SHORT QUESTIONS

1. What is ASP.NET?

- ASP.NET means "Active Server Pages" which is frontend language & it is established & supported by Microsoft

2. Explain objective of ASP.NET

- ASP.NET is a web application which allows the programmer to build dynamic web sites, Web applications & web services.
- ASP.NET allows to develop web applications using a full featured programming language such as C#, VB.NET, Jscript .

3.Explain compile code.

- When code is compiled, it is first converted into Microsoft Independent Language (MSIL). Then this independent language is further converted into Machine Specific instructions.

4. Why is ASP.NET event driven.

- When something specific has occurred in significance for system hardware or software is call an Event.
- A segment of code that is executed in response to an event is called an event handler.

5. Explain ISPOSTBACK

- IsPostBack is a propety of the Asp.Net page that tells whether or not the page is on its initial load or if a user has perform a button on your web page that has caused the page to post back to itself.

6. What is the use of PostBackUrl Property?

- The PostBackUrl property can be useful for scenarios where you want to collect data from multiple pages and process it on a final page.

7. What is the use of web.sitemap file?

- The SiteMapPath control is also used to display the Navigation information on the site

- It display the current page's context within the entire structure of a website
- You can bind this control to TreeView and Menu controls

8. What is APP_CODE and APP_DATA application folders.

- ASP.NET compiles the code in the app_code folder on the initial request to your application and recompiles it when any changes are detected.
- The app_data folder contains application data files, such as .mdf database files, XML files, and other data store files.

9. What is the use of Image Map control?

- It is used to show image with in the page.
- An Image Map control is an ASP.NET web control that allows you to create an image with defined hot spot regions.
- For example, you can use this control to display a map of a geographical region and link each region to a different page.

10. What is inline code and code behind page?

- ASP.NET 2.0 provides two paths for structuring the cod of ASP.NET pages: Code Behind and Code Inline
- Inline Coding :- Inline Code is embedded directly within the ASP.NET page that has an extension of .aspx.
- Code Behind :- code is separated from content completely and stored in a separate file.

11. What are the advantage of server side state management?

- Server side state management is a technique of storing data on the server instead of the client's browser.

12. Explain @ Register directive.

- The @ Register directive is a directive used to register user-defined controls on a web page.
- <code> <%@ Register tagprefix="my" namespace="MyNamespace" %> <my:CustomControl runat=server /> </code>

13. What is importance of Themes?

- Themes are a feature of ASP.NET that help you maintain a consistent look and feel across your web pages and controls. Themes can also improve the performance and security of your web application by reducing the bandwidth and hiding the confidential state data from the client

14. What is SOAP?

- SOAP is a protocol for accessing web services over HTTP. It stands for Simple Object Access Protocol and it is based on XML

15. What is WSDL?

- WSDL stands for Web Services Description Language. It is an XML-based language for describing web services. A WSDL file is written in XML.

16. What is the use of global.asax file in ASP.NET?

- There are 2 types of events available in ASP.NET. We can handle both sessions in a global.asax file.

- Session_Start(): When the new session is initialized then the session_start event is invoked.
- Session_end(): When the session is expires then the Session_End event is invoked.

17. Explain Server.Transfer().

- It sends (transfers) all the information created in one ASP file to a second ASP file.
- Server.Transfer() has some advantages and disadvantages compared to Response.Redirect(), which is another method to navigate to a different page.

18. What is the importance of Query String?

- Query string is a way to pass some information from one page to another in ASP.NET. It is useful when you want to transfer a value or a parameter to a web application or a database

19. List out Data Adapter's methods and properties.

- The DataAdapter has properties of type Command, which represent the ways it can query, insert, delete, and update the database.

20. Why we use the app_code and app_data folder?

- ASP.NET compiles the code in the app_code folder on the initial request to your application and recompiles it when any changes are detected.
- The app_data folder contains application data files, such as .mdf database files, XML files, and other data store files.

21. What are advantages of validator control?

- Validation allows you to validate an input of the user.
- ASP.NET validation controls validate the user input data to ensure that useless, unauthenticated, or contradictory data don't get stored
- ASP.NET provides the following validation controls :-
  1. RequiredFieldValidator
  2. RangeValidator
  3. CompareValidator
  4. RegularExpressionValidatator
  5. CustomValidator
  6. ValidationSummary

22. What is the use of System.Data.Oledb?

- To use Microsoft Access as a database, System.Data.oleDb namespace is used. It provides classes which can work with OLE-DB data sources using the .NET OleDb data provider.
- For Example:- OleDbConnection and OleDbCommand.

23. How CSS is applied to the ASP.NET page?

- Inline style: This is the simplest way to apply CSS to an ASP.NET page. You can use the style attribute of any HTML or ASP.NET element to specify the CSS properties and values for that element.
- Internal style sheet: This is a way to apply CSS to an ASP.NET page by using the <style> tag inside the <head> section of the page. You can use selectors to target specific elements or groups of elements and define the CSS rules for them

- External style sheet: This is a way to apply CSS to an ASP.NET page by using a separate file that contains the CSS code. You can use the <link> tag inside the <head> section of the page to link to the external style sheet file. You can also use the @import rule inside a <style> tag to import an external style sheet file.

24. List data bind controls in ASP.NET. Write its usage.

- The binding of data with the controls of Web Forms is known as Data Binding. Data binding is the ability to bind some elements of a data source with the controls of an application.
- On the basis of the number of bound values that can be displayed through a control of a Web Form, It can be divided in two types:
- 1)Simple data binding. 2)Complex data binding.

25. Explain Querystring. How can you make and use of QueryString?

- A query string is use to send data from one webform to another through the URL. A query string consists of two parts,
- 1. field and 2. value, and each of pair separated by ampersand (&) symbol. The ?(question mark) symbol in a query string indicates the beginning of a query string and it's value. There is a limit on the Query string length. Hence, Query strings cannot be used to send more than 100kb data
- Request.QueryString(variable)[lindex)|.Count]

26. What is the difference between page and Web user control?

- A page and a Web user control are both components that can be used to create dynamic web applications using ASP.NET. However, they have some differences in their structure, functionality, and usage.
- A page is a file that has the extension .aspx and contains an @ Page directive that defines its configuration and other properties
- A Web user control is a file that has the extension .ascx and contains an @ Control directive that defines its configuration and other properties

27. How structured exception handling is implemented?

- Try-catch block :- This is the most common way of handling exceptions. You can use the try, catch, and finally keywords to enclose the code that might throw an exception, and provide the logic to handle it or clean up resources

28. How Asp.net maintain View State?

- View State provides page level client side state management which is as long as the user is active on the current page, state is available and as the user redirects to the next page and the current page state is lost.
- By default View state is enabled for all server side controls of ASP.NET with a property called EnableviewState set to true.

29. Use of custom validator.

- The CustomValidator control allows writing application specific custom validation routines for both the client side and the server side validation
- The client side validation is accomplished through the ClientValidation Function property

30. What is SiteMapPath Control?

- The SiteMapPath control is also used to display the Navigation information on the site. It display the current page's context within the entire structure of a website.
- You can bind this control to TreeView and Menu controls

31. Differentiate dataset and datareader.

- The key components of DataSet are :-
  1. DataTableCollection
  2. DataRelationCollection
  3. DataTable
  4. DataRowCollection
  5. DataColumnCollection

32. What is UDDI in webservice?

- UDDI registries can be public or private. Public registries are accessible to anyone on the Internet and provide a global directory of web services

33.What is session? how to increase session time [Note: default session time is 20 MIN

- "The time duration for which the user interacts with the web application is called session"

<sessionState mode"InProc"

stateConnectionString="tcpip=127.0.0.1:42424"

sqlConnectionString="Data Source=\sQLEXPRESS;
Trusted_Connection=Yes;" cookieless="false" timeout="20"/>

34. How theme set at page level and application level.

- To set a theme at the page level, you can use the Theme or StyleSheetTheme attribute of the @ Page directive in the .aspx file. For example:- <%@ Page Language="C#" Theme="FirstTheme" %>

35. How to use structured error.

- Structured error handling is a way of handling errors and exceptions in your ASP.NET code using the try-catch-finally blocks.
- To use structured error handling, you need to enclose the code that might throw an exception in a try block, followed by one or more catch blocks that handle different types of exceptions

36. What is caching in short

- Caching is a technique of storing frequently used information of user in memory, so that, when the same information is needed by user next time, it could be directly retrieved from the memory instead of being generated by the application.

