1. Define State Management Technique in Asp.net?

Ans: State Management Technique-

It is a mechanism of maintaining the state of the response either on browser or on server.

These are two types-

- **Client-side state management technique:** Client -side state management technique is a mechanism of maintaining the state of response on browser.
  Types of Client-side SMT-
  a) View State
  b) Query String
  c) Hidden Field
  d) Cookies

- **Server-side state management technique:** Server-side state management technique is a mechanism of maintaining the state of response on web server.
  Types of Server-side SMT-
  a) Session
  b) Application
  c) Caching

2. Explain following in short.

I. View State
II. Cookies
III. Query String
IV. Hidden Field

Ans:

I. ViewState-

- ViewState is a client-side SMT which is used to maintain the state of response on web browser only.
- Scope of ViewState is within the page that is we can not maintain the values between multiple web pages.

Syntax- To store the value in ViewState-

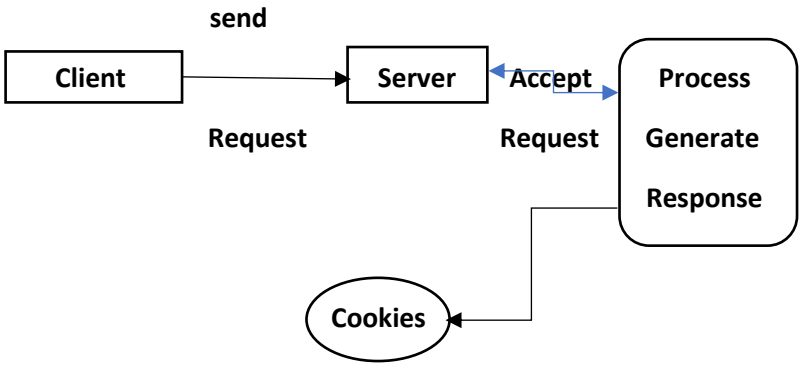
```
ViewState["Varname"]=value;
```

To read the value from ViewState-

```
ViewState["Varname"];
```

II. **Cookies-**

- Cookies is a client-side SMT which is used to maintain the response on browser in plain text format.
- Whenever client sends the request to server (IIS) , Server will accept the request , process and generate the response and store the response in cookies format.



III. **QueryString:**

QueryString will maintain the values between one page to another page.

Syntax-

To pass value in QueryString-

Response.Redirect("destinationUrl?QueryStringName=value");

To read value from QueryString-

Request.QueryString["QueryStringName"];

IV. **HiddenField:**

Through HiddenField we hide the query string data within browser url.

How to hide the query string data within browser url.

Ans: By using Server. Transfer ();

3. Explain following in short.

a) **Session**
b) **Application**
c) **Caching**

Ans:

a) **Session-**

- Session is server-side SMT which is used to maintain the data on web server within a particular time period between login and logout.
- Whenever client sends the request , web server will accept the request , process and generate the response apart from that a session is created on web server and session was assigned with same id that is session id.

b) Application-

- Application State is a state management technique. Application State is stored in the memory of the the server and is faster than storing and retrieving information in a database.
- Application State does not have a default expiration period. When we close the worker process the application object will be lost.
- Technically the data is shared amongst users by a HTTPApplcationState class and the data can be stored here in a key/value pair. It can also be accessed using the application property of the HTTPContext class.

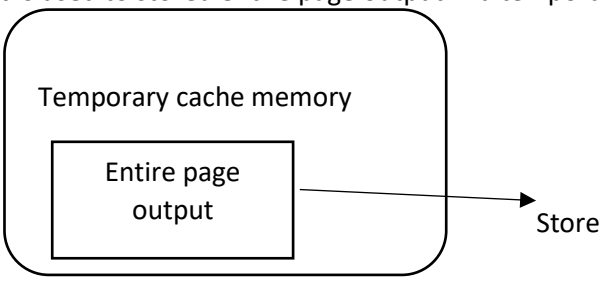
c) Caching-

- Caching is a Server-side SMT which is used to store frequently data in a temporary cache memory.
- Whenever client send the request to the server , server will accept the request , process and generate response store in cache memory.

These are 3 types-

a) Output caching-

It is used to stored entire page output in a temporary cache.



b) Fragment caching- It is used to apply caching for the portion of the page. We can achieve by using web-user controls.

c) Data caching- It is used to cache the data.

