

2.3 Validation

Validation stands for checking data for as per the application requirements. ASP.NET provides various types of Validations controls to validate the data.

2.3.1 Validation Controls

Q. What is use of validation controls? List the different validation controls. (4 Marks)

- Validation controls in ASP.NET validate the user input data to assure that unauthenticated data should not get stored in database.
- Validation controls are used to :
 - i) Apply presentation logic.
 - ii) Verify data entered by user.
 - iii) Validate Data format, data type and data range of data entered by user

Validation control has two types

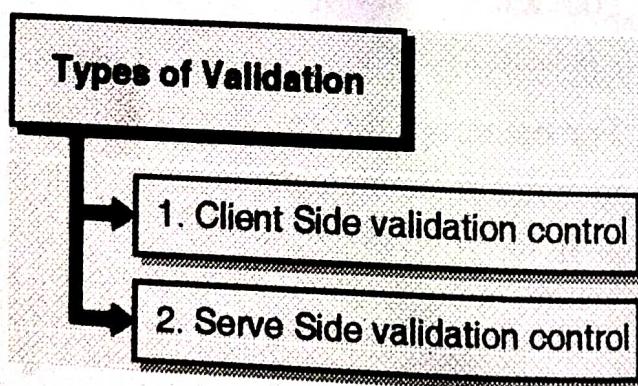


Fig C2.4 : Types of Validation Control

- **1. Client Side validation control**
- Client side validation is suitable to users as they get the feedback immediately.
 - The core advantage is that it stops the process of postback the page to server until the client validation is executed successfully.

- Client side validation is good but in it we have to be dependent on browser and scripting language support.

- For client script .NET uses JavaScript. WebUI Validation.js javascript file is used by .NET for client validation.

→ 2. Serve Side validation control

According to developer point of view, serve side validation controls are suitable because server control will not fail and not dependent on browser and scripting language.

- We can use ASP.NET validation controls which will assure the client side as well as server validation.
- ASP.NET validation controls work at both the ends. First it will work on client validation and than on server validation.
- ASP.NET provides a set of validation controls which are user friendly and they provide powerful way to check for errors and if needed show error messages to the user.
- ASP.NET provides the following validation controls :
 1. Required Field Validator
 2. Range Validator
 3. Compare Validator
 4. Regular Expression Validator
 5. Custom Validator
 6. Validation Summary

Q. Important points for validation controls

1. Control to Validate property is compusary for all validatation controls.
2. One validation control can be used to validate only one input control but multiple validation controls can be assigned to one input control.

5. Set the value of CauseValidation property to true for submit button as following :

```
<asp:Button ID="Button2" runat="server" Text="Submit"  
CausesValidation=true />
```

2.3.4 HTML5 Validation

Q. List all HTML5 validation controls and explain any 2 HTML validation controls in details. **(4 Marks)**

Following are the HTML5 Form Validation controls in ASP.NET :