37. Write an extention of:

1. WEB FORM :- .ASPX
2. USER CONTROL :- .ASCX
3. WEB SERVICE :- ASMX
4. GLOBAL CLASS FILE :- .ASAX
5. VB CODE FILE :- .VB
6. SKIN FILE :- .SKIN
7. CSS FILE :- .CSS

LONG QUESTIONS

1. Write a note on .NET Framework.

➢ **.NET Framework:**

- NET is Software Platform. It is a language neutral environment for developing applications. It also provides an execution environment and integration with various programming languages for building, deploying and running web-based and standalone enterprise applications. All aspects to manage the program execution like memory allocation for data storage and instructions, granting and denying permission to the application, managing execution of application and reallocation of memory for resources which are not needed is managed easily in ASP.NET. It consists of class libraries and reusable components.

The block diagram of .NET framework is as follows:

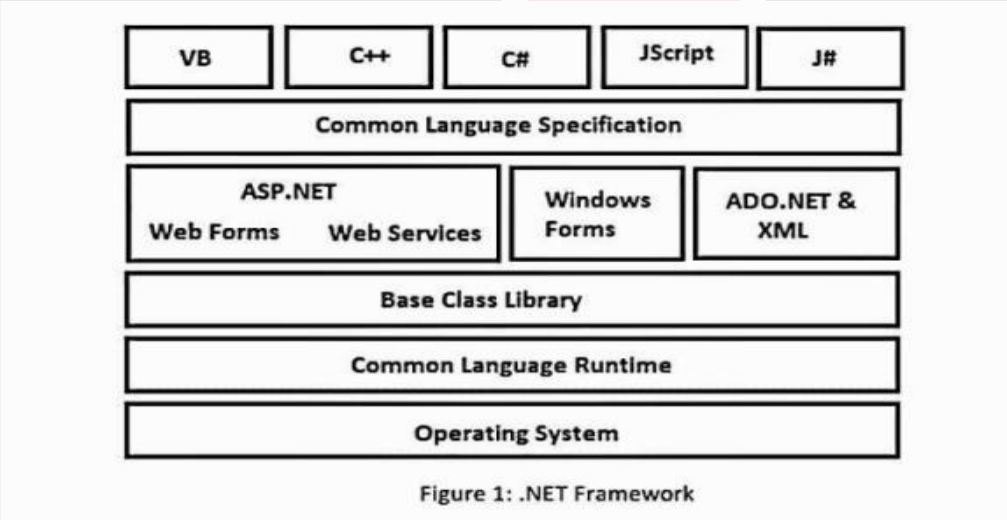


Figure 1: .NET Framework

Common Language Specification (CLS):

- CLS is the collection of rules and constraints that every language compatible to .NET must follow. It is one of the core components of the .Net Framework which helps to build communication between different components written in different programming language and reuse the common logics

Microsoft has defined three level of CLS compatibility as follow:

- **Compliant Producer**: The component developed in this type of language can be used by any other language.
- **Consumer**: In this category, the language can use classes produced in any other language.
- **Extender**: In this category, languages cannot use classes as in CONSUMER category but can also extend classes using inheritance

- **Base Class Library (BCL):**
  .Net framework consists of classes, interfaces and value types that help in speeding up the development process and provide access to the system functionality. The class library is a collection of methods & functions that can be reusable types meant to be used by managed code.
  The namespaces are logically defined by their functionality. Most of the methods are split into either System. or Microsoft." Namespaces
  For example:- The System.Data namespace contains the functionality available for accessing database.

Common Language Runtime (CLR):

- It is the runtime environments which do both compile and run the application. So it is also known as the heart of .NET framework.
- MSIL which is language independent code, so CLR uses this code for the execution of the application.
- The MSIL code is translated by JIT (Just-in Time) compliler.

Microsoft Intermediate Language (MSIL):

- it is also known as Intermediate Language (1L) or Common Intermediate Language (CIL). A.NET programming language does not compile the code into executable code; instead it compiles the code into an intermediate code called Microsoft Intermediate Language (MSIL) or Intermediate Language (L) or Common Intermediate Language (CIL).
- This IL or MSIL or CiIL code is machine independent code which is then send to CLR which converts this machine independent code into native code with the help of JIT (Just-in Time Compiler) avalable in CLR

2. Explain page life cycle in detail.

➢ **Page Life Cycle:**

- Whenever you request an ASP.NET page, a particular set of events is raised in particular sequence. This sequence of events is called the page execution lifecycle.

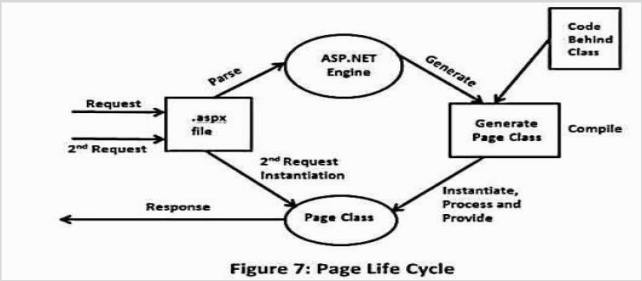


Figure 7: Page Life Cycle

When the request comes from the .aspx page, it is given for parsing to asp.net engine. Then a combined page class is generated from that parsed page and the code behind class and then it is compiled. At last, the instance of the class is created and provided to the user.

Page Life Cycle Stages:

1. **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.
2. **Start**: In this phase, 2 objects, known as the Request and Response object are set. If the request is an old request or post back, the IsPostBack property of the page is set to true. The UICulture property of the page is also set.
3. **Page Initialization**: In this phase, controls on the page are available and each control's UniquelD property is set. A master page and themes are also applied to the page if applicable
4. **Load**: During this phase, if the current request is a postback, control properties are set to utilize the view state and control state values like if a textbox is supposed to have a default value, that value is loaded during the page load time.
5. **Validation**: In this phase, when the validation controls are present on the page, than on its successful execution, IsValid property of the page is set to true
6. **Postback Event Handling**: In this phase, control event handlers are called if the request is a postback. That means this event is triggered if the same page is being loaded again. This happens in response to an earlier event.
7. **Rendering**: Before this phase, view state is saved for the page and all controls. During this stage, the page calls the Render() method for each control and writes its output to the OutputStream object of the page's Response property.
8. **Unload**: The unload method takes after the page is fully loaded and is ready to terminate. At this point, rendered page is sent to the client and page properties such as Response and Request are unloaded and clean-up is done.

3. Explain Request object.

- Request object is used to retrieve information from a user. It is an instance of the system.Web.HttpRequest class. Request Object's collections, properties, and methods are described below:

Collections of Request Object

Collection	Description
ClientCertificate	It contains all the field values stored in the client certificate.
Cookies	It contains all the cookie values sent in a HTTP request.
Form	It contains all the form (input) values from a web form that uses the post method.
QueryString	It contains all the variable values in a HTTP query string.
ServerVariables	It contains all the server variable values.

Table 1: Collection of Request Object

Properties of Request Object

Property	Description
ApplicationPath	It gets the virtual application root path on the server.
Browser	It gets or sets information about the requesting client's browser capabilities.
CurrentExecutionFilePath	It requests the current execution path.
FilePath	It gets the virtual path of the current request.
HttpMethod	It gets the HTTP data transfer method (such as GET, POST, or HEAD) used by the client.
TotalBytes	It gets the number of bytes in the current input stream.
UserAgent	It gets the raw user agent string of the client browser.
UserHostAddress	It gets the IP host address of the remote client.

Table 2: Properties of Request Object

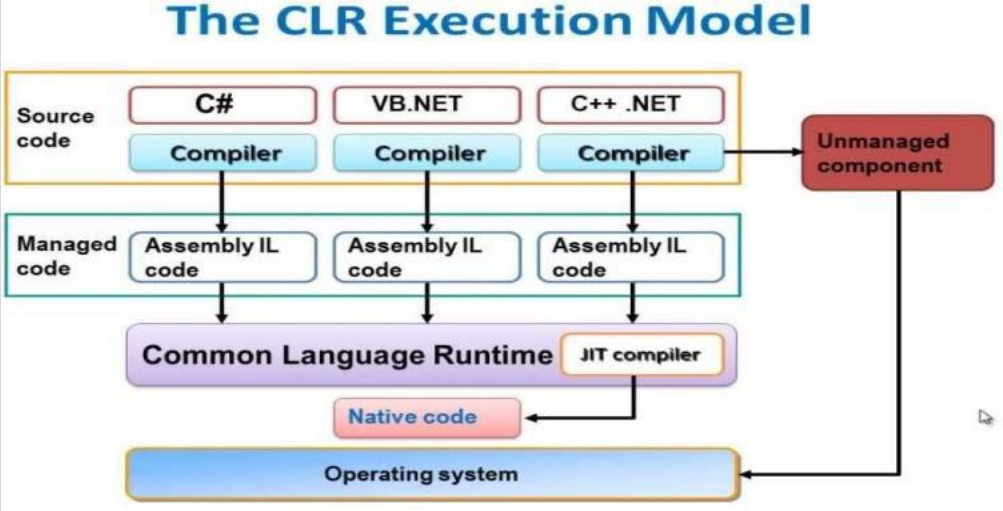
Methods of Request Object

Method	Description
BinaryRead	It performs a binary read of a specified number of bytes from the current input stream.
Equals(Object)	It determines whether the specified object is equal to the current object.
GetType	It gets the Type of the current instance.
MapImageCoordinates	It maps an incoming image-field form parameter to appropriate x-coordinate and y-coordinate values.
MapPath(String)	It maps the specified virtual path to a physical path.
SaveAs	It saves an HTTP request to disk.
ToString	It returns a String that represents the current object.
ValidateInput	It causes validation to occur for the collections accessed through the Cookies, Form, and QueryString properties.

