Adding Roles and User Roles in ASP.NET Core MVC

Step 1: Setup Identity in your MVC App

```
Install required packages:
dotnet add package Microsoft.AspNetCore.Identity.EntityFrameworkCore
dotnet add package Microsoft.EntityFrameworkCore.SqlServer

DbContext:
using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore;

public class ApplicationDbContext : IdentityDbContext<IdentityUser>
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
    {
    }
}
```

Step 2: Register Identity in Program.cs

```
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

builder.Services.AddIdentity<IdentityUser, IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>()
    .AddDefaultTokenProviders();

builder.Services.AddControllersWithViews();
```

Step 3: Create Roles

```
async Task CreateRolesAsync(IApplicationBuilder app)
{
    using var scope = app.ApplicationServices.CreateScope();
    var roleManager = scope.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>();

    string[] roleNames = { "Admin", "Manager", "User" };

    foreach (var roleName in roleNames)
    {
        if (!await roleManager.RoleExistsAsync(roleName))
        {
            await roleManager.CreateAsync(new IdentityRole(roleName));
        }
    }
}

await CreateRolesAsync(app);
```

Step 4: Assign a Role to a User

```
public class AccountController : Controller
{
    private readonly UserManager<IdentityUser> _userManager;

    public AccountController(UserManager<IdentityUser> userManager)
    {
        _userManager = userManager;
    }

    public async Task<IActionResult> AddUserToRole(string userEmail, string roleName)
    {
        var user = await _userManager.FindByEmailAsync(userEmail);

        if (user == null)
        {
            return NotFound("User not found");
        }

        var result = await _userManager.AddToRoleAsync(user, roleName);

        if (result.Succeeded)
        {
            return Ok($"User {userEmail} added to role {roleName}");
        }

        return BadRequest(result.Errors);
    }
}
```

Step 5: Check User Roles

```
if (User.IsInRole("Admin"))
{
    // Do admin stuff
}

[Authorize(Roles = "Admin")]
public IActionResult AdminOnlyPage()
{
    return View();
}
```