Syntax-

Store the value-

Cache [“Varname”]=value;

Read the value-

Cache [“Varname”]

4. Define validation controls in Asp.net.

Ans:

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:

- **RequiredFieldValidator**
- **RangeValidator**
- **CompareValidator**
- **RegularExpressionValidator**
- **CustomValidator**
- **ValidationSummary**

1. RequiredFieldValidator-

- It is used to make a field as mandatory in the form.
- Without filling the field user can’t submit the form.

properties and the methods-

Members	Description
ControlToValidate	Indicates the input control to validate.
Display	Indicates how the error message is shown.
EnableClientScript	Indicates whether client side validation will take.
Enabled	Enables or disables the validator.
ErrorMessage	Indicates error string.
Text	Error text to be shown if validation fails.
IsValid	Indicates whether the value of the control is valid.
SetFocusOnError	It indicates whether in case of an invalid control, the focus should switch to the related input control.
ValidationGroup	The logical group of multiple validators, where this control belongs.
Validate()	This method revalidates the control and updates the IsValid property.

2. **RangeValidator-**
    - It is used to validate if its given data is in between specified range or not.
      It has three specific properties:

Properties	Description
Type	It defines the type of the data. The available values are: Currency, Date, Double, Integer, and String.
MinimumValue	It specifies the minimum value of the range.
MaximumValue	It specifies the maximum value of the range.

3. **CompareValidator-** It is used to compare two values.

   It has the following specific properties:

Properties	Description
Type	It specifies the data type.
ControlToCompare	It specifies the value of the input control to compare with.
ValueToCompare	It specifies the constant value to compare with.
Operator	It specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, and DataTypeCheck.

4. **RegularExpressionValidator-**
    - The RegularExpressionValidator allows validating the input text by matching against a pattern of a regular expression.
    - The regular expression is set in the ValidationExpression property.

5. **CustomValidator-**
    - 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 ClientValidationFunction property.
    - The server side validation routine must be called from the control's ServerValidate event handler.

6. ValidationSummary-

- The ValidationSummary control does not perform any validation but shows a summary of all errors in the page.
- The summary displays the values of the ErrorMessage property of all validation controls that failed validation.

The following two mutually inclusive properties list out the error message:

- **ShowSummary :** shows the error messages in specified format.
- **ShowMessageBox :** shows the error messages in a separate window.

5. What is authentication and authorization in Asp.net. Define types of authentications.

Ans:

Authentication: It is nothing but validating that exist in your database and it's a proper user.

Authorization: It is nothing but accessibility that he accesses to a particular resource on the IIS website.

Types of authentications:

a. **Window-** In this methodology Asp.net web page will use local windows users and group to authenticate and authorize resources.

b. **For authentication-** This is a cookie-based authentication where username and password are stored on client machine as cookie files or they are sent through URL for every request.

c. **Password Authentication:** It is based on password website provided by Microsoft. So, when the user login with credentials. It will reach to the password website where authentication will happen. Its authentication is successful, it will return a token to your website.

6. What is the difference between Session and Application.

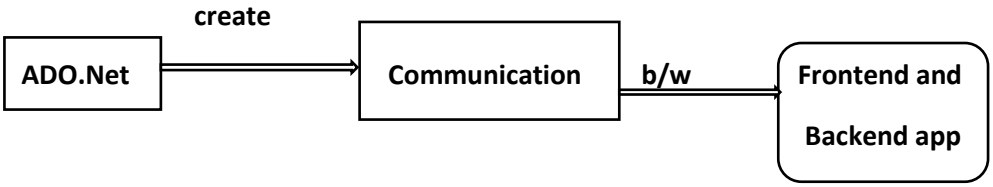
Ans: Differences between Session and Application-

Sno	Session	Application
1.	Session will allocate a separate memory for every user.	Application will allocate a common memory for all the users.
2.	Session will have expiry time.	Application will not have expiry time.
3.	Session supports webform and web garden.	Application does not supports webform and web garden.

7. What is ADO.Net?

Ans: ADO.Net-

- ADO.Net is stand for (Active X database object network enables technology).
- ADO.Net is a technology which is used to create the communication between Front-end application and Backend application.



8. What is Front-end Application and Backend Application?

Ans:

Front-end Application-

An application where end user will interact is called Front-end Application.

Ex- Facebook, irctc , icici etc.

Backend Application-

An application where user data is saved permanently is called backend application.

It can be developed by using backend tool like- **SQL Server, MySQL and Oracle etc.**

9. Explain Different types of Architecture in ADO.Net?

Ans: Types of Architecture in ADO.Net-

- **Connected Oriented Architecture**
- **Disconnected Oriented Architecture**

a. **Connected Oriented Architecture-** In this Architecture we need to create the connection to database server-

Process-

- **Create the connection**
- **Open the connection**
- **Pass the query**
- **Execute the query**
- **Close the connection**

