Unit-2

PAGE NO. :
DATE : / /

Programming in C# with Asp.Net :⇒

Introduction to C# Programming Language :⇒ the C# language pronounds 'C Sharp' or 'See Sharp' is an Completley object oriented Programming language (oops) developed by Microsoft Corporation to become a key Pair of .Net S/w development Plateform. the C# language is based on the ~~C++~~ C++ language. but it is Mostly developed on the lines of Microsoft visual basic

⇒ this Principle designer of C# language Anders Hejlesberg C# was designe to take advantage of the Common language Runtim (CLR) or application written in C# required CLR to run all the .Net Programming language have the .Net Framwork Class library the .Net class library also support function Such as file IO, database operation, XML & SOAP (Simple object Acces Protocol)

Example of C#

```
Public class Class1
{
    Static Void Main(string[] args)
    {
        System.Console.WriteLine("Hello world!");
    }
}
```

Characteristic of C# Programming :⇒

PAGE NO. :
DATE : / /

① Object oriented :⇒ C# Content Completley Object oriented features or golden Principle of object oriention, inheritance, Encapsulation, overloading, overridding, Polymorphism has Made C# Programming 'A' grade Choice of .Net application's developers also C# has Components of high level business object and wild range of Components to system level sku application's C# Contract their Components into XML web service which Permits them to be invoked ~~approach~~ access the internet from any language running on any O.S.

② Simple and Modern language :⇒

⇒ Pointer's are Missing in C#
⇒ unsafe operation such as direct memory Manipulation are Not allowed
⇒ in C# there is No usage of "::" or "→" Operator's.
⇒ C# inherit's the feature automatic Memory Management and the garbadge Collection.
⇒ versions of the Primitive data types like integer, float etc.
⇒ integer value of 0 and 1 are No longer accepted. as boolean value's are true or false values in C# C-sharp for No more error's of "=" or "==" Operator
⇒ "==" operator's is use for Comperision operation's
⇒ "=" operator is use for Assignment operations.
⇒ C# has a based according to the current trained and is very Powerful and simple for building Inter Opuble, Skilable, Robust application
⇒ C# include building support to turn any Component into a web service and Can be invoke over the internet from any application running on any Plateform.

PAGE NO.:
DATE: / /

③ typesafe :⇒

⇒ in c# we cannot Perform unsafe Caste like Convert double to a boolean

⇒ Value type (INT, Float, double) are initilize to 0 and refrence type (Object, class, array, string) are initilize to NULL by the Compiler automaticalley

⇒ array are 0 with indexing and are bound check

⇒ overflow of type can be check

④ Exception handling :⇒

⇒ .Net standrize the exception handling across language c# offeres's the Condition keyword too Control flow and Make the Code More Readible

⑤ ~~Scable~~ Scalable and Adaptable :⇒ .Net has introduce assemblies which are Salfed describing by means of their Manifest

⇒ Manifest Contain the assembley identity, version, Culture and digital Signature etc.

⇒ in our application ~~the~~ we delete the old files and updating them New ones without No registring of dll (dynamic link library)

⑥ interopesability :⇒ c# include Native Support for the COM and windows based application

⇒ c# allowed the usage to use Pointer's and unsafe code block to Manipulate old code

make

⇒ Component from Vb.Net and other Managed Code language direct the use in C#

(7) C# indexes :⇒ C# has indexes which helps to access value in a class with an array like Syntax

Data type :⇒ the following are two types in C#

① Value type
② Refrence type

① Value type :⇒

(a) integral type (Signed ~~whi~~ byte, unsigned byte, char, short, unsigned short, INT, unsigned INT or long).

(b) Floating and decimal type (float, double or decimal)

② Refrence type :⇒

Object type, class type, interfaces, delegets, string, array, struct

Boxing & unboxing :

Boxing :⇒ Convert Value type to refrence type

```
Int32 a = 10;
System.Console.WriteLine(Convert.ToString(a));
```

PAGE NO. :
DATE : / /

Unboxing :⇒ Convert reference type to value type

```
Stri String a :"20"
    System.Console.WriteLine (Convert.ToINT32(a));
```

Access Modifier :⇒

Private :⇒ access limited to Containing type or Can only read by Members of the same class

Public :⇒ it's Member Can be reached from anywhere

Protected :⇒ it's Member Can only be reached within the same class or from a Class which is inherits from this class

internal :⇒ it's Member Can be reached from within the same Project only.

Protected internal :⇒ it is same as internal accept the class which inherit's from this class Can reached it's Member, even from Another Project

Introduction to Asp.Net :⇒ Asp.Net is a technology used to create dynamic web application's or websites (internet based application) this is based on the .Net platform and it support Most .Net language for it's application.

Asp.Net Provides an infrastructure for faster and easier development of reliable and secure web application Asp.Net Provides Easy Programming Model flexible language option

and Compiled execution

features of Asp. Net :⇒

Asp. Net improve performance by using server side caching it allows to cache the entire output of a web page for use by other client

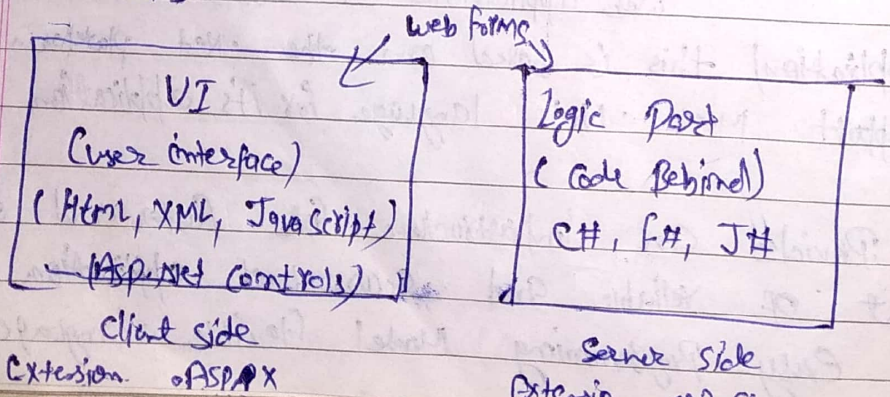
Asp. Net having functionality can be coded using different language like C# or Vb. Net However only one language can be used for coading in a single page.

Asp. Net is a container with many built in a server controls that have the common required functionalities.

Asp. Net is contain within Hierarchical namespace that we can organise coding structure manner.

