Chapter-5 Active Server Page (ASP)

- ASP stands for Active Server Pages.
- Commonly known as Classic ASP or ASP Classic.
- Microsoft developed this server side scripting environment (Microsoft's first server side scripting engine), specifically for dynamically generated web pages.
- The default scripting language used for writing ASP is VBScript, although you can use other scripting languages like JScript (Microsoft's version of JavaScript).
- Any web pages containing ASP cannot be run by just simply opening the page in a web browser. The page must be requested through a web server that supports ASP, this is why ASP stands for Active Server Pages, no server, no active pages
- While the initial release classic ASP was an add-on to the Internet Information Services (IIS) component of Windows NT 4.0, it was later incorporated into the Windows Server operating system.
- ASP was Microsoft' alternative to Common Gateway Interface (CGI) scripts and Java Server Pages (JSPs), both intended to allow clients to interact with server-side databases and enterprise services.
- Originally released as part of Windows NT 4.0 Option Pack, it is now superseded by ASP.NET, another product by Microsoft.
- Though, ASP.NET is not strictly an enhanced version of ASP; the two technologies have completely different underlying implementations. ASP.NET is a compiled language and relies on the .NET Framework, while ASP is strictly an interpreted language
- Mainly ASP.NET used to run and create dynamic Web server applications, it allows one to combine HTML pages, script commands, and COM components. It makes it easier to modify and develop Web applications.

What is ASP.Net?

- ASP.Net is a web development platform provided by Microsoft.
- It is used for creating web-based applications.
- ASP.Net was first released in the year 2002.
- ASP stands for Active Server Pages, and .NET is Network Enabled Technologies.

ASP.NET Architecture and its Components

❖ ASP.Net is a framework which is used to develop a Web-based application.

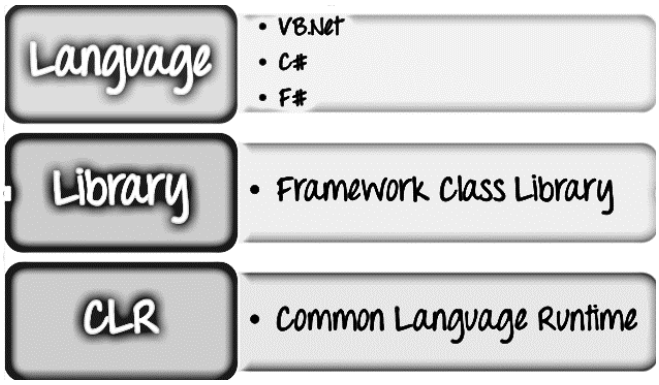


Figure: basic architecture of ASP.NET

The architecture of the.Net framework is based on the following key components

1. **Language**

   - A variety of languages exists for .net framework.
   - They are VB.net, F#, C# etc.
   - Languages can be used to develop web applications.

2. **Library**

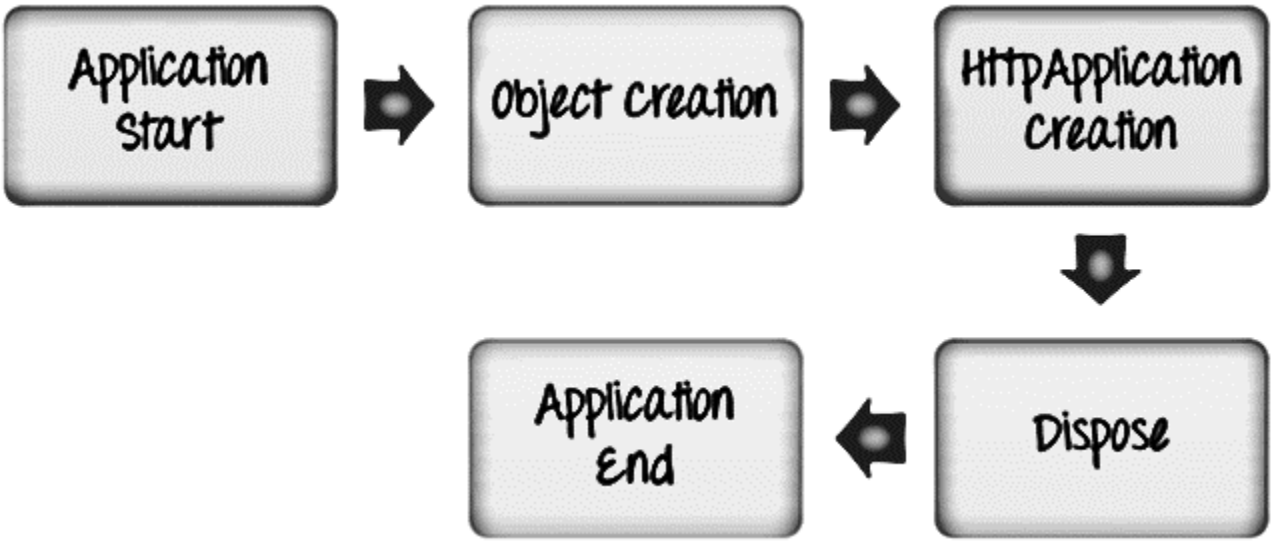
   - The .NET Framework includes a set of standard class libraries.
   - The most common library used for web applications in .net is the Web library.
   - The web library has all the necessary components used to develop.Net web-based applications.

