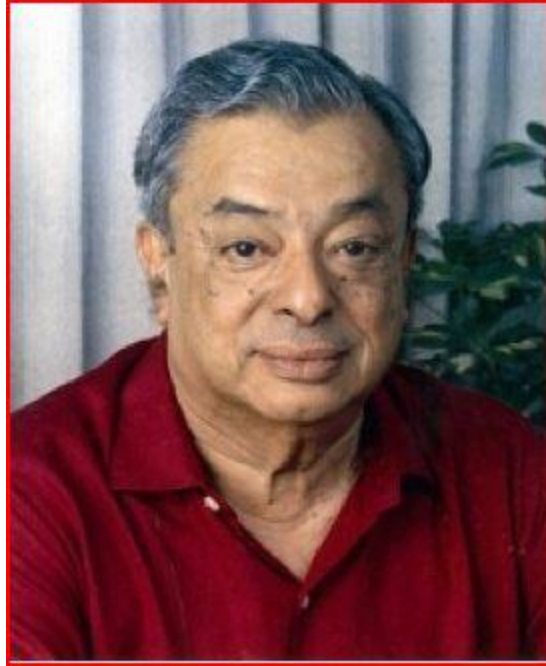


ASP.NET Web Validation Controls

Prepared for Vth semester DDU-CE students
2022-23 WAD

Apurva A Mehta



Introduction

- Can we trust users?
- Quick and precise feedback about mistakes users make
- Usability and Likeability
 - A judgemental user!

Client side Vs Server side Validations

- Client-side
 - Convenient to users
 - Avoids unnecessary communication with the server
 - Security
 - Malicious user can bypass the client-side validations
 - Platform dependency
- Server-side
 - Secure

Validations Control

Validation Control	Description
RequiredFieldValidator	Makes an input control a required field
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value
RangeValidator	Checks that the user enters a value that falls between two values
RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern
CustomValidator	Allows you to write a method to handle the validation of the value entered
ValidationSummary	Displays a report of all validation errors occurred in a Web page

Validation Process: CausesValidation

- Validation, Page, Modify Page, Submit
 - CausesValidation (Property)
 - True (Default)
 - ASP.NET validate each control on button click
 - Client side validations provided browser support
 - False
 - ASP.NET ignores validation controls

BaseValidator Class

- Validation control classes
 - System.Web.UI.WebControls
 - BaseValidator
- Members of BaseValidator
 - ControlToValidate
 - EnableClientScript
 - Enable
 - Text
 - ErrorMessage
 - IsValid
 - SetFocusOnError
 - ValidationGroup

RequiredFieldValidator

- Ensures that control is not empty
- Client Side Validations and Server Side Validation


```

<tr>
    <td>Name</td>
    <td>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
            ControlToValidate="TextBox1" Text="Please enter a Name!"
            ErrorMessage="Name is required" ForeColor="Red" ></asp:RequiredFieldValidator>
    </td>
</tr>

```

```

<tr>
    <td>Gender</td>
    <td>
        <asp:DropDownList ID="DropDownList1" runat="server">
            <asp:ListItem Selected="True" Value="-1">Select a gender</asp:ListItem>
            <asp:ListItem>Female</asp:ListItem>
            <asp:ListItem>Male</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
            ControlToValidate="DropDownList1" ErrorMessage="Gender is required"
            ForeColor="Red" InitialValue="-1"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>
        <asp:Button ID="Button1" runat="server" Text="Save" OnClick="Button1_Click" />
    </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>

```

```
protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}
```

0 references

```
protected void Button1_Click(object sender, EventArgs e)
{
    ////disable javascript and run demo: client validations are skipped
    //Label1.Text = "Data Saved successfully";

    //Solution
    if (Page.IsValid)
    {
        Label1.Text = "Data Saved successfully";
        Label1.ForeColor = System.Drawing.Color.Green;
    }
    else
    {
        Label1.ForeColor = System.Drawing.Color.Red;
        Label1.Text = "Validation Failed, Data not Saved";
    }
}
```

localhost:51249/WebForm1.aspx x +

localhost:51249/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Name

Gender

Please enter a Name!
Gender is required

localhost:51249/WebForm1.aspx x +

localhost:51249/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Name

Gender



localhost:51249/WebForm1.aspx



localhost:51249/WebForm1.aspx



Apps



Electronic library. D...