Web Services
Master Pages
ADO. Net
Ajax Control toolkit
Model View Controller (MCV)

Web forms



PAGE NO. :
DATE : / /

Web forms in Asp. Net have user interface, control, all interfacing with the users through form Controls such as function, list box, text and soon these tools for creating web application or these Controls are called web forms.

the Asp. Net web Forms is divided into two parts visual Part or user interface. the visual Part or UI Part used to interact with the clients by the Controls the UI Page stored in a file with the extension of .Aspx

Logic or Code behind :⇒ the Code are written in a seperate file dynamically interaction with the UI Page 'Code behind file' this file extension is .cs or .aspx.cs

Web Forms inherits from System.web UI.Page namespace the body of the Page using the standard Html Forms tage
<form id = "form1" Root = "server> ... </form>

Asp. Net web Controls / web server Control Server :⇒ Asp. Net web Page Controls are object of an Asp. Net web Page that run when the Page requested and Server. Makeup as a browser Many web Server Controls or similar to Html elements such as function and text box etc other Asp. Net Control have Complex behaviours Such as Calender Control and Control other Controls that Manage data Connection Control datalist Control Called data Controls

PAGE NO.:
DATE: / /

the following are the categories of ASP.Net web Controls

Simple web services | List controls | Rich web Controls | Validation Controls.

- Label
- textbox
- Image
- RadioButton
- Checkbox
- HyperLink
- LinkButton
- ImageButton
- Button
- ~~Image~~B

- Drop Down list
- Radio Button list
- Check Box List
- Button List
- List Box

- Calender
- File uploading
- Ad Rotater

Validator
- Required field validation
- Range validation
- Compare's validation
- Regular Expression validation
- Custom validation
- Validation Summary

Data Controls
- grid view
- Data list
- Data grid
- Repeater
- Detail view

Simple web Control :=>

Label Control => this control is used to display server's of text or a message on clients browser for text

```
<asp: label Id = "LB1" runat = "server" Text = "Hello world" >
        <asp: label >
```

Output Hello world

Properties - text, FontPage, Border, Style

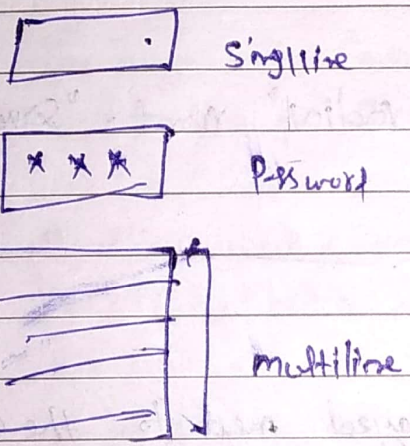
② TextBox :- textbox Control is same as Html text element this Control is used to input any text by the server on element

For

<asp: Textbox ID = "textbox1" Runat = "server" TextMode = "Single Line" MaxLength = "10" />

* Password
* Multilevel

Output



③ Image :-

<asp: Image ID = "Image1" Runat = "server" ImageUrl = "Image/logo.png" AlternateText = "Logo" Border = "0" />

④ HyperLink :-

<asp: HyperLink ID = "HyperLink1" Runat = "server" ~~PostBackUrl~~ NavigateUrl = "http://google.co.in" Text = "Click Here" />

PAGE NO. :
DATE : / /

Check Boxes and Radio Buttons

A check box displays a single option that the user can either check or uncheck and radio buttons present a group of options from which the user can select just one option.

To create a group of radio buttons, we specify the same name for the GroupName attribute of each radio button in the Group. If more then one group is required in a single form, the specify a different group Name for each group

```
<asp:CheckBox ID="ChkOption" runat="server">
</asp:CheckBox>

<asp:RadioButton ID="radio1" runat="Server">
</asp:RadioButton>
```

Property	description
Text	the text displayed onext to the check box or radio Button
checked	specifies whether it is selected or not, default is false
GroupName	Name of the group the control belongs to

Button Controls:-

Button :- it displays text within a rectangular area,
Link Button :- it displays text that looks like a hyperlink
ImageButton:- it display's an image

PAGE NO. :
DATE : / /

Button Syntax

<asp:Button ID = "Button1" runat = "server" onClick = "Button1_click"
Text = "click" />

Property	description (Common Properties of the Button Control)
Text	The text displayed on the Button. this is for button and link button Controls only.
ImageUrl	For image button Control Only. the image to be displayed for the button.
AlternateText	For image button Control only. the text to be displayed if the browser Can't display the image.
CausesValidation	Determines whether Page validation occurs when a user clicks the button. the default is true.
CommandName	A string value that is passed to the Command event when a user clicks the button.
CommandArgument	A string value that is passed to the Command event when a user clicks the button.
PostBackUrl	The Url of the Page that is requested when the user clicks the Button.

List Controls :-

Drop-down list
List - box
Radio Button list
check box list
Bulleted list

these Control let a user choose from one or more item from the list. list box and drop-down lists Contain one or more list items

Property of each list Controls

Property	Description
Text	The text displayed for the item.
Selected	indicates whether the item is selected.
Value	A string value associated with the item.

<asp: DropDownList ID="DropDownList1" Runat="Server"
AutoPostBack="True">

<asp: ListItem Value="0" Text="...Select any language"></asp: ListItem>
<asp: ListItem Value="1" Text="C++"></asp: ListItem>
<asp: ListItem Value="2" Text="C#"></asp: ListItem>
<asp: ListItem Value="3" Text="Java"></asp: ListItem>
<asp: ListItem Value="4" Text="F#"></asp: ListItem>
<asp: ListItem Value="5" Text="Vb.net"></asp: ListItem>
</asp: DropDownList>

<asp:ListBox ID="ListBox1" Runat="server" AutoPostBack="true" >

</asp:ListBox>

Properties of list box and drop-down list

Property	Description
Items	the collection of ListItem objects that represent the items in the control, this Property returns an object of type ListItemCollection
Rows	Specifies the Number of items displayed in the box. if actual list Contains more rows than displayed then a scroll bar is added
SelectedIndex	the index of the currently selected item. if more than one item is selected, then the index of the first selected item. if no item is selected, the value of this Property is -1.
SelectedValue	the value of the currently selected item. if more than one item is selected, then the value of the first selected item. if no item is selected, this value of this Property is an empty string "".
SelectionMode	indicated whether a listbox allows single selections or multiple selections.

A Radio Button list Presents a list of mutually exclusive options. A check box list Presents a list of independent options. these Controls Contain a Collection of List Item Objects that Could be refereed to through the item Property of the Control

<asp: CheckBoxList ID = "CheckBoxList1" Runat = "server"
AutoPostBack = "true">

</asp: CheckBoxList>

<asp: RadioButton ID = "RadioButton1" Runat = "server"
AutoPostBack = "true">

</asp: RadioButton>

Property	description
RepeatLayout	this attributes Specifies whether the table bags or the normal html flow to use while formatting the list when it is rendered, the default is table.
RepeatDirection	it specifies the direction in which the Controls to be repeated. the values avilble are Horizontal and vertical, Default is vertical.

PAGE NO.:
DATE: / /

RepeatColumns it specifies the number of columns to use when repeating the controls; default is 0.

Bulleted lists :⇒ it creates bulleted lists or Numbered list. these Controls Contain a Collection of ListItem Objects that Could be referred to through the items property of the Control.