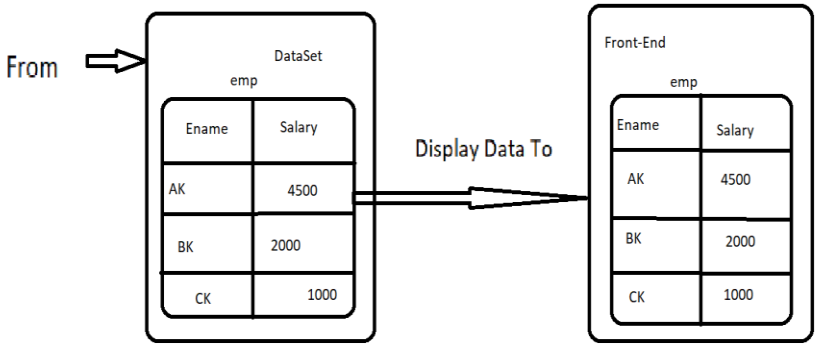
b. **Disconnected Oriented Architecture-** In this architecture we don't need to open and close the connections.
In this architecture open and close connection will be done by using SQL Data Adapter Class.

There are 3 steps to work with Disconnected Oriented architecture-

1) Design the Data bound control
2) Prepare Data Source
3) Attached the data source to data bound control.

10. What is Data Bound control and its type?

Ans: Data Bound Control- It is used to display the data from dataset to frontend.



Types of Data Bound Control:

A. **Repeater Control**
B. **DataList Control**
C. **GridView Control**
D. **ListView Control**
E. **DetailsView Control**
F. **DataPager Control**

11. What is SQL Command and define its predefined methods?

Ans: SQL Command: An SQL Command is a class of connection oriented architecture of ADO.Net who is inherits by "System.Data.SqlClient" namespace.

It Provides 3 predefined methods-

- **ExecuteNonQuery()-**
  This is used for DML operation.

- **ExecuteScalar()-**
  This is used for return single cell value and its returntype will be an object.

- **ExecuteReader()-**
  If we want to return entire table or specific value then we use ExecuteReader().

12. What is PostBack Request and IsPostBack Request?

Ans:

PostBack Request-

Whenever user will interact with webpage through controls then PostBack Request will go to server that is user click on controls like – Buttons etc.

IsPostBack Request-

It is a Boolean property which will return either true or false. By Default, IsPostBack is false.

It is used to check whether request is first or post back.

13. What is GridView? Define its properties and events.

Ans:

GridView-

Grid View is a Data Bound control which is used to display the data from data set to front end.

We can perform deleting, editing, paging and sorting operation within Grid View control.

Properties-

- Id
- AutoGenearteColumns=True/False
- AutoGenerateEditButton=True/False
- AutoGenerateDeleteButton=True/False
- AutoGenerateSelectButton=True/False
- AllowPaging=True/False
- AllowSorting=True/False
- ShowFooter=True/False
- ShowHeader=True/False
- PageSize

Events-

- RowDeleting
- RowEditing
- RowUpdating
- RowCancelingEdit
- SelectedIndexChanged
- PageIndexChanging
- Sorting

14. What are Templates? Define its types.

Ans: Templates- It is used to design the DataBound Control.

Types

- ✓ Header Template- Display the column Heading.
- ✓ ItemTemplate- Bind Dataset data column by column.
- ✓ EditItemTemplate- For Perform editing operation
- ✓ FooterTemplate- Add controls within FooterRow.

15. What are Cookies? Define its types of cookies?

Ans: Cookies-

- **Cookies** are small pieces of text that accompany requests and pages as they travel between the web server and the browser.
- Cookies are often used in modern web applications development to track user history, states, etc.
- Cookies are transported to and from the server and client machine using the HTTP protocol.

Types of Cookies-

1) Temporary Cookies/ In memory/ Non-Persistent Cookies
2) Permanently Cookies/ Out memory/ Persistent Cookies

- **Temporary Cookies-**
  When cookies are saved within the browser's temporary folder on the client's machine, such cookies are said to be **persisted**, because information saved can be retrieved at any time, so long the cookies' expiration date has not elapsed.

- **Permanently Cookies-**
  Also known as **session** or **in-memory cookies**, these are only useable while the browser is open. They are not saved locally on the client's machine.

16. Define Steps to work with cookies class?

Ans: Steps to work with cookies class-

1) **Create a cookie by using HttpCookie class.**
   HttpCookie obj=new HttpCookie("cookiename");

2) **Store the value in cookie object.**
   Cookieobjectname.value=some value;
   Obj.value="Manish";

3) **Add the cookie object to cookies collection table**
   Response.cookies. Add(cookiesname)

Syntax- To read the value from cookies-

Request. Cookie[“cookiename”].value;

17. What are the differences between Inmemory and Outmemory Cookie?

Ans: Differences between Inmemory and Outmemory Cookie-

Inmemory Cookie	Outmemory Cookie
Cookie value will store on RAM	Cookie value will store on HardDisk
The life span of Inmemory cookie until user close browser.	The life span of Outmemory cookie after expiry time.
We can’t set expiry time for Inmemory cookie.	We can set expiry time for Outmemory cookie.

