

29/11/19

* Rich controls go Asp.net

a) Calendar Control :-

Step 1: Drag and drop a TextBox, TextBox, ImageButton and a CalendarControl on the webform. Create an Images folder and add the following calendar.png to the Images folder.

Step 2: Set the Image URL property of the image button to Calendar.png.

Image URL = "~/Images/Calendar.png"

.aspx

Step 3: Change the following properties of a control.

Control	Property	Value
---------	----------	-------

1. TextBox	- ID	TextBox1
	- Width	115px

2. ImageControl	- ID	ImageButton1
	- ImageURL	"~/Images/Calendar.png"
	- OnClick	ImageButton1_Click
	- Click	

DATE: / /

3. Calendar - ID "Calendar1"
- onselectionchanged "SelectionChanged"

Step 4:- Code behind page (aspx.cs)

Page_Load()

{
if (!IsPostBack)

{

}
} Calendar1.Visible = false;

}

ImageButton1_Click()

{
if (Calendar1.Visible)

{
}

Calendar1.Visible = false;

{

else

{
Calendar1.Visible = true;

}

}

Calendar1_SelectionChanged()

{
}

TextBox1.Text = Calendar1.Selected

Date.ToString("ShortDateString");

"}
} Calendar1.Visible = false;

}

(on ToString("LongDateString"))

Step 5:- (F5)

Run the application.

b) AdRotatorControl :-

- (i) It is used to display random ads. The ads information can be stored in XML file or in a database table.
- (ii) XML attributes -
- ImageURL
 - NavigateURL
 - AlternateText
 - Keyword
 - Impressions

Step 1:- Create an asp.net web application project, add XML file. Name the XML file as AdsData.xml.

write the following XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Advertisements>
<Ad>
<ImageURL>~/Images/Google.png
</ImageURL>
<NavigateURL>http://google.com
</NavigateURL>
```

Date : _____
26/

```
<AlternateText>Please visit  
http://www.Google.com</Alternate  
Text>  
<Impressions>10</Impressions>  
</Ad></Advertisements>
```

Step 2: Create a folder in Solution Explorer with a name Images and copy and paste 3 images (for eg - google.png, youtube.png)

Step 3: Drag and drop AdRotator control in web form set the property Advertisementfile = " ~ /AdsData.xml"

Step 4: To open the target webpage in a separate browser window, set Target = "-blank".

e) Multiview Control :-

- * FileUpload control in asp.net?
- is a combination of TextBox and BrowseButton that enable users to select a file to upload to the server.
 - Steps :-

Step 1:- Create a web application, add a webform into it.

Step 2:- Drag and drop the fileUpload control on the webform.

Step 3:- To upload the selected file, drag and drop Button control and change the following properties.

	Control	property	Value
1.	Button	- ID	btnUpload
2.		- Text	Upload
3.			

Step 4:- Also, drag and drop Label control and change the following properties.

	Control	property	value
1.	Label	- ID	objMessage
		- Text	"blank"
		- Font-Bold	true

Step5:- Right click on the web application project and add a folder with name "uploads". (This folder is going to store all the uploaded files.)

Step6:- Write the following code in btnUpload_Click() event handler.

btnUpload_Click()

{

// If the user has selected a file
if (fileUpload1.HasFile)

{

// upload the file .

fileUpload1.SaveAs(Server.MapPath
(" ~ /uploads / " + fileUpload1.FileName))

lblMessage.ForeColor = System.Drawing.

Color.Green;

lblMessage.Text = " File uploaded
successfully " ;

}

else

{

lblMessage.ForeColor = System.Drawing.
Color.Red;

lblMessage.Text = " Please select a
file " ;

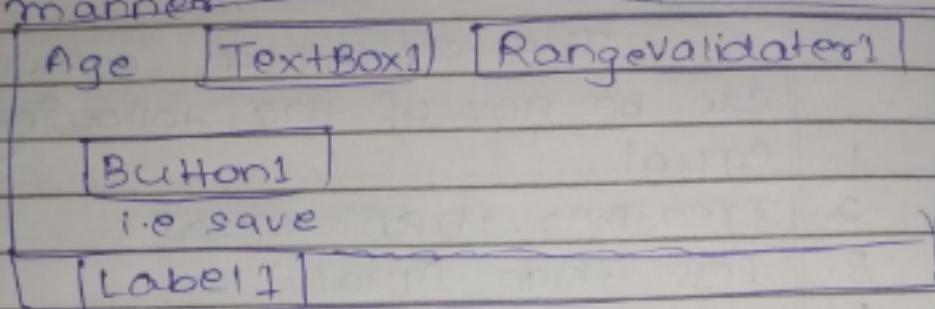
}

}

- * RangeValidator control in asp.net :-
- Is used to check if the value is within a specified range of values.
- Steps :-

Step 1:- Create a web application and add one webform.

Step 2:- Design a webform in a following manner



Step 3:- Change the following control properties

Control	Property	Value
1. TextBox	- ID	txtAge
2. Button	- ID - Text	btnSave Save
3. Label	- ID - Text	lblStatus "blank"
4. RangeValidator	- ID - ErrorMessage - ForeColor - ControlToValidate - MaximumValue - MinimumValue	RangeValidatorAge Age must be between 18 to 100 Red txtAge 100 18

- Type

Integer

27/11/19.

* CompareValidator Control :-

If is used to compare the value of 1 control with the value of another control or a constant value.

The comparison operation can be any of the following.

1. equal
2. greater than equal
3. less than equal
4. not equal
5. DataType check

step1:-

Password RequiredfieldV.

Rtype CompareValidator

Password RequiredFieldV.

Age CompareValidator

RequiredFieldV.

btsave → Button

lblMessage → Label

Step 2 - Change the following control property:

control	Property	Value
1. TextBox1	- ID - TextMode	txtPassword Password
2. TextBox2	- ID - TextMode	txtRePassword Password
3. Button	- ID - Text	btnSubmit Submit
4. Label1	- ID - Text	lblMessage "blank"
5. CompareValidator1	- ID - Error Message - ControlToValidate - ControlToCompare - Type - Operator	CompareValidator Password and Retype Password must match Red txtRePassword txtPassword string equal

6. TextBox3	- ID	txtAge
7. CompareValidator2	- ID	CompareValidatorAge
	- ErrorMessage	Age must be a number
	- Forecolor	Red
	- Type	Integer
	- Operator	Data Type check
	- SetFocusOnError	True
• forall validation controls	- Display	Dynamic

28/11/19.

- * Multiview control :-
- Multiview is made up of multiple view controls and each view control in turn can have controls inside it.
- Steps :-

Step 1 :- Create a web application and add new web form by the name Multiview.aspx.

Step 2 :- Drag and Drop Multiview Control

on the webform. Inside a MultiView control, drag and drop three different view controls.

- Example :- we want to capture employee information on a step by step basis.

- (i) First capture employee personal details.
- (ii) Second capture employee contact details.
- (iii) Show summary for confirmation, upon confirmation, save the data on a new web page.

Control	Property	Value
1. MultiView1	- ID	MultiViewEmployee
2. View1	- ID	ViewPersonalDetails
3. View2	- ID	ViewContactDetails
4. View3	- ID	ViewSummary

Multiview Employee

View Personal Details

Step 1 - Personal Details

First name

Last name

Gender Male

Button ← Step 2 >>

View Contact Details

Step 2 - Contact Details

Email Address

Mobile
Contact No.

[< Step 1]

[Step 3 >>]

View Summary

Step 3 - Summary

Personal Details

First Name [lbl FirstName]

Last Name [lbl LastName]

Gender [lbl Gender]

Contact Details

Email Address [lbl Email]

Mobile Number [lbl Mobile]

[< Step 2]

[Submit >>]

Step 3:- Apply the ~~ApplyViewIndex~~ property for each button control to display the view. (This property is used to determine the view i.e. visible or active).

- for example :-