<asp: BulletedList ID="List1" runat="server" BulletStyle="Circle"
DisplayMode="Text" ImageUrl="Images/Arrow.ico" /> "Disc"
"HyperLink" "Square"
"LinkButton" "Numbered"

</asp:BulletedList>

Rich web controls functionality know as to do some advanced

Rich web Controls :⇒ the Control which are not standard Control and allow

Calender :⇒ the Calender Control is a as functionalities ~~which~~ Rich web Control's which Provides a following Capabilities

a) displaying one Month at a time.
b) selecting a days a week or a Month
c) Selecting a range of days
d) Moving from Month to Month
e) Controlling the display of the day.

PAGE NO.:
DATE: / /

<asp:Calendar ID="Cal1" Runat="server" SelectionMode="DayWeekMonth" />

event :- SelectionChanged

```
lbl1.Text = Cal1.TodaysDate.ToShortDateString();
lbl2.Text = Cal1.SelectedDate.ToShortDateString();
```

file upload :=> Asp.Net has two controls that allow to user to upload file to the web server. Once the server receive the posted file data, the application can save it, check it, or ignore it, the following controls allows the file uploading.

- HtmlInputfile - an HTML server control.
- FileUpload :- and Asp.NET web control.

both control allow file uploading, but the file upload control automatically sets the encoding of the form, whereas the HtmlInputfile does Not do so.

=> the FileUpload control allow the user to browser for and select the file to be uploaded, providing a browse button and a text box entering the filename.

=> once the user entered the filename in the text box by typing the name or browing the SAve As Method of the FileUpload Controlled can be called to save the file to the disk.

Month

Syntax of FileUpload is :-

<asp: FileUpload ID = "uploader" runat = "server" />

* the fileUpload class is derived from the WebControl class and inherits all its members. Apart from those, the FileUpload class has the following read-only Properties.

Properties	description
FileBytes	Returns an array of the bytes in a file to be uploaded.
FileContent	Return the stream object pointing to the file to be uploded.
FileName	Returns the Name of the file to be uploaded.
HasFile	specifies whether the control has a file to upload.
PostedFile	Returns a refrences to the uploaded file.

the PostedFile is encapsulated in an object of type HttpPostedFile, which could be accessed through the PostedFile property of the FileUpload class.

Properties	description
ContentLength	Returns the size of the uploaded file in bytes.
ContentType	Returns the MIME type of the uploaded file.
FileName	Returns the fullfilename.
InputStream	Returns a stream object Pointing to the uploaded file.

```
<body>
    <form id="form1" runat="server">
    <div>
        <h3> File upload :</h3>
        <br/>
    <asp: fileUpload ID="file1" runat="server" /> <br/> <br/>
    <asp: Button ID="Button1" runat="server" onclick="Button1_Click"
        Text="Save"  style="width:85 px" />
            <br/> <br/>
    <asp: Label ID="label1" runat="server" />
    </div>
</form> </body>
```

```
Protected void Button1_Click (Object sender, Event Args e)

if (file1.HasFile)
 {
   if (file1.PostedFile.ContentLength < 1024)
    {
      String fn = System.IO.Path.GetFileName(file1.FileName);
    file1.SaveAs (Server.MapPath ("~/user photo/") +fn;
    Image1.ImageUrl = "~/user photo" /+ fn;
    label1.Text = "Photo upload Successfully";
    }
else
 {
   label1.text = "file Should be less than 1 Mb";
 }
else
```

PAGE NO.:
DATE: / /

```
{
    label1.Text = "file choose first for upload";
}
```

AdRotater Control :⇒ the AdRotater Control yamelamley select or rotate advertaisment / banner / graphics from a list which is specified in an external XML schedule file. this external XML schedule file is called the advertisement file that is contain information above the advertisement to be display.

```
<asp: AdRotater ID="AdRotater1" Runat="server"
    AdvertisementFile = "Ads.xml" target = "_blank" />
```

```
<Advertisements>
  <Ad>
  <ImageUrl> Images/flipkart logo.jpg </ImageUrl>
  <NavigateUrl> https://www.flipkart.com </NavigateUrl>
  <AlternateText> flipkart online shopping </AlternateText>
  <Keyword> E-Commerce, e-Shopping </Keyword>
  <Impression> 40 </Impression>
  <Height> 2vw </Height>
  <Width> 2vw </Width>
  </Ad>
  <Ad>                                  Snapdeal, Amazon
  - - -                                 OLX, e-bay
  - - -
  - - -
  </Ad>
</Advertisements>
```

Element	Description
Advertisements	Enclose the Advertisements file
Ad	enclose the seperate Ad
ImageUrl	the Path of image that will be displayed
NavigateUrl	the link that will be followed when the user Click's the Ad.
AlternateText	the text that will be displayed instead of Pictures if it can't be display
Keyword	the keyword identify a group of advertisment this is use for filtering
Impressions	the number integrating How often an advertisements will appear.
Height	Height of image to be displayed.
width	width of the image to be displayed.

Properties and Events of the AdRotater Class.

- the AdRotater class is derived from the web control class and inherits its Properties.

Properties	Description
Advertisement file	the Path to be advertisement file.
AlternateText field	the element name of the field where alternate text is provided.
Data Member	the name of the specific list of data to be bound when Advertisement file is Not used.
Data Source	Control from where it would retrive data

DataSourceID	ID of the Control from where it would retrieve data.
Font	specifies the font properties associated with the advertisement banner Control.
ImageUrlField	the element name of the field where the URL for the image is provided.
KeywordFilter	for displaying the keyword based ads only.
NavigateUrlField	the element name of the field where the URL to Navigate to is provided.
Target	the browser window or frame that displays the content of the page linked.
UniqueID	Obtains the unique, hierachically qualified identifier for the AdRotator Control.

Events	Description
AdCreated	it is raised once per round trip to the server after creation of the Control, but before the Page is rendered
DataBinding	occurs when the Server Control binds to a data source.
DataBound	occurs after the server controls binds to a data source
Disposed	occurs when a server Control is released from memory, which is the last stage of the Server Control life cycle when An Asp.Net Page is requested
Load	occurs when the Server Control is loaded into the Page Object
PreRender	occurs after the Control object is loaded but prior to rendering.

Validation Control :⇒ Asp. Net Validation Control Validate the user input data to ensure that useless, unauthentic or wrong data doesn't get Stored.

Asp Common Properties Of Validation Control

	Properties	Description
①	ControlToValidate	indicate the input Control to validate
②	Display	indicates how the error Message Shown
③	ErrorMessage	indicates error String
④	Text	Error text to be Shown if validation fails
⑤	IsValid	indicates whether the value of Control is valid
⑥	ValidationGroup	the logical group of multiple data where this Control belongs
⑦	Enabled	enables or disable the valid data.

① RequiredFieldValidator :⇒ the RequiredFieldValidation Control ensure that the required field is Not empty or a text box to fault input into the text.

Syntax :-

<asp:RequiredFieldValidator ID="RequiredI" Runat="server"
ControlToValidate="TextUser" Display="Dynamic"
ErrorMessage="User name Cannot be left blank" />

Enter User Name [ ] username Can Not be left blank

[Submit] — click

② Range Validator :⇒ the Range Validator Contr. verify that input value in the textbox without a Predetermind range.

