* Working With Master pages, in Asp.net :-

→ A Master page is similar to Web user Control. Which will provide reusability facility for the design of the web pages.

→ To Create Every Masterpage will have a default Extension of .Master.

(Master page hee) ×

(To Create a web, user control) ×

→ To Create a Master page control we use web-forms master page template in VS.net.

→ Master page will contain HTML, Head, body tags.

→ Every Master page will be Executed only for the 1st request of the client.

→ Every Master page we create is a class which is Inherited from masterpage class of system.Web.UI. namespace

* Example with master page :-

Web App Master

- → Home.aspx
- → NEWS.aspx
- → Aboutus.aspx
- → contactus.aspx
- → Navigate.Master

→ Create a NEW Website With the Name WebApp. Master.

* Creating Master page:-

→ Go to Solution Explorer select the Solution, click With right Mouse button, click on Add, click on NEW Item, select Web forms ~~pa~~ Master Page template. Type the master page name Navigate. Master. click on OK.

→ Then Create hyper links In Master page.

* Creating content pages:-

→ Go to Solution Explorer, select the Solution, click With Right Mouse button, click on Add, click on NEW Item, select Web forms ~~App~~ Master page Template. Type the page name With Home.aspx click on add. select Master page Navigate.Master, click on ok. perform the design of Home.aspx With in the Content place holder. Similarly Create the others Content pages. Run the application and check.

* Customizing the Master page from the Content page:-

→ Create a New label in Master page. With the Name lbl display.

→ Go to Home.aspx.cs Write the following code.

```
Protected Void page-Load (          ''          )
{
  Label L1 = (Label)Master.Find Control ("lblDisplay");
  L1.Text = " Welcome To My Website";
}
```

→ Similarly same code we can write the News.aspx.cs, contactus.aspx.cs. ~~aboutus.aspx.cs.~~

→ Code for aboutus.aspx.cs.

```
LinkButton LB1 = (LinkButton)Master.FindControl
                 ("LinkButton2");
LB1.Visible = false;
Label L1 = (Label)Master.FindControl("lbldisplay");
L1.Text = "Welcome to About us page";
```

* Differences b/w Webuser control And Master page:-

S.No.	Web user control	Master page
1.	User control will have a default extension of .ascx.	Master page will have a default extension of .Master.
2.	User control does not contain <HTML><HEAD> <BODY> Tags.	Master pages contain <HTML> <HEAD> <BODY> Tags.
3.	To Create a user control we use Web user control Template in vs.net.	To Create a Master page we use WebForms Master page template in vs.net.
4.	A user control will be executed for every client request	A Master page will be executed for the first request of client only.
5.	Execution of user control is slow.	Execution of Master page is fast.

6.	User control can be used to implement partial page cache.	Master page can be used to implement partial page cache.
7.	User control cannot be customized from other web pages.	Master page can be customized from the content pages..
8.	To consume the user control with in the web page we need to register the user control with the web page using <%@ Register → page directive.	To use the Master page with in the content pages it is not required to register the Master page rather use Master page file attribute with the web page directive.
9.	Web user control will be created by using <%@ - control .... %> directive	Master page will be created by using <%@ Master...%> Directive.
10.	Any web user control is inherited from user control class of system. Web. UI Namespace.	Any web page is inherited from Master page class of system. Web. UI Namespace.
11.	Supports in all versions of Asp.net (1.0, 1.1, 2.0, 3.0, 3.5, 4.0, 4.5, 5.0)	Does Not support in all versions of Asp.net, included in 2.0 version of Asp.net.

