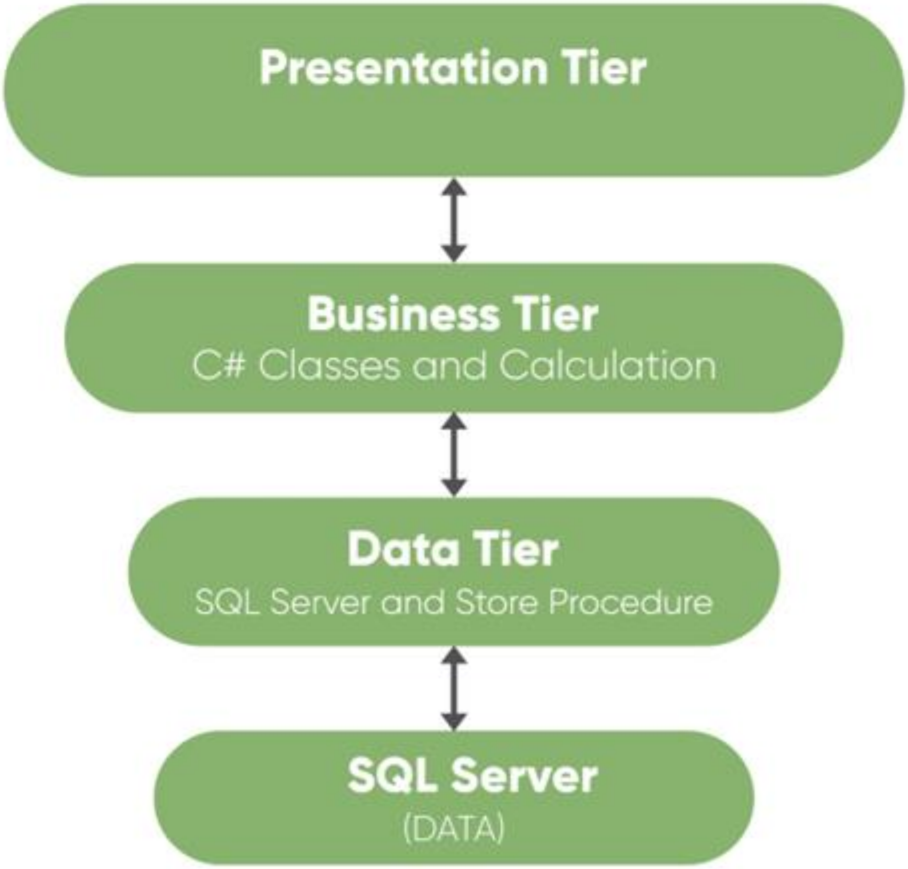
Three Tier – کارگاه معماری سه لایه

در این کارگاه یک پروژه عملی با معماری سه لایه را انجام میدهیم.

Three-layer architecture is dividing the project into three layers that are *User interface layer*, *business layer* and data(database) layer where we separate UI, logic, and data in three divisions. Suppose the user wants to change the UI from windows to the phone than he has to only make change in UI layer, other layers are not affected by this change. Similarly, if the user wants to change the database then he has to only make a change in the data layer, rest everything remains the same.

Why to use 3-Tier Architecture in Projects?

We use 3 Tier architecture to manage large projects. We can take an example: Suppose you have created 2000 entity code and you have coded all your code in only one layer. Now suppose you want to change from windows application to mobile or web application then again you have to write code for all 2000 entities or you want to change database provider than again you have to change database connection in all 2000 entities. Hence writing such a large amount of code is a time and money wasting. Also writing a large amount of code in one layer only makes new people difficult to understand that code. So, to provide maintenance, flexibility, updation flexibility without affecting other parts, clean and understandable code we should use 3-Tier Architecture in Projects. This architecture basically contains **three layers**:



1. Presentation Layer

This is the top layer of architecture. The topmost level of application is the user interface. It is related to the user interface that is what the user sees. The main function of this layer is to translate tasks and results in something which the user can understand. It contains pages like web forms, windows form where data is presented to the user and use to take input from the user. The presentation layer is the most important layer because it is the one that the user sees and good UI attracts the user and this layer should be designed properly.

2. Business Layer

This is the middle layer of architecture. This layer involves C# classes and logical calculations and operations are performed under this layer. It processes the command, makes logical decisions and perform calculations. It also acts as a middleware between two surrounded layers that is presentation and data layer. It processes data between these two layers. This layer implements business logic and calculations. This layer also validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. This validation of input is called business rules.

3. Data Layer

This layer is used to connect the business layer to the database or data source. It contains methods which are used to perform operations on database like insert, delete, update, etc. This layer contains stored procedures which are used to query database. Hence this layer establishes a connection with the database and performs functions on the database.

Advantages:

- Updation to new graphical environments is easier and faster.
- Easy to maintain large and complex projects.
- It provides logical separation between presentation layer, business layer and data layer.
- By introducing application layer between data and presentation layer, it provides more security to database layer hence data will be more secure.
- You can hide unnecessary methods from the business layer in the presentation layer.
- Data provider queries can be easily updated and OOP’s concept can be easily applied on project.

Disadvantages:

- It is more complex than simple client-server architecture.
- It is difficult to build and time consuming.
- User should have good knowledge of object-oriented concepts.