ASP. NET

New Notes

By

Sudhkar Sharma

Naresh Technology

21/05 Windows Applications Vs Web applications:

feature	Window Application	Web Application.
1) Build	They are easy to build	They are complex to build, requires serval impement-ations.
2) Installing	To installed on every machine	To be installed only on server.
3) Upgrade	Must be upgraded on every machine	Upgraded only on server.
4) framework	Requires framework on every machine	Requires framework only on server.
5) Catastropic failures	Leads to the failure of individual machine	Leads to the failure of entire system.

★ Web Terminology:

1. Network
2. Types of network
3. Web
   - Tim Berners Lee   web is part of internet
   - IETF
   - W3C (HTML5)
   - W3C ~~WAT~~ WHATWG
4. Web server:
   - Microsoft IIS
   - IBM Lotus.
5. Web site - is a virtual directory in webserver
   - User cannot interact
6. Web Application - User interacts
7. Blog
8. wiki

1) Network : A computer network comprises of group of computers Connecting with each other for sharing of information and resources.

2. Types of networks :
Computer networks are classified into three major types based on their range and capacity. They are -
a) LAN - (Local area network)
b) MAN - (Metropolitian area network)
c) WAN - (Wide area network)

3. About Internet : It is a wide area network that connects computers all over the world.

4. Web : 1) Web is a portion of internet.
2) The concept of web was introduced by Tim Berners Lee
3) Later web was developed by IIEF. (Internet Engineering task force)
4) W3C : World wide web consortium. It maintains the stand- -ards of web.
5) The latest version of HTML is being developed by 2 groups.
1) W3C
2) WHATWG - Web hypertext application technology work group.

5) Web Server : A web server resembles both hardware and soft- -ware. It satisfies the request of clients by sending and receiving the data. The popular webserver softwares are
1) Microsoft IIS (Internet information services)
2) Apache Tomcat → PHP,
3) JBOSS →
4) Light PGD
5) IBM Lotus - offline server.
6) Web sphere etc.

5) Website: It is a virtual directory in the webserver. A typical website will not allow any interactions.

Ex. www.nareshit.in.

6) Web Application: It is similar to a website but allows interaction with users.

Ex. www.bookmyshow.com
irctc.gov.in

7. Web page: Web page are of types:

~~1) Static~~

Information in a website is stored in the form of HyperText documents known as webpage. They are classified into two types:

a) Static page

b) Dynamic page.

a) Static page: The pages that are predefined in the server and are ready for access are known ~~as~~ as static pages.

Ex. Home.html
index.html

When we export to server - htm
When we directly save it to server - html.

b) Dynamic pages: The pages that are generated as a response to the client request are known as dynamic pages.

Ex. ~~Res~~ results. ~~of~~ aspx   .NET → .SPX
cricket.asp
movies.php
pnr.jsp.

8) URL: http - Normal protocol
https - secured protocol
↓
facebook : https://facebook.com.
Bank sites.

URL: (Uniform resource locator)

It is a virtual path generated by a webserver in order to access the resources of a web site.

blogspot.com

Ex. $\underbrace{\text{http}}_{\text{Protocol}}$:// www. $\underbrace{\text{nareshit . in}}_{\text{Domain name.}}$

7) Protocols: Computers in the network communicate with each other by using a set of rules known as protocols. Web uses the protocols in http

a) HTTP

b) HTTPS → Secured.

★ Blog : (Web-log)   Micro blogging - Twitter

Blogs are journals ~~of~~ on internet usually published by individual users and updated periodically. If many users post their personal information on a single then it is ~~cat~~ refered as microblog.

Ex. Twitter.

★ HTML -5   IIS → 6

★ JavaScript

★ ASP.

Requirements of designing web applications, page

★ Webpage - is having two paths -

1) Virtual →

2) Physical → Cannot visible to the user.

★

★ WIKI (Quick) :- A WIKI allows any user to ~~g~~ edit its content.

Ex. www.wikipedia.com

www.imdb.com.

★ Web debugger : A web debugger tracks ~~a~~ performance of your ~~p~~ webpage. Which includes the request and response time, files accessed, bytes received, etc.

Ex. 1) fiddler

2) Internet Explorer debugger (F12)

22/05 Note: ~~Req~~ S/W Requirements to develop an web Application; fiddler

1. HTML
2. CSS
3. Client side Script
   Javascript / Jquery, Angular Js, BackBone js, KnockOut Js. –
4. Server side script
   ASP, ASPX, JSP, PHP
5. CMS (Content management system)
   Dream, Weaver, Micromedia Home suite, Visual Studio, Telerik Developer express, etc.
6. Web Debugger: fiddler, Internet explorer debugger.
7. UI Designing Tools:
   Photoshop, Flash, GIF Animator
8. Database
   SQL server, Oracle, MS Access, MYSQL, Sybase, DB2 ...

★ Locating web Server:
1) Open windows control panel.
2) Switch to ~~A~~ large icon view
3) goto Administrative tools and look for IIS (internet information services Manager). If it is there then its ok.
4) If it is not there we have to add it.

★ Adding IIS to our computer.
1) Open control panel
2) program & ~~Fea~~ features → then select the turn windows features ON or OFF
3) Select checkbox for IIS. then click OK.

Shortcut to open server:
Run → Inetmgr.
IIS - Manages Applications.

★ Creating a new website on IIS.

1. Open IIS. i.e Run → inetmgr ↵
2. Expand local computer
   (http://localhost)
3. Expand "sites" folder.
4. ~~Wa~~ Right click on default website.
5. Select the option add virtual directory
6. Give an alias name for website.
   Ex. snapdeal.
7. Select a physical path ~~and~~ i.e ~~C:\Snapdeal~~
   D:\ Snapdeal website and $\boxed{ok}$.

Note: The default location of website on IIS is "c:\netpub\wwwRoot"

28/05

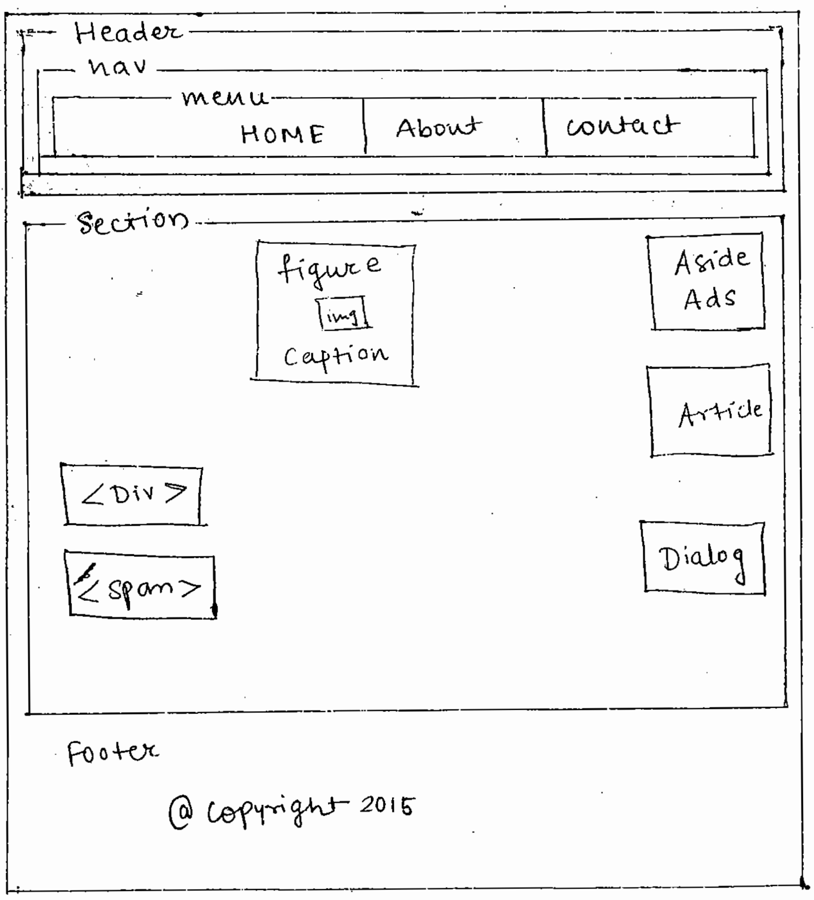
★ Static pages:

1. A static page is predefined in the webserver and it is ready for access.
2. The language HTML is used to design static pages
3. HTML is a presentation language.
4. HTML comprises of a set of commands enclosed in "< >". known as elements or tags.
5. The following are basic html elements.

Element	Description
1. <!DOCTYPE html>	It specifies that page is using HTML 5.
2. ~~HTML~~ <html>	It indicates the language used in webpage < ~~html~~ >.
3. <head>	Describes the head section of a page, which comprises of title, link and meta.
4. <title>	It describes the title to be displayed in the browser title bar.

★ HTML file body elements.

Element	Description.
1. Aside	Contains information that is not relevant to website. Ex. Ads.
2. Article	Publishesh information about website.
3. Dialog	Allows interaction with the user.
4. Header	Header section of page
5. Footer	Bottom Margine of page.
6. Section	Content to be displayed in body.
7. nav	Navigation area
8. Menu	Menu inside the navigation area
9. figure	Tinnage image with caption.
10. Span	Container ~~with~~ ~~line~~ ~~break~~.
11. div	Container with line break.



Source code :

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<header style = "background - color : red ; color : white ;
 text - align : centre ">
 <nav>
 <menu>
  Home <span> |</span>
 About <span > | </span>
  Contact <span> | </span>
 </menu>
 </nav>
 </header>
 <section>
 < aside> Ads come here </aside>
 < article> Special offers </article>
<dialog> post your comment </dialog>
 <figure style = " background - color : yellow ">
 <img serc = "g.jpg" width = "100". height = "100" >
<caption > Figure 1.1 </ caption>
 </figure>
 <div > Welcome to Naresh IT </div>
 </section >
 <footer style = " background - color : red; color : white ; text-
   align : centre " >
   &copy copyright 2015
  </footer >
  </body >
  </html>
```

<link>	links the external files ~~link~~ like CSS, ~~JS~~ JS, Shortcut icons, etc.
<meta>	It describes the metadata of your website, which is used by SEO (Search engine optimizatn)

SignalR.

★ Creating first static page with a favourite ICON.

1. Open mspaint → set page size to 16X16 pixels. Drop an icon and save the file in your website physical path by name "favicon."
   (D:\Snapdeal\favicon.png)

3. ~~to~~ Open windows command prompt. (Run→cmd)

4. Change to your website location.

```
C:>CD  D:\Snapdeal.
```

5. Rename the ~~icon~~ favicon into icon file.

```
D:\>rename  favicon.png  favicon.ico.
```

6. Open notepad and type the following code

```
<!DOCTYPE html>
<html>
<head>
<title>  Snapdeal | Hyd </title>
<link rel = " Shortcut icon "
      href = "favicon.ico">
</head>
</html>
```

7. Save the file in your website folder by name "index.html".

8. Open browser any... Chrome, IE, mozila, ...etc and type following URL

   http://localhost/snapdeal/index.html.

★ Creating a static page with meta.
Meta represents metadata; It contains information about your application.
SEO use your website meta contain to search and summarize your website information in the search results.
Html provides the following attributes for <meta>eleme--nts. in <head> section.

a) Charset
b) Name = Keywords
c) Name = Description
d) Http-equiv = Refresh

★ Source code : index.html.

Code ⇒
```
 <!DOCTYPE html>
 <html>
 <head>
<title> Snapdeal | HYD </title>
<link rel = " shortcut icon".
      href = "favicon.ico">
<meta charset = "utf=8">
<meta name = "Keywords">
  Content = "best online shopping Buy Online">
<meta name = "Description"
  Content = "best shopping site for electronics,
    furniture, cars,...>
<meta http-equiv = "refresh".
     content = "3">
 </head>
 </html>
```

★ Client side script :

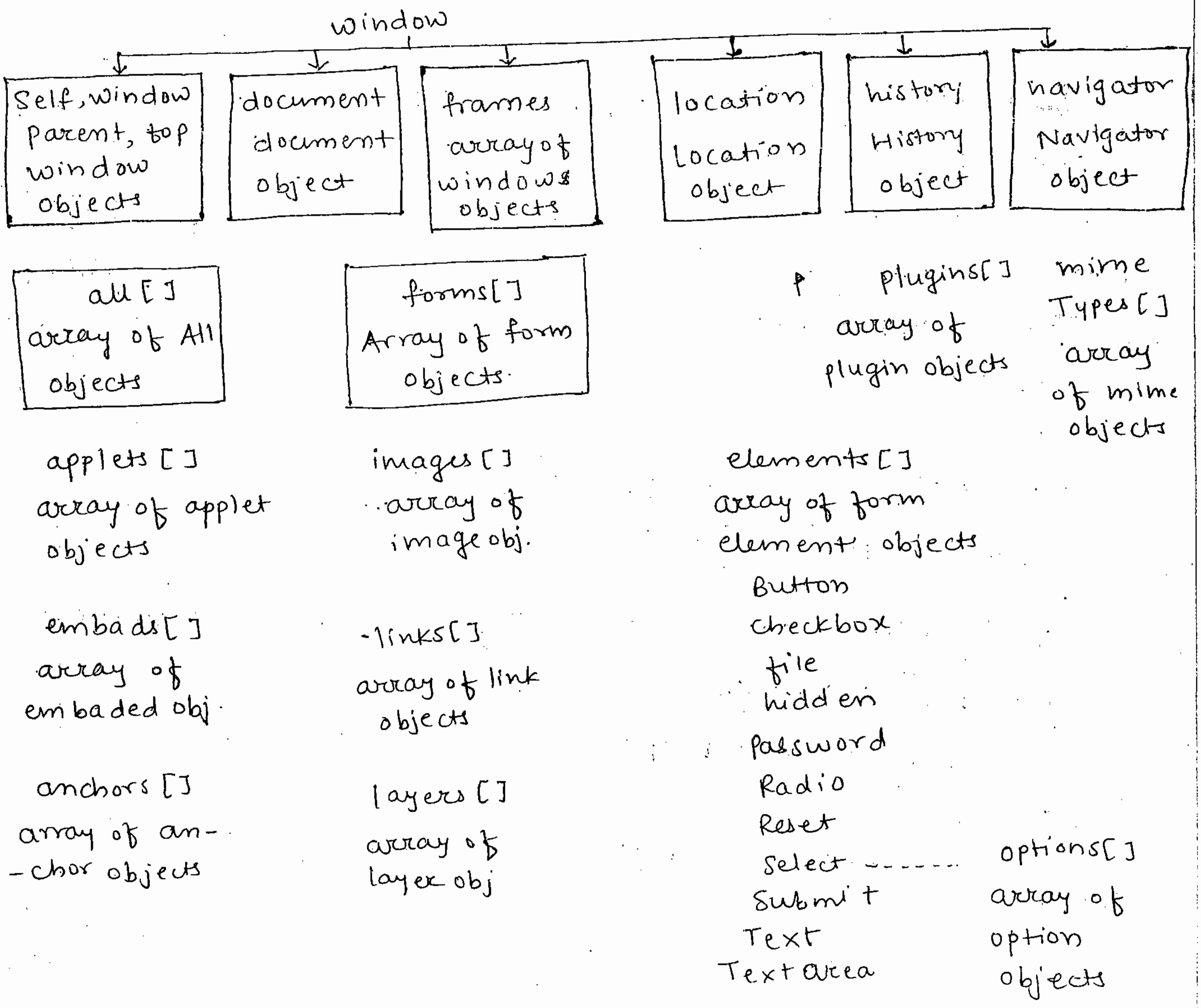
A client side script provides set of statements that are executed on the client machine. This will reduce the burden on server.

Client side scripts are mostly used for client side validations and client side interactions. The commonly used client side scripts are JS, jquery, angular JS, Backbone JS, knockout JS, etc.

★ Using Javascript for clientside interactions :

Javascript is an object based programming system (OBPS) that provides a collection of build-in objects, which controls client side interactions.

★ Javascript Document object model (DOM)


window
Self, window parent, top window objects
document document object
frames array of windows objects
location Location object
history History object
navigator Navigator object
all [ ] array of All objects
forms [ ] Array of form objects
plugins [ ] array of plugin objects
mime Types [ ] array of mime objects
applets [ ] array of applet objects
images [ ] array of image obj.
elements [ ] array of form element objects
Button
checkbox
file
hidden
Password
Radio
Reset
Select
Submit
Text
Textarea
options [ ] array of option objects
embads [ ] array of embaded obj
links [ ] array of link objects
anchors [ ] array of anchor objects
layers [ ] array of layer obj

Example: Javascript Window Object to print page.

```
<! DOCTYPE html>
<html>
<head>
</head>
<body>
<form>
<h1> Click print Button to print this page </h1>
<input type = "Button" name = "btnPrint" value= "Print"
 onclick = "window.print ()" >
</form>
</body>
</html>
```

★ Client side validations using Javascript :-

Validations are required in webapplications to ensure that contradictonary and unautherized data is not get stored into the db.

Ex.
```
<! DOCTYPE html>
<html>
<head>
<script type = "text/javascript" >
function RegisterClick()
{
 var mobile = document.frmRegister.txtMobile.value;
 var p = /\+91[0-9]{10}/;
 if (mobile.match(p))
 {
 document.write ("Registered Successfully...");
 }
 else
 {
 document.getElementById("msg").style.color = "red";
document.getElementById ("msg").innerHTML = "Invalid Mobile";
}
}
</script>
</head>
<body >
< form name = "frmRegister">
Mobile Number :
<input type = "text " name = "txtMobile" placeholder = "eg :
+91 0000000000 ">
< spanid = " msg "></span>
<br>
<input type = "button"
name = "btn Register "value = "Register " onclick = "RegisterClick()">
</form>
</body>
</html>
```

ASP.NET
(Active Server Pages)

- Asp.Net is a server side technology
- It satisfies the request of clients by sending and receiving the data.
- Technically Asp.net is a framework that provides set of classes to build rich interactive and responsive web application.
- The classes of ASP.NET framework are defined by the library System.Web.UI.

★ ASP Vs ASP.NET

ASP	ASP.NET.
1. It is microsoft's earlier server side technology.	1) It is microsoft's new Server side technology build on .NET framework
2. It has no language, uses VB as server side language	2. It have its own languages, uses all ~~..~~ .NET framwork languages like C#, VB, etc.
3. It doesn't have its own controls, uses HTML controls	3. It have a huge heap of controls and doesn't require HTML controls.
4. It uses inline documentation where code and design are in same page	4. It supports inline and code behind technique where code and design are present in different pages.
5. Not ASP.NET compact	5. It is fully ASP compact

★ Features of ASP.NET.

1) Buit on .NET framework.
2. Simple programming model
3. Multibrowser support
4. XCOPY deployment
5. XML configuration
6. Debugging
7. Extensibility (Loosely coupled and extensible Architecture)
8. Seperation of code and UI
9. security
10. ASPX, ASP side by side.

★ Drawbacks of ASP.NET.

1. Doesn't support Test Driven Application development
~~2.~~ (Unit testing is not possible)
2. Doesn't support complete HTML
3. Applications are heavy and Not light weight
4. lots of server side interaction.

★ Test driven Application development :

★ Solⁿ introduced by microsoft to overcome these problems, is :
~~MVC~~ ASP.NET MVC - model-View-Control.

★ What's New in ASP.NET 4.5 ?

1. Bundling and Minification :
~~Bundling~~ It is reduceing the no. of lines in coding by changing the logic.

2. Rounting : Youtube in india Browser asks you ~~for~~ to access ~~your~~ location . ~~get your~~ locatn related Advertisements.

3. Bootstrap : Website which provides hundred's of styles. ~~for~~ designing. It is time saving

4. Signal R : Without refreshing a page we get a message

5. Open ID - Identity .
Whenever we want to ~~you~~ use a website like a tutorials or something but the condition is you have to register and the person registering will get ~~to the updates of~~ this tutorials then we go to register.
But instead of registering we can login with facebook or google+ . That is Open ID.

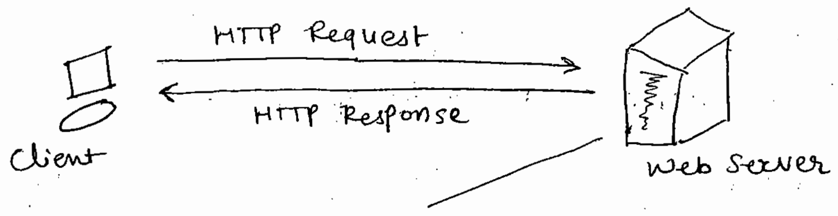
6. WEB API . 7. Facebook Application.

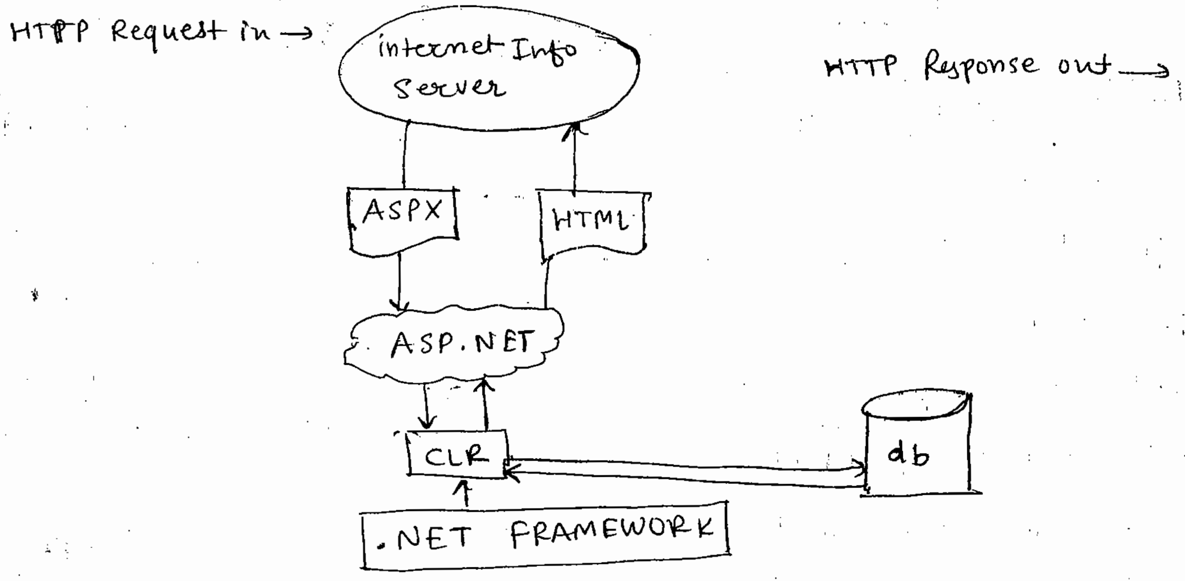
★ Whats new in ASP.NET 4.6 ?

1. ASP.NET VNext
2. Cloud Computing
3. Rosyln Compiler : without recompiling ~~a~~ ~~po~~ the code it refreshes a pages
4. Side by side execution :
5. "Monaco" online Visual Studio Editor
6. Background Garbage collector.

www.channel9.msdn.com
www.asp.net

28-05 ASP.NET architecture :-





.VB or .CS HTML → Razor Engine
.ASPX → ASPX engine.

★ ASP.NET Page contains :

1. Directives - mandatory
Describes whether it is page, master control and what it does. @ → Reference symbol.
Mandatory directive = @page. → It indicates you are designing a web page.

★ Attributes of directives:

I. language : C#, VB → It will ~~say~~ tell which language we are going to use

2. Culture :

ASP.NET Page structure:

- ASP.NET pages are controlled by using various of types of View engines like ASPX, Razor, Spark, etc.

ASPX engine recognizes the pages with extension ASPX or ASCX (for controls).

A Razor engine recognizes the pages with extension "cshtml" or "VBhtml".

A typical ASPX page comprises of following elements:-

1. Directive
2. Code declaration block → int i;
3. Code rendering block → i=10;
4. Server side commands
5. XML commands
6. Server side Controls
7. HTML controls
8. User controls
9. Server side includes
10. Literal text

1) Directives: The directives defines type of content, which includes web forms, control, master pages, Application files, etc.

Directives provide a set of attributes that are ~~using~~ used to control the behaviour of a page or application.

ASP.NET supports the following directives:

1) @page	6) @Security
2) @control	7) @OutputCache
3) @master	8) @Imports, etc
4) @Application	
5) @Register	

1) @ page: The page directive defines attributes for a webform

Attributes	Description.
1. language	Indicates the page language for codebehind i.e, C# or VB.
2. Codebehind	Defines the name of codepage for your design page. (home.aspx.cs)
3. Inherits	Specifies the class name of codepage.
4. Autoeventwireup	Events will wireup with control if ~~send to~~ set to true
5. Culture	Sets page culture.

Syntax: <% @ page Language = "C#" CodeBehind = "home.aspx.cs" Inherits = "Home". AutoEventWireUp = "True". Culture = "en-In" %>

~~29~~29/05
★ ASP uses aspx ~~$~~ engine therefore the extension is .aspx.

★ Server side code must be write in <% %>

★ <asp. TextBox ID =
↓
tag prefix

★ viewstate: Run at "server", It is like memory. HTML do not have viewstate

★ in Descriptor trace = ~~boolean~~ True

★ Creating a dynamic page and hosting on IIS.

1. Create a new website on IIS by name ~~asp~~ AspProject.
2. Select the physical path for website as "D:\AspProject"
3. Open notepad application and type the following code