Properties types :- it define the type of the data. The avilable values are Concurrency data double, integer and String.

Maximum value :⇒ it specify the Maximum value of the Range.

Minimum values :⇒ it specify the Minimum value of the Range.

Syntax :-

<asp: RangeValidator ID = "Range1" Runat = "Server" ControlToValidate = "textamount" Display = "Dynamic" Type = "Currency" MinimumValue = "100" MaximumValue = "1000" ErrorMessage = "Please input valid currency (100-1000)" />

Text Amount

Enter currency [2000] Please input valid currency (100-1000)

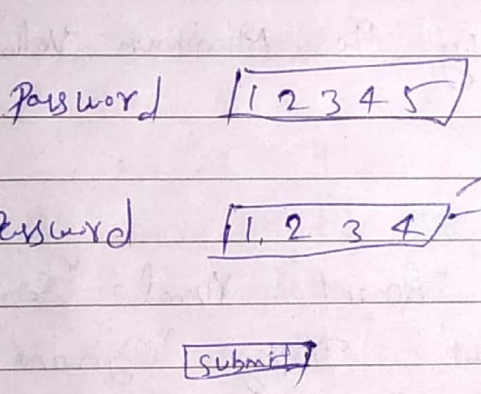
[Submit]

PAGE NO.:
DATE: / /

(5) Compare Validator :=> this Control Compares a value in one Control with a fixed value or a value in another Control.

Properties

(i) types :- it specifies the types of data
(ii) Control to Compare :- it specifies the value of the input Control to Compare with
(iii) Value to Compare :- it specifies the constant value to Compare with
(iv) Operator :- it specifies the Comparision operators the avilable values are equals, not equals, greater then, greater then equal, less then, less then equal and data type check.



Syntax :-

<asp: CompareValidator ID = "Compare" runat = "Server" ValueToCompare = "Password Control To Compare = "text pass 1" ControlToValidate = "text pass 2" ErrorMessage = "Password does Not Match" Display = "Dynamic" />

PAGE NO. :
DATE : / /

Regular Expression Validator :=> the Regular Expression Validator allows Validating the input text by Matching against a Pattern ~~Expression~~, of a regular expressions, the regular Expressions is set in the Validation Expression Property.

<asp: RegularExpressionValidator ID="string" runat="server" ErrorMessage="string" ValidationExpression="string" ValidationGroup="string"> </asp: RegularExpressionValidator>

Custom Validator :=>

the Custom Validator Control allows writing application specific custom Validation routines for both the client side and the server side Validation

the client side Validation is accomplished through the ClientValidationFunction Property. The client side validation routine should be written in Scripting language, such as JavaScript or VBScript, which the browser can understand
The server side Validation routine must be called from the control's ServerValidate even handler. The server side validation routine should be written in any .Net language like C# or Vb.net.

<asp: CustomValidator ID="Custom1" runat="server" ClientValidationFunction=".cvf-func. ErrorMessage:" Custom Validator"> </asp: CustomValidator>

PAGE NO. :
DATE : / /

Validation Summary Control :⇒ Validation Summary Control allows to supplies or validation error Message from all Validator's is a single location the error Message display in this control in Specified by the error Message Property of each Validation Control.

Property	Description
DisplayMode	How to display the summary legal Values are Bullet List, List or Single Paragraph
HeaderText	A Header in the Validation Summary Control
ShowMessageText	the Boolean Value get specifies wor the Summary should be display in a Message box or Not
ShowSummary	the Boolen Values specifies the Validation Summary Should be display or hidden.

<asp : ValidationSummary ID = "VS" Runat = "Server" HeaderText = "the following fields are Medentory:" ShowMessageText = "True" ShowSummary = "false" DisplayMode = "BulletList" />

PAGE NO.:
DATE: / /

Working with events :=>

An event is an action or occurrence such as a Mouse Click, a key Press, Mouse Movements or any System-generated notification. A Process Communicates through events.

Events in Asp.Net raised at the client Machine, and handled at the Server Machine

eg: A user clicks a button displayed in the browser. A click event is raised. the browser handles this client side event by Posting it to the Server.

The Server has a Subroutine describing what to do when the event is raised : it is called the event handler

Event Arguments

Asp.Net event handlers generally take two Parameters and return void. the first Parameter represent the object raising the event and the second Parameter is event argument

```
Private Void EventName (object sender, EventArgs e);
```

Application and Session Events :-

Application_start -> it is raised when the application is started
Application_End -> it is raised when the application is stopped
Session_start :- it is raised when a user first request a page from the application
Session_End :- it is raised when the session ends.

Event	Attribute	Controls
Click	OnClick	Button, ImageButton, LinkButton, ~~Image~~, imagemap
Command	OnCommand	Button, imagbutton, linkbutton
Text Changed	OnTextChanged	Textbox
SelectIndexChanged	OnSelectIndexChanged	Drop-down list, list box, radioButton list, checkbox list
CheckedChanged	Oncheckedchanged	checkbox, radio button
Selection changed	OnSelectionChanged	Clander

Some events cause the form to be posted back to the server immediately, these are called the postback events.

eg. the Click event such as, Button.Click

Some events are not Posted back to the server immediately, these are called non-postback events.

eg! the change events or selection events such as TextBox, TextChanged or CheckBox.CheckedChanged. the non postback events could be Made to post back immediately by setting their AutoPostBack property to true.

PAGE NO. :
DATE : / /

* Some events cause the form to be Posted back to the server immediately, these are called the Postback events.

* Some events are not Posted back to the server immediately these are called Non-Postback events.

Asp.net life cycle :⇒

* Asp.NET life cycle could be divided into two groups.

Application Life Cycle
Page Life Cycle

Asp.NET Application Life Cycle :⇒

→ User Makes a request for accessing application resource, a Page. Browser sends this request to the web server.

→ A Unified Pipeline receives the first request and the following events take place :

→ An object of the class ApplicationManager is created.

→ An object of the class HostingEnvironment is created to provide information regarding the resources.

→ Top level items in the application are compiled.

→ Response objects are created, the application objects such as HttpContext, HttpRequest and HttpResponse are

Created And initialized

⇒ An instance of the HttpApplication object is created and assigned to the request.

⇒ The request is processed by the HttpApplication class. Different events are raised by this class for processing the request.

Asp.NET Page Life Cycle :-

When a Page is requested, it is loaded into the server Memory, Processed and sent to the browser. Then it is unloaded from the Memory. At each of these steps, Methods and events are available, which could be overridden according to the Need of the application, in other words, we can write our own Code to override the default Code.

the Page class creates the hierachical tree of all the controls on the Page All the component on the Page, except the directives, are part of this (Contro) tree

different Stage of Asp.NET Page ⇒

→ Page request ⇒ When ASP.NET gets a Page request, it decides whether to Parse and Compile the Page, or there would be a Cached version of the Page; accordingly the response is sent.