Manthan Movie - Y...

Name

Apurva

Gender

Male



Save

Data Saved successfully

RangeValidator

- The RangeValidator control verifies that an input value falls within a predetermined range
- It has three specific properties: MinimumValue, MaximumValue, and Type.

```
<tr>
  <td>Age</td>
  <td>
    <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
  </td>
  <td>
    <asp:RangeValidator ID="RangeValidatorAge" runat="server"
      ControlToValidate="txtAge" Display="Dynamic"
      ErrorMessage="Age must be in 1 and 100" ForeColor="Red"
      MaximumValue="100" MinimumValue="1" SetFocusOnError="True"
      Type="Integer">Enter Age: 1-100</asp:RangeValidator>
    <asp:RequiredFieldValidator ID="RequiredFieldValidatorAge"
      runat="server" ControlToValidate="txtAge" Display="Dynamic"
      ErrorMessage="Age is a required field" ForeColor="Red">
    </asp:RequiredFieldValidator>
  </td>
</tr>
```

```
<tr>
  <td>Date Available</td>
  <td>
    <asp:TextBox ID="txtDateAvl" runat="server"></asp:TextBox>
  </td>
  <td>
    <asp:RangeValidator ID="RangeValidatorDateAvl" runat="server"
      ControlToValidate="txtDateAvl" Display="Dynamic"
      ErrorMessage="Date must be 1/1/2019 to 31/12/2019"
      ForeColor="Red" SetFocusOnError="True" Type="Date"
      MinimumValue="01/01/2019" MaximumValue="31/12/2019">
    </asp:RangeValidator>
  </td>
</tr>
<tr>
  <td>
    <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="Save" />
  </td>
  <td>&nbsp;</td>
  <td></td>
</tr>
<tr>
  <td>
    <asp:Label ID="lblStatus" runat="server"></asp:Label>
  </td>
  <td>
  </td>
  <td>
  </td>
</tr>
```

```
protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}
```

0 references

```
protected void btnSave_Click(object sender, EventArgs e)
{
    ////disable javascript and run demo: client validations are skipped
    //Label1.Text = "Data Saved successfully";

    //Solution
    if (Page.IsValid)
    {
        lblStatus.Text = "Data Saved successfully";
        lblStatus.ForeColor = System.Drawing.Color.Green;
    }
    else
    {
        lblStatus.ForeColor = System.Drawing.Color.Red;
        lblStatus.Text = "Validation Failed, Data not Saved";
    }
}
```


localhost:51739/WebForm1.aspx x +

localhost:51739/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Age

Date Available

Save

localhost:51739/WebForm1.aspx x +

localhost:51739/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Age

Date Available

Save

Age is a required field

localhost:51739/WebForm1.aspx x +

localhost:51739/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Age

Date Available

Save

Enter Age: 1-100

localhost:51739/WebForm1.aspx x +

localhost:51739/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Age

Date Available

Save

Date must be 1/1/2019 to 31/12/2019

localhost:51739/WebForm1.aspx x +

localhost:51739/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Age

Date Available

Save

Data Saved successfully

CompareValidator

- Compares
 - Value in one control with a fixed value
 - Value in one control with value in another control
- Properties
 - ValueToCompare
 - ControlToCompare
 - Type
 - Operator

```
<tr>
    <td>Password</td>
    <td>
        <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
    </td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>Confirm Password</td>
    <td>
        <asp:TextBox ID="txtRePassword" runat="server" TextMode="Password"></asp:TextBox>
    </td>
    <td>
        <asp:CompareValidator ID="CompareValidatorConfirmPassword" runat="server"
            ControlToCompare="txtPassword" ControlToValidate="txtRePassword"
            Display="Dynamic" ErrorMessage="Password and Confirm Password must be same"
            ForeColor="Red" Text="Password and Confirm Password must match.">
            
        </asp:CompareValidator>
    </td>
</tr>
```

```

<tr>
    <td>Date of Application</td>
    <td>
        <asp:TextBox ID="txtDateofApp" runat="server" TextMode="Date"></asp:TextBox>
    </td>
    <td>
        <asp:CompareValidator ID="CompareValidatorDateofApp" runat="server"
            ControlToValidate="txtDateofApp" Display="Dynamic"
            ErrorMessage="Date of App must be greate than 10/07/2019"
            ForeColor="Red" Operator="GreaterThan" Type="Date"
            ValueToCompare="10/07/2019"></asp:CompareValidator>
    </td>
</tr>

