Documentation

Solution Structure

This solution is made up of three main projects:

- **UserManagement.API**
  This is the ASP.NET Core Web API. It contains:

    - Database models for Users, Groups, and Permissions

    - Entity Framework Core DbContext and migrations

    - API controllers for creating, updating, deleting, and reading users

    - Endpoints for total user count and users per group

    - Global error handling

- **UserManagement.Web**
  This is the web interface built with Razor Pages. It:

    - Provides pages to add, edit, and delete users

    - Calls the API for all data operations

    - Does not access the database directly

- **UserManagement.Tests**
  This project contains automated tests using xUnit.
  The tests verify:

    - Creating a user

    - Updating a user

    - Returning the total user count

    - Returning the number of users per group

The main solution file is KingPriceAssessment.sln.

Key Technical Decisions

These are design choices made to meet the requirements in a clean and maintainable way:

- **Separation between API and Web UI**
  The Web project does not talk to the database directly. All data is handled through the API. This keeps responsibilities clear and makes the system easier to maintain and extend.

- **Use of DTOs and mapping**
  The API uses DTOs to control what data is sent and received. Mapping methods convert between DTOs and database models. This helps keep the API contract clean and prevents exposing database entities directly.

- **Centralised error handling**
  A global exception handler is used in the API so errors are handled in one place. This keeps controllers simpler and gives consistent error responses.

- **Simple test setup using in-memory database**
  Automated tests use an in-memory database so they can run quickly and without needing a real SQL Server. This makes it easy for anyone to run tests after cloning the repository.

- **Clear folder and naming structure**
  Files are grouped by purpose (Controllers, Models, DTOs, Mappers, Tests) to make the project easy to understand and navigate.