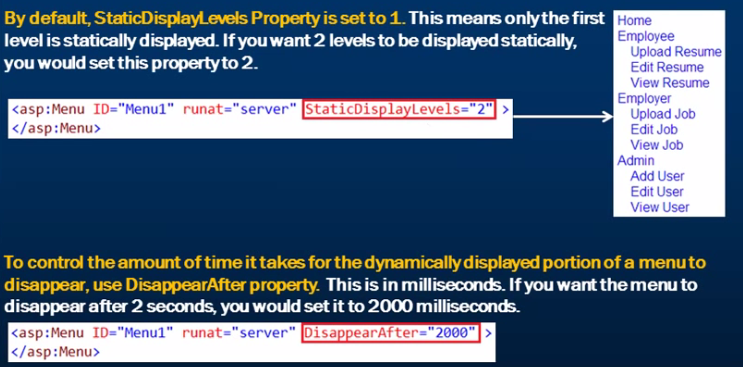
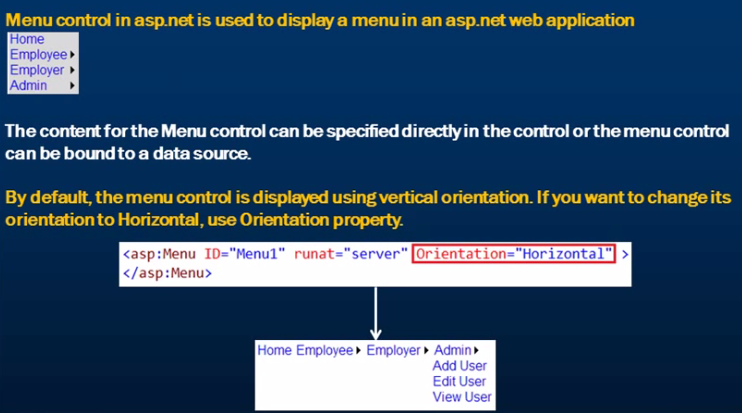
**Unit – 3**

**Navigation Control:** Navigation controls are basically **used to navigate the user through webpage** . It is more helpful for making the navigation of pages easier . There are three controls in ASP.NET ,Which are used for Navigation on the webpage.

1. Menu
2. SiteMapPath
3. TreeView

**Menu Control:**The Menu Control can be used to display hierarchical data. It is very easy to navigate through the Menu Control when there are many options in the hierarchy. Menus occupy less space in the web page.

The Menu control can be bound to a SiteMapDataSource to display the information available in the Web.sitemap file.