```

```

<tr>
    <td>Age</td>
    <td>
        <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
    </td>
    <td>
        <asp:CompareValidator ID="CompareValidatorAge" runat="server"
            ControlToValidate="txtAge" Display="Dynamic" ErrorMessage="Age must ne a number"
            ForeColor="Red" Operator="DataTypeCheck" SetFocusOnError="True"
            Type="Integer">Age is a number</asp:CompareValidator>
    </td>
</tr>
<tr>
    <td>
        <asp:Button ID="btnSubmit" runat="server" OnClick="btnSubmit_Click" Text="Submit" />
    </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>
        <asp:Label ID="lblStatus" runat="server"></asp:Label>
    </td>

```

```
protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}
```

0 references

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    ////disable javascript and run demo: client validations are skipped
    //lblStatus.Text = "Data Saved successfully";

    //Solution
    if (Page.IsValid)
    {
        lblStatus.Text = "Data Saved successfully";
        lblStatus.ForeColor = System.Drawing.Color.Green;
    }
    else
    {
        lblStatus.ForeColor = System.Drawing.Color.Red;
        lblStatus.Text = "Validation Failed, Data not Saved";
    }
}
```

localhost:52007/WebForm1.aspx x +

localhost:52007/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Password

Confirm Password

Date of Application

Age

Password and Confirm Password must match

Date of App must be greate than 10/07/2019

Age is a number

Error for Password comparison is in form of an image

localhost:52007/WebForm1.aspx x +

localhost:52007/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Password

Confirm Password

Date of Application

Age

Regular Expression

- To specify pattern

Quantifier	Description
*	Zero or more matches
+	One or more matches
?	Zero or one matches
{N}	N matches
{N,}	N or more matches
{N,M}	Between N and M matches (inclusive)

Cont.

Character Class	Description
.	Matches any character except \n.
[aeiou]	Matches any single character specified in the set.
[^aeiou]	Matches any character not specified in the set.
[3-7a-dA-D]	Matches any character specified in the specified ranges (in the example the ranges are 3–7, a–d, A–D).
\w	Matches any word character; that is, any alphanumeric character or the underscore (_).
\W	Matches any nonword character.
\s	Matches any whitespace character (space, tab, form feed, newline, carriage return, or vertical feed).
\S	Matches any nonwhitespace character.
\d	Matches any decimal character.
\D	Matches any nondecimal character.

RegularExpressionValidator

- Checks if the value of an associated input control matches the pattern specified by a regular expression.
- Property
 - ValidationExpression

```

<tr>
    <td>Email</td>
    <td>
        <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
    </td>
    <td>
        <asp:RegularExpressionValidator ID="RegularExpressionValidatorEmail"
            runat="server" ControlToValidate="txtEmail"
            Display="Dynamic" ErrorMessage="Invalid Email Address"
            ForeColor="Red" SetFocusOnError="True"
            ValidationExpression="\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*">
            Need a suitable Email</asp:RegularExpressionValidator>
    </td>
</tr>

```

```

<tr>
    <td>Mobile No</td>
    <td>
        <asp:TextBox ID="txtMobileNo" runat="server"></asp:TextBox>
    </td>
    <td>
        <asp:RegularExpressionValidator ID="RegularExpressionValidatorMobileNo"
            runat="server" ControlToValidate="txtMobileNo" Display="Dynamic"
            ErrorMessage="Mobile no must be of 10 digits" ForeColor="Red"
            SetFocusOnError="True" ValidationExpression="\d{10}">10 digit mobile no
        </asp:RegularExpressionValidator>
    </td>
</tr>

```

```

<tr>
    <td>
        <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="Save" />
    </td>
    <td>&nbsp;  </td>
    <td>&nbsp;  </td>
</tr>
<tr>
    <td>
        <asp:Label ID="lblStatus" runat="server" ForeColor="Black"></asp:Label>
    </td>
</tr>

```

```
protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}
```

0 references

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    ////disable javascript and run demo: client validations are skipped
    //lblStatus.Text = "Data Saved successfully";

    //Solution
    if (Page.IsValid)
    {
        lblStatus.Text = "Data Saved successfully";
        lblStatus.ForeColor = System.Drawing.Color.Green;
    }
    else
    {
        lblStatus.ForeColor = System.Drawing.Color.Red;
        lblStatus.Text = "Validation Failed, Data not Saved";
    }
}
```

localhost:50033/WebForm1.aspx x +

localhost:50033/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Email

Mobile No

Need a suitable Email
10 digit mobile no

Save

localhost:50033/WebForm1.aspx x +

localhost:50033/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Email

Mobile No

Save

CustomValidator



shutterstock.com • 1129766546



Cont.

