**School Management System MVC Project**

(Database-first Approach)

Design Pattern – IoC(Inversion of Controll)

Docker Configuration:

* SQL Server Image:

docker pull mcr.microsoft.com/mssql/server

* Create Docker Container:

docker run -e 'ACCEPT\_EULA=Y' -e 'SA\_PASSWORD=UPPERlower123@#' -p 1400:1433 -d mcr.microsoft.com/mssql/server

Database:

* Create DB

CREATE DATABASE SchoolDB

GO

USE SchoolDB;

* Create Tables

CREATE TABLE Students(

    Id INT PRIMARY KEY IDENTITY,

    FirstName NVARCHAR(50) NOT NULL,

    LastName NVARCHAR(50) NOT NULL,

    DateOfBirth DATE,

)

CREATE TABLE Lecturers(

    Id INT PRIMARY KEY IDENTITY,

    FirstName NVARCHAR(50) NOT NULL,

    LastName NVARCHAR(50) NOT NULL,

)

CREATE TABLE Courses(

    Id INT PRIMARY KEY IDENTITY,

    Name NVARCHAR(50) NOT NULL,

    Code NVARCHAR(5) UNIQUE,

    Credits INT,

)

* Database Connection String (appsettings.json)

"ConnectionStrings": {

    "DefaultConnection": "Server=localhost, 1400;Database=SchoolDB;Trusted\_Connection=false;MultipleActiveResultSets=true;Encrypt=false;User Id=sa;Password=UPPERlower123@#;"

}

* Database Scaffolding (Generate Code from Database Tables and Fields):

dotnet ef dbcontext scaffold "Server=localhost, 1400;Database=SchoolDB;Trusted\_Connection=false;MultipleActiveResultSets=true;Encrypt=false;User Id=sa;Password=UPPERlower123@#;" Microsoft.EntityFrameworkCore.SqlServer -o Data -f --no-onconfiguring

* Configur eDependancy Injection DBContext in Program.cs

var config = builder.Configuration.GetConnectionString("DefaultConnection");

builder.Services.AddDbContext<SchoolDbContext>(options =>

    options.UseSqlServer(config));

Packages

Installed Packages:

* 1. Microsoft.EntityFrameworkCore
  2. Microsoft.EntityFrameworkCore.Design
  3. Microsoft.EntityFrameworkCore.SqlServer
  4. Microsoft.EntityFrameworkCore.Tools
  5. Microsoft.VisualStudio.Web.CodeGeneration.Design

Terminal CLI Commands:

dotnet add package Microsoft.EntityFrameworkCore

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Design

dotnet add package Microsoft.EntityFrameworkCore.Tools

dotnet add package Microsoft.VisualStudio.Web.CodeGeneration.Design

Tools

Installed Tools:

1. Entity Framework
2. Code Generator

Commands:

dotnet tool install --global dotnet-ef

dotnet tool install --global dotnet-aspnet-codegenerator

Controllers

Created Controllers vis Codegenerator. Code Generator creates Controller Class, Razor view page (Create, Edit, Details, Delete and Index page)

1. CoursesController
2. LecturersController
3. StudentsController

Code Generator CLI Command:

dotnet aspnet-codegenerator controller -name CoursesController -m Course -dc SchoolDbContext --relativeFolderPath Controllers --useDefaultLayout --referenceScriptLibraries -f

dotnet aspnet-codegenerator controller -name LecturersController -m Lecturer -dc SchoolDbContext --relativeFolderPath Controllers --useDefaultLayout --referenceScriptLibraries -f

dotnet aspnet-codegenerator controller -name StudentsController -m Student -dc SchoolDbContext --relativeFolderPath Controllers --useDefaultLayout --referenceScriptLibraries -f

Frontend -Razor Pages

* Alert – Sweetalert2

View > Shared > \_Layout.cshtml (below jquery script)

<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

jquery code

@section Scripts{

    <script>

        $(function () {

            $('.dltBtn').click(function (e){

                //sweetAlert code

            })

        });

    </script>

}