**1. Create a Windows application in C# for registration form and fill the details and when you click the submit button, it displays all the details. irstname, lastname, middlename, gender, city, hobby, image, email id, password, phone no, about us.**

**Code:- ASPX.CS:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace program1

{

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

{

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

{

}

protected void TextBox1\_TextChanged(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

// Retrieve values from the form fields

}

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

{

string firstName = textBoxFirstName.Text;

string lastName = textBoxLastName.Text;

string middleName = textBoxMiddleName.Text;

string gender = comboBoxGender.SelectedValue;

string city = textBoxCity.Text;

string hobby = textBoxHobby.Text;

string email = textBoxEmail.Text;

string password = textBoxPassword.Text;

string phoneNo = textBoxPhoneNo.Text;

string aboutUs = textBoxAboutUs.Text;

// Validate that all fields are filled

if (string.IsNullOrEmpty(firstName) || string.IsNullOrEmpty(lastName) ||

string.IsNullOrEmpty(gender) || string.IsNullOrEmpty(city) ||

string.IsNullOrEmpty(hobby) || string.IsNullOrEmpty(email) ||

string.IsNullOrEmpty(password) || string.IsNullOrEmpty(phoneNo) ||

string.IsNullOrEmpty(aboutUs))

{

labelResult.Text = "All fields are required.";

labelResult.ForeColor = System.Drawing.Color.Red;

}

else

{

labelResult.Text = $"Registration successful!<br />" +

$"First Name: {firstName}<br />" +

$"Last Name: {lastName}<br />" +

$"Middle Name: {middleName}<br />" +

$"Gender: {gender}<br />" +

$"City: {city}<br />" +

$"Hobby: {hobby}<br />" +

$"Email ID: {email}<br />" +

$"Phone No: {phoneNo}<br />" +

$"About Us: {aboutUs}";

labelResult.ForeColor = System.Drawing.Color.Green;

}

}

}

}

**WEB FORM:**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="program1.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<h2>Registration Form</h2>

<label>First Name:</label>

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

<label>

<br />

Last Name:</label>

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

<label>

<br />

Middle Name:</label>

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

<label>

<br />

Gender:</label>

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

<asp:ListItem Value="">Select</asp:ListItem>

<asp:ListItem Value="Male">Male</asp:ListItem>

<asp:ListItem Value="Female">Female</asp:ListItem>

<asp:ListItem Value="Other">Other</asp:ListItem>

</asp:DropDownList>

<label>

<br />

City:</label>

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

<label>

<br />

Hobby:</label>

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

<label>

<br />

Email ID:</label>

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

<label>

<br />

Password:</label>

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

<label>

<br />

Phone No:</label>

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

<label>

<br />

About Us:</label>

<asp:TextBox ID="textBoxAboutUs" runat="server" TextMode="MultiLine" Rows="4"></asp:TextBox>

<br />

<br />

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

<br />

<br />

<asp:Label ID="labelResult" runat="server" CssClass="result" ForeColor="Green"></asp:Label>

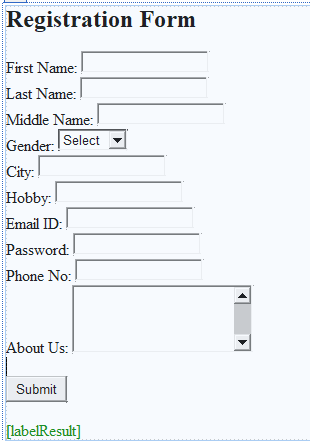
</div>

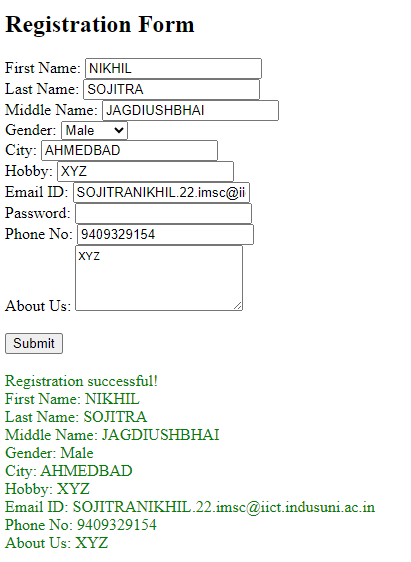
</form>

</body>

</html>

**DESIGN:**

****



**2. Change listbox values from listbox1 to listbox2 and vice versa.**

**ASPX.CS**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace program2

{

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

{

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

{

}

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

{

foreach (ListItem item in listBox2.Items)