- Properties
 - OnServerValidate
 - ClientValidationFunction
 - ValidateEmptyText

```
<tr>
  <td class="auto-style2">Enter a positive even number</td>
  <td class="auto-style2">
    <asp:TextBox ID="txtEven" runat="server"></asp:TextBox>
  </td>
  <td class="auto-style2">
    <asp:CustomValidator ID="CustomValidatorEven" runat="server"
      ControlToValidate="txtEven" Display="Dynamic"
      ErrorMessage="Not a positive even number" ForeColor="Red"
      OnServerValidate="CustomValidatorEven_ServerValidate"
      ValidateEmptyText="True">Positive even number is required
    </asp:CustomValidator>
  </td>
</tr>
```



```
protected void CustomValidatorEven_ServerValidate(object source, ServerValidateEventArgs args)
{
    if(args.Value=="")
    {
        args.IsValid = false;
    }
    else
    {
        int Number;
        bool isNumber = int.TryParse(args.Value, out Number);

        if (isNumber && Number >= 0 && Number%2 == 0)
        {
            args.IsValid = true;
        }
        else
        {
            args.IsValid = false;
        }
    }
}
```

```
<tr>
  <td class="auto-style2">Enter a positive odd number</td>
  <td class="auto-style2">
    <asp:TextBox ID="txtOdd" runat="server"></asp:TextBox>
  </td>
  <td class="auto-style2">
    <asp:CustomValidator ID="CustomValidatorOdd" runat="server"
      ControlToValidate="txtOdd" Display="Dynamic"
      ErrorMessage="Not a positive odd number" ForeColor="Red"
      ValidateEmptyText="True" ClientValidationFunction="IsOdd">
      Positive odd number is required</asp:CustomValidator>
  </td>
</tr>
```

```
<script type="text/javascript">
    function IsOdd(source,args)
    {
        if(args.Value == "")
        {
            args.IsValid = false;
        }
        else
        {
            if(args.Value % 2 == 1)
            {
                args.IsValid = true;
            }
            else
            {
                args.IsValid = false;
            }
        }
    }
</script>
```

```

<tr>
    <td>
        <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="Save" />
    </td>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
    <td>
        <asp:Label ID="lblStatus" runat="server"></asp:Label>

```

```

protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}

```

0 references

```

protected void btnSave_Click(object sender, EventArgs e)
{
    ////disable javascript and run demo: client validations are skipped
    //lblStatus.Text = "Data Saved successfully";

    //Solution
    if (Page.IsValid)
    {
        lblStatus.Text = "Data Saved successfully";
        lblStatus.ForeColor = System.Drawing.Color.Green;
    }
    else
    {
        lblStatus.ForeColor = System.Drawing.Color.Red;
        lblStatus.Text = "Validation Failed, Data not Saved";
    }
}

```

localhost:50391/WebForm1.aspx x +

localhost:50391/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Enter a positive even number

Enter a positive odd number

Positive odd number is required

Save

localhost:50391/WebForm1.aspx x +

localhost:50391/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Enter a positive even number

Enter a positive odd number

Positive even number is required

Save

Validation Failed, Data not Saved

ValidationSummary

- Display a summary of all validation errors occurred in a Web page, at once
- Properties
 - HeaderText
 - ShowSummary
 - ShowMessageBox
 - DisplayMode

```

<tr>
  <td>Email</td>
  <td>
    <asp:TextBox ID="txtEmail" runat="server" ValidationGroup="Email_Gender">
    </asp:TextBox>
  </td>
  <td>
    <asp:RequiredFieldValidator ID="RequiredFieldValidatorEmail"
      runat="server" ErrorMessage="Email is required" Text="*"
      ControlToValidate="txtEmail" Display="Dynamic" ForeColor="Red">
    </asp:RequiredFieldValidator>
    <asp:RegularExpressionValidator ID="RegularExpressionValidatorEmail"
      runat="server" ControlToValidate="txtEmail" Display="Dynamic"
      ErrorMessage="Invalid Email Address" ForeColor="Red"
      ValidationExpression="\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*">
    </asp:RegularExpressionValidator>
  </td>
</tr>