Table 3: Methods of Request Object

4. Write a note on CLR.

- It is the core component of .NET Framework. It is also known as an execution Environment. It is used to compile the IL or MSIL code to native code.



The main function of CLR is to convert the MSIL code to native code and then execute the program. The managed code a code that is developed with a language compilerthat targets the CLR is compiled only when it is needed, that means it converts the appropriate instructions when each functin is called.

- The unmanaged code - a code thaat is developed without considering the conventions and requirements of CLR, is executed with minimal service of CLR. The CLR's JIT (Just-in Time) compilation converts MSIL to native code on demant at application run.
- During the execution of the program, the CLR provides functionaloty such as memory, Debugging, Exception Handling, Security and versioning support to any languages that target it. In short, When .NET programs are executed, the CLR activates the JIT compiler.
- The JIT Compiler converts MSIL into native code on demand basis. Thus the program executes as a native code to run the program fast. That is how NET framework achieves the portability.

5. Write a note on Response object.

- Response object is used to provide output to the user from the server. ASP.NET provides a class called HttpResponse which is defined in the namespace called System.Web. Response object's collections, properties, and methods are described below:

Collections of Response Object

Collection	Description
Cookies	It sets a cookie value. If the cookie does not exist, it will be created, and take the value that is specified.

Table 4: Collection of Response Object

Properties of Response Object

Property	Description
Buffer	It specifies whether to buffer the page output or not.
CacheControl	It sets whether a proxy server can cache the output generated by ASP or
	not.
Charset	It appends the name of a character-set to the content-type header in the response object.
ContentType	It sets the HTTP content type for the response object.
Expires	It sets how long (in minutes) a page will be cached on a browser before it expires.
ExpiresAbsolute	It sets a date and time when a page cached on a browser will expire.
IsClientConnected	It indicates whether the client has disconnected from the server.
Pics	It appends a value to the PICS label response header.
Status	It specifies the value of the status line returned by the server.

Table 5: Properties of Response Object

Methods of Response Object

Method	Description
AddHeader	It adds a new HTTP header and a value to the HTTP response.
AppendToLog	It adds a string to the end of the server log entry.
BinaryWrite	It writes data directly to the output without any character conversion.
Clear	It clears any buffered HTML output.
End	It stops processing a script and returns the current result.
Flush	It sends buffered HTML output immediately.
Redirect	It redirects the user to a different/provided URL.
Write	It writes a specified string to the output.

Table 6: Methods of Response Object

6. Write a note on WEB.CONFIG file.

- A WEB.CONFIG file is a configuration file that is used to control the behavior of an ASP.NET web application. It is an XML file that is stored in the root directory of the application.
- It contains settings that override the default settings in the Machine.config file, which is located in the systemroot\Microsoft.NET\Framework\versionNumber\CONFIG\ folder and applies to all .NET applications on the system.
- The WEB.CONFIG file can have various sections that specify different aspects of the application, such as authentication, authorization, caching, compilation, custom errors, globalization, HTTP modules, HTTP handlers, session state, tracing, and more.
- Each section can have its own attributes and elements that define the configuration options. The WEB.CONFIG file can also reference external files or assemblies that contain additional configuration settings or custom code.
- The WEB.CONFIG file is read by IIS and the ASP.NET Core Module to configure the app hosted with IIS.
- The file must be present at the content root path of the deployed app and must have a format similar to the Machine.config file. The file must also contain the <configuration> element and the <system.web> element as the minimum structure.
- The ASP.NET Core Module is configured with the aspNetCore section of the system.webServer node in the WEB.CONFIG file.
- The WEB.CONFIG file can be created and modified by using a text editor such as Notepad. There is no Microsoft-provided administration tool for this purpose.
- However, some third-party tools or IDEs may offer graphical interfaces or wizards to create or edit the WEB.CONFIG file.

7. Explain Adrotator Control.

- An AdRotator control is a web server control that allows you to display a sequence of advertisement images on your web page. The images are randomly selected from a list that is specified in an external XML file, called the advertisement file.
- The advertisement file contains information about each image, such as the URL, the alternate text, the impressions, the keyword, and the dimensions. You can also include custom attributes in the advertisement file for your own purposes.

- The AdRotator control has two main properties: AdvertisementFile and Target. The AdvertisementFile property sets the path of the XML file that contains the list of images. The Target property sets the type of window that will open when the user clicks on an image. You can use values such as "_blank", "_self", "_parent", or "_top" for this property.
- The AdRotator control is useful for creating dynamic and attractive web pages that can generate revenue from advertising. You can use the AdRotator control to display different ads based on the user's preferences, location, or other criteria.
- You can also track the performance of your ads by using the impressions attribute or by using custom code.

8. Explain calendar control and hidden control.

➢ Calendar control and hidden control are two types of web server controls that are used in ASP.NET applications. They have different purposes and functionalities, as explained below:

- **Calendar control**: This control allows you to display a calendar on your web page and let the user select a date or a range of dates. You can customize the appearance and behavior of the calendar control by using its various properties and events. For example, you can set the caption, the format, the style, the selection mode, and the navigation elements of the calendar control. You can also handle events such as SelectionChanged, DayRendered, and VisibleMonthChanged to perform some actions when the user interacts with the calendar control. The basic syntax of a calendar control is:

```
<asp:Calender ID = "Calendar1" runat = "server">
</asp:Calender>
```

- **Hidden control**: This control allows you to store a value on your web page that is not visible to the user. You can use this value for passing data between pages or for storing some information that you do not want to expose to the user. The hidden control is rendered as an HTML input element of type hidden. The basic syntax of a hidden control is:

```
<asp:HiddenField ID = "HiddenField1" runat = "server" Value =
"some value"></asp:HiddenField>
```

9. Master page and nestedmaster page.

➢ Master page: It allows you to create a consistent layout for the pages in your application. Following are the steps to embed master page in your application

- Right click on solution explorer of your website.
- Select Add new item.
- Select Master Page option from Add new item Dialog Box.

- Change name of Master Page.
- Click on Add button.
- Add desired controls on the master page that you want on all the pages. Drag and Drop HTML Table and then keep all controls. E.g Menu control from Navigation, Image, Heading, Footer etc.
- Perform necessary Design steps and save the Master Page
- Nested master pages are a feature of ASP.NET that allow you to create master pages that inherit from other master pages. This way, you can reuse common layout and functionality across different master pages. For example, you can have a main master page that defines the header, footer, and navigation of your site, and then have nested master pages that define the content area for different sections of your site.

- To create a nested master page, you need to set the MasterPageFile attribute of the @Master directive to point to the parent master page. For example, if you have a parent master page called MainMaster.master and a nested master page called NestedMaster.master, you can use this code in NestedMaster.master:
  <%@ Master Language="C#" MasterPageFile="~/MainMaster.master" AutoEventWireup="true" CodeFile="NestedMaster.master.cs" Inherits="NestedMaster" %>
  you can use this code in NestedMaster.master:
  <asp:Content ID="Content1" ContentPlaceHolderID="MainContent" Runat="Server">
  <asp:ContentPlaceHolder ID="NestedContent" runat="server">
  </asp:ContentPlaceHolder>
  </asp:Content>

10.Expain HTML control.

- The HTML server controls are basically the standard HTML controls enhanced to enable server side processing.
- The HTML controls such as the header tags, anchor tags, and input elements are not processed by the server but are sent to the browser for display
- They are specifically converted to a server control by adding the attribute runat="server" and adding an id attribute to make them available for server-side processing.
- For example, consider the HTML input control: <input type="text" size="40"> It could be converted to a server control, by adding the runat

and id attribute: < input type="text" Id="testtext" size="40" runat="server">

- **Advantages of using HTML Server Controls**: Although ASP.NET server controls can perform every job accomplished by the HTML server controls, the later controls are useful in the following cases:
- Using static tables for layout purposes.
- Converting a HTML page to run under ASP.NET

11.Explain gridview control.

- The GridView control enables us to display, select, sort, page, and edit data items such as database records.
- The GridView control in ASP.NET is a powerful and flexible data-bound control used to display tabular data from various data sources, such as databases, XML, or object collections. It is commonly used in web applications to create data grids or tables with features like sorting, paging, and editing. Here's a detailed explanation of the GridView control:

Key Features and Properties of GridView Control:

1. Data Binding: You can bind the GridView to a data source using the `DataSource` property. This can be a database query, an object collection, or other data sources like XML or JSON.

