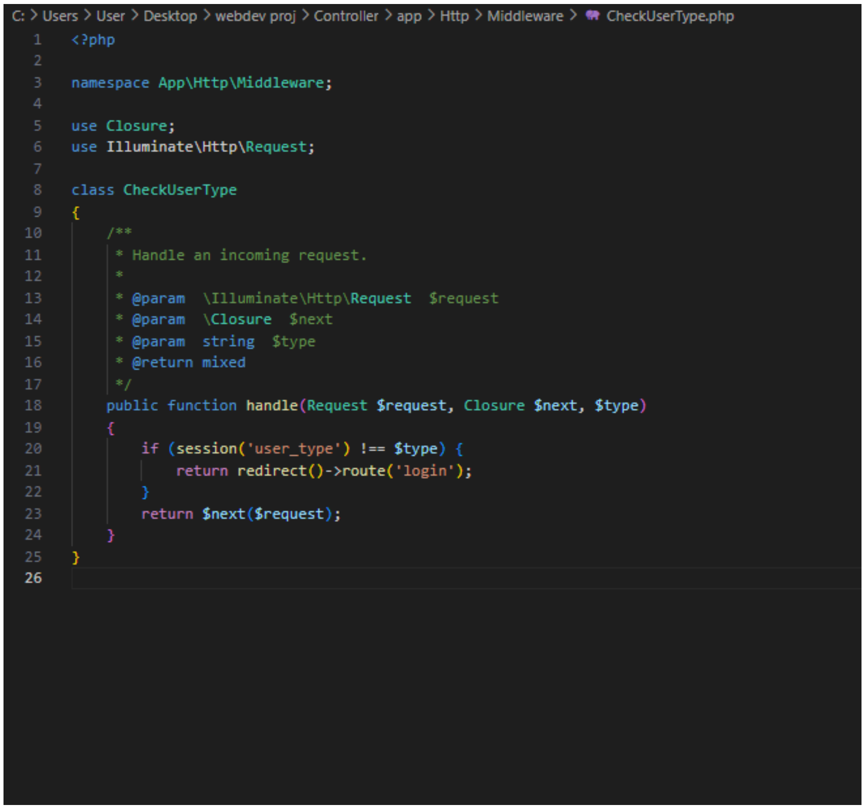


```php
<?php

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 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

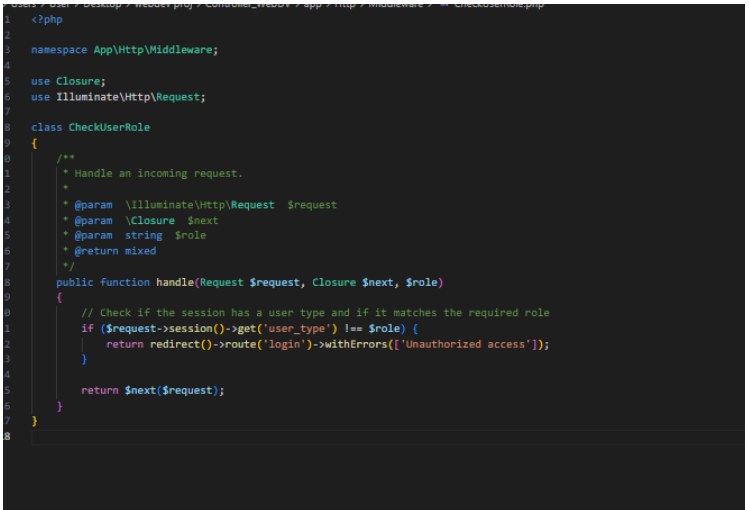
checkusertype.php



C: > Users > User > Desktop > webdev proj > Controller > app > Http > Middleware > CheckUserType.php

