Phase-2 Course End Project

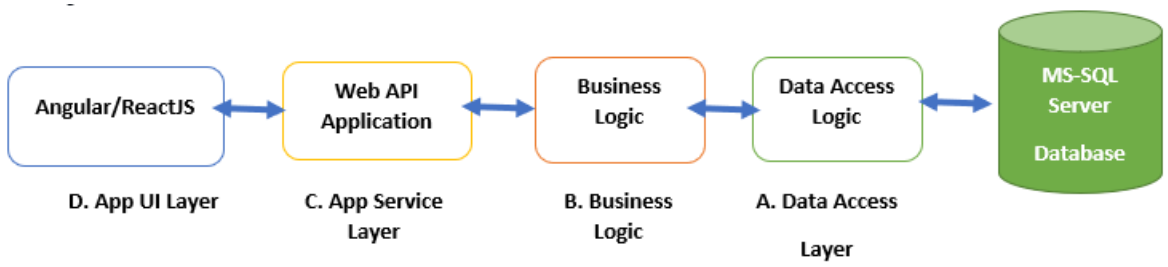
EMS Full Stack App-Requirement

Case Study:

Simplona Tech. Solutions require a Full Stack application EMS (Employee Management System) to maintain their employee profile details.

You need to create a Full Stack Application to maintain the details of employees using the multi-layered application described in the system overview.

1. System Overview:



A. Data Access Layer (DAL): This layer will be a type of Class Library. This application will create a database using the Entity Framework Code First Approach and implement Database functionalities such as Saving Employee details, Get All Employee Details, Get Employee by Code, Update Employee Details, and Delete Employee Details.

Consider the below Entity Model classes to create a database for the same.

1. Add entity class **DeptMaster** and given public properties

Property Name	**Type**
DeptCode	int
DeptName	string
virtual EmpProfiles	ICollection<EmpProfile>

2. Add entity class **EmpProfile** and the following public properties

Property Name	**Type**
EmpCode	int
DateOfBirth	DateTime
EmpName	string
Email	string
DeptCode	Int
virtual DeptMaster	DeptMaster

3. After creating the above entity classes, create the Context class by inheriting **DbContext** class

4. Add one more class by inheriting **DropCreateDatabaseIfModelChanges<ContextClass>** and add some default data into the **DeptMaster** table by overriding the Seed method

5. Write database functionalities for **EmpProfile** entity using the Data Repository Pattern: Saving Employee details, Get All Employee Details, Get Employee by Code, Update Employee Details, and Delete Employee Details

B. Business Logic Layer (BLL): This layer will be a type of Class Library. This application will invoke functionalities from the DAL class.

1. Add a DAL library reference to this project
2. Add a class to invoke functionalities from the DAL class

C. App Service Layer: This layer will be a type of ASP.Net Web Application (Web API). This application contains RESTful services to consume functionalities from BLL Class.

1. Add Web API controller and write action methods to issue a GET and POST, PUT and DELETE request perform: Saving Employee details, Get All Employee Details, Get Employee by Code, Update Employee Details, and Delete Employee Details
2. Use attribute-based routing while implementing the above functionalities
3. Enable Swagger support for documentation to test this application as shown in the Output