```
Button1_Click() {  
    MultiViewEmployee.ActiveViewIndex = 0;  
}
```

Step 4:- After entering all details submit the details and redirect to the new webpage i.e.

```
btnSubmit_Click() {  
    Response.Redirect("~/confirmation.aspx");
```

* MasterPages in Aspx :-

* ValidationSummary in Asp.net :-

It is used to display a summary of all validation errors occurred in a webpage, at one place.

In general, in a real time application it is very common to display a red star symbol-text next to the input control.

- where the error occurred and then detail error msg in the . . .
- properties :
 - 1. HeaderText - It displays the header text for the ValidationSummary control
Eg - userpage.
 - 2. DisplayMode - It can be list, bullet list, single paragraph.
 - 3. ShowMessageBox - whether to display a messagebox with all the validation errors. (Boolean - True/False) (to display error msg format)

* Navigation control in asp.net :

1. MenuControl :-

• properties :

- (i) Orientation - by default menucontrol is displayed in vertical design. Orientation if we want to change its orientation to horizontal, use "orientat" prop

- (ii) StaticDisplayLevels - This means only the first level is statically displayed. By default value is 1.

- (iii) DisplayAfter - To control the

(iii) Disappear after - To control the amount of time it takes for the dynamically displayed position of menu to disappear, use this property. This is in millisecs. - 500 half sec.

Eg - if you want the menu to disappear after 2 sec, you would set it to 2000 milisec

2. Treeview Control :

- (i) It is useful to display hierarchical data in a tree structure.
- (ii) The content for the tree view control can be specified directly in the control or the tree view control can be bound to
 - ① XML file ② Database file.
- (iii) A treeview control is a collection of tree-node objects.

Step1: Drag and drop treeview control from navigation section and edit the node object by using editor windows.

Step 2: To configure the look and feel of the tree-node, the following styles can be used

- (i) HoverNodeStyle -
- (ii) LeafNodeStyle -
- (iii) RootNodeStyle -

3. SiteMap pathfile control :-

- ① This control displays the navigation path.
- ② The navigation path is often called as BreadCrumb
 `

Step 1: Right click on the solution explorer & add sitemapfile and write the following code.

