Project Name: **Contact Management** Author: **Sanjay Kumar Sharma**

GitHub URL: <https://github.com/Sanjay-Kr-Sharma/ContactManagement>

**Technologies Used**:

Dotnet Framework 4.5, ASP .Net Web API, Code First Entity Framework 6, MS Test Framework (Automated Unit Testing) and Moq Framework (to mock the database for unit testing)

**Database :** SQL Server 2008R2

**IDE :** Visual Studio 2017.

**User Interface :** jQuery and jQuery DataTables

**DotNet, Entity and MS Test Framework**:

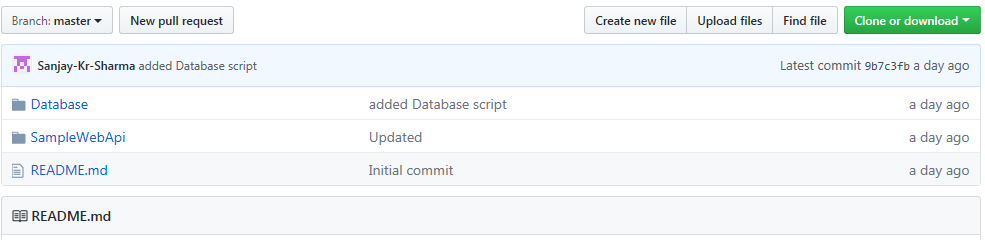
Entity framework that offers an automated mechanism to developer for storing and accessing the data in the database. Coding standard used by following *Solid Principles* and *Dependency Injection* are also used)

**Service**: To extend with Service layer (REST API): principle follow (Object and JSON)

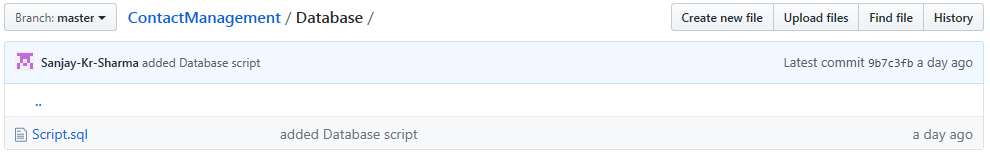
**Install Visual Studio 2017 and SQL server 2008R2**

1. Open solution file in Visual Studio 2017
2. Change Connection String in ***SampleWebAPI/web.config*** <ConnectionString > pointing to your local sql server database.
3. Change Connection String in ***SampleDataLayer/App.config*** <ConnectionString > pointing to your local sql server database.
4. Goto Tool->NuGet Package Manager-> Package Manager Console.
5. Set Default Project as ***SampleDataLayer***
6. Run ***Enable-Migration*** command and then ***Update-Database***
7. The above command will create database and table schemas.
8. Open SQL Server Management studio connect to Database Contact DB.
9. Run Query “*Insert Into User Values (‘Admin’, Admin’)*” to create master user to access the application. (Note: We can also create sign-up link to create first user in login page, that is not cover as of now)
10. Set ***SampleWebAPI*** as default/start-up project and run.

**Project Container Details in GitHub:**

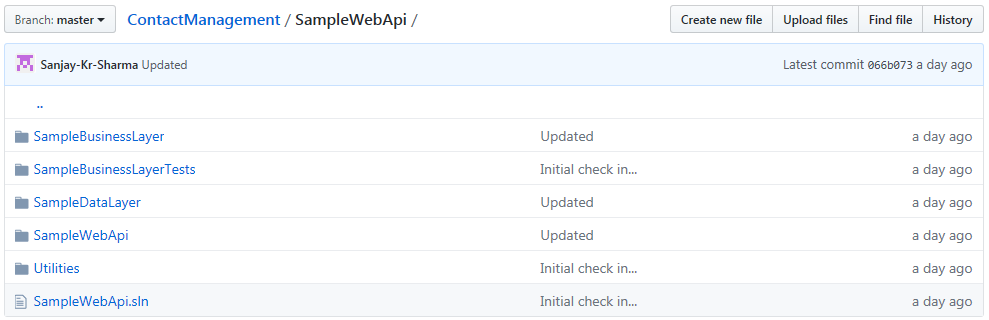


**[1] Database Container:**

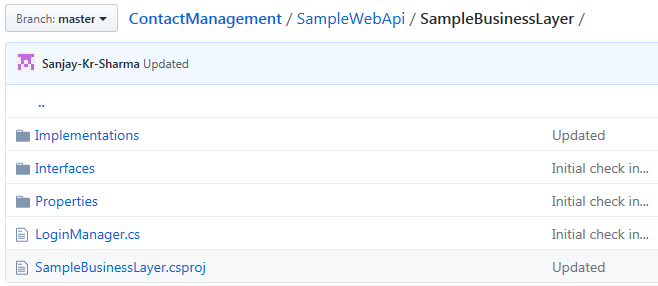


**Script.sql**: Contains database script (this is for reference purpose, as we are using Code first approach which will automatically generate database schema)