```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class CheckUserType
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @param  string  $type
     * @return mixed
     */
    public function handle(Request $request, Closure $next, $type)
    {
        if (session('user_type') !== $type) {
            return redirect()->route('login');
        }
        return $next($request);
    }
}
```

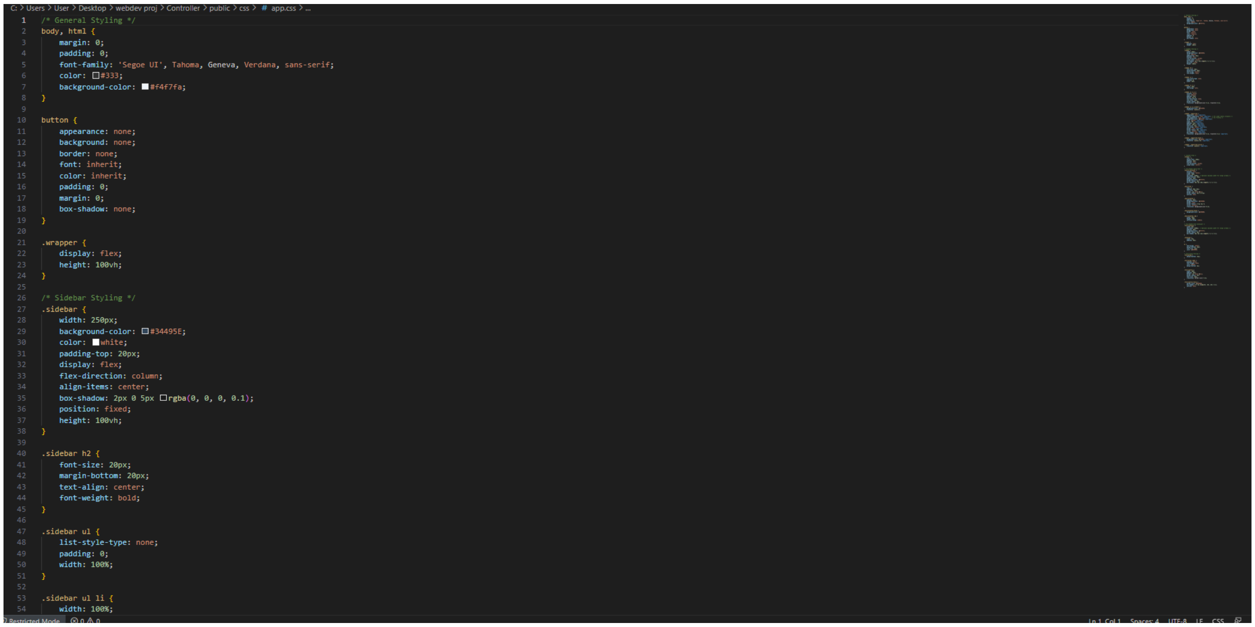
- Middleware that checks if the authenticated user before proceeding to routes
  checkuserrole.php



```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;

class CheckUserRole
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @param  string  $role
     * @return mixed
     */
    public function handle(Request $request, Closure $next, $role)
    {
        // Check if the session has a user type and if it matches the required role
        if ($request->session()->get('user_type') !== $role) {
            return redirect()->route('login')->withErrors(['Unauthorized access']);
        }

        return $next($request);
    }
}
```

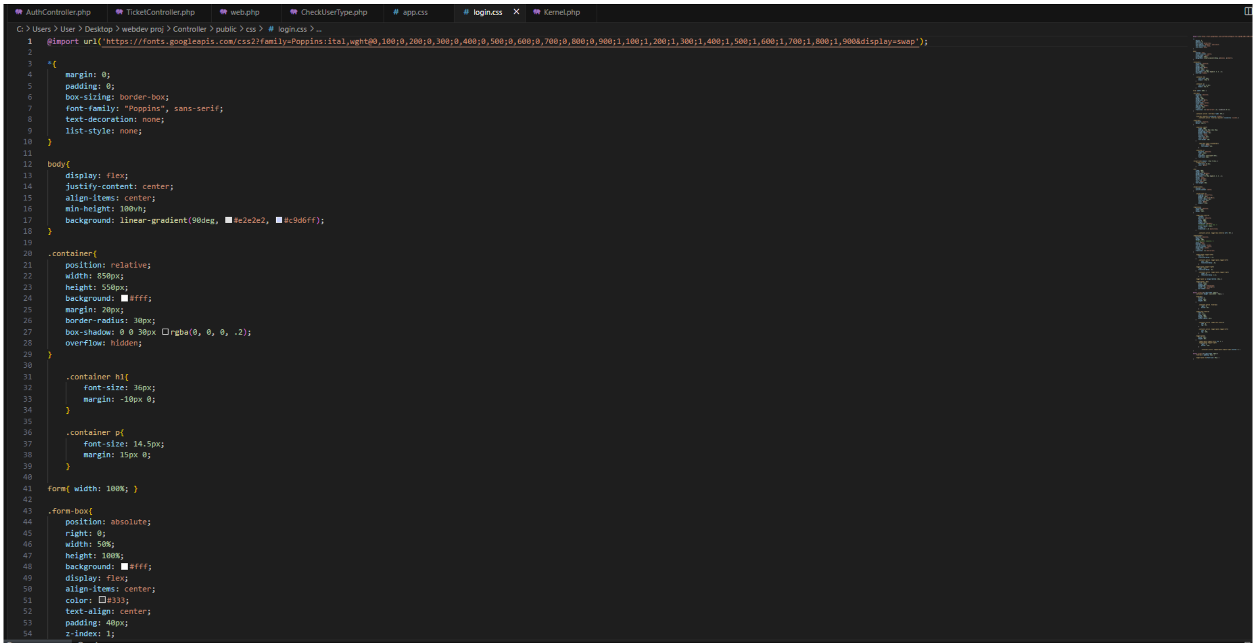
4. Source codes (CSS Files)

app.css


C: > Users > User > Desktop > webdev proj > Controller > public > css > # app.css > ...
Restricted Mode
⊗ 0 ⚠ 0
Ln 1, Col 1
Spaces: 4
UTF-8
LF
CSS