```
<siteMapNode url = "Home.aspx"  
title = "Home" description = "">  
<siteMapNode url = "Services.aspx"  
title = "Services" description = "">  
<siteMapNode url = "consulting.aspx"  
title = "Consulting"></siteMapNode>  
<siteMapNode url = "outsourcing.aspx"  
title = "Outsourcing"></siteMapNode>  
</siteMapNode>
```

```
<siteMapNode url = "Aboutus.aspx"  
title = "About us"></siteMapNode>
```

DATE / /

```
<siteMapNode url = "contact us.aspx"  
title = "contact us" ></siteMapNode>
```

Drag and drop SiteMapPath control
from navigation control section on
the respected webform.

5/12/19

PAGE NO.:

DATE: / /

* Web services in Asp.net.

7/10/19

2/1/20

* Asp.net calendar control properties.

(i) Caption - This is a string read/write property.

(ii) CaptionAlign - Use to align the caption.

(iii) DayHeaderStyle - Style properties that can be used to customize the look and feel of the Day Header in the calendar.

(iv) DayNameFormat - can be full, short, firstLetter, firstTwoLetters, shortest.

(v) DayStyle - Style properties that can be used to customize the look and feel of the date in the calendar.

- (vii) firstDayOfWeek - which day of the week is displayed first.
 - (viii) NextPrevFormat - Can be ShortMonth, FullMonth, CustomText
 - (ix) NextMonthText - The text to use for the next month button
 - (x) PrevMonthText - The text to use for the previous Month button.
 - (xi) SelectionMode - Can be Day, DayWeek, DayWeekMonth.
If the SelectionMode is set to Day, then the user can select only one day
 - (xii) IsPostBack
- * IsPostBack in Asp.net :-
It is a page level property, that can be used to determine whether the page is being loaded in response to a client post back, or if it is being loaded and accessed for the first time.

* Asp.net page lifecycles event:
Every time when a request is made for a web form, the following sequence of events occur

- (i) web application creates an instance of the requested web form.
- (ii) Processes the events of the webform.
- (iii) Generates the html, and sends the html back to the requested client.
- (iv) The webform gets destroyed and removed from the memory.

The following are some of the commonly used events in the lifecycle of Asp.net webform. These events are shown in order of occurrence, except for "Error" event, which occurs only if there is unhandled exception.

* Page Lifecycle event .

1. preInit :

As the name suggest this event happens just before page initialization event starts.

IsPostBack, IsCallback and IsCrossPagePostBack properties are set at this stage.

2. Init : Page init event occurs after the Init event of all the individual controls on the webform. Use this event to read or initialize control properties.

3. Init complete : This event get raised immediately after page initialization.

4. Preload : Happens just before the page load event . Eg - drag & drop button.

5. Load : Occurs before the load event of all the individual controls on that webform . Eg - when execution

6. ControlEvents : After the page load event, the control events like buttons click, dropdownlist, selectIndexChanged events are raised .

7. loadComplete : It is raised after the control event are handled
8. preRender : This event is raised just before the rendering stage of the page .
9. preRenderComplete - Raised immediately after preRender event .
10. Unload : Raised for each control & for the page .
At this stage the page is unloaded from the memory . Eg - debugging stop .
11. Error : This event occurs only if there is an unhandled exception . It can occur in any stage .

- * Asynchronous javascript and XML:
- This is a cross platform technology which speeds up response time
 - The Ajax server controls add script to the page which is executed and processed by the browser .
 - The control loadBox in Visual

studio contains a group of controls called 'Ajax extension'.

* Ajax Controls :-

* Script Manager Control :-

- (i) It is the most important control & must be present on the page for other controls to work.

- (ii) It has the basic structure syntax :

```
<asp: ScriptManager ID="ScriptManager1"  
runat="server" >  
</asp: ScriptManager >
```

* UpdatePanel Control : + no interface .

- (i) It is a container control & derives from the control class.

- (ii) It acts as a container for the child controls within it & does not have its own interface.

- (iii) for eg : If a button control is inside the update panel and it is clicked, only the controls within the update panel will be affected, the controls on the other part of the page will not be affected. This is called partial postback or the asynchronous postback .

	Property	Description
①	Content Template	It defines what appears when in the updatepanel when it's rendered
②	Triggers	defines the collection triggers object each corresponding to an event causing the panel to refresh automatically

2. UpdateProgress control :

- It provides a sort of postback on the browser while one or more updatepanel controls are being updated.
- for eg: while a user login or waits for a server response while performing some database oriented goal.
- It provides a visual acknowledgement like "loading page . . ." indicating the work is in progress.
- Syntax for updateProgress control is :

