----------ASP. NET----------

1: What is ASP.NET?

Answer: ASP.NET was developed in direct response to the problems that developers had with classic ASP. Since ASP is in such wide use, however, Microsoft ensured that ASP scripts execute without modification on a machine with the .NET Framework (the ASP engine, ASP.DLL, is not modified when installing the .NET Framework). Thus, IIS can house both ASP and ASP.NET scripts on the same machine.

Advantages of ASP.NET

Separation of Code from HTML

Support for compiled languages

Use services provided by the .NET Framework

Graphical Development Environment

State management

Update files while the server is running

XML-Based Configuration Files

2: What are the different validators in ASP.NET?

Answer: ASP.NET validation controls define an important role in validating the user input data. Whenever the user gives the input, it must always be validated before sending it across to various layers of an application. If we get the user input with validation, then chances are that we are sending the wrong data. So, validation is a good idea to do whenever we are taking input from the user.

There are the following two types of validation in ASP.NET:

- Client-Side Validation
- Server-Side Validation

Client-Side Validation: When validation is done on the client browser, then it is known as Client-Side Validation. We use JavaScript to do the Client-Side Validation.

Server-Side Validation:

When validation occurs on the server, then it is known as Server-Side Validation. Server-Side Validation is a secure form of validation. The main advantage of Server-Side Validation is if the user somehow bypasses the Client-Side Validation, we can still catch the problem on server-side.

The following are the Validation Controls in ASP.NET:

- RequiredFieldValidator Control
- CompareValidator Control
- RangeValidator Control
- RegularExpressionValidator Control
- CustomFieldValidator Control
- ValidationSummary

3: What is View State?

Answer: View State is the method to preserve the Value of the Page and Controls between round trips. It is a Page-Level State Management technique. View State is turned on by default and normally serializes the data in every control on the page regardless of whether it is actually used during a post-back.

A web application is stateless. That means that a new instance of a page is created every time when we make a request to the server to get the page and after the round trip our page has been lost immediately

Features of View State

These are the main features of view state:

1. Retains the value of the Control after post-back without using a session.
2. Stores the value of Pages and Control Properties defined in the page.
3. Creates a custom View State Provider that lets you store View State Information in a SQL Server Database or in another data store.

Advantages of View State

1. Easy to Implement.
2. No server resources are required: The View State is contained in a structure within the page load.
3. Enhanced security features: It can be encoded and compressed or Unicode implementation.

4: What are the different Session state management options available in ASP.NET?

Answer: State Management in ASP.NET

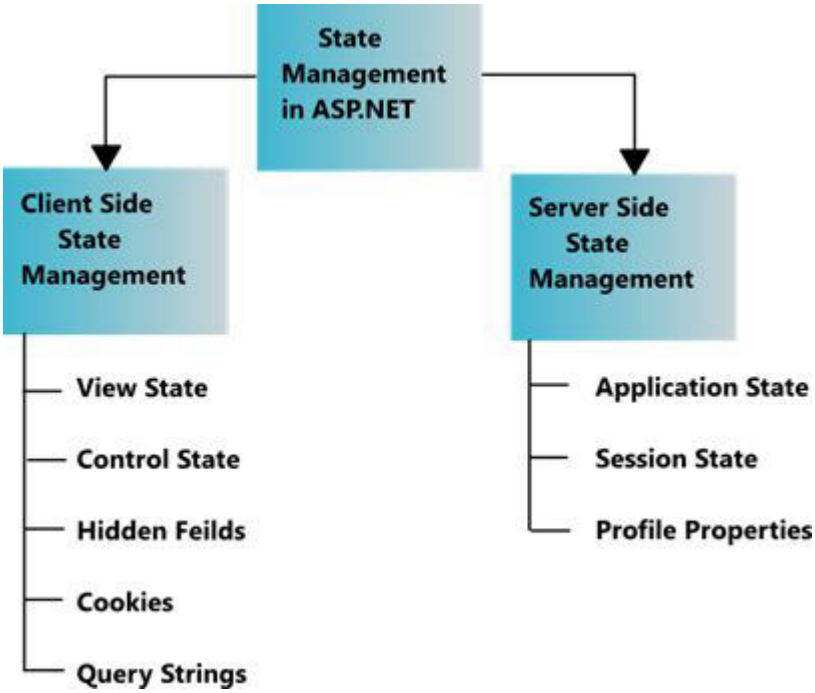
- A new instance of the Web page class is created each time the page is posted to the server.
- In traditional Web programming, all information that is associated with the page, along with the controls on the page, would be lost with each roundtrip.
- The Microsoft ASP.NET framework includes several options to help you preserve data on both a per-page basis and an application-wide basis.
  These options can be broadly divided into the following two categories:
- **Client-Side State Management Options**
- **Server-Side State Management Options**

Client-Side State Management

- Client-based options involve storing information either in the page or on the client computer.
- Some client-based state management options are:
  - Hidden fields
  - View state
  - Cookies
  - Query strings

Server-Side State Management

- There are situations where you need to store the state information on the server side.
- Server-side state management enables you to manage application-related and session-related information on the server.
- ASP.NET provides the following options to manage state at the server side:
    - Application state
    - Session state



5: What is caching in ASP.NET?

