



Unit-3

.Net Using C#
Sem-5

Validation Control

- Form Validation
- Validates the user input data to ensure that only correct authentication data get stored.
- Ex. Name, Username, Email-id, Mobile no, Gender
- Validation Work Form-> User Input -> Validation Rules-> Correct data.

Validation Control

- Validation control are used to validate the user input. The validation control can be attached to input control to check the values that are entered by user.
- For ex. You can use validation control to check whether user has entered a password contains a minimum of 4 digit and maximum 20 characters the entered password is combination of later and numbers..

- **Server side Validation:-** Many protocols www,http => Web used internet connectivity => Runs for web browser. => Web application used diffraints user accessing the program. => speed slow => web 24*7 hrs.
- **Client side validation:-** used on application layes => no need to use internet connectivity. => run is computer based application run through o.s. => access by the particular computer. => speed is fast => not available in 24*7




There are Six Types of Validation Control

- Required Field Validator
- Regular Expression Validator
- Compare Validator
- Range Validator
- Custom Validator
- Validation Summary

Required Field Validator

- It is used to check whether a server control added to a web form has a value or not.
- Ex. You can use this validation control to check whether a user has entered any value in the password textbox.
- This validator properties is **Control To Validate, Error Message, Text.**
- Control to validate:- This properties is used to specify the control to be validated.

- 
- **Error Message:-** This properties is used to specify the error message displayed when the validation condition fetched.
 - **Text:-** This properties is used to specify the error message displayed by the control.

Regular Expression Validator

- It is used to check whether a server control added to web form matches with a specific regular expression or not.
- The regular expression can be the format of a telephone number or email-id address.
- This validator properties is **ControlToValidate, ErrorMessage, Text**.
- This property is used to specify the regular expression when performing validation.
- Ex. `[\w\w\=-] + @ +[a-z] [z-a] $`

Compare Validator

- It is used to compare the entered value with another value the other value can be a number or a value entered into another control.
- This validator properties is **Control To Validate, Control To Compare, Error Message, Text.**
- Control to validate:- This properties is used to specify the control to be validated.
- Control to compare:- use to specify id of the control that will be used to compare values.

Range Validator

- It is used to check whether the value of a particular web form field is within the range of minimum value and maximum value.
- The minimum value can be date, number, currency amount or strings.
- This validator properties is **Control To Validate, Error Message, Minimum value, maximum value.**
- Minimum value:- used to specify the minimum value.
- Maximum value:- used to specify the maximum value.

Custom Validator


- It is used to perform user define validation that cannot be perform by standard validation controls.
- This validator properties is **Control To Validate, Error Message.**
- Method:- OnServerValidate:- Use to raise the server validate event.
- Server validate is commonly use event of custom validator. You can write event handler for server validate event and perform the required validation.
- Custom validation is more secure.

Validation Summary

- It is used to summarize all errors and display list in a user specified location on the web form.
- Properties:- **HeadText**- it is used to set the text display at the top of the summary.
- ShowMessageBox:- Used to display the error message in a pop of msgbox when the value of this property is true.
- Showsummary:- Used to enable or disable the summary of the messages.

Navigation Control

- Navigation controls are very important for websites.
- Navigation controls are basically used to navigate the user through webpage.
- It is more helpful for making the navigation of pages.
- The user interface for navigation can be a simple, static, hyperlink, or involve the use of trees and menus.



There are three controls in ASP.NET ,Which are used for Navigation on the webpage.

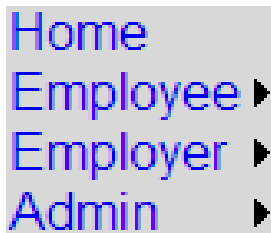
- Menu Control
- Tree View Control
- Sitemaps Control

Menu Control

- The menu control is another rich control that supports hierarchical data to try the menu control.
- Menu control is also used to display site navigation information. The menu control display the site structure vertically as well as horizontally.
- It can also retrieve the data to be display from the item that are added to the items collection of the menu control at runtime.
- The menu control consists of one or more menu items properties display at different levels of hierarchical.

Menu Control


- The menu control provides an overwhelming number of styles. Like the tree view the menu adds a custom style class. Which is name menu styles.
- This style adds spacing properties such as ItemSpacing, HorizontalPadding, and varticalPadding.
- Static that main the root level items that are displayed in the page when it is first generated.
- Dynamic that main that are added when the user moves the mouse over a portion of the menu.