```

<asp:UpdateProgress ID="Update
Progress" runat="server">
<progressTemplate>
</progressTemplate>

```

</asp:UpdateProgress>

- Properties Description
- ① ProgressTemplate- Indicates the template displayed during an asynchronous postback which takes more time than the display after time.

- ② DisplayAfter gets & sets the time in milliseconds after which the progress template is displayed. the default is 500.

3. Timer Control :

- Timer control is used to initiate the postback automatically
- This could be done in following two ways:

(i) setting the triggers property of the update panel control.

Syntax: <triggers>

</asp:AsyncPostBackTrigger ControlID = "Event" Name = "id" />
</triggers>

(ii) Placing a timer control directly inside the update panel to act as a child control triggers. A single trigger can be triggered for multiple update panel.

gl120

- * Data Binding in asp.net
 - 1. GridView
 - 2. DetailsView
 - 3. FormView

* GridView control in Asp.net :-

- (i) It used to display whole table data on the webpage.
- (ii) In gridview each columns defines a field or title, by each rows represents a record or data.
Eg - id , name , etc .
- (iii) Gridview control displays data with rows and column wise , i.e we can display whole table in gridview and also display only required columns from table in gridview control.
- (iv) In gridview control , we can easily do sorting , paging , selection and inline editing .
Also we can perform editing and deleting operation of displayed data with gridview control .

(v) Syntax :

```
<asp:GridView ID="GridView1"  
runat="server">  
</asp:GridView>
```

* DetailsView in Asp.net :-

- (i) It is also a databound control
- (ii) DetailsView is used to display one row at a time.

Step 1: Create a table tblEmployee with 10 columns. Use the SQL script and create a table with sample data.

- Create table tblEmployee (
id int primary key identity(1,1),
FirstName nvarchar(50),
LastName nvarchar(50),
City nvarchar(20),
Gender nvarchar(10),
DateOfBirth DateTime,
Country nvarchar(10),
Salary int,
DateOfJoining DateTime,
MaritalStatus nvarchar(20)) .

- Insert into tblEmployee values ('Tom', 'Minst', 'Alexandria', 'Male', '10/10/1980', 'USA', 5000, '09/09/2003', 'Married').
- insert 4 more values.

Step 2: Drag and drop GridView control on a webform. (Here, we have to display only 3 columns i.e id, FirstName, City from tblEmployee table).

Step 3: Drag and drop DetailsView control on the same webform.

Step 4: Drag and drop 2 SqlDataSource controls.

Step 5: Configure SqlDataSource1 to retrieve [ID], [FirstName], [City] columns from tblEmployee table.

Step 6: Associate SqlDataSource1 with GridView1 control.

Step 7: Configure SqlDataSource2 to retrieve all columns from tblEmployee table. Add a WHERE clause to filter the rows based on the selected row in GridView1 control.

Step 8: Associate SqlDataSource2 with DetailsView1 control.

* Datasource Controls :-

- (1) A Datasource control interacts with the databound controls and hides the complex data-binding processes.
- (2) These are the tools that provide data to the data bound controls and support execution of operations like insertion, deletion, sorting and updates.
- (3) Each datasource control wraps a particular data provider relational database, XML documents and helps in:
 - (i) Managing connection.
 - (ii) Selecting data.
 - (iii) Managing presentation aspects like paging, etc.
 - (iv) Manipulating data.

1. SQLDataSource Control :-

- (i) It represents a connection to a relational database such as SQL server or oracle database or ODBC (open database connectivity).
- (ii) Connection to data is made through two important properties

i.e ConnectionString and ProviderName.

- Properties :-

Property group	Description
1. DeleteCommand , DeleteParameter , DeleteCommandType .	Gets or Sets the SQL statement , parameters and type for deleting rows in the underlying data .
2. FilterExpression , FilterParameters .	Gets or Sets the data filtering string and Parameters .
3. InsertCommand , InsertParameter , InsertCommandType .	Gets or sets the SQL statement , parameters and type for inserting rows in the underlying data .
4. SelectCommand , SelectParameter , SelectCommandType .	Gets or sets the SQL statement , parameters and type for retrieving rows in the underlying data .

5 UpdateCommand, Gets or sets the UpdateParameter, SQL statement, UpdateCommandType, parameter and type for updating rows in the underlying data

* Insert Data ~~into~~ Into Database

Step 1: Create a table into your database with the name customer_detaile primary key (ID)

Step 2: Create columns i.e ID, fname, lname, city → varchar(50)

Step 3: Create a webform and design it in the following way:

Insert Data into Database

First Name

Last Name

City

Submit

[Label]

sql Databas eSource

Step 4: Add a connection to ~~say~~
SqlDataSource control by configuring
the Datasource.

Step 5: Generate Sql connection, open
connection, creating sql query and
close the connection in ~~web~~
`WebForm1.aspx.cs`

Step 6: ~~using~~ also add namespaces i.e using
`System.Data;`
`using System.Data.SqlClient;`
`using System.Configuration;`

Step 7: `WebForm1.aspx.cs`
public partial class

{
 // Generate Connection
 SqlConnection con = new SqlConnection
 (ConfigurationManager.ConnectionStrings
 ["SYDBCConnectionString"].
 ConnectionString);

Page_Load ()
{ // open connection
con.Open ();
}

Button1_Click()

{

// Insertion of data

```
SqlCommand cmd = new SqlCommand  
("Insert into customer_details values  
('" + txtFname.Text + "' ,  
'" + txtLname.Text + "' ,  
'" + txtCity.Text + "')", con);
```