```css
/* General Styling */
body, html {
    margin: 0;
    padding: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    color: #333;
    background-color: #f4f7fa;
}

button {
    appearance: none;
    background: none;
    border: none;
    font: inherit;
    color: inherit;
    padding: 0;
    margin: 0;
    box-shadow: none;
}

.wrapper {
    display: flex;
    height: 100vh;
}

/* Sidebar Styling */
.sidebar {
    width: 250px;
    background-color: #34495E;
    color: white;
    padding-top: 20px;
    display: flex;
    flex-direction: column;
    align-items: center;
    box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
    position: fixed;
    height: 100vh;
}

.sidebar h2 {
    font-size: 20px;
    margin-bottom: 20px;
    text-align: center;
    font-weight: bold;
}

.sidebar ul {
    list-style-type: none;
    padding: 0;
    width: 100%;
}

.sidebar ul li {
    width: 100%;
```

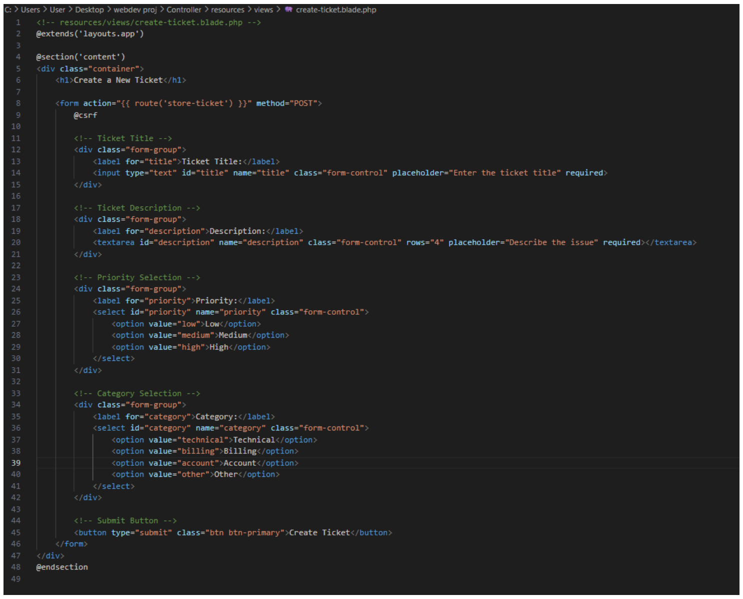
Login.css



```css
@import url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: "Poppins", sans-serif;
    text-decoration: none;
    list-style: none;
}

body{
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background: linear-gradient(90deg, #e2e2e2, #c9d6ff);
}

.container{
    position: relative;
    width: 850px;
    height: 550px;
    background: #fff;
    margin: 20px;
    border-radius: 30px;
    box-shadow: 0 0 30px rgba(0, 0, 0, .2);
    overflow: hidden;
}

