Slip4

Q1. WAP to Validate student details Form using Validation Control

**Contact.aspx**

Page Title="Contact" Language="C#" MasterPageFile=\*-/Site. Master" AutoEventWireup="true" CodeBehind "Contact.aspx.cs Inherits="studentForm.Contact">

<asp:Content ID="BodyContent ContentPlaceHolderID="MainContent" runat="server"> <nain aria-labelledby="title">

<h2 id="title"><%: Title > &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="Labell" runat="server" Text="Registration Forn"></asp:Label>

</h2>

<p>UserName: &nbsp;&nbsp;&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; <asp:TextBox ID="TextBox1" runat="server" Height="18px"

Midth="146px"></asp:TextBox>

<asp:RequiredFieldValidator ID="RequiredFieldValidatorl" runat="server" ControlToValidate="TextBox1" Display \*Dynanic" ErrorMessage "RequiredFieldValidator" ForeColor="Red" SetFocusOnError="True">enter name</asp:RequiredFieldValidator>

</p>

<p>Password:

<asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>

</p>

<p>Confirm Password:

<asp:TextBox ID=\*TextBox7" runat="server"></asp:TextBox>

<asp:Comparevalidator ID="CompareValidatorl runat="server" ControlToCompare="TextBox6" ControlToValidate="TextBox7" Display="Dynamic"

ErrorMessage="Password and Confirn ForeColor="Red" SetFocusOnError="True">Password Missatched</asp:CompareValidator>

</p>

<p>Address:&nbsp;&nbsp;&nbsp;&nbsp; <asp:TextBox ID="TextBox2"

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

</p>

<p>DOB &nbsp; &nbsp; &nbsp; &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <asp:TextBox

ID="TextBox3" runat="server"></asp:TextBox>

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

ControlToValidate="TextBox3" Display="Dynamic" ErrorMessage="age between 18 to

45" ForeColor="Red" SetFocusOnError="True" Type=\*Date">Age not in

range</asp: RangeValidator>

</p>

p>Email ID :&nbsp;&nbsp;&nbsp;&nbsp;

<

<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>

<asp: RegularExpressionValidator ID="RegularExpressionValidatorl

runat="server" ControlToValidate="TextBox4" Display="Dynamic" Errorfñessages enail id is incorrect format ForeColor="#FF3300" SetFocus On Error="True"

ValidationExpression=\*\w+([-.]\w+)\*@\w+([-.]\w+)\*\.\w+([-]\w+)\*\*>invalid

enailid</asp: RegularExpressionValidator>

</p>

<asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>

<asp: RegularExpressionValidator ID="RegularExpressionValidator2"

<p>Mobile No:

runat="server" ControlToValidate="TextBox5" Display="Dynamic

ErrorMessage="RegularExpressionValidator" ForeColor="Red" SetFocusOnError="True" ValidationExpression=\*\d{10}">invalid number</asp: RegularExpressionValidator>

|  |  |
| --- | --- |
|  |  |

</p>

<p>

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

</p>

</main>

</asp:Content>

**contact.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace studentForn

{

public partial class Contact: Page {

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

{ RangeValidatorl. MinimumValue = DateTime.Now. AddYears(- ToShortDateString(); 45).

RangeValidatorl. MaximumValue = DateTime.Now.AddYears(-18) ToShortDateString();

}

}

}

|  |  |
| --- | --- |
|  |  |

**Q2.** Create the application that accepts name. password . age . email id. and user id. All the information entry is compulsory Password should reconfirmed\_ Age should within 21 to 30. Eman id should valid, User id should have at least a capital and digit a,' well as length should between 7 and 20 characters.

**Aspx:**

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

<!DOCTYPE html>

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

<head runat="server">

<title>User Registration</title>

</head>

<body>

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

<div>

<h2>User Registration Form</h2>

<div>

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

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

</div>

<div>

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

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

</div>

<!-- Add similar markup for Age, Email ID, and User ID -->

<!-- You can also add validation controls for each field -->

<div>

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

</div>

</div>

</form>

</body>

</html>

**Aspx.cs:**

**using System;**

**using System.Web.UI;**

**namespace YourNamespace**

**{**

**public partial class RegistrationForm : Page**

**{**

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

**{**

**}**

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

**{**

**// Retrieve values from textboxes**

**string name = txtName.Text;**

**string password = txtPassword.Text;**

**int age = Convert.ToInt32(txtAge.Text);**

**string email = txtEmail.Text;**

**string userId = txtUserId.Text;**

**// Validate age**

**if (age < 21 || age > 30)**

**{**

**// Display error message for invalid age**

**// You can use a label to display error messages**

**lblError.Text = "Age must be between 21 and 30.";**

**return;**

**}**

**// Validate email format**

**if (!IsValidEmail(email))**

**{**

**lblError.Text = "Invalid email address.";**

**return;**

**}**

**// Validate user ID format**

**if (!IsValidUserId(userId))**

**{**

**lblError.Text = "User ID must have at least one capital letter, one digit, and length between 7 and 20 characters.";**

**return;**

**}**

**// If all validations pass, you can proceed with storing the user data**

**// You can implement database insertion or any other desired action**

**// For simplicity, let's just display a success message**

**lblError.Text = "Registration successful!";**

**}**

**private bool IsValidEmail(string email)**

**{**

**// Implement email validation logic**

**// You can use regular expressions or any other method to validate email format**

**// For simplicity, let's just check if it contains '@'**

**return email.Contains("@");**

**}**

**private bool IsValidUserId(string userId)**

**{**

**// Implement user ID validation logic**

**// You can use regular expressions or any other method to validate user ID format**

**// For simplicity, let's check if it has at least one capital letter, one digit, and length between 7 and 20 characters**

**bool hasCapital = false;**

**bool hasDigit = false;**

**foreach (char c in userId)**

**{**

**if (char.IsUpper(c))**

**hasCapital = true;**

**if (char.IsDigit(c))**

**hasDigit = true;**

**}**

**return hasCapital && hasDigit && userId.Length >= 7 && userId.Length <= 20;**

**}**

**}**

**}**