2. Auto-Generation of Columns: The GridView can automatically generate columns based on the data source schema, which simplifies the setup process. You can also define custom columns manually.

3. Sorting: GridView provides built-in sorting functionality. Users can click on column headers to sort the data in ascending or descending order.

4. Paging: You can enable paging to split large datasets into manageable pages. This is crucial for performance when dealing with extensive data.

5. Editing and Updating: GridView supports inline editing of data rows. You can specify which columns are editable and handle events like `RowEditing` and `RowUpdating`.

6. Deleting and Inserting: It allows users to delete rows and insert new rows. You can handle events like `RowDeleting` and `RowInserting` for custom logic.

7. Templates: GridView supports templates for customizing the appearance and layout of data rows, headers, footers, and more. You can use templates to display data in a more user-friendly way.

8. Data Formatting: You can format data within columns using various formatting options, making it suitable for displaying dates, currency, or custom text.

9. Data Source Controls: GridView works well with data source controls like SqlDataSource and ObjectDataSource, simplifying data retrieval and management.

10. Client-Side Integration: It can be integrated with client-side scripts and AJAX for enhanced user interactions without full-page postbacks.

Here's a simplified example of using a GridView control in ASP.NET:

html

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="true">
</asp:GridView>
```

In this basic example, the GridView is configured to auto-generate columns based on the data source. In your code-behind, you would set the `DataSource` property and call the `DataBind()` method to populate the GridView with data.

The GridView control is highly customizable and can be extended to meet specific requirements in web applications, making it a fundamental tool for displaying and interacting with tabular data.

12.Explain server object in detail.

- The ASP.NET Server object is used to access properties and methods of the server. Server object's properties and methods are described below:

Properties of Server Object

Property	Description
ScriptTimeout	It sets or returns the maximum number of seconds a script can run before it gets terminated.

Table 7: Properties of Server Object

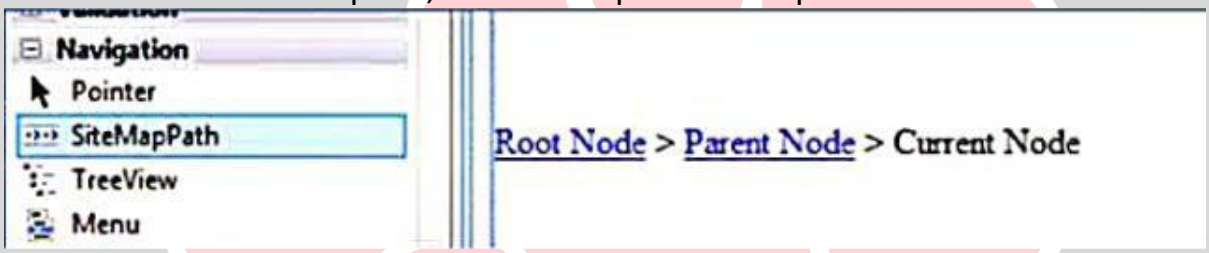
Methods of Server Object

Method	Description
CreateObject	It creates an instance of an object.
Execute	It executes an ASP file from inside another ASP file.
GetLastError()	It returns an ASPError object that describes the error condition occurred.
HTMLEncode	It applies HTML encoding to a specified string.
MapPath	It maps a specified path to a physical path.
Transfer	It sends (transfers) all the information created in one ASP file to a second ASP file.
URLEncode	It applies URL encoding rules to a specified string.

Table 8: Methods of Server Object

13.Write a note on SITEMAPPATH.

- The SiteMapPath control is also used to display the Navigation information on the site.
- It display the current page's context within the entire structure of a website. You can bind this control to TreeView and Menu controls.
- SiteMapPath is a web server control that is used to display a set of text or image hyperlinks that enable users to more easily navigate a web site, while taking a minimal amount of page space.
- It reflects the data provided by the SiteMap object and shows the current position of the user within the site hierarchy. It also allows the user to go back to any previous page in the navigation path.
- To add SiteMapPath to your web page, you need to use the asp:SiteMapPath tag and set its properties and styles according to your preferences.
- You can also use templates to customize the appearance and behavior of each node type, such as RootNodeTemplate, CurrentNodeTemplate, ParentNodeTemplate, and PathSeparatorTemplate .



14.Short note:

1. **USER CONROL**
2. **USER PROFILE**

- **USER CONROL:**
  - User control in ASP.NET is a way to create reusable components that can be embedded in web pages. User control is a file with .ascx extension that contains HTML and server-side code.
  - User control can have properties, methods, and events, just like a web form or a custom server control. User control can also contain other ASP.NET controls, such as text boxes, labels, buttons, and so on.
  - To create a user control, you need to use Visual Studio or any text editor that supports ASP.NET. You can design the user control visually or write the code manually. You can also use code-behind files to separate the presentation logic from the business logic. You can test the user control by adding it to a web form and running the application.

- User control is useful for creating common UI elements that can be shared across multiple pages or applications.
- For example, you can create a user control for a contact form, a navigation menu, a header, or a footer. You can then include the user control in any page that needs it by using the asp:UserControl tag or the @Register directive.
- You can also customize the user control by passing parameters or setting attributes.

➢ **USER PROFILE:**

- User profile in ASP.NET is a feature that allows you to store and retrieve user-specific information for your web application. You can use user profile to customize the user experience, such as remembering their preferences, settings, or personal data. User profile can also be used to track the user's activity, such as the last visit time, the number of visits, or the pages viewed.

  To use user profile in ASP.NET, you need to do the following steps:
- Configure the profile provider in the web.config file. The profile provider is responsible for storing and retrieving the user profile data from a persistent source, such as a database or a file system. You can use the default SqlProfileProvider or create your own custom profile provider.
- Define the user profile properties in the web.config file. The user profile properties are the attributes that you want to store for each user, such as name, email, age, gender, etc. You can specify the data type, default value, and serialization mode for each property. You can also group the properties into profiles for different types of users.
- Access and modify the user profile data in your code. You can use the Profile object to get or set the values of the user profile properties. The Profile object is available in any page or class that inherits from Page or UserControl. You can also use the ProfileManager class to perform bulk operations on user profiles, such as deleting, migrating, or finding profiles.

15.Explain checkboxlist control.

➢ A checkboxlist control is a web server control that allows you to display a list of items with checkboxes that the user can select or clear. The checkboxlist control is useful for creating multiple-choice questions, surveys, or preferences. The checkboxlist control has the following features and properties:

- It can be bound to a data source, such as a database, an XML file, or an array, to dynamically generate the list items.

- It has an Items collection that contains the ListItem objects that represent each item in the list. You can access and modify the items by using the collection methods and properties, such as Add, Remove, Count, and Clear.
- It has a SelectedIndex property that returns the index of the first selected item in the list, or -1 if no item is selected. You can also use the SelectedIndices property to get a collection of all the selected indices in the list.
- It has a SelectedItem property that returns the first selected item in the list, or null if no item is selected. You can also use the SelectedItems property to get a collection of all the selected items in the list.
- It has a SelectedValue property that returns the value of the first selected item in the list, or an empty string if no item is selected. You can also use the SelectedValues property to get a collection of all the selected values in the list.

16.Explain OOP concept.

- OOP stands for Object-Oriented Programming, which is a programming paradigm that organizes data and behavior into reusable units called objects.
  - Objects have properties that describe their state and methods that define their actions. Objects can also interact with each other through messages or events.
  - ASP.NET is a web development framework that supports OOP concepts in various ways. For example, ASP.NET web pages are derived from the Page class, which is an object that has properties and methods for handling requests, rendering output, and managing the page life cycle.
  - ASP.NET web controls are also objects that inherit from the Control class, which provides common functionality for rendering HTML elements, handling events, and maintaining state.
  - ASP.NET also allows developers to create their own custom classes and objects to implement business logic, data access, or presentation logic.
  - Developers can use languages such as C#, VB.NET, or JScript.NET to write OOP code in ASP.NET applications.
  - These languages support OOP features such as abstraction, encapsulation, inheritance, and polymorphism, which enable developers to create modular, reusable, and maintainable code.

17.Explain REPEATER AND LISTBOX.

- **REPEATER**: A REPEATER in ASP.NET is a web server control that allows you to display a repeated list of items that are bound to the control. The REPEATER control can be bound to a data source, such as a database, an XML file, or

another list of items. The REPEATER control is useful for creating custom layouts and formats for displaying data, such as tables, lists, or grids.

The REPEATER control has five templates that define how the data is rendered:

- HeaderTemplate: This template is used for elements that you want to render once before the item section.
- FooterTemplate: This template is used for elements that you want to render once after the item section.
- ItemTemplate: This template is used for elements that are rendered once per item in the data source. It is the main template that displays the data.
- AlternatingItemTemplate: This template is used for elements that are rendered every second item in the data source. This allows you to alternate the appearance of the items.
- SeparatorTemplate: This template is used for elements that are rendered between each item, such as line breaks or commas.

➢ **LISTBOX**: A LISTBOX in ASP.NET is a web server control that allows you to display a list of items with checkboxes that the user can select or clear. The LISTBOX control is useful for creating multiple-choice questions, surveys, or preferences.

The LISTBOX control has the following features and properties:

- It can be bound to a data source, such as a database, an XML file, or another list of items, to dynamically generate the list items.
- It has an Items collection that contains the ListItem objects that represent each item in the list. You can access and modify the items by using the collection methods and properties, such as Add, Remove, Count, and Clear.
- It has a SelectedIndex property that returns the index of the first selected item in the list, or -1 if no item is selected. You can also use the SelectedIndices property to get a collection of all the selected indices in the list.
- It has a SelectedItem property that returns the first selected item in the list, or null if no item is selected. You can also use the SelectedItems property to get a collection of all the selected items in the list.
- It has a SelectedValue property that returns the value of the first selected item in the list, or an empty string if no item is selected. You can also use the SelectedValues property to get a collection of all the selected values in the list.
- It has a RepeatColumns property that specifies the number of columns or rows to display in the list. The value depends on whether you use Table or Flow layout and Horizontal or Vertical direction.

- It has a TextAlign property that specifies how the text is aligned with respect to the checkbox. You can choose between Left and Right alignments.

18.What is theme?Explain various way to apply theme.

- **Theme**: They are skin template that allows you to define the look of pages and controls, which can then be applied to all the pages in your application to provide consistent look for your application. It applies to any Server control. It is used to set Height, width, Forecolor, backcolor, bordercolor of any asp.net server controls
- Applies to all the server controls Is applied on the server rather than in the browser But we cannot apply multiple themes to a single page. Only one theme we can apply for a single page.
- But themes does not support cascading For property values defined in a theme, the theme property overrides the property values declaratively set on a control, unless the StyleSheetTheme property is explicitly defined.
- Can be applied through Configuration Files.
- All theme and Skin files should be placed in a special Asp.net folder called the "ApP_Themes in order for the themes to work and behave normally.
- Each theme should be associated with at least one Skin file.
- But a theme can define multiple properties of a control not just style properties such as we can specify the graphies property for a control, template layout of a GridView control etc.

There are two ways to Apply Theme to your web form

1. Add Theme Attribute inside @page Directive Theme="Skin File" (<%@Page Language="CH" MasterPageFile="/MasterPage.master" AutoEventWireup="true" CodeFile="Registration.aspx.cs" Inherits="Registration" Theme="SkinFile"9%>)
2. Right click on page-> goto properties-> In Theme property select the SkinFile that you have created. Note: This step is only useful when you have not applied master page to the form.

19.Write code snipped to fill dropdown from the database.