Sno	Web User Control	Master Page
01	User control will have a Default extension of .ascx	Master page will have a Default extension of .master
02	User control does not contain <Html> <Head> <Body> tags	Master page contains <Html> <Head> <Body> tags
03	To create a user control we use WebUserControl template in VS.Net	To create a Master Page we use WebForms MasterPage template in VS.Net
04	A user control will be executed for every client request	A Master Page will be executed for the first request of client only
05	Execution of user control is slow	Execution of master page is fast
06	User control can be used to implement partial page cache	Master page can be used to implement partial page cache
07	User control can not be customized from other web pages	Master page can be customized from the content pages

06	User control can be used to implement partial page cache	Master page can be used to implement partial page cache
07	User control can not be customized from other web pages	Master page can be customized from the content pages
08	To consume the user control with in the web page we need to register the user control with the web page using <%@Register —> Page directive	To use the master page with in the content pages it is not required to register the master page rather use <u>MasterPageFile</u> attribute with the Web page directive
09	Web User Control will be created by using <%@ Control ... %> Directive	Master Page will be created by using <%@ Master ... %> Directive
10	Any Web User Control is inherited from <u>UserControl</u> class of <u>System.Web.UI</u> Namespace	Any Master page is inherited from <u>MasterPage</u> class of <u>System.Web.UI</u> Namespace
11	Supports in all Versions of ASP.Net (1.0,1.1,2.0,3.0,3.5,4.0, 4.5, 5.0)	Does not support in all Versions of ASP.Net, Included in 2.0 version of ASP.Net

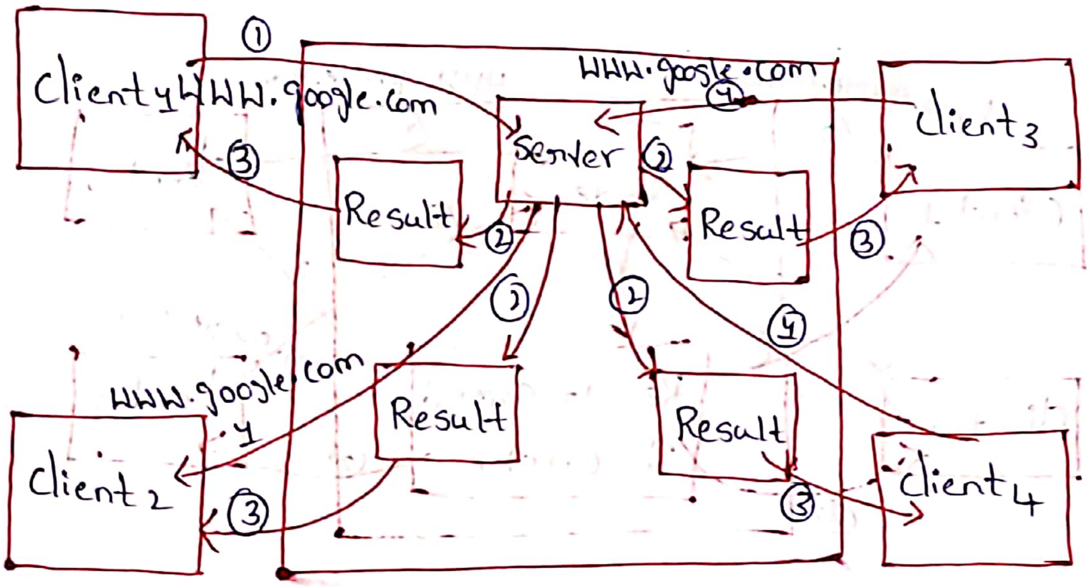
* Catche :- In Asp.net :-

→ Catche in Asp.net is similar to catche memory In hardware.

→ Catche is used to store the required data temporarorily with in the memory for required Period of time.

→ Catche will help to improve the application performance

* Senario Without Catche Implementation :-



Step1 :- Client send Request for a page to the server

Step2 :- Server will process the client Request and generates the Result.

Step3 :- Result is delivered to the client

→ Once Result is delivered to the client, the Execution result of server side will be destroyed.