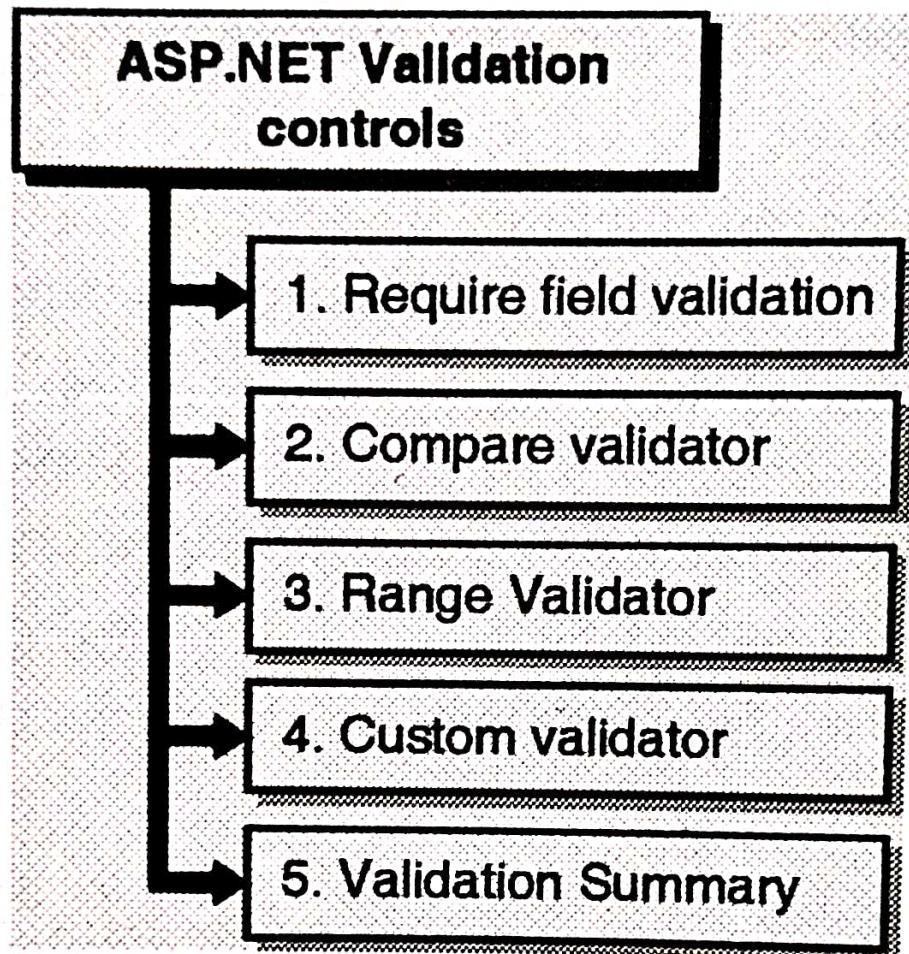


Fig. C2.5 : ASP.NET validation controls

→ 1. **Require field validation**

- The Require field validation is used to verify user enter data in form filed such as textbox,checkbox , dropdownlist etc .
- Html5 attribute "required" is used to do Required Field validation.
- It is generally tied to a textbox to force input in it.

☞ **Syntax**

```
<asp:RequiredFieldValidatorID="rfvcandidate"  
runat="server"ControlToValidate="emailid"  
ErrorMessage="Please enter proper Emailid"  
InitialValue=" Please enter proper Emailid ">  
  
</asp:RequiredFieldValidator>
```

A program to check whether name, email and age fields are filled by user or not. If not then display proper error message using Required field validation.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm6.aspx.cs"
Inherits="WebApplication7.WebForm6" %>

<html>
<head runat="server">
<title></title>
</head>

<body style = "font-family:Arial;font-size:10pt">
<form id="form1" runat="server">

