Practice Project-4

Section 9 WriteUp

**ASP.NET Core Web API for Managing Student Subjects and Marks**

ASP.NET Core is a versatile framework that allows developers to create powerful web applications, including Web APIs. In this, I will be setting up an ASP.NET Core Web API project for managing student subjects and their marks.

**Step 1: Project Setup**

1. Open Visual Studio and create a new project.

2. Choose the "ASP.NET Core Web API" template.

3. Provide a suitable name for your project and click "Create."

**Step 2: Model Creation**

1. Inside your project, create a "Models" folder.

2. In the "Models" folder, create a class for Mark include properties for Student, Subject and Mark. This class will define the structure of your data.

**Step 3: Installing Entity Framework Core Packages**

1. To work with a database, you'll need Entity Framework Core. Install the necessary packages by running the following commands in the Package Manager Console:

Install-Package Microsoft.EntityFrameworkCore.SqlServer

Install-Package Microsoft.EntityFrameworkCore.Tools

**Step 4: Controller Generation**

1. Now, let's generate a controller for your models. Right-click on the "Controllers" folder.

2. Select "Add" -> "Controller."

3. In the pop-up window, choose "API" in the right pane and select "API Controller with actions, using Entity Framework."

4. Choose your model class, DbContext, and give your controller a name.

This step will create a controller and configure the necessary routes for CRUD operations.

**Step 5: Database Configuration**

1. The Entity Framework controller generation process also adds a connection string to your `appsettings.json` file.

2. In `appsettings.json`, modify the database name and server name to match your SQL Server configuration.

**Step 6: Database Migration**

1. In the Package Manager Console, run the following commands to create a database migration and apply it to your database:

Add-Migration YourMigrationName

Update-Database

These commands will create the database tables based on your model classes.

**Step 7: API Operations**

1. With the database set up, you can now perform various API operations:

- `GET`: Retrieve a list of students, subjects, or marks.

- `GET by ID`: Retrieve a specific student, subject, or mark by its unique identifier.

- `POST`: Create a new student, subject, or mark.

- `PUT by ID`: Update an existing student, subject, or mark.

- `DELETE by ID`: Delete an existing student, subject, or mark.

Your ASP.NET Core Web API is now ready to manage student subjects and marks. You can use tools like Postman or Swagger to test and interact with your API.

**Conclusion**

In this project, we set up an ASP.NET Core Web API project for managing student subjects and marks. This powerful framework allows you to create a robust backend for various applications, including mobile apps, web apps, and more.