    .container h1{
        font-size: 36px;
        margin: -10px 0;
    }

    .container p{
        font-size: 14.5px;
        margin: 15px 0;
    }

form{ width: 100%; }

.form-box{
    position: absolute;
    right: 0;
    width: 50%;
    height: 100%;
    background: #fff;
    display: flex;
    align-items: center;
    color: #333;
    text-align: center;
    padding: 40px;
    z-index: 1;
```

5. Source codes (view files)

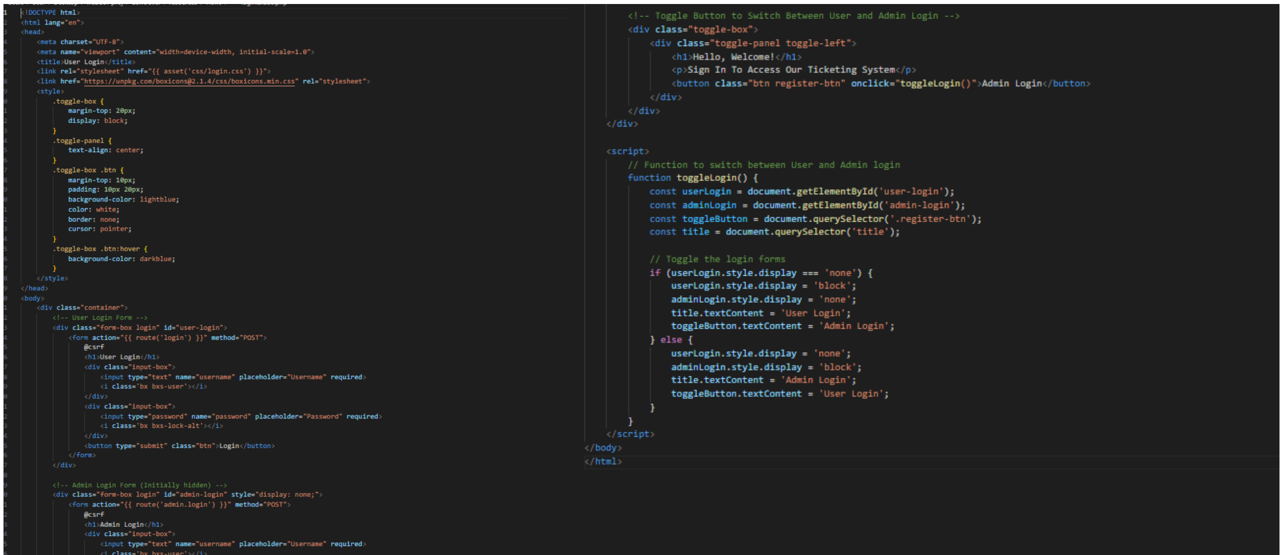
create-ticket.blade.php



C: > Users > User > Desktop > webdev proj > Controller > resources > views > create-ticket.blade.php

```
<!-- resources/views/create-ticket.blade.php -->
@extends('layouts.app')

@section('content')
<div class="container">
    <h1>Create a New Ticket</h1>

    <form action="{{ route('store-ticket') }}" method="POST">
        @csrf

        <!-- Ticket Title -->
        <div class="form-group">
            <label for="title">Ticket Title:</label>
            <input type="text" id="title" name="title" class="form-control" placeholder="Enter the ticket title" required>
        </div>

        <!-- Ticket Description -->
        <div class="form-group">
            <label for="description">Description:</label>
            <textarea id="description" name="description" class="form-control" rows="4" placeholder="Describe the issue" required></textarea>
        </div>

        <!-- Priority Selection -->
        <div class="form-group">
            <label for="priority">Priority:</label>
            <select id="priority" name="priority" class="form-control">
                <option value="low">Low</option>
                <option value="medium">Medium</option>
                <option value="high">High</option>
            </select>
        </div>

        <!-- Category Selection -->
        <div class="form-group">
            <label for="category">Category:</label>
            <select id="category" name="category" class="form-control">
                <option value="technical">Technical</option>
                <option value="billing">Billing</option>
                <option value="account">Account</option>
                <option value="other">Other</option>
            </select>
        </div>