Home
Employee ►
Employer ►
Admin ►

Tree View Control

- The tree view at work for displaying navigational information.
- The tree view can show a portion of the full site map or the entire site map. Each node becomes a link that when clicked takes the user to the new page.
- The tree view is one of the most impressive controls in asp.net. Not only you show site maps.
- The tree view has properties that let you change how its displayed on the page.

- 
- There are some properties that are useful for defining the behavior of the TreeView.
 - **ShowCheckBox:** This property is used to set whether the checkbox is required or not.
 - **ShowExpand:** Hides the expand collapse boxes when set to false.
 - **ShowLines:** Adds lines that connect every node when set to true.
 - **NodeWrap:-** Lets a node text wrap over more than one lines when set to true.
 - **MaxDataBindDepth:-** Determines how many levels the tree view will show. By default maxdatabinddepth is -1.



Site Maps Control


- A sitemap is a list of your website's URL.
- It acts as a roadmap to tell search engines what and where content is available..
- You probably want some sort of navigation system to let users move from one page to the next page.
- A way to define the navigational structure of your website. This part is xml sitemap which is stored in a file.
- A way to use the site map information to display the user current portion and give the user the ability to easily move from one place to another.

Site Maps Control

- Every web sitemap file begins by declaring the `<sitemap>` element and ends by closing that element you place the actual site map information between the start and end tag.
- For ex. `<url>`
`<loc>http://ngu.ac.in/Colleges.aspx</loc>`
- `<changefreq>daily</changefreq> </url>`

Web Control Event

- Events are generated in response to a user action.
- For Ex. :- When user clicks on button, then click event is generated. Events are messages that contain details about the user action. Such as cause of the event and the control on which the event occurred.
- Asp.net all events pass two arguments.
- **Object**:- This is the first argument and is an instance of **System.object class**. This represents the control that raised the events.
- **Event object** :- this is an instance of **System.EventArgsClass**. This contain information about the event is generated.

- 
- Events can be generated **on the server side** or **the client side** but the processing of events happens only on the server.
 - **PostBack Events:-** Events that can cause the roundtrip to the server are called postback events.
 - **NonPostBack Events:-** Events that do not cause the roundtrip to the server.
 - **Autopostback** is the mechanism by which the page will be posted back to the server automatically based on some events in the web controls. In some of the web controls, the property called auto post back, if set to true, will send the request to the server when an event happens in the control

Web Control Event


Event	Web controls that provide it	AutoPostBack
Click	Button, Image button	True
TextChanged	TextBox	False
CheckChanged	Checkbox, Radio button	False
SelectedIndex Changed	DropDownlist, List Box, CheckBox List Radiobutton List	False

Master Page

- If you decide to update your navigation bar or change its position later.
- You need to modify every web page to apply the same change.
- The complexity of different pages that need to look and act the same one option is to subdivide the page into frame.
- Master page are similar to ordinary pages like master page are text file that can contain HTML, Web control and code.
- Master pages have a different file extension .master file.


A Simple Master Page & Content Page


- To see how this works it helps to create a simple example. To create a master page in visual studio- select website-> add new item -> select master page. Give it a file name and click add.
- When you create a new master page in .net. You start with a black page that include a ContentPlaceHolder control.
- If you want to give the content page the opportunity to supply content in a specific section of the page you need to add a contentplaceholder.

- 
- When you first create a master page. You start with two ContentPlaceHolder control.
 - One is defined in the <head> section which gives content pages the add page metadata such as search keywords and style sheet link.
 - Second content place holder is defined in the <body> section and represents the displayed content of the page. It appears on the page as a outlined box.

Themes

- Themes allow you to define a set of style details that you can apply to controls in multiple pages.
- Themes are implemented by the browser instead asp.net processes your themes when it create the page.
- A theme can include skin files, cascading style sheet files (.css files), and graphics. By applying a theme, one can give Web pages a consistent appearance across the website.
- **Creating a Theme:-** A theme is used in Web applications by creating a skin file and attaching it to the Web pages on which the theme has to be applied.

- 
- Skin files are used to define the property settings for ASP.NET Web server controls. Skin files are created and stored in a theme folder.
 - This theme folder is placed inside the folder named App_Themes. All the themes in a Web application are placed inside the App_Themes folder. This folder is placed inside the top-level directory of the Web application.
 - Multiple themes can be defined for the Web application. Each theme in a Web application can be defined in a separate folder inside the App_Themes folder which contains the skin file(s) for the theme.