```
<%@ page Language = "C#" %>
< !DOCTYPE html>
~~<hea~~
<html>
< head > </head>
<body>
< div >
Today: <% Response.Write (DateTime.Now.ToString()); %>
</div>
< /body>
</ html >
```

4. Save the file in website physical path by name "welcome.aspx".
5. Open browser - IP, Chrome and type the following URL:

   http :// localhost /Asp Project /Welcome.aspx.

★ ASP.NET Server controls :-

A typical ASP.NET application may contain the following types of controls:

1. HTML control
2. ASP.NET Server controls
3. Web user controls.

The ASP.NET Server side controls are ~~divi~~ derived from the assembly "system.web.UI.WebControls".

★ Creating ASP.NET Server control:

1. All server controls in a webform must be placed in the "form" tag.
2. Every server control must have following attributes :
   1. ID
   2. Runat
3. Server control will have a "tag prefix" and "tagname" with attri-butes.

Syntax:

```
<asp.Texbox · id = "txtName" · runat = "server">
</asp:TextBox>
```

Tag structure:

```
<TagPrefix : TagName · attributes >
</TagPrefix : TagName >
```

★ A webform can have multiple forms

★ We can end Server control with /> if there are not chain control.

Ex.:
```
<%@ page Language= "C#"%>
  <!DOCTYPE html>
  <html>
  <head> </head>
  <body>
  <div>
  <form id ="frmWelcome" runat = "server">
  HTML control :
  <input type="text">
  <br> <br>
  ASP control :
  <asp: TextBox ·id = "~~txtBo~~ "txtName"
```

```
runat = "server" />
<br>
<asp:button  id = "btnSubmit"
 text = "Submit"  runat = "server" />
</form>
~~</body>~~ </div>
</body>
</html>
```

★ / sender is a object
lb1Text → label. , GET /post request : !Page.IsPostBack

★ Page events :
A typical ASP.NET Page comprises of several events that specify the actions to be performed in various situations. ~~w~~ however the default event for page is "Page_Load"

★ Request types :-
Every ASP.NET Page responds to two types of request :
a) GET
b) POST
The "get" request controls the page actions whenever client requests the page for the first time from server.
The "post" request controls the page actions whenever client post data to server.
You can identify the request type of a page by using the property "Page.IsPostBack".

* Writing events for page:-

If you are using inline documentation then the events of a page must be written in the head section by using the tag<Script>

Program: Writing events

```
<%@ Page Language = "C#" %>
<!DOCTYPE html>
<html>
<head>
<script runat ="server">
protected void Page_Load (object sender , EventArgs e)
{
 if (! Page.IsPostBack)
 {
  lblTit~~e~~le.Text = "Welcome to ~~C#~~ ASP.NET";
 }
 else
 {
  lblTitle.Text = " Page Posted on :" + DateTime.Now.ToString();
 }
}
</script>
</head>
<body>
<form runat= "server" id= "frmDemo">
<asp: Label id = "lblTitle" runat = "server" />
<br>
< asp: Button id = "btnSubmit" runat= "server" text =
    "Submit" />

</form>
</body>
</html>
```

★ Developing the web Applications using Visual Studio.

★ n

★ database files - .mdf or .sdf

★ App start : Collection of ~~severtal~~ classes. when a program starts, it will automatically start.

★ Content : Contains non dynamic files. Ex. images, css-style sheets.

★ models : Class responsible for communication with dB

Script : Dynamic files like Js,

★ Account : Template files

★ Global.asax : global application class file. If we declare any value in this, we can access it from any locatn.

It contains global declarations.

★ web.config :-

It contains global configurations.

<globalization culture = "en-In" />

★ Adding pages :

Creating pages at run time.

When we use internal server → URL comes with unknown characters.

To run it on IIS : open VS as an Administrator

Soln Exp : Project name Right click → properties → web-cate → servers select local IIS and then look for URL → ADD /create Virtual directory

Click ok. Save and open IIS and goto sites → Default site → Your website is there

★ Creating ASP.NET applications using visual studio.

Visual studio is an IDE (Integrated devel env) for dev--eloping .NET applications. However from the latest version ASP.NET is an open source and can be used with any one of the following development tools.

1. Wiwet ASP.NET Template
2. ASP.NET Intellisense Generator
3. Microsoft visual studio
4. Microsoft visual web developer express
5. CodeGear Delphi → C# is derived from delphi, C++
6. Macromedia Homesite
7. Microsoft expression web
8. Microsoft Sharepoint designer
9. Monodevelop
10. Sharpdevelop
11. Eiffel for ASP.NET
12. Adobe Dreamweaver
13. Stadium.
14. Telerik.

★ Creating a new web application:-

1. Open VS 2013. Run → devenv
2. goto file menu → new project
3. Select visual C# choose → web category
   Select the template → ASP.NET Web Application
4. Specify name and location for application
5. Click OK.
6. IDE Prompts ~~to select~~ you to select template.
7. Select "Webforms"
8. Select checkbox for "webforms" as references

4. Click OK.

★ File system :-

Asp. NET Application file system :-

A typical ASP.NET 4.5 Application comprises of the following files and folders.

Sr.	file/folder	Description
1.	Properties	Contains assembly info.
2.	References	Contains collection of namespaces or assemblies (libraries) used in application.
3.	Account	Contains template files that are used to create and manage accounts. (login, Register, etc)
4.	App-Data	Contains local database files. (.mdf, .sdf)
5.	App-Start	Contains collection of classes that are intended to run on application start
6.	Content	Contains all non-dynamic files (images, css)
7.	Fonts	Contains all fonts installed for the application
8.	Models	Contains a collections of classes that are responsible for interacting with database.
9.	Script	Contains all dynamic files like JS, Jquery.
10.	Global.asax	contains global declarations that are accessable from any page in website.
11.	Web.Config	Contains the configuration settings that control the behaviour of application.

★ Adding webpage to application :-

1. Right click on solution explorer → project name.
2. goto the option add "new item".
3. Select the template "web form".
4. Name it as home.aspx
5. Click "add" button

★ Adding controls to page :

1) You can drag and drop the control from toolbox into design or source of page.

2) You can write the control syntax in page source.

```
<asp:button id = "btnSubmit" runat = "server" text = "Submit"
                                                        />
```

★ Absolute position for controls : Two ways

Method 1 : Using options.

1. goto tools menu
2. Select option → goto category "Html designer"
3. Select the sub category "css styling".
4. Select the checkbox "Change positioning.......".

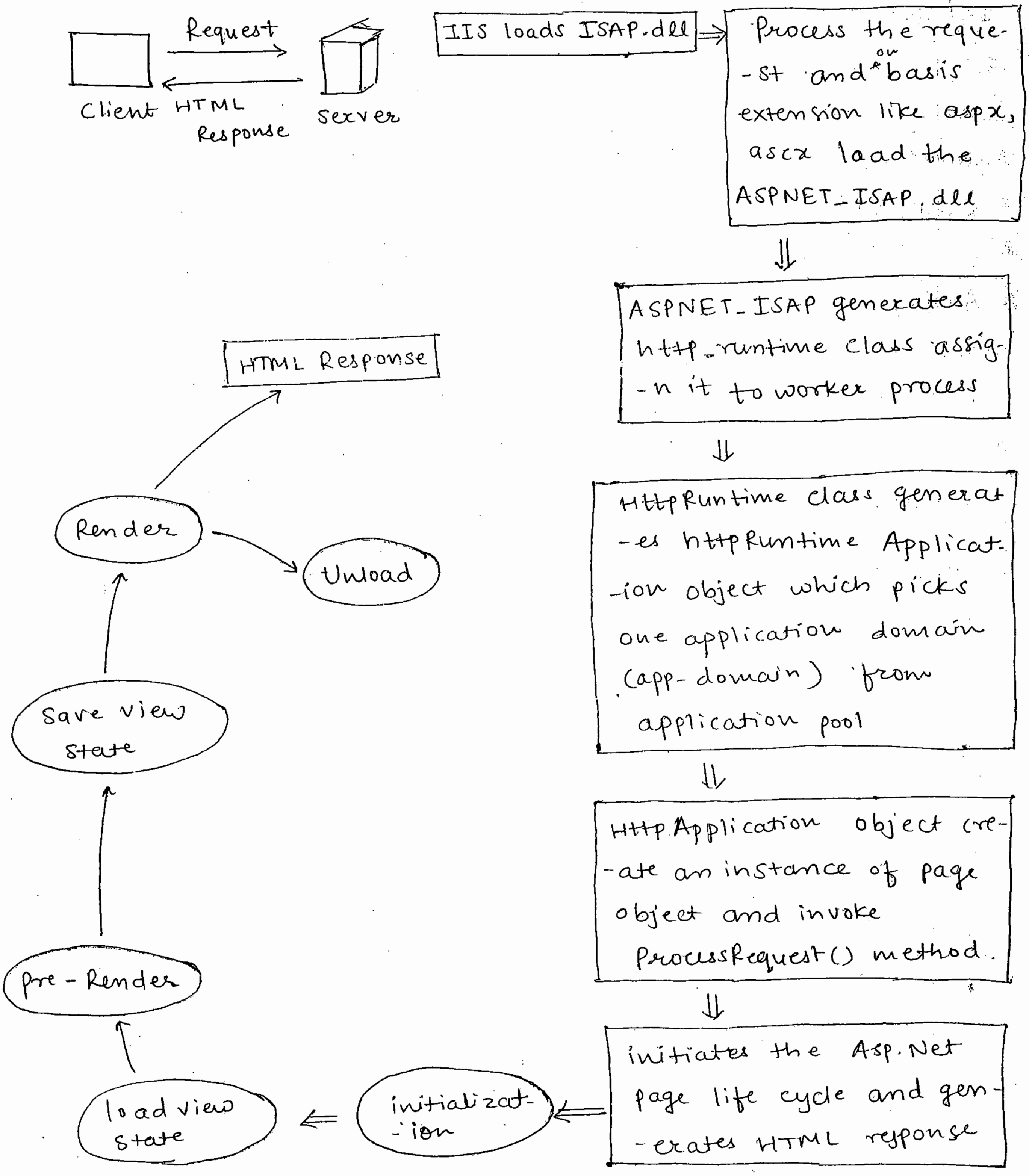
Method 2 : Using style.

```
<asp:button style = "position : absolute "/>
```

★ Better way of ~~ab~~ organizing info is using tables. We can use div but ~~for~~ we have to use div with tables.

★ ~~Open~~ Right click on Project name and go to second last option and go to bin you will get all dll files.

★ ASP.NET Application life cycle.



IIS loads ISAP.dll ⇒ Process the request and on basis extension like aspx, ascx load the ASPNET_ISAP.dll

⇓

ASPNET_ISAP generates http_runtime class assign it to worker process

⇓

HttpRuntime class generates httpRuntime Application object which picks one application domain (app-domain) from application pool

⇓

HttpApplication object create an instance of page object and invoke ProcessRequest() method.

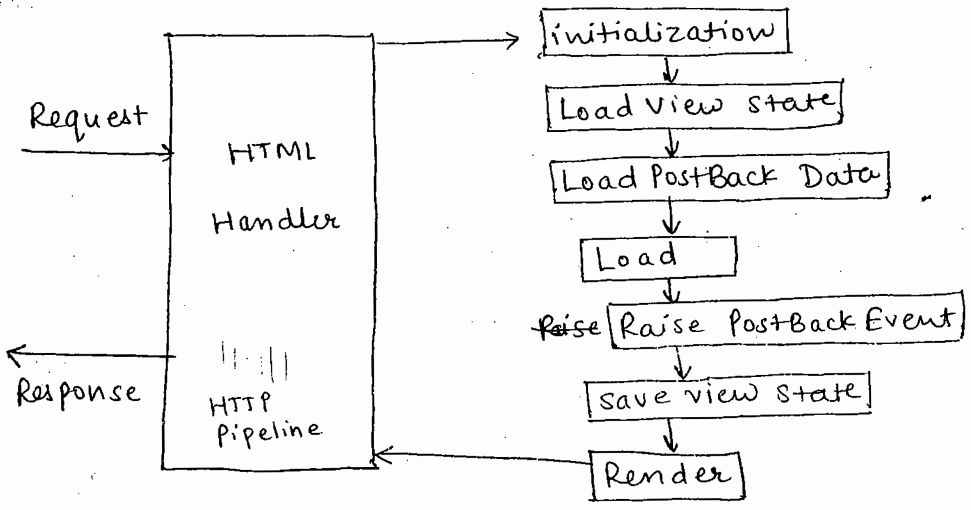
⇓

initiates the Asp.Net page life cycle and generates HTML response

Asp.NET page life cycle.

Page Stage	Description
1. Initialization	In this stage the controls are created and are assigned with unique ID's
2. Load view State	In this stage the values of controls are saved to provide on postBack.
3. Pre-Render	The control events are fired up and various actions are performed
4. Save view State	The results are computed and are saved in a view state.
5. Render	The Html response is generated for the client request and the response is send to client
6. Unload	It unloads the page and performs cleanup. i.e all traces of page are removed from server.

★ Asp.Net high level design



★ **Page events:**

There are four Stages in page events.

1) init

2) Load

3) Rendering

4) Unload.

3-06

★ Page Events :-

Every ~~t~~ typical ASP.NET Page comprises of several ~~eve~~ events classified into four categories

1. Initialization
2. Load
3. Rendering
4. Unload

State	event	viewstate
init	pre init int init complete	
Load	Preload	
	load Load complete Validation Control events	Available
Rendering	preRender Pre Rendering complete	
	Save state complete Render	
Unload	Unload	

★ How to write page events :

1. Add a new web form" by name "Demo.aspx".
2. In solution explorer select "demo.aspx.cs" file
3. Add the following events in page class.

~~4~~ protected void Page_Init (object sender, EventArgs e)
{

```
Response.write ("Controls Created :.." + "<br>");
 }
Protected void Page_Load (object sender, EventArgs e)
{
 Response.write ("page successfully ... loaded");
 }
```

★ Color : system.Drawing.

Properties = Statically → from properties.
Runtime → from coding

★ ASP.NET server side controls :

ASP.NET Provides a heap of controls that enables the UI to make more interactive and responsive.

All server controls in ASP.NET are defined under the library "system.web.UI.webcontrols".

1. Label :- A label control is used to display titles and captio--ns that are not editable manually during the run-time.

Properties : 1) ID
2) Runat
3) Text
4) Forecolor
5) Backcolor
6) Font.

Syntax :

```
<asp:label id = "lblTitle" runat = "server" Text = "Welcome
to ASP.NET    Forecolor = "white" Backcolor = "Red" />
```

Ex : Applying properties for label dynamically.

~~Protecte~~

```
using System.Drawing;

 Protected void Page-Load ('object sender, EventArgs e)
 {
  lblMsg.ForeColor = Color.White;
  lblMsg.BackColor = Color.Blue;
  lblMsg.Text = "Hello!";
 }
```

★ Buttons :- AutoEventWireUp = True then
 Command:- Writing one event which is available for several buttons.

Buttons in web forms are used to perform record actions, record navigation, and miscellenous actions.

ASP.NET Provides three types of button controls :

	Control	Description	Properties
1.	Button	It is an ordinary button with text.	ID Runat Text
2.	LinkButton	It is similar to a Button but shows the text as a hyperlink	ID Runat Text
3.	~~[illegible]~~ ImageButton	It is similar to other button controls but contains an image instead of text.	ID Runat ImageUrl

★ Button Events :

1.	Click	It specifies the actions to be performed when button is clicked.
2.	Command	It identifies a specific button click from a group of buttons. It requires a "<u>command Name</u>"

	to be defined for every button

Syntax :

<asp: Button id = " btnOrdinary" runat = "server"
text = " Submit" />

<asp: LinkButton id = " btnLink". runat = "Server" ~~text~~
text = " SignOut" />

<asp: Image Button id = " btnImage" runat = "Server"
ImageUrl = " ~ / Images / about. jpg" />
↓
tild

Ex. Using command Event for buttons :

1. Add a new web form by name "Demo.aspx".
2. Demo. aspx (design).

[lblTitle]

[insert]   [Update]   [Delete]

Control	properties.

1. label 1	id = lbltitle
	Text = " "

2. Button 1	id = "btnInsert"
	Text = "Insert"
	Command Name = Insert

3. Button 2	id = btn Update
	Text = Update
	Command Name = Update

4. Button 4	id = btn Delete
	Text = Delete
	Command Name = Delete

3| Select any button control and open properties window (F4)

4. goto events category and Double click on "command". event.

5. Change the event name ~~with~~ and code as shown below:

```
Protected void database_Command (object sender,
                                  CommandEventArgs e)
 {
    switch (e.commandName)
    {
     case "Insert" : lblTitle.Text = "Record Inserted";
                     break;
     case "Update" : lblTitle.Text = "Record Updated";
                     break;
     case "Delete" : lblTitle.Text = "Record Deleted";
                     break;
    }
 }
```

6. Goto ~~Aspx~~ "Demo.Aspx" Design

7. Select "Button 1" (Insert) and open properties.

8. Goto Events category and set command :=
   Database_Command

9. Repeat the same for all other buttons.

04/06

(18)

★ How do we remove the underline for link button?

Style = "text-decoratíon: none".

⇒ Using style.

```
<asp: LinkButton  id = "btnLink" runat = "server" text = "SignOut".
    Style = "text_decoration = none" />
```

★ ServerMap Path()

'~', ~~'@'~~ '.', '..'

↓ root Path

↓ current path

↳ less 1 than current

★ Virtual path and Physical path :

Virtual path is the path generated by webserver in order to access the resources from a website.

Ex. http://localhost/website/page.~~name~~ aspx

Every website is hosted on webserver and the files for web--site are stored in its physical path.

Ex. D:\WebsiteFolder.

~~But~~ You can track the physical path of any website by using the method "Server.MapPath()".

	Character	Description
1.	Tild	Returns path upto the website root directory
2.	. (dot)	Returns path upto the current directory
3.	..	Returns path one level back to the current directory

Syntax :

Server.MapPath("~"); OR Server.MapPath("..");

Ex. ImageButton1.ImageUrl = "~/Images/about.jpg";

★ TextBox control:-

All textBox properties we cannot you because some brow-
-ser may not support them
Some properties are not visible but they are available
to use.

⇒ TexBox is the basic input control in webforms which allows
the UI to input, read, and edit the values during runti-
-me.

- Properties : 1 ID
  2 Runat
  3. Text
  4. MaxLength
  5. ReadOnly
  6. TextMode →
    - SingleLine
    - MultiLine
    - Date
    - DateTimeLocal
    - Month
    - Year
    - Week
    - URL
    - Range
    - Number
    - Color
    - Phone
    - File

  7. PlaceHolder
  8. Pattern
  9. Focus() → method

Syntax:

1) Password ⇒ < asp: TextBox id = "txtPassword" ~~r~~ runat = "server"
TextMode = "Password" />

2) Date of Birth ⇒
<asp: TextBox id = " txtDob" runat = "server" TextMode = "Date" />

Event: TextChanged ⇒ finishing typing ~~wt~~ after what should happen.

Button-Clicks → pages are posted to server. ← Note.
For All controls except Buttons you set "AutoPostBack = True".
then only the event will fire up. otherwise not.
To button-click ~~thes~~ by default AutoPostBack is set to "true".
AutoPostBack = To convert the text. to Upper case - that is to fire up the text changed Event.

To Do this → go to the TextChanged Event.

* Event:

TextChanged: The Text Changed event for textbox indi-cates the actions to be performed when you finished typing the text and loose focus ~~form~~ from the control.

Note → The text changed event will not wireup untill or unless you set "autoPostBack to True" for TextBox Control.

Ex.

1. Add a new webform "Register.aspx".
2. Register.aspx (Design)
3.

Enter Name (Block Letters	▭	→ id = txtName
Password	▭	→ id = txtPassword

3. Double Click on "txtName" textBox and add the following code.

```
protected void txtName_TextChanged (object sender, EventArgs e)
{
 String name = txtName.Text;
 txtName.Text = name.ToUpper();
}
```

4. Goto Design and Open txtName Properties and set AutoPostBack = True for textbox.

☆ Types of redirection in Asp.NET

1. Response.redirect 2) Server.Transfer()

3. Cross Page PostBack 4. Server. Execute() 5. Hyperlink.

05/06

Redirections in Asp.Net :-

Redirection is the process of navigating from one page to another page or ~~any~~ to any named location within the same page. Asp.Net provides the following redirection mechanisms

1. Response.redirect()
2. Server.Transfer()
3. Server.Execute()
4. Cross page PostBack
5. Hyperlink.

Response.Redirect() Vs.	Server.Transfer()	Feature
1. Slow and uses round trip	1. fast and doesn't require round trip.	1. Accessibility
2. Not secured, it returns the URL of target page in address bar	2. Secured as it will not return the URL of target page.	2. Security

3. Can access the pages within Application or from any another appli--cation.	3. Can access pages only within application	3. Page Access--ibility
4. Allows you to Bookmark any random page while accessing in sequential order	4. Allows you to bookmark only the first page	4. BookMarking
		~~5. Syntax~~

Syntax: 1. for Server.Transfer :-

Server.Transfer ("welcome.aspx");

2. for response.Redirect :-

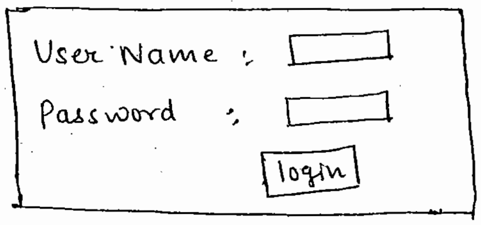
Response.Redirect ("http://localhost /website / page.html ");

Ex :

1) Add the following pages to your website.

1. Login.aspx
2. welcome.aspx
3. Error.aspx

2) Login.aspx (Design) :-



TextBox1 Id = txtName

TextBox2 id = txtPassword

TextMode = Password.

3) Login.aspx.cs (Code)

// Login Button Click code

```
if (txtName.Text == "manager" && txtPassword.Text ==
                                        "nareshit")
  {
  Response.Redirect
("http://localhost/Naresh IT/home.html");
  }
  else
  {
  Server.Tranfer ("Error.aspx");
  }
```

4) Welcome.aspx (Design)

```
<h1> Login success... </h1>
```

5) Error.aspx (Design)

```
<h1> Invalid User Name / password </h1>
```

☆ Server.Transfer Vs Server.Execute.

⇒ The method Server.Execute is similar to Server.Transfer in syntax but functionally server.Execute will execute the target page and render's its output in the same page.

<u>Syntax</u>: Server.Execute ("Error.aspx");

☆ Cross Page PostBack ~~Important~~

⇒ When I click the button then the same page will come again is the corouss page PostBack.

⇒ PostBack is a mechanism where the current page containts are posted to server. You can design the postBack in such a way so that the current page contents are posted to any another page in the website. This is known as cross page postBack.

1) Add a new webform by name "Demo.aspx".

2) Add a button control to webform and set the text as postBack.

3) Goto Button properties and set the following attribute.

PostBackUrl = ~/Welcome.aspx

Note: what happens when you click a button on Demo.aspx page : It will post the contents to welcome.aspx.

★ Hyperlink : It is a control, ~~property~~.

A hyperlink is clickable text, picture or graphic that links to any another document or a named location in the same document.

ASP.NET provides a hyperlink control that manages navigation in website.

★ Properties for hyperlink:

1) Id
2) Runat
3) Text
4) ImageUrl
5) Navigate Url
6) Target

★ Syntax : <asp:Hyperlink id = "link1" runat = "server"
Text = "Goto C# Basics" Navigate Url = "#CSharp"
Target = "_blank" />

★ Intra Document links : It refers to a hyperlink that navigates to any named location within the same page

How do we do that?

1) Name a location in your document.
< h1 id = "asp"> ASP.NET Basics </h1>

2) Refer the named location using an hyperlink.

```
<asp: Hyperlink id = "link1" runat= "server" Text= "ASP.NET"
 NavigateUrl = "#asp"/>
```

★ Inter document links : A hyperlink allows to navigate, to read any document or to download a file. If it is redirecting to any another document or URL then it is reffered as inter document link.

Ex:

1. Add a new web form by name "Home.aspx".
2. Add following hyperlink controls to page.

