

## DOCUMENT 10

### USER LOGIN, LOGOUT, AND PROTECTING ADMIN PAGES

#### Goal

Create login and logout, then restrict Product pages so only Admin role gets access.

#### Step 1. Create a Login DTO

RIGHT CLICK ON MODEL FOLDER AND THEN CLICK ON ADD AND THEN CLICK ON CLASS AND THEN NAME THE FILE LoginDTO

#### Code to be put inside file below

```
1. using System.ComponentModel.DataAnnotations;
2.
3. namespace RTWebApplication.DTOs
4. {
5.     public class LoginDTO
6.     {
7.         [Required]
8.         [EmailAddress]
9.         public string Email { get; set; } = "";
10.
11.         [Required]
12.         public string Password { get; set; } = "";
13.
14.         public bool RememberMe { get; set; }
15.     }
16. }
17.
```

#### Explanation

RememberMe controls cookie persistence.

## Step 2. Add Login actions in AccountController

Add these methods.

```
1.      [HttpGet]
2.      public IActionResult Login()
3.      {
4.          return View();
5.      }
6.
7.      [HttpPost]
8.      public async Task<IActionResult> Login(LoginDTO loginDTO)
9.      {
10.         if (!ModelState.IsValid)
11.         {
12.             return View(loginDTO);
13.         }
14.
15.         var signIn = await signInManager.PasswordSignInAsync(loginDTO.Email,
loginDTO.Password, loginDTO.RememberMe, false);
16.         if (signIn.Succeeded)
17.         {
18.             return RedirectToAction("Index", "Home");
19.         }
20.         else
21.         {
22.             ViewBag.ErrorMessage = "Invalid Login Attempt";
23.
24.         }
25.         return View(loginDTO);
26.     }
27.
```

### Explanation in detail

**PasswordSignInAsync** checks credentials and creates the auth cookie.

**UserName** is set to **Email** in registration, so logging in with email works.

### Step 3. Create Login view

Create Views/Account/Login.cshtml.

@model LoginDTO

```
1. <h2>Login</h2>
2.
3. <form asp-action="Login" method="post">
4.     <div>
5.         <label>Email</label>
6.         <input asp-for="Email" />
7.         <span asp-validation-for="Email"></span>
8.     </div>
9.
10.    <div>
11.        <label>Password</label>
12.        <input asp-for="Password" type="password" />
13.        <span asp-validation-for="Password"></span>
14.    </div>
15.
16.    <div>
17.        <label>Remember me</label>
18.        <input asp-for="RememberMe" type="checkbox" />
19.    </div>
20.
21.    <button type="submit">Login</button>
22. </form>
23.
```

### Step 4. Add Logout action

Add to AccountController.

```
1. [HttpPost]
3. public async Task<IActionResult> Logout()
4. {
5.     await _signInManager.SignOutAsync();
```

```
6.         return RedirectToAction("Index", "Home");
7.     }
8.
```

## Explanation

SignInAsync clears the auth cookie.

## Step 5. Add Login and Logout links in \_Layout.cshtml

Add the Identity injections at the top of \_Layout.cshtml.

```
1. @using Microsoft.AspNetCore.Identity
2. @using RTWebApplication.Models
3. @inject SignInManager<ApplicationUser> SignInManager
4. @inject UserManager<ApplicationUser> UserManager
5.
```

Then add conditional nav links.

**RIGHT AFTER THESE INITIAL LINKS YOU HAVE UNDER THE LAST**

```
<div class="collapse navbar-collapse" id="mainNavbar">
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link text-dark" asp-controller="Home" asp-action="Index">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark" asp-controller="Home" asp-action="Privacy">Privacy</a>
        </li>
    </ul>
```

**</LI> FROM THIS PICTURE**

**THEN YOU ADD THIS**

```
@if (User.Identity != null && User.Identity.IsAuthenticated && User.IsInRole("Admin"))
{
    <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Product" asp-action="Index">Products</a>
    </li>
}
</ul>
```

```

<ul class="navbar-nav ms-auto align-items-center">

    @if (SignInManager.IsSignedIn(User))
    {
        var user = await UserManager.GetUserAsync(User);
        <li class="nav-item">
            <span class="navbar-text me-3">Hello, @user.FirstName!</span>
        </li>

        <li class="nav-item dropdown">
            <a class="btn btn-light dropdown-toggle"
                href="#"
                role="button"
                data-bs-toggle="dropdown"
                aria-expanded="false">
                Account
            </a>

            <ul class="dropdown-menu dropdown-menu-end">
                @if (User.IsInRole("Admin"))
                {
                    <li>
                        <a class="dropdown-item" asp-controller="Admin" asp-action="Index">Admin dashboard</a>
                    </li>
                }

                <li>
                    <a class="dropdown-item" asp-controller="Account" asp-action="Profile">Profile</a>
                </li>

                <li><hr class="dropdown-divider"></li>

                <li>
                    <form asp-controller="Account" asp-action="Logout" method="post" class="px-3">
                        <button type="submit" class="btn btn-danger w-100">Log out</button>
                    </form>
                </li>
            </ul>
        </li>
    }
    else
    {
        <li class="nav-item me-2">
            <a asp-controller="Account" asp-action="Login" class="btn btn-outline-primary">Log in</a>
        </li>
        <li class="nav-item">
            <a asp-controller="Account" asp-action="Register" class="btn btn-primary">Register</a>
        </li>
    }
</ul>

```

Then run the program to check how it works