```
cmd.ExecuteNonQuery();
```

```
con.Close();
```

```
label1.Visible = true;
```

```
label1.Text = "Your Data stored  
successfully";
```

```
txtFname.Text = "" ;
```

```
txtLname.Text = "" ;
```

```
txtCity.Text = "" ;
```

```
}
```

```
}
```

* Definition of ADO.NET.

ADO.NET :-

(ActiveX Data Object)

1. Whenever we create a software of website, it is not possible to implement it using only programming languages. Regarding any software of website, there is always important data which we have to store permanently.
 2. Storing Data permanently is not the facility given by any programming languages. For this purpose we have to use Database. That means software or website is developed by the combination of programming language or database.
 3. ADO.NET is a model used by the .NET framework to communicate with database for retrieving or storing data with the help of various built-in classes.
 4. There are several different types of databases available i.e
- ✓ (i) Microsoft SQL Server

- (ii) Microsoft Access
- (iii) Oracle
- (iv) Borland Interbase
- (v) IBM DB2 etc.

* Objects of ADO.net :-

Objects of ADO.net

- 1) The connection object
- 2) The command object
- 3) The DataReader object
- 4) The Dataset object
- 5) The DataAdapter object

- ### 1. The Connection Object :
- To communicate or interact with a database, you must have a connection with a database.
 - The connection helps out to recognize the database server.

The Database name, username, Password and other parameters, that are required for connecting to the database.

- A connection object is used by command to know for which database to execute the command.

2. The Command Object :

- Interacting with the database does not mean only creating a connection; it means you must stay state the actions that you want to occur. This can be done with the help of command object. The command object can be used to send SQL statements to the database.
- A command object makes use of connection object to tell which database to communicate with. The command object can be used alone, to execute a command directly.

B. The DataReader Object :

- It is a stream based, read only forward only retrieval of query results from the Datasource, which never update the data.
- A connection object can contain only one data reader at a time.
- While data is being accessed, the connection in the data reader remains open (i.e. `open();`) and cannot be used for any other purpose.

4. Dataset Object :

- It is used to store the data of database in application. It represents a collection of data retrieved from the Datasource.
- Dataset is a tabular representation of data means it represents data into row and column format.
- We can use Dataset in combination with DataAdapter class as the Dataset object work in disconnected database architecture. It provides a better advantage over DataReader.

5. The DataAdapter object:
- It provides the communication between the Dataset and Datasource.
 - Data Adapter can be used in combination with the Dataset object so that two objects enable both Data retrieval and Data manipulation capabilities.
 - The DataAdapter is also used to reflect changes made to the Dataset back to the Datasource.

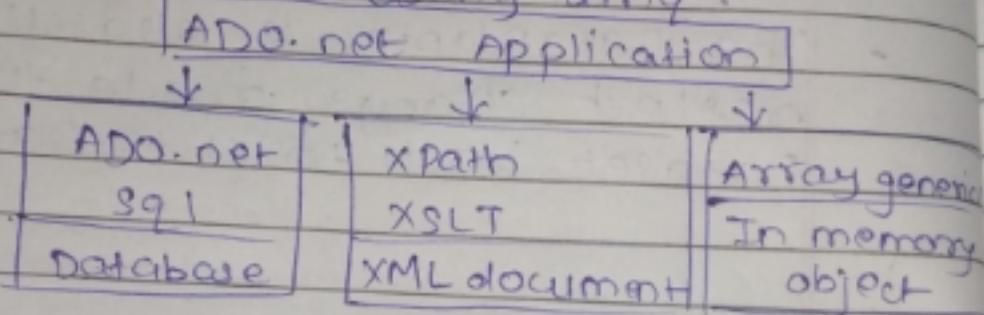
* State management in Asp.net;

* Linq :

- Definition :-

- Linq stands for language ^{query} integrated
- Linq enable us to query any type of Data store (sql server, xml document, object string memory, etc)

- Benefit of using linq :-



If the .net application that is being developed

- a) requires data from Sql server :- Then the developer has to understand ADO.net code & sql specific to Sql server database.
- b) requires dat from XML document :- Then the developer has to understand XSLT & XPath queries
- c) need to query objects in Memory. (`list < customer >, list < order >, etc`) Then the developer has to understand how to work with objects in memory.

- linq enables us to work with this different datasources using a similar coding style without having the need to known the syntax specific to the datasource.

linc provide intelligence & compile time error checking.

- linq query can be written using any .net supported programming language.

Eg: querying the customers table using linq in c#, the code would be