{

if (item.Selected)

{

listBox1.Items.Add(new ListItem(item.Text, item.Value));

}

}

// Remove selected items from listBox2

for (int i = listBox2.Items.Count - 1; i >= 0; i--)

{

if (listBox2.Items[i].Selected)

{

listBox2.Items.RemoveAt(i);

}

}

}

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

{

foreach (ListItem item in listBox1.Items)

{

if (item.Selected)

{

listBox2.Items.Add(new ListItem(item.Text, item.Value));

}

}

// Remove selected items from listBox1

for (int i = listBox1.Items.Count - 1; i >= 0; i--)

{

if (listBox1.Items[i].Selected)

{

listBox1.Items.RemoveAt(i);

}

}

}

}

}

**WEBFORM**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="program2.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

<title></title>

</head>

<body>

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

<div>

<h2>ListBox Transfer Example</h2>

<asp:ListBox ID="listBox1" runat="server" CssClass="list-box" SelectionMode="Multiple">

<asp:ListItem Value="Item 1">Item 1</asp:ListItem>

<asp:ListItem Value="Item 2">Item 2</asp:ListItem>

<asp:ListItem Value="Item 3">Item 3</asp:ListItem>

<asp:ListItem Value="Item 4">Item 4</asp:ListItem>

</asp:ListBox>

<div>

<asp:Button ID="buttonToRight" runat="server" Text=">>" OnClick="buttonToRight\_Click" CssClass="button" />

<asp:Button ID="buttonToLeft" runat="server" Text="<<" OnClick="buttonToLeft\_Click" CssClass="button" />

</div>

<asp:ListBox ID="listBox2" runat="server" CssClass="list-box" SelectionMode="Multiple">

<asp:ListItem Value="Item A">Item A</asp:ListItem>

<asp:ListItem Value="Item B">Item B</asp:ListItem>

<asp:ListItem Value="Item C">Item C</asp:ListItem>

<asp:ListItem Value="Item D">Item D</asp:ListItem>

</asp:ListBox>

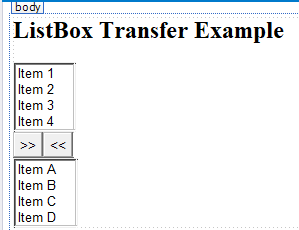
</div>

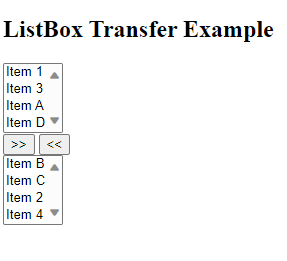
</form>

</body>

</html>

**DESIGN**



**OUTPUT**  


**3. Create a rich text box with customised output.**

**CODE**

**ASPX.CS**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

richTextBox1.Font = new Font(richTextBox1.Font, FontStyle.Bold);

}

private void button1\_Click(object sender, EventArgs e)

{

richTextBox1.Font = new Font(richTextBox1.Font, FontStyle.Italic);

}

private void button4\_Click(object sender, EventArgs e)

{

richTextBox1.Clear();

}

private void button3\_Click(object sender, EventArgs e)

{

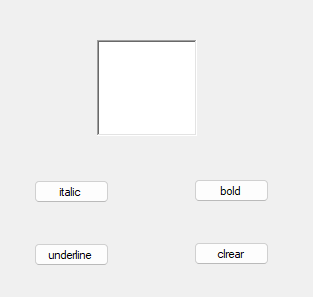
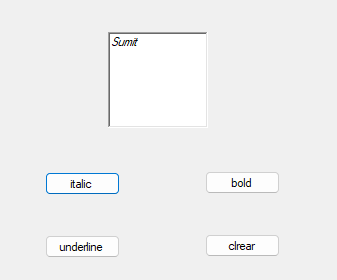
richTextBox1.Font = new Font(richTextBox1.Font, FontStyle.Underline);