```
Imports System.Data
    Imports System.Data.SqlClient
Imports System.Configuration
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
If Not IsPostBack Then
Dim connectionString As String =
ConfigurationManager.ConnectionStrings("dbConnection").ConnectionS
tring
```

```
Dim command As New SqlCommand("SELECT ProductID, ProductName
FROM Products", connection)
Dim adapter As New SqlDataAdapter(command)
Dim products As New DataTable()
adapter.Fill(products)
ddlProducts.DataSource = products
ddlProducts.DataTextField = "ProductName"
ddlProducts.DataValueField = "ProductID"
ddlProducts.DataBind()
ddlProducts.Items.Insert(0, New ListItem("--Select Product--", "0"))
End If
        End Sub
```

20.Difference between CSS and skin file.

- **CSS**:
  - Applies to all HTML Controls
  - Is applied on the Client Side in the Browser
  - We can apply multiple style sheets to a single page
  - The CSS supports cascading
  - The CSS cannot override the property values defined for a control.
  - Cannot be applied through the configuration files
  - Can be used directly via a reference to the css file location
  - Do not require any other resource like Skin files
  - In case of CSS you can define only style properties
- **Themes**:
  - Applies to all the server controls
  - Is applied on the server rather than in the browser
  - But we cannot apply multiple themes to a single page. Only one theme we can apply for a single page.
  - But themes does not support cascading
  - For property values defined in a theme, the theme property overrides the property values declaratively set on a control, unless the StyleSheetTheme property is explicitly defined.
  - Can be applied through Configuration Files.
  - All theme and Skin files should be placed in a special Asp.net folder called the "ASP_Themes in order for the themes to work and behave normally.
  - Each theme should be associated with at least one Skin file.
    But a theme can define multiple properties of a control not just style properties such as we can specify the graphies property for a control, template layout of a GridView control etc.

21.Difference between label and literals.

- **Label**: lable in ASP.NET is a web server control that displays text on a web page. It can be used to create captions, titles, headings, or other textual elements. Label in ASP.NET has the following features and properties:
  - It can be associated with another input control, such as a text box or a radio button, by using the AssociatedControlID property. This allows the user to focus on the input control by clicking on the label text
  - It can be styled by using the CssClass property or the Style attribute. It can also inherit the styles from the parent container or the theme of the web page.
  - It can be accessed and modified by using the Text property or the InnerText property in the code-behind file. It can also display dynamic content by using data binding expressions or inline code blocks.
- **Literals**: Literals in ASP.NET are web server controls that display text or HTML content on a web page without any additional HTML tags. Literals in ASP.NET have the following features and properties:
  - They can be used to insert dynamic content, such as data binding expressions, inline code blocks, or scripts, into the web page. They can also be used to add static content, such as headings, paragraphs, or comments.
  - They have a Text property that sets or gets the content of the control. The Text property can contain plain text or HTML markup, depending on the Mode property of the control.
  - They do not generate any HTML tags by themselves, unlike other web server controls such as Label or TextBox. This makes them lighter and faster than other controls. However, this also means that they cannot be styled or associated with other controls.

22.Differentiate session and application variable.

- Session and application variables are two ways to store data in ASP.NET web applications. They have different scopes and lifetimes, as explained below:
  Session variables are used to store data that is specific to a single user or session. Each user who visits the web application has a unique session ID that is stored in a cookie or in the URL.
  - The session ID is used to identify the user and access the session variables that belong to that user. Session variables are stored in the server memory or in an external state server or database.
  - Session variables are created when a user starts a session and are destroyed when the user ends the session or w hen the session times out due to inactivity.

- Session variables can be used to store information such as user preferences, shopping cart items, or authentication status.

Application variables:

- Application variables are used to store data that is shared by all users and sessions of the web application.
- Application variables are stored in the server memory and are accessible by any page or class in the web application.
- Application variables are created when the web application starts and are destroyed when the web application ends or when the server shuts down.
- Application variables can be used to store information such as site configuration, global counters, or cached data.

23.What are Web services? and how can the interaction with web services are done?

➢ A web service is a web-based functionality that runs on Web Server. It is accessed using the protocols of the web to be used by the web applications. They are the small unit of code which is design to handle limited set of task. They are independent of operating system and programming language.

- Using HttpClient class: This class provides a base class for sending HTTP requests and receiving HTTP responses from a web service identified by a URI.
- You can use various methods of this class, such as GetAsync, PostAsync, PutAsync, and DeleteAsync, to perform CRUD operations on the web service.
- You can also use helper methods, such as ReadAsJsonAsync, PostAsJsonAsync, and PutAsJsonAsync, to work with JSON data
- Using WebRequest and WebResponse classes: These classes provide an abstract base class and an interface for making requests and receiving responses from a web service.
- You can use the Create method of the WebRequest class to create a request object for a specific URI and protocol. You can then set

the properties and headers of the request object, such as Method, ContentType, ContentLength, and Credentials.

- You can also write data to the request stream by using the GetRequestStream method. To get the response from the web service, you can use the GetResponse method of the request object, which returns a WebResponse object.
- You can then read data from the response stream by using the GetResponseStream method

24.Explain FileUpload control with example.

➢ **FileUpload Control**:

The FileUpload control is a common feature in web development frameworks, including ASP.NET. It allows users to select and upload files from their local devices to a web server. Here's an example of using the FileUpload control in an ASP.NET web form:

asp

```
<asp:FileUpload ID="fileUploadControl" runat="server" />
<asp:Button ID="uploadButton" runat="server" Text="Upload File" OnClick="UploadFile" />
```

In the code-behind (C# or VB.NET), you would handle the file upload process in the `UploadFile` method:

csharp