Answer: Caching is one of the most interesting concept and operation in ASP.NET. If you can handle it, you can run any web application by applying the caching concept depending on the requirements.

Caching is for providing solutions or the results to the users depending on their request, admin needs to recreate the pages often depending on user requests…STOP!!! "A cache simply stores the output generated by a page in the memory and this saved output (cache) will serve us (users) in the future.".

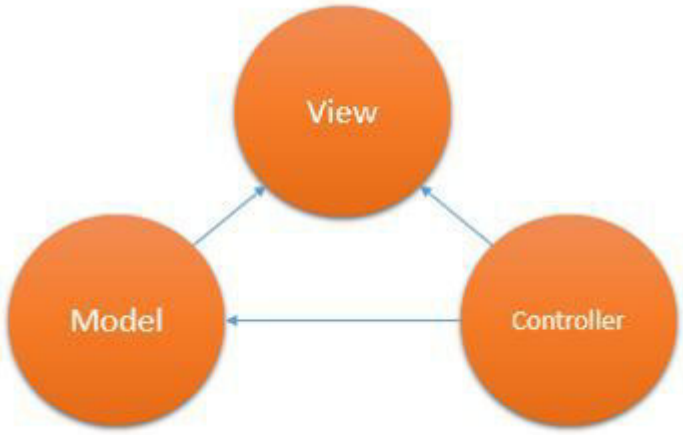
Types

6: What is MVC?

Answer: Model-View-Controller (MVC) is a pattern to separate an application into the following three main components:

1. Model
2. View
3. Controller

The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating web applications. The ASP.NET MVC Framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentications. The MVC framework is defined in the **System.Web.Mvc** assembly. It provides full control over HTML, JavaScript and CSS. It's the better as well as a recommended approach for large-scale applications where various teams are working together.



The ASP.NET MVC framework offers the following advantages:

- It makes it very easy to manage complexity by dividing an application into the Model, View and Controller.
- It does not use view state or server-based forms.
- Full control over HTML, JavaScript and CSS.
- It provides better support for Test-Driven Development (TDD).
- It works well for Web applications that are supported by large teams of developers and for web designers who need a high degree of control over the application behaviour.
- By default support of Facebook and Google Authentication.
- It easy to manage a large application to divide in multiple areas.

7: What are Cookies in ASP.NET?

Answer: Cookies are a State Management Technique that can store the values of control after a post-back. Cookies can store user-specific Information on the client's machine like when the user last visited your site. Cookies are also known by many names, such as HTTP Cookies, Browser Cookies, Web Cookies, Session Cookies and so on. Basically cookies are a small text file sent by the web server and saved by the Web Browser on the client's machine.
List of properties containing the HttpCookies Class:

1. **Domain:** Using these properties we can set the domain of the cookie.
2. **Expires:** This property sets the Expiration time of the cookies.
3. **HasKeys:** If the cookies have a subkey then it returns True.
4. **Name:** Contains the name of the Key.
5. **Path:** Contains the Virtual Path to be submitted with the Cookies.
6. **Secured:** If the cookies are to be passed in a secure connection then it only returns True.
7. **Value:** Contains the value of the cookies.

Limitation of the Cookies

1. The size of cookies is limited to 4096 bytes.
2. A total of 20 cookies can be used in a single website.

8: What is Ajax in ASP.NET?

Answer. Ajax stands for Asynchronous JavaScript and XML; in other words Ajax is the combination of various technologies such as a JavaScript, CSS, XHTML, DOM, etc.
AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the entire page.
We can also define Ajax is a combination of client side technologies that provides asynchronous communication between the user interface and the web server so that partial page rendering occurs instead of complete page post back.
Ajax is platform-independent; in other words AJAX is a cross-platform technology that can be used on any Operating System since it is based on XML & JavaScript. It also supports open source implementation of other technology. It partially renders the page to the server instead of complete page post back. We use AJAX for developing faster, better and more interactive web applications. AJAX uses a HTTP request between web server & browser.

- With AJAX, when a user clicks a button, you can use JavaScript and DHTML to immediately update the UI, and spawn an asynchronous request to the server to fetch results. When the response is generated, you can then use JavaScript and CSS to update your UI accordingly without refreshing the entire page. While this is happening, the form on the users screen doesn't flash, blink, disappear, or stall.
- The power of AJAX lies in its ability to communicate with the server asynchronously, using a XMLHttpRequest object without requiring a browser refresh.
- Ajax essentially puts JavaScript technology and the XMLHttpRequest object between your Web form and the server.

9: What are Web Services in ASP.NET?