```
1. <asp: Hyperlink ID= "HyperLink1" runat= "server"
   Navigate-Url = "~/Tutorial . aspx " Target ="_blank">
    GotoTutorial .
   </asp : HyperLink >
         <br/>

2. //Redirecting a website
  <asp: HyperLink ID = "HyperLink2" runat= "server"
   NavigateUrl = "http:// www. NareshIt.in" >
  NareshIT Website </asp : HyperLink >
    <br/>

3. <asp : HyperLink ID = " HyperLink3 " runat = "server "
   NavigateUrl = "~/Content /asp.pdf " >Read ASP Tutorial
    </asp : HyperLink >
      <br/>

4. <asp : HyperLink ID = " HyperLink4 " runat= "server "
   NavigateUrl= "~ / Content/ Adobe. exe ">
    Download Adobe reader .
    </asp : HyperLink>
```

```
< br/>
5. <asp:HyperLink ID = "HyperLink5" runat = "server"
   ImageHeight = "100px"  ImageUrl = "~/Images/3.jpg"
  ImageWidth = "100px"  NavigateUrl = "~/Images/3.jpg">
  </asp:HyperLink>
```

★ File Upload Control:

The file upload control unables the UI to select a file from your computer and to upload into ~~servered~~ system.

★ Properties and methods:

1) Hasfile : Returns "true" if file selected and returns

2) Saveas : Saves the selected file in specified location.

3) filename : Returns the selected file name

4) PostedFile.ContentLength : Returns the selected file size, ~~to~~

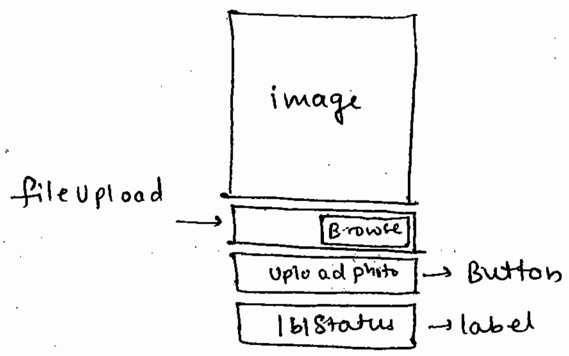
5) PostedFile.ContentType : Returns the selected file mime type

Ex:

1) Create a new folder in your website by name "Images".

2) Add ~~an~~ new webform by name "Upload.aspx".

3. upload.aspx (Design)



Source code:

```
// ~~inter~~ import namespace
   Using system.Drawing;
```

```
// Upload Button Click code

Protected void Button1_Click (object sender, Eventargs e)
 {
 if (FileUpload1.HasFile)
 {
  string fileType = System.IO.Path.GetExtension
    (FileUpload1.FileName);
   if (fileType != ".jpg" && fileType != "png" &&
     fileType != ".gif")
     {
  lblStatus.ForeColor = Color.Red;
  lblStatus.Text = "You can upload only Images";
     }
  else
  {
 int fileSize = FileUpload1.PostedFile.ContentLength;
   if (fileSize > 1048576)
  {
   lblStatus.foreColor = Color.Red;
   lblStatus.Text = "You can upload only 1 MB";
  }
 else
 {
 FileUpload1.SaveAs (Server.MapPath ("~/Images/") +
 FileUpload1.FileName);
 lblStatus.ForeColor = Color.Green;
 lblStatus.Text = "File Uploaded Successfully";
Image1.ImageUrl = "~/Images/" + FileUpload1.
                                FileName;

  }
 }
}
 else
{
lblStatus.ForeColor = Color.Red;
lblStatus.Text = "Please Select a File";
  }
 }
```

08/06

★ Add Rotator :

The add rotator control is used to display advertisements in a webform and change the advertisements on every postback.

The advertisements for add rotator comes from an XML file which comprises of the following elements.

Element	Description
1. <Advertisements>	It specifies a collection of advertisements.
2. ~~<ad~~ <ad>	It represents an individual add in a collection of advertisements.
3. <ImageUrl>	It specifies the name and path of image to be displayed as advertisement.
5. <NavigateUrl>	It indicates the virtual path for redirection when an advertisement is clicked
6. <AlternateText>	Text to be displayed when image fails to load.
7. <Impressions>	It specifies the priority of an advertisement.
8. <Keyword>	It describes the keyword that enables filtering of advertisements.

★ Syntax: `<asp:AdRotator id="ads" runat="server" AdvertisementFile="ads.xml" KeywordFilter="Pepsi" height="100" width="400"/>`

Ex.

1. Right click on project name in solution explorer and add new folder by name ads.
2. Right click on Ads folder and select ~~new~~ add → new items.
3. goto visual C# category and select XML ~~file~~ file. ↑
4. Name the file as ads.XML.
5. Write the following code in ads.XML file

5.
```
<Advertisements>
<ad>
< ImageUrl > pepsi.jpg </ImageUrl>
<AlternateText> Pepsi foods.Ltd.</AlternateText >
<NavigateUrl> http://www.pepsifoods.co.in </NavigateUrl>
<Impressions> 40</Impressions >
<Keyword> Pepsi </Keyword>
</Ad>
<Ad>
< ImageUrl > ~~Reliance~~.rDigitial.jpg < /ImageUrl>
<AlternateText> Reliance Digital </ AlternateText>
< NavigateUrl > http://www.Reliance-Digital.com </NavigateUrl>
<Impressions > 30 < Impressions>
<Keyword > Reliance </ keyword>
</Ad>
</Advertisements>
```

Note :- Keyword → Pepsi or Reliance if we give the keyword Filter ~~the~~ property then it will show the particular add only.

6. Add a new webform "Home.aspx"

7. Drag and drop ~~a~~ adRotator.

8. goto the properties of adRotator and set the following

* AdvertisementFile	: Ads.xml
KeywordFilter	: Reliance
Width	: 400
height	: 100.

09/06

*

Note : Smarttag - In ~~calender~~ control it is ~~the~~ will Tooltip..

* Calender Control:-

The calender control provided a month view of calender that enables the user to select a date, week or month.

Properties:

1) ID
2) Runat
3) DayNameFormat = short or full
4) First Day -ofweek
5) Showtitle
6) titleformat :- Month, MonthYear
7) SelectorMode : Day, week, DayWeekMonth
8) Select Week Text :
9) Select Month Text

* Events:

1) Selection Changed
2) DayRender

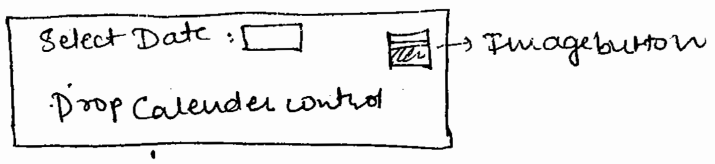
* Syntax:

<asp:Calender Id = "Cal 1" runat = "server" SelectionMode = "Day". DayNameFormat = "short" / >

Ex. Enable the user to select a date from calender.

1. Add a new webform → "Trip.aspx".

2. Trip.aspx (Design)



3. Trip.aspx : Code.

* page_load() event code

```
if (! page.IspostBack)
 {
  Calender 1.visible = false;
  }
```

4 * Image Button - Click()

```
    {
     if (calender1. visible == true)
     {
      Calender1. visible = false;
     }
    else
     if (calender1 . visible == false)
     {
      Calender1. visible = true;
      }
     }
```

⑤ Calender 1 - Click().

```
  {
   TextBox1. Text = Calender 1 .Selected Date. To string ("D");
  }
```

<u>Ex2</u>: Enable calender to select a week or a month - or day

1. Goto Calender property and set selectionMode = Dayweek
   Month

2. write following code in calender selection changed event.

3. Calender1_Click()

```
{ Textbox1.Text = Calender1. SelectedDate. Tostring ("D");
foreach ( Datetime d in calender1. SelectedDate)
{ label1. Text + = d. Tostring ("d") + "<br";
}
}
```

Ex 3: Design calender so that user can't select other month ~~p~~ day and weekend.

1. goto calender properties and set First Day of week ="Monday".
2. goto calender event and double click on DayRen-der event and write the following code

```
Calender1_DayRender()
{
 if(e.Day.IsotherMonth || e.Day .IsWeekend)
 {
 e.Day.IsSelectable = false;
 }
 if (e.Day.IsWeekend)
 { e.cell.Backcolor = System.Drawing.color.Red;
   e.Cell.Forecolor = System.Drawing.color.white;
   e.cell.Text = "N/A".
   e.cell.ToolTip = "Booked";
 }
}
```

Ex.4: Enable calender to block specific Date. write the following code on Day-Render event

```
if ( e.Day.Date.Day = =17)
 {e.cell.Text = "Holiday";
 }
```

Note :

Note: Mutual ~~Exclustion~~ Exclusion (Multi Threading)

→ RadioButton: This control enables the user to select only one option from the group of choices.
~~If~~ It uses the mechanism of multithreading called "Mutex". [Mutual Exclusion]

Properties: Id, Text, Runat, Checked, groupName ***

Note: In order to group the radioButtons into one category set groupName and it should be same for all radioButtons.

Syntax: <asp: RadioButton Id = "opt1" runat= "server"
Text = "~~Mal~~ ~~People~~ Male", GroupName = "Gender"/>
<asp: RadioButton Id = "opt2" runat = "server"
Text = "Female" groupName = "Gender"/>

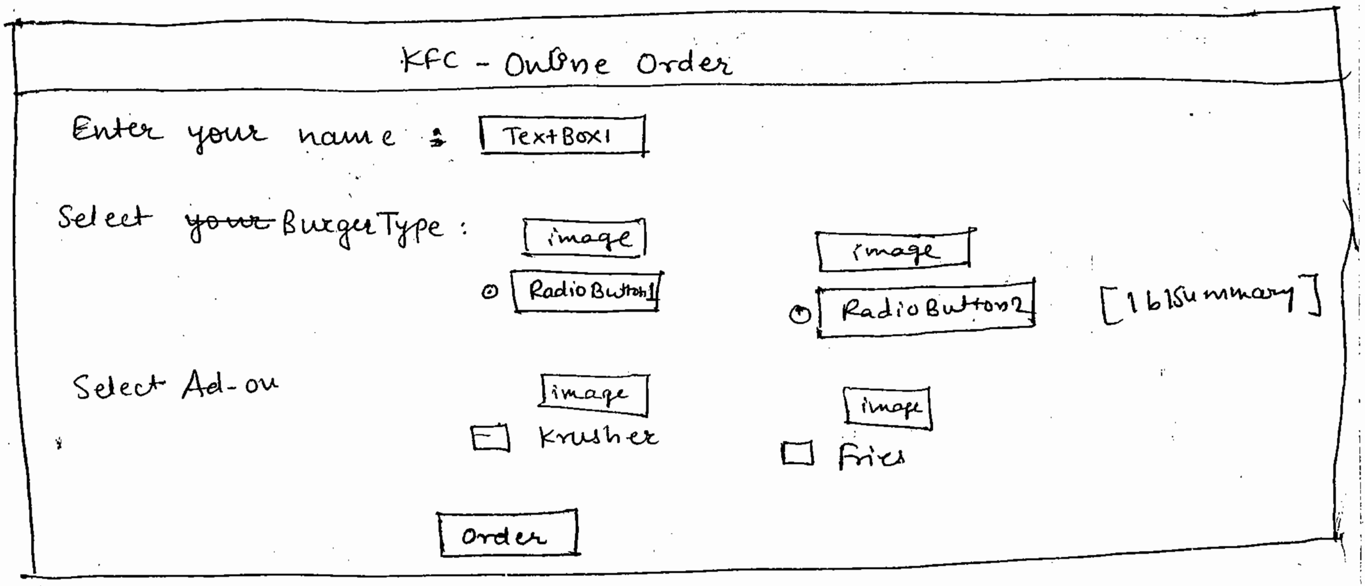
Event: CheckedChanged
Note: To enable CheckedChanged event you have to set AutoPostBack To "True".

☆ CheckBox: It is similar to RadioButton in properties and event but enables the user to select multiple option at a time from a group of choices.

Properties: Id, runat, Text, checked
Event: CheckedChanged.

Ex Using RadioButtons and CheckBoxes

1. ~~Add~~ a new webform by name " KFC Order . aspx " ~~(Design)~~
2. KFCOrder. aspx (Design)


KFC - Online Order
Enter your name : TextBox1
Select ~~your~~ BurgerType : image RadioButton1 image RadioButton2
[lblSummary]
Select Ad-on image Krusher image Fries
Order

A Control	Properties	
1. TextBox1	ID = TextBox1 , ~~RadioButton1~~ .	
2. RadioButton1	ID = optChicken · Text = Chicken Burger	GroupName : Burger
3. RadiOButton 2	ID = optVeg Text = Veg Burger	
4. CheckBox1	id = optfries Text = fries .	
5. CheckBox 2	id = optKrusher Text = Krusher	
6. Button 1	id = BtnOrder Text = Order	
7. label1	id = lblsummary Text = Empty " "	

3) KFCOrder . aspx . cs (Code)

// order button click event code

```
int bcost , acost , total ;
Protected void Button1_click (object sender , EventArgs e)
{
 if (optChicken.Checked)
 {
     bcost = 150;
 }
 if (optVeg.Checked)
 {
     bcost = 100;
 }
 if (optFries . Checked)
 {
   acost = 40;
 } bcost = bcost + acost ;
 }
 if (optkrusher . checked)
 {
   acost = 60;
   bcost = bcost + acost;
  }
   total = bcost ;
  lblsummary.Text = "Hello!" + TextBox1.Text + "<br>"
  + "You have to pay :" + total.ToString("c");
 }
```

4. Goto Html source of KFC Order.aspx and set culture
`<%@ Page culture = "en-In" ... %>`

★ <u>List Controls</u>:- The list controls are collection controls that enables the UI to maintain a collection of items so that user can select or manipulate the list.
A list control resembles an "<u>array list</u>" which can store any type of value and allows accessing by their index. Every item in a list control is of type "<u>list item</u>" → class Name

The following are Asp.Net list controls :-

1. DropDownList
2. ListBox
3. Bulleted List
4. RadioButton List
5. CheckBox List

★ DropDownList : A dropdownlist provides collection of items that enable the user to select any one item ~~f~~ from list. All items in the list are of type "list item" and contains the following properties :-

a) Text
b) Value
3) Enabled
4) Selected.

★ Members of dropdown list :

Member	Description
1. Item.Add()	Adds a new item to list
2. Item.Remove()	Removes the specified item from list.
3. Item.RemoveAt()	Removes an item by its index number
4. Item.Clear()	Removes all items from list.
5. Items.Contains()	Returns boolean true if specified item exists in list.
6. Items.count()	Returns the total count of items.
7. SelectedItem.Value()	Returns the selected item value
8. Selected Item.Text()	Returns the selected item text.
9. Selected Item()	Returns both value and text.
10. SelectedIndex()	Returns the index number of selected item

Events :- SelectedIndexChanged.

Syntax :

1) Adding items to dropdown list in html source:

```
<asp: DropdownList ID = " lstPayment" runat= "server">
<asp: ListItem Text= "Cash" Value = "1" >
</asp: ListItem >
<asp: ListItem Text = "Credit card" . value = "2" selected=
"True" enabled = "True" >
</asp: ListItem >
<~~/~~asp: DropDownList >
```

Syntax 2: Adding items dynamically during runtime using code.

1. Goto page Code (C#) .

2. Create a new List in page class.

```
3. List<ListItem > cities = new List<ListItem > ( )
{
  new ListItem { Text = "Select a city" , Value = "-1",
  selected = true } ,
  new ListItem { Text = " Chennai", Value = "650004",
  Enabled = true ; Selected = false } ,
  new ListItem { Text= "Hyd" , value = "500045" } ,
  new ListItem { Text= "Mumbai", value = "400020" } ;
};
```

3. Write the following code in page-Load Event -

```
Protected void page_Load ( object sender, EventArgs e)
{
   if( ! page . IsPostBack)
   {
        foreach (var item in cities)
      {
        ~~IsCity~~ lstCity.Items.Add(item);
        }
      }
   }
```

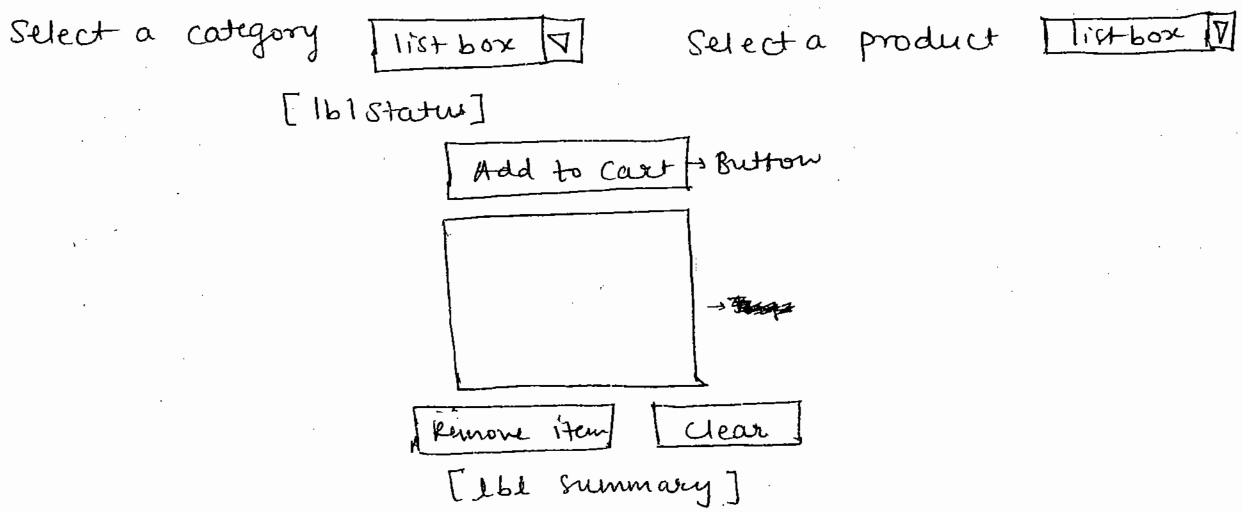
__Syntax 3__ : Adding items in Design mode

1. Add a dropdown list on page
2. Open properties window
3. Goto "Items" collection
4. Click "Add" button and add the listItems with "text; value, etc...".

12/06

★ __Ex__: Online shopping cart with list box and dropdown list.

1. Add a new webform by name shopping.aspx.
2. Shopping.aspx (Design)



★ __Code for this__ :-

1)

Control	Properties

1. Dropdown List1	id = lstCategories
	AutopostBack = True

2. DropDownList 2	id = lstProducts
3. ListBox 1	id = lstCart Selection Mode = multiple.

4. Button 1	id = "btnAdd"
	Text = 'Add to cart

5. Button 2	id = "btn Remove"
	Text = "Remove item"

6. Button 3	id = " btn Clear"
	Text = "Clear

7. Label1	id = lblstatus
	Text = " "

8. Label 2	id = 'lblSummary
	Text = " "

☆ Shopping.aspx.cs :
Write the following code in page Class. → partial class <u>Shopping</u>

```
public partial class Shopping (

{
 List< string> Categories = new List<string>()
  {
   "Select a category", "Electronics", "Shoes",
  };
List< ListItem > electronics = new List< ListItem>() .
 {
```

```
new ListItem { Text="Mobile", value="6000"},
new ListItem { Text="LG LED TV", value="67000"}
};

List<ListItem> shoes =new List<ListItem>()
{
new ListItem { Text = "Nike", value="4500" },
new ListItem { Text = " Lee Cooper", value = "8000"}
};
```

★ // Write the following code on page-Load event.

```
if (! Page.IsPostBack)
{
foreach( String item in categories)
 {
  lstCategories . Items. Add ( item);
 }
}
```

★ // List categories selected index changed code.

```
switch ( lst Categories . SelectedIndex)
   {
Case 0 : lbl Status . Text= "Please select a category";
 lstProducts . Items. Clear();
 lstProducts. Items . Add ( " select a product");
        break;

Case 1 : lst Products . Items. Clear ();
foreach ( var item in electronics ) → if simple list then
 {                                      use string at place
lstProducts .Items. Add (item);          of var
}
 break;
Case 2 : lstProducts .Items. Clear();
 foreach ( var item in shoes)
 {
 lstProducts. Items . Add (item);
  }
  break;
}
// create a new method by name GetBill.
 int bill=0;
 Private void GetBill ()
 {
for (int i = 0; i< lstCart .Items. Count ; i++)
{
bill = bill + Convert To Int32 ( lstCart . Items[i].Value);
 }
lblsummary. Text = "Total amount:" + bill.ToString ("C");
lblStatus . Text = "Total no of items = " + lstCart. Items.
 }                                          Count;

// Add to Cart Button Click code:
 if ( lstCart. Items. Contains (lstProducts . SelectedItem))
  {
 lblStatus.Text = "Item Exists";
  }
  else
  {
  lstCart . Items. Add (lstProducts . SelectedItem);
   GetBill();
  }
 // Remove item button click code
```

```
lstCart.Items.Remove (lstCart.SelectedItem);
    GetBill();
// clear button clickcode :
  lstCart.Items.Clear();
   GetBill();
```

13/06
★ RadioButton List :
It is a collection of radiobuttons that enables the user to select any of option from the list from a group of choices. Its properties and methods are similar to other list controls. like drop down list and List box.

Syntax: `<asp: RadioButton List ID = " RadioButton List1" runat =" ser-ver" AutoPostBack = "True" >`

```
<asp: ListItem > Cash on delivery
</asp : ListItem >
<asp : ListItem > Credit card </asp: ListItem >
</ asp : RadioButton List1 >
```

★ Panel : Panel is a container control that contains a group of Asp or html controls so that you can hide or unhide the panels during runtime and enable the user to view multiple pages information on single page.

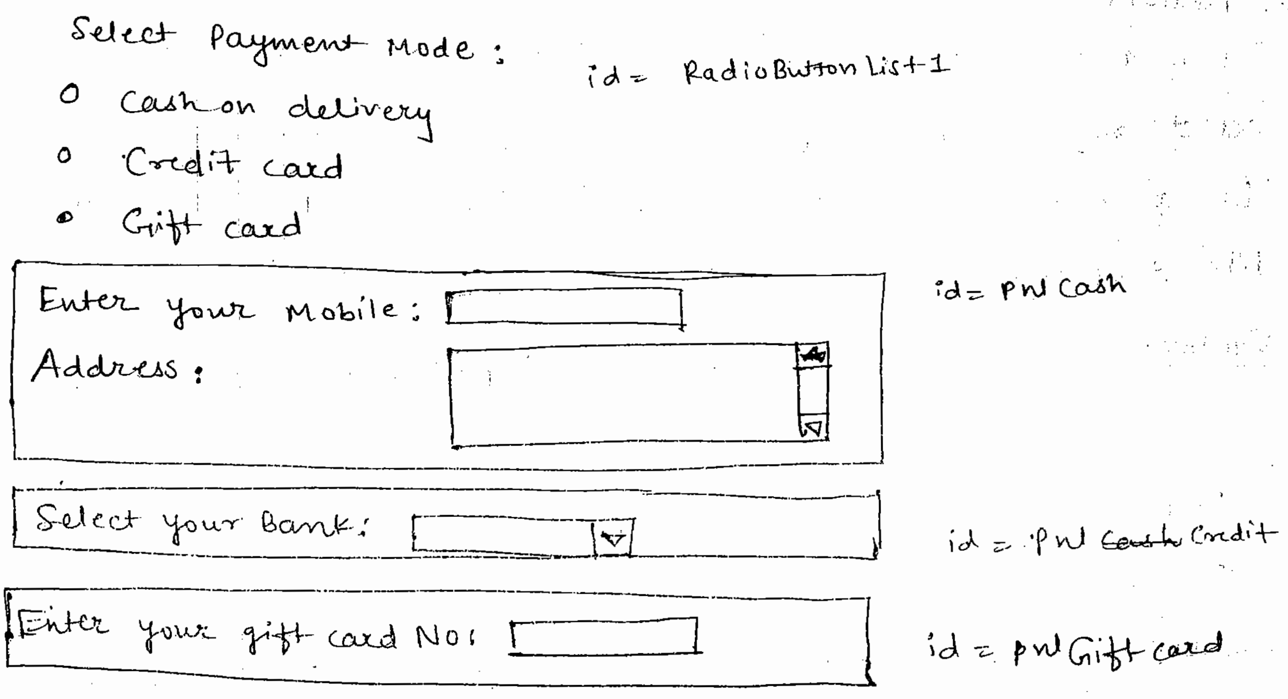
★ Properties:
1. id
2. runat
3. Visible = true or false
4. Backcolor

Syntax:

```
<asp: panel id = "pnl1" runat= "server" >
      you can put controls inside panel.
```

Ex:

1. Add a new webform by name "payment.aspx"
2. "Payment.aspx" (Design)



Select Payment Mode : id= RadioButtonList1

o Cash on delivery

o Credit card

● Gift card

Enter your Mobile : id= pnlCash

Address :

Select your Bank: id = pnl ~~Cash~~ Credit

Enter your gift card No: id = pnlGiftcard

3. Payment.aspx.cs (code)

```
// Page-Load event code :
    if (!Page.IsPostBack)
    {
      pnlCash.Visible =false;
      pnl CreditCard. Visible= false;
      pnl GiftCard.Visible =false;
    }

// RadioButton1_ SelectedIndexChanged Event Code:
      (set AutoPostBack= True)
```

```
Switch ( RadioButtonList1. SelectedIndex)
 {
 case 0: pnl Cash . Visible = true;
         pnl Credit Card. visible = false;
         pnl GiftCard. Visible = false;
         break;
Case 1: pnl Cash. Visible = false;
        pnl CreditCard. Visible = true;
        pnl GiftCard. Visible = false;
        break;
Case 2: pnl Cash . Visible = false;
        pnl CreditCard . Visible = false;
        pnl GiftCard. Visible = true;
        break;
```

* Checkbox List :- A checkbox list is similar to a listbox but it enables the user to select ~~any~~ multiple items. The condition for selected items is satisfied based on the checked property of checkbox.

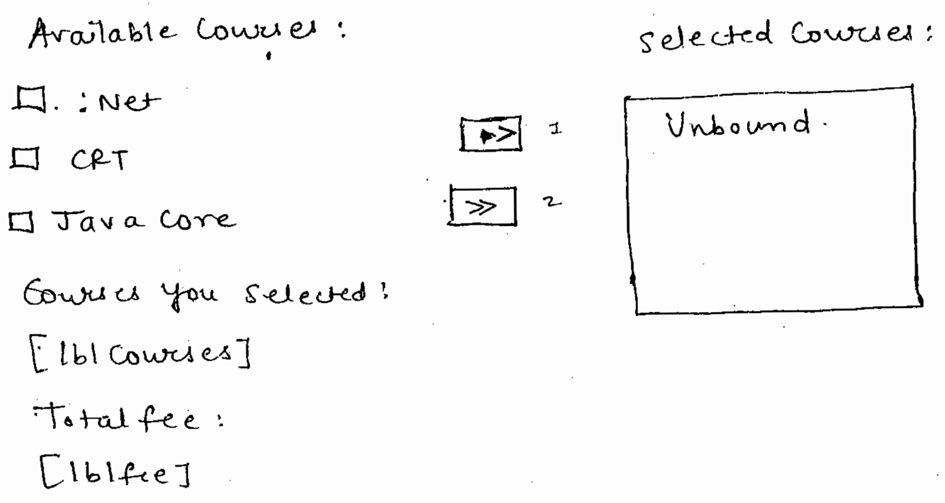
All the methods and properties of checkbox are similar to the dropdown control.

Syntax:

```
<asp: CheckBoxList id = "lst" runat = "server" >
<asp; ListItem > Item 1 </asp: ListItem >
<asp: ListItem > Item 2 </asp: ListItem >
</asp: CheckBoxList>
```

★1. Add a new webform by name "courses.aspx".

2. Courses.aspx (Design)



3. Courses.aspx.cs (Code)

// Create a new method by name "getdetails".