}

}

}

**DESIGN**  **OUTPUT**



**4. Create a MDI form. If button 1 clicked , form1 opened, button 2 clicked form2 opened, button 3 clicked form3 opened and in this do addition and subtraction.**

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace program4

{

public partial class Form1: Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.MdiParent = this; // Set Parent

form2.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

Form3 form3 = new Form3();

form3.MdiParent = this; // Set Parent

form3.Show();

}

}

}

**Form2.cs**

using System;

using System.Windows.Forms;

namespace program4

{

public class Form2 : Form

{

public Form2()

{

InitializeComponent(); // This should be inside the constructor

}

private void InitializeComponent()

{

this.Text = "Form2"; // Set Form Title

this.Size = new System.Drawing.Size(300, 200); // Set Form Size

}

}

}

**Form3.cs**

using System;

using System.Windows.Forms;

namespace program4

{

public class Form3 : Form

{

// Declare UI components

private TextBox textBox1, textBox2;

private Button btnAdd, btnSubtract;

private Label labelResult;

public Form3()

{

this.Text = "Addition & Subtraction"; // Set Form title

this.Size = new System.Drawing.Size(300, 200); // Set Form size

InitializeUI(); // Create UI elements dynamically

}

private void InitializeUI()

{

// Create TextBox1

textBox1 = new TextBox();

textBox1.Location = new System.Drawing.Point(20, 20);

textBox1.Width = 100;

this.Controls.Add(textBox1);

// Create TextBox2

textBox2 = new TextBox();

textBox2.Location = new System.Drawing.Point(20, 50);

textBox2.Width = 100;

this.Controls.Add(textBox2);

// Create Add Button

btnAdd = new Button();

btnAdd.Text = "Add";

btnAdd.Location = new System.Drawing.Point(20, 80);

btnAdd.Click += new EventHandler(btnAdd\_Click);

this.Controls.Add(btnAdd);

// Create Subtract Button

btnSubtract = new Button();

btnSubtract.Text = "Subtract";

btnSubtract.Location = new System.Drawing.Point(100, 80);

btnSubtract.Click += new EventHandler(btnSubtract\_Click);

this.Controls.Add(btnSubtract);

// Create Result Label

labelResult = new Label();

labelResult.Text = "Result: ";

labelResult.Location = new System.Drawing.Point(20, 120);

labelResult.Width = 200;

this.Controls.Add(labelResult);

}

private void btnAdd\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text, out int num1) && int.TryParse(textBox2.Text, out int num2))

{

labelResult.Text = "Sum: " + (num1 + num2);

}

else

{

MessageBox.Show("Please enter valid numbers.");

}

}

private void btnSubtract\_Click(object sender, EventArgs e)

{

if (int.TryParse(textBox1.Text, out int num1) && int.TryParse(textBox2.Text, out int num2))

{

labelResult.Text = "Difference: " + (num1 - num2);

}

else

{

MessageBox.Show("Please enter valid numbers.");

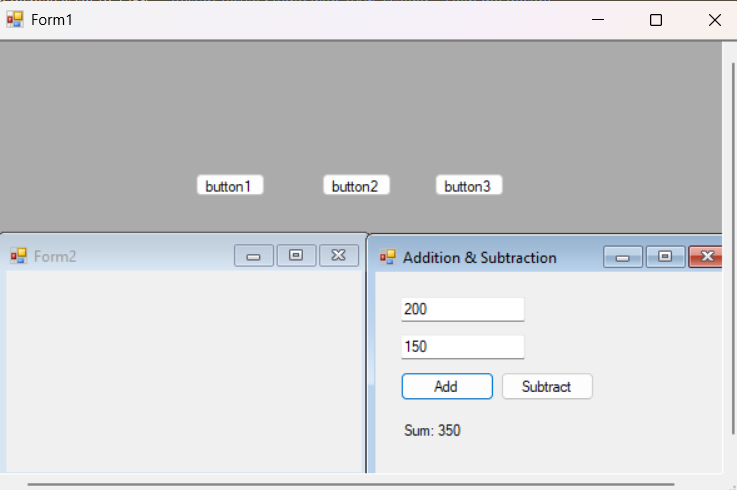
}

}

}

}

**DESIGN** **OUTPUT**



**5. Display date and time using calendar control.**

**Form1.cs**

using System;

using System.Windows.Forms;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.Load += new System.EventHandler(this.Form1\_Load); // ✅ Fix 1: Link Form Load Event