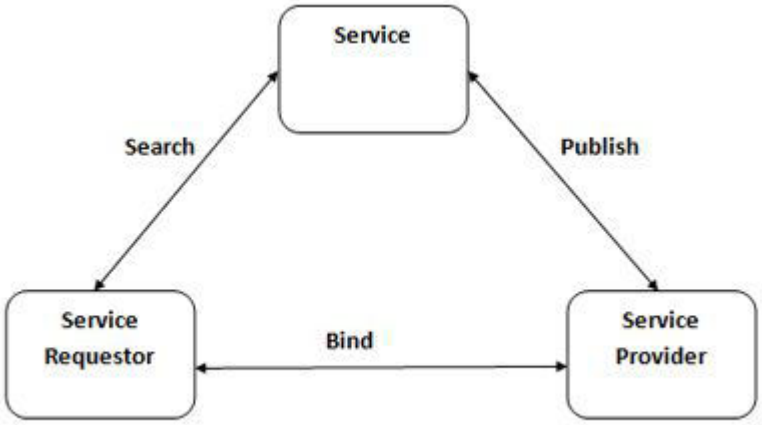
Answer: A Web Service is a software program that uses XML to exchange information with other software via common internet protocols. In a simple sense, Web Services are a way for interacting with objects over the Internet.

A web service is:

- Language Independent.
- Protocol Independent.
- Platform Independent.
- It assumes a stateless service architecture.
- Scalable (e.g. multiplying two numbers together to an entire customer-relationship management system).
- Programmable (encapsulates a task).
- Based on XML (open, text-based standard).
- Self-describing (metadata for access and use).
- Discoverable (search and locate in registries)- ability of applications and developers to search for and locate desired Web services through registries. This is based on UDDI.

Key Web Service Technologies:

- **XML-** Describes only data. So, any application that understands XML-regardless of the application's programming language or platform-has the ability to format XML in a variety of ways (well-formed or valid).
- **SOAP-** Provides a communication mechanism between services and applications.
- **WSDL-** Offers a uniform method of describing web services to other programs.
- **UDDI-** Enables the creation of searchable Web services registries.



10: What are the Advantages of ASP.NET?

Answer: ASP.NET provides services to allow the creation, deployment, and execution of Web Applications and Web Services like ASP, ASP.NET is a server-side technology. Web Applications are built using Web Forms. ASP.NET comes with built-in Web Form controls, which are responsible for generating the user interface. They mirror typical HTML widgets such as text boxes or buttons. If these controls do not fit your needs, you are free to create your own user controls.

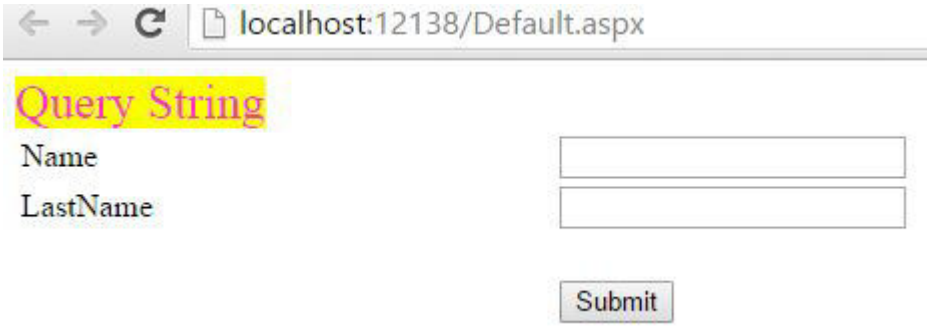
Advantages of ASP.NET:

- Separation of Code from HTML
- Support for compiled languages
- Use services provided by the .NET Framework
- Graphical Development Environment
- Update files while the server is running
- XML-Based Configuration Files

11: What is Query String in ASP?

Answer: A QueryString is a collection of characters input to a computer or web browser. A Query String is helpful when we want to transfer a value from one page to another. When we need to pass content between the HTML pages or aspx Web Forms in the context of ASP.NET, a Query String is Easy to use and the Query String follows a separating character, usually a Question Mark (?). It is basically used for identifying data appearing after this separating symbol. A Query String Collection is

used to retrieve the variable values in the HTTP query string. If we want to transfer a large amount of data then we can't use the Request.QueryString. Query Strings are also generated by form submission or can be used by a user typing a query into the address bar of the browsers.
Syntax of Query String -*Request.QueryString(variable)[(index).count]*



Advantages:

- Simple to Implement
- Easy to get information from Query string.
- Used to send or read cross domain (from different domain).

Disadvantages:

- Human Readable
- Client browser limit on URL length
- Cross paging functionality makes it redundant
- Easily modified by end user

12: Use of CheckBox in .NET?