89-SU-SB-86

PAGE NO. :
DATE : / /

→ Starting of Page life cycle :⇒ At this stage, the request and Response objects are set. If the request is an old request or post back, the IsPostBack Property of the Page is set to true. The UICulture Property of the Page is also set.

→ Page Initialization :⇒ At this stage, the controls on the Page are assigned unique ID by setting the UniqueID Property and the themes are applied. For a New request, Postback data is loaded and the control properties are restored to the view-state values.

→ Page load :⇒ At this stage, control properties are set using the view state and control state values.

→ Validation :⇒ Validate Method of the validation control is called and on it's successful execution, the IsValid event handler is invoked.

→ Page rendering :⇒ At this stage, view state for the Page and all controls are saved the Page calls the Render Method for each control and the output of rendering is written to the OutputStream class of the Response Property of Page.

→ Unload → the rendered Page is sent to the client and Page Properties, such as Response and Request, are unloaded and all cleanup done.

PAGE NO.:
DATE: / /

Asp.NET Page Life Cycle Events :-

At each stage of the page life cycle, the page raises some events, which could be coded. An event handler is basically a function or subroutine, bound to the event, using declarative attributes such as OnClick or handle.

→ Preinit :→ Preinit is the first event in page life cycle. it checks the IsPostBack property and determines whether the page is a postback. it sets the themes and master pages, creates dynamic controls, and gets and sets profile property values. this event can be handled by overloading the OnPreInit Method or creating a Page_PreInit handler.

→ Init → Init events initialize the control property and the control tree is built. this event can be handled by overloading the OnInit method or creating a Page_Init handler.

→ Init Complete :→ Init Complete event allows tracking of view state. All the controls turn on view-state tracking.

→ LoadViewState :→ LoadViewState event allows loading view state information into the controls.

→ LoadPostData :→ During this phase, the contents of all the input fields defined with the <form> tag are processed.

PAGE NO.:
DATE: / /

→ Preload → Preload occurs before the Post back data is loaded in the controls. this event can be handled by overloading the onpreload method or creating a Page-preload handler.

→ Load :⇒ the load event is raised for the Page first and then recursively for all child controls. The controls in the control tree are created. this event can be handled by overloading the OnLoad method or creating a Page-Load handler.

→ Load Complete :⇒ the loading process is completed, control event handlers are run, and Page validation takes place. this event can be handled by overloading the OnLoad Complete method or creating a Page-LoadComplete handler

→ Pre Render :⇒ The PreRender event occurs just before the output is rendered. By handling this event, Pages and Controls can perform any updates before the output is rendered.

→ PreRender Complete :⇒ As the PreRender event is recursively fired for all child controls, this event ensures the Completion of the Pre-rendering phase

→ Save State Complete :⇒ State of control on the Page is saved Personalization, control State and View State information is saved. The Html Markup is gener This stage can be handled by overriding the Render Method or creating a Page-Render handler.

Unload :⇒ The Unload Phase is the last phase of the Page life cycle. it raises the Unload event for all controls recursively and lastly for the Page itself. Final cleanup is done and all resource and References such as database connections are freed. this event can be handled by Modifying the onUnload Method or Creating a Page-Unload handler.

What is Master Page ?

Ans Asp.NET master Page allow to create a Consistent layout for the Page in our application A Single Master Page defines the look and feel and standard behaviour that we want for all the Pages (or a group of Pages) in our application. we can then create individual Content Pages that contain the Content we want to display. when user request the Content Pages, they merge with the master Page to Produce output that combines the layout of the Master Page with the Content from the Content Page.

State Management in Asp.Net :⇒

State Management is the Process by which we Maintain State and Page information over Multiple request for the same and different Page.

⇒ HTTP is the stateless protocol. when the client disconnects from the server, the Asp.NET engine discards the Page

objects. This way, each web application can scale up to service numerous requests simultaneously without running out of server memory

However, there needs to be some technique to store the information b/w requests and to retrieve it when required. this information that the current value of all the controls and variable for the current user in the current session is called the state.

Asp.NET includes several options that help we preserve data on both a per-page basis and on application-wide basis

there are two state Management technique / Methodlogies

State Management in Asp.NET

Client Side
- View State
- Query String
- Hidden field
- Control State
- Cookies

Server Side
- Application State
- Session State

PAGE NO.:
DATE: / /

④ Client side

① View State :→ Asp.NET usage View State to track the values in the Controls it is used by the Asp.NET page framework to automatically save the value of the Page and of each Control just re-entring to the Page.

The View State Property provides a dictionary object for retaining values b/w multiple request for the same Page. this is the default Method that the Page uses to preserve Page and Control Property values b/w round trips. when the Page is processed, the current State of the Page and Controls is hashed into a string and saved in the Page as a hidden field or multiple hidden fields if the amount of data store in the View State property exceeds the specified value in the MaxPageStateFieldLength Property. When the Page is posted back to the server, the page parses the view-state string at Page initialization and restores property information in the Page.

eg:-

```
if (IsPostBack)
{
  if (ViewState["Count"] != null)
  {
    int32 ViewStateValue = Convert.ToInt32(ViewState["Count"]) + 1;
    Label1.Text = ViewStateValue.ToString();
    ViewState["Count"] = ViewStateValue.ToString();
  }
else
  {
    ViewState["Count"] = "1";
  }
}
```

Label1.Text = ViewState["Count"].ToString();

(2) Control State :=> If we Control a Custom Control that required View State to work Property, we Should use Control State to ensure other developer don't break our Control by disabling View State.

(3) Hidden Field :=> A hidden field is use for storing small amount of data on the client side

Asp.Net allows to store information in a HiddenField Control, which renders as a standard HTML hidden field. A Hidden Field does not render visibly in the browser, but we can set it's properties just as we can with a standard Control. When a Page is submitted to the server, the content of a hidden field is sent in the HTTP form Collection along with the value of other Controls. A hidden field acts as a repository for any Page-specific information that we want to store directly in the Page. In most simple word it is just a container of some object but there result is known reentered on our web browser it is individual in the browser

```
Int32 Count = Convert.ToInt32(HiddenField1.Value)+1;
HiddenField1.Value = Count.ToString();
Label1.Text = CountToString();
```

④ Cookies :⇒ Cookie stored a value in the user browser send with every Page request to the same server. Cookies are the best way to stored data that must be avilable for Multiple web pages or web - in other words a Cookie is a small amount of data that is stored either in a text file on the client file system or in-memory in the client browser session it contains site-specific information that the server sends to the client along with Page output. Cookies can be temporary or Persistent.

the Cookies are store information about a Particular client, session or application. The Cookies are saved on the client device and when the browser request a Page, the client sends the information in the Cookie along with the request information, the server can read the Cookie and extract it's value. A typical use is to store a token indicating that the user has already been authenticated in our application.

```
Int32  PostBacks = 0;
if (Request.Cookies["count"] != NULL)
{
    Postback = Convert.ToInt32(Request.Cookies["count"]+1);
}
else
{
  Postbacks = 1;
}

Response.Cookies("count").Value = Postbacks.Testing();
lbl1.Text = Response.Cookies["count"].Value;
```

Cookies are two types

① Persistent ⑪ NON-Persistent

① Cookies having an expiring date has called Persistent Cookies this type of Cookies riches there expiration date come to an end or expire in this Cookies we send an Expiration date

② NON-Persistent types of Cookies are Not Stored in the client harddrive Permanantley it Maintainance users information as long as the user accen or usage the service.

