* Steps to Work With Cookies:-

step 1: Create object for Http Cookie class

syntax: Http Cookie class object Name = NEW Http Cookie class ("Cookie Name");

example: Http Cookie obj1 = NEW Http Cookie ("a");

step 2: Store the data into cookie

syntax: Cookie objectName.Value = "Data";

example: obj1.Value = "10";

Set the othere properties based on Requirement

:

step 3: Add the Cookie to Cookies Collection of Response object

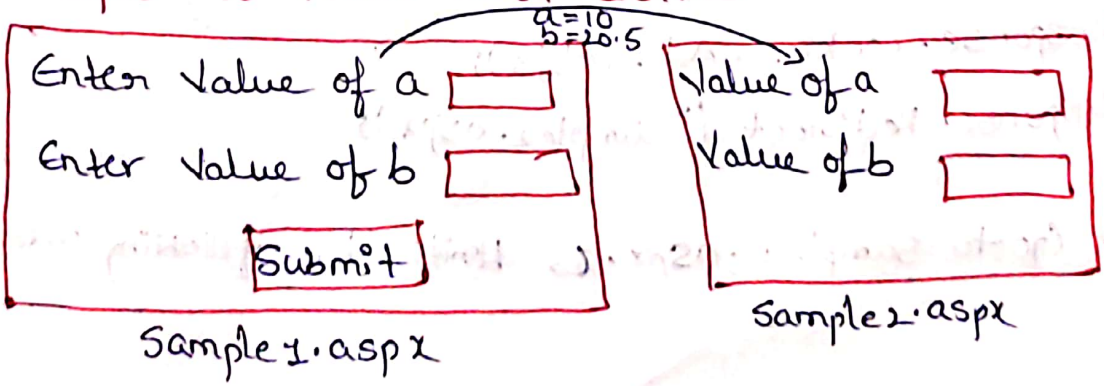
syntax: Response.Cookies.Add (Http Cookie class object);

example: Response.Cookies.Add (obj1);

step 4: Repeats the above steps for Number of Cookies.

step 5 (optional): Redirect the request to the Target Page.

* Example to Work With Cookies



Scanned with CamScanner

→ Create a NEW Website With the Name WebApp Cookies.

→ Create a NEW Web pages With the Name Sample1.aspx and Sample2.aspx.

→ Write the following Code for Submit button of Sample.aspx.

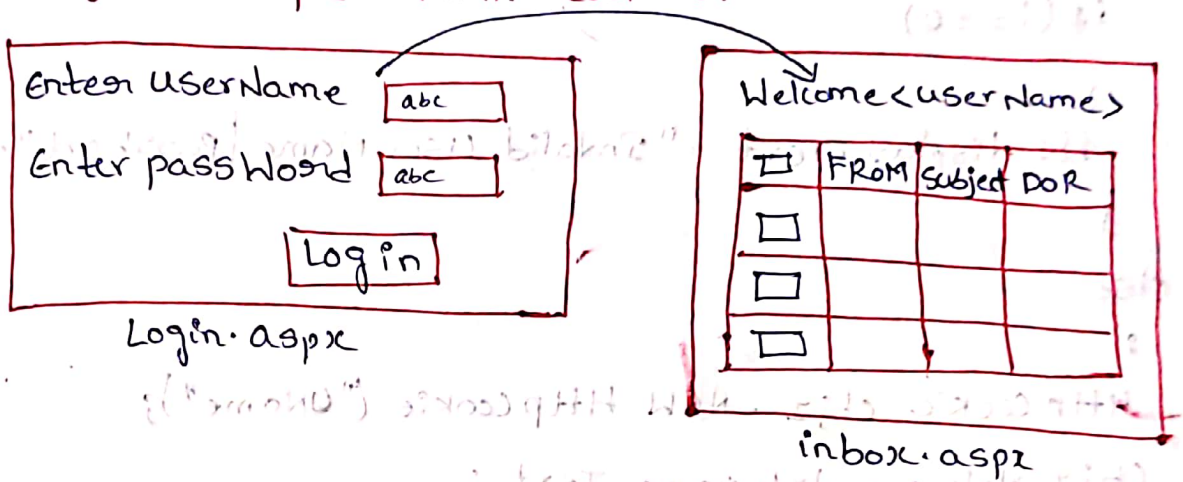
```
Protected void btnsubmit_click ( " " )
{
Http Cookie obj1 = NEW Http Cookie ("a");
Obj1.Value = txt Sample1.Text;
obj1.Expires = DateTime.NoW.Add Days (4);
Response.Cookies.Add (obj1);
Http Cookie obj2 = NEW Http Cookie ("b");
Obj2.Value = txtsample2.Text;
obj2.Expires = DateTime.
Response.Cookies.Add (obj2);
Http Cookie obj3 = NEW Http Cookie ("Institute");
Obj3.Value = "Nagesh it";
Obj3.Expires = DateTime.NoW.Add Days (7);
Response.Cookies.Add (obj3);
Response.Redirect ("Sample2.aspx");
}
```

→ Go-to Sample2.aspx.cs Write the following code.