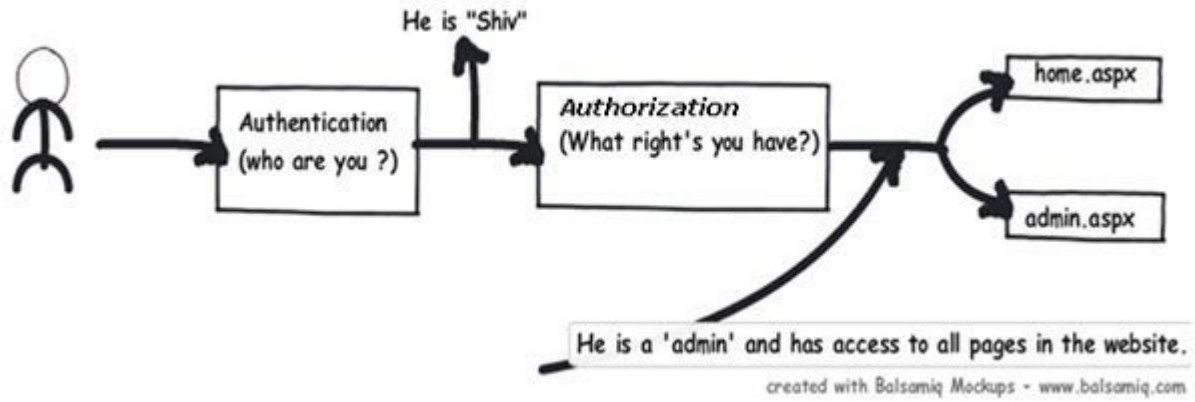
Answer: The CheckBox control is a very common control of HTML, unlike radio buttons it can select multiple items on a webpage. The CheckBox control in ASP.NET has many properties and some of them are listed below.

Property	Description
AutoPostBack	Specifies whether the form should be posted immediately after the Checked property has changed or not. The default is false.
CausesValidation	Specifies if a page is validated when a Button control is clicked.
Checked	Specifies whether the check box is checked or not.
InputAttributes	Attribute names and values used for the Input element for the CheckBox control.
LabelAttributes	Attribute names and values used for the Label element for the CheckBox control.
runat	Specifies that the control is a server control. Must be set to "server".
Text	The text next to the check box.
TextAlign	On which side of the check box the text should appear (right or left).
ValidationGroup	Group of controls for which the Checkbox control causes validation when it posts back to the server.
OnCheckedChanged	The name of the function to be executed when the Checked property has changed.

13: What is the authentication and authorization in ASP.NET?

- **Authentication:** Prove genuineness
- **Authorization:** process of granting approval or permission on resources.

In ASP.NET authentication means to identify the user or in other words its nothing but to validate that he exists in your database and he is the proper user.
Authorization means does he have access to a particular resource on the IIS website. A resource can be an ASP.NET web page, media files (MP4, GIF, JPEG etc), compressed file (ZIP, RAR) etc.



Types of authentication and authorization in ASP.NET
There are three ways of doing authentication and authorization in ASP.NET:

- **Windows authentication:** In this methodology ASP.NET web pages will use local windows users and groups to authenticate and authorize resources.
- **Forms Authentication:** This is a cookie based authentication where username and password are stored on client machines as cookie files or they are sent through URL for every request. Form-based authentication presents the user with an HTML-based Web page that prompts the user for credentials.
- **Passport authentication:** Passport authentication is based on the passport website provided by the Microsoft .So when user logins with credentials it will be reached to the passport website (i.e. hotmail,devhood,windows live etc) where authentication will happen. If Authentication is successful it will return a token to your website.
- **Anonymous access:** If you do not want any kind of authentication then you will go for Anonymous access.

In 'web.config' file set the authentication mode to 'Windows' as shown in the below code snippets.

```
<authentication mode="Windows"/>
```

We also need to ensure that all users are denied except authorized users. The below code snippet inside the authorization tag that all users are denied. '?' indicates any unknown user.

```
<authorization>
   <deny users="?"/>
</authorization>
```

14: What are the HTML server control in ASP.NET?

Answer: The Microsoft.NET Framework provides a rich set of server-side controls for developing Web applications. You can add these controls to WebForms pages just as you add Windows controls to a form. Server-side controls are often called server controls or Web Forms controls. There are four types of Server controls: HTML server controls. Web server controls, validation control, and user controls.

- **HTML Server controls**
  HTML developers must be familiar with old HTML controls, which they use to write GUI applications in HTML. These controls are the same HTML controls; you can run these controls on the server by defining the runat ="server" attribute. These control names start with Html.

Controls	Description
HtmlForm	Create an HTML form control, used as a place holder of other controls.
HtmlInputText	Creates an input text box control used to get input from user.
HtmltextArea	Creates multiline text box control.
HtmlAnchor	Creates a Web navigation.
HtmlButton	Creates a button control.
HtmlImage	Creates an image control, which is used to display an image.
HtmlInputCheckBox	Creates a check box control.
HtmlInputRadioButton	Creates a radio button control.
HtmlTable	Creates a table control.
HtmlTableRow	Creates a row within a table.
HtmlTableCell	Creates a cell with in a row.

