

Assignment 02 - Summer 2023 HCM

Building ASP.NET Web API with Entity Framework Core and Web Application

1. Introduction

Implement a part of Project Participant Management System. The situation is described as a project has many employees involved and an employee can be participated in many projects with difference time (note that the time involved in projects does not overlap).

Imagine you're a developer, your manager has asked you to develop a Web application for employee management, company project management, participating management The application has a default account whose email is “**admin@eprojectmanagement.com**” and password is “**admin@@**” that stored in the **appsettings.json**.

The application has to support adding, viewing, modifying, and removing information - a standardized usage action verb better known as Create, Read, Update, Delete (CRUD) and Search. This assignment explores creating an ASP.NET Core Web API with OData, and ADO.NET or Entity Framework Core, the client application (using ASP.NET Core Web MVC or Razor Pages or Desktop, Mobile app). An MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

2. Assignment Objectives

In this assignment, you will:

- Use the Visual Studio.NET to create a Web application and ASP.NET Core Web API project (with OData support).
- Perform CRUD actions using Entity Framework Core with *Code First approach*.

- Using OData (a data access protocol for the web) to query and manipulate data sets.
- Apply 3-layers architecture to develop the application.
- Apply Repository pattern and Singleton pattern in a project.
- Add CRUD and searching actions to the Web application with ASP.NET Core Web API.
- Apply to validate data type for all fields.
- Run the project and test the actions of the Web application.

3. Database Design

A part of *Project Participant Management* database is described as the below:

Department(DepartmentID, DepartmentName, DepartmentDescription)

All fields are required.

CompanyProject(CompanyProjectID, ProjectName, ProjectDescription,
EstimatedStartDate, ExpectedEndDate)

All fields are required.

ParticipatingProject(CompanyProjectID, EmployeeID, StartDate, EndDate,
ProjectRole)

All fields are required.

Project Role: Project Manager = 1; ProjectMember = 2

Employee(EmployeeID, FullName, Skills, Telephone, Address, Status, DepartmentID,
EmailAddress)

All fields are required. Status = Active/Inactive. EmailAddress is unique.

4. Main Functions

- Create database with Forward Engineering approach (create model classes and DB context class then make migration).
- Create Web API with OData: Employee management, project management, Participant management: Read (list all, search), Create, Update and Delete actions.

- Create Client application (with Desktop/Web/Mobile application) interactive with Web API to perform these functions:
 - Manage employee information, manage company project information.
 - Get all participating information of *specific company project* (the information includes EmployeeID, EmployeeName, EstimatedStartDate, EstimatedEndDate, ProjectRole).
 - Employee authentication by Email and Password. If the user is “**Admin**” then allows to perform all actions, otherwise, the normal user is allowed to view personal information/view list of participating projects/update the profile.

5. Notes

- You must use **Visual Studio 2019 or above** (.NET5/.NET6/.NET7), **MSSQL Server 2012 or above** for your development tools.
- To do your BE program, you must use **ASP.NET Core Web API with OData**. *Note that you are not allow to connect direct to database from ASP.NET Core Web API, every database connection must be used through Repository and Data Access Objects. The database connection string must get from appsettings.json file.*
- Create Solution in Visual Studio named **StudentName_ClassCode_A02.sln**. Database after migration should be named: **StudentName_ClassCode_A02**