EMS Full Stack App For Simplona Tech

Creating an ASP.NET Core Web API for Employee Management involves a series of well-defined steps. Here's a detailed walkthrough of the process:

Create ASP.NET Core Web API Project:

Start by setting up a new ASP.NET Core Web API project in your development environment. Ensure that you select the appropriate project template designed for creating APIs.

Model Classes for Employee and Department:

Design the database schema by creating model classes that represent your data entities. In this case, we have two main entities: EmpProject and DeptMaster.

Define the necessary variables within these classes to accurately map to the database tables.

Install Entity Framework Core Packages:

Utilize NuGet Package Manager to install the required packages: Microsoft.EntityFrameworkCore.Tools and Microsoft.EntityFrameworkCore.SqlServer. These packages are essential for working with Entity Framework Core and SQL Server.

Create Controllers:

Develop controllers for each model class. You will need controllers named EmpProjectsController and DeptMastersController. These controllers handle HTTP requests and responses related to employee projects and department masters.

Configure AppSettings:

In the appsettings.json file, configure the server's name and specify the appropriate database name. This configuration ensures that the application connects to the correct database.

Entity Framework Migrations:

Employ Entity Framework Migrations to manage the database schema effectively. Use the Package Manager Console (PMC) to create a database table and generate migration files.

Utilize commands such as add-migration and update-database to create and update the database based on your defined models and their relationships.

Data Population:

Populate the database tables with initial data. In this context, we will insert records for employees and departments. These initial records provide a foundation for your Employee Management System.

Debug and Execution:

Debug and execute your project to ensure its functionality. This step involves running your API to test its endpoints and verify that they operate as intended.

Testing with Swagger:

Enhance the testing process by integrating Swagger into your application. Swagger provides a user-friendly interface for testing API endpoints.

Simulate various scenarios, including adding, updating, and deleting records, to validate that the API behaves correctly.

GitHub Repository:

Once we are satisfied with the changes, we commit them using Git and push to our GitHub repository. This will track our SQL scripts and VS code.

git init

git add .

git commit -m "Creating a MVCSchoolDatabase"

git branch -M SimplonaTech

git remote add origin https://github.com/PrashastVats1/Phase2Projects.git

git push -u origin SimplonaTech

GitHub Link :

git remote add origin
https://github.com/PrashastVats1/Phase2Projects/tree/SimplonaTech