**HTML Source Code:**

**<asp:Menu ID="Menu1" runat="server" DisappearAfter="2000" >**

**<Items>**

**<asp:MenuItem Text= "Home" Value ="Home" NavigateUrl="~/Home.aspx"></asp:MenuItem>**

**</Items>**

**<Items>**

**<asp:MenuItem Text= "Employee" Value ="Employee" NavigateUrl="~/Employee.aspx">**

**<asp:MenuItem Text= "Upload Resume" Value ="UploadResume"**

**NavigateUrl="~/UploadResume.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "Edit Resume" Value ="EditResume"**

**NavigateUrl="~/EditResume.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "View Resume" Value ="ViewResume"**

**NavigateUrl="~/ViewResume.aspx"></asp:MenuItem>**

**</asp:MenuItem>**

**</Items>**

**<Items>**

**<asp:MenuItem Text= "Employer" Value ="Employer" NavigateUrl="~/Employer.aspx">**

**<asp:MenuItem Text= "Upload Job" Value ="UploadJob"**

**NavigateUrl="~/UploadJob.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "Edit Job" Value ="EditJob"**

**NavigateUrl="~/EditJob.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "View Job" Value ="ViewJob"**

**NavigateUrl="~/ViewJob.aspx"></asp:MenuItem>**

**</asp:MenuItem>**

**</Items>**

**<Items>**

**<asp:MenuItem Text= "Admin" Value ="Admin" NavigateUrl="~/Admin.aspx">**

**<asp:MenuItem Text= "Add User" Value ="UploadJob"**

**NavigateUrl="~/AddUser.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "Edit User" Value ="EditJob"**

**NavigateUrl="~/EditUser.aspx"></asp:MenuItem>**

**<asp:MenuItem Text= "View User" Value ="ViewJob"**

**NavigateUrl="~/ViewUser.aspx"></asp:MenuItem>**

**</asp:MenuItem>**

**</Items>**

**</asp:Menu>**

**SiteMapPath Control:**

The SiteMapPath control basically is used to access web pages of the website from one webpage to another.

The Site Map provides a logical structure of the pages in a website. The Site Map information is stored in a file named Web.sitemap. The Web.sitemap file is stored in the root directory of the web application. ASP.NET has the XmlSiteMapProvider which is used to access the information from the Web.sitemap file. It automatically reads data in the Web.sitemap file present in the root folder of the web application and stores it in a SiteMap object.

We have to take 1 file web.sitemap in which all hierarchy is available.

<?xmlversion="1.0"encoding="utf-8" ?>

<siteMapxmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0">

<siteMapNodetitle="Dummy Root">

<siteMapNodeurl="~/Home1.aspx"title="Home" />

<siteMapNodeurl="~/Employee1.aspx"title="Employee" />

<siteMapNodeurl="~/UploadResume1.aspx"title="Upload Resume"></siteMapNode>

<siteMapNodeurl="~/EditResume.aspx"title="Edit Resume"></siteMapNode>

<siteMapNodeurl="~/ViewResume.aspx"title="View Resume"></siteMapNode>

<siteMapNodeurl="~/Employer.aspx"title ="Employer" />

<siteMapNodeurl="~/UploadJob.aspx"title ="Upload Job"></siteMapNode>

<siteMapNodeurl="~/EditJob.aspx"title ="Edit Job"></siteMapNode>

<siteMapNodeurl="~/ViewJob.aspx"title ="View Job"></siteMapNode>

<siteMapNodeurl="~/Admin.aspx"title ="Admin" />

<siteMapNodeurl="~/AddUser.aspx"title ="Add User"></siteMapNode>

<siteMapNodeurl="~/EditUser.aspx"title ="Edit User"></siteMapNode>

<siteMapNodeurl="~/ViewUser.aspx"title ="View User"></siteMapNode>

</siteMapNode>

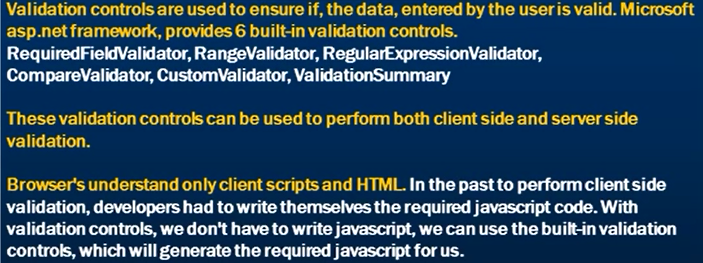
</siteMap>

Our SiteMapPath Control is connected withWeb.sitemap xml file.

There are different properties of SiteMapPath Control, which are following:

| PathSeparator | Gets or sets Path separator text. (By default it is >.) |
| --- | --- |
| NodeStyle | Sets the style of all nodes that will be displayed. |
| CurrentNodeStyle | Sets the style on node that represent the current page. |
| RootNodeStyle | Sets the style on the absoulte root node. |
| PathSeparatorStyle | Sets the style of path separator. |