```
    int fee = 0;
    String courses;
    private void GetDetails()
    {
     for (int i=0; i< lstSelectedCourses.Items.Count; i++)
     {
     fee = fee + Convert.ToInt32 (lstSelectedCourses.
                                    Items[i].Value);

          courses = Couses + lstSelectedCourses.Items[i].
                          Text + "<br>";
     }
   lblFee.Text = fee.ToString ("c");
   lblCourses.Text = courses;
  }
```

// Adding selected items. (>) button click code:

```
   for (int i=0; i< lstAvailCourses.Items.Count; i++)
   {
    if (lstAvailCourses.Items.~~Contains~~ [i] . Selected  &&
lstSelectedCourses.Items.Contains(lstAvailCourses.Items[i])
                                                  == false)
{
 lstSelectedCourses.Items.Add(lstAvailCourses.Items[i]);
   }
  }
  GetDetails();
```

// Adding all items (>>) Button click code:

```
for (int i=0; i< lstAvailCourses.Items.Count; i++)
{
 lstSelectedCourses.Items.Add(lstAvailCourses.Items[i]);
 }
GetDetails();
```

15/06. www.gifanimations.com ~~[illegible]~~

From properties we can change the display form from text to hyperlinks.

★ Bulleted list : It is also a list control similar to other list controls in ASP.NET but it allows the ~~[illegible]~~ UI to display items in the form of bulleted or numbered list. Its properties and methods are similar to other list controls. You can dynamically add and manipulate the list.

1) Properties :-

1) ID

2) Runat

3) BulletStyle - numbered
 - Alphabets
 - CustomImage, etc

4) Displaymode - text
 - hyperlink
 - link Button

5) BulletImageUrl → path and name of image.

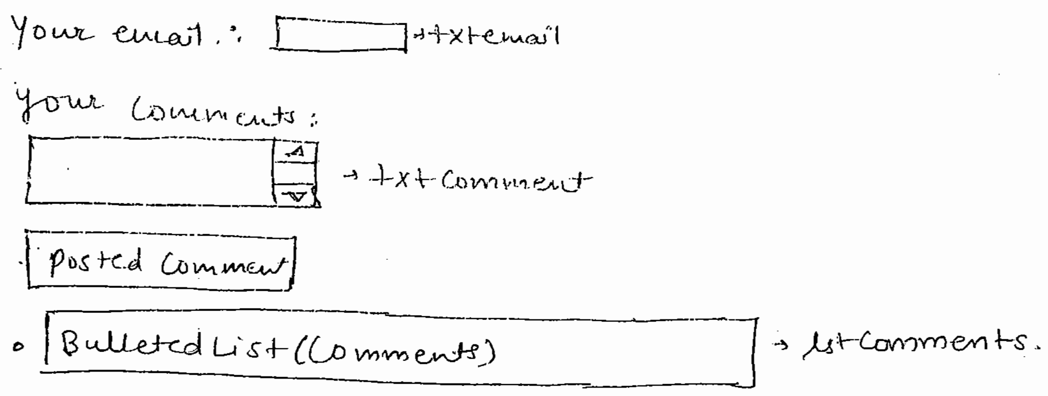
Syntax: ~~AS~~.

```
<asp : BulletedList · id= "lst1" runat = "server"
  ~~BulletedList~~ Bulletstyle = "Numbered" Displaymode = "Text">
<asp : ListItem  Text = "Item 1">
</asp: ListItem>
</asp : Bulleted List>
```

Ex: BulletedList with displayMode as Text.

1) Add a new webform "Comments.aspx".

2) Comments.aspx (Design)



★ Code : Comments.aspx.cs (Code):-

```
// postComment  Button Click Code

  string email = txtEmail.Text;
  string comment = txtComment.Text;
  lstComments.Items.Add(Comment + " - posted By " +
  email + "On :" + DateTime.Now.ToString("D"));
      txtEmail.Text = "";
      txtComment.Text = " ";
```

Ex: Bulleted List with display mode Hyperlink.

1. Add a new webform "Site.aspx".
2. Add bulleted List Control to page
3. Goto Bulleted List properties and set Display Mode = "hyperlink".
4. ~~Goto In~~ ~~Butt~~ Bulleted List properties, Select "items-collection" and add the following items.

5. item-1 :
   Text = Gmail
   Value = http:// www. Gmail.com .

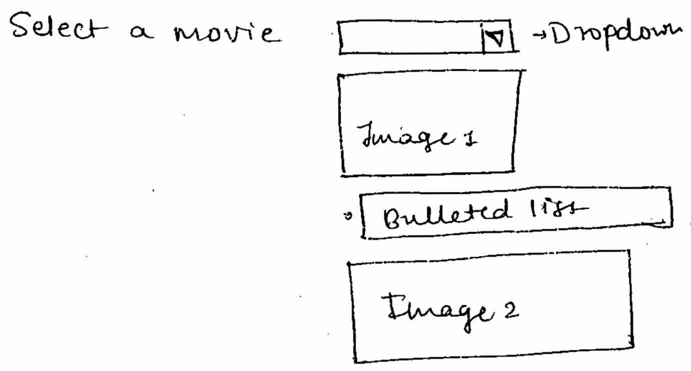
   item-2
   Text = Youtube Videos.
   Value = http:// www. Youtube.com

~~* Bul~~

* Bulleted List Items with display mode as "Link Button"

① Add a new webform "Movie.aspx"

② ~~IA~~ Movie.aspx (Design)



Control	Properties
1. DropdownList 1	id = lstMovies
2. Bulleted List 1	id = lstTimings
	Displaymode = LinkButton
3. Image1	id = Image1

4) image 2 ID= Image2

5) ~~lb~~ label 1 id = lblstatus

Text = " "

★ Code : Movie.aspx.cs (code)

```
// Create the following list in page class.
 // partial class ______

 List<string> movies = new List<string>()
      { "Select a Movie",
        "San Andreas",
        "Jurrasic World".
      };

 List < ListItem > Jurrasic Timings = new List < List Item > ()
 {
 new ListItem { Text = "10:20 AM", Value = "10:20 AM" },
new List Item { Text = "02:45 PM", Value = "02 : 45 PM" }.
 };

List< ListItem > San Timings = new List < ListItem>()
{
new ListItem { Text = "11:00 AM", value = "11 :00' AM" },
new ListItem { Text = "02:30 PM" value = "02:30 PM" },
new ListItem { Text = "10:00 PM" value = "10:00 PM" },
 };

// Page_Load event Code :
     if (!page.IsPostBack)
     {
      foreach (string item in movies)
     {
      lstMovies.Items.Add(item);
```

```
            }
          }
// Dropdow List selected Index changed event code :
   switch ( lstMovies . SelectedIndex)
     {
  case 0 : Image1 . ImageUrl = "~/Images/ inox . jpg";
           break;
  case 1 : lstTimings . Items . Clear();
           Image1 . ~~@~~ImageUrl = "~/Images/ sam.jpg";
       foreach ( var  item  in  samTimings)
        {
     lstTimings . Items . Add (item);
      }
      break;

  case 2 :  lstTimings . Items . Clear();

       Image1 . ImageUrl = "~/Images / jurassic.jpg";
  foreach ( var  item  in  jurassicTimings)
  {
  lstTimings . Items . Add (item);
  }
   break;
  }

// BulletList 1 _ Click event code :
 protected void lstTimings_click ( object sender , BulletedList Event
                                                   Args e)

  { switch ( e. Index)
    {
  case 0 : Image2 . ImageUrl = "~/Images /1.png";
        lblStatus . Forecolor = Color. Green;
     lblstatus . Text = "200 seats Available";
```

```
            break;
    case 1 : Image2.ImageUrl = "~/Images/2.png";
            lblstatus.Forecolor = Color.Yellow;
           lblStatus.Text = "100 seats available - fast filling";
            break;
    case 2 : Image2.ImageUrl = "~/Images/3.png";
           lblstatus.ForeColor = Color.Red;
         lblstatus.Text = "Sold out";
          break;
    }
  }
```

16/06 ★ Managing multiple views in web Applications:-

You can access multiple pages information through a single page. So that it will reduce the number of request and improves the performance of application.

The following methods can be used to handle multi-ple views :-

1) Frames - HTML
2) iFrames - HTML5
3) MultiView
4) wizard
5) panel

★ Using frames of HTML.

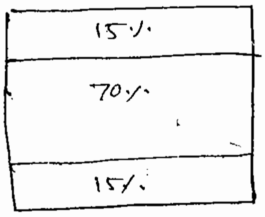
1) Add the following pages to your application.

- index.html
- footer.html
- menu.aspx
- home.aspx
- about.aspx
- contact.aspx

★ Index. html source:-

Remove head and body in html page and write:

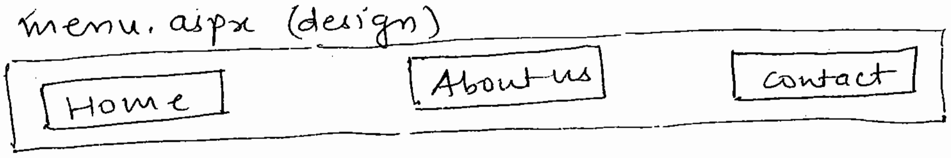
```
<!DOCTYPE html>
<html>
< frameset rows = "15%, 70%, 15%">
 <frame src = " menu.aspx"
  name = "framehead" ></frame>
  <frame src = " home.aspx"
   name = "framebody" > </frame>
   <frame src = " footer.html"
    name = " frame~~body~~Footer " ></frame>
  </frameset>
  </html>
```



3. Footer.html source:

```
<centre>
<div> & copy Copyright 2015 </div>
</centre>
```

4. menu.aspx (design)



5. Control and properties:

Control	property
1. Hyperlink 1	ImageUrl = home.png
	NavigateUrl = home.aspx
	Target = frame Body
2 Hyperlink 2	ImageUrl = about.png
	NavigateUrl = about.aspx
	Target = frameBody

3) Hyperlink 3 ImageUrl = Contact.png
NavigateUrl = Contact.aspx
Target = frameBody.

5. Put any information in the following pages:

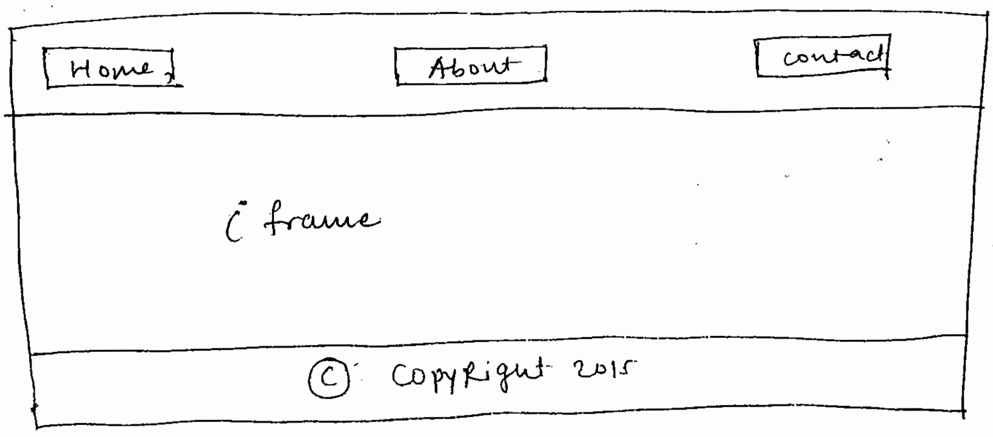
- home.aspx
- about.aspx
- Contact.aspx

Note: Start with "index.html".
- Right click on index.html ~~in~~ in sol^n explorer.
- Select "set as start page".

★ Using iframes of HTML-5.
The major drawback with frames in html is they are not supported for several browsers, which are compatible with mobile devices. Hence, html 5 introduces iframes that creates an embaded frame inside the body and allows to access any page or url.

★ Add following pages to application:
- New Index.aspx
- Home Aspx
- About.aspx
- Contact. aspx

★ New Index.aspx (Design) take row 3 colum 1



3. To add iframe in an new Index.aspx page.
Goto HTML Source and Add the following in second row.

```
<tr>
<td Colspan = "3">
<iframe src = "Home.aspx",
  name = "frameBody" >
</iframe>
</td>
</tr>
```

4. Set the following properties for hyperlink.

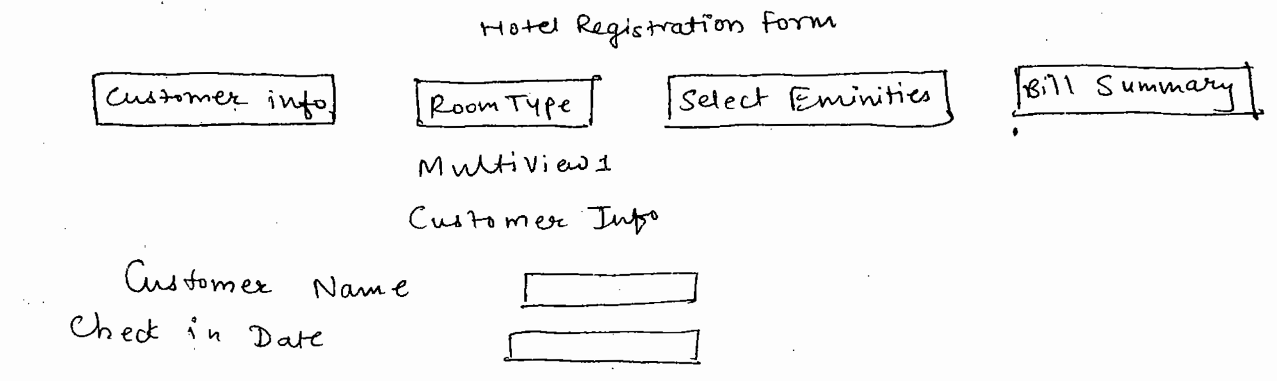
ImageUrl	=	home.png
NavigateUrl	=	home.aspx
Target	=	frameBody

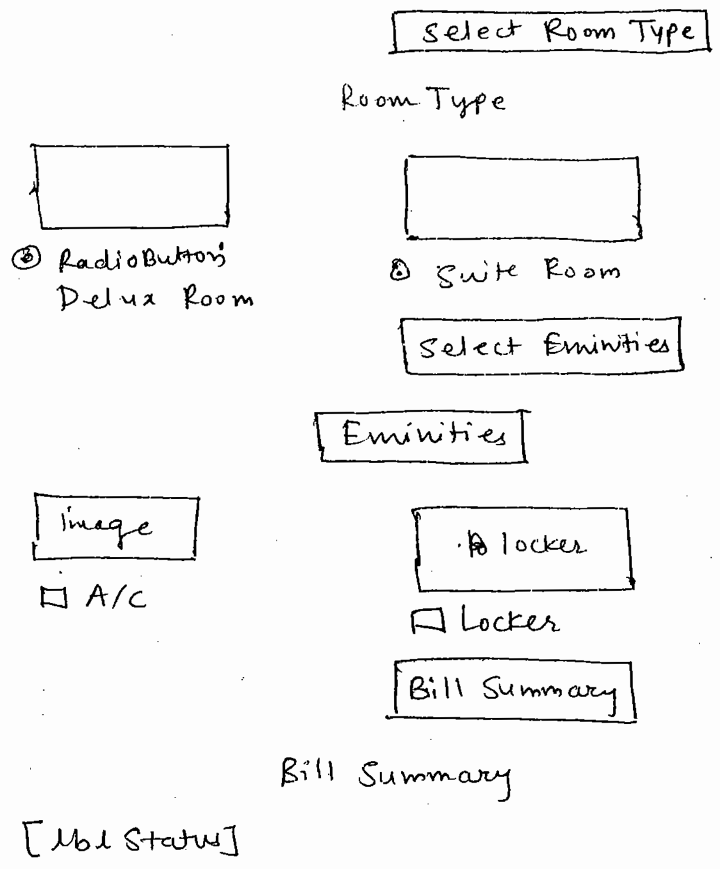
17/06
5. Multiview :- A multiview control is server side control, which is collection of panels and enables the user to switch between the panels during runtime.

Syntax :-

```
<asp: Multiview  id = "multiview1"  runat = "server"
  ActiveViewIndex = "0" >
  <asp: View  id = "View1"  runat = "server".  --Any content--
  </asp:View>
</asp: MultiView>
```

Ex:-.
1. Add a new webform by name "Hotel.aspx".
2. Hotel.aspx (Design)





③ Hotel . aspx . cs (code)

```
// Select Room type button click.
 Multiview 1 . Active ViewIndex = 1;

// Select Eminities  Button Click.
 MultiView 1 . ActiveViewIndex = 2;

// Bill Summary
 int rcost, ecost, total;
 protected void BillSummary _Click (object sender, EventArgs e)
 {
 MultiView 1 . Active ViewIndex = 3;
  if (RadioButton 1 . Checked)
   {
   rcost = 2500;
   }
  if (Radio Button2 . Checked)
  { rcost = 5000;
  }
if (CheckBox1 . Checked)
{
  ecost = 500;
  rcost = rcost + ecost;
}
if (CheckBox2. checked)
{
  ecost = 1000;
      rcost = rcost + ecost;
  }
  total = rcost ;
lblBill .Text = " Total Amount:"+ total .Tostring ("C");
}
```

Note : 1. Set command name for all buttons in Navigation area.

- Customer
- Room
- Eminities
- Bill

2. Open "Command" event for Button and set as Navigation_command.

3. Inside Navigate_command write code

```
Switch (e. CommandName)
{
}   =
```

★ Wizard control: alert()

- On Client Click

★ Wizard Control: A wizard is step by step guidance followed by the user to accomplish any task. In ASP.NET wizard is like a multiView Control that can display multiple pages information. from single page. It is a collection of wizard steps.

Syntax:
```
<asp: Wizard id = "Wizard1" runat= "server",
    HeaderText = " Registration form" >
 <wizard Steps >
 <asp : WizardStep . runat= "server" Title = " Step-1" >
 </asp : Wizard Step >
 </wizardSteps>
</asp : Wizard >
```

Wizard properties:

1. ID
2. Runat
3. HeaderText
4. Wizard Steps (Collection)
5. Display Cancel Button - True/false
6. Cancel Destination PageUrl
7. Finish Destination PageUrl
8. Display sideBar - True/false

★ Wizard Events:

1. Active Steps Changed
2. Cancel Button Click
3. Next Button Click
4. Previous Button Click
5. Finish Button Click
6. SideBar ButtonClick

Wizard Templates :

★ Displaying Summary.

1. Start Navigation Template
   - Next Button
   - Cancel Button
2. Step Navigation Template
   - Next Button.
   - Previous Button
   - Cancel Button
3. Finish Navigation Template
   - Finish Button
   - Previous Button
   - Cancel Button

18/06

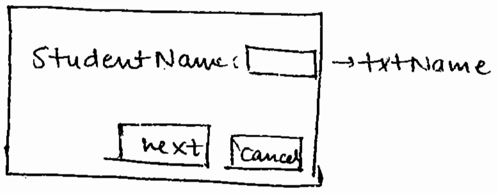
★ Ex: Wizard Control :

1. Create a new webform by name "Courses.aspx".
2. Add Wizard control to form.
3. Goto Wizard properties and select add wizard steps.

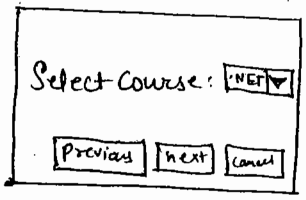
   (properties in Smart Tag)

4. Add following steps :

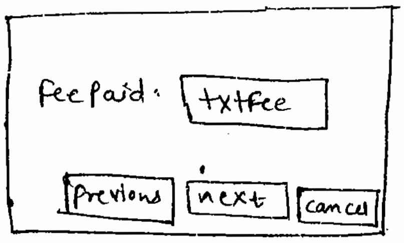
Step: 1) Student info (design)



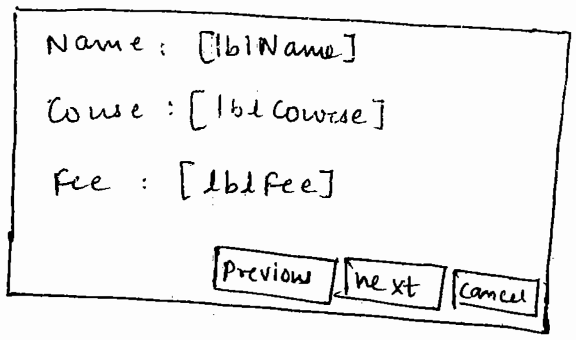
Step 2 : Course Info (Design)



★ Step-3 : Fee Info . (Design)



Step - 4 : Summary . (Design)



5. Set the following properties for Wizard.
   1. DisplaySideBar - False
   2. DisplayCancelButton - True
   3. CancelDestinationPageUrl = cancel.aspx
   4. FinishDestinationPageUrl : Finish . aspx

6. Adding client Script for finish Button Click.
   a) Goto wizard Smart tag
   b) Select "Convert FinishNavigation Template"
   c) Goto Html source of wizard.

```
<finishNavigationTemplate>
<asp: Button ID = "FinishButton" runat = "server"
    Text= " Finish " On ClientClick = " FinishClick () "/>
</finishNavigationTemplate>
```

   d) Write the javascript function in head section of page.

```
<script type = " text/javascript " >
```

```
   function FinishClick()
   {
    alert("wizard Completed");
   }
</script>
```

7. Goto wizard Event and double click on "ActiveStepChanged" event

```
protected void wizard1_ActiveStepChanged(object sender,
                                         EventArgs e)
{
Switch (Wizard1.ActiveStepIndex)
{
Case 0: Wizard1.HeaderText = "Student Info";
        break;
Case 1: Wizard1. Header Text = "Course Info";
        break;
Case 2: wizard1. HeaderText = " Fee Info";
        break;
Case 3: Wizard1.HeadText = "Student Summary";

     lblName.Text = txtName.Text;
    lblCourse.Text = DropDownList1.SelectedItem.Text;
    lblfee.Text = txtfee.Text;
               break;
    }
  }
```

★ <u>Validations</u>: Unobtrusive Validations → uses jquery

jquery → uses Auto correct & Auto Complete
↓
white type it will validate

Validations in web Applications are required to ensure that contradictonary and unautherized data is not get stored into the database.

Validations can be controlled in three different ways:

1) Client side (Using client side script)
2) Server side (Using server side controls)
3) Remote (Using jquery and json)

★ Validations in asp.net are unobtrusive validations from the version 4.5. It uses Jquery and requires and requires jquery script manager for validations

Asp.Net provides the following validation controls.

1) Required field validator
2) Compare validator
3) Range validator
4) Regular Expression Validator
5) Custom Validator
6) Validation summary

The validations in ASP.NET requires unobtrusive Jquery script mapping, which is predefined for a template and requires manual configuration if it is an empty website.