Scanned with CamScanner

```
Protected Void page_Load ( " " )
{
txtSample1.Text = Request.Cookies["a"].Value;
txtSample2.Text = Request.Cookies["b"].Value;
}
```

* LoGin Example With Cookies:-



→ Create a NEW WEBpages With the Name Login.aspx and inbox.aspx. design both the pages. & go to Login.aspx.cs file and Write the following code.

```
Using system.Data;
Using system.Data.SqlClient;
Protected Void btnLogin_click ( " " )
{
String sqlconstring = "Server = ; UserId = ;"
Sql connection con = NEW Sql connection (sqlconstring);
String Query =" Select count(*) From Users Where
                UserName = @P1 and passWord =@P2";
Sql command cmd = NEW Sql command (Query, (on);
Cmd.command Type = command Type.Text;
```

Scanned with CamScanner

```
cmd.parameters.AddWithValue("@P1", txtuName.Text);
cmd.parameters.AddWithValue("@P2", txtpassWord.Text);
Con.open();
int i = (int)cmd.ExecuteScalar();
Con.close();
if(i==0)
{
lbl display.Text = "Invalid. User Name | passWord";
}
else
{
Http Cookie obj1 = NEW Http Cookie ("UName");
Obj1.Value = txtuName.Text;
if (chk Remember.checked)
Obj1.Expires = DateTime.Now.Add days (7);
Response.Cookies.Add(obj1);
Response.Redirect("Inbox.aspx");
}
```

→ Go to inbox.aspx.cs file Write the following code.

```
Protected void page-load (          "          )
{
lbl display.Text = "Welcome" + Request.Cookies
                              ["UName"].Value;
```

Scanned with CamScanner

→ Advantages using cookies:-

1. Simple implementation.
2. No server resources are required.
3. Compared with query string cookies can store more amount of data.

→ Disadvantages using cookies:-

1. Security concerns. as cookies stores the data in plain text format cookies can't be used to transfer sensitive information like password etc.
2. Size limitation:

Though every cookie can transfer 4 Kb of data We can't use cookie to transfer large amount of data.

3. Cookies are browser dependent. if user is blocking the cookies or deleting the cookies as a developer We cannot restrict the user.

* Working With Sessions in Asp.net:-

→ A session is the memory created for every client at server side. At first request of the client Web server will allocate memory for the client, this memory is known as session and the unique Id allocated for this session is known as session Id. The data related to that particular client or user will be stored in this session.

Scanned with CamScanner

→ One user of client session is entirely different from the other user of other client session.

→ Once session is created the request and response from them always will travel using the session Id only.

→ The webserver will identify the first request of the client is if the request is coming without session Id.

→ A session will be destroyed or killed in two senarios.

1. When session time out period completes
2. When developes kills the session programmatically using session.abandont method.

→ The default time out period of a session is 20min

Scanned with CamScanner

→ One user of client session is entirely different from the other user of other client session.

→ Once session is created the request and response from them always will travel using the session Id only.

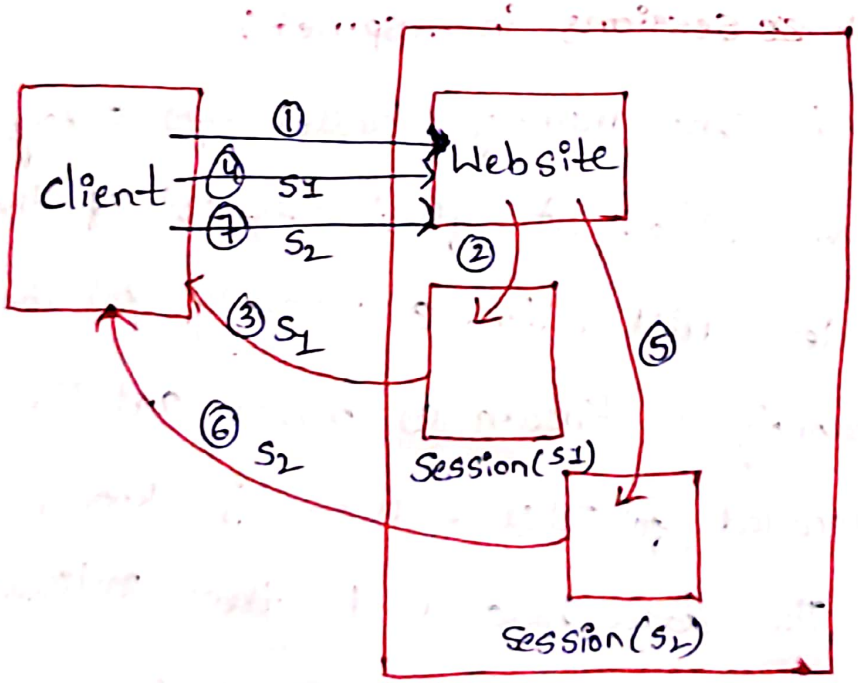
→ The webserver will identify the first request of the client is if the request is coming without session Id.