}

private void Form1\_Load(object sender, EventArgs e)

{

timer1.Interval = 1000; // 1 second

timer1.Tick += Timer1\_Tick;

timer1.Start();

label1.Text = "Current Time: " + DateTime.Now.ToString("hh:mm:ss tt");

}

private void Timer1\_Tick(object sender, EventArgs e)

{

label1.Text = "Current Time: " + DateTime.Now.ToString("hh:mm:ss tt");

}

private void monthCalendar1\_DateChanged(object sender, DateRangeEventArgs e)

{

label1.Text = "Selected Date: " + e.Start.ToShortDateString() + " | Time: " + DateTime.Now.ToString("hh:mm:ss tt");

}

private void label1\_Click(object sender, EventArgs e)

{

// ✅ Fix 2: Empty Event Handler for label1\_Click

}

private void timer1\_Tick(object sender, EventArgs e)

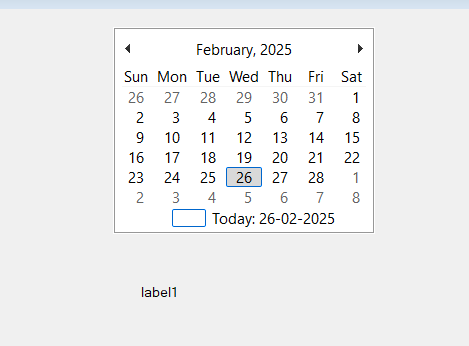
{

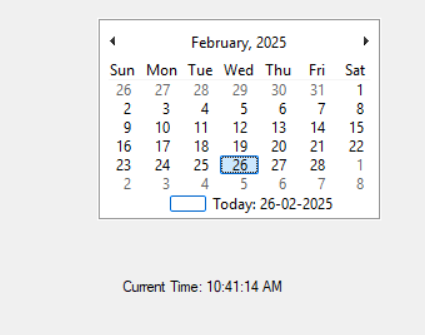
Timer1\_Tick(sender, e); // ✅ Fix 2: Call Timer1\_Tick

}

}

}

**DESIGN** **OUTPUT**



**6. Create a form for checking if a given number is odd or even.**

**FORM1.cs**

using System;

using System.Windows.Forms;

namespace program6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnCheck\_Click(object sender, EventArgs e)

{

// Check if the input is a valid integer

if (int.TryParse(txtNumber.Text, out int number))

{

// Check if the number is Even or Odd

if (number % 2 == 0)

{

lblResult.Text = $"{number} is an Even number.";

lblResult.ForeColor = System.Drawing.Color.Green;

}

else

{

lblResult.Text = $"{number} is an Odd number.";

lblResult.ForeColor = System.Drawing.Color.Red;

}

}

else

{

lblResult.Text = "Please enter a valid integer.";

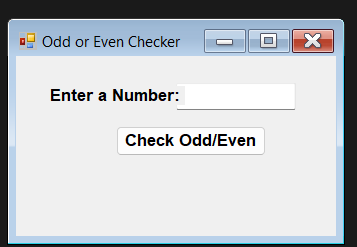
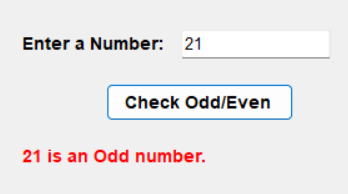
lblResult.ForeColor = System.Drawing.Color.Orange;

}

}

}

}

**DESIGN** **OUTPUT**

**7. Store user value in a hidden field and display it in another control.**

**FORM1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace prgram7

{

public partial class Form1: Form

{

public Form1()

{

InitializeComponent();

}

private void btnStore\_Click(object sender, EventArgs e)

{

// Store user input in the hidden field

txtHiddenField.Text = txtUserInput.Text;

}

private void btnShow\_Click(object sender, EventArgs e)