```
var data = from c in dataContext  
           .customers  
           where c.Country = "Spain";  
           select c;
```

- (i) The 'from' keyword logically loops through the contents of the collection.
- (ii) The expression with the 'where' keyword is evaluated for each object in the collection.

Select' the select statement selecting
the evaluated object to add to
the list being return.

Step 1: Select sys database from SQL
server management studio 2013

Step 2: click on new query and write
create table students
(

Id int primary key Identity step
first_name varchar (20),
last_name varchar (20),
Gender varchar (20)

)

- execute .

Step 3: Insert into students table name
values
('Mark', 'xyz', 'Male')

Insert 5 records where 1 record
must be for female .

- execute .

Step 4: Create a empty web application and name it as Demo →
Click on View menu item and select "server explorer" →
in server explorer window, right click on "data connection".
Select add connection option.

- Adding linq to sql database

Step 5: Adding linq to sql classes
(i) Right click on demo project in solution explorer and select add new item → Data → linq to sql classes → Name as 'sample.dbms' → add .

From server explorer window, drag and drop "Students-table" on to sample.dbms designer table .

Add a webform and drag and drop review control .

Sample.aspx.cs .

- * State Management in Asp.net :-
- * Introduction -
 - Hypertext transfer protocol is a stateless protocol. When the client disconnects from the server, the asp.net engine discards the page object. This way, each web application can still scale up to the server numerous request simultaneously without running out of server memory.
 - However there needs to be some technique to store the information between request and to retrieve it when required. This information i.e. the current value all the controls and variables for the current user in the current session is called State.
 - Asp.net manages three types of states :
 - (i) ViewState
 - (ii) Session State
 - (iii) Application State

1. ViewState :-

- The ViewState is the state of the page and all its controls. It is automatically maintained across posts by the asp.net framework.
- When a page is sent back to the client, the changes in the properties of the page and its controls are determined and stored in the value of hidden input field named as "VIEWSTATE".

When the page is again posted back, "VIEWSTATE" field is send to the server with the http request.

- Example :

Step 1: Create asp.net empty web application → add new web form → name it as Viewstate1.aspx -

Step 2: Drag & drop a button & textbox control on to the web form → double click on the button.

Step 3: Now add another webform to the project → name it as Viewstate2.aspx → drag & drop textbox and button control on to this webform as well.

Step 4: write the following code in Viewstate1.aspx.cs and Viewstate2.aspx.cs

Page_Load()

{

if (!IsPostBack)

{

if (ViewState["clicks"] == null)

{

ViewState["clicks"] = 0;

}

TextBox1.Text = ViewState["clicks"].ToString();

}

}

Button1_Click()

{

int ClicksCount = (int)ViewState["clicks"] + 1;

TextBox1.Text = ClicksCount.ToString();

ViewState["clicks"] = ClicksCount;

}

Step5: Now run the application and navigate to Viewstate1.aspx

Click the button control. Every time you click the button, the ClickCount get incremented and is displayed in the textbox as expected.

Now, Navigate to Viewstate2.aspx
Click the button control. Notice that the value starts from 0, each page has its own ViewState["Clicks"].

2. SessionState:-

Note: ViewState of a webform is available only within that webform, by default.

2. SessionState :-

- When the user connects to an asp.net website, HttpSession object is created.
- When session state is formed - time on, a new session object is created for each new request.

This session state object becomes part of the context and is available through the page.

Example :-

Step 1: Add a new web-form with name SessionState1.aspx → Drag and drop button and textbox control.

Step 2: Do the same thing by adding a page with name SessionState2.aspx

Step 3: Write the following code in SessionState1.aspx.cs and SessionState2.aspx.cs.

Page_Load()

{

if (!IsPostBack)

{

if (Session["clicks"] == null)

{

Session["clicks"] = 0;

}

TextBox1.Text = Session["clicks"].ToString();

}

Button1_Click()

{

int ClicksCount = (int)Session["Clicks"]
+ 1;

TextBox1.Text = ClicksCount.ToString();
Session["Clicks"] = ClicksCount;

Step4: Add the following sessionstate element to web.config file, under the system.web . This setting specifies the webapplication to use cookieless sessions.

<sessionState mode="InProc"

<sessionState mode="InProc"

cookieless="true" >

</sessionState >

Step5: Run the application and navigate to SessionState1.aspx.

Click the button 3 times and notice that the value 3 is displayed in the TextBox.

Now navigate to SessionState2.aspx , the value 3 is displayed in the textbox on SessionState2.aspx.

3. ApplicationState :-

- The asp.net application is collection of all webpages, code and other files within a single virtual ^{directory} on a web-server.
- When information is stored in ApplicationState, it is available to all the users.
- To provide for the use of applicationstate, asp.net creates an applicationstate object for each application from the httpApplication state class and stores this object in server memory.

Example :

Step 1: Add a new webform with name "ApplicationState1.aspx". Drag and drop a button and textbox control onto ApplicationState1.aspx.

Step 2: Do the same thing by adding a page with name "ApplicationState2.aspx".

Step3: write the following code in "ApplicationState1.aspx.cs" and "ApplicationState2.aspx.cs".

Page - load ()

{

if (! IsPostBack)

{

if (Application ["Clicks"] == null)

{

Application ["Clicks"] = 0 ;

}

TextBox1 . Text = Application ["Clicks"] . ToString ();

}

Button1 - click ()

{

int ClickCount = (int) Application ["Clicks"]

+ 1 ;

TextBox1 . Text = ClickCount . ToString ();

Application ["Clicks"] = ClickCount ;

}

Step4: Run the application and navigate to ApplicationState1.aspx

Click the button 3 times and notice that the value 3 is displayed in the TextBox. Now navigate to ApplicationState2.aspx, the value 3 is displayed in TextBox of ApplicationState2.aspx. This proves that an Application-State variable is accessible across all pages in a web-application.

Now open a new browser window and navigate to ApplicationState1.aspx. Notice that the value in the TextBox is still 3.

* Difference between ViewState, SessionState and ApplicationState

1. ViewState :

- ViewState of a webform is available only within that webform.
- ViewState is stored on the page, in a hidden field called as 'Viewstate'. Because of this, the ViewState will be lost, if you navigate away.

from the page , or if the browser is closed .

- Viewstate is used by all asp.net controls to retain their state across postback .

2. Session State :

- Session State Variables are available across all pages , but only for a given single session . Session Variables are like single user , - global data .
- Session State Variables are stored on the webserver .
- Session State Variables are cleared , when the user session times out . The default is 20 minutes . This is Configurable in " web.config " .

3. Application State :

- Application State Variables are available across all pages and across all sessions . Application State Variables are like multi-user global data .
- Application State Variables are stored in the webserver .

- Application state variables are cleared, when the process hosting the application is restarted.

- Asp.net server control events:-
 - Asp.net server controls such as TextBox, Button and DropDownList has their own events.
 - For example a Button has a click event, TextBox has TextChanged event and DropDownList has SelectIndexChanged event.
 - We have a set of asp.net validation controls that has validation events. The events that all these controls exposed, can be broadly divided into three categories
 - (i) PostBack events - click event of a button control is an example of PostBack event.
 - (ii) Cached events - These events are saved in the PageViewState to be processed when a PostBack event occurs for example : TextChanged event of a TextBox control and SelectIndexChanged event of a DropDownList control.
 - (iii) Validation events - These events occur on the client, before the page is posted back to the server. All validation Controls uses these types of events.

25/11/20
protected void Page_PreInit()

{
Response . write (" Page-Init " +
"
 ");
}

Page_Init()

{

Response . write (Page-Init" +
"
 ");
}

{

Page-InitComplete()

{

Response . write (" Page-Initcomplete"
+ "
 ");
}

{

Page-PreLoad()

{

Response . write (" Page-PreLoad" +
"
 ");
}

{

Page_LoadComplete()

{

Response . write (" Page_LoadComplete"
+ "
 ");
}

{

Page_PrefRender()

{

Response . write (" Page_PrefRender" +
"
 ");
}

{

Page - PreRenderComplete ()

{

Response.write ("Page - PreRenderComplete"
+ "
");

}

TextBox1_TextChanged ()

{

Response.write ("Text changed Event"
+ "
");

}

Button1_Click ()

{

Response.write ("Button click" +
"
");

}

Step1: Add a web form with TextBox,
RequiredFieldValidator and a
Button control.

Step2: Double click the TextBox control and
the event handler TextBox1_
TextChanged() will be automatically
generated. Also double click the
button control, Button1_Click()
event handler will be generated

Step 3: Change the property of RequiredFieldValidator control to validate to TextBox1.

Step 4: Now run the project and when the webform renders, the page level event occur in the following order:

1. Page - PreInit
2. Page - Init
3. Page - InitComplete
4. Page - PreLoad
5. Page - LoadComplete
6. Page - PreRender
7. Page - PreRenderComplete

Notice that TextChanged and Button click events are not fired.

Step 5: Don't type anything in TextBox and click the button control.

The RequiredFieldValidator error message is displayed.

No other events get processed and the page is not posted back to the server.

Step 6: Now enter some text into the TextBox and click the button. TextChanged event and Button click event, execute after PageLoad and before PageLoadComplete event. The execution order is shown below :

1. Page - PreInit
2. Page - Init
3. Page - InitComplete
4. Page - PreLoad
5. Text changed Event
6. Button Click
7. Page - LoadComplete
8. Page - PreRender
9. Page - PreRenderComplete

* IsPostBack :-

- IsPostBack is a Page Level property, that can be used to determine whether the page is been loaded in response to a client PostBack or if it is been loaded and accessed for the first time.
- In Real Time, there are many situation where IsPostBack property is used.

8- for eg. :-

Consider the web form used to register Employee details. The form has FirstName, LastName and city fields.

Employee details form

First Name :

Last Name :

City ▾

Register Employee

Write the following code in
Code behind file of the
web form.

Page_Load()

{

LoadCityDropDownList();

}

public void LoadCityDropDownList()

{

List li1 = new List(
 "London")

ddlCity.Items.Add(li1);

List li2 = new List(
 "Sydney");

ddlCity.Items.Add(li2);