18. What are the differences between Response. Redirect () and Server. Transfer ()?

Ans: Differences between Response. Redirect () and Server. Transfer ()-

	Response. Redirect ()	Server. Transfer ()
I.	It is used to redirect the user request between multiple web servers.	It is used to redirect the user request within the same web servers.
II.	It will not hide the destination URL address.	It will hide the destination URL address.
III.	Response. Redirect(“URL”)	Server. Transfer(“URL”)

19. Define Asp.net Page Life Cycle?

Ans:

ASP.NET Page Lifecycle:

In ASP.NET, a web page has execution lifecycle that includes various phases. These phases include initialization, instantiation, restoring and maintaining state etc. it is required to understand the page lifecycle so that we can put custom code at any stage to perform our business logic.

Page Lifecycle stages – The following table contains the lifecycle stages of ASP.NET web page.

i. **Page request**
ii. **Start**
iii. **Initialization**
iv. **Load**
v. **Postback event handling**
vi. **Rendering**
vii. **Unload**

Stage	Description
Page request	This stage occurs before the lifecycle begins. When a page is requested by the user, ASP.NET parses and compiles that page.
Start	In this stage, page properties such as Request and response are set. It also determines the Request type.
Initialization	In this stage, each control's UniqueID property is set. Master page is applied to the page.
Load	During this phase, if page request is postback, control properties are loaded with information.
Postback event handling	In this stage, event handler is called if page request is postback. After that, the Validate method of all validator controls is called.
Rendering	Before rendering, view state is saved for the page and all controls. During the rendering stage, the page calls the Render method for each control, providing a text writer that writes its output to the OutputStream object of the page's Response property.
Unload	At this stage the requested page has been fully rendered and is ready to terminate.at this stage all properties are unloaded and cleanup is performed.

A requested page first loaded into the server memory after that processes and sent to the bowser. At last, it is unloaded from the server memory. ASP.NET provides methods and events at each stage of the page lifecycle that we can use in our application. In the following table, we are tabled events.

ASP.NET Life Cycle Events-

Page Lifecycle events – The following table contains the lifecycle events of ASP.NET web page.

- ✓ **PreInit**
- ✓ **Init**
- ✓ **InitComplete**
- ✓ **PreLoad**
- ✓ **Load**
- ✓ **Control events**
- ✓ **LoadComplete**
- ✓ **PreRender**
- ✓ **PreRenderComplete**
- ✓ **SaveStateComplete**
- ✓ **Render**
- ✓ **UnLoad**

Page Event	Typical Use
PreInit	This event is raised after the start stage is complete and before the initialization stage.
Init	This event occurs after all controls have been initialized. We can use this event to read or initialize control properties.
InitComplete	This event occurs at the end of the page's initialization stage. We can use this event to make changes to view state that we want to make sure are persisted after the next postback.
PreLoad	This event is occurs before the post back data is loaded in the controls.
Load	This event is raised for the page first time and then recursively for all child controls.
Control events	This event is used to handle specific control events such as Button control' Click event.
LoadComplete	This event occurs at the end of the event-handling stage. We can use this event for tasks that require all other controls on the page be loaded.
PreRender	This event occurs after the page object has created all controls that are required in order to render the page.
PreRenderComplete	This event occurs after each data bound control whose DataSourceID property is set calls its DataBind method.
SaveStateComplete	It is raised after view state and control state have been saved for the page and for all controls.
Render	This is not an event; instead, at this stage of processing, the Page object calls this method on each control.
Unload	This event raised for each control and then for the page.

20. What is the difference between Hyperlink button and Link button?

Ans:

Hyperlink-

- Hyperlink button is just use to navigate from one page to another page.
- We can't post back request to the server.

Link button-

- We can post back request to the server.
- When user click on button controls, writes some logic, and want to perform some operation, it means that is a post back request to the server. In this case we use Link button.

21. What is Master Page in Asp.net? Explain How Master page work?

Ans:

Master Page-

- ASP.NET master pages allow you to create a consistent layout for the pages in your application.
- A single master page defines the look and feel and standard behaviour that you want for all of the pages (or a group of pages) in your application.
- You can then create individual content pages that contain the content you want to display.
- When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page.

How Master Pages Work-

- ✓ Master pages actually consist of two pieces, the master page itself and one or more content pages.
- ✓ A master page is an ASP.NET file with the extension. master (for example, MySite.master) with a predefined layout that can include static text, HTML elements, and server controls.
- ✓ The master page is identified by a special @ Master directive that replaces the @ Page directive that is used for ordinary .aspx pages.