{

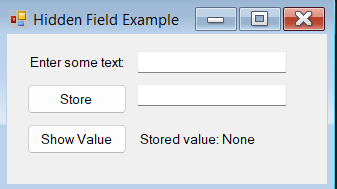
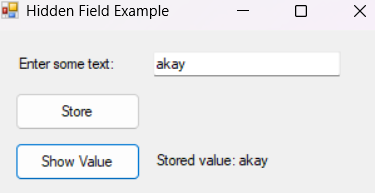
// Display hidden field value in the label

lblDisplay.Text = "Stored value: " + txtHiddenField.Text;

}

}

}

**DESIGN** **OUTPUT**

**9. Create a window form by using combobox control to add, delete , count and select the items.**

**Form1.cs**

using System;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Tab;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace program9

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

// Add Item to ComboBox

private void btnAdd\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrWhiteSpace(txtItem.Text))

{

comboBox1.Items.Add(txtItem.Text);

txtItem.Clear();

}

else

{

MessageBox.Show("Please enter an item.");

}

}

// Delete Selected Item

private void btnDelete\_Click(object sender, EventArgs e)

{

if (comboBox1.SelectedItem != null)

{

comboBox1.Items.Remove(comboBox1.SelectedItem);

}

else

{

MessageBox.Show("Please select an item to delete.");

}

}

// Count Items in ComboBox

private void btnCount\_Click(object sender, EventArgs e)

{

lblResult.Text = "Total Items: " + comboBox1.Items.Count;

}

// Select an Item

private void btnSelect\_Click(object sender, EventArgs e)

{

if (comboBox1.SelectedItem != null)

{

lblResult.Text = "Selected Item: " + comboBox1.SelectedItem.ToString();

}

else

{

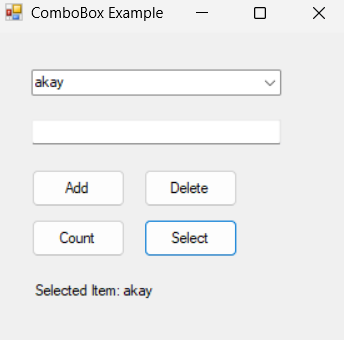
MessageBox.Show("Please select an item.");

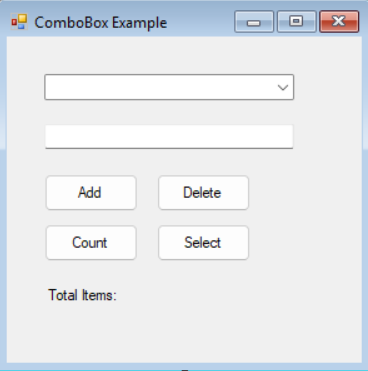
}

}

}

}

**DESIGN** **OUTPUT**



**10.Create a web form by using listbox control to add, delete , count and select the items.**

**Webform.axps**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="program10.WebForm1" %>

<!DOCTYPE html>

<html>

<head>

<title>ListBox Example</title>

</head>

<body>

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

<div>

<asp:ListBox ID="lstItems" runat="server" SelectionMode="Multiple"></asp:ListBox>

<br /><br />

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

<asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd\_Click" />

<asp:Button ID="btnDelete" runat="server" Text="Delete" OnClick="btnDelete\_Click" />

<asp:Button ID="btnCount" runat="server" Text="Count Items" OnClick="btnCount\_Click" />

<asp:Button ID="btnSelect" runat="server" Text="Show Selected" OnClick="btnSelect\_Click" />

<br /><br />

<asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>

</div>

</form>

</body>

</html>

**Webform.cs**

using System;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace program10

{

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

{

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

{

if (!IsPostBack)

{

// Add some default items to the ListBox

lstItems.Items.Add("Apple");

lstItems.Items.Add("Banana");

lstItems.Items.Add("Orange");

}

}

// Add Item to ListBox

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

{

if (!string.IsNullOrWhiteSpace(txtItem.Text))

{

lstItems.Items.Add(txtItem.Text);

lblMessage.Text = "Item added successfully!";

}

else

{

lblMessage.Text = "Please enter an item.";

}

}

// Delete Selected Item

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

{

if (lstItems.SelectedIndex != -1)

{

lstItems.Items.Remove(lstItems.SelectedItem);

lblMessage.Text = "Item deleted successfully!";

}

else

{

lblMessage.Text = "Please select an item to delete.";

