Chapter- Five
Introduction to Advanced Server Side Issues

FORM HANDLING

- A form is a section of an HTML document where you put user-input controls, like text boxes, check boxes, radio buttons, and pull-down lists. You use forms when you want to collect and process user input.
- ASP.NET provides important feature event handling to Web Forms. It let us to implement event-based model for our application. As a simple example, we can add a button to an ASP.NET Web Forms page and then write an event handler for the button's click event. ASP.NET Web Forms allows events on both client and server sides.
- In ASP.NET Web Forms pages, however, events associated with server controls originate on the client but are handled on the Web server by the ASP.NET.
- ASP.NET Web Forms follow a standard .NET Framework pattern for event-handler methods. All events pass two arguments: an object representing the object that raised the event, and an event object containing any event-specific information.

ASP.Net Web Form Features

- ASP.NET has multiple features and provides many tools to create and develop web applications. Here are some of the features of web forms:
  i. Server Controls
  ii. Master Pages
  iii. Working with data
  iv. Membership
  v. Client Script and Client Frameworks
  vi. Routing
  vii. State Management
  viii. Security
  ix. Performance
  x. Error Handling

i. Server Control

- It provides a vast set of server controls. These controls are like objects, and they run when they are requested and rendered to the browser. Some web

pages are similar to HTML elements like text-box, button, checkbox, and hyperlink.

ii. Master Pages

☞ Mater Pages is responsible for the consistent layout of our web applications. It gives a proper appearance and standard to different pages.

iii. Routing

☞ URL routing can be configured to a web application. A request URL is a URL that a user enters in a browser to browse in a specific place.

iv. Security

☞ Security always plays a crucial role in software development. ASP.NET provides different configuration options and extensibility points to make our systems more secure.

ASP .NET FORM LABEL

☞ This control is used to display textual information on the web forms.
☞ It is mainly used to create caption for the other controls like: textbox.
☞ This is server side control, asp provides own tag to create label.
Example:
<asp: labelID = "label1" runat= "server" Text = "label1"></asp: label>

Property	Description
Access Key	It is used to set a keyboard shortcut for the label.
Tab Index	The tab order of the control.
Back Color	It is used to set the background color of the label.
Border Color	It is used to set border color of the label.
Border width	It is used to set the width of the border of the label.
Font	It is used to set the font for the label text.
Fore Color	It is used to set the color of the label text.
Text	It is used to set text to be shown for the label.
ToolTip	It displays the text when the mouse is over the label.
Visible	To set the visibility of control on the form
Height	It is used to set the height of the control.
Width	It is used to set the width of the control.

ASP .Net WEB FORM TEXTBOX

☞ This is an input control which is used to take user input.
☞ To create a textbox either we can write a code or use the drag and drop facility of visual studio IDE.
☞ This is server side control, run asp provides own tag to create it.

<asp:TextBoxID= “TextBox1” runat= “server”></asp:TextBox>

Property	Description
Access Key	It is used to set keyboard shortcut for the control.
Tab Index	The tab order of the control.
Back Color	It is used to set the background color of the control.
Border Color	It is used to set the border color of the control.
Border Width	It is used to set width of border of the control.
Font	It is used to set font for the control text.
Fore Color	It is used to set color of the control.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when mouse is over the control.
Visible	To set the visibility of control on the form.
Height	It is used to set height of the control.
Width	It is used to set width of the control
Max length	It is used to set maximum number of characters that can be entered.
Read Only	It is used to make control read only.

ASP .NET DROP DOWN LIST

- The DropDownList is a web server control which is used to create an HTML Select component.
- It allows us to select an option from the dropdown list.
- It can contain any number of items ASP.Net provides a tag to crate DropDownList for web application.

Example:

<asp:DropDownlist id= “DropDownlist1” runat = “server” DataSource = “<% databindingexpression %>”

DataTextField = “DataSourceField”

DataValueField = “DataSourceField”

AutoPostBack = “True | False”

OnSelectedIndexChanged = “OnSelectedIndexChangedMethod”>

<asp: ListItem Value = “value” selected = “True | False”>

Text

</asp: ListItem>

</asp: DropDownList>

CODE:

```
protected void Button1_click(object sender, EventArgs e)
{
        if(DropDownlist1.SelectedValue == " ")
        {
                Label1.Text = "Please Select a City";
        }
        else
        {
                Label1.Text = " Your Choice is:" + DropDownList1.SelectedValue;
        }
}
```

ASP .NET WEB FORMS RADIO BUTTON

- ☞ It is an input control which is used to take input from the user. It allows user to select a choice from the group of choices.

<asp: RadioButtonID = "RadioButton1" runat = "server" Text = "Male" GroupName = "Gender"/>

Property	Description
Access key	It is used to set a keyboard shortcut for the control.
TabIndex	It is used to set Tab order of the control.
BackColor	It is used to set the background color of the control.
BorderColor	It is used to set the border color of the control.
BorderWidth	It is used to set the width of the border of the control.
Font	It is used to set the font for the control text.
ForeColor	It is used to set the color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It display the text when the mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set the height of the control.
Width	It is used to set the width of the control.

GroupName	It is used to set the name of the radio button group.

FORM CHECKBOX

- Form CheckBox is used to get multiple inputs from the user.
- It allows user to select choices from the set of choices.
- It takes user input in yes or no format.
- It is useful when we want multiple choices from the user.

Property	Description
Access Key	It is used to set a keyboard shortcut for the control.
Tab Index	The tab order of the control
Back Color	It is used to set the background color of the control.
Border Color	It is used to set the border color of the control.
Border Width	It is used to set the border width of the control
Font	It is used to set the font for the control text.
Fore Color	It is used to set the color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when the mouse is over the control
Visible	To set visibility of control on the form
Height	It is used to set height of the control
Width	It is used to set width of the control.
Checked	It is used to set check state of the control either true or false.