        <!-- Submit Button -->
        <button type="submit" class="btn btn-primary">Create Ticket</button>
    </form>
</div>
@endsection
```

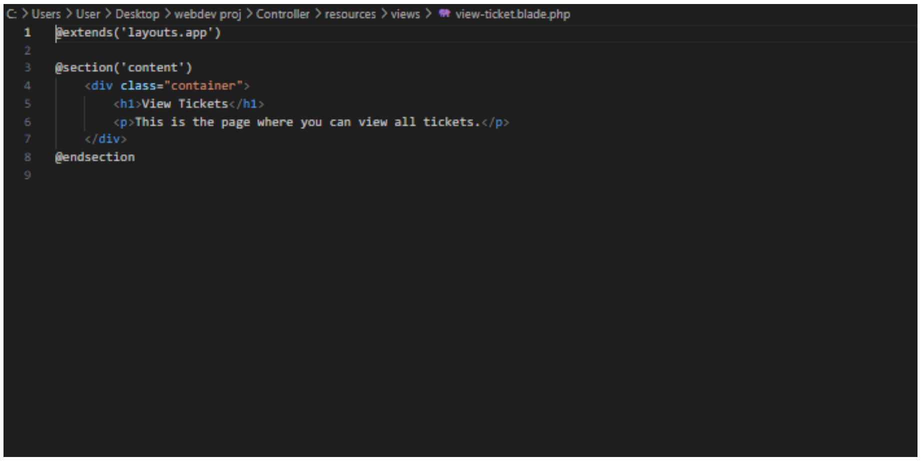
Login.blade.php



```
<!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">
    <style>
        .toggle-box {
            margin-top: 20px;
            display: block;
        }
        .toggle-panel {
            text-align: center;
        }
        .toggle-box .btn {
            margin-top: 10px;
            padding: 10px 20px;
            background-color: lightblue;
            color: white;
            border: none;
            cursor: pointer;
        }
        .toggle-box .btn:hover {
            background-color: darkblue;
        }
    </style>
</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 bxs-user'></i>
                </div>
                <div class="input-box">
                    <input type="password" name="password" placeholder="Password" required>
                    <i class='bx bxs-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 bxs-user'></i>
```

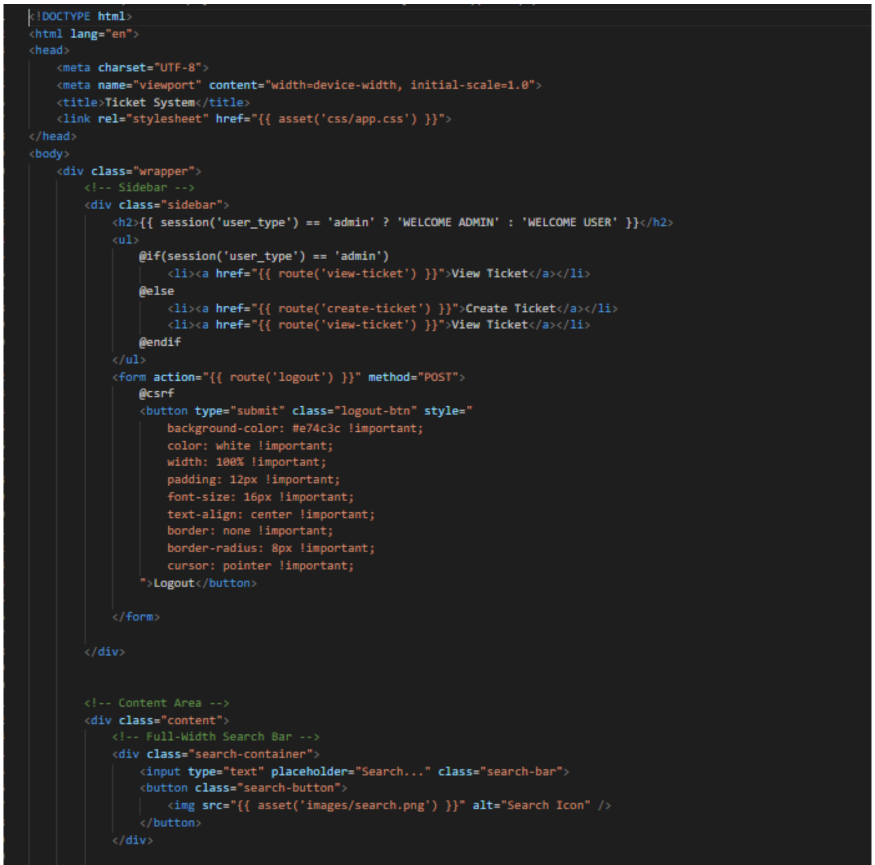
```
        <!-- Toggle Button to Switch Between User and Admin Login -->
        <div class="toggle-box">
            <div class="toggle-panel toggle-left">
                <h1>Hello, Welcome!</h1>
                <p>Sign In To Access Our Ticketing System</p>
                <button class="btn register-btn"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

