Advantages of Asp.net N-Tier Architecture over single tier architecture

1. Code Maintainability
2. Standardization
3. Reduce the lines of code
4. Separation of layer is concern
5. Security
6. Reusability

The N-Tier architecture which contains the more than 2 layers. Below are the some of the sample layers in N-Tier architecture

1. Presentation Layer
2. Business Layer
3. Data access Layer
4. Common Logic Layer etc..

The N-Tier which contains lot of advantages over the single-tier architecture

Code Maintainability

As I told the above line, the N-Tier Architecture which contains lot of layers. So we can define our logics in the appropriate layer so the layers are separated here based on the usage but the single tier architecture which contains all the business logic, data base query and all the codes are maintained in single layer itself so it’s is difficult to maintain the code.

Standardization

In the N-Tier architecture which contains the layers based on the developers usage so anyone can easily identify the logic while debugging the code and also if anyone can implement the new logic they can segregate their code based on the layers it is called standardization but the single layer we cannot do like that.

Reduce the lines of Code

Here the codes are segregated based on the layers so each layers have that particular code only so automatically the number of code will be reduced but in the single layer all the logics, db queries are maintained that layer itself.

Separation of layer is concern

The Data Access Layer which contains the DB connections, Object relational mappings and direct the calls from business layers to database.

The Business layer which contains all the logics which is involving in the application

Presentation layer which contains the UI design and get the input / output and display to the end user

Common layer which contains all the common logic like Encryption / Decryption etc..

The DB layer which contains the Store Procedure details, Functions, Views, etc..

So the layers are separated here based on the usage but the single tier architecture which contains all the business logic data base query all the codes are maintain single layer itself so it’s is difficult to maintain the code.

Security

Compare with single tier architecture the N-Tier is more secured one here the data base oriented things are maintained DB itself say for Example in SQL server database all the store procedures, functions, views are managed in DB itself. Sometimes the DB and code will be maintained separate server so it will be more secured.

Reusability

If we develop the application using N-Tier Architecture then we can reuse it.

Below are the one example for reusability.

Example:

The presentation layer has been designed using ASP.net and the Data Access layer is designed using ADO.net the DB is Sql server. Now the client wants to change the presentation layer as MVC or they want to change the DB as oracle so now no need to develop the application from the scratch we just reuse the business logics here the DB / Presentation layer only will be changed based on the client requirement