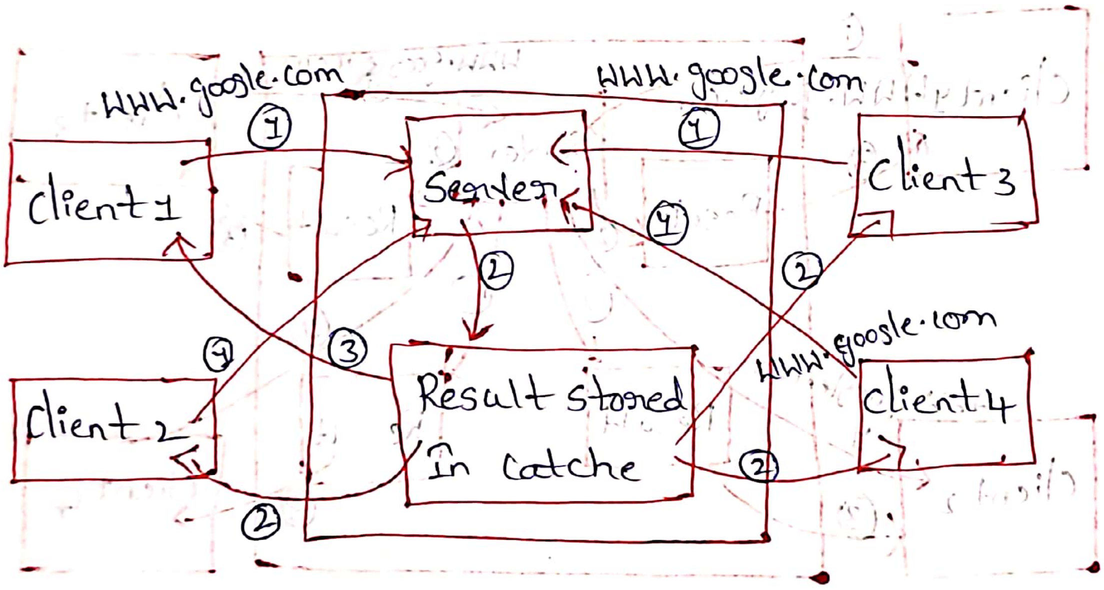
→ for Every other request of the same client or different client for the same page request will be

Represented again.

→ This will increase burden on the server and will reduce the application performance because though there are no modifications in the page, page Request is reprocessed again and again unnessarily.

→ To overcome this drawback catche is used in Asp.net.

* Senario With Catch Implementation:-



Step1:- Client sends request for a page to the Server

Step2:- client Request is processed at server And the

Step3:- Result is stored in catche Memory.

→ Response is delivered to the client

after delivering Response to the client, result is not destroyed, rather result is stored in catche.

→ If same client or some other client sends requests for the same page, Then Request will not be Reprocessed, rather Catche ~~a~~ result is delivered to the client, This Will Make application Execution at faster rate.

*** Types of Catche In Asp.net:**

→ Asp.net Will supports Three Types of catche

1. Data catche.
2. Page output catche.
3. Partial page catche | Fragment catche.

1. Data Catche:-

→ Data catche is used to (implem) store the Required data With in the catche and can be used to access the catche accross Multiple pages.

→ Catche Will store the data with object type.

Syn:- Catche ["Variable Name"] = Data;

```
Ex:-  Cache ["a"] = 20;
      Cache ["b"] = 30.5;
      Cache ["c"] = "Welcome";
```

*** Example With Data Catche:-**

→ Create a New Website With the Name WebApp Catche

→ Create two pages With the Name Sample1.aspx, Sample2.aspx.

→ Design of Sample1.aspx.

Enter Value of a [ ]

Enter Value of b [ ]

[Submit]

→ Design of Sample2.aspx

Enter Value of a [ ]

Enter Value of b [ ]

→ Code for Sample1.aspx.cs

```
Protected void btnsubmit_click (        ''            )
{
Cache ["a"] =txtSample1.Text;
Cache ["b"] =txtSample2.Text;
Response.Redirect ("sample2.aspx");
```

→ Code for sample2.aspx.cs

```
Protected Void page_Load (    ''        )
{
txtSample1.Text = Cache ["a"] . To string ();
txtsample2. Text = Cache["b"] . To string ();
}
```