3. **Common Language Runtime**

   - The CLR is used for performing key activities. Activities include Exception handling and Garbage collection.

ASP.Net Lifecycle

- When an ASP.Net application is launched, there are series of steps which are carried out. These series of steps make up the lifecycle of the application.



ASP.Net Lifecycle

Application Start:

The life cycle of an ASP.NET application starts when a request is made by a user. This request is to the Web server for the ASP.Net Application. This happens when the first user normally goes to the home page for the application for the first time. During this time, there is a method called Application_start which is executed by the web server. Usually, in this method, all global variables are set to their default values.

Object creation

The next stage is the creation of the HttpContext, HttpRequest & HttpResponse by the web server. The HttpContext is just the container for the HttpRequest and HttpResponse objects. The HttpRequest object contains information about the current request, including cookies and browser information. The HttpResponse object contains the response that is sent to the client.

HttpApplication creation

This object is created by the web server. It is this object that is used to process each subsequent request sent to the application.

For example, let's assume we have two web applications. One is a Daraz, and the other is a onlinekhabar. For each application, we would have 2 HttpApplication objects created. Any further requests to each website would be processed by each HttpApplication respectively.

Dispose

This event is called before the application instance is destroyed. During this time, one can use this method to manually release any unmanaged resources.

Application End

This is the final part of the application. In this stage, the application is finally unloaded from memory.

What is ASP.Net Page Lifecycle

- When an ASP.Net page is called, it goes through a particular lifecycle.
- This is done before the response is sent to the user.
- There are series of steps which are followed for the processing of an ASP.Net page.

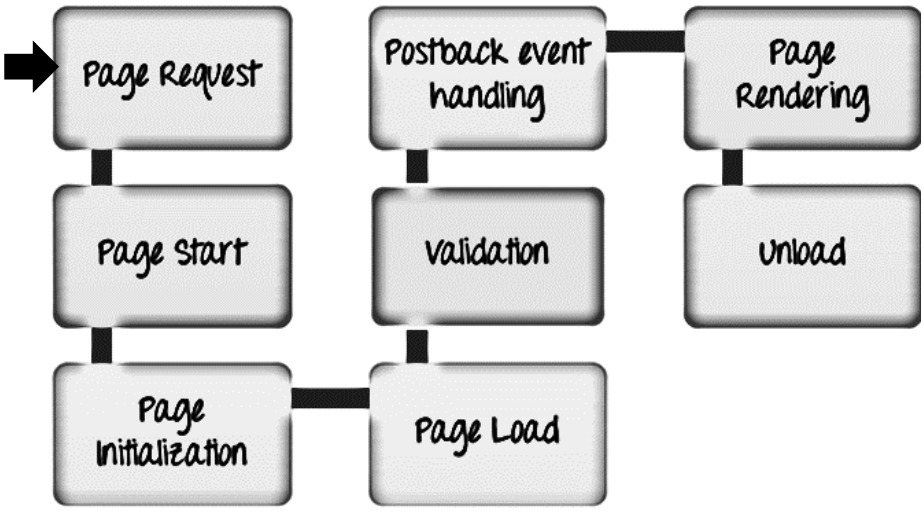


Figure: ASP.Net Page Lifecycle

Page Request

- This is when the page is first requested from the server.
- When the page is requested, the server checks if it is requested for the first time.
- If so, then it needs to compile the page, parse the response and send it across to the user.
- If it is not the first time the page is requested, the cache is checked to see if the page output exists. If so, that response is sent to the user.

Page Start

- During this time, two objects, known as the **Request** and **Response** object are created.
- The Request object is used to hold all the information which was sent when the page was requested.
- The Response object is used to hold the information which is sent back to the user.

Page Initialization

- During this time, all the controls on a web page is initialized.
- So if you have any label, textbox or any other controls on the web form, they are all initialized.

Page Load

- This is when the page is actually loaded with all the default values.
- So if a textbox is supposed to have a default value, that value is loaded during the page load time.

Validation

- Sometimes there can be some validation set on the form.
- For example, there can be a validation which says that a list box should have a certain set of values.
- If the condition is false, then there should be an error in loading the page.

PostBack event handling

- This event is triggered if the same page is being loaded again.
- This happens in response to an earlier event.
- Sometimes there can be a situation that a user clicks on a submit button on the page. In this case, the same page is displayed again. In such a case, the Postback event handler is called.

Page Rendering

- This happens just before all the response information is sent to the user.
- All the information on the form is saved, and the result is sent to the user as a complete web page.

Unload

- Once the page output is sent to the user, there is no need to keep the ASP.net web form objects in memory.
- So the unloading process involves removing all unwanted objects from memory.