```
protected void Page_Load (object sender, EventArgs e)
{
  page . UnobtrusiveValidationMode = UnobtrusiveVali-
        - datioMode . None;
}
```

Note: The page property "Page.IsValid" returns boolean true if page is valid and have no validation error. If it returns false then it fires up the validation messages.

19/06

Validation: Required Field Validator: It is used to define man-datory fields in a form i.e, the field cannot be null when you submit the form data to server.

Properties

1. ID
2. Runat
3. Control to Validate
4. Error Message
5. Forecolor
6. Text

* Syntax: <asp: RequiredField Validator ID = "rfv 1" runat = "server" ControlToValidate = "txtName" Forecolor = "Red" />

* A compare Validator:- It is used to compare values in two different fields and also the specific datatype entered into the field.

* Properties

1. ID
2. Runat
3. ControlToValidate
4. ControlTo compare
5. Type
6. Operator
7. Error message
8. Forecolor.

★ <u>Syntax 1</u>: Validating password and confirm password.

<asp: CompareValidator id="CV1" runat="server"
ControlToValidate = "txtCompare" ControlToCompare =
"txtpassword" Operator = "Equal" Type = "String "
ErrorMessage = "password Mismatch" foreColor = "Red" />

<u>Syntax 2</u>: Validate date

<asp: CompareValidator id = "CV2" runat = "server"
ControlToValidate = "txtDOB" Operator = "DataTypeCheck"
Type = "Date".
ErrorMessage = "~~Inav~~ Invalid Date"
Forecolor = "Red" />

★ How do we use Range Validator:

It ensures that the input value falls within the specified range.

properties:

1) Id
2) Runat
3) ControlToValidate
4) Minimum Value
5) Maximum Value
6) ForeColor
7) Error Message

<u>Syntax</u>:

<asp: RangeValidator id="rv1" runat="server"
ControlToValidate = "15" Maximum Value = "30"
ErrorMessage = "Age 15 to 30 only"
ForeColor = "Red"
Type = "Integer" />

★ Regular Expression Validator:

Meta character   Quantifiers

(?=.*[A-Z]) ⇒ At least one Upper case Alphabet

[A-Z(?=.*[A-Z]) → At least one Upper case, lower case and number.

(?=.*[0-9]) = At least one no.

(?=.*[{$,@#%&] = At least one

Cannot use '/' it is a group not a individual chara.

{7,15} = length of password. 7 to 15

★ Regular Expression Validator:

A regular expression validator enables the user to validate any field by using a validation expression to build the validation expressions we have to use the meta characters and quantifiers.

Meta Characters :-

Meta Characters	Description
1. \w	Any word i.e. [A-Z a-z 0-9 _]
2. \d	Any number [decimal)
3. \s	Spaces allowed
4. [A-Z]	Upper case Alphabet
5. [a-z]	lower case Alphate
6. [0-9]	Numbers 0 to 9
7. [a-Z] or [A-Za-z]	Upper and lower case both
8. [a, s, d]	Characters in specified range
9. [^a, s, d]	excluding specified characters.

10] [a-mA-M4-8]	Characters in specified range.
11) ?=.*	At least one
12) ~~###~~ \+\@\.\-	Special characters must precide with a "\".

13)

★ Quantifiers :-

	Quantifier	Description.
1.	{n}	specified number of ~~digits~~ characters
2.	{n,m}	Characters range from n to m (min=n , max=m)
3)	{n,}	min=n and max = any number

~~Sy~~

★ Properties of Regular Expression :-

1) ID
2) Runat
3) ControlToValidate
4) ForeColor
5) ErrorMessage
6) Validation Expression

Syntax: Validate 10 digit Mobile no. starting with +91.

```
<asp: Regular ExpressionValidator id= "rev1" runat=
"server" ControlTo Validate = "txt Mobile "
Validation Expression = "\+91[0-9]{10}"
ErrorMessage = " Invalid Mobile".
ForeColor = "Red"/>
```

Syntax: Validate password that must be between (7-15) Characters and with at least one number and special Characters.

~~(?=.*~~

```
<asp: Regular Expression Validator id = "rev2" runat="Server"
ControlToValidate = "txtPassword" ValidationExpression =
"(?=.*[0-9])(?=.*[!@#$%^&*])(a-zA-Z 0-9!
@#$%^&*] {7,15}".
Error Message= "Invalid password Format"
ForeColor = "Red" />
```

★ BulidIn Validation Expression:

Custom validation:

Custom Validator allows any field by using a ~~Cliend~~ client side/ server side function. It comprises of arguments that evaluates to true or false. The Validation m are shown when the argument evaluates to false.

Properties:

1) ID
2) Runat
3) ControlToValidate
4) Error Message
5) ForeColor
6) ClientValidation Function

Event: Server Validate

Syntax: Validating even number :-

```
<asp: Custom Validator id = "CV1" runat = "server"
Control to Validate = "txtform" Error Message = "Not an even
number". ForeColor = "Red" OnSeverValidate = " ServerValidate"/>

  Protected void CustomValidator1_ServerValidate (object sender,
  ServerValidate EventArgs e)
  {
   if ( Convert.ToInt16(args.Value) % 2 ==0)
    {
     args.IsValid = true;
     }
    else
    {
     args.IsValid = false;
     }
  }
```

★ Validation Summary: It makes the summary of all the errors on a page and shows that error messages as bulleted list or paragraph.

<u>Properties:</u>

1) ID
2) Runat
3) ForeColor
4) DisplayMode : List, Bulleted List...
5) Show Message Box
6) Header Text

<u>Syntax</u>: < asp: Validation Summary id = "VS1" runat="server"
Header Text = " Please Check the following errors"
Display Mode = " Bulleted List" Show Message Box = "True" />

<u>Ex</u>: 1) Add a new webform "Register.aspx"

2) (Design)

Registration form

UserName:	txtName	[Required field Validator]
Password:	txtPassword	[compare Validator 1]
Confirm password:	txtConfirm	[compare Validator 2]
Date of Birth:	txtDOB	-
Age:	txtAge	[Range validator1]
Bank code:	txtBank	[Regular Expression Validator1]
Mobile:	txtMobile	[Regular Expression Validator 2]
Email Address:	txt Email	[Regular Expression validator 3]
Enter code:	txtCode	[custom validator1]
img		
Enter Even no:	txtEven	[Custom Validator 2]
	Register	

[lblTitle]

Please Check the following errors: [Validation summary]

- Error Message 1
- Error Message 2
- Error Message 3

⋮

3) Validation Controls and their properties:

a) ~~Required field Validator~~:

	Control	properties
1)	Required field Validator	- ForeColor (Red)
		- Text = *
		- ControlToValidate = txtName
		- Error Message = Name Required

2) Compare Validator1.

- Text = *
- Control to validate = txtConfirm
- Control to Compare = txtPassword
- ErrorMessage = Password Mismatch
- FaceColor = Red
- Type = String
- Operator = Equal

3) CompareValidator 2

→ Text = *
→ ControlToValidate = TextDOB
→ ErrorMessage = Invalid Text
→ ForeColor = Red
→ Type = Date
→ Operator = Data Type Check.

4) RangeValidator1

→ Text = *
→ ControlToValidate = txtAge
→ ErrorMessage = Age 15 to 30
→ ForeColor = Red
→ Minimum Value = 15
→ Maximum Value = 30
→ Type = Integer

5) Regular Expression Validator1

→ Text = *
→ ControlToValidate = txtBank
→ Error Message = InvalideCode
→ Fore Color = Red
→ ValidationExpression =

[A-Z] {3} [0-4] {4} [A-z] {2}

6) RegularExpressionValidator 2
- → Text = *
- → ForeColor = Red
- → ErrorMessage = InvalidMobileNumber
- → ControlToValidate = txtMobile
- → Validation Expression = \+91[0-9]{10}

7) RegularExpressionValidator3
- → Text = *
- → ForeColor = Red
- → ControlToValidate = txtEmail
- → ErrorMessage = InvalidEmail
- → ValidationExpression : Invalid Email (Select from List)

8) Custom Validator 1
- → Text = *
- → ForeColor = Red
- → ControlToValidate = txt Captcha
- → Error Message = Invalid Code Entered

9) CustomValidator 2
- → Text = *
- → ForeColor = Red
- → ControlToValidate = txt Even
- → Error Message = Not an Even Number

10) Validation Summary
- → HeaderText = Please Check following errors :-
- → DisplayMode = BulletedList
- → ForeColor = Red
- → Show MessageBox = True

4. Source Code :

// On Register Button Click Code :

```
if (!Page.IsValid)
    lblTitle.Text = "Registered ....";
```

```
// Custom Validator 1. Server Validate Event code:-
    if (args.Value == "T7AG po 75")
        args.IsValid = true;
    else
      args.IsValid = false;

// Custom Validator 2. server Validate Event code
    if (Convert.ToInt16(args.Value) % 2 == 0)
        args.IsValid = true;
    else

    args.IsValid = false;
```

☆ ~~Sy~~ Styles & Themes :-

Styles are required in web Development to make the pages more responsive and interactive ASP.Net web Applications can use styles in three different ways:-

1) Inline Styles

2) Embeded Styles

3) CSS

1) Inline Style: In style are defined within the element by using a "style attribute". These are individual to every element and cannot be accessed by other elements.

~~6) Regular Expression Validator 2~~

Syntax:

```
<h1 Style = "background-Color : red ; color : White ;
text-align : Centre ; " > welcome to ASP.Net
</h1>
```

2) Embeded Styles : The styles are defined in the head section of page so that they are accessible by all elements within the page.

Syntax:

```
<head>
<style>
~~[illegible]~~ h1
 {
  background-Color : red;
  Color : White
 }
</style>
</head>

<body>
<h1> Welcome to Asp.Net </h1>
</body>
```

22/06

3) CSS (Cascade Style Shades)

The styles are maintained in a seperate style sheet so that they are accessible from any page in the website.

a) Right click on content folder in your website.

b) Select the option "add→new item".

c) Select the item type as style-sheet and name it as demo.css.

d) Write the styles in stylesheet.

```
h1
{
  background-Color : red ;
   color : white
 }
```

e) Link the style sheet to any page.

```
<head>
<link rel = "Stylesheet". href = "~/ Content / Demo.css"/>
</head>
```

★ Smalify or web Essentials.

www. NuGet.org → Download Thirdparty tools for VS.

★ Minification: It is the new ~~concept~~ feature introdu--ced with asp.Net 4.5. It is the process of creating mi--nified version of css and Javascript Files, which will redu--ce the file size by removing unnecessary blank spaces, converting lengthy variables and logics ~~to~~ into shortcut form.

Ex: 1. Download any minification tool.

Ex. Smalify.

2. Open Smalify tool. drag and drop your content folder into Smalify

3. Click "minify now".
4. Minified version of css and Js files are created.

Ex. Sample.min.css
 Sample.min.Js.

6. Link the minified file to your webform.

```
<link rel = "Stylesheet" href = "~/Content/Sample.min.
                                           css"/>
```

★ Bundling:

System.web.optimization → required for bundling

tool → library package manager console → Package manager console. there ~~ta~~ write

 Install-package microsoft.asp.net.web.optimization

Sol^n Explorer → references → system.web.optimization.

★ Add stylesheets: Dont want to use directly

 App-start → Bundleconfig → ~~Bundle~~.

Bundling → reduce request numbers.

★ Bundling: It is one of the key feature of ASP.NET introd-uced with the version 4.5. You can reduce the no. of request while accessing multiple css or Js files, by making a bundle of styles or scripts.

It improves the performance of an application and reduces burden on server.

The assembly system.web.optimization provides a collection of "style bundle" and "script bundle."

Ex. 1) Create a new Asp.Net Application

2) Add following ~~Sta~~ style sheets into content folder

1) headstyles.css.

And write the following code:

```
.h1
{
background-color: green;
 color: white;
 text-align: centre
}
```

2) parastyles.css

And write the following code into that

```
p
{
 background-color : Yellow;
 text-align : justify
}
```

3. Goto the folder "App-start".

4. Open the file "Bundle Config.css" and create a style bundle

```
public class BundleConfig
{
public static void RegisterBundles (BundleCollection
                                        bundles)
  {
  bundles.Add (new StyleBundle ("~/Content/Demo").
      Include (" ~/Content/headStyles.css",
       "~/Content/parastyles.css"));
  }
```

↑ Param array concept is used

5. Add a new webform "Home.aspx" with some heading and paragraph we have to link a bundle,

In Home.aspx (HTML Source) → in head Section →

```
<head>
<%: Styles.Render("~/Content/Demo") %>
</head>
```

if it is script bundle then write script instead of style and rest of the code will be same

★ **Bootstrap**: www.getBootStrap.com → Justified Nav.css
↓
To adjust contents according to size of screen

★ How to use the styles?

★ Bootstrap is a repository of css and javascript files. It provides 1000's of predefined styles and scripts that you can import and use with your application.
Microsoft integrated bootstrap services from the version 4.5.

Ex: Using css :-
1. Visit the website www.getBootStrap.com.
2. Click on css category. Select any specific css like menu buttons, header, etc.
3. Goto Examples of selected category.
Ex. http://getbootstrap.com/examples/justify-nav/justified-nav.css.
4. Copy the css code.
5. goto your website and add a new style sheet by name justified-nav.css.
7. Paste the copied css code into the file.
8. goto bootstrap website and copy the html code for the style you selected.

9. Paste the html code into your html page. i.e, "Home.aspx".

10. Link the following files to your page.

- bootstrap.css
- justified-nav.css
- bootstrap.js

23/06

<u>Syntax for writing styles</u>:

★ Writing styles in a webform

<u>Syntax</u>:
```
selector
{
attribute: value;
attribute: value;
}
```

★ Types of selectors in css.

1. Type selector :- It refers the element name to which you want to apply the styles.

<u>Ex</u>:
```
h1
{
background-Color: Green;
Color: white;
}
```

<u>Note</u>: The styles are applied to all h1 elements in the page. You cannot ignore any element.

2. ID Selector : It refers to an "id" so that the styles are accessible with reference of "ID". The styles are affected only to the elements that are using the "ID".

<u>Syntax</u>:
```
#headings
{
background-Color: red;
Color: White;
}
```

```
<h1 id = "headings"> Welcome </h1>
<p id = "headings"> para-1</p>
```

Note: Every element can have only one id i.e multiple styles cannot be applied to a single element through "ID"s. Then solution for that is class selector.

3) Class Selector: The styles can be defined as classes so that a single element can implement multiple classes.

Ex:
```
.backStyle
{
 background-color: Red;
}

.textStyle
{
 color: white;
 text-align: centre;
}
```

```
<h1 class = "backStyle textStyle"> welcome </h1>
```

Note: Multiple styles are seperated using space.

4) Descendent Selectors: It refers to a child element to which you want to apply the styles

Ex:
```
ol > li
{
 Color: Red;
}
<ol>  // Order list(ol)
<li> Item 1 </li>
<li> Item 2 </li>
```

5. Attribute Selector: You can apply the styles to an attribute of any element so that element with same type of attri-bute will acquire the styles.

```
input [type=text]
{
 background-Color : red;
 color : white;
}
<input type= "text" name= "txtName">
```

★ Background Styles :-

Ex:

- Background-color
- Background-attachment
- Background-image
- Background-position
- Background-repeat.

Ex :

```
<Style>
 body
 {
 background-image: url ('images/1.jpg');
 background-repeat: repeat;
 background-attachment: fixed;
 font-size: xx-large;
 }
</style>
```

★ Themes:

Q:- when the themes are applied to a page?

→ On "page-preinit" not on page_Load.

and write Page.Theme = "ThemeName"; int Page-Prei
event.

*Themes:- Themes are also ~~set of~~ attributes defined for elements to control their behaviour and apperance dynamically during run-time.

Themes are defined in a "skin file" that have the extension ".skin". And all skin files must be maintained in the folder "app_themes".

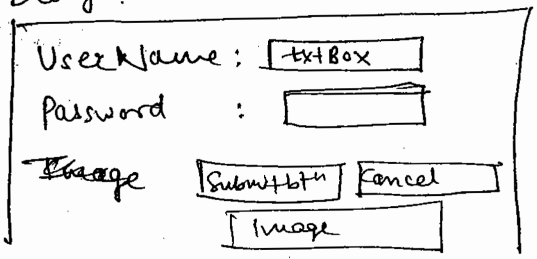
The themes are applied dynamically to a page on or before the "preinit" ~~#~~ i.e, "page_preinit" event.

Themes can be applied to a page by using the "theme" property in page directive and by using "Page.theme" attribute.

Ex: 1. Right click on "Project Name".
2. goto "add→new item"
3. Select Skin file.
4. Name it as "independence.skin".
5. Write the following code in skin file:

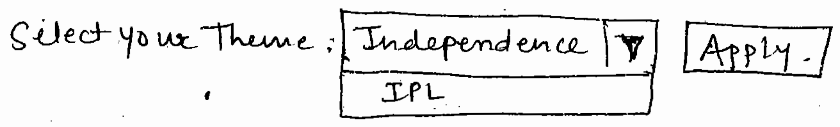
```
<asp:TextBox runat = "server" BackColor = "Organge" Forecolor =
    "Black" />    ⇒// for this we dont give id & text
<asp : Button runal= "server" BackColor = "Green" ForeColor =
    "White" />
<asp: ~~Butt~~ Image runat = "server" ImageUrl = "~/Images/
    ind.png" />
```

6. Similarly add another skin file by name "ipl.skin".
7. Add a new web form by name "login.aspx".
8. Design:



8. Add another webform by name "Select_theme.aspx".

9. (Design) :-



10. Apply Button Click code

```
Context.Items.Add ("theme", Dropdownlist1.SelectedValue);
Server.Transfer ("Login.aspx");
```

// Goto Login page, Add the following event.

Change page_Load to Page_Preinit.

```
protected void page_PreInit (object sender, EventArgs e)
{
  page.Theme = Context.Items ["theme"].ToString();
}
```

24/06 ★ <u>Master Page</u> : Master pages are blueprints for a website or application. Every website may ~~contains~~ contain a collection of several pages and all pages must be uniform in Appearance, which can be achieved by using master page.

1) Master Pages in Asp are derived from the base master directive "@Master". It provides the set of attributes to configure the master page.

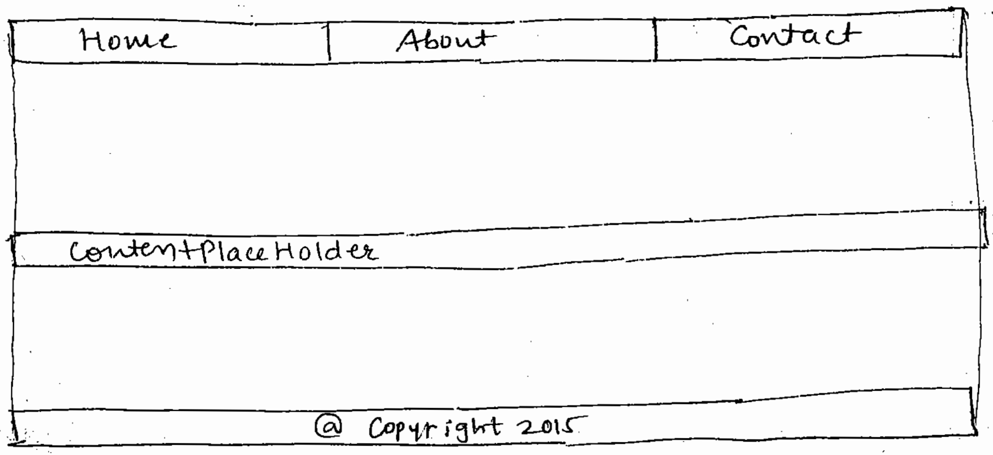
2) The major components in a master page are "<u>directives</u>, <u>markup</u>, and <u>content placeholder</u>.

3) ContentPlaceHolder In master page describes the location where the child page content is rendered.
Every master page can have more than one contentPlace holder.

4) The child pages that are using master page cannot contain markup. Entire information is present inside the "content" control

★ Using Master page for a website:

1. Create a new Asp.Net Application
2. Add new item
3. Select the item type as "webform master page"
4. Name the page as site.master → Traditional Name
5. *Site.Master (Design)



6. Html Source (Site.Master)

```
<%@Master Language ="C#"%>
<!DOCTYPE html>
<html>
<head runat = "server" >
<asp: ContentPlaceHolder ID="head" runat= "server" >
</asp: ContentPlaceHolder>
</head>
<body>
<form id = "form 1" runat = "server" >
<div>
<asp: ContentPlaceHolder ID = "bodyContent" runat= "server" >
```

```
</asp: ContentPlaceHolder>
</div>
</form>
</body>
</html>
```

7. To add Child pages with master page.
   - Select "Add new Item"
   - Select "Webform with MasterPage" or "Webform using masterpage".
   - Give the name as "Home.aspx"
   - Select Master Page file as "site.Master".

8. How to add a master page for existing webforms:-
   - Goto Webform ~~Desio~~ Html source.
   - Remove the markup (complete html code)
   - Add the following in page directive.

```
<%@ page language = "C#" MasterPageFile = "~/site.
                                          Master" %>
```

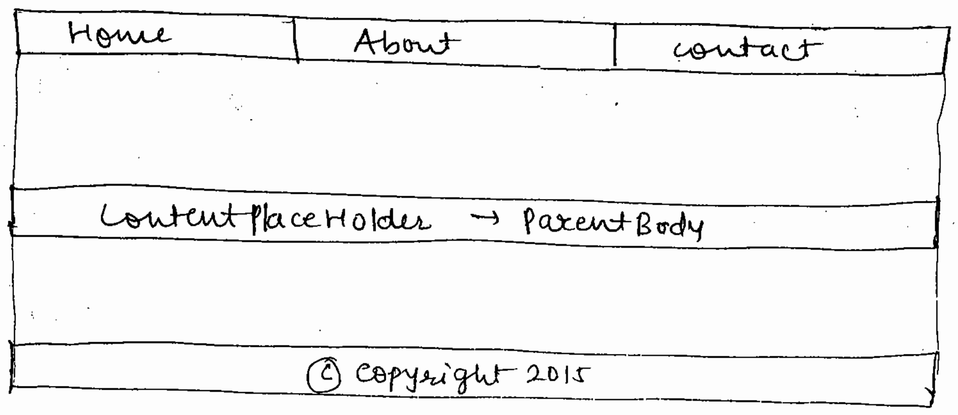
   - Add a content control and put your information in "content".

