**Simple validation- exp 14&15**

**14)** **Develop a feature-rich ASP.NET web application that includes**

**various validation techniques and utilizes rich controls. The**

**application should have the following functionalities:**

**Form Validation:**

**a) Create a web page with a form that collects user registration**

**information (e.g., username, email, password).**

**b) Extend the registration form to include a custom validation**

**rule (e.g., mobile numbers must be unique and 10 digits).**

**Implement a custom validator to enforce this rule on the**

**server-side**

**15)** **Develop an ASP.NET web application that incorporates validation**

**controls and rich user interface elements. The application should**

**have the following features:**

**Form with Validation Controls:**

**a) Create a web form with various input fields (e.g., textboxes,**

**dropdowns, checkboxes).**

**b) Implement validation controls (e.g., RequiredFieldValidator,**

**RegularExpressionValidator) to ensure data integrity.**

**c) Include a custom validator for a specific validation scenario**

**Add->webform:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication2.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<asp:Label ID="Label1" runat="server" Text="Label">Enter Username</asp:Label>

<input id="Text1" runat="server" type="text" />

<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ForeColor="Red" ControlToValidate="Text1" ErrorMessage="Pls enter username"></asp:RequiredFieldValidator><br /><br />

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

<input id="Text2" runat="server" type="text" />

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="Text2" ForeColor="Red" ErrorMessage="Pls enter Email"></asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="Text2" ForeColor="Red" ValidationExpression="^[a-zA-Z0-9+\_.-]+@[a-zA-Z0-9.-]+$" ErrorMessage="Enter valid email"></asp:RegularExpressionValidator>

<br /><br />

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

<input id="Password1" runat="server" type="password" />

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="Password1" ForeColor="Red" ErrorMessage="Pls enter Password"></asp:RequiredFieldValidator>

<br /><br />

<asp:Label ID="Label4" runat="server" Text="Label">Enter Mobile Number</asp:Label>

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

<asp:CustomValidator ID="CustomValidator1" runat="server" ControlToValidate="TextBox1" ErrorMessage="Mobile numbers must be unique and 10 digits" OnServerValidate="ValidateMobile" ForeColor="Red"></asp:CustomValidator>

<br /><br />

<asp:Label ID="Label5" runat="server" Text="Label">Select Gender</asp:Label>

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem Text="Select" value="" />

<asp:ListItem Text="Male" value="M" />

<asp:ListItem Text="Female" value="F" />

<asp:ListItem Text="Other" value="O" />

</asp:DropDownList>

<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ControlToValidate="DropDownList1" ForeColor="Red" ErrorMessage="Pls Select"></asp:RequiredFieldValidator>

<br /><br />

<asp:Label ID="Label6" runat="server" Text="Label">Are you sure to submit? </asp:Label>

<asp:CheckBox ID="CheckBox1" runat="server" />

<br /><br />

<asp:Button ID="Button1" runat="server" Text="Submit" />

</div>

</form>

</body>

</html>

**Webform1.aspx->webform1.aspx.cs=**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication2

{

public partial class WebForm1 : System.Web.UI.Page

{

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

{

}

protected void ValidateMobile(object source, ServerValidateEventArgs args)

{

string mobileNumber = TextBox1.Text;

bool isValid = IsMobileValid(mobileNumber);

args.IsValid = isValid;

}

private bool IsMobileValid(string mobileNumber)

{

return Regex.IsMatch(mobileNumber, @"^\d{10}$");

}

}

}