view-ticket.php


C: > Users > User > Desktop > webdev proj > Controller > resources > views > view-ticket.blade.php

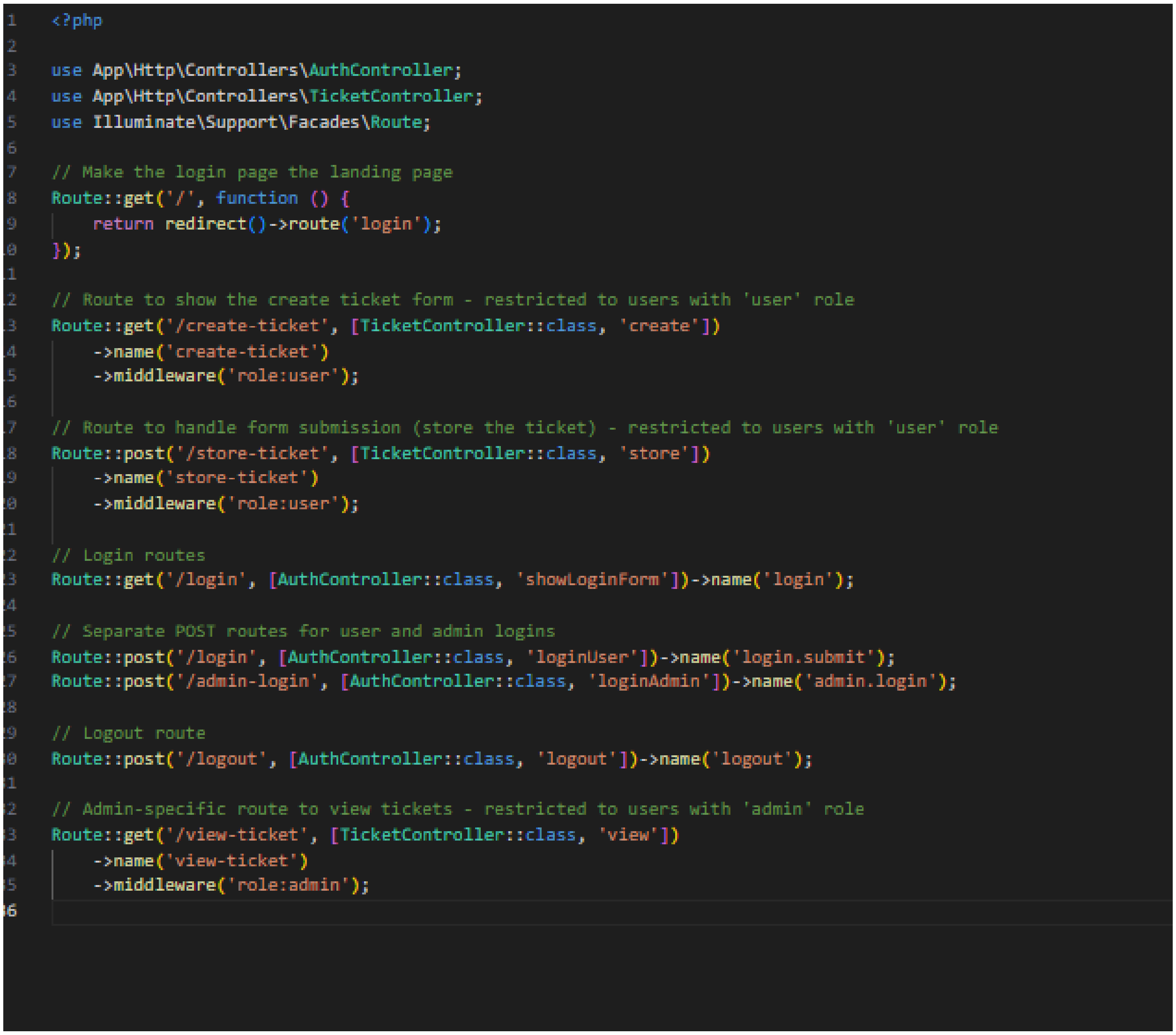
```
@extends('layouts.app')

@section('content')
    <div class="container">
        <h1>View Tickets</h1>
        <p>This is the page where you can view all tickets.</p>
    </div>
@endsection