```
<asp: Content ID = "C1" runat = "server" ContentPlaceHolder
    ID = "bodyContent" >
   <h1> Naresh IT - Home </h1>
  </asp: Content >
```

★ Nested Master Pages :-

Master pages are supported with "multilevel Inheritances". i.e A master page can implement another master page while the webforms will use the derived master page so that it aquires the properties of both parent and child master page

1. Add new item to your website
2. Select webform master page. Name it as "parent Master"
3. (Design) parent. Master :



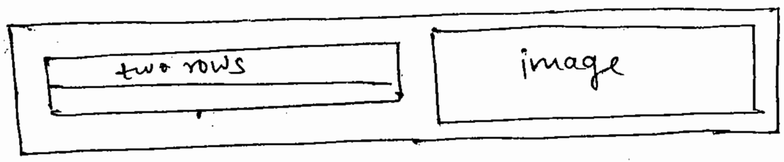
4. Add another new item to website
5. Select "webforms master page (nested)
6. Give the name as "Child.master".
7. Select the master page file as "Parent_Master".
8. Goto Child. Master (html source)
9. Add a content PlaceHolder ~~(ID~~

```
<asp: Content ~~PlaceHolder~~ ID = " Content2 "
 ContentPlaceHolder ID = " ParentBody " . runat= "server" >
 <asp: ContentPlaceHolder  ID= "ChildBody" runat= "server">
 </asp: ContentPlaceHolder >
 </asp: Content >
```

10. Now add the following design to "Child. Master".

11) Child Master page (Design)

In ContentPlaceHolder



12. Add New item :
12. Select Webform with Master Page "
13. Name it as "Home.aspx".
14. Select the master page as "Child. Master"
15. Clicked "Add".
16. Goto "smart-tag" of "child.master" page Body and select .
"Default to MasterPage".
(Click Yes to Continue)
17. Again Goto smart-tag and select "Create Custom Content".

25/06
★ State Management → Roslyn Compiler
1. Http is a stateless Protocol. It cannot remember inform-
-ation between pages.
2. It uses the mechanism go - get - forget.
2 Go : Sends request to server.
Get : gets response from server
forget : Performs cleanup i.e, It removes information of page from server, After sending the response.
3. The Stateless nature of http is an advantage to server as it will reduce the burden on server, but it is an drawback for client because, client must remind the Server about information between pages.

• This requires the implementation of several state management techniques, like
1. Context
2. querystring
3. Cookies
4. Session
5. Application
6. ViewState
7. Cache

★ **Context:** The context memory of any webapplication is avail-able only with the context of previous page. It cannot span information to multiple pages and also cannot be used to acess information across websites.

Context provides a dictionary type collection that allows to store values under the reference of keys and are accessi-ble by using the keyname.

★ **Syntax:** Creating an context object.

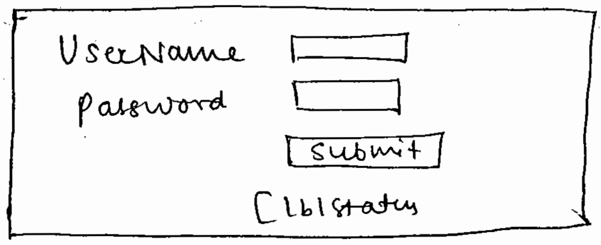
$\text{Context.Items.Add(}\underbrace{\text{"Key", "Value"}}_{\text{object type}}\text{);}$

★ **Syntax:** Access the context values.

Context.Items.["KeyName"];

Ex: Add following pages to website
1. login.aspx
2. Welcome.aspx

2) login.aspx (Design)



8) login button click code :-

```
if ( txtPassword.Text= "admin" )
 {
Context.Items.Add ( "uname", txtUserName, Text) ;
 Server.Transfer (" Welcome.aspx") ;
  }
  else
 {
lblError.Text = "Invalid Password" ;
 }
```

4. Welcome.aspx (Design)

[lbltitle]

5. Welcome.aspx.cs (code)

```
// Page-load event code.
 ~~if(~~
 lblTitle.Text = "Hello" + Context.Items ["uname"].ToString();
```

★ Query string:-

A query string is passed in the address bar of browser & it have the ability to transport values across websites However, it is not secured as it exposes the data in the address bar and it is accessible from any page.

A request object is required to collect the values from query string.

The size of query string will vary according to the browser

~~IE~~ Browser	query string length	
IE	2083 , 2048	
Firefox	65,536	characters
Safari	80000	
Opera	1,90,000	
Apache	4000	

Netscape → They all belongs to the Netscape (Browsers)

Syntax :- Creating query string

Page.aspx ? Key = Value & Key = Value

Syntax : To access query string value

request.querystring ["key"~~Name~~]

Ex :- Add following pages to website.

- Login.aspx
- Welcome.aspx

2. Login.aspx.cs (code)

```
// Login Button Click code.

  if (txtPassword .Text == "admin")
   { response.redirect ("welcome.aspx ? uname=" +
  txtUserName.Text + "&pwd=" + txtpassword.Text);
  }
  else
   {
 lblError.Text = "Invalid Password" ;
   }
```

3. Welcome.aspx.cs (code)

~~lbltitle~~ page_Load event code

```
lbltitle .Text = "Hello ! " + Request . Querystring ["uname"] +
"<br>" + " Your Password : " + Request. Querystring ["Pwd"];
```

★ Configuration property ~~Attribute~~ :

msdn . microsoft . com

Javascript document objects .

26/06 Cookies : 1. Cookies are simple text files.
Server stores the information of client in cookie so that it can access the client information automatically from cookies.

2. The cookies represent a collection defined by the class: "http Cookie".

3. The cookies can be a -
a) Inmemory (Temporary)
b) Persistant (Permanant)

Some of the drawbacks of cookies are -
a) Many browsers do not support cookies
b) They can be infected with virus.
c) They do not exihibit cross platform (Not understandable to all the browsers)
d) Their functionality may vary according to the browser memory
e) Creating and manipulating of cookies includes following steps -

1) Step 1 : Create a cookie

Syntax: httpCookie obj = new httpCookie("name");

Step 2 : Assign value to cookie.

Syntax; obj.value = value

Step 3 : Set expiry Date for cookie

Syntax: obj.expires = DateTime

Step 4 : Add cookie into memory

Syntax: Response.AppendCookie(obj)

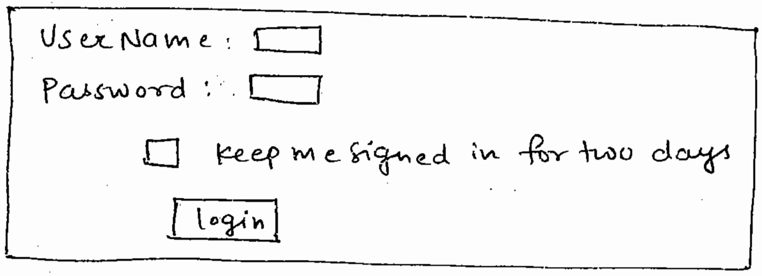
★ Note: The cookies in memory are accessible by using their their index or theme

Syntax:

```
Request.Cookies ["KeyName"];
```

Example:

1. Add following pages to website
   - login.aspx
   - welcome.aspx.
2. login.aspx (Design)



3. login.aspx.cs (code)

```
// login button Click code
protected void Button1_Click(object sender, EventArgs.e)
 {
  if ( CheckBox1. Checked)
  {
  Http Cookie obj1= New HttpCookie ("uname");
 Http Cookie Obj 2 = new HttpCookie ("pwd");
  obj1 .Value = txt UserName. Text;
  obj2. Value = txtPassword. Text;
  obj1 . Expires = DateTime. Now. AddDays (2);
  obj2. Expires = DateTime. Now. Add Days(2);
 Response. Append Cookie (obj1);
 Response. Append Cookie ('obj2');
  }
server. Transfer (" welcome.aspx");
```

}

4. // welcome.aspx (design)

[lblUser]

5. Welcome.aspx.cs (code)

```
protected void Page_Load (object sender, EventArgs e)
{
HttpCookie obj;
    obj = Request.Cookies ["uname"];
    lblUser.Text = "Hello!" + obj.Value;
}
```

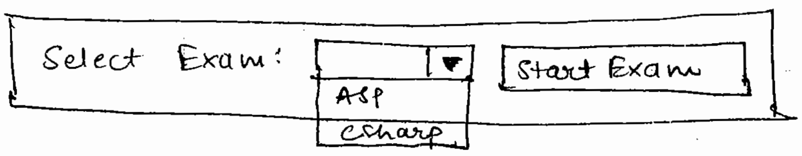
Example 2 : Example for inMemory Cookies

1. Add following ~~pages~~ ~~to~~ folders to website
   - ASP folder
   - CSharp folder.

2. Add a new webform
"Select_Exam.aspx"

3. Add following pages into ASP folder
   - Q1.aspx
   - Q2.aspx
   - Q3.aspx
   - Result.aspx

4. SelectExam.Aspx (design)

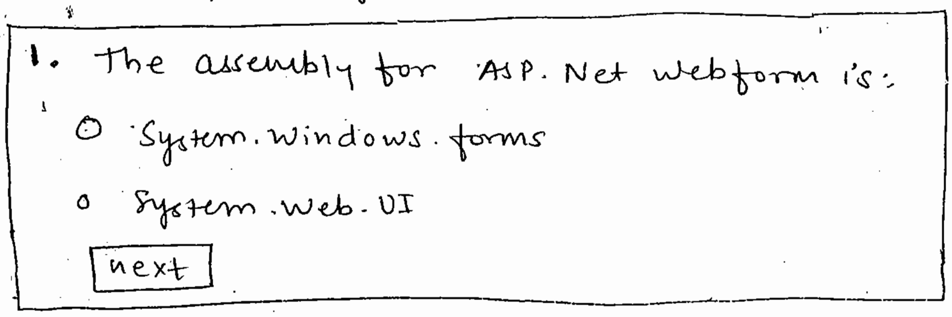


5. Code for this :
Select_Exam.aspx (code)

★ Start Exam Button Click code

```
Switch (DropdownList1.SelectedIndex)
  {
  case 0: Response.Redirect("~/Asp/Q1.aspx");
          break;
  case 1: Response.Redirect("~/CSharp/Q1.aspx");
          break;
  }
```

6// Q1.aspx (Design)



1. The assembly for ASP.Net webform is:

O System.windows.forms

O System.web.UI

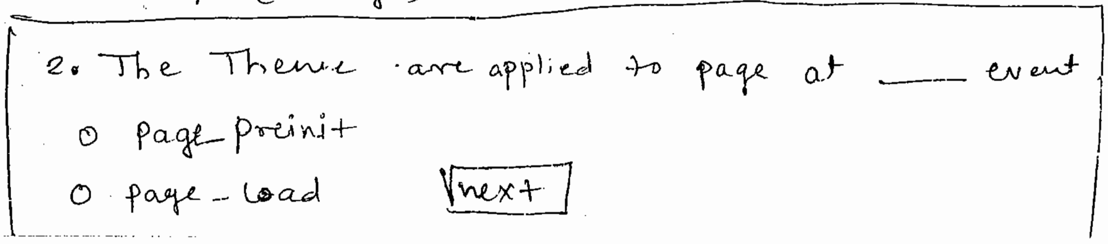
next

7. Code :- "Q1.aspx.cs"

nextButton Click (code).

```
HttpCookie obj = new HttpCookie("Q1");
 if (RadioButton1.Checked)
  {
   ~~obj.~~ obj.value = "n";
  }
  if (RadioButton2.Checked)
  {
  obj.value = "y";
  }
  Response.AppendCookie(obj);
  Server.Transfer("Q2.aspx");
```

8. Q2.aspx (Design)



2. The Theme are applied to page at ____ event

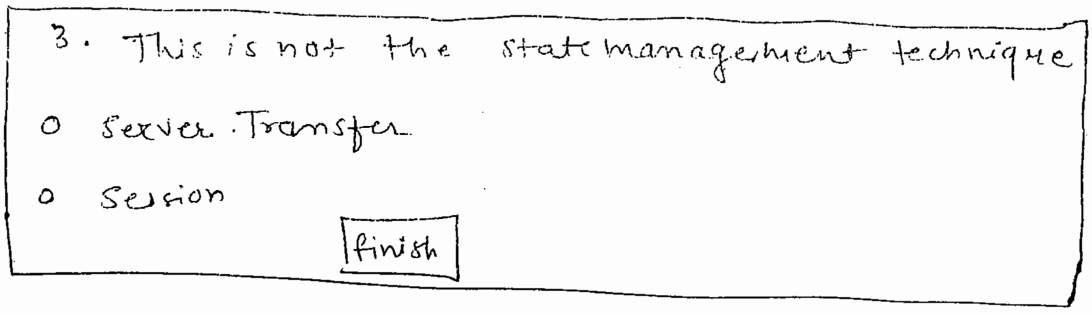
O Page_Preinit

O Page_Load   next

9. Code for Q2.aspx.cs :-

```
~~I~~' HttpCookie obj = new HttpCookie ("Q2");
    if (RadioButton1.Checked)
    {
      obj.Value = "Y";
    }
    if (RadioButton2.Checked)
    {
      obj.Value = "N";
    }
    ~~r~~response.AppendCookie(obj);
     Server.Transfer("Q3.aspx");
    }
```

10. "Q3.aspx.cs" (Design)



3. This is not the state management technique

o Server.Transfer

o Session

Finish

11. Code for "Q3.aspx.cs":-

finishButton_Click Code :

```
HttpCookie obj = new HttpCookie ("Q3");
 if (RadioButton1.Checked)
 { obj.Value = "Y";
  }
 if (RadioButton 2.Checked)
  { obj.Value = "N";
 }
 ~~Se~~ response.AppendCookie(obj);
    Server.Transfer(" Result.aspx");
 }
```

12. Result.aspx.(Design)

```
[lblResult]
```

13. "Result.aspx.cs" (code)

```
// Page-Load event code
  int count;
  HttpCookie obj;
  String s = " ";
Protected void Page-Load (object sender, EventArgs e)
{
 for(int i=0; i< ~~response~~ request.Cookies [i]; i++)
 {
  obj = Request.Cookies [i];
  if (obj.Value == "Y");
   {
    count = count + 1;
   }

 s = s + "Q" + (i+1) + "." + obj.Value + "<br>";
  }
 lblResult.Text = "<br> " + " Your ~~total~~ Ans ~~score~~ = " + "<br>"
 + s + "<br>" + " Total correct =" + count;
}
```

★ Q. How to check if cookies are enabled or not on browses

→ Using "navigator object".

Ex: ~~<head~~

```
<!DOCTYPE html >
<html>
<head  runat = "server" >
<script type = " text/javascript " >
  function  CookiesStatus( )
{
document . getElementById ("cookie") . innerHTML = "Cookie
Enabled:"+ navigator.cookieEnabled ;
}
</script>
</head>
< boady  onLoad= "CookiesStatus()">
< form Id = "form1" runat= "server" >
<div id = " cookie">
</div>
</form >
</body>
```

★ Application & session :

Single Call : It is a remoting technique where the server system creates an object for every client to access the application, and this individual object is reffered as session object. It contains the declarations that are ~~b~~ accesible from session start to session end.

★ Singleton: It is also a remoting context where an object is created for the first client request and the same object is provided to multiple clients. This is ~~reff~~ referred as an application object, which is accessible from any session.

★ Application Object: It represents a singleton technique that provides access to the resources from any client.

Syntax:

```
Application ["key"] = Value;
```

Note: All application objects are defined in a global appli-
-cation class file (global.asax). It contains the following application events. i.e. Application_Start, Application_End, Session_start, session-End, Application_error.

Example : 1) Goto global.asax file. and write the following code:

```
void Application_Start( object sender, eventArgs e)
    {
     Application ["n"] = 0;
    }
 Void Session_Start ( object sender, eventArgs e)
  {
  Application ["n"] = (int) Application ["n"] + 1;
Void Session_End ( object sender, eventArgs e)
 {
  Application [" n"] = (int) Application ["n"]-1;
 }
Void Application_End (object sender, eventArgs e)
{
```

```
void Application - End (

{
 Application ["n"]=0;
 }
```

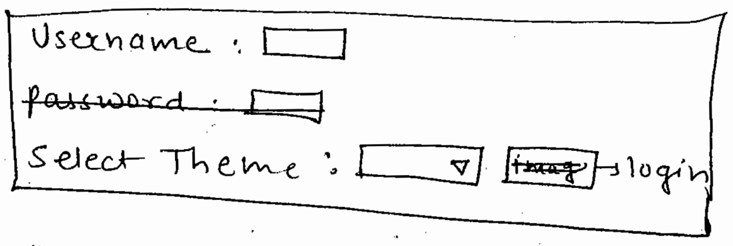
2. Add a new webform "Home.aspx"
   ~~[lbltitle]~~ [lblStatus]
3. Home.aspx page-Load Event code.
   lblStatus.Text = "You are User No :" + Application["n"];

☆ session :
The session state will use a single call mechanism where an Object is created and it is made available from session_start to session_end.
The following methods are used to abandant and remove the sessions.
a) Session. Abandon();
b) Session. RemoveAll();
c) Session. Remove();
d) Session. Add();

☆ Examples
1) Create following themes in your website.
   - Independence.skin
   - Ipl.skin
2) <asp: Image runat = "server" ImageUrl = "~/Content/ Ipl.png">

2) Add following pages to website.
   - Login.aspx
   - Inbox.aspx
   - Contact.aspx

* login.aspx (Design)

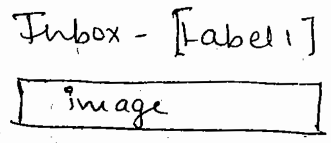


* Login Button Click Code.

Login.aspx.cs.

```
Session["theme"] = Dropdownlist1.SelectedValue;
Session["~~theme~~ uname"] = TextBox1.text;
Server.Transfer("inbox.aspx");
```

2) Inbox.aspx (Design)



Contacts → linkbutton / hyperlink.

event: Page_Load, Page-Render.

* Inbox.aspx.cs code

* Page-Load event code.

```
Label1.Text = Session["uname"].ToString();
// page_preinit code
Page.Theme = Session["theme"].ToString();
```

/ Contact:

LinkButton_Click code

```
Server.Transfer("Contact.aspx");
```

7. Contacts.aspx (Design)

Contact - [lbl1]

| image |

[Signout] → linkButton

8. Contacts.aspx.cs (Code)

// Page_Load and Page_PreInit code is same as inbox.aspx

// linkButton Click code

```
{ ~~Server Transf~~
    Session.abandon();
  Server.Transfer("login.aspx");
}
```

★ View State :-

Viewstate is a dictonary type collection that allows you to save and restore the values of a server control across multiple request for the same page

Syntax:

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

[msdn. microsoft. com] ⟹ Download.

★ Add a new webform "facebook.aspx" (Design).

★

| image |

[label1] Like → Button

3. facebook.aspx.cs (code)

Page_Load Event code

{

```
    if (!page.IsPostBack)
    {
  labael1.Text = "Be first to like";
    }
// Link Button Click code.
  int ClicksCount = 1;
  protected void LinkButton1_Click (object sender, EventArgs e)
  {
   if (ViewState ["Clicks"] != null)
    {
    Clicks Count = Convert.ToInt16 ( ViewState ["Clicks"]) + 1;
    }
  label1.Text = ClicksCount + "Like (s)";
  ViewState ["Clicks"] = clicksCount;
   }
```

★ Cache & Memory Buffer.

Caching: Cache is ~~one~~ one of the statemanagement technique that allows the server to store frequently accessed data into the buffer so that it can be accessed from buffer without the server interaction.

If the page or data is required very frequently, then it can [illegible] in the cache and accessible within a span of specified time interval

Ex: Add a new webform "Demo.aspx"

Design : [161 I]

Goto html source of Demo.aspx and add the following code

```
< %@ Output Cache Duration = "30" VaryByParam = "None" %>
```

3. Page-Load Event Code:

Label1.Text="Page Accessed on:-"+ DateTime.Now.ToString();

29/06 Design Patterns:

The design patterns are solutions to software design problems that you find in real world application development. They are about reusable designs and interaction of objects.

The design patterns are categorized into three groups-

a) Creational
b) Structural
c) Behavioural.

① Creational: The creational patterns deal with instantiation that is creating of objects.

Ex: 1) Abstract factory
2) Builder
3) ~~Str~~ Singleton
4) Factory Method
5) Prototype

② Structural patterns: The structural patterns deal with ~~Crea~~ designing of classes. They are:-

1) Adapter
2) Bridge
3) Composite
4) Decorator
5) Facade
6) Flyweight
7) Proxy

(3) Behavioural patterns: The behavioural patterns define scope of object. It includes the location where and object is created and how it consumes the resources.

Ex 1. Chain of responsibility
2. Command
3. Interpreter
4. Iterator
5. State
6. Mediator
7. memento
8. Observer
9. Strategy
10. Template Method
11. Visitor.

★ Architectural patterns: ~~tti~~ It will describe how an application will run. Applications are build using a layered architecture (3-layers) and Applications run in a tiered architecture :-

Ex : 3 Tier
MVP
MVC
MVVM -- SPA -- Twitter

★ ADO. Net: (Activex Data Objects)

It is a framework that provides a set of classes, which are responsible for communication between the application & the database in multitier Architecture.

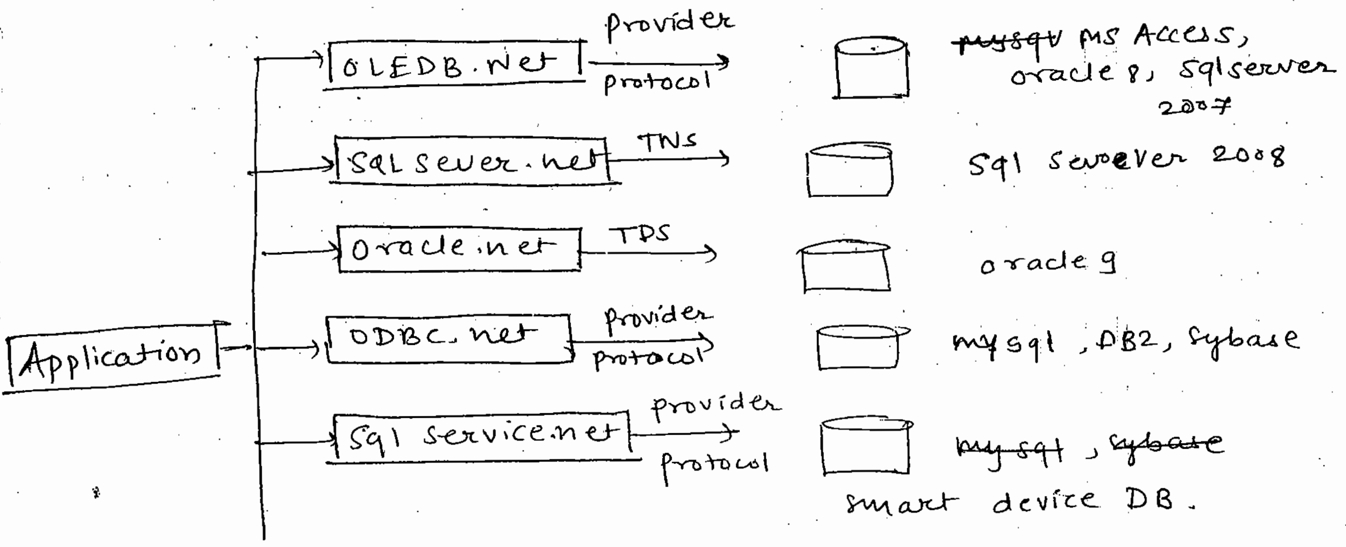
ADO. Verses ADO.NET

~~Feature ADO~~	ADO	~~ADO.NET~~ ADO.NET
primary Aim	client server coupled i.e. connected	Disconnected Architecture
form of data in memory	Uses Recordset	uses Dataset
Disconnected access	Uses Recordset object & connection object with OledB	Uses DatasetCommand object with OleDB
Disconnected access across multitiers	Uses com to marshal Recordset	Transfers Dataset object via XML.
XML capabilities	XML aware	XML is a native transfer medium for objects
firewalls	firewall block system-level com marshelling	XML flows through the firewall via http
Code	coupled to the langua-ge used, various implementation.	Managed code library - Uses Common lang. Runtime therfore lang agnostic

* Data Providers : The data providers are responsible for comm-unication betn Application & Database. There are several databases globally used by various clients. However, there is no single provider that can communicate with all databases. Microsoft introduced the following datapro-viders -

1) OledB.net
2) Sqlserver.net
3) oracle.net
4) ODBC.net
5) SQL service.net

Data Provider



OleDB : Object linking & embaded dB.

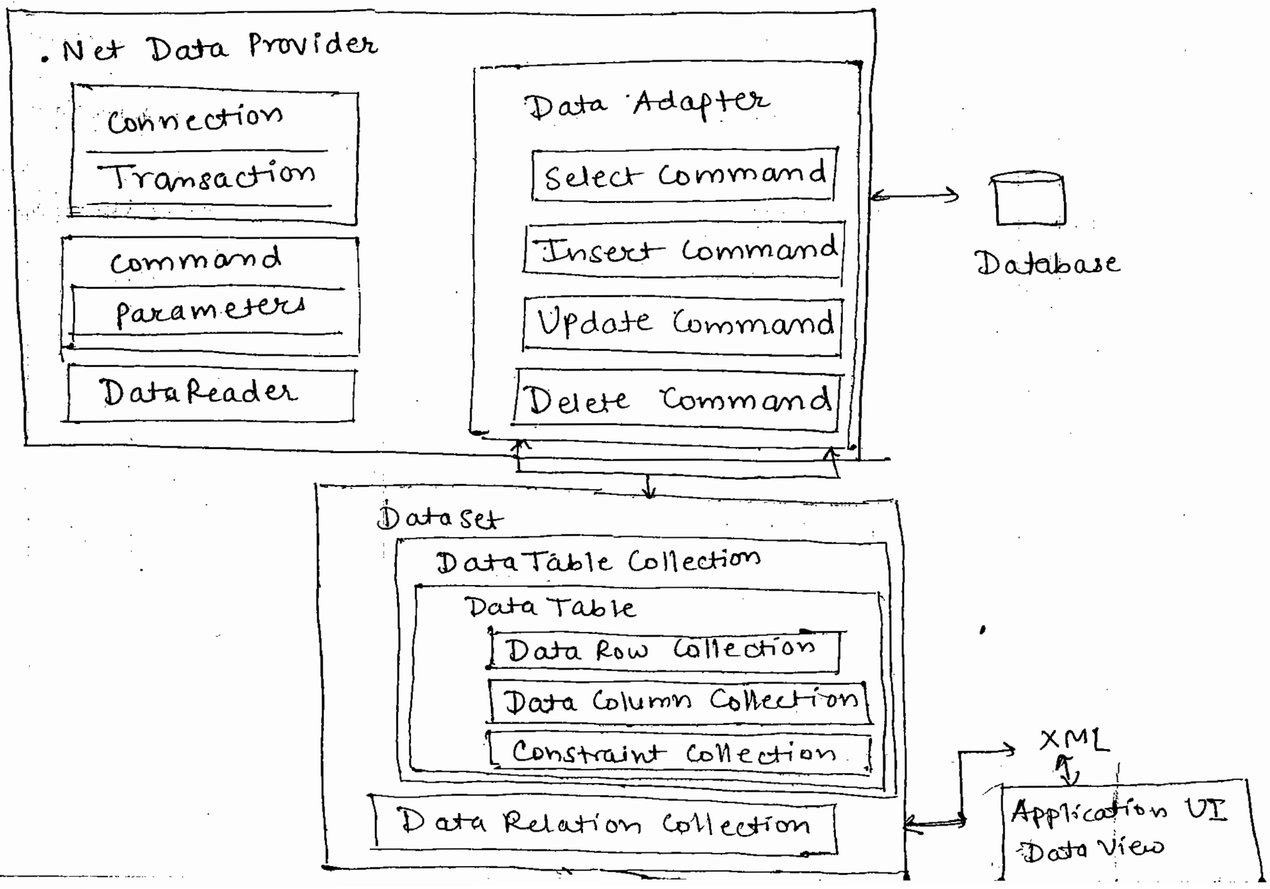
ODBC : Open database connectivity

TNS : Tabular network stream

TDS : Tabular data ~~strea~~ stream

☆ TDS : Tabular Data Stream

ADO.Net Architecture Diagram



- The connection object provides set of properties and methods to connect with the database
- The connections can be opened explicitly and implicitly
- The query is submitted to database by using command object
- The data Reader object is read only and forward only. It can only read the data from the data source
- The adapter object implicitly opens the connection and execu--tes the command.
- Adapter requires a dataset or a DataTable to store the data.
- Dataset is in memory database, which contains a collection by tables and relations.
- DataTable represents a recordset, which contains only one Table
- Adapter requires a dataset or table but they can be used without adapter
- Dataset need not required Adapters to communicate. They can take help of the command object

☆ OLEDB Data Provider :

- The OLEDB provider is responsible for connecting with the databases like MS-access, Oracle 8, sql server 2008 & below
- The OLEDB classes that are responsible for communica--ting with database are defined by the assembly – "System. Data. Oledb".
- The classes are :-
  - Oledb Connection
  - Oledb Command
  - Oledb Data Reader
  - Oledb DataAdapter

MSDRORA :
Microsoft Data Adapter for Oracle

30/06

* OledB.Net Data providers:

The oledB provider is responsible for connecting with the databases like * MsAccess, Oracle 8 and sql server 2008 below.versions.

The OledB classes that are responsible for comm--unicating with database are defined by the assembly "system.data.OledB.

* The classes are -

1. OledB Connection
2. OleDBCommand
3. OleDBDataReader
4. OleDBDataAdapter.

1) OledB Connection :-

1) The connection class is derived from the base "db Connection."

2) It provides a ~~sett~~ set of properties and methods to connect with database

3) Connections can be opened in two ways.

a) - Implicitly - using 'DataAdapter

b) - Explicitly - using ~~Data~~ 'Open method.

4) Members → Connection class members -

- Open → opens connection
- Close → Closes connection
- Dispose → Closes and erases the connection traces
- State → ~~Shows to~~ gets the connection status.

~~Open~~
and that connection status are -

1) open()
2) close()
3) fetching()
4) broken()
5) Connecting.

<u>Syntax</u>:

OledbConnection con = new OledbConnection ("Co-
-nnectionString");

<u>Connection String</u> :-

1. provider = providerName;
2. Datasource = DatabaseName;
3. Userid = UserId;
4. Password = Pwd;.

Datasource	provider Name
1. MsAccess 2003 (.mdb)	Microsoft.jet.Oledb. 4.0
2. Ms Access 2007 (.accdb)	Microsoft.ace.Oledb. 12.0
3. Orale	MSDAORA
4. Sql server	SQLOLEDB.

<u>Ex</u>:

1. Open Ms Access.
2. Select "Blank→Database"
3. Specify the name and location for database
   Ex. F:\\ ProductsDb.accdb.

4. Click create database
5. Goto Create menu and select table.
6. Save the table by name "tblProeuct"

Field p	Datatype
1. ProductID (PK)	AutoNumber
2. Name	Text
3. price	Currency

7. Double Click on table name to add records into the table~~Name~~
8. Create a new webApplication
9. Add a new webform "Demo.aspx".
10. Add a button control with text "Connect to MsAccess".
11. write the following code for connect button click event.

