**Description of the folder structures of the solution.**

**The solution has the following folders**:

1. Core
2. Host
3. Services
4. UnitTest
5. **The Core folder**:

It has two projects in it: The Api.Contracts project and the Api.DAL project.

**The Api.Contracts** project has the following folders:

1. Error- This folder has all the files used for error handling. It has two sub folders:

* Exceptions: User defined exceptions are coded here
* Resources: It has the fault code and fault message resource files, used to give the user the correct error message in case the validation fails. The request to add a contact or update a contact has been validated. Should the validation fail the user will get the error message describing why the request was bad. Also the delete, deactivate, reactivate calls have been validated against the id provided.

1. Interfaces- All the interfaces used in the application has been put in this folder.
2. Models – All the models used in the application has been put in this folder.

**The Api.DAL** project is the data access layer of the application. It used entity framework to communicate with the underlying database.

1. **The Host folder** : This folder contains the code of the host application. It has the following sub folders:
2. Controllers: The ContactController (api controller) code used for that application has been put in this folder.
3. Middleware: The ErrorHandlingMiddleware used for handling any error occurring in the application has been put in this folder. This middleware will catch any exceptions/error occurring in the application and return the user with the correct error response. The middleware concept of ASP.NET core has been used to construct this custom middleware.
4. **The Services folder**: This is the middle layer between the controller and the data access layer. Build with the thought that if in future the data access layer changes there will be minimal changes required in the application. The data access layer has been injected in the service layer. So if a new data access layer has to be incorporated in the application , change will be only in the Startup.cs file where the DAL has been registered.
5. **UnitTest folder**: This folder contains the unit test for the ContactService.