PAGE NO.:
DATE: / /

Query String :->

first.aspx

Username [Admin] TxtUser

Second.Aspx

⇒ Query String used for handling some value for different Page and Move these value to the different page. The information stored in it can be easily navigated to one Page to another or to the same Page as well for example:-

first.aspx

Username [Admin] - TextUser

Email [xyz@gmail.com] - TextMail

[Submit]

Second.aspx

btn_Click

Step-1

Response.Redirect ("Second.aspx?User=" TextUser.Text "+" &Email = "+ TextEmail.Text);

Step2

Local host/website/Second.Aspx?User=Admin&Email=Admin@

Page Load

Step-3

Label User.Text = Request.QueryString["User"].ToString();
Label Email.Text = Request.QueryString["Email"].ToString();

PAGE NO.:
DATE: / /

(B) Server side state Management technique

(1) Application State :- Application State is a Server Side State Management technique. Application State is a Stored in the Memory of the Server and is faster than storing and Retriving information in a data base.

→ Application State does Not have a default. When we close the Process application State will be lost

→ Application State is for all users and sessions

ex.

Application Start

Application ["Counts"] = 0;

Session - start

Application.lock();

Application ["Counter"] = Convert.ToInt32

Application.Unlock;

Session - End

Application.Lock();

Application ["counter"] = Convert.ToInt32 (Application ["counter"])

Application.Unlock();

Page load

lblUser.Text = "you are visitior = " + Application ["counter"].

ToString();

Session.Abandon(); - Close Complete Session

PAGE NO.:
DATE: / /

(17) Session State :=>

=> Session State information is avilable to all Pages opened by a user during a single visit

=> Asp.Net allows to Save Values using Session State a Storage Machinasm that is accessible from all Pages requested by a single web browser session we Can use Sesion State to store Specific user information like userid Password etc.

=> Session State Similar to application State it is specked to the current browser session.

=> If different user are using our website, each user session has a different session State.

Eg. One.aspx

User [ ] textUser

Pass [ ] textPass

Login

Button_click

Session["user"] = textUser.text;
Session["pass"] = textPass.text;

Another.aspx

Lab1.User = Session["user"].ToString();
Lab1.pass = Session["pass"].ToString();

PAGE NO. :
DATE : / /

Different between Array and ArrayList

Array	ArrayList
Array is strongly types. This means that an array can store only specific type of items/elements.	ArrayList can store any type of item/elements.
Array stores fixed number of elements. size of an array must be specified at the time of initialization	ArrayList Grows automatically and we don't need to specify size.
No Need to cast elements of an array while retriving because it is strongly type and stores type of items only.	items of ArrayList need to be cast to appropriate data type while retriving
use static helper class Array to perform different tasks on the Array	Arraylist itself includes various utility methods for various task.

type Conversion :⇒ type Conversion is Converting one type of data to another type. it is also known as type Casting
two types of type casting

① Implicit type Conversion :⇒ these Conversion are performed by C# in a type-safe Manner
for ex;- are Conversion from Smaller to larger integral types and Conversions from derived classes to base classes.

PAGE NO. :
DATE : / /

② Explicit type Conversion :- These Conversion are done explicitly by users using the pre-defined functions.
Explicit Conversions require a Cast Operator.

eg:-
```
using System;
    namespace TypeConversion
    {
        class ExplicitConversion
        {
            static void Main(String[] args)
            {
                double d = 5673.74;
                int i;
                i = (int) d;
                Console.WriteLine();
                Console.ReadLine();
            }
        }
    }
```

3
5673

Operator :- ⇒ An operator is a symbol that tells the Compiler to perform specific mathematical or logical Manipulations

- Arithmatic opt.
- Relational opt.
- Logical opt.
- Bitwise opt.
- Assignment opt
- misc opt.

PAGE NO. :
DATE : / /

Arithmatic opt :=>

A=10, B=20.

operator		Example
+	A+B	30
-	A-B	-10
*	A*B	200
/	A/B	2
%	A%B	0
++	A++	11
--	A--	9

Relational opt :=> A=10 B=20

operator	Example	
==	A==B	false
!=	A!=B	true
>	A>B	false
<	A<B	true
>=	A>=B	false
<=	A<=B	true

Logical opt :=>

A = holds Boolean value
B = hold's Boolean falg

operator		Example
&&	A&&B	false
||	A||B	true
!	!(A&&B)	true

Bitwise Opt:-

P	q	P&q	P|q	P^q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Ex: Here A=60 B=13 the

A = 0011 1100
B = 0000 1101

A&B = 0000 1100
A|B = 0011 1101
A^B = 0011 0001
~A = 1100 0011

A << 2 = 1111 0000
A >> 2 = 0000 1111

Assignment Opt:=>

operator	Example
=	C=A+B
+=	C=C+A
-=	C=C-A

*=	C = C*A
/=	C = C/A
%=	C = C%A
<<=	C = C<<2
>>=	C = C>>2
&=	C = C&2
^=	C = C^2
|=	C = C|2

Miscellaneous opt :⇒

Size of	Returns the size of data type
type of	Returns the type of a class
&	Returns the address of an variable
*	Pointer to a variable
?:	Conditional expression
is	Determines whether an object is of a certain type
as	Cast without raising an exception if the cast fails

Exception Handling :⇒ An exception is a Problem that arises during the execution of a Program. A C# exception is a response to an exception circumstance that arise while a Program is running, such as an attempt to divide by zero.

Exception handling is built upon four keyword's :-
try, catch, finally and throw.

→ try :⇒ A try block identifies a block of code for which Particular exception is activated it is followed by one of

PAGE NO.:
DATE: / /

More Catch block

Catch :→ A Program Catches an exception with an exception handler at the place in a Program where we want to handle the problem. The Catch keyword indicates the Catching of an exception.

finally :→ the finally block is used to execute a given set of statements, whether an exception is thrown or Not thrown.

for ex:- if we open a file, it must be closed whether an exception is raised or Not.

throw :- A Program throws an exception when a problem shows up. this is done using a throw keyword.