→ A session will be destroyed or killed in two scenarios.

1. When session time out period completes
2. When developes kills the session programmatically using session.abandant method.

→ The default time out period of a session is 20min

* Working Nature of session in Asp.net When there is no data stored in Session.



Scanned with CamScanner

step1:- Client sends first request to the Web Server

step2:- Client request is processed, memory allocation is done for the client and unique Id is allotted to the client, The Memory is known as Session and the Id is known as Session Id.

step3:- Response to the client is delivered along with session Id.

step4:- client sends next request with the session Id.

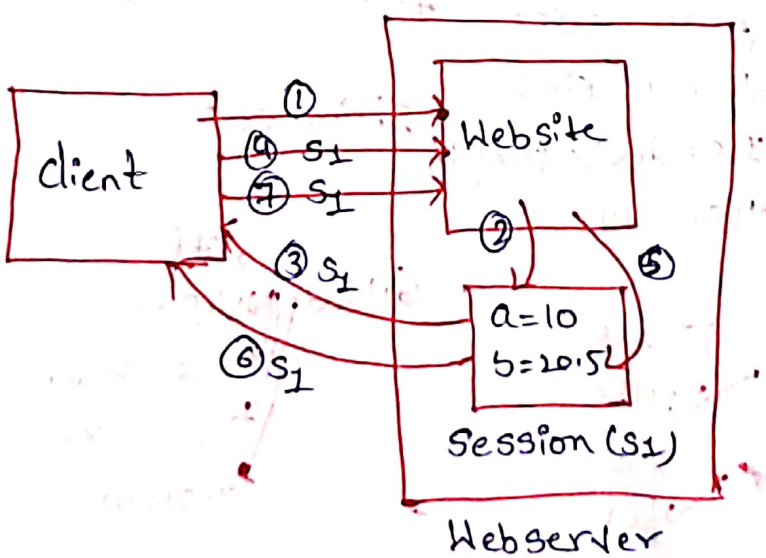
step3:-(i) once response is delivered to the client the session at server is destroyed.

step5:- Server will create new session for the client request with NEW session Id.

step6:- Response is delivered to the client with NEW session Id.

step7:- Client sends next Request with NEW session Id.

* In case data is stored with in the session Working Nature of the session will change like in following Senario.



Scanned with CamScanner

Step1: Client sends first request to the Webserver.

Step2: Server will process the client request, Creates the session, stores the user data with in the session and allocates session Id

Step3: Response is delivered to the client along with session Id.

→ Once Response is delivered to the client session will not be destroyed at server.

Step4: client sends next Request along with session Id.

Step5: Client Request is processed and stored in same session with same session Id.

Step6: Response to the client is delivered with same session Id.

Step7: client will send the next request with the same session Id. This process is repeated until the session is destroyed.

* Properties with session in Asp.net :-

1. Contents
2. Cookie mode → Auto detect, Use cookies, Use device profile, Use URL
3. Count
4. Is Cookie less → True, false
5. Is New session → True, false
6. Mode → Off, In proc, State server, Sql server, Custom
7. Session Id
8. Time out
9. Mode

Scanned with CamScanner

1. Content :-

This property is used to get the contents of the sessions. i.e. It is returns to the reference to the session content.

2. Cookie Mode :-

used to set or get the required cookie mode value for the session.

→ All cookie mode values are available in an Enumeration. Known as http cookie mode.

3. Count :-

This property will return the count of elements or items available with in the session.

4. Is Cookie less :-

When set to false session can be worked with cookies When set to true session can be worked without cookies.

5. Is New Session :-

This property will return true if session is the new otherwise will return false.

6. Mode :-

used to set or get the required session mode where the session is to be created and maintained.

→ default session state mode is Inproc

Scanned with CamScanner

7. Session Id :-

This property is used to get the Id of the session that is created.

8 Timout :-

Used to set of get the required timeout Period for a session.

Methods With Session :-

1. Abandon ( )

This Method is used to Kill the Session Programmatically.

Scanned with CamScanner