**[2] Sample Web API:** Contain the whole project.



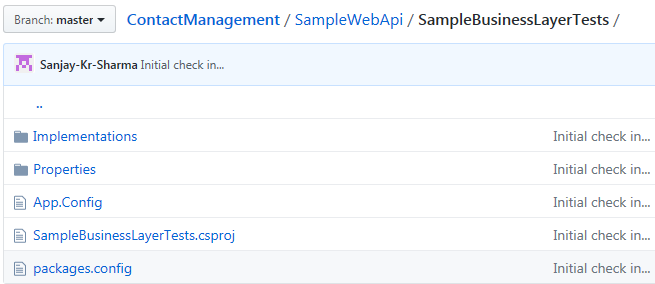
**[A] Sample Business Layer:**



Note:

1. Its contain the business logic of the project.
2. Receive input from *SampleWebAPI*, manipulate data as per business logic and pass to *SampleDataLayer*

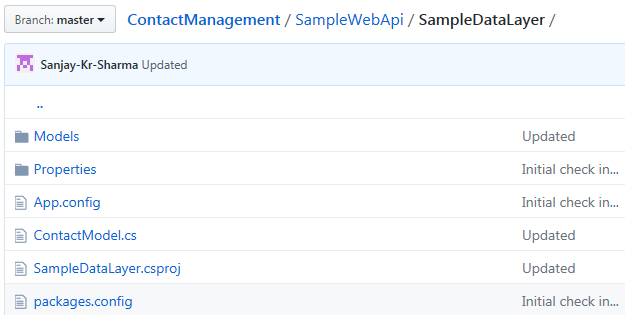
**[B] Sample Business Layer Tests**



**Note:**

1. Its contain all the automated unit test (ensure that connection string in *App.config* is updated)

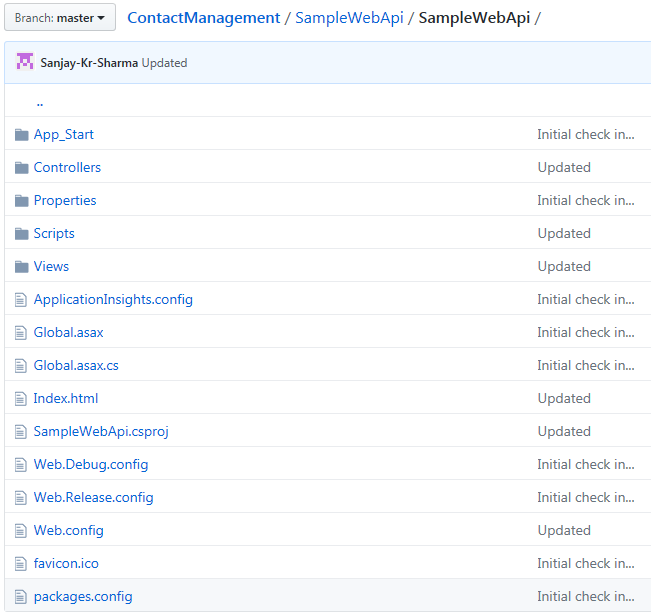
**[C] Sample Data Layer**



Note:

1. Its contain all the ORM (Object relational mapping) classes.
2. Receive input from *SampleBusinessLayer* and perform CURD operation in database.

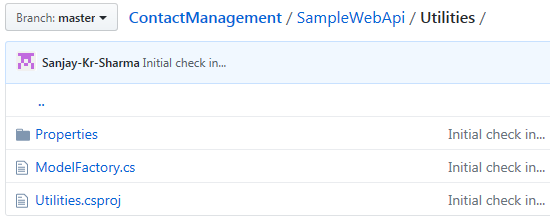
**[D] Sample Web API:**



Note:

1. Its contain the user interface details of the project.
2. Perform user action i.e login into application, add/update/delete/active contact information.
3. Capture user action and call *SampleBusinessLayer* method to perform operation.

**[E] Utilities**



Note:

1. Its contain the single utility class which return DB context (i.e Singleton framework approach).

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Thank You \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***