PRACTICAL NO. 1: Introduction to ASP.NET

Aim: To develop simple ASP.NET web pages demonstrating data types, variables, operators, ASP.NET objects, server-side controls, cross-page posting, postback, and autopostback.

1. Data Types in ASP.NET

Value Types: Store the actual value and are stored on the stack. They are typically more efficient but have limited flexibility. Examples: int, float, bool, struct.

Reference Types: Store a reference to the actual data (i.e., a pointer) and are stored on the heap. They allow for more complex structures and behaviors. Examples: string, class, object, array. Value Types:

Data Type	Description	Example
int	Represents a 32-bit signed integer.	int age = 25;
float	Represents a single-precision floating point number.	float temperature = 36.5f;
double	Represents a double-precision floating point number.	double price = 19.99;
decimal	Represents a 128-bit precise decimal number, suitable for financial calculations.	decimal totalAmount = 12345.67M;
char	Represents a single Unicode character.	char grade = 'A';
bool	Represents a Boolean value (true/false).	bool isActive = true;
byte	Represents an 8-bit unsigned integer.	byte byteValue = 255;
short	Represents a 16-bit signed integer.	short shortNumber = 32767;
long	Represents a 64-bit signed integer.	long population = 7834000000L;
sbyte	Represents an 8-bit signed integer.	sbyte smallValue = -120;
ushort	Represents a 16-bit unsigned integer.	ushort ushortValue = 65535;
uint	Represents a 32-bit unsigned integer.	uint uintValue = 100000U;
ulong	Represents a 64-bit unsigned integer.	ulong largeValue = 1000000000UL;
Guid	Represents a globally unique identifier (GUID).	Guid guid = Guid.NewGuid();
DateTime	Represents an instance of time.	DateTime currentDate = DateTime.Now;