Example:

<asp: checkbox ID = "CheckBox1" runat = "server" Text = "MVC" />

<asp: checkbox ID = "CheckBox2" runat = "server" Text = "CORE" />

<asp: checkbox ID = "CheckBox3" runat = "server" Text = "ASP.NET" />

ASP .NET FORM BOTTON

- This control is used to perform events. It is also used to submit client request to the server.

Property	Descriptions
Access Key	It is used to set a keyboard shortcut for the control.
Tab Index	It is used to set Tab order of the control.
Back Color	It is used to set the background color of the control.

Border Color	It is used to set the background color of the control.
Border Width	It is used to set the width of the border of the control.
Font	It is used to set the font for the control text.
Fore Color	It is used to set the color of the control text.
Text	It is used to set text to be shown for the control.
ToolTip	It displays the text when the mouse is over the control.
Visible	To set visibility of control on the form.
Height	It is used to set the height of the control.
Width	It is used to set the width of the control.

Example:

<asp: ButtonID = "Button1" runat= "server" Text = "Submit" BorderStyle = "Solid" ToolTip = "Submit"/>

<asp: Button ID = "Button1" runat = "server" Text = "Click here" OnClick = "Button1_Click"/>

DATABASE CONNECTIVITY

- A database connection is a facility in computer science that allows client software to communicate with database server software, whether on the same machine or not. A connection is required to send commands and receive answers.

ADO.NET

- ADO.NET is a module of .Net Framework which is used to establish a connection between application and data sources. Data sources can be such as SQL Server and XML. ADO.NET consist of classes that can be used to connect, retrieve, insert and delete data.

ADO.NET CLASSES

- Some important ADO.NET objects that are responsible for the CRUD operations is as follows:
  - **DataSet:**
    Think about DataSet as a copy of a database stored in server's memory. It is used only for querying multiple SQL tables at once.
  - **SqlDataReader:**
    It is used for querying data from a single SQL table.

- **DataTable**
  Data Table is a sub item of a DataSet and represents a database table stored in the memory.
- **SqlConnection**
  Object responsible with storing the data.
- **SqlCommand**
  Object responsible with sending the SQL query to the server and returning the results.
- **SqlDataAdapter**
  SqlDataAdapter is responsible with filling a DataSet with the data returned from the database.
- **DataReader**
  This retrieve data in forward only and read only form.
- **DataAdapter**
  This acts as a bridge between dataset and data source to load the dataset and reconcile changes made in dataset back to the source.

Web.config file with the new connection string

Syntax:

<connectionStrings>

<add name = "your connectionStringName" connectionString = "Data Source = DatabaseServerName; Integrated Security = true; Internal Catalog = YouDatabaseName; uid= YourUserName; Password=yourpassword;" providerName= "System.Data.SqlClient"/>

</connectionStrings>

Example:

<connectionStrings>

<add name = "myconnection" connectionString = " Data Source = miniproject; Integreated Security = true; Internal Catalog = MyWebDatabase" providerName = "System.Data.SqlClient"/>

</connectionStrings>

Database Connection into .cs file:

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["connectionstrings"].ToString());

Example:

SqlConnection con = new SqlConnection(ConfigruationManager.ConnectionStrings["myconnection"].ToString());

Here **myconnection** is connection string that we create into Web.config file.

Creating SQL Statements

1. **How to create database**
   CREATE DATABASE database_neme;
   USE Database_name;

   Eg. CREATE DATABASE imsdb;
   USE imsdb;
2. **Dropping the Database (Deleting the database)**
   Syntax: DROP DATABASE database_name;
   Eg. DROP DATABASE imsdb;
3. **How to create Table**
   CREATE TABLE Table_name
   (
   ID INT NOT NULL IDENTITY(1,1),
   FirstName VARCHAR(50),
   SecondName VARCHAR(50)
)
4. **Adding Column on Table**
   Syntax: ALTER TABLE table_name
   ADD column_name datatype;
   Eg. ALTER TABLE tbl_user
   ADD Email varchar(255);

5. **Deleting Column from Table**
   Syntax: ALTER TABLE table_name
   DROP COLUMN column_name;

Eg. ALTER TABLE tbl_user
DROP COLUMN Email;

6. **Modifying Column (changing the data type of a column in a table)**
Syntax: ALTER TABLE table_name
MODIFY COLUMN column_name datatype;
Eg. ALTER TABLE tbl_user
MODIFY COLUMN class int;
7. **Deleting Table**
Syntax: DROP TABLE table_name;
Eg. DROP TABLE tbl_user;

SQL Statement to insert, select, update and delete data

8. **Inserting data into table**
Syntax: INSERT INTO table_name(column1, column2, column3….)VALUES(value1, value2, value3,…….);
Eg. INSERT INTO tbl_user(uid, uname, upassword)VALUES(01,'Saroj', 'singh');

9. **SELECT QUERY**
SELECT * FROM Table_name;
Eg. SELECT * FROM tbl_user;

10. **Selecting data using condition**
Syntax: SELECT column1, column2,…
FROM table_name
WHERE condition;
Eg. SELECT uname, upassword FROM tbl_user WHERE uid=2;

11. **Updating data (modifying the existing record in a table)**

Syntax: UPDATE table_name
SET column1=value1 , column2=value2
WHERE condition;
[:. WHERE condition is optional]

Eg. UPDATE tbl_user SET uname='Ram'
WHERE uid=01;

12. **Deleting data**

Syntax: DELETE FROM table_name WHERE condition;

Eg. DELETE FROM tbl_user WHERE uname='Ram';