```
protected void UploadFile(object sender, EventArgs e)
{
    if (fileUploadControl.HasFile)
    {
        string fileName = Path.GetFileName(fileUploadControl.FileName);
        string filePath = Server.MapPath("~/Uploads/") + fileName;
        fileUploadControl.SaveAs(filePath);
        // Perform actions on the uploaded file here
    }
}
```

25.Write a note on user profile.

➢ **USER PROFILE**:

- User profile in ASP.NET is a feature that allows you to store and retrieve user-specific information for your web application. You can use user profile to customize the user experience, such as remembering their preferences, settings, or personal data. User profile can also be used to

track the user's activity, such as the last visit time, the number of visits, or the pages viewed.

To use user profile in ASP.NET, you need to do the following steps:

- Configure the profile provider in the web.config file. The profile provider is responsible for storing and retrieving the user profile data from a persistent source, such as a database or a file system. You can use the default SqlProfileProvider or create your own custom profile provider.
- Define the user profile properties in the web.config file. The user profile properties are the attributes that you want to store for each user, such as name, email, age, gender, etc. You can specify the data type, default value, and serialization mode for each property. You can also group the properties into profiles for different types of users.
- Access and modify the user profile data in your code. You can use the Profile object to get or set the values of the user profile properties. The Profile object is available in any page or class that inherits from Page or UserControl. You can also use the ProfileManager class to perform bulk operations on user profiles, such as deleting, migrating, or finding profiles.

26.Explain global.asax file in detail. OR write a note on global application class.

➢ **Global.asax File:**

- The `Global.asax` file is a special file in ASP.NET that serves as the Global Application Class.
- It contains application-level events and code that are executed during the application's lifecycle. Some of the key events in the `Global.asax` file include `Application_Start`, `Application_End`, `Session_Start`, and `Session_End`.
- You can use these events to perform tasks like initializing application-wide settings, handling errors, and managing session data.
- In ASP.NET, the "global.aspx" file, also known as the Global Application Class file, is a special file used to define and handle global application-level events and settings. It serves as a central location for managing application-wide configurations, authentication, and event handling. Here are some key points about the "global.aspx" file:
- **Global.asax File**: The file is named "Global.asax" and is typically located in the root directory of an ASP.NET web application. It has a code-behind file called "Global.asax.cs" where you write the server-side code to handle events.

- **Application Events:** The primary purpose of the Global.asax file is to define and handle application-level events. These events include Application_Start, Application_End, Session_Start, Session_End, and various error-handling events like Application_Error.
- **Application Lifecycle:** Global.asax allows you to hook into the application's lifecycle events. For example, Application_Start is triggered when the application first starts, and Application_End is triggered when the application is unloaded.
- **Session Events**: You can also use Global.asax to track session-related events. Session_Start occurs when a new user session is initiated, and Session_End occurs when a session ends, typically due to user inactivity.
- **Error Handling**: Global.asax is valuable for global error handling. Application_Error can be used to catch unhandled exceptions and implement custom error logging or redirection logic.
- **Application Settings:** You can define application-level variables or settings in Global.asax, making them accessible throughout the application.
- **Security and Authentication:** Global.asax can be used to handle authentication and authorization logic at the application level, ensuring consistency across the entire application.

27.List out state management techniques.Explain cookie in detail.

➢ Client-Side State Management techniques are:

1. **View State**
2. **Hidden Fields**
3. **Cookies**
4. **Query Strings**

- "A cookie is a small file capable of storing user information". cookies do not use server memory. "A small text file created by the client's browser and stored on the client hard disk by the browser is called Cookie". System.Web.HttpCookie namespace is required to create and access cookies in ASP.NET web applications
- When the next time the user makes a request for the same site for the same or another page, the browser checks for cookie for that site in the folder. If the cookie already exists it sends a request with the same cookie, else that request is considered as a new request.
- It is clear text so user is able read it. It helps us to store user preference information like like Username, Password,City and PhoneNo etc on the client

machine. It is easy way to maintain user's information. Cookies are Fast accessing.

28.Write a note on exception handling.justify Exception handling play major role to improve quality of software.

➢ **Exception Handling:**

Exception handling is a crucial aspect of software development that involves dealing with unexpected or exceptional situations that may arise during the execution of a program. Exception handling improves software quality in several ways:

- **Robustness**: Properly handling exceptions makes your software more robust and resilient to unexpected errors. It prevents crashes and provides graceful error recovery mechanisms.
- **Debugging**: Exception information helps developers identify and diagnose issues during development and testing, making it easier to fix problems.
- **User Experience:** Handling exceptions gracefully provides a better user experience by displaying helpful error messages and preventing application crashes.

29.Explain session in detail.

➢ Session The time duration for which the user interacts with the web application is called session". In ASP.NET session is a state that is used to store and retrieve values of a user. In ASP.NET, Session is the instance of HttpSessionState Class.

- The server maintains the state of user's information by using a session ID. When user makes a request without a session ID, ASP.NET first creates a session ID and then sends it with every request and response to and for the same user.
- There are 2 types of events available in ASP.NET. We can handle both sessions in a global.asax file.
  1. **Session_Start():** When the new session is initialized then the session_start event is invoked.
  2. **Session_end()**: When the session is expires then the Session_End event is invoked.

The session is stored in the following for ways in ASP.NET.

1. **InProcMode**: It is a default session mode and a value store in web server memory (11S). In this the session values are stored with server start and it ends when the server is restarted.

2. **State Server Mode**: In this mode session data is stored in separate server.
3. **SQL Server Mode:** In this session is stored in the database. It is a secure mode.
4. **Custom Mode:** Generally, session data is stored in InProc, Sql Server, State server, etc. but ASP.NET allows session data to be stored with other new techniques developed by developer also.

30.Difference between HTML Server controls and Web Server Controls.

➢ **Html Server**: The HTML server controls are basically the standard HTML controls enhanced to enable server side processing.

- The HTML controls such as the header tags, anchor tags, and input elements are not processed by the server but are sent to the browser for display
- hey are specifically converted to a server control by adding the attribute runat="server" and adding an id attribute to make them available for server-side processing.
- For example, consider the HTML input control: <input type="text" size="40"> It could be converted to a server control, by adding the runat and id attribute: < input type="text" Id="testtext" size="40" runat="server">
- Advantages of using HTML Server Controls: Although ASP.NET server controls can perform every job accomplished by the HTML server controls, the later controls are useful in the following cases:
- Using static tables for layout purposes.
- Converting a HTML page to run under ASP.NET

31.What is CSS? Discuss how CSS is more powerful than HTML (5) formatting .

➢ **CSS (Cascading Style Sheets):**

- CSS is a style sheet language used for describing the presentation and formatting of web documents, including HTML. It is more powerful than HTML in terms of formatting because:
  - **Separation of Concerns**: CSS separates the content (HTML) from the presentation (styling). This makes it easier to maintain and update the appearance of a website without changing the underlying content.
  - **Reusability**: CSS allows you to define styles once and apply them to multiple elements or pages, promoting code reusability.
  - **Control**: CSS offers fine-grained control over the layout, colors, fonts, and other aspects of a webpage, enabling precise and consistent styling.

- **Media Queries**:CSS allows responsive design by using media queries to adapt styles based on the device or screen size.

32.What is advantage of validation control? List validation controls and Explain range validator and regular expression validator in detail.

➢ **Validation Controls in ASP.NET:**

- Validation controls in ASP.NET are used to validate user input on web forms. Some common validation controls include:
- **RequiredFieldValidator**:Ensures that a field is not empty.
- RangeValidator: Validates that a value falls within a specified range.
- **RegularExpressionValidator**:Uses a regular expression pattern to validate input.
- **CompareValidator**: Compares the input value to another value or a control's value.
- **RangeValidator**: This control checks if the input falls within a specified numeric or date range. For example:

asp