Syntax :

```
try
{
    Statement Causing exception
}
Catch (Exception e1)
{
    // error handling Code
}
catch (Exception e2)
{
    // error handling Code
}
finally
```

{

// statements to be executed

}

S.No	Exception class	Description
1	System.IO.IOException	Handles I/O errors.
2	System.IndexOutOfRangeException	Handles errors generated when a method refers to an array index out of range
3	System.ArrayTypeMismatchException	Handles error generated when type is mismatched with the array type
4	System.NullReferenceException	Handles errors generated ~~when type is Mismatched~~ from referencing a null object
5	System.DivideByZeroException	Handles errors generated from dividing a dividend with zero
6	System.InvalidCastException	Handles errors generated during typecasting
7	System.OutOfMemoryException	Handles error generated from insufficient free memory
8	System.StackOverflowException	Handles errors generated from stack overflow

```
using System;
  namespace ErrorHandlingApplication
  {
    class DivNumbers
     {
       int result;
       DivNumbers()
        {
          result = 0;
       }
      Public void division (int num1, int num2)
       {
         try
          {
            result = num1 /num2;
          }
        Catch (DivideByZeroException e)
        {
          Console.WriteLine("Exception Caught : {0}", e);
        }
        Finally
          {
            Console.WriteLine ("Result: {0}", result);
          }
      }
      Static void Main (String[] args)
       {
         DivNumbers d = new DivNumbers();
          d.division (25, 0);
          Console.ReadLine();
      }}
```

PAGE NO. :
DATE : / /

String :⇒ String is immutable, immutable means if we create String object then we cannot Modify it and it always create New object of String type in Memory

Ex. String strMyValue = "Hello Visitor";
strMyValue += "How Are";
strMyValue += "you ??";

String builder :⇒ StringBuilder is Mutable Means if create string builder object then we can Perform any operation like, insert, replace or append without creating New instance for every time, it will update string at one place in Memory doesnt create New Space in Memory

ex:- StringBuilder sbMyValue = new StringBuilder("");
sbMyValue.Append("Hello Visitor");
sbMyValue.Append("How are you?");
String strMyValue = sbMyValue.ToString();

IsPostBack Property :⇒ In Asp.Net when a Page is the loaded, it is Possible to check with her and .Aspx Page is Posted Back to the Server with the help of the IsPostBack Property it determines whether or Not Post Back

eg:- Someone clicking a button on the Page always Page Load event fires while a Page that is being loaded for the first time check should be a Postback or Not.

800 kg [मिलर ओरा 187 → फेंगी 240 / चेरी 175 / पनाली 135 / सासपेन 250 / चानी 176] 150 kg — आर metal 280 — 3715

PAGE NO.:
DATE: / 07/12/02

```
If (! IsPostback)
{
    Label1.Text = "Page is Post back first time";
}
else
{
    Label.Text = "Page is Postback Again";
}
```

Page Directives :-> the Page directives set up the environment for the Page to run. The @Page directive defines Page-specific attributes used by Asp.NET Page Parser and Compiler, page directives specify how the Page should be Processed and which assumptions need to be taken about the Page.

Code Section :=> the Code Section Provides the handlers for the Page and Control event along with other functions required. We Mentioned that, Asp.NET follows an Object Model. Now these objects raise events when some events take place on the user interface; like a users click a button or Moves the cursor. the Kind of response these events Need to reciprocate is Coded in the event handler functions. the event handler are Nothing but functions bounds to the Control

the Code section or the Code behind file provides all these events handler routines, and other functions used by the developer. the Page Code Could be precompiled and deployed in the form of a binary assembly.

Scanned by CamScanner

PAGE NO. :
DATE : / /

Creating and deploying Asp. Net application :=>

two categories of deployment

→ Local deployment :=> in this case, the entire application is contained with a virtual directory and all the contents and assembilies are contained within it and avilable to the application

→ Global deployment :=> in this case, assemblies are avilable to every application running on the server.

different technique used for deployment

① → Xcopy deployment
② → Copying a website
③ → Creating a set up Project

① Xcopy deployment :=> Xcopy deployment means making recursive copies of all the files to the target folder on the target machine

→ FTP transfer
→ Using Server Management tool that Provide replication on a remote site
→ MSI installer application.

Xcopy deploymnt simply copies the application file to the production server and sets a virtual directory there. we need to set a virtual directory using the internet information

Manages Microsoft Management Console.

(2) Copying a website :⇒ the copy website is avilable in visual studio option. it is avilable from the website copy web site menu option. this menu item allows copying the current web site to another local or remote location. it is a sort of integrated FTP Pool.

(3) Creating a setup Project :⇒ in this Method we use windows Installer and Package our web applications so it is ready to deploy on the Production Server. Visual Studio allow to build deployment Package.

Step 1 :⇒ Select file → Add → New Project with the website root directory highlighted in the Solution explorer.

Step 2 :→ Select setup and Deployment, Under other Project Types, Select setup wizard.

Step 3 :- chosing the default location ensures that the set up Project will be located in its own folder under the root directory of the site. Click on OKay to get the first Splash screen of the wizard.

Step 4 :- Chose a Project type. Select "creat a setup for a web application"

Step 5 Next, the third screen asks to chose Project output from all the Projects in the solution check the check box to Content files from...

Step 6:- the fourth screen allows including other files like Read Me. after ~~However~~, in our ~~case~~ there is no such file. click to finish

Step 7:- the final screen displays a summary of setting for the setup Project.

Step 8:- the setup Project is added to the Solution Explorer and the Main design window shows a file system editor.

Step 9:- Next step is to build the setup Project. Right click on the Project name in the solution Explorer and select Build.

Step 10 When build is Completed, ~~be~~ we get the message

two files are created by the build Process.

- setup.exe
- setup.databinding.msi

Debugging :=> Debugging allows the developers to see How the code work's in a step by step Manner How the values of the variables Changes and How the object are created & destoryed etc.

When the site is executed for the first time visual Studio displaying a from asking with her it should be enable for debugging.

PAGE NO.:
DATE: / /

Break Point :-> Break Point specifies the Runtime to Run a specific line of code and then stop execution so that the code could be and Perform various debugging such as Changing the Value of variable Error through the code, Moving in and out function Method.

* How to set a Break Point :-> to set a break Point right Click on the code and Choose insert break Point.

a red dot appears on the left Margin and the line of code is highlighted.

Virtual directory :-> a virtual directory is a directory Name that we specified in IIS (internet information services) and Map to Physical directory on a local server hard drive or a directory on a another server (remote server) who can use IIS Manager to create a virtual directory for an asp.Net web application that is hosted in IIS How to create in IIS Manager, expect the local Computer and the website which we want to add virtual directory

Right click the side of folder in which we want to create the virtual directory Click New and click virtual directory

in the add virtual directory dialog box at the minimum enter information in the alias and Physical Path and the Click OK.

How to Run IIS :-> in Run dilog box type INETMGR and click OK.