}

}

// Count Items in ListBox

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

{

lblMessage.Text = "Total items: " + lstItems.Items.Count.ToString();

}

// Show Selected Item(s)

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

{

if (lstItems.SelectedIndex != -1)

{

lblMessage.Text = "Selected Item(s): ";

foreach (ListItem item in lstItems.Items)

{

if (item.Selected)

{

lblMessage.Text += item.Text + " ";

}

}

}

else

{

lblMessage.Text = "No item selected.";

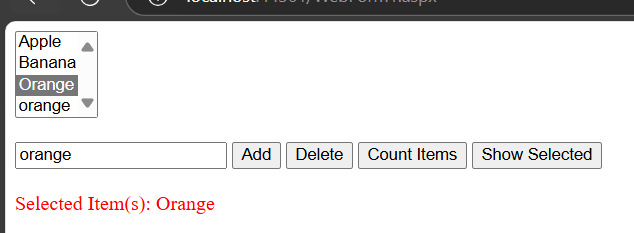
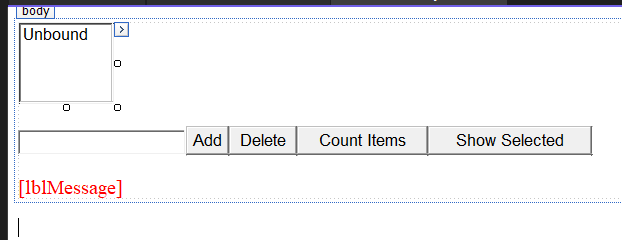
}

}

}

}

**DESIGN** **OUTPUT**



**11. Create a web form to use AD Rotator in asp.net .**

**Webform.aspx**

<%@ 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>

</div>

<asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1" Height="200px" Width="200px" />

<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile="~/XMLFile1.xml"></asp:XmlDataSource>

</form>

</body>

</html>

**Xml**

<?xml version="1.0" encoding="utf-8" ?>

<Advertisements>

<Ad>

<ImageUrl>img/ad1.jpg</ImageUrl>

<NavigateUrl>https://indusacademy.com</NavigateUrl>

<AlternateText>Indus Academy</AlternateText>

<Impressions>50</Impressions>

<Keyword>Meera</Keyword>

</Ad>

<Ad>

<ImageUrl>img/ad2.jpg</ImageUrl>

<NavigateUrl>http://academic.com</NavigateUrl>

<AlternateText>University</AlternateText>

<Impressions>100</Impressions>

<Keyword>Academy</Keyword>

</Ad>

<Ad>

<ImageUrl>img/ad3.jpg</ImageUrl>

<NavigateUrl>https:// indusacademy.com </NavigateUrl>

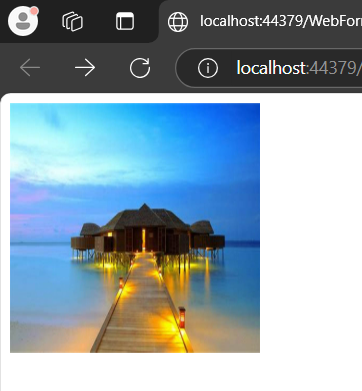
<AlternateText>Indus Aademy</AlternateText>

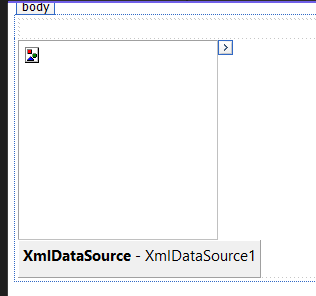
<Impressions>50</Impressions>

<Keyword>Academy</Keyword>

</Ad>

</Advertisements>

**DESIGN** **OUTPUT**



**12. Create a web form using all controls of Validation Control.**

**Webform.aspx**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ValidationExample.aspx.cs" Inherits="WebApplication1.ValidationExample" %>

<!DOCTYPE html>

<html lang="en">

<head runat="server">

<title>Validation Example</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

</head>

<body>

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

<div>

<h2>ASP.NET Validation Controls Example</h2>

<!-- RequiredFieldValidator -->

<label for="txtName">Name:</label>

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

<asp:RequiredFieldValidator ID="rfvName" runat="server"

ControlToValidate="txtName"

ErrorMessage="Name is required!"