```

```

<tr>
  <td>Username</td>
  <td>
    <asp:TextBox ID="txtUsername" runat="server" ValidationGroup="Email_Gender">
    </asp:TextBox>
  </td>
  <td>
    <asp:RequiredFieldValidator ID="RequiredFieldValidatorUsername" runat="server"
      ErrorMessage="Username is required" Text="*" ControlToValidate="txtUsername"
      ForeColor="Red"></asp:RequiredFieldValidator>
  </td>
</tr>

```

```

<tr>
  <td>Password</td>
  <td>
    <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
      ValidationGroup="Password_ConfirmPassword"></asp:TextBox>
  </td>
  <td>
    <asp:RequiredFieldValidator ID="RequiredFieldValidatorPassword"
      runat="server" ErrorMessage="Password is required" Text="*"
      ControlToValidate="txtPassword" ForeColor="Red">
    </asp:RequiredFieldValidator>
  </td>
</tr>

```

```

<tr>
  <td>Confirm Password</td>
  <td>
    <asp:TextBox ID="txtConfirmPassword" runat="server" TextMode="Password"
      ValidationGroup="Password_ConfirmPassword"></asp:TextBox>
  </td>
  <td>
    <asp:RequiredFieldValidator ID="RequiredFieldValidatorConfirmPassword"
      runat="server" ErrorMessage="Confirm password is required" Text="*"
      ControlToValidate="txtConfirmPassword" Display="Dynamic" ForeColor="Red">
    </asp:RequiredFieldValidator>
    <asp:CompareValidator ID="CompareValidatorConfirmPassword" runat="server"
      ControlToCompare="txtPassword" ControlToValidate="txtConfirmPassword"
      Display="Dynamic" ErrorMessage="Confirm Password must match with password"
      ForeColor="Red">*</asp:CompareValidator>
  </td>
</tr>

```



```

<tr>
    <td>
        <asp:Button ID="btnSave" runat="server" OnClick="btnSave_Click" Text="Save" />
    </td>
    <td>
        <asp:Button ID="Button1" runat="server" CausesValidation="False"
            OnClick="Button1_Click" Text="Clear" />
    </td>
    <td>&nbsp;   </td>
</tr>
<tr>
    <td>
        <asp:ValidationSummary ID="ValidationSummary1" runat="server"
            DisplayMode="List" ForeColor="Red" HeaderText="Validations Errors" />
    </td>
</tr>

```

```

protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}

```

0 references

```

protected void btnSave_Click(object sender, EventArgs e)
{
}

```

0 references

```

protected void Button1_Click(object sender, EventArgs e)
{
    txtEmail.Text = "";
    txtUsername.Text = "";
    txtPassword.Text = "";
    txtConfirmPassword.Text = "";
}

```

localhost:50686/WebForm1.aspx x +

localhost:50686/WebForm1.aspx

Apps Electronic library. D... Manthan Movie - Y...

Email	<input type="text" value="test@test.test"/>
Username	<input type="text"/>
Password	<input type="password" value="...."/>
Confirm Password	<input type="password" value="...."/>

Validations Errors

Username is required

Confirm Password must match with password

CausesValidations

- Default: true for any web control
 - Validation happens
 - Clear button event also tests validations
- CausesValidations=false
 - Prevent validations from happening.
 - This property is mostly used to prevent validation when a cancel or reset button is clicked.

ValidationGroup Property

- Fields of certain categories are required to validate
 - e.g. Login Fields, Personal Info Fields, etc...
- Default
 - All validations control on page get validated
- ValidationGroup property to group validation controls using same group name

```

<tr>
    <td>Personal Information</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>Name</td>
    <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorName" runat="server"
            ControlToValidate="txtName" Display="Dynamic" ErrorMessage="Name is required"
            ForeColor="Red" SetFocusOnError="True" ValidationGroup="Personal">*
        </asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>Age</td>
    <td><asp:TextBox ID="txtAge" runat="server"></asp:TextBox></td>
    <td>
        <asp:CompareValidator ID="CompareValidatorAge" runat="server"
            ControlToValidate="txtAge" Display="Dynamic" ErrorMessage="You must be 18 or older"
            ForeColor="Red" Operator="GreaterThanOrEqual" Type="Integer" ValidationGroup="Personal"
            ValueToCompare="18">*</asp:CompareValidator>
    </td>
</tr>