```

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">
                    <img src="{{ asset('images/search.png') }}" alt="Search Icon" />
                </button>
            </div>
```

6. Routes (Web.php)



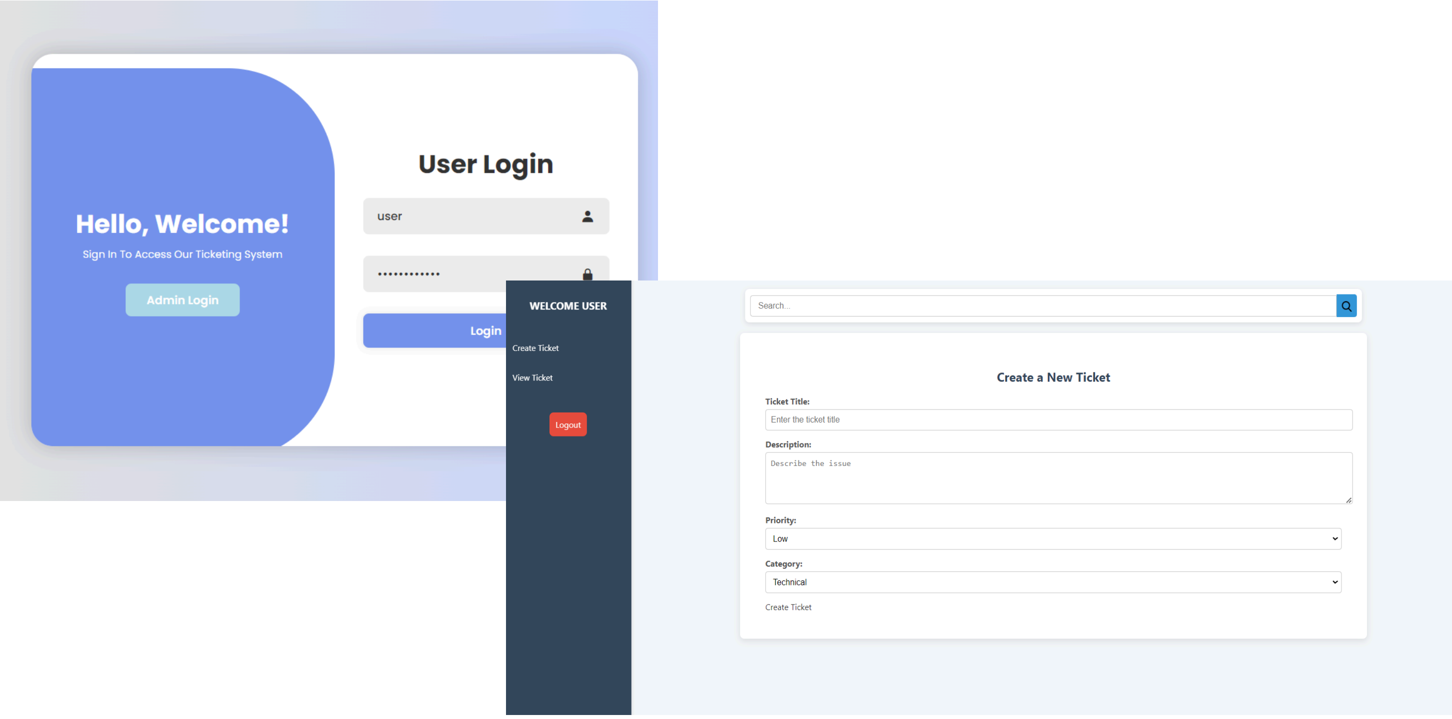
```php
<?php

use App\Http\Controllers\AuthController;
use App\Http\Controllers\TicketController;
use Illuminate\Support\Facades\Route;