- 
- Themes can contain CSS files, images and skins, you can change colors, fonts, positioning and images simply by applying the desired themes.
 - A skin file is a text file with the .skin extension asp.net never skin files directly instead. They are used behind the scenes to define a themes.

Data Controls

- One of the important goals of the Asp.net language is code minimization. Data controls play an important role in this purpose.
- Data controls used to display the records in the form of reports.
- To enable you to work efficiently with the data in your system, ASP.NET offers two sets of data-aware controls: The **Data-bound controls** and **The Data source controls**.
- The first group contains controls that you use to display and edit data, like the GridView, Repeater, and the new ListView control.

- The data source controls are used to retrieve data from a data source, like a database or an XML file, and then offer this data to the data-bound controls. shows you the complete list of available data controls in the Data category of the Toolbox.
- **The standard ASP.NET data presentation controls are:**
 - DetailsView.
 - FormView.
 - GridView.
 - ListView.
 - Repeate
 - DataList

GridView Control

- GridView control is a successor to the asp.net dataGrid control.
- It is the only dataBound control which has inbuilt Edit, Update, Delete, Sort, Paging data functionality. We can enable these functionalities by setting the properties of gridview without even writing a single line code.
- It provides more flexibility in displaying and working with data from your database in comparison with any other controls.
- The gridview control enables you to connect to a datasource and display data in tabular format however you have many options to customize the look and feel.

GridView Control

- Using this control is very easy when compared to other databound controls to display data. Just assigning the datasource property to a DataTable and calling DataBind() method will render the data.
- When it is rendered on the page generally it is implemented through <Table> html tag.
- Each column will be rendered as separate TD which makes it not possible to display data in any other format.
- Displays the values of a data source in a table where each column represents a field and each row represents a record. The GridView control enables you to select, sort, and edit these items.

GridView Events

Events	Description
PageIndexChanged	Both events occur when the page link is clicked.
RowCancelingEdit	Fires when cancel button is clicked in edit mode of the grid view
RowCommand	Fires when a button is clicked on any row of grid view.
RowCreated	Fires when a new row is created in grid view.
RowDataBound	Fires when row is bound to the data in grid view.
RowDeleting	Both events fire when delete button of a row is clicked.
RowEditing	Fire when a edit button of the row is clicked but before the grid view handles the edit operation.
RowUpadeted	Both events fire when a update button of the row is clicked.
Sorting	Both events fire when column header link is clicked.

div

Id	name	city	mobile	gender
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc
abc	abc	abc	abc	abc

SqlDataSource - SqlDataSource1

Output


localhost:50040/WebForm1.asp

localhost:50040/WebForm1....

Id	name	city	mobile	gender
101	abc	idar	85956	male
102	xyz	hmt	52565	female
103	bca	idar	46541	male
104	pqr	idar	78945	male
105	bcd	hmt	78656	female

Details View Control

- The Details View Control enable us to work with a single data item at a time.
- This control enable us to display, edit, insert, and delete data items such as database records.
- The details view displays single record at a time. It places each field in a separate row of a table. How to create a basic details view to show the currently selected record.
- The details view also allows you to move from one record to the next using paging controls.
- If you see the allow Paging property to true. You can configure the paging controls using the PagerStyle and PagerSettings Properties in the same way as you tweak the pager for the grid view.

- 
- Its tempting to using the details view pager controls to make a handy record browser. One problem is that a separate post back is required each time the user moves from one record to another record.
 - The details view user reflection to generate the filed it shows. This means it examiner the data object and creates a separate row for each filed.
 - Just like the grid view. You can disable this automatic generation by setting `AutoGenerateRows` to false. Then you to declare information you want to display.

Id	abc
name	abc
city	abc
mobile	abc
gender	abc
<u>Edit</u> <u>Delete</u> <u>New</u>	
1 2	


Id name city mobile gender					
<u>Select</u>	101	RRR	IDAR	89890	MALE
<u>Select</u>	104	xyz	idar	78945	female