```

```

<asp:Button ID="btnPersonal" runat="server" Text="Validate Personal Info"
    ValidationGroup="Personal" />

```

```

<asp:ValidationSummary ID="ValidationSummary2" runat="server" ForeColor="Red"
    ValidationGroup="Personal" />

```

```

<tr>
    <td>Professional Information</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>Highest Degree</td>
    <td><asp:TextBox ID="txtDegree" runat="server"></asp:TextBox></td>
    <td>
        <asp:RequiredFieldValidator ID="RequiredFieldValidatorDegree" runat="server"
            ControlToValidate="txtDegree" Display="Dynamic"
            ErrorMessage="Specify your highest degree" ForeColor="Red"
            SetFocusOnError="True" ValidationGroup="Professional">*
        </asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>Years of Experience</td>
    <td><asp:TextBox ID="txtExperience" runat="server"></asp:TextBox></td>
    <td>
        <asp:CompareValidator ID="CompareValidatorExperience" runat="server"
            Display="Dynamic" ErrorMessage="Must be an integer" ForeColor="Red"
            Operator="DataTypeCheck" Type="Integer" ValidationGroup="Professional"
            ControlToValidate="txtExperience">*</asp:CompareValidator>
    </td>
</tr>

```

```

<asp:Button ID="btnProfessional" runat="server" Text="Validate Professional Information"
    ValidationGroup="Professional" />

```

```

<asp:ValidationSummary ID="ValidationSummary1" runat="server" ForeColor="Red"
    ValidationGroup="Professional" />

```

```

protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;
}

```

Personal Information

Name

Age

Validate Personal Info

- Name is required
- You must be 18 or older

Professional Information

Highest Degree

Years of Experience

Validate Professional Information

Professional Information

Highest Degree

Years of Experience

Validate Professional Information

- Specify your highest degree
- Must be an integer

1. Which validation control can be used to determine whether data that is entered into a TextBox control is of type Currency?
2. What control can be used to display all validation errors on a page inside a pop-up window?

You need to validate a vendor ID entered by a user. The valid vendor IDs exist in a database table. How can you validate this input?

A. Provide a `RegularExpressionValidator` control and set the `ValidationExpression` property to `/DbLookup{code}`.

B. Provide a `RangeValidator` control, and set the `MinValue` property to `DbLookup(code)` and the `MaxVaue` property to `DbLookup(code)`.

C. Provide a `CustomValidator` control with server-side code to search the database for the code.

D. Provide a `CompareValidator` control and set the compare expression to the name of a server-side function that performs a database lookup of the code.

You have created a webpage that contains many controls that are validated by using validation controls. This page also contains Button controls that perform postbacks. You disabled all of the client-side validation and noticed that when you clicked any of the Button controls, the code in the Click event handler was executing even when some of the controls did not have valid data. How can you best solve this problem to ensure that code is not executed when invalid data exists?

- A. In the Click event handler method for each of your Button controls, test the webpage's IsValid property and exit the method if this property is false.
- B. In the Load event handler method of the webpage, test the webpage's IsValid property and exit the method if this property is false.
- C. Re-enable the client-side script to disable postback until valid data exists.
- D. Add the runat="server" attribute to all of the validation controls.

You have created an elaborate webpage that contains many validated controls. You want to provide a detailed message for each validation error, but you don't have space to provide the detailed message next to each control. What can you do to indicate an error at the control and list the detailed error messages at the top of the webpage?

- A. Set the Text property of the validator control to the detailed message and set the ErrorMessage property to an asterisk. Place a ValidationSummary control at the top of the webpage.
- B. Set the ErrorMessage property of the validator control to the detailed message and set the Text property to an asterisk. Place a ValidationSummary control at the top of the webpage.
- C. Set the ToolTip property of the validator control to the detailed message and set the ErrorMessage property to an asterisk. Place a ValidationSummary control at the top of the webpage.
- D. Set the ToolTip property of the validator control to the detailed message and set the Text property to an asterisk. Place a ValidationSummary control at the top of the webpage.