ForeColor="Red">\*</asp:RequiredFieldValidator>

<br /><br />

<!-- CompareValidator (Check Age is a Number) -->

<label for="txtAge">Age:</label>

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

<asp:CompareValidator ID="cvAge" runat="server"

ControlToValidate="txtAge"

Operator="DataTypeCheck"

Type="Integer"

ErrorMessage="Age must be a valid number!"

ForeColor="Red">\*</asp:CompareValidator>

<br /><br />

<!-- RangeValidator (Restrict Age between 18-60) -->

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

ControlToValidate="txtAge"

MinimumValue="18" MaximumValue="60"

Type="Integer"

ErrorMessage="Age must be between 18 and 60!"

ForeColor="Red">\*</asp:RangeValidator>

<br /><br />

<!-- RegularExpressionValidator (Email Format) -->

<label for="txtEmail">Email:</label>

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

<asp:RegularExpressionValidator ID="revEmail" runat="server"

ControlToValidate="txtEmail"

ValidationExpression="^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$"

ErrorMessage="Enter a valid email address!"

ForeColor="Red">\*</asp:RegularExpressionValidator>

<br /><br />

<!-- CustomValidator (Check for Specific Value) -->

<label for="txtCustom">Enter 'Yes' or 'No':</label>

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

<asp:CustomValidator ID="cvCustom" runat="server"

ControlToValidate="txtCustom"

ErrorMessage="Only 'Yes' or 'No' allowed!"

OnServerValidate="cvCustom\_ServerValidate"

ForeColor="Red">\*</asp:CustomValidator>

<br /><br />

<!-- ValidationSummary (Show all Errors) -->

<asp:ValidationSummary ID="vsSummary" runat="server"

ForeColor="Red"

HeaderText="Validation Errors:" />

<br /><br />

<!-- Submit Button -->

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

</div>

</form>

</body>

</html>

**Webform.cs**

using System;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication1

{

public partial class ValidationExample : Page

{

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

{

}

// Custom Validator Logic (Validates that input is 'Yes' or 'No')

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

{

if (args.Value.ToLower() == "yes" || args.Value.ToLower() == "no")

{

args.IsValid = true;

}

else

{

args.IsValid = false;

}

}

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

{

if (Page.IsValid)

{

// Show a success message

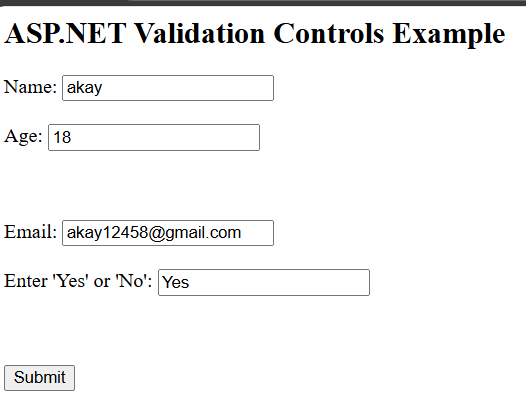
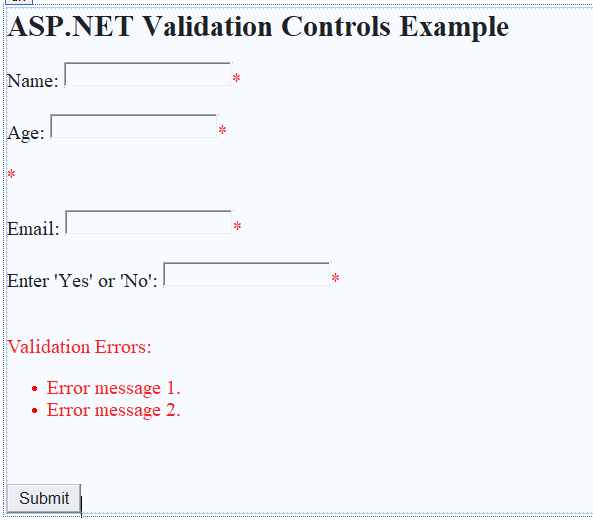
Response.Write("<script>alert('Form submitted successfully!');</script>");

}

}

}

}

**DESIGN**