1.Create main project

2.Add second project as class library names it mainproject.entites (this project will only contain database)

3.Add third project as class library name it mainproject.repository(this will contain interface and repository class)

4.Install required nuget package to solution such as ef core tool, ef core sql, ef core design.

5.Add connection string to appsetting json file

for example

"ConnectionStrings": {

"Employee": "Server=PCLPT160\\SQL2019;Initial Catalog=Muhafil;Persist Security Info=False;User ID=sa;Password=Tatva@123;MultipleActiveResultSets=False;Encrypt=False;TrustServerCertificate=False;Connection Timeout=30;"

}

6.Open nugent package manger console now select project mainproject.enties so that we can execute scaffold command there

Sample command :

Scaffold-DbContext Name=ConnectionStrings:Employee Microsoft.EntityFrameworkCore.SqlServer -OutputDir DataModels -Context EmployeeDbContext

7. create folder interface and repository inside mainproject.repository

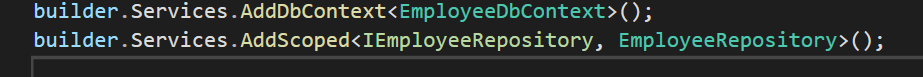
8.add Iemployeerepository interface to interface folder and employeerepository class to repository folder.

9.Now define method name in interface for example public employee getemployeedata().

10. Now define above interface body in repository class and implement interface in repository class.

Here in repository we will injecting dbcontext class to access database model and fetch the data or perform CRUD operation.

11. add dbcontext and addscoped reference to program.cs file



12. Now inject your Iemployeerepository interface in home controller and use its method to perfume operation such as CRUD.

Question – why we are using interface ?

Answer – To make it loosely couple , if we use direct class without interface and suppose one of your class method is used by many other controller, and if you change anything in class file it will reflect in all controller and it will increase if else statement in class file to handle such scenario.

So we introduce interface which handle the request and call the repository class so in case if changes are there we can add another method to interface and make changes instead of direct changes in class file.