  - Web Server Controls
  - Validation Controls
  - User Controls

15: What is the web API in ASP.NET?

Answer: It is a framework provided by Microsoft for writing HTTP services. There are many frameworks available to build HTTP based services. They follow a common guideline of international standardization but with different flavors.
For example, all framework must adhere to these status codes-

- **1xx** - Informational Message
- **2xx** - Successful
- **3xx** - Redirection
- **4xx** - Client Error
- **5xx** - Server Error

Features:

- It is light weight and thus good for small devices also like tablets, smart phones.
- No tedious & extensive configuration like WCF REST is required.
- MediaTypeFormatter makes easy to configure your APIs response type in single line (JSON, XML and so on).

- IIS Hosting dependency is no more and it can be hosted in application too.
- Easy and simple control with HTTP features such as Caching, Versioning, request/response headers and its various content formats.
- It support content-negotiation (deciding the best response data format that client can accept).

16: What is the code behind and Inline Code?

Code Behind

Code Behind refers to the code for an ASP.NET Web page that is written in a separate class file that can have the extension of .aspx.cs or .aspx.vb depending on the language used. Here the code is compiled into a separate class from which the .aspx file derives. You can write the code in a separate .cs or .vb code file for each .aspx page. One major point of Code Behind is that the code for all the Web pages is compiled into a DLL file that allows the web pages to be hosted free from any Inline Server Code.

Inline Code

Inline Code refers to the code that is written inside an ASP.NET Web Page that has an extension of .aspx. It allows the code to be written along with the HTML source code using a <Script> tag. It's major point is that since it's physically in the .aspx file it's deployed with the Web Form page whenever the Web Page is deployed.

17: What is the ASP.NET page life Cycle?

Answer: When a page is requested by the user from the browser, the request goes through a series of steps and many things happen in the background to produce the output or send the response back to the client. The periods between the request and response of a page is called the "Page Life Cycle".

- **Request:** Start of the life cycle (sent by the user).
- **Response:** End of the life cycle (sent by the server).

There are four stages that occur during the Page Life Cycle before the HTML Response is returned to the client. Later in this article we"ll study all these stages and their sub events.

1. Initialization
2. Loading
3. Rendering
4. Unloading

Initialization	During this stage the IsPostback property is set. The page determines whether the request is a Postback (old request) or if this is the first time the page is being processed (new request). Controls on the page are available and each control's UniqueID property is set. Now if the current request is a postback then the data has not been loaded and the value of the controls have not yet been restored from the view state.
Loading	At this stage if the request is a Postback then it loads the data from the view state.
Rendering	Before rendering, the View State is saved for the page and its controls. During this phase, the page calls the render method for each control, providing a text writer that writes its output to the OutputStream of the page's Response property.
Unloading	Unload is called after the page has been fully rendered, sent to the client and is ready to be discarded. At this point also the page properties such as Response and Request are unloaded.

18: What is the ASP.NET page life cycle events?

Answer: We have many events in ASP.NET page life cycle let's see some most important events:

- **Page request**
  When ASP.NET gets a page request, it decides whether to parse and compile the page or there would be a cached version of the page; accordingly the response is sent,
- **Starting of page life cycle**
  At this stage, the Request and Response objects 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.
- **Page initialization**
  At this stage, the controls on the page are assigned unique ID by setting the UniqueID property and themes are applied. For a new request postback data is loaded and the control properties are restored to the view-state values.
- **Page load**
  At this stage, control properties are set using the view state and control state values.
- **Validation**
  Validate method of the validation control is called and if it runs successfully, the IsValid property of the page is set to true.
- **Postback event handling**
  If the request is a postback (old request), the related event handler is called.
- **Page rendering**
  At this stage, view state for the page and all controls are saved. The page calls the Render method for each control and the output of rendering is written to the OutputStream class of the Page's Response property.
- **Unload**
  The rendered page is sent to the client and page properties, such as Response and Request are unloaded and all cleanup done.

ASP.NET Page Life Cycle Events

Following are the page life cycle events:

- **PreInit**
  PreInit is the first event in page life cycle. It checks the IsPostBack property and determines whether the page is a postback. It sets the themes and master pages, creates dynamic controls and gets and sets profile property values. This event can be handled by overloading the OnPreInit method or creating a Page_PreInit handler.
- **Init :-** Init event initializes the control property and the control tree is built. This event can be handled by overloading the OnInit method or creating a Page_Init handler.
- **InitComplete**
  InitComplete event allows tracking of view state. All the controls turn on view-state tracking.

- **LoadViewState**

LoadViewState event allows loading view state information into the controls.