```
// Import the Name Space
  Using System.Data.OledB.
// Button_click code:-

Oledbconnection con= new OledbConnection("provider = Micro-
-soft.ace.Oledb.12.0 ; Data Source = ~~Products~~ F:\\ Products
Db.accdb);
  con.Open();
 con.Close();
Response.write(" Connected ...." + "<br>" + "Connection
  Status = " + con.State);
```

★ Writing connection strings in webApplication:-

It is always recommended not to write the connection string in page, you can use any one of the following methods -

a) Writing in global.asax
b) Writing in web.Config

Ex: For global.asax.

1. Open global.asax and Add a connection string as Application, object

```
Void Application_Start (object sender, Eventargs e)
{
Application ["dbCon"] = @"provider = Microsoft. ace. ole-
-db. 12.0; Data Source = F:\productsDB. accdb";
}
```

2. Goto "Demo.aspx" and write the following code for button click event.

```
String strCon = Application ["dbCon"]. ToString();
OledbConnection con = new OledbConnection (strCon);
con. Open ();
```

★ Method - 2

Ex: writing connection ~~is~~ string in web.config app settings.

AppSettings are used to store Application Variables that can be changed dynamically during the runtime

Ex 1) Goto web.config file and add the following -

```
<app Settings>
 <add key = "dbCon".
  Value = "provider = Microsoft. ace. Oledb. 12.0;
  DataSource = F:\\ EmployeeDb.accdb"/>
 <app Settings>
```

2. Goto Demo.aspx and write the following code:

```
// Import Namespace
  Using System.Configuration;
  using System.Data.Oledb;

// Button click_code .

 string strcon = ConfigurationManager.AppSettings
   ["dbcon"].ToString();
 OledbConnection con = new Ole~~B~~dbConnection (StrCon);
   con.Open();
```

Note :
Add the add key under Connection string and write the new connection String in web.config form .

Note: what is benefit of defining connections ~~in~~ in global.aspx or ~~A~~ web.config page?
Why it is not recommended to write the changes in Page_Load event.?

Ex2: Connection Strings in web.config.

1. Add the following code in web.config

```
< Connection Strings>
<add name'= "'productsConnection"
providerName= "System.Data.Oledb"
ConnectionString = "provider= Microsoft.ace Oledb.12.0;
Datasource = F:\\ ProductsDB.accdb" />
</ connectionString>
```

2. Demo.aspx.

```
// Button Click - code.
String strclon = Configuration Manager. Connection strings
[" products Connection "].To string();
~~Oledb~~ OleDbConnection   con = new OleDbconnection (strCon);
con. Open();
```

★ How do maintain connection string globally to access it from any Application in computer. -
- we can do it ~~we~~ using. "ODBC".

→ Namespace changed to OBBC for Oledb.

→ while doing connection put DSN = DataSourceName. given by you

→ If we want access it from any other application then .

* ODBC connections :

An open database Connectivity allows to maintain the connection string globally and locally on server so that it is accessible from any Application.

1) Open control panel → goto administrative tools ~~an~~.
2) Open "ODBC datasources".
3) goto "System.DSN" property and click Add
4) Select ~~microsoft~~ ~~provider for~~ microsoft access provider (.mdb, .accdb).
5) Type the data Source Name DSN = products Connection
6) Type any Description
7) Click select Button and select the database file name. "F:\\ ProductsDb.accdb".
8) Click finished
9) ~~Goto~~ goto demo.aspx and write following code

```
// Import the namespace
  Using System.Data.ODBC

// Button_Click code

OdbcConnection con = new Odbc connection ( ~~DATS~~ "DSN=
 products Connection");
  con.Open();
 Response.write(" Connected...");
```

21/07

About Command:

OleDbCommand:-

1) It represents an sql statement or stored procedure to execute against a data source.

2) Command Implicitly executes by using an adapter.

3) Command Requires the following methods to execute explicitly.

4) Methods are-

1) ExecuteReader: Use when command returns more than one value.

Ex: OledBCommand cmd = new Oledbcommand ("select * from tblProduct", con);

Executing: cmd. ExecuteReader();

2) Execute Scalar: Used when command Returns only one value.

Ex: OledbCommand cmd = new OledbCommand ("select count ("*") from tblProduct");

To execute: cmd. ExecuteScalar()

3) ExecuteNonQuery: Used when command is affecting the database.

Ex OledbCommand cmd = new Oledbcommand ("Delete from tbl Product where ProductId =1");

To execute: cmd. ExecuteNonQuery();

☆ The commands are derived from the base "dbCommand"
The command Text or String can be anyone of the follow-
-ing types -
a) Text
2) Stored Procedure.
3) Table Direct

☆ OledbDataReader() :-
1) The data reader provides a forward - only ~~String~~ stream of data rows from data source.

2) It is readonly and will not support Manipulations.
3) It can read only once from the datasource for connection i.e it is just like a Constructor.
4) It provides the following properties and methods.
a) GetName : Returns the fieldName at specified Index
b) GetFieldType : Returns the datatype of field at specified index
c) FieldCount : ~~Counts~~ ~~howman~~ returns the total count of fields present in a table.
d) Read : Returns true if there are no more records to read from a table

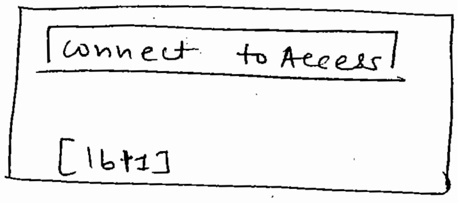
Ex: Using DataReader And Command:
1) Write the connection string in "web.config"

```
<connectionStrings>
<add Name = "productConnection" providerName = "System.
Data.Oledb"

ConnectionString = "provider = Microsoft.ace.oledb.12.0;
```

Data Source = R:\\ · productsDB. accdb ";" />
</connectionStrings>

2) Add a new, webform "Demo. aspx" and "Demo.aspx" Design



3) In Demo. aspx , Button_Click code :-

```
// Import namespace
 Using . System . Configuration;
 Using . System . Data . Oledb;

// Button Click - code .

   String strCon = ~~Oled~~ ConfigurationManager . ConnectionStrings
 [" ProductsConnection" ] . ToString ();

 OledbConnection con = new OledbConnection ( strCon);
  con.Open();

 OleDbCommand cmd = new OledbCommand ( "select *
  from tbl product " , con);

~~OledDbCo~~
  OledbDataReader dr ;
  ~~dr : command~~ . dr. cmd . ExecuteReader () ;
   while ( dr. Read ())
  {
 Response . Write ( " Product ID = " + dr [ " ProductID " ] + "." "
 + " product Name: " + dr [" productName"] + " - " + "price:"
 + dr. ["price"] + " <br> ");
 }
```

```
Label1.Text = "Total No. of Fields :" + dr.FieldCount +
"<br>" + "Field at Index No1 :" + dr. GetName(1) + "<br>"
 + Data Type of price = " + dr. GetFieldType(2);
  con.Close();
```

★ OLEDB Data Adapter:

1) It represents a set of data commands and a database connection that are used to fill the dataset or a table and update the data ~~store~~ source.

2) It opens the connection and executes the command implicitly.

3) Adapter provides the following methods:-

   a) Fill():- Fills the data into a table or a dataset

   b) Update(): Updates the table or dataset.

★ <u>DataTable:</u>

1) It represents a record set of ADO-

2) It is transported across tiers by using Marshal by value component.

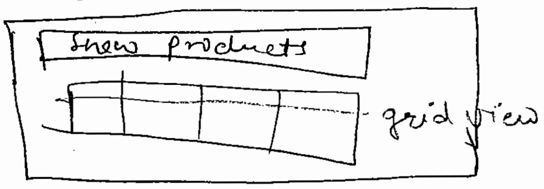
3) It doesn't support Relationships.

4) It is connected in access.

5) It provides the set of properties and methods that are defined under "System.Data".

★ <u>EX</u>: 1. Add a new webform "Demo.Aspx"

2. Design



3. Demo.aspx.cs (code)

```
1. // Import the namespace
   Using system.Configuration;
   Using System.Data.Oledb;
   Using System.Data

// Button-Click code
   string strCon= ConfigurationManager.ConnectionStrings["pro-
   -ductsConnection"].ToString();
   OleDbConnection con = new OleDbConnection (strCon);
   OledbCommand cmd = new OledbCommand( "select *from
   tblProduct", con);
   OledbDataAdapter dt= new OledbDataAdapter (cmd);
   DataTable dt = new DataTable();
   da.Fill (dt);
   GridView1. DataSource =dt;
   GridView1. DataBind();
   Response.write(" Connection Status = " + con.State);
```

★ Dataset: 1) Dataset represents in memory database.

2) It is a collection of tables, constraints and relations

3) It is disconnected in access.

4) It is bidirectional in navigation and manipulations
read write

5) It supports manipulations.

6) It is fully XML featured

7) Doesn't require any conversions.

8) It comprises of set of properties and methods that are

defined under "system. Data".

9) The tables in dataset are accessible by IndexNumber or the reference name

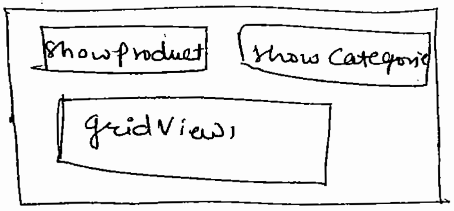
reference name - given by us, no need to match with ProductName

10) Ex 1. Create Databases in MS Access.

- Tbl Product
- tbl Categories

2. Add a new webform by Name "Demo Aspx".

3. (Design)



4. Code Demo.aspx.cs.

1. //Import Namespaces

Using. System Configuration;
using System. Data · Oledb;
Using System Data;

2. // Button 1 _ Click code: (Show Products)

Publicly declare ↓

Oledb Connection con;
string strcon = Configuration Manager = Connection Strings ["
ProductsConnection"].To string();
OledbCommand cmd;
Dataset ds = ~~new~~ new Dataset();

Now, Button1_Click code

```
protected void Button1_click( Object sender, eventArgs e)
 {
  con = new OledbConnection (strCon);
 cmd = new Oled Command ("select * from tbl Product ",
   con);
OledBData Adapter ProductsAdapter = new OledbDataAdapter
                                          ~~(new)~~ (cmd);
Products Adapter . Fill ( ds, " ProductsTable ") ;
GridView1 . DataSource = ds. Tables [ " productsTable "] ;
 GridView1. DataBind ();
}
```

// Button2_ Click code :

```
Protected void Button2_click ( object sender, eventArgs e)
 {
 con = new OledbConnection (strCon) ;
Cmd = new Oledb Command ( "select * from tabl Categories ",
    con) ;
OledbDataAdapter P CategoriesAdapter = new OledbDataAda-
       -pter ( cmd);
P CategoriesAdapter . Fill (ds, " CategoriesTable") ;
 GridView1 . Data Source = ds. Tables [ "CategoriesTable"] ;
  GridView1. € DataBind ();
 }
```

★ How to delete the table from database if it is ~~used~~ currently used by one client and another client tries to delete ?

Syntax: Alter database Aspdb set
~~RollBack~~ ~~Set~~ Single_User Rollback Immediate

★ Sql Server Data Binding

02/07 ★ Sql Server Data provider: The sql provider is responsible for communicating with sql ~~databe~~ server database. It provides a set of classes defined under "System.Data.SqlClient". The classes are -

1. SqlConnection
2. Sql Command
3. Sql DataReader
4. SqlData Adapter

1> Sql Connection:

Syntax: Sqlconnection con = new SqlConnection("ConnectionString");

Connection String = "Data source = ServerName, Initial Catlog = Database Name, Integrated Security = SSPI, user id = UserName, password = password".

Ex. :

1) Create a new Database in sqlserver by Name "AspDb".

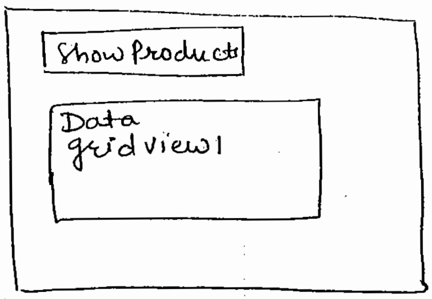
2) Create a new table by name "tbl products".

	Field	Datatype
1)	ProductId (PK)	int (isIdentity = true)
2)	~~IsIdentity~~ Name	~~true~~ Varchar (50)
3)	price	Money

3. Add records into the table.
4. Create a new Asp.Net Application.
5. goto web.config and write the connection string

```
<connectionStrings>
<add name = "ProductsConnection" ProviderName = " ~~Microsoft~~
  System.Data.SqlClient"
"Connection string = " Data Source = .; initial Catlog= AspDb;
Integrated ~~secu~~ security = SSPI , user id= userid, password=
  123"./>
</connection strings>
```

6. Add a new webform "Home.aspx".
7. Home.aspx (design)



// show_product Button click code:-

1) // Import Namespaces

```
Using system.Configurations;
Using system.Data;
using system.Data.SqlClient;
```

// showButton_Click code.

```
string strCon = ConfigurationManager.ConnectionStrings["
  ProductsConnection"].ToString();
SqlConnection con = new SqlConnection (strcon);
```

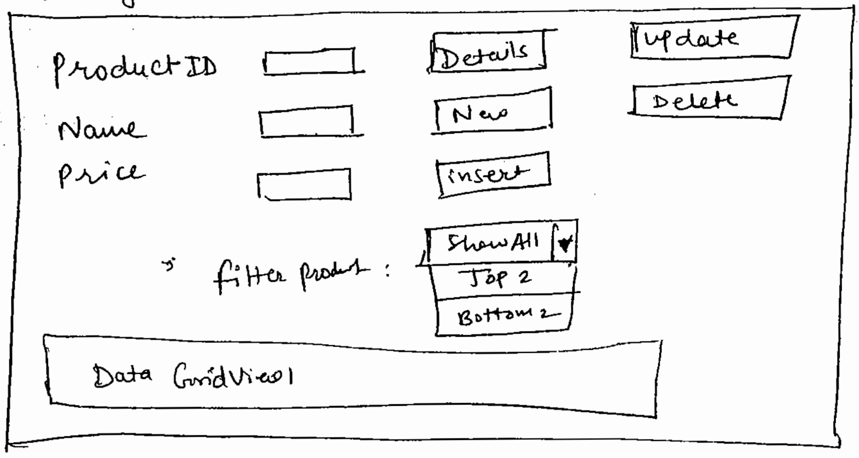
```
SqlCommand cmd = new SqlCommand ("select * from
tblproducts", con);
SqlDataAdapter da = new SqlDataAdapter (cmd);
Dataset ds = new Dataset();
da.Fill (ds, "ProdTable");
GridView1.DataSource = ds.Tables["ProdTable"];
GridView1.DataBind();
}
```

★ SQL DataBase Binding Operations using layer based architecture (CRUD operations)

1. Create Database by name "AspDb".
2. Create table by name "tblProducts":
3. Create a stored procedure to get product list.

```
~~Gre~~
Create procedure spGetProduce
 As
Select ProductID, Name, price from tblProducts
Go.
```

4. Create a new Asp.net Application by name "SQL CRUD".
5. goto web.config and write the connection string
6. ~~Goto~~ ~~file~~ Add a new webform by name "Details.aspx"
7. Design



8. Goto File Menu → Add new project

9. Select ~~a~~ Class library Project and named it as "DataAccess Layer".

10. goto References in "Data Access Layer" → Add reference → "system. configuration".

11. Add a new class in "Data Access Layer" by name "product.cs".

```
public class Product
{
~~product~~ Public int ProductID {get; set;}
public string Name {get; set;}
public double ? price {get; set;}
}
```

12. Add another class into "DataAccessLayer" by name ~~"~~ "productsCRUD.cs" and write the following code there.

```
// Import Namespaces.
using System.Configuration;
using system.Data.SqlClient;
using System.Data;
~~using system./~~
namespace DataAccessLayer
{
public class ProductsCRUD
{
string strCon = ConfigurationManager.ConnectionStrings["
  ProductsConnection"].ToString();
Sql Connection con;
Sql Command cmd;
```

```
   // Read Operation
public IEnumerable <product> products
   {
   get
   {
   List <product> products = new List <product> ();

    con = new sql Connection (strCon);
    cmd = new Sql Command ("SpGetProcedure", con);
    SqlDataReader dr;
     dr = cmd. ExecuteReader ();
   while (dr.Read())
   {
   Product Product = new Product();
   product. ProductID = Convert. ToInt16 (dr ["ProductID"]);
   Product. Name = ~~T~~ dr.["Name"] . ToString();
   Product . Price = Convert . ToDouble (dr[ "price"]);
    Products . Add (product);
    }
    return products;
   }
  }
 }
}
```

13. Goto "SQL CRUD" ~~sql P.~~ Asp Project and add reference for "Data AccessLayer".

   - Right Click on references
   - Add reference
   - goto solution Category
   - Select Data Access Layer

14. Details .aspx.cs (code)

```
// Import Namespace     ProductCRUD db = new ProductsCRUD();
   using DataAccessLayer;

// Details ButtonClick code

 int id = int.Parse(txtProductID.text);
  foreach ( var item in db.Products)
  {
   if (item.ProductID==id)
   {
    txtName.text = item.Name;
     txtPrice.text = item.Price.ToString();
   }
  }

// DropDown list Selected Index Changed Code.

  Switch (DropdownList1.SelectedIndex)
  {
   case 0:   GridView1.DataSource = db.Products;
             GridView1.DataBind();
             break;
  Case 1 :   List <product> prods = db.Products.OrderByDescen-
      -ding (x => x.Price).Take (2).ToList();
             GridView1. DataSource = prods;
              GridView1. DataBind();
              break;

   case 2:   List <product> prod = db.Products.OrderBy(x =>
          x.price).Take (2).ToList();
             GridView1.Datasource = prod;
```

```
        GridView1.DataBind();
        ~~B~~ break;
    }
```

03/07

Create Operation:

1. Create a stored procedure to insert records

```
Create procedure spAddProducts
(
@Name Varchar(50),
@price ~~Varchar 50)~~ money.
)
 As
 insert into tblproducts ( Name , price) values (@Name,
          @Price)
 Go
```

2. ~~Gto~~ goto DataAccessLayer and add the following method in "ProductsCRUD.cs".

```
 public void AddProduct (product product)
 {
 con = new SqlConnection (strCon);
 con.Open();
 Cmd = new ~~Go~~ SqlCommand ("spAddProducts", con);
Cmd.CommandType = CommandType.StoredProcedure;
  SqlParameter paramName = new SqlParameter();
   ParamName.ParameterName = "@Name";
    paramName.Value = product.Name ;
    Cmd.Parameters.Add (ParamName);

   Sql Parameter paramPrice = new SqlParameter();
    ParamPrice.Parameter~~Price~~Name = "@price";
paramPrice. value = Product. Price ;
 Cmd. Parameters. Add (paramprice);
  Cmd. ExecuteNonQuery ();
  con. close();
 }
```

3. Goto "Details.aspx"

// "New" Button Click code.

```
   txtProductID. text = " ";
   txtName. Text = " ";
   txtPrice. Text = " ";
   txtProductID. enabled = false ;
   txtName. focus();
```

// Insert Button Click code.

Note: Create a method "GetList()" to display the records in GridView.

```
  private void GetList()
  {
   GridView1. DataSource = db. products;
   GridView1. DataBind();
  }
```

// Insert Button Click code.

```
 product product = new Product();
 Product. Name = txtName. Text;
 product. Price = Convert. ToDouble (txtprice. Text);
     db. Add Product (product);
     GetList();
     txtName. Text = " ";
```

```
    txtprice.Text = " ";
    txtProductID.Text = " ";
    txtProductID.Enabled = true;
    txtProductID.Focus();
```

★ Update operation :-

1) Create a stored procedure to update records

```
    Create procedure spUpdateProducts
  (
  @productId int,
  @Name varchar(50),
  @Price money
  )
  AS
  Update tblProduct SET
  Name=@Name, Price= @price where ProductID=@ProductID
  go
```

2. goto DataAccessLayer and add the following method in "productsCRUD".cs'.

```
                                  Class    obj
                                   ↓        ↓
  public void UpdateProduct( product product)
   {
   con = new SqlConnection (strcon);
   Con.Open();
   Cmd = new SqlCommand ( "spUpdateProduct", con);
   Cmd.CommandType = CommandType.StoredProcedure;
   Cmd.Parameters.AddWithValue ( "@productID",
                     Product.ProductID);
   Cmd.Parameters.AddWithValue ("@price," Product.Price);
   Cmd.ExecuteNonQuery();
   Con.Close();
   }
```

3. Goto Details.aspx.

Update Button Click code

```
product product = new Product();
product.ProductID = int.Parse (txtProductID.Text);
Product.Name = txt~~Product~~Name.Text;
Product.Price = ~~txtPrice.Text;~~
        Convert.ToDouble(txtPrice.Text);

db.UpdateProduct (product);
GetList();
```

☆ Delete Operation:-

1. Create a stored procedure to Delete Product

```
Create Procedure spDeleteProduct
(
@productID int
)
AS
Delete from tblProducts where productID =@ProductID
Go
```

2. Goto Data Access Layer and add the following method in "productsCRUD.cs".

```
public void DeleteProduct(int Id)
 {
con = new SqlConnection (strcon);
con.Open();
cmd = new SqlCommand ("spDeleteProduct", con);
cmd.CommandType = CommandType.StoredProcedure;
cmd. Parameters. AddWithValue ("@ ProductID ", Id);
 cmd. ExecuteNonQuery ();
  con. Close();
 }
```

3) Details. aspx.