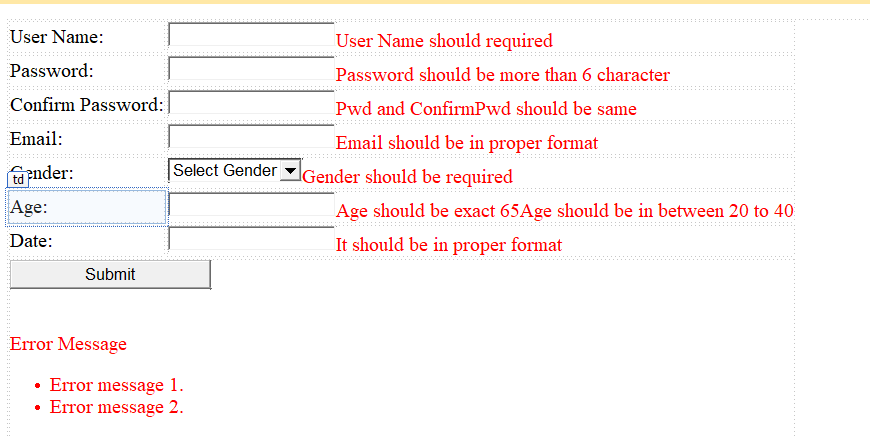
**Validation Control:**



here are six types of validation controls in ASP.NET

1. RequiredFieldValidation Control
2. CompareValidator Control
3. RangeValidator Control
4. RegularExpressionValidator Control
5. CustomValidator Control
6. ValidationSummary

Example:



**RequiredFieldValidation Control**

The RequiredFieldValidator control is simple validation control, which checks to see if the data is entered for the input control.

In Our example we have validate UserName and Gender for the compulsory field.

In which we need to set Following properties.

**For User Name:**

ControlToValidate – txtUserName

ErrorMessage – UserName should required

ForeColor- Red

**For Gender (DropDownList):**

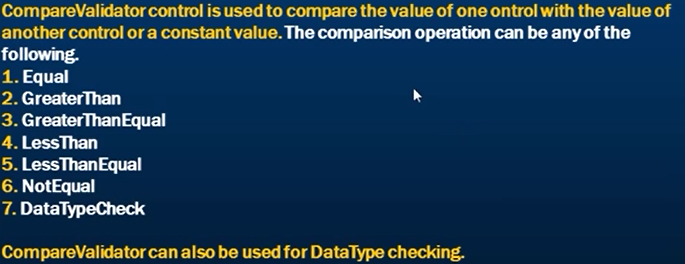
ControlToValidate – txtUserName

ErrorMessage – UserName should required

ForeColor- Red

InitialValue – 1 (hint – we have given Value = 1 for Select Gender Text)

**CompareValidator:**



In our example, we used CompareValidator for 3 ways:

1. we have validate 2 control Password and Confirm Password.

Properties are:

ControlToValidate – txtConfirmPwd

ControlToCompare - txtPwd

ErrorMessage – Pwd and ConfirmPwd should be same.

ForeColor- Red

1. Validate textbox for datatype check

ControlToValidate – txtDate

ErrorMessage – It should be in proper format.

ForeColor- Red

Operator- DataTypeCheck

Type- Date

1. Validate textbox for exact value

ControlToValidate – txtAge

ValueToCompare - 65

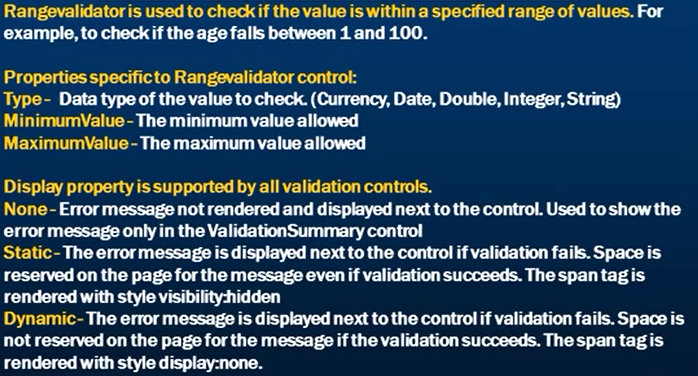
ErrorMessage – Age should be exact 65

ForeColor- Red

Operator- Equal

Type- Integer

**RangeValidator:**



In Our example we have set properties:

ControlToValidate – txtAge

MinValue – 20

MaxValue - 40

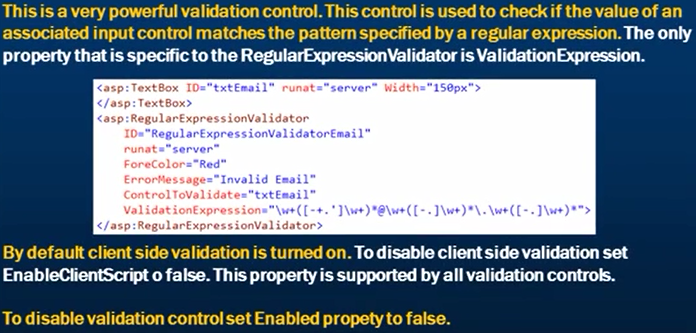
ErrorMessage – Age should be in between 20 and 40

ForeColor- Red

Type- Integer

Display - Dynamic

**RegularExpressionValidator:**



**Properties we have set:**

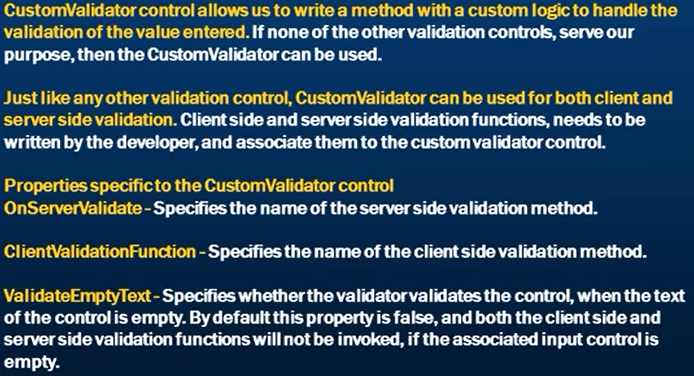
ControlToValidate – txtEmail

ErrorMessage – Email should be in proper format.

ForeColor- Red

ValidationExpression – Internet Email

**CustomValidator:**



**In our example we have written script :**

<scripttype="text/javascript">

functionValidateLength(sorc, args)

{

if (args.Value.length>= 6)

{

args.IsValid = true;

}

else

{

args.IsValid = false;

}

}

</script>

We have check here our password should be 6 digit or more.

Properties:

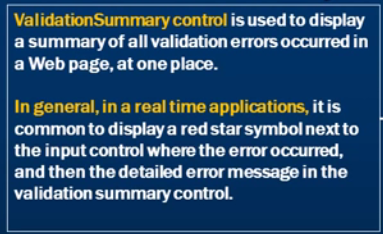
ControlToValidate – txtPwd

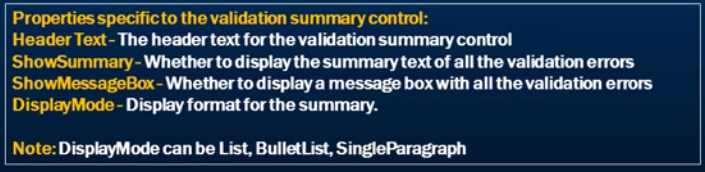
ErrorMessage – Pwd should be more than 6 character