- **LoadPostData**
  During this phase, the contents of all the input fields defined with the <form> tag are processed.
- **PreLoad**
  PreLoad occurs before the post back data is loaded in the controls. This event can be handled by overloading the OnPreLoad method or creating a Page_PreLoad handler.
- **Load**
  The Load event is raised for the page first and then recursively for all child controls. The controls in the control tree are created. This event can be handled by overloading the OnLoad method or creating a Page_Load handler.
- **LoadComplete**
  The loading process is completed, control event handlers are run and page validation takes place. This event can be handled by overloading the OnLoadComplete method or creating a Page_LoadComplete handler.
- **PreRender:-** The PreRender event occurs just before the output is rendered. By handling this event, pages and controls can perform any updates before the output is rendered.
- **PreRenderComplete**
  as the PreRender event is recursively fired for all child controls, this event ensures the completion of the pre-rendering phase.
- **SaveStateComplete**
  State of control on the page is saved. Personalization, control state and view state information is saved. The HTML markup is generated. This stage can be handled by overriding the Render method or creating a Page_Render handler.
- **UnLoad**
  The UnLoad phase is the last phase of the page life cycle. It raises the UnLoad event for all controls recursively and lastly for the page itself. Final cleanup is done and all resources and references, such as database connections, are freed. This event can be handled by modifying the OnUnLoad method or creating a Page_UnLoad handler.

19: Describe login Controls in ASP?

Answer: The Login control provides the user interface to log a user into a web site. The Login control uses the Membership service to authenticate the user in your membership system. The default Membership service from your configuration file will be used automatically, however you can also set the Membership provider that you would like used as a property on the control.
The Login Control consists of:

- **Username Label and Textbox:** Collects the string used to identify the user in the membership system.
- **Password Label and Textbox:** Collects the password for the specified user. The textbox text is always obscured.
- **LoginButton:** The button to submit the users request for authentication.
- **RememberMe:** Configurable to display a checkbox giving the user the option to store a persistent cookie on the user's machine.
- **Title and Instruction:** Text to orient and guide the user through the process.

- **Links:** Configurable links to help, password recovery and user registration information.
- **Validators:** Required field Validators for the username and password textboxes.

20: What are the data controls available in ASP.NET?

Answer: The Controls having DataSource Property are called Data Controls in ASP.NET. ASP.NET allows powerful feature of data binding, you can bind any server control to simple properties, collections, expressions and/or methods. When you use data binding, you have more flexibility when you use data from a database or other means.
Data Bind controls are container controls.
Controls -> Child Control
Data Binding is binding controls to data from databases. With data binding we can bind a control to a particular column in a table from the database or we can bind the whole table to the data grid.
Data binding provides simple, convenient, and powerful way to create a read/write link between the controls on a form and the data in their application.
Data binding allows you to take the results of properties, collection, method calls, and database queries and integrate them with your ASP.NET code. You can combine data binding with Web control rendering to relieve much of the programming burden surrounding Web control creation. You can also use data binding with ADO.NET and Web controls to populate control contents from SQL select statements or stored procedures.
Data binding uses a special syntax:
<%# %>
The *<%#,* which instructs ASP.NET to evaluate the expression. The difference between a data binding tags and a regular code insertion tags <% and %> becomes apparent when the expression is evaluated. Expressions within the data binding tags are evaluated only when the DataBind method in the Page objects or Web control is called.
Data Bind Control can display data in connected and disconnected model.
Following are data bind controls in ASP.NET:

- **Repeater Control , DataGrid Control , DataList Control, GridView Control, DetailsView**
- **FormView, DropDownList, ListBox, RadioButtonList, CheckBoxList**
- **BulletList etc.**

21: How to use repeater control in ASP.NET?

Answer: A Repeater is a Data-bound control. Data-bound controls are container controls. It creates a link between the Data Source and the presentation UI to display the data. The repeater control is used to display a repeated list of items.
The main use of Repeater Control is for displaying a repeated list of items bound to the control. A Repeater Control is faster and lightweight for displaying data compared to a GridView or DataGrid. With the Repeater control we can display data in a custom format. The main drawback of a Repeater Control is that it doesn't support paging and sorting.
The Repeater Control has the following types of template fields:

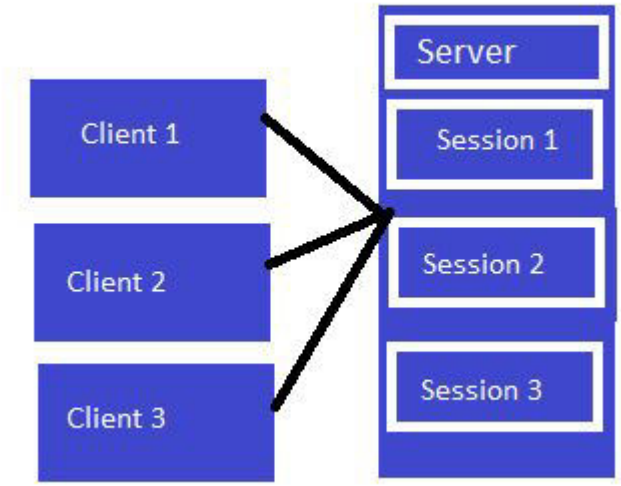
- Item Template
- AlternatingItem Template
- Header Template
- Footer Template
- Separator Template

22: What are different methods of session maintenance in ASP.NET?