```
// Delete Button Click code

  int id = int. Parse (txtProductID. text) ;
     db.Delete Product (id);
      GetList();
     txtproductID. text = " ";
     txtName. Text = " ";
     txtPrice. Text = " ";
      txt Product ID. Focus();
  }
```

★ Data Sources :-

The data source controls provides set of properties & meth-ods to bind and communicate with various datasources implicitly. Asp.net provides the following datasources.

1. Sql Datasource
2. Entity Datasource
3. Linq Datasource
4. Object Datasource
5. Sitemap Datasource
6. XML Data source

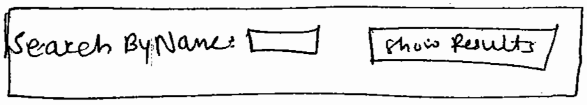
★ SqlData Source:- It provides a set of properties and meth-ods to communicate with the databases like sqlServer, oracle and oledb.

(4)

Ex: ① Add ~~the~~ following pages to website.

- search.aspx
- Results.aspx

② Search.aspx (Design)

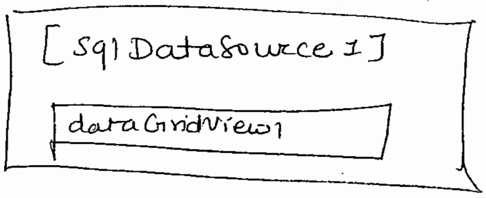


③ Search.aspx.cs (Code)

// ~~Search~~ Button click code:

```
Session ["product"] = TextBox1.Text;
Response.Redirect ("Results.aspx");
```

4. Results.aspx (Design)



5. Goto SqlDataSource1 smarttag and select "Configure Data Source".

6. A configuration wizard starts with following steps:

Step1 :- Click new connection
- Select Data Source sql server
- Specify server Name as "(local)"
- Select Authentication (windows / server ✓)
- Specify Userid & password
- Select Database Name "AspDb".
- Click Next

Step 2 :- Select TableName "tblProducts".
- Select fields
- Click "Where" Button
- Select "Column, Operator, Source & parameter"

Column Name
Operator = LIKE
Source = Session
SessionValue = Product.

- Click Add
- Click Next

Note : Advance - for insert, Update & Delete.

Step3 :- Click "Text Query"
- Click "Finish"

7. goto GridView1 properties and select Datasource = "sqlDatasource1"

★ Data Bound Controls :

The Data Bound controls Enables the UI to interact with Database like performing ~~operas~~ CRUD operations. Asp.net provides the following Data Bound Controls :-

① Data List
② Details View
③ form View
④ List View
⑤ Repeater

① DataList : It enables the UI to read the data from Datasource and Display in page so that, it cannot be modified p

Properties :

1) Id
2) format
3) Datasource - SqlDatasource
4). Repeat Columns
5) Repeat Direction
6) Show Grid - Hor, Ver

② Details View : It is the Data Bound Control that enables the user to perform all the CRUD operations.

Properties:

1) Id
2) Runat
3) DataSource
4) HeaderText
5) FooterText
6) Auto Generate ~~link~~ edit Button
7) Auto Generate Delete Button
8) Auto Generate InsertButton
9) AutoFormat
10) Enable paging

★ FormView : The form View control enables the UI to perform any specific operation like Edit, Read or Insert

★ Properties.

1) Id
2) Runat
3) DataSource
4) HeaderText
5) FooterText
6) Default Mode : Insert, ~~Delete~~, Read., Edit

★ ListView :

It is the Data Bound control that enables the UI to perform all CRUD operations and display the data in tabular form

★ properties :

1) ~~Go~~ ID
2) Runat
3) DataSource
4) Configure~~Vi~~ ListView

5. Enable Insert
6. Enable Edit
7. Delete
8. Paging
9. InsertItem Position : FirstItem , Last
10. DisplayLayout : Grid, SingleLine ....

04/07

Repeater control :-

A repeater control is a data bound control that enables the UI to customize the Appearance by using following templates

1) ~~The~~ Header Template
2) Footer template
3) Item template

Properties:

1) ID
2) Runat
3) DataSourceID

Note : The data bound controls can bind database fields using the following methods -

a) Bind
b) Eval

The Bind method binds database field with any specified control. whereas Eval is used to show the database field as a literal

Ex. 1) Goto the products table and add the following field

- field	Datatype
Photo	varchar

2) Store the photo information as virtual path
- ~/photo/mobile.jpg

3. Add a new webform by name "products.aspx"
4. Add following controls to page on - SqlDataSource1
= Repeater.
5. Configure SqlDataSource with products table.
6. Set DataSource for repeater as "sqlDataSource1".
7. goto html source of repeater
8. Add the following code.

```
<asp: repeater ID = " Repeater1" runat = Server.
DataSourceID = " SqlDataSource1" >
<HeaderTemplate> <h1 align = "centre" >< products Info
   </h1>

 </HeaderTemplate> <ItemTemplate> < table border=
   "1" width = "200" >

<tr> <td> <asp: Image ID = "img1". runat = "server"
ImageUrl = '<%# Bind ("photo") %>' Width = "100".
Height = "100" /></td>
<td>
 <table broder = "1" width = "200" >
 <tr> <td> <%# Eval ( "productID") %></td>
     </tr>
<tr> <td> <%# Eval ("Name")%></td></tr>
<tr> <td> <%# Eval ("Price") %></td></tr>
   </table>
   </td>
   </tr>
   </table>
  </ItemTemplate>
```

```
<Footer Template>
<h3 align = "Centre" > & copy .copyright 2015 </h3>
<~~¢~~/ Footer Template>
</asp: Repeater>
```

★ Customizing Grid View Control:

1. Hiding Any specific field inside the grid View
   goto the source of field and set visible="false".
   1) goto html source of grid view.

```
<asp: BoundField DataField = "Name" HeaderText = "Name"
visible = "false" / >
```

2. ~~2)~~ you can also Delete the bound Field from grid View Source.

3. Changing the header text for any field in grid View

Syntax:

```
<asp: Bound Field DataField = "Name" Header Text= "Product
                                                  Name" />
```

4. Applying DataFormats to grid View Data Fields.
   (Date, Currency, Time)

Syntax: for currency

```
<asp: BoundField DataField = "Price" HeaderText= "ProductPrice"
DataFormat string = "{0: C}" / >
```

0:d - Short Date Format
0:D - long Date
0: c - Currency
0:t → Short Date
0:T → long Date

⑤ Adding controls to gridView.
(Image in GridView)

a) Goto Grid view properties and select ~~edit~~ "edit columns."

b) In selected columns category select the field "photo".

c) Click on the link ~~convert field~~ → "Convert To Template Field".

d) Goto Html source of grid view you will find "Template Field" for photo

e) Change the item template as shown below.

Syntax:

```
<asp: TemplateField  HeaderText = "photo">
<ItemTemplate> <asp: ~~Temp~~ ImageID = "img1" runat =
"server"  ImageUrl = '<%# Bind ("photo") %>'
width = '100'  Height = '100' />
</ItemTemplate>
</ asp: TemplateField>
```

~~Note for PHP~~

5) Customizing grid view by adding checkbox into column

→ goto html source of grid view and add following template in grid view.

// ~~inside~~ gridview ID under.

```
< columns>
< asp: Template Field >  <ItemTemplate > <asp:
  CheckBox  ID = "C1"  runat= "server" />
 </ ItemTemplate>
 </asp. Template Field>

</ columns>
```

6. Adding insert operation to the grid view by customizing to the footer template.

a) Goto gridView properties and select Edit-columns.

b) Convert all field into template fields.

c) Add a footer template for every field with the controls you want to display

```
<asp: TemplateField   HeaderText= "ProductID " >
<Footer Template >
<asp: ButtonID = "btnInsert" runat= "server" Text="Insert"
 Onclick = "btnInsert_Click" />
</footer Template>
</asp: TemplateField>
```

7 ★ Adding Validation Inside the GridView ~~&~~ BoundFields.

a) goto grid view properties and select "edit columns"

b) convert the price field to template field.

c) goto html source of gridView and add a validation control for price

d)
```
<asp: TemplateField  HeaderText= "price" >
 ~~<EditTemplate~~ <EditItem Template >
  <asp: TextBox ID = "TextBox1"   runat = server" />
 <asp: RequiredFieldValidator ID = "rfV1" runat =
 "server"   ControlToValidate = "TextBox1"
 Error Message= "Price Requred"  ForeColor = "Red" />
  </ EditItem Field >
```

AJAX | WCF | XML | Authentications

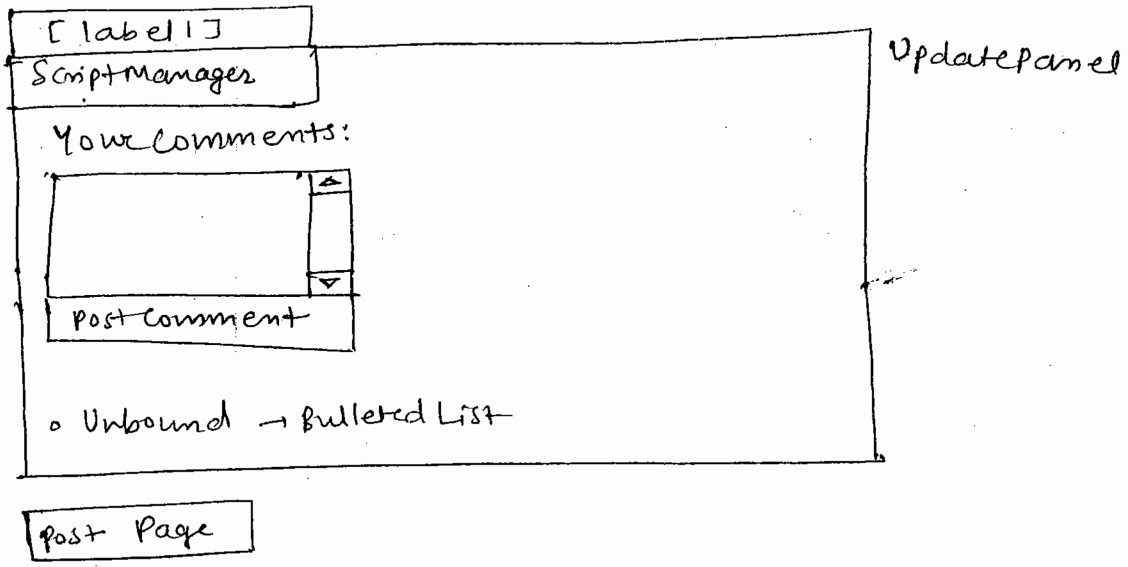
A AJAX (Asynchronous Javascript & XML)

1. AJAX is used web Development to make the applications more interactive and responsive.

2. It can be used to maintain partial PostBacks so that only a portion of the page is posted to server.

3. Asp.net provides several AJAX extensions, which includes
   - a) Update Panel
   - b) Script Manager
   - c) Timer ...etc—

4. Ex:
   1) Add a new webform by Name "Comments.aspx".



2) Source code

// Page Load Event code:

Label1.Text = "page posted on:" + DateTime.Now.ToString();

// Post Comment Button Click code

BulletedList1.Items.Add (TextBox1.text + " - posted on :" + DateTime.Now.ToString());

05/07 Default user for ASP.NET is ASPNET.

★ Security in

ASP.NET provides several security features that allows the application to restrict the accessibility. The following are the security features in ASP.NET.

1. Impersonation
2. Identity
3. (OpenId)
3. Member Ship
4. Autherization and authentication

★ Impersonation:-

Every website is process under the request of the default user (ASPNET) Changing the default user account is known as "impersonation".

This type of security is Applicable for only intranet based websites. This requires a setting in "web.Config" file

~~<System.web>~~

1. Add a new folder by name "Manager".
2. Add a webform Home.aspx into the folder
3. Add new item into manager folder and select the item type as "web.config"

```
<System.web>

<identity impersonate = "true".
UserName = "manager"
```

```
    password ="1234"/>
  </System.web>
```

☆ Open Id is 4.5

It is the new feature introduced from ASP.net 4.5 It allows third party logins that is you can use google, facebook and twitter accounts to access website

1. Create a new ASP.net Application.
2. goto "App-Start" folder
3. goto the file "StartUp-Auth.cs"
4. Uncomment the following line

   app.UseGoogleAuthentication();

5. Now press "f5" to run Application
6. Click "Login" from Default page
7. Google option will be available at "other service Logins"

☆ Autherization and Authentication:

Authentication is the process of checking user credentials like (UserName, Password), Security token, etc).

Autherization is the process of allowing access to the resources of a website. ASP.net provides three types of ~~all~~ authentications.

1) window Authentication
2) form Authentication
3) Passport Authentication

① Windows Authentication:

It is a process of giving accessibility to the resources of a website by using windows credentials. This is applicable for intranet based websites and this requires the following configuration in web.config file.

```
<System.web>
<authentication mode = "Windows" >
</authentication>
<allow Users = "Manager Rahul" />
<deny users = "?"/>
</autherization>
</ system.web>
```

* = deny All users.
? = Deny Users without security token.

★ Form Based Authentication:

It is the process of authenticating the user by checking his credentials with a login form and Database. Users are given access to application only when their crendentials match with database. This type of authentication is Applicable to both intranet and internet based websites. This requires configuration in the web.config file

```
< System. web>
< Authentication mode = "Forms" >
<forms loginUrl = "~/Login.aspx" >
</ forms >
```

```
</authentication>
< authorization>
<deny Users ="?" /> → *
</authorization>
</system.web>
```

Ex: 1) Create a new table in sql Database by name "tblUsers".

~~Uerid~~ Userid (PK)
UserName
Password } varchar(50)
Mobile

2) Create a stored procedure "sp Register"

3. Add the following pages to your website

- Home. aspx
- Register. aspx
- Login. aspx
- Error. Aspx
- Success. aspx
- Tutorial. aspx

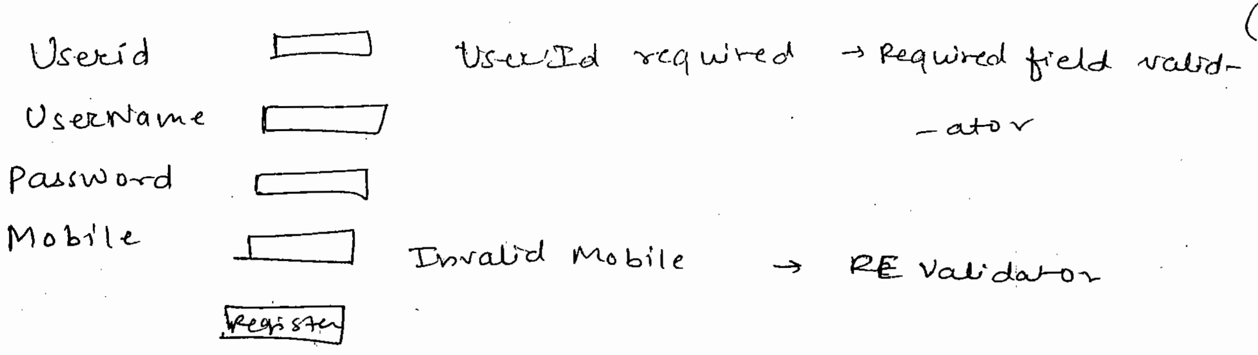
4. Home. aspx Design.

~~Register~~ ~~Us~~ NewUser <a href "

New User Register

Existing User Login

5. Register aspx Design


Userid
UserName
Password
Mobile
Register
UserId required → Required field valid-ator
Invalid Mobile → RE Validator

6. Register.aspx.cs code

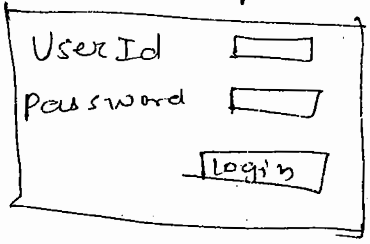
```
// Import Namespace
using System.Data.SqlClient;
using System.Data;
```

7. Register Button click code.

```
if (Page.IsValid)
 {
 SqlConnection con = new SqlConnection ("Data Source=
 .; initial catlog = AspDb; Integrated Security = SSPI;
 Userid = sa; password = 123 ");
 con.Open();
 SqlCommand cmd = new SqlCommand ( " ~~SpSto~~ spRegister",
                                                   con);
 cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue ( "@UserId," txtUserID.Text);
cmd.Parameters.AddWithValue ("@ ~~Pass~~ UserName", txtUserName.
                                                      text);
Cmd.Parameters.AddWithValue ("@Password", txtpassword.text);
Cmd.parameters.AddwithValue ("@Mobile", txtMobile.text);
   cmd.ExecuteNonQuery();
   con.Close();
```

Server.Transfer ("~~Login~~ Success.aspx");
}

7. Login.aspx (Design)



8. Login.aspx.cs (code)

~~SqlConnection con = new SqlConnection (~~

```
// Import Namespace
   using System.Data;
   Using System.Data.SqlClient;

// Login Button Click code

SqlConnection con = new SqlConnection ("____");
con.Open();

~~Col~~ ~~Cm~~
SqlCommand cmd = new SqlCommand (" Select
UserId, pass~~p~~word from tblUsers where UserID =
@UserId and password = @password"; con);

cmd.parameters.AddWithValue ( "@UserId", txtUserID.
                                          text);

cmd.Parameters.AddWithValue ("@Password", txtPassword.
                                          text);

DataTable dt = new DataTable();

SqlDataAdapter da = new SqlDataAdapter (cmd);

da.Fill (dt);

if (dt.rows ~~[illegible]~~.count > 0)
```

```
{
Server.Transfer("~/NareshIT website/Tutorial.aspx");
}
else
{
Server.Transfer("~/NareshIT Website/Error.aspx");
}
con.Close();
```

9) Success.aspx.

```
<h1> Register succesfully.... </h1>
<a href = "Login.aspx"> Login </a>
```

8. Error.Aspx

```
<h1> Invalid UserId/ password </h1>

<a href = "Login.aspx"> try again </a>
```

★ Membership: It is the process of providing autherisation and authentication to a website without manually approaching towards a database. Asp.net provides several login controls for this ~~authenti~~ membership. They are-

- Change password
- Create User Wizard
- Login
- LoginName
- Login Status
- Login View
- password Recovery

Ex : 1. Goto file new project. Select Visual C#, web
Visual Studio 2012.

2 Select the template Asp.net ~~web~~forms Application

3. Add following folder to website.

- public
- secured

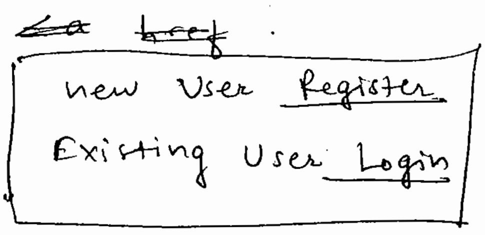
4. Add following files to public folder

- Home.aspx
- Login.aspx
- Register.aspx
- Change-Password.aspx
- Forgot Password.aspx
- Signout.aspx.

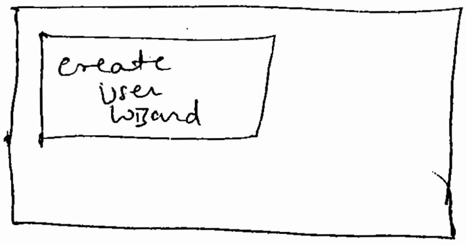
5. Add following pages to secured folder

- welcome.aspx
- Tutorial.aspx.

6 Home.aspx (Design)



7. ~~Home.aspx~~ Register.aspx Design



- Add create User Wizard ctrl
- set following properties

~~Contro~~

1. ContinueDestinationPageUrl = Login.aspx.
2. finishDestinationPageUrl = Success.aspx
3. CreateUserButtonText = Register.

8. Login.aspx (Design)

- Add Login ctrl
- Set Properties:

1. DestinationPageUrl = ~/Secured/ welcome.aspx
2. PasswordRecoveryText = forgot password.
3. PasswordRecovery Url = forgot password.aspx

9. Change Password.aspx (Design)

- Add Change Password ctrl
- Set properties:

1. SuccessPageUrl = Login.aspx

10. forgot password.aspx (Design)

- Add ~~Pass~~ forgot Password ctrl

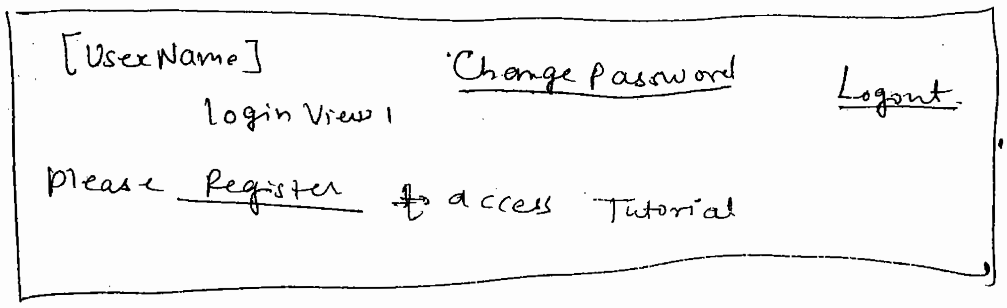
1. Success PageUrl = Login.aspx

11. Signout.aspx (Design)

<h1> Signed out successfully.... </h1>

<a href = "Login.aspx" > Login </a>

12. Welcome.aspx Design



· Properties :

Login Status :-
- Select "Logged · In"
- Action "Redirect"
- URL = SignOut.aspx

LoginView – Two templates .

```
~~<tor~~ <asp : LoginView1 ID = "LoginView1" . runat = "server"
<Anonymous Template >
 <h1> Please
 <a href = "…/public /Register.aspx">
Register </a> to access tutorial </h1>
</Anonymous Template>
~~</asp. LoginView~~ .
< LoggedIn Template >
  <h1> Goto to Tutorial <a href = "Tutorial.aspx">
  Click Here . </a> </h1>

 </ LoggedIn Template >
</asp : LoginView >
```

☆ XML

1. Offline Storage
2. Cross Platform

☆ XML:

XML is used in applications to achieve
1) Offline storage
2) Cross Platform

It can store the data offline so that the data can be acc-essed without the communication with database server and it is crossed platform, which is understandable to multiple browsers and clients

XML documents are classified into two types -

1) General XML Document : which contains only Data

Ex: Employee.xml

2) Structured XML Document : which contains both Data and structure

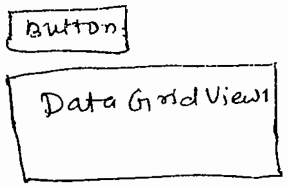
Ex : Dataset.xsd.

☆ Serialization :

The process of writing data into XML is known as Serialization. It can be achieved by using the method "writeXML()".

Ex: 1. Create a new webform "Home.aspx".

2. Home.aspx (Design)



3. Button_Click (code)

SqlConnection con = new SqlConnection ("Data Source = . ; initial Catlog = AspDb ; Integrated Security = SSPI ; User id = sa ; password = 123").

```
con.Open();
SqlCommand cmd = new {SqlCommand ( select *
 from tblProducts", con);
SqlDataAdapter da = new SqlDataAdapter (cmd);
Dataset ds = new Dataset ();
da.fill(ds, "prodTable");
ds.WriteXml (@"D:\products.xml");
GridView1.Datasource = ds.Tables[0];
 GridView1.DataBind();
```

★ **Deserialization:**

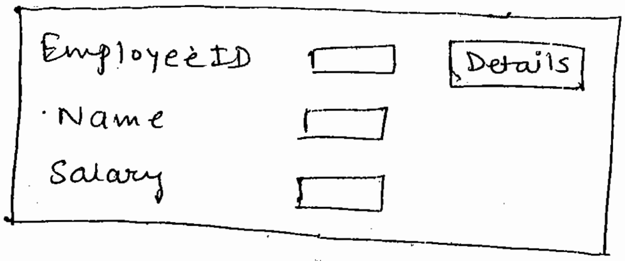
It is a process of reading data from an XML file. C# provides several classes for deserialization like
- XDocument
- X Element
- Read XML ()

Ex. 1. Add a new XML file into your website by name "~~Employee.xml~~ "Employees.xml". [[XML with Linq]]

```
<Employees>
<Employee>
<EmployeeID> 101 </EmployeeID>
 <Name> John </Name>
<Salary> 45000 </salary>
 </Employee>
 </Employees>
```

2. Add a new webform "Emp.aspx".

3. Emp.aspx (Design.



4. Details_Button Click (code)

```
// Import Namespaces
  using System.XML.Linq;
// Details Button Click Code
XElement xelement = XElement.Load (@ "E:\...\EmployeeXML");
IEnumerable < XElement > employees = xelement.Elements();
int id = int.Parse (TextBox1.Text);
Var res = from emp in employees
        where Convert.ToInt16 (emp.Element ("EmployeeID").
                                                Value) == id
    select emp;
    foreach ( var item in res)
    {
     TextBox2.Text = item.Element ("Name").Value;
    TextBox3.Text = item.Element (" Salary").Value;
    }
```

☆ Distributed Computing :