```
    listItem li3 = new listItem("Munich");
    addCity.Items.Add(li3);
}
Button1_Click()
```

Run the application. The cities are currently shown as expected. Just click the button one more time. Notice that the city name in the dropdownlist are duplicated.

- Note:
1. Asp.net server control retain their state across post back. These controls make use of viewstate. So, the first time when the web form load, the cities get correctly added to the dropdownlist and send back to the client.
 2. Now, when the client clicks the button control and the web form is posted back to the server for processing. During the page initialization, View state restoration happens.

~~get~~ ~~100~~

SOAP: Simple Object Access Protocol.
WSDL: Web service description language.

Page No.:	
Date:	/ /

3 How to solve dropdown list items duplication.

- (i) one of the best way to do this, is to use IsPostBack property . so, in the Page_Load, call LoadCityDropDownList(); if the request is not a Post back request i.e only if the web form is being loaded and accessed for the first time.

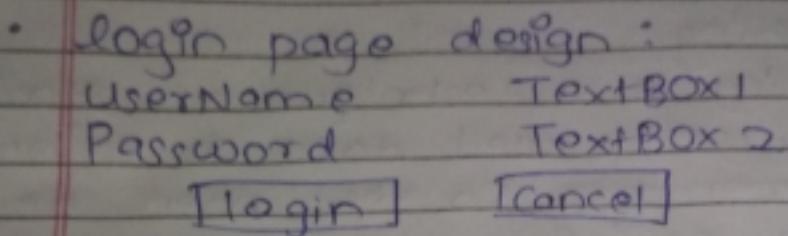
* Web Services :

1. what are web services and why should we use them ?

28/1/20

- x Create a web form registration where accept the value from user that is username, date of birth, email, city, state, contact no, password, confirm password, security question, accept the date save it in 'Registration details' table. After successful registration display a label with 'Data saved successfully' message.
- Apply appropriate validation controls

Username	TextBox1	RFV 1
Date of birth	" 2	RFV 2
Email	" 3	RFV 3 REV
State ↪	DDL1	RFV 5
City ↪	DDL2	RFV 4
Contact no	TextBox4	RFV 6
Password	" 5	RFV 7
Confirm Password	" 6	RFV 8 CVI
Security ↪	" 7	RFV 10
Security Ques ↪	DDL3	RFV 9



hyperlink → forgot password ?

- forgot Password webform :
- ① Enter Email Text Box 1
- ② Enter security ques DD 1

Answer
 Reset password →

Ans is not match ← label

Answer is match then webform
 will open Ret.

Retype Password TB1

Confirm retype password TB2 CV1

Password changed successfully

↑
 message box

1/2/20

* Caching

- Caching is a technique of storing frequently used data or information in memory, so that when the same data or information is needed next time, it could be directly retrieved from the memory instead of generated by the application.
- Caching is extremely important for performance boosting in Asp.net, as the pages and controls are dynamically generated.
- It is especially important for ~~data related~~ transaction as there are expensive in terms of response time.
- Caching places frequently used data in quickly accessed media such as the random access memory of a computer.
- Caching improves the performance and scalability of an application.

1. Output Caching :

Step 1: Create tblProducts table in sql server.