```
<asp:RangeValidator ID="rangeValidator" runat="server"
ControlToValidate="TextBox1"
    Type="Integer" MinimumValue="1" MaximumValue="100"
    ErrorMessage="Value must be between 1 and 100." />
```

33.List types of control used to create ASP .NET page. Explain TreeView control in detail.

➢ ASP.NET provides various controls that can be used to create web pages. Here are some types of controls commonly used:

1. **HTML Controls**: Basic HTML elements like buttons, textboxes, and checkboxes can be used in ASP.NET pages. They are lightweight and easy to work with.

2. **Web Controls**: These controls are specifically designed for web development and offer more functionality than HTML controls. Examples include GridView, DropDownList, and Calendar controls.

3. **Validation Controls**: ASP.NET provides validation controls like RequiredFieldValidator and RegularExpressionValidator to validate user input easily.
4. **Data Controls**: These controls, such as Repeater and DataGrid, help in displaying data from various sources like databases.
5. **User Controls:** User controls are custom controls created by developers that can be reused across multiple pages.
6. **AJAX Controls** : ASP.NET AJAX controls enable you to build responsive and interactive web applications.
7. **Login Controls**: These controls, like Login and PasswordRecovery, simplify user authentication and password management.

Now, let's explain the TreeView control in detail:

TreeView Control:

The TreeView control in ASP.NET is used to display hierarchical data in a tree-like structure, much like the folder structure in Windows Explorer. It's particularly useful for representing categories, organizational structures, or any data with a parent-child relationship.

Here are its key features and properties:

- Nodes: The TreeView consists of nodes, each representing an item in the hierarchy. You can add nodes manually in the markup or programmatically in code-behind.
- **Hierarchical Structure**: Nodes can have child nodes, creating a hierarchical structure. This is ideal for displaying parent-child relationships.
- **Templates**: You can customize the appearance of nodes using templates for various states such as normal, selected, and expanded.
- **Data Binding**: You can bind the TreeView to data sources like XML, databases, or object collections. This allows dynamic population of the tree.
- **Expand/Collapse**: Users can expand and collapse nodes to navigate through the hierarchy.
- **Client-Side Events**: You can handle client-side events for user interactions like node selection or expansion without postbacks, improving performance.

34.What are the advantages of themes over CSS. Explain various ways to apply skin file.

- **Theme**: They are skin template that allows you to define the look of pages and controls, which can then be applied to all the pages in your

application to provide consistent look for your application. It applies to any Server control. It is used to set Height, width, Forecolor, backcolor, bordercolor of any asp.net server controls

- Applies to all the server controls
- Is applied on the server rather than in the browser
- But we cannot apply multiple themes to a single page. Only one theme we can apply for a single page.
- But themes does not support cascading
- For property values defined in a theme, the theme property overrides the property values declaratively set on a control, unless the StyleSheetTheme property is explicitly defined.
- Can be applied through Configuration Files.
- All theme and Skin files should be placed in a special Asp.net folder called the "ApP_Themes in order for the themes to work and behave normally.
- Each theme should be associated with at least one Skin file.
- But a theme can define multiple properties of a control not just style properties such as we can specify the graphies property for a control, template layout of a GridView control etc.
  There are two types of control skins, default skins and named skin:
- A default skin automatically applies to all controls of the same type when a theme is applied to a page. A control skin is a default skin if it does not have a SkiniD attribute. For example, if you create a defalt skin for a Calendar control, the control skin applies to ll Calendar controls on pages that use the theme. (Default skins are matched exactly by control type, so that a Button control skin applies to all Button controls, but not to LinkButton controls or to controls that derive from the Button object.
- A named skin is a control skin with a SkinID property set. Named skins do not automatically apply to controls by type. Instead, you explicitly apply a named skin to a control by setting the control's SkinID property. Creating named skins allows you to set different skins for different instances of the same control in an application.

35.Differentiate between DataSet and DataReader. Explain Read() method of DataReader with example.

➢ **DataSet vs. DataReader:**

- **DataSet**: It's an in-memory, disconnected representation of data retrieved from a database. It allows for data manipulation and can store multiple tables. It's suitable for situations where data needs to be manipulated extensively.

- **DataReader**: It's a forward-only, read-only stream of data retrieved from a database. It's more efficient than a DataSet when reading data sequentially. It's suitable for scenarios where you need to quickly read and process data, but not modify it.

The `Read()` method of a DataReader is used to advance the reader to the next record in the result set. Here's an example:

```
csharp

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

   // Access data using reader["columnName"]

}

reader.Close();
```

36.Write a note on ADO.NET architecture.

➢ The two core components of ADO.NET which are used to access and manipulate data are:

1. NET Framework Data Provider.
2. Data Set.

The ADO.NET architecture is based on a few key components, such as:

- **Connection**: This component is used to establish a connection between the .NET application and the data source. It requires a connection string that specifies the location and the credentials of the data source. Different types of connection classes are available for different data sources, such as SqlConnection, OracleConnection, OleDbConnection, etc.
- **Command**: This component is used to execute commands or queries on the data source. It requires a connection object and a command text that contains the SQL statement or the stored procedure name. Different types of command classes are available for different data sources, such as SqlCommand, OracleCommand, OleDbCommand, etc.
- **DataReader**: This component is used to read data from the data source in a fast and forward-only manner. It requires a command object and an

open connection object. It returns a stream of data that can be accessed by using methods and properties, such as Read, NextResult, GetOrdinal, GetName, GetValue, etc.

- **DataAdapter**: This component is used to transfer data between the data source and the DataSet object. It requires a connection object and one or more command objects for performing select, insert, update, and delete operations. It uses methods such as Fill and Update to populate and modify the DataSet object.
- **DataSet**: This component is used to store data in memory in a disconnected and relational manner. It consists of DataTables, DataColumns, DataRows, DataRelations, and Constraints that represent the structure and the content of the data. It can be filled by using a DataAdapter or by using methods such as ReadXml and WriteXml. It can also be bound to various controls for displaying or editing data.
- **DataView**: This component is used to sort, filter, or search data in a DataTable or a DataSet. It provides a customized view of the data without modifying the original data. It can also be bound to various controls for displaying or editing data.

37.Differentiate between error and exception. Explain block level exception handling .

➢ **Error vs. Exception:**

- **Error**: Errors are typically low-level issues that can lead to program crashes or unexpected behavior. They often indicate serious problems like memory exhaustion or hardware failures.
- **Exception**: Exceptions are higher-level events that can be caught, handled, and gracefully managed within the program. They represent situations where the program can continue running but may need to take specific actions to recover or report the issue.
- **Block-Level Exception Handling**: This refers to catching and handling exceptions at a specific block of code within the program. It allows you to isolate error-handling logic to specific sections where exceptions are likely to occur, enhancing code maintainability and robustness

38.Write a note on command builder.

➢ A command builder is a class in the ADO.NET that helps developers automatically generate SQL statements for updating, inserting, and deleting data in a database. It is used in conjunction with a data adapter to synchronize data between a Dataset and a database.

- To use a command builder, you first need to create a new instance of the command builder class and associate it with a data adapter. You can then use the command builder's methods to generate the SQL statements for the desired operations.
  The following are the steps on how to use a command builder in ASP.NET

- Create a new instance of the command builder class.
- Associate the command builder with a data adapter.
- Use the command builder's methods to generate the SQL statements for the desired operations.
- Execute the SQL statements using the data adapter.

Here is an example of how to use a command builder in ASP.NET

```
Dim commandBuilder As New SqlCommandBuilder(myDataAdapter)
Dim insertCommand As SqlCommand =
commandBuilder.GetInsertCommand()
insertCommand.Parameters.AddWithValue("@ProductID", productID)
insertCommand.Parameters.AddWithValue("@ProductName",
productName)
insertCommand.Parameters.AddWithValue("@QuantityInStock",
quantityInStock)
myDataAdapter.Update(insertCommand)
```

Here are some of the benefits of using a command builder:

- It can save time and effort by automating the generation of SQL statements.
- It can help to ensure that the SQL statements are compatible with the data adapter.
- It can help to prevent errors in SQL statements.

Here are some of the drawbacks of using a command builder:

- The command builder may not always generate the most efficient SQL statements.
- The command builder can be inflexible in some cases.
- The command builder can be difficult to use in complex scenarios.

39.Explain page processing sequence of ASP .NET web page.

➢ The page processing sequence of an ASP.NET web page is as follows:

- **Start stage**: The page properties such as Request and Response are set. At this stage, the page also determines whether the request is a post back or a new request and sets the Is PostBack property. The page also sets the UICulture property.

- **Init stage**: The page is initialized and the controls on the page are instantiated. A master page and themes are also applied to the page if applicable.
- **InitComplete stage**: This stage is executed after the Init stage and before the Load stage. This stage is typically used to initialize custom controls or perform other tasks that need to be done before the controls are loaded.
- **Load stage:** The controls on the page are loaded and their properties are set. The view state is restored if it is available.
- **PostBack stage**: This stage is executed when the page is posted back to the server. This happens when the user submits a form or clicks a button.
- **LoadComplete stage**: This stage is executed after the PostBack stage and before the PreRender stage. This stage is typically used to perform tasks that need to be done after the controls are loaded but before the page is rendered.
- **PreRender stage**: This stage is executed before the page is rendered to the client. This stage is typically used to perform tasks such as validating the form data or setting the values of the controls.
- **Render stage**: The page is rendered to the client.
- **Unload stage**: The page is unloaded from memory.
    - The page processing sequence can be affected by the following factors:
    - The type of request (postback or new request)
    - The presence of custom controls
    - The use of view state
    - The use of master pages

It is important to understand the page processing sequence in order to write code that works correctly at the appropriate stage of the page life cycle.

40.Differentiate between client side and server side validation. Explain range and regular expression validator in detail.

- Client-side validation is done in the user's browser using JavaScript. It is performed before the form data is sent to the server, so it can provide a quick and immediate feedback to the user. However, it is not as secure as server-side validation because the user can disable or change the JavaScript code.

- Server-side validation is done on the server side using code written in VB.NET. It is performed after the form data is sent to the server, so it can be more secure. However, it is slower than client-side validation because it requires an additional round trip to the server.

Here are the details of range and regular expression validators in ASP.NET:

- Range validator is used to validate the value of a control to be within a specified range. For example, you can use a range validator to ensure that the value of a textbox is between 1 and 100.Regular expression validator is used to validate the value of a control against a regular expression.
- A regular expression is a pattern that can be used to match text. For example, you can use a regular expression validator to ensure that the value of a textbox is a valid email address.

To use a range or regular expression validator in ASP.NET, you can add the appropriate validation control to the form. For example, the following code shows how to add a range validator to a textbox:

