

Assignment 03

Building a Post Management Website using Real-Time Communication with SignalR

Introduction

Imagine you are an employee of IT department of a media company. Your manager has asked you to develop a web application for post management. The post management system must follow the real-time communication application. When user makes the post, this post will automatically update for all connected clients.

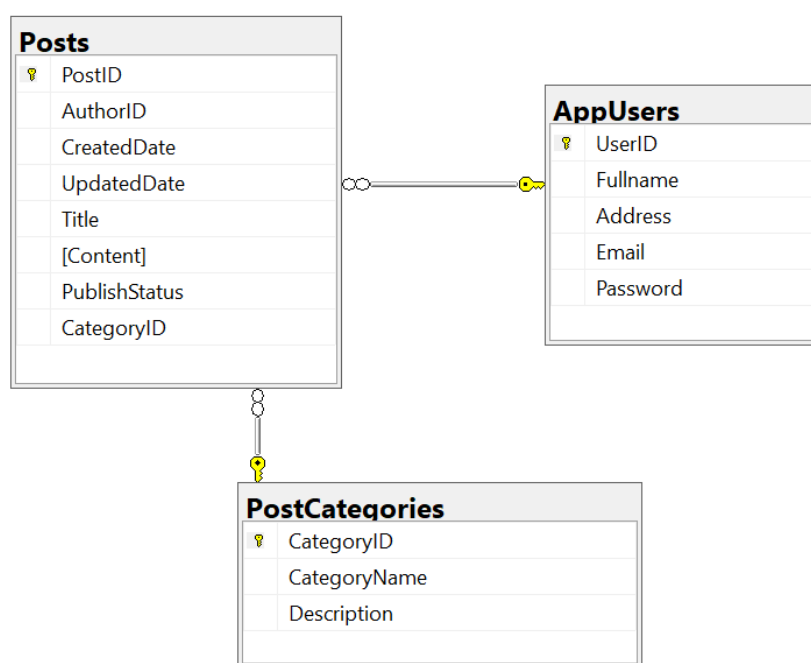
The application has to support adding, viewing, modifying, and removing posts; standardized usage action verbs better known as Create, Read, Update, Delete (CRUD) and Search. This assignment explores creating an application using Real-Time Communication with SignalR .NET Core Web Application, C#, and Entity Framework Core. The MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

Assignment Objectives

In this assignment, you will:

- Use the Visual Studio.NET to create ASP.NET Core Web Application (Model-View-Controller) project.
- Perform CRUD actions using Entity Framework Core combined with Real-Time Communication with SignalR.
- Apply to validate data type for all fields.
- Run the project and test the ASP.NET Core Web Application actions.

Database Design



PostCategories (CategoryID, CategoryName, Description)

AppUsers (UserID, Fullname, Address, Email, Password)

Posts (PostID, AuthorID, CreatedDate, UpdatedDate, Title, Content, PublishStatus, CategoryID)



Main Functions

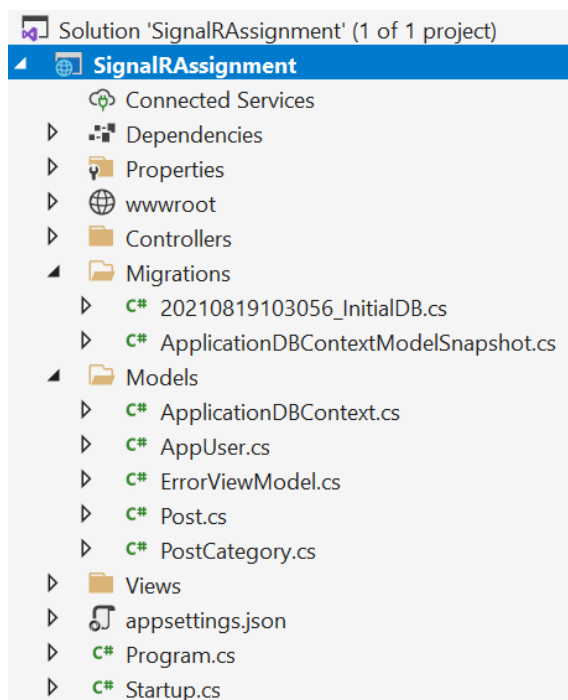
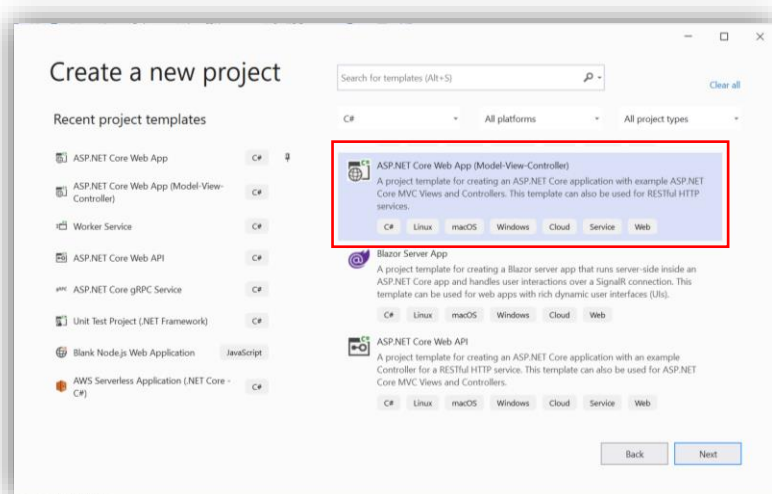
- Create database using Entity Framework Core.
- Member management, post management, and registration management:
Read, Create, Update and Delete actions.
- Search Post by ID , Title or Description
- Create a report statistics posts by the period from StartDate to EndDate, and
sort post in descending order
- Member registration by Email and Password.

