**Remaining Work Custom Validator:**

**WFValidation.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WFValidation.aspx.cs" Inherits="PracticalNo2All.WFValidation" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label1" runat="server" Font-Size="Large" ForeColor="#CC0000" Text="Validation Demo"></asp:Label>

<br />

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label2" runat="server" Text="Enter Name"></asp:Label>

:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

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

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="txtName" Display="Dynamic" ErrorMessage="Name Required!!!" ForeColor="Red">Name is Mandatory!</asp:RequiredFieldValidator>

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label3" runat="server" Text="Password"></asp:Label>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtPass" runat="server" TextMode="Password"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="txtPass" Display="Dynamic" ErrorMessage="Password Please" ForeColor="Red">Password Mandatory!</asp:RequiredFieldValidator>

<asp:CustomValidator ID="CustomValidator2" runat="server" ControlToValidate="txtPass" Display="Dynamic" ErrorMessage="Check Password" OnServerValidate="CustomValidator2\_ServerValidate">Please Enter Valid Password</asp:CustomValidator>

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label4" runat="server" Text="Confirm Password"></asp:Label>

&nbsp;&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtConPass" runat="server" TextMode="Password"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="txtConPass" Display="Dynamic" ErrorMessage="Confirm Password please" ForeColor="Red">Confirm Password is Mandatory!</asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCompare="txtPass" ControlToValidate="txtConPass" Display="Dynamic" ErrorMessage="Password not same" ForeColor="#CC00FF"></asp:CompareValidator>

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label5" runat="server" Text="Enter Age:"></asp:Label>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtAge" runat="server" TextMode="Number"></asp:TextBox>

<asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="txtAge" Display="Dynamic" ErrorMessage="Age is Not in Age limit" ForeColor="#CC0000" MaximumValue="50" MinimumValue="18"></asp:RangeValidator>

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Label ID="Label6" runat="server" Text="Enter Email:"></asp:Label>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:TextBox ID="txtEmail" runat="server" TextMode="Email"></asp:TextBox>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="txtEmail" Display="Dynamic" ErrorMessage="Check Email ID" ForeColor="#CC0000" SetFocusOnError="True" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*">Please Enter Proper Email</asp:RegularExpressionValidator>

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<br />

<br />

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="btnSubmit" runat="server" Font-Size="Large" Text="Submit" OnClick="btnSubmit\_Click" />

<br />

<br />

<br />

</div>

</form>

</body>

</html>

**WFValidation.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace PracticalNo2All

{

public partial class WFValidation : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblCustm1.Text = "";

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

}

protected void CustomValidator2\_ServerValidate(object source, ServerValidateEventArgs args)

{

if (txtPass.Text.Length == 8)

{

string passWord = txtPass.Text;

if (!ValidatePassword(passWord))

{

lblCustm1.Text = "Please enter a valid Password with these constraints: One Special Character, One Lowercase, One Uppercase, and Any Digit 0-9.";

}

}

else

{

lblCustm1.Text = "Please enter a valid Password of length 8.";

}

}

static bool ValidatePassword(string passWord)

{

int validConditions = 0;

foreach (char c in passWord)

{

if (c >= 'a' && c <= 'z')

{

validConditions++;

break;

}

}

if (validConditions == 0) return false;

foreach (char c in passWord)

{

if (c >= 'A' && c <= 'Z')

{

validConditions++;

break;

}

}

if (validConditions == 1) return false;

foreach (char c in passWord)

{

if (c >= '0' && c <= '9')

{

validConditions++;

break;

}

}

if (validConditions == 2) return false;

// Check for special characters

char[] special = { '@', '#', '$', '%', '^', '&', '+', '=' };

foreach (char c in passWord)

{

if (special.Contains(c))

{

validConditions++;

break;

}

}

// Ensure that all conditions are met

return validConditions == 4;

}

}

}

**Output:**