ForeColor- Red

ClientValidationFunction - ValidateLength

**SummaryValidator:**

****

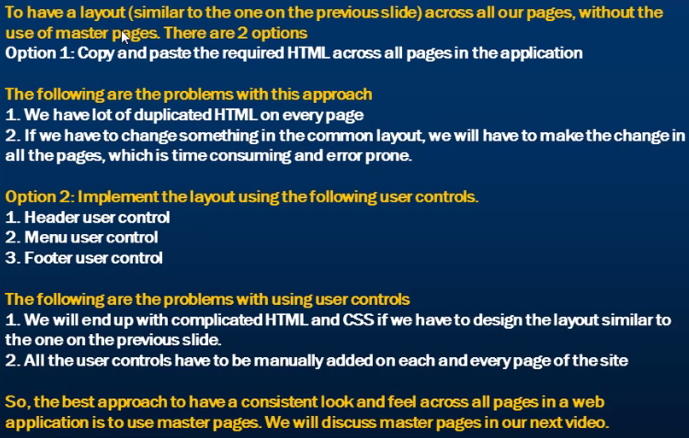
****

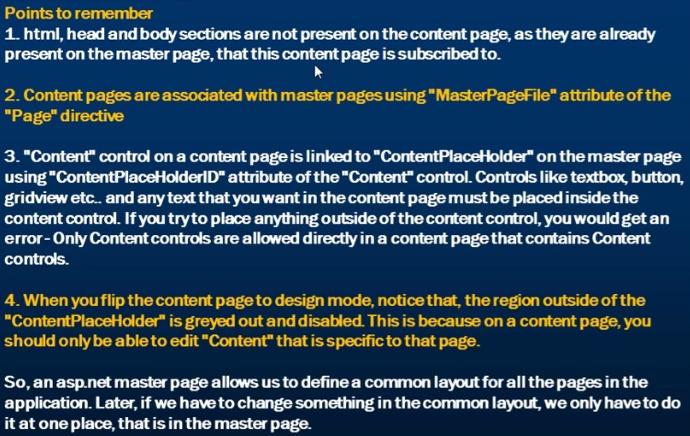
**Master Page:**

**It is common for a web site to have a consistent look and behaviour across all the pages in the web application.**

****

**Why user Master page in asp.net?**

****

****

**Source Code:**

<tablestyle="font-family: Arial">

<tr>

<tdcolspan="2"style="width: 800px; height: 80px; background-color:

#BDBDBD; text-align: center">

<h1>

WebSite Header

</h1>

</td>

</tr>

<tr>

<tdstyle="height: 500px; background-color: #D8D8D8; width: 150px">

<h3>

Menu

</h3>

</td>

<tdstyle="height: 500px; background-color: #E6E6E6; width: 650px">

<asp:ContentPlaceHolderID="ContentPlaceHolder1"runat="server">

<h2style="height: 409px">

This section changes on a page by page basis

</h2>

</asp:ContentPlaceHolder>

</td>

</tr>

<tr>

<tdcolspan="2"style="background-color: #BDBDBD; text-align: center">

<b>WebSite Footer</b>

</td>

</tr>

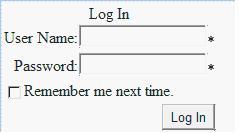
</table>

**ASP.NET LOGIN CONTROL:**

ASP.NET provides user with login controls in the Login tab of the toolbox of the Visual Studio application. The following controls are provided to the user.

1. [Login Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#login)
2. [The LoginView Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#LoginView)
3. [The LoginStatus Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#LoginStatus)
4. [LoginName Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#LoginName)
5. [PasswordRecovery Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#PasswordRecovery)
6. [CreateUserWizard Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#CreateUserWizard)
7. [ChangePassword Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#ChangePassword)

[Login Control](https://www.go4expert.com/articles/exploring-login-controls-aspnet-t30438/#login)

The Login control at the design time is as shown below:  
  


We can set different properties of Login Control:

| **Properties of the Login Control** | |
| --- | --- |
| TitleText | Indicates the text to be displayed in the heading of the control. |
| InstructionText | Indicates the text that appears below the heading of the control. |
| UserNameLabelText | Indicates the label text of the username text box. |
| PasswordLabelText | Indicates the label text of the password text box. |
| FailureText | Indicates the text that is displayed after failure of login attempt. |
| UserName | Indicates the initial value in the username text box. |
| LoginButtonText | Indicates the text of the Login button. |
| LoginButtonType | Button/Link/Image. Indicates the type of login button. |
| DestinationPageUrl | Indicates the URL to be sent after login attempt successful. |
| DisplayRememberMe | true/false. Indicates whether to show Remember Me checkbox or not. |
| VisibleWhenLoggedIn | true/false. If false, the control is not displayed on the page when the user is logged in. |
| CreateUserUrl | Indicates the url of the create user page. |
| CreateUserText | Indicates the text of the create user link. |
| PasswordRecoveryUrl | Indicates the url of the password recovery page. |
| PasswordRecoveryText | Indicates the text of the password recovery link. |
| **Style of the Login Control** | |
| CheckBoxStyle | Indicates the style property of the Remember Me checkbox. |
| FailureStyle | Indicates the style property of the failure text. |
| TitleTextStyle | Indicates the style property of the title text. |
| LoginButtonStyle | Indicates the style property of the Login button. |
| TextBoxStyle | Indicates the style property of the TextBox. |
| LabelStyle | Indicates the style property of the labels of text box. |
| HyperLinkStyle | Indicates the style property of the hyperlink in the control. |
| InstructionTextStyle | Indicates the style property of the Instruction text that appears below the heading of the control. |

**CreateUserWizard control**:-  
This control uses the membership service to create a new user in the membership data store. The CreateUserWizard control is provides by the CreateUserWizard class and can be customized by using template and style properties .Using this control any user can easily create an account  and login to the web page.  
You can drag and drop CreateUserWizared control on the web page as shown below:-



**PasswordRecovery Control:**

The PasswordRecovery control available in ASP.NET 3.5 assists is a part of Login controls and helps to recover their password who have forgotten their passwords. It enables a user to request an e-mail message containing either a new password or the password already associated with his or her user name or email.  
  
The PasswordRecovery Web control uses Internet e-mail services to send recovered or new passwords to users. There are inherent security risks with sending passwords in e-mail. You should determine whether these security risks are acceptable to your site.

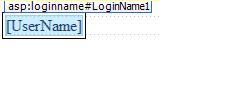
The e-mail message is sent using the MailDefinition class. To be able to send e-mail to users, you must configure a mail server in your application's Web.config file. You can change the content of the e-mail sent to users by setting a custom message in the MailDefinition property.

**ChangePassword Control**

The control helps user to change the password. The user adds the current password and adds the new password. If the old password is incorrect, the new one cannot be added.  
  
**Properties of the control**

1. CancelDestinationPageUrl: It accesses or retrieves the URL of the page that the user is shown once it clicks the Cancel button.
2. CurrentPassword: It retrieves the current password of a user.
3. DisplayUserName: It retrieves the value indicating whether the ChangePassword control should be display the control and label
4. NewPassword: It retrieves the new password entered by the user
5. UserName: It shows the username for which the password is to be modified.

**LoginName Control**

It is used for displaying the name of the authenticated users. The Page.User.Identity.Name is used for returning the user name. The control is not displayed if it does not contain any logged in user. The LoginName class is used for the control.  
  
The control does not contain any method, property or events associated with it. The FormatString property is used for displaying the string in the control.  
The LoginName control at the design time is as shown below:  
  


**The LoginStatus Control**

It specifies that a particular user has logged into the web site. The login status is displayed as a text. The login text is displayed as a hyperlink but provides the navigation to the login page. The authentication section of the web.config file is useful for accessing the login page URL.  
  
The LoggedIn and LoggedOut are the two status provided by the LoginStatus control. TheLoginStatus class provides the control.

**Properties of the LoginStatus Control**

1. LoginImageUrl: It accesses or specifies the URL of the image used for the login link.
2. LoginText: It access the text added for the login link
3. LogoutAction: It retrieves the value for determining the action when the user logs out of the web site.
4. LogoutText: It retrieves the text used for logout the link

**The LoginView Control**

The LoginView control is a web server control used for displaying the two different views of a web page. It helps to alter the page view for different logged in users. The current users status information is stored in the control. The control displays appropriate information depending on the user.

It used template for the loginview.

1. loggedIn Template
2. Anonymous Template