// Make the login page the landing page
Route::get('/', function () {
    return redirect()->route('login');
});

// Route to show the create ticket form - restricted to users with 'user' role
Route::get('/create-ticket', [TicketController::class, 'create'])
    ->name('create-ticket')
    ->middleware('role:user');

// Route to handle form submission (store the ticket) - restricted to users with 'user' role
Route::post('/store-ticket', [TicketController::class, 'store'])
    ->name('store-ticket')
    ->middleware('role:user');

// Login routes
Route::get('/login', [AuthController::class, 'showLoginForm'])->name('login');

// Separate POST routes for user and admin logins
Route::post('/login', [AuthController::class, 'loginUser'])->name('login.submit');
Route::post('/admin-login', [AuthController::class, 'loginAdmin'])->name('admin.login');

// Logout route
Route::post('/logout', [AuthController::class, 'logout'])->name('logout');

// Admin-specific route to view tickets - restricted to users with 'admin' role
Route::get('/view-ticket', [TicketController::class, 'view'])
    ->name('view-ticket')
    ->middleware('role:admin');

```

- 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



Create a New Ticket

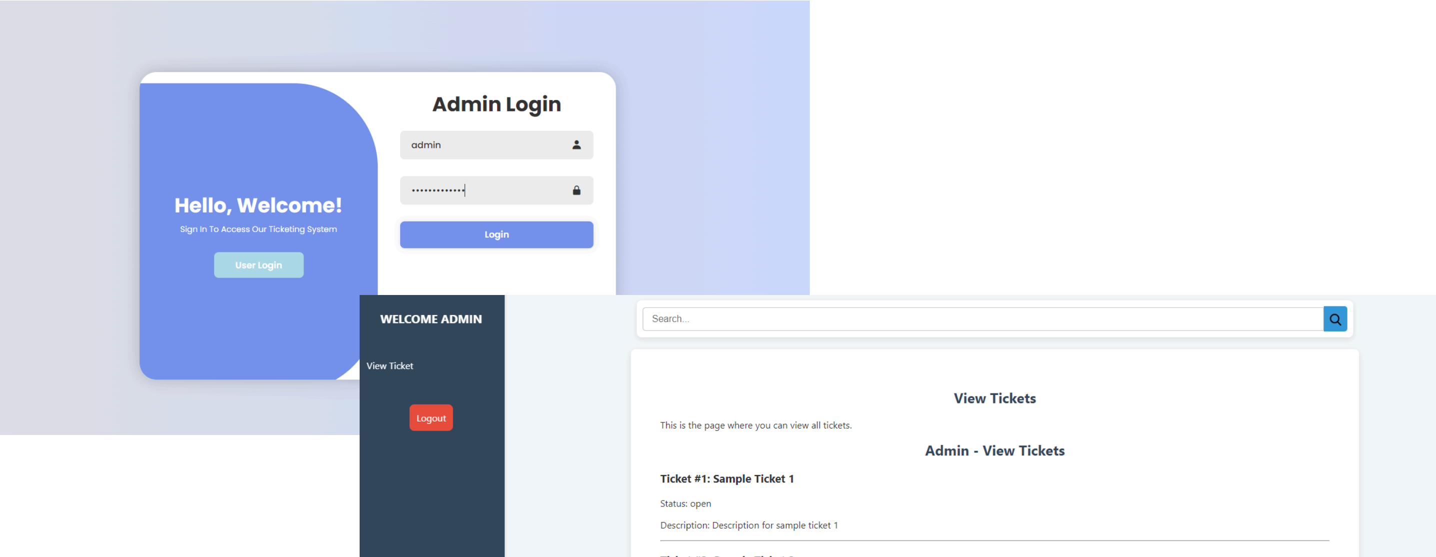
Ticket Title: Enter the ticket title

Description: Describe the issue

Priority: Low

Category: Technical

Admin Login & Dashboard



View Tickets

This is the page where you can view all tickets.

Admin - View Tickets

Ticket #1: Sample Ticket 1

Status: open

Description: Description for sample ticket 1