PAGE NO.:
DATE: / /

web.Config file :⇒ it is the Main setting and Configration file for an asp.Net application. it is an XML documents that decide in the root directory of the site and the application and Contain data about How the application will act.

this information Controls Module loading, Security, Configuration, Session State Configuration and application language, Compilation Setting and database Connection etc.

difference b/w Response.Redirect and Server.transfer Method

	Response.Redirect	Server.transfer
①	Response.Redirect has a round trip in the combination of a request sent back to browser	where Server.transfer has Not round trip
②	Response.Redirect is a client Process	it is a Server Process
③	it Preserve query string and variable from the origional request	it doesn't preserve query string (optional)
④	when we want to allow website URL can be copied then Response.Redirect is better	but security reason Server.transfer is better because URL can't be copied.
⑤	Response.Redirect URL changed	URL Not changed
⑥	Response.Redirect the can redirect the user to both type page .Html or .Aspx	but in case Server.transfer we can redirect user to .asp or .aspx Page

PAGE NO. :
DATE : / /

⑦ Response.Redirect Send Message to the browser saying it to More Some different Page

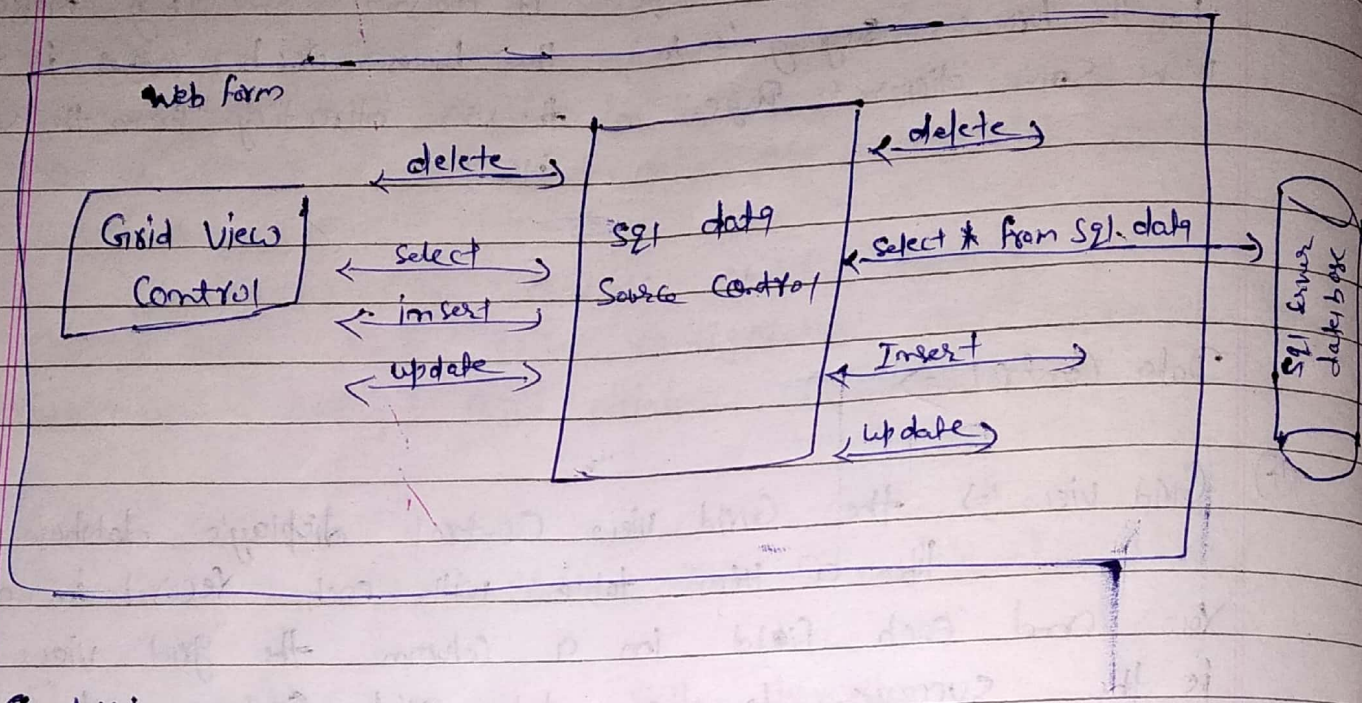
It doesn't Send any Message to the browser or but rather it redirect the user directory from the server itself.

* Data Control :=>

① Grid view :=> the Grid view Control display's database records like a Html table with each record in a table row and each field in a column the grid view Control is the successor to the data grid Control. the Grid view Control is a feature rich and versitile Control use to accept display and edit data on a web page it is Commonly used Control in Asp.Net web application.

=> a grid view Control and a Sql data Source Control is placed on the web form the select, delete, insert and update Command are use to intract with the Sql Server database using data Source ID Property

=> grid view support Programatic access to the grid view object Model and set Properties and handles event dynamicalley.



Syntax :-

<asp: GridView ID = "GridView1" Runat = "Server" AutoGenerateColumns = "false" DataSourceID = "SqlDataSource1" AllowPaging = "true" PageSize = "10" />

Data List :⇒

⇒ Data List is an Unformatted Data Control like Repeater Control in Asp.Net

⇒ Data List Controls are used to display a list of items.

⇒ the Data List Controls is useful for displaying data in any repeating structure.

⇒ Eval() and Bind() can be used to bind data in Data List

Details view!=>

=> Details view is a formatted data Control like GridView Control in Asp.Net
=> The Detail View Control display only a single data record at a time, even if its data source Exposes multiple records
=> CRUD operation are Possible in Detail View
=> Asp:Boundfield is used to bind data in Details View.

Database Control:=> the database Control display data in a formed that we can define using templates and Styles.
the datalist Control is useful for data in any repeating structure such as a table the data list Control can display rows in different layout such as ordering them in column or rows
the datalist Control is like the repeater Control we to display repeated list item that are bound to the Control However the data list Control add a table around the data items by default

<asp:DataList ID="datalist1" Runat="Server"
Repeat Columns="5" Repeat direction="Horizantly"
Cell spacing="5" Cell padding="5">
<HeaderTemplet> ... </HeaderTemplate>
<ItemTemplet> ... <ItemTemplete>
<FotterTemplet> ... </FotterTemplete>
</asp:DataList>

Repeater Control :⇒ Repeater Control is a Control which is used to display the repeated list of item.

⇒ Repeater Control is used to display Repeated List of items that are bound to the Control Same as grid view and data grid.

⇒ Repeater Control is light weight and faster to display data then grid view and data grid.

⇒ Repeater Control display data in Custom format but it's Not Possible Grid view and data Grid.

⇒ Repeater Control doesn't Support Pagging and Shorting

⇒ the Repeater Control work by looping through the records in our data Source and then Repeating the rendring of it's template called item template.

⇒ Repeater Control Contains different type of template fields those are

① ItemTemplete
② AlternatingItemTemplate
③ HeaderTemplate
④ footerTemplate
⑤ SeperatorTemplate

PAGE :
DATE : / /

How to Populate data grid view Control

Note:- dbshopping - database
tblProduct - table

```
System.Data;
System.Data.SqlClient;

SqlConnection Con = new SqlConnection ("server=localhost; UId=
        sa; Password=sql; database=dbshopping")
    Con.Open();
SqlDataAdapter Adap = new SqlDataAdapter ("Select * from tblProduct",
    Con);
DataSet ds = new DataSet();
Adap.Fill (ds, "tblProduct");
  GridView1.DataSource = ds;
  GridView1.DataBind();
Con.Close();
```