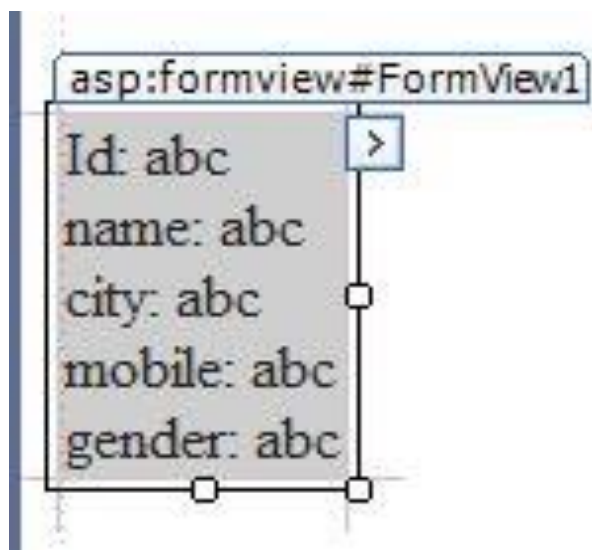
Output

Id	101
name	Ramesh
city	IDAR
mobile	89890
gender	MALE
<u>Edit</u> <u>Delete</u> <u>New</u>	

Form View Control

- If you need the ultimate flexibility of templates.
- The form view provides a template only control for displaying and editing a single record.
- The beauty of the form view template model is that it matches the model of the templateField in the gridview closely. This means you can work with template.
- Item Template
- EditItem Template
- InsertItem Template
- Footer Template
- Header Template
- EmptyData Template
- Pager Template

- 
- The grid view and details view which allow you to add as many TemplateField objects as you want the form view allows just a single copy of each template. If you show multiple values you must add multiple binding expression to the same item template.
 - The FormView control is a web server control that displays a single record from a data source. It's similar to the DetailsView control, but it uses user-defined templates instead of fields. You can use templates to display and edit data-bound values, and to create a free-form layout for a data record.



Id: 101
name: Ramesh
city: IDAR
mobile: 89890
gender: MALE


1 2

ListView Control

- The ListView control displays columns and rows of data and allows sorting and paging.
- It is by far the most popular data display control, and is ideal for understanding how data display controls interact with data retrieval controls and code.
- SQL Server Express (Graphics)
- Microsoft SQL Server (code)

Repeater Control

- The Repeater control is used to display a repeated list of items that are bound to the control. The Repeater control may be bound to a database table, an XML file, or another list of items.
- Repeater is a Data Bind Control. Data Bind Controls are container controls. Data Binding is the process of creating a link between the data source and the presentation UI to display the data. ASP .Net provides rich and wide variety of controls, which can be bound to the data.

- 
- **Repeater has 5 inline template to format it:**
 - <HeaderTemplate>
 - <FooterTemplate>
 - <ItemTemplate>
 - <AlternatingItemTemplate>
 - <SeperatorTemplate>
 - <AlternatingItemTemplate>

DataList


- DataList is a Databound control to display and manipulate data in a web application. It is a composite control that can combine other ASP.Net controls and it is present in the form. The DataList appearance is controlled by its template fields.
- **Itemtemplate:** It specifies the Items present in the Datasource, it renders itself in the browser as many rows present in the data source collection.
- **EditItemTemplate:** Used to provide edit permissions to the user.
- **HeaderTemplate:** Used to display header text to the data source collection.

- **FooterTemplate:** Used to display footer text to the data source collection.
- **ItemStyle:** Used to apply styles to an ItemTemplate.
- **EditStyle:** Used to apply styles to an EditItemTemplate
- **HeaderStyle:** Used to apply styles to a HeaderTemplate
- **FooterStyle:** Used to apply styles to a FooterTemplate.

EventName	Description
OnEditCommand	It will be raised when user clicks on a button(present in DataList) whose CommandName is "Edit"
OnUpdateCommand	It will be raised when user clicks on a button(present in DataList) whose CommandName is "Update"
OnCancelCommand	It will be raised when user clicks on a button(present in DataList) whose CommandName is "Cancel"
OnDeleteCommand	It will be raised when user clicks on a button(present in DataList) whose CommandName is "Delete"

DetailsView & Formview Control

- The DetailsView control always renders each field in a separate HTML table row.
- The Gridview excels at showing a dense table with multiple rows of information. Sometimes you want to provide a details look at a single record.
- You could work out a solution using a templatecolumn in a gridview. But includes two controls that are this purpose the details view & the form view.

- 
- Both show a single record at a time but can include optional page buttons that let you step through a series of records.
 - Both give you an easy way to insert a new record which the grid view doesn't allow. And both support templates but the form view require them.
 - One other difference is the fact that the details view renders its content inside a table while the form view gives you the flexibility to display your content without a table.
 - Details view gives you a simpler model that lets you build a multirow data display out of field objects, much the same way that the grid view is built out of column objects.