    Name: (Required)<br />
    <asp:TextBox ID="txtName"
    runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator
    ID="RequiredFieldValidator1" CssClass
    ="Validators"
    Display="Dynamic" ControlToValidate="txtName"
    runat="server"
    ErrorMessage="Name is
    required."></asp:RequiredFieldValidator>
<hr />
```

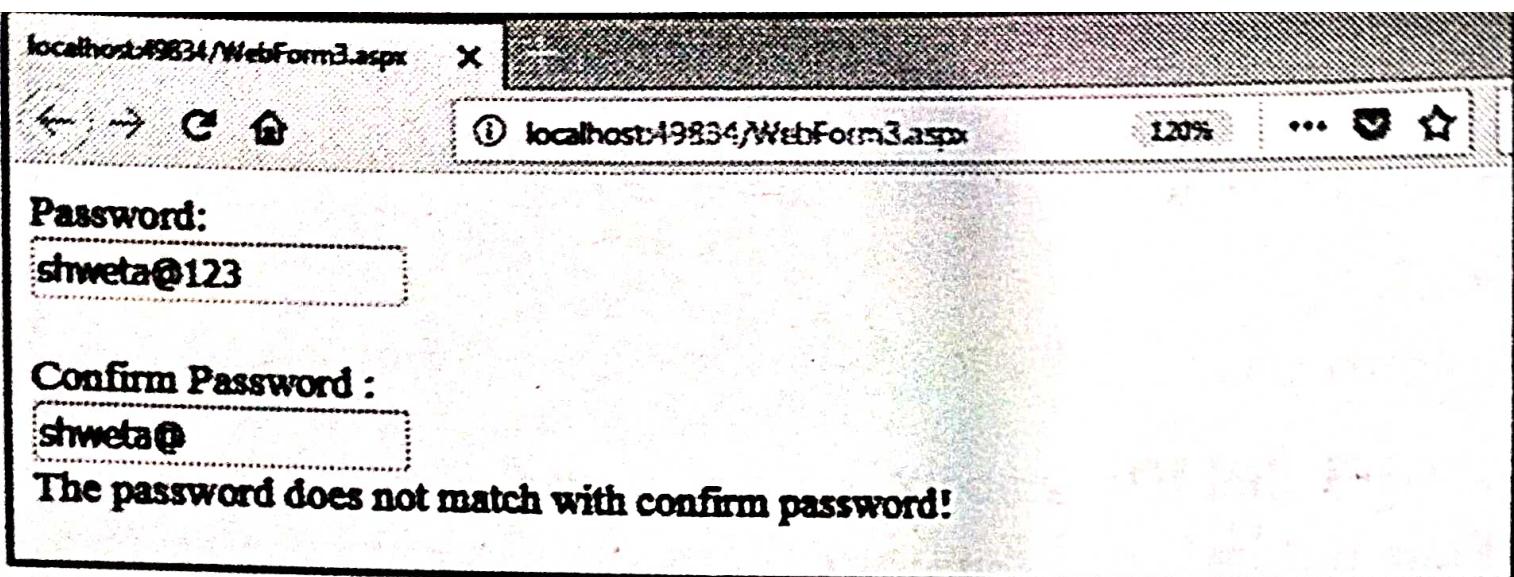
Entering value in Name field is compulsory, if not enter then show message "Name is required".

2. Compare Validator

- Compare Validator control compares value in one control with value in another control.
- For Example we can use Compare validator to reconfirm the password provided in one textbox by user with value in another textbox.

```
<form id="form1" runat="server">
<div>
    Password:<br />
    <asp:TextBox runat="server" id="txtpwd" /><br />
    <br />
    Confirm Password :<br />
    <asp:TextBox runat="server" id="txtconfirmpwd" />
    <br />
    <asp:CompareValidator runat="server" id="cmppwd"
        controltovalidate="txtpwd"
        controltocompare="txtconfirmpwd"
        errormessage="The password does not match with
        confirm password!">
        Type="String" />
    <br />
</div>
</form>
</body>
</html>
```

Compare value in password field
with confirm password field if values
are not matched than display error
message.



→ 3. Range Validator

- The RangeValidator control verifies that the input value should be in between the range which is specified.

☞ Range validator has three specific properties

Properties	Description
Type	This property state the type of the data. The available values for this property are: String, Integer, Currency, Date, Double.
MinimumValue	This property state lower limit of range.
MaximumValue	This property state upper limit of range.



Syntax

```
<asp:RangeValidator ID="rangevalidationclass"
runat="server" ControlToValidate="classtext"
ErrorMessage="Enter your class (1 - 10)"
MaximumValue="10"
MinimumValue="1" Type="Integer">

</asp:RangeValidator>
```

```
<form id="form1" runat="server">  
<div>  
  
<asp:Label ID="Label2" runat="server"  
Font-Bold="True" Font-Size="Small"  
Text="Your Age:></asp:Label>
```


Create
textbox which
accept age in
range of 18
and 100

```
<asp:TextBox ID="TextBox1" runat="server"  
Width="170px"></asp:TextBox>
```

```
<asp:RangeValidator ID="RangeValidator1"  
runat="server"
```

ControlToValidate="TextBox1"

ErrorMessage="Invalid Age. Please enter the age
between 18 to 100."

```
    MaximumValue="100" MinimumValue="18"  
    Type="Integer">> </asp:RangeValidator>
```

```
&nbsp;<br />  
</div>  
</form>  
</body>  
</html>
```

Output