Answer: Session is a State Management Technique. A Session can store the value on the Server. It can support any type of object to be stored along with our own custom objects. A session is one of the best techniques for State Management because it stores the data as client-based, in other words the data is stored for every user separately and the data is secured also because it is on the server.

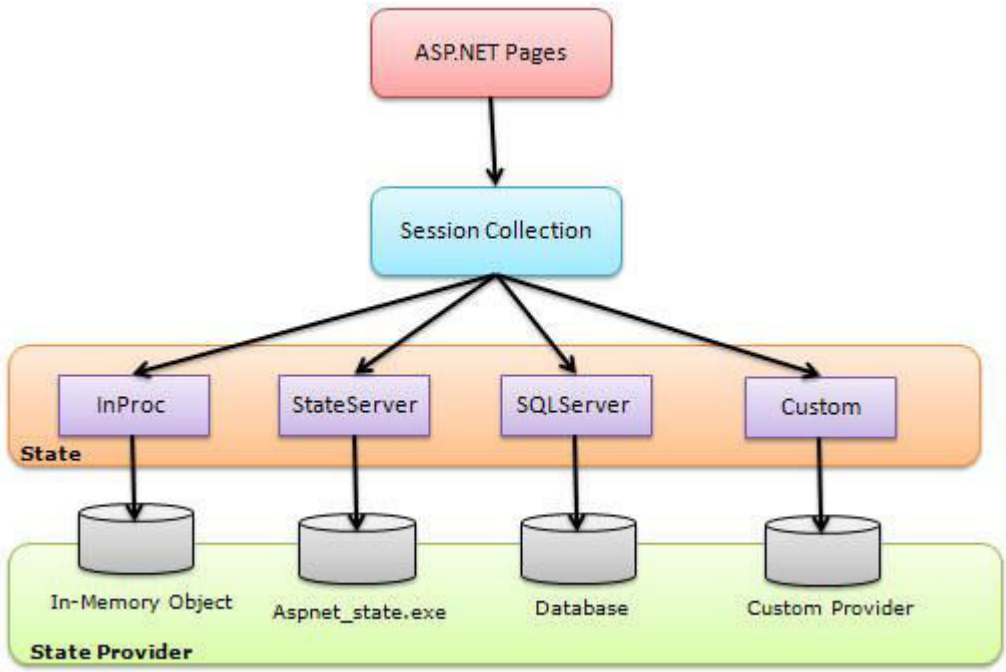
We can set the session on one of the following 2 types of configuration files:

1. **Machine Configuration file:** Machine Configuration is applied for all application.
2. **Application Configuration file:** It's applied for only application by application basis.



Session Mode

In ASP.NET there are 4 types of Session Mode.
Off: We can disable the session mode for the entire application using the off mode.



According to performance and durability the difference between InProc,State Server and SQL Server is:

Session mode	Performance Durability
InProc	More(1 processor and 1 server) less.
State Server	Medium(n processor and 1 server) Medium
SQL Server	Less More

23: What is the Difference between session and caching?

Answer: The first main difference between session and caching is: a session is per-user based but caching is not per-user based, So what does that mean? Session data is stored at the user level but caching data is stored at the application level and shared by all the users. It means that it is simply session data that will be different for the various users for all the various users, session memory will be allocated differently on the server but for the caching only one memory will be allocated on the server and if one user modifies the data of the cache for all, the user data will be modified.

24: What is page directives in ASP.NET?

Answer: Basically Page Directives are commands. These commands are used by the compiler when the page is compiled.

How to use the directives in an ASP.NET page

It is not difficult to add a directive to an ASP.NET page. It is simple to add directives to an ASP.NET page. You can write directives in the following format:

<%@[Directive][Attributes]%>

See the directive format, it starts with "<%@" and ends with "%>". The best way is to put the directive at the top of your page. But you can put a directive anywhere in a page. One more thing, you can put more than one attribute in a single directive.

Here is the full list of directives:

- @Page
- @Master
- @Control
- @Import
- @Implements
- @Register
- @Assembly
- @MasterType
- @Output Cache
- @PreviousPageType
- @Reference

25: Explain GridView control in ASP.NET?

Answer: The GridView control displays the values of a data source in a table. Each column represents a field, while each row represents a record. The GridView control supports the following features:

- Binding to data source controls, such as SqlDataSource.
- Built-in sort capabilities.
- Built-in update and delete capabilities.
- Built-in paging capabilities.
- Built-in row selection capabilities.
- Programmatic access to the GridView object model to dynamically set properties, handle events, and so on.
- Multiple key fields.
- Multiple data fields for the hyperlink columns.
- Customizable appearance through themes and styles.

