

Course Code: CT2207	Course Name: Lab: Web Technologies
----------------------------	---

Name of Students: Badal Wanjari	Semester/ Section: 3 rd / B
Roll No: 140	Enroll No: 20011045

Practical No: 10

Aim: Write Theory regarding ASP.Net Data Validation Controls. Create a Web Form to demonstrate the use of ASP.Net Data Validation Controls

THEORY:

Data Validation

Data validation is important part of any web application. User's input must always be validated before sending across different layers of the application.

Validation controls are used to,

- Implement presentation logic.
- To validate user input data.
- Data format, data type and data range is used for validation.

Validation is of two types

1. Client Side
2. Server Side

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

Client side validation is considered convenient for users as they get instant feedback. The main advantage is that it prevents a page from being postback to the server until the client validation is executed successfully.

For developer point of view server side is preferable because it will not fail, it is not dependent on browser and scripting language.

You can use ASP.NET validation, which will ensure client, and server validation. It work on both end; first it will work on client validation and than on server validation. At any cost server validation will work always whether client validation is executed or not. So you have a safety of validation check.

For client script .NET used JavaScript. WebUIValidation.js file is used for client validation by .NET

Validation Controls in ASP.NET

An important aspect of creating ASP.NET Web pages for user input is to be able to check that the information users enter is valid. ASP.NET provides a set of validation controls that provide an easy-to-use but powerful way to check for errors and, if necessary, display messages to the user.

There 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

The below table describes the controls and their work,

Validation Control	Description
RequiredFieldValidation	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

All validation controls are rendered in form as (label are referred as on client by server)

ASP.NET CompareValidator Control

This validator evaluates the value of an input control against another input control on the basis of specified operator.

We can use comparison operators like: less than, equal to, greater than etc.

Competitive questions on Structures in Hindi

Keep Watching

Note: If the input filed is empty, no validation will be performed.

CompareValidator Properties

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
BorderWidth	It is used to set width of border of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
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 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.
ControlToCompare	It takes ID of control to compare with.
ControlToValidate	It takes ID of control to validate.
ErrorMessage	It is used to display error message when validation failed.
Operator	It is used set comparison operator.

ASP.NET RangeValidator Control

This validator evaluates the value of an input control to check that the value lies between specified ranges.

It allows us to check whether the user input is between a specified upper and lower boundary. This range can be numbers, alphabetic characters and dates.

Note: if the input control is empty, no validation will be performed.

The **ControlToValidate** property is used to specify the control to validate.

The **MinimumValue** and **MaximumValue** properties are used to set minimum and maximum boundaries for the control.

RangeValidator Properties

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
TabIndex	The tab order of the control.
BackColor	It is used to set background color of the control.

BorderColor	It is used to set border color of the control.
BorderWidth	It is used to set width of border of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
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 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.
ControlToValidate	It takes ID of control to validate.
ErrorMessage	It is used to display error message when validation failed.
Type	It is used to set datatype of the control value.
MaximumValue	It is used to set upper boundary of the range.
MinimumValue	It is used to set lower boundary of the range.

ASP.NET RegularExpressionValidator Control

This validator is used to validate the value of an input control against the pattern defined by a regular expression.

It allows us to check and validate predictable sequences of characters like: e-mail address, telephone number etc.

The **ValidationExpression** property is used to specify the regular expression, this expression is used to validate input control.

RegularExpression Properties

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
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 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.
ErrorMessage	It is used to set error message that display when validation fails.
ControlToValidate	It takes ID of control to validate.
ValidationExpression	It is used to set regular expression to determine validity.

ASP.NET RequiredFieldValidator Control

This validator is used to make an input control required. It will throw an error if user leaves input control empty.

It is used to mandate form control required and restrict the user to provide data.

Note: It removes extra spaces from the beginning and end of the input value before validation is performed.

The ControlToValidateproperty should be set with the ID of control to validate.

RequiredFieldValidator Properties

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
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 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.
ErrorMessage	It is used to set error message that display when validation fails.

ControlToValidate

It takes ID of control to validate.

ASP.NET ValidationSummary Control

This validator is used to display list of all validation errors in the web form.

It allows us to summarize the error messages at a single location.

We can set **DisplayMode** property to display error messages as a list, bullet list or single paragraph.

ValidationSummary Properties

This control has following properties.

12.4M

202

Triggers in SQL (Hindi)

Property	Description
AccessKey	It is used to set keyboard shortcut for the control.
BackColor	It is used to set background color of the control.
BorderColor	It is used to set border color of the control.
Font	It is used to set font for the control text.
ForeColor	It is used to set color of the control text.
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 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.
ShowMessageBox	It displays a message box on error in up-level browsers.
ShowSummary	It is used to show summary text on the form page.
ShowValidationErrors	It is used to set whether the validation summary should be shown or not.

Important points for validation controls

- ControlToValidate property is mandatory to all validate controls.
- One validation control will validate only one input control but multiple validate control can be assigned to input control.