Difference between ASP and ASP.NET

ASP	ASP .NET
1. ASP is the interpreted languages.	ASP.NET is the compiled language.
2. ASP uses **ADO (ActiveX Data Object)** technology to connect and work with database.	ASP.NET uses **ADO.NET** to connect and work with database.
3. ASP is partially object oriented.	ASP.NET is fully object oriented.
4. ASP Pages have the file extension **.asp**.	ASP.NET Pages have the file extension **.aspx**
5.ASP doesn't have the concept of inheritance	ASP.NET use full-fledged programming language
6.ASP pages use scripting language	ASP.NET use full-fledged programming language.
7.Error handling is very poor in ASP	Error handling is very good in ASP.NET

8.ASP has maximum four in-built classes i.e. Request, Response, Session and Application.	ASP.NET has more than 2000 in-built classes

ASP .Net Controls

- Controls are small building blocks of the graphical user interface, which include text boxes, buttons, check boxes, list boxes, labels, and numerous other tools.
- Using these tools, the users can enter data, make selections and indicate their preferences.
- Controls are also used for structural jobs, like validation, data access, security, creating master pages, and data manipulation.

ASP.NET uses **five types of web controls**

1. HTML controls
2. HTML Server controls
3. ASP.NET Server controls
4. ASP.NET Ajax Server controls
5. User controls and custom controls

ASP.NET **server controls** are the primary controls used in ASP.NET. These controls can be grouped into the following categories:

1. **Validation controls** - These are used to validate user input and they work by running client-side script.
2. **Data source controls** - These controls provides data binding to different data sources.
3. **Data view controls** - These are various lists and tables, which can bind to data from data sources for displaying.
4. **Personalization controls** - These are used for personalization of a page according to the user preferences, based on user information.
5. **Login and security controls** - These controls provide user authentication.
6. **Master pages** - These controls provide consistent layout and interface throughout the application.
7. **Navigation controls** - These controls help in navigation. For example, menus, tree view etc.

8. **Rich controls** - These controls implement special features. For example, AdRotator, FileUpload, and Calendar control.

The syntax for using server controls is:

```
<asp:controlType  ID ="ControlID" runat="server" Property1=value1
[Property2=value2] />
```

Database Connection in ASP .NET

- ASP.Net has the ability to work with different types of databases (majority of databases).
- It can work with the most common databases such as **Oracle** and **Microsoft SQL Server**.
- It also has the ability to work with new forms of databases such as **MongoDB** and **MySQL.**

Let's established the database connection by using Microsoft SQL Server as database.

While working with databases, the following concepts which are common across all databases.

1. **Connection**

   - It is first step to work with database connection. The connection to a database normally consists of following parameters.

     - **Database name or Data Source**
       - Each connection can only work with one database at a time.
     - **Credentials**
       - Credentials are used to establish a connection to the database.
       - username and password of SSMS
     - **Optional parameters**
       - Optional parameters specifies on how .net should handle the connection to the database.

- For example, one can specify a parameter for how long the connection should stay active.

2. **Selecting data from the database**

   - ❖ Once the connection is established, data is fetched from the database.
   - ❖ ASP.Net has the ability to execute 'sql' select command against the database.
   - ❖ The 'sql' statement can be used to fetch data from a specific table in the database.

3. **Inserting data into the database**

   - ❖ ASP.Net is used to insert records into the database. Values for each row that needs to be inserted in the database are specified in ASP.Net.

4. **Updating data into the database**

   - ❖ ASP.Net can also be used to update existing records into the database.
   - ❖ New values can be specified in ASP.Net for each row that needs to be updated into the database.

5. **Deleting data from a database**

   - ❖ ASP.Net can also be used to delete records from the database. The code is written to delete a particular row from the database.

Code to connect database:

i. Create a windows form project
ii. Make a button and named is as 'ClickToConnect'
iii. Write a code inside button click event as follows

```
string connetionString;
            SqlConnection con;
            connetionString = @"Data Source=Muku;Initial Catalog=db_EEC;User
ID=sa;Password=p@ssw0rd";
            con = new SqlConnection(connetionString);
            con.Open();

MessageBox.Show("Database Connected Successfully. Congratulations..!");
            con.Close();
```

Access data with the SqlDataReader

To showcase how data can be accessed using C#, let us assume that we have the following artifacts in our database.

- ✓ A table called EecInfo. This table will be used to store the ID, name and address of EEC.
- ✓ The table will have 2 columns, one called "name" and the other called "address".
- ✓ For now, the table will have 1 rows

    - ❖ Construct “select" statement, which will be used to read the data from the database then execute the "select" statement against the database and fetch all the table rows accordingly.
    - ❖ The DataReader object is used to get all the data specified by the SQL query. We can then read all the table rows one by one using the data reader.
    - ❖ Finally, we will just display the output to the user and close all the objects related to the database operation.

```
SqlCommand command;
            SqlDataReader datareader;
            string sql, output = "";
            sql = "SELECT *FROM EecInfo";
            command = new SqlCommand(sql, con);
            datareader = command.ExecuteReader();
            while (datareader.Read())
            {
                output = output + datareader.GetValue(0) + "-->" + datareader.GetValue(1)
+ "-->" + datareader.GetValue(2) + "\n";
            }

            MessageBox.Show(output);
            datareader.Close();
            command.Dispose();

            cnn.Close();
```

**End of Chapter **