Guidelines

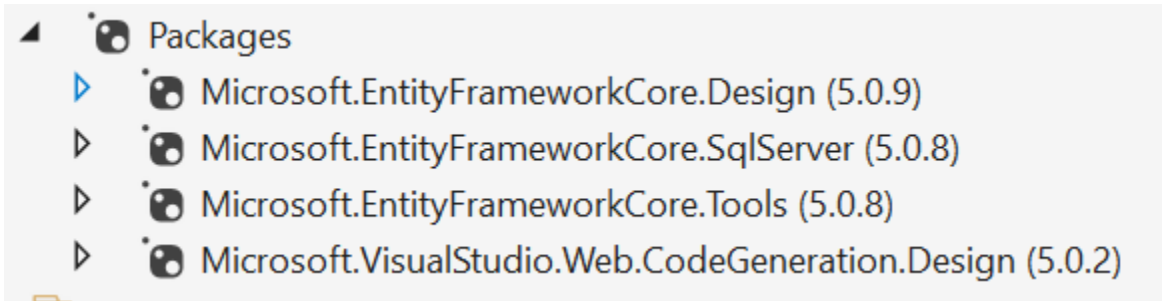
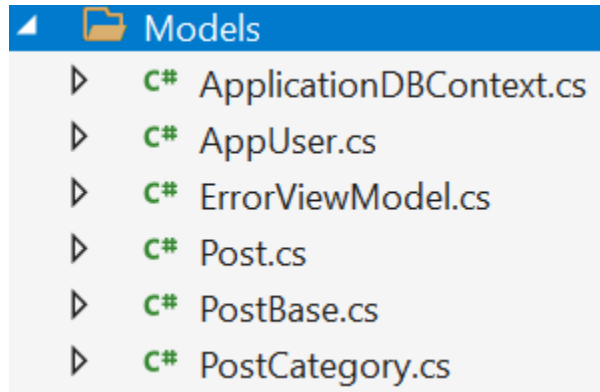
Activity 01: Build a solution [01 mark]

Step 01. Create Project/Solution named SignalRAssignment

Step 02. Choose the Project template: ASP.NET Core Web Application (Model-View-Controller)



Activity 02: Create database with Entity Framework [03 marks]



Step 01. Create Model

Step 02. Create database context with Entity Framework
ApplicationDbContext

Step 03. Add-Migration and then Update Database

Activity 03: CRUD post item using Real-Time Communication with SignalR [05 marks]

Step 01. Create Registration with the information

Step 02. Create a new post. When user create a new post, this information will automatically update with all other connected users using .NET Core SignalR.

Step 03. Update an existing post. When user update an existing post, this information will automatically notify with all other connected users using .NET Core SignalR.

Step 04. Delete an existing post. When user delete an existing post (owner), this information will automatically update with all other connected users using .NET Core SignalR.

Step 05. View detail of the post.

Step 06. View list of posts with paging.

Activity 04: Run the project and test all actions [01 mark]