Program:

- **Default.aspx**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
<style>

    .main {
        width: 100%;
        height: 90vh;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
    }
    #TextBox1, #CustomValidator1, #RequiredFieldValidator1, #Button1, #Label1{
        margin: 10px auto;
    }
    div{
        text-align: center;
    }
    #Label1{
        font-size: 1.2rem;
        width: 50%;
    }

</style>
```

```

</head>
<body>
    <form id="form1" runat="server">
        <div class="main">
            <div>
                <h1>Practical-10</h1>
                <h4>Create a Web Form to demonstrate the use of ASP.Net Data Validation
Controls</h4>
            </div>

            <asp:Label ID="Label2" runat="server" Text="Enter userID with atleast 1 capital
alphabet, 1 small alphabet and 1 number"></asp:Label>
            <asp:Label ID="Label1" runat="server" Text="User ID:"></asp:Label>

            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

            <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="User id
required"></asp:RequiredFieldValidator>

            <asp:CustomValidator ID="CustomValidator1" runat="server"
OnServerValidate="UserCustomValidate" ControlToValidate="TextBox1" ErrorMessage="User ID
should have atleast a capital, small and digit and should be greater than 5 and less than
26 letters" SetFocusOnError="True"></asp:CustomValidator>

            <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Submit" />
        </div>
    </form>
</body>
</html>

```

- **Default.aspx.cs**

```

using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class _Default : System.Web.UI.Page
{
    protected void UserCustomValidate(object source, ServerValidateEventArgs args)
    {
        string str = args.Value;
        args.IsValid = false;
        //checking for input length greater than 6 and less than 25 characters
        if (str.Length < 6 || str.Length > 25)
        {
            return;
        }
        //checking for a atleast a single capital letter
        bool capital = false;
        foreach (char ch in str)
        {
            if (ch >= 'A' && ch <= 'Z')

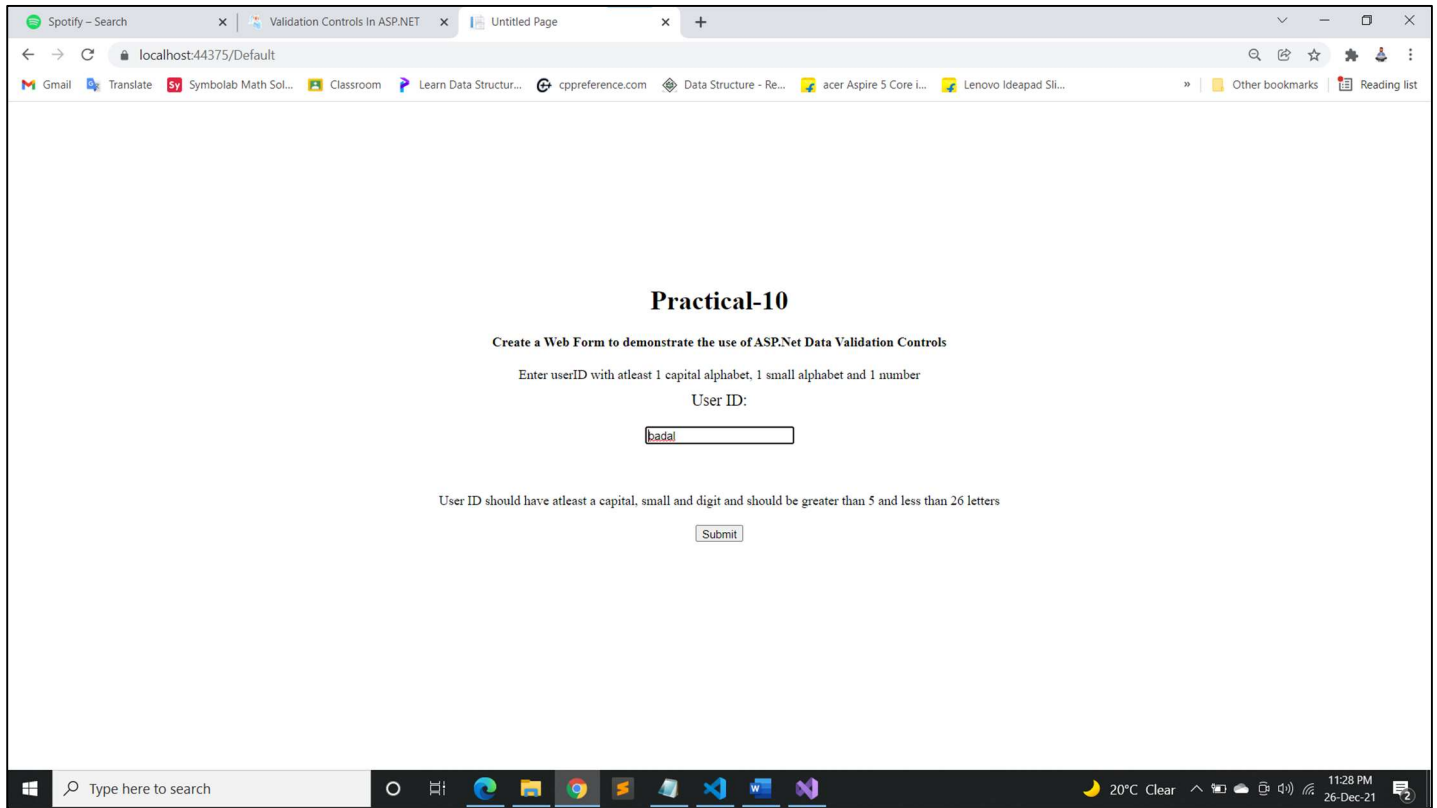
```

```

        {
            capital = true;
            break;
        }
    }
    if (!capital)
    {
        return;
    }
    //checking for a atleast a single lower letter
    bool lower = false;
    foreach (char ch in str)
    {
        if (ch >= 'a' && ch <= 'z')
        {
            lower = true;
            break;
        }
    }
    if (!lower)
    {
        return;
    }
    bool digit = false;
    foreach (char ch in str)
    {
        if (ch >= '0' && ch <= '9')
        {
            digit = true;
            break;
        }
    }
    if (!digit)
    {
        return;
    }
    args.IsValid = true;
}
protected void Page_Load(object sender, EventArgs e)
{
}
protected void Button1_Click(object sender, EventArgs e)
{
}
}

```

OUTPUT SCREEN SHOTS:



CONCLUSION:

I have studied about ASP.Net Data validation and successfully completed Practical-10.