```
Create Table tblProducts
(
```

```
[id] int identity primary key,
[name] nvarchar(50),
[Description] nvarchar(250)
)
```

Popular tblProducts with
Sample data

```
Insert into tblProducts value
('Laptop', 'Dell Laptop')
```

```
Insert into tblProducts value
('iPhone', 'iPhone 4S')
```

```
Insert into tblProducts value
('LCD TV', 'Samsung LCD TV')
```

```
Insert into tblProducts value
('Desktop', 'HP Desktop Computer')
```

Step 2: Create spGetProducts stored
procedure.

Create Procedure spcretProducts

As

Begin

Wait For Delay '00:00:05'

Select ID , Name , Description

From tblProducts

End .

Execute spcretProducts

Step 3: Let us the stored procedure in an asp.net application and display tblProducts data in a GridView Control .

Drag and drop a GridView control onto the web form and write the following code in the code behind file under Page_Load () event .

Sql @

```
SqlConnection con = new SqlConnection  
("Data Source = ADMIN; Integrated  
Security = SSPI; initial catalog =  
SYNEW");
```

```
Page_Load () :
```

```
{
```

```
SqlDataAdapter da = new SqlDataAdapter  
("SpCreateProducts", con);  
da.SelectCommand.CommandType =  
CommandType.StoredProcedure;
```

```
DataSet DS = new DataSet();  
da.Fill(DS);
```

```
GridView1.DataSource = DS;
```

```
GridView1.DataBind();
```

```
Label1.Text = DateTime.Now.  
ToString();
```

```
}
```

Also make sure you have the following "using" declaration.

```
using System.Data;
```

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

also add a client time in a script tag that is:

Client time :

```
<script>
```

```
document.write (date());
```

```
</script>
```

C	C

→ GridView1

Server time : [Label1] → text = " - Baner"

Client time :

Step 4: Run your application the page takes 5 seconds to load this is because when you request the webform, the webserver have to process the web-form events, execute the stored procedure, create objects, generate HTML and send that HTML to the client browser.

Step 5: To cache a webform use the `@OutputCache` page directive, the `@OutputCache` directive has mandatory attributes.

- ① Duration specifies the time in seconds for which the webform should be cached.
- ② VaryByParam - Cache multiple responses of a single webform.

webform with following "Output Cache" directive is cached for 30 seconds.

```
<? @OutputCache Duration="30"  
VaryByParam="None" ? >
```

Step 6: Run the application.

Note: When any user request this webform for the first time, the webserver will process the webform event, execute the stored procedure, Create objects, generate html & send that html to client browser and retains a copy of the response in memory for the next 30 sec.

Any subsequent request during that time receive the cache response.

After the cache duration has expired, the next request for the webform, has to process WebForm events, execute the stored procedure, create objects, generate HTML, which is then cached for another 30 secs.

so this webform is processed by the server, once every 30 secs at the most.