StudentPortal Project Documentation

\*\*Created by:\*\* Sneha Mane

\*\*LinkedIn:\*\* https://www.linkedin.com/in/sneha-mane-697a97156/

# 1. Project Overview

StudentPortal is a .NET 6 MVC-based web application designed to manage student-related data and operations. It serves as a sample application that showcases clean architecture, proper use of Entity Framework Core, and good project structuring for learning and demonstration purposes.

# 2. Technology Stack

- ASP.NET Core MVC (.NET 6)  
- Entity Framework Core  
- C#  
- Razor Pages  
- SQLite (app.db)  
- Visual Studio

# 3. Project Structure

- StudentPortal.sln – Solution file  
- StudentPortal.Web – Main Web Application  
 - Program.cs – Application startup configuration  
 - appsettings.json – Main configuration file  
 - app.db – SQLite database  
 - \*.csproj – Project file defining dependencies  
- .vs/ and obj/ – IDE and build generated folders (can be ignored for source control)

# 4. Architecture Flow

Below is a basic logical flow of the MVC architecture used in the application:

User Request -> Controller -> Service/Business Logic -> Data Access (EF Core) -> Database  
 ↓  
 View (Razor Pages)

# 5. Key Features

- Modular project setup using ASP.NET Core MVC  
- Razor page integration  
- SQLite database for development/demo purposes  
- Clean code structure following best practices

# 6. How to Run the Project

1. Open `StudentPortal.sln` in Visual Studio 2022 or higher.  
2. Restore NuGet packages (Visual Studio will usually prompt this).  
3. Build the solution.  
4. Run the application (F5 or Ctrl+F5).

# 7. Future Scope / Enhancements

- Add authentication and user roles.  
- Deploy using Azure or another cloud platform.  
- Extend with CRUD operations for students and related entities.  
- Add unit testing and logging.