Slip 9

Q1.Write program on state management

. **.aspx**

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

<!DOCTYPE html>

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

<head runat="server">

<title>State Management Demo</title>

</head>

<body>

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

<div>

<asp:Label ID="lblCounter" runat="server" Text="Counter:"></asp:Label>

<br />

<asp:Button ID="btnIncrement" runat="server" Text="Increment" OnClick="btnIncrement\_Click" />

<asp:Button ID="btnReset" runat="server" Text="Reset" OnClick="btnReset\_Click" />

</div>

</form>

</body>

</html>

**aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System;

namespace StateManagementDemo

{

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

{

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

{

if (!IsPostBack)

{

// Check if session variable exists

if (Session["Counter"] == null)

{

// If not, initialize it

Session["Counter"] = 0;

}

// Display current value of the counter

lblCounter.Text = "Counter: " + Session["Counter"].ToString();

}

}

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

{

// Increment the counter value stored in session

Session["Counter"] = Convert.ToInt32(Session["Counter"]) + 1;

// Display updated counter value

lblCounter.Text = "Counter: " + Session["Counter"].ToString();

}

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

{

// Reset the counter value stored in session

Session["Counter"] = 0;

// Display reset counter value

lblCounter.Text = "Counter: " + Session["Counter"].ToString();

}

}

}

**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)**

**{**

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

**}**

**private bool IsValidUserId(string userId)**

**{**

**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;**

**}**

**}**

**}**