Creating a GridView

1. <asp:GridView ID="gridService" runat="server"> , 2. </asp:GridView>

26: What is the difference between ASP.NET Web API and WCF?

Answer: The ASP. NET Web API is a framework that uses the HTTP services and makes it easy to provide the response to the client request. The response depends on the request of the clients. The Web API builds the HTTP services, and handles the request using the HTTP protocols. The request may be GET, POST, DELETE, PUT. We can also say that the ASP. NET Web API:

- Is an HTTP service.
- Is designed for reaching the broad range of clients.
- Uses the HTTP application.

We use the ASP. NET Web API for creating the REST ful (Representational State Transfer) services. The following are some important points of the ASP. NET Web API:

- The ASP. NET Web API supports the MVC application features that are controller, media formatters, routing etcetera.
- It is a platform for creating the REST services.
- It is a framework for creating the HTTP services.

- Responses can be formatted by the APIs MediaTypeFormatter into the Java Script Object Notation (JSON) and Extencible Markup Language (XML) formats.

27: What is the PostBack property in ASP.NET?

Answer: If we create a web Page, which consists of one or more Web Controls that are configured to use AutoPostBack (Every Web controls will have their own AutoPostBack property), the ASP.NET adds a special JavaScipt function to the rendered HTML Page. This function is named _doPostBack() . When Called, it triggers a PostBack, sending data back to the web Server.
ASP.NET also adds two additional hidden input fields that are used to pass information back to the server. This information consists of ID of the Control that raised the event and any additional information if needed. These fields will empty initially as shown below,

1. <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
2. <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />

The following actions will be taken place when a user changes a control that has the AutoPostBack property set to true:

1. On the client side, the JavaScript _doPostBack function is invoked, and the page is resubmitted to the server.
2. ASP.NET re-creates the Page object using the .aspx file.
3. ASP.NET retrieves state information from the hidden view state field and updates the controls accordingly.
4. The Page.Load event is fired.
5. The appropriate change event is fired for the control. (If more than one control has been changed, the order of change events is undetermined.)

6. The Page.PreRender event fires, and the page is rendered (transformed from a set of objects to an HTML page).
7. Finally, the Page.Unload event is fired.
8. The new page is sent to the client.

28: Explain Cookie-less Session in ASP.NET.

Answer: By default a session uses a cookie in the background. To enable a cookie-less session, we need to change some configuration in the Web.Config file. Follow these steps:

1. Open Web.Config file.
2. Add a <sessionState> tag under <system.web> tag.
3. Add an attribute "cookieless" in the <sessionState> tag and set its value to "AutoDetect" like below:
    1. <sessionState cookieless="AutoDetect" regenerateExpiredSessionId="true"/>

The possible values for "**cookieless**" attribute are:

- **AutoDetect:** Session uses background cookie if cookies are enabled. If cookies are disabled, then the URL is used to store session information.
- **UseCookie:** Session always use background cookie. This is default.

- **UseDeviceProfile:** Session uses background cookie if browser supports cookies else URL is used.
- **UseUri:** Session always use URL.

"**regenerateExpiredSessionId**" is used to ensure that if a cookieless url is expired a new new url is created with a new session. And if the same cookieless url is being used by multiple users an the same time, they all get a new regenerated session url.

29: What are the Navigations technique in ASP.NET?

Answer: Navigation can cause data loss if it not properly handled. We do have many techniques to transfer data from one page to another but every technique has its own importance and benefits.
We will discuss the following techniques in this article.

- Response.Redirect
- Server.Transfer
- Server.Exceute
- Cross page posting

30: What are master pages?

Answer: Some points about Master Pages:

1. The extension of MasterPage is '.master'.
2. MasterPage cannot be directly accessed from the client because it just acts as a template for the other Content Pages.
3. In a MasterPage we can have content either inside ContentPlaceHolder or outside it. Only content inside the ContentPlaceHolder can be customized in the Content Page.
4. We can have multiple masters in one web application.
5. A MasterPage can have another MasterPage as Master to it.
6. The content page content can be placed only inside the content tag.
7. Controls of MasterPage can be programmed in the MasterPage and content page but a content page control will never be programmed in MasterPage.
8. A master page of one web application cannot be used in another web application.
9. The **MasterPageFile** property of a webform can be set dynamically and it should be done either in or before the Page_PreInit event of the WebForm. **Page.MasterPageFile = "MasterPage.master"**. The dynamically set Master Page must have the ContentPlaceHolder whose content has been customized in the WebForm.
10. The order in which events are raised: Load (Page) a Load (Master) a LoadComplete (Page) i.e. if we want to overwrite something already done in Load event handler of Master then it should be coded in the **LoadComplete** event of the page.
11. Page_Load is the name of method for event handler for Load event of Master. (it's not **Master_Load**).