```
<asp:TextBox ID="txtNumber" runat="server"></asp:TextBox>
<asp:RangeValidator ID="valNumber" ControlToValidate="txtNumber"
MinimumValue="1" MaximumValue="100" ErrorMessage="The value must be
between 1 and 100." runat="server"></asp:RangeValidator>
```

- The MinimumValue and MaximumValue properties of the RangeValidator control specify the range of values that are allowed. The ErrorMessage property specifies the error message that is displayed if the value is not within the range.
- To use a regular expression validator, you can use the RegularExpressionValidator control. The ValidationExpression property of the RegularExpressionValidator control specifies the regular expression that is used to validate the value. The ErrorMessage property specifies the error message that is displayed if the value does not match the regular expression.

Here are some of the benefits of using client-side validation:

- It can provide a quick and immediate feedback to the user.
- It can improve the user experience by preventing the user from submitting invalid data.
- It can reduce the load on the server by validating the form data before it is sent to the server.

Here are some of the benefits of using server-side validation:

- It is more secure than client-side validation because the user cannot disable or change the code.

- It can be used to validate data that cannot be easily validated on the client side, such as data that is stored in a database

41.Explain with example DataAdapter.

- DataAdapter is a class in the ADO.NET that is used to fill and update data between a DataSet and a data source. It is a bridge between a DataSet and a data source.

The DataAdapter has two main methods:

- **Fill**(): This method is used to fill a DataSet with data from a data source.
- **Update**(): This method is used to update a data source with data from a DataSet.

To use a DataAdapter, you first need to create a new instance of the DataAdapter class and associate it with a DataSet and a data source. You can then use the Fill() or Update() method to fill or update the DataSet.

For example:

```
Dim dataAdapter As New SqlDataAdapter("SELECT * FROM Products",
connectionString)
Dim dataSet As New DataSet()
dataAdapter.Fill(dataSet)
' Do something with the data in the DataSet
dataAdapter.Update(dataSet)
```

In this example, we first create a new instance of the SqlDataAdapter class and specify the SQL SELECT statement that is used to fill the DataSet. We then create a new instance of the DataSet class and associate it with the DataAdapter object. We then use the Fill() method to fill the DataSet with data from the data source. Finally, we use the Update() method to update the data source with changes made to the DataSet.

Here are some of the benefits of using a DataAdapter:

- It can be used to connect to a variety of data sources, such as SQL Server, Oracle, and MySQL.
- It can be used to perform both read and write operations on the data source.
- It can be used to handle transactions, which ensures that the data is always consistent.

  Here are some of the drawbacks of using a DataAdapter:
- It can be complex to use, especially for complex data operations.
- It can be slow for large datasets.
- It can be difficult to debug.

42.Write a note on navigation controls.

- Navigation controls are very important for websites. Navigation controls are basically used to navigate the user through webpage .It is more helpful for making the navigation of pages easier .There are three controls in ASP.NET ,Which are used for Navigation on the webpage
  1. Pointer
  2. TreeView control
  3. Menu Control
  4. SiteMapPath control

- **Pointer**: It is just a pointer. If we drag any other control on form it causes to create that control on form but pointer does not create any control on form. In other word we can say, we select it for to ignore any other selected control.

- **Menu**: This control is used to create one or more MenuItems typically organized into different levels of a hierarchy. In this each MenuItem consists of properties that determine the look and feel of the MenuItem like text, navigateURL etc. Using CSS we can make this menu dynamic or cool look.

Properties of Menu Control:

- **DataSourceID**: This property is used to specify the data source to be used using sitemap file as data source.
- **CssClass**: This property is used to specify the CSS class attribute for the control.
- **ImgeUrl**: This property is used to specify the image that appear next to the menu item.
- **Orientation**: This property is used to specify the alignment of menu control. It can be horizontal or vertical.
- **Tooltip**: This property is used to specify the tooltip of the menu item when you mouse over.
- Text: This property is used to specify the text to display in the menu.
- **NavigateUrl**: This property is used to specify the target location to send the user when menu item is clicked.

- **Target**: This property is used to specify the target page location. It can be in new window or same window.
- **Value**: This property is used to specify the unique id to use in server side events.

- **SiteMapPath**: This control is used to displays a list of links representing the user's current page and the hierarchal path back to the root of the web application. In other word SiteMapPath is a way to present all folders and pages of the website. sitemap extension .<sitemap> element is the root node of the sitemap file.

It has three attributes:

- Title: It provides textual description of the link.
- URL: It provides the location of the valid physical file.
- Description: It is used for tooltip of the link.

Properties of SiteMapPath Control:

- **PathSeparator**: This property is to get or set the out separator text.
- **NodeStyle**: This property is used to set the style of all nodes that will be displayed.
- **RootNodeStyle**: This property is used to set the style on the absolute root node.
- **PathDirection**: This property is used to set the direction of the links generated in the output.
- **CurrentNodeStyle**: This property is used to set the style on node that represent the current page.
- **ShowToolTips**: This property is used to set the tooltip for the control. Default value is true.
- PathSeparatorStyle: This property ja used to set the style of path separator

- **TreeView**: This control is used to display the item in tree view format. We can expand and collapse the data list using tree root +/- sign.

Properties of TreeView Control:

- **DataSourceID**: This property is used to specify the data source to be used using sitemap file s data source.
- **ShowLines**: This property is used to specify the lines to connect the individual item in the tree.
- **CssClass**: This property is used to specify the CSS class attribute for the control.
- **ExpandDepth**: This property is used to specify the level at which items in the tree are expanded.

43.Write a note on login control.

- **The Login Controls**:- There are following Login controls developed by the Microsoft which are used in ASP.NET Website as given below:-
    1. **Login**
    2. **LoginView**
    3. **LoginStatus**
    4. **Loginname**
    5. **PasswordRecovery**
    6. **ChangePassword**
    7. **CreateUserWizard**

1.) The Login Control:-

The Login control provides a user interface which contains username and password, that authenticate the username and password and grant the access to the desired services on the basis of the credentials. There are used some methods ,properties and events in this Login control, You can check manually after drag and drop this control on your web form .

2.) The LoginView Control:-

The LoginView Control is a web server control ,Which is used to display two different views of a web page of any website , dependending on whether the any user has logged on to a web page as anonymous user or registered user .If the user is authenticated,the control displays the appropriate to the person with the help of the following views template.

Anonymous Template :- This template (default of all) will be displayed when any user just open the web page but not logged in.

LoggedInTemplate:- This Template (page)will be displayed when the user in logged in.

RoleGroups:- This template will be displayed when user logged in, that is the member of the specific role (defined role group).

3.) The LoginStatus Control :- The Login Status control specifies whether a particular user has logged on the website or not . When the user is not logged in, this control display the Login text as a hyperlink When the user is logged in, this control display the logout text as a hyperlink. To do this, Login Status control uses the authentication section of the web.config file. This control provides the following two views:-

1. **LoggedIn** --> It will be displayed, when the user is not Logged In.
2. **Logout** --> It will be displayed when the user is Logged In.

4.) The LoginName Control :-

The LoginName Control is responsible for display the names of all the authenticated users.If no users id logged in ,then this control is not displayed on the page. This control uses the page .User.Identity.Name namespace to return the value for the user's name.

5.) Passwordrecovery Control:-

The Passwordrecovery control is used to recover or reset the forgotten password of a user . This control does not display the password on the browser,but it sends the password to the respective email address whatever you have specified at the time of registration. This control has included three views as given below:-

1. **UserName** :- It refers to the view that accepts the username of a user.
2. **Question** :- It accepts the security questions asked from the users.
3. **Success** :- It display a message to the user that retrieved password has been set to the user.

6.) CreateUserWizard control:-

This control uses the membership service to create a new user in the membership data store. The CreateUserWizard control is provides by the CreateUserWizard class and can be customized by using template and style properties .Using this control any user can easily create an account and login to the web page

7.) The ChangePassword Control:-

Using this control, user can easily change your existing password (old password) on the ASP.NET Website. This control prompts uses to provide the current password first and then set the new password first and then set the new password. If the old password is not correct then new password can't be set. This is also helps to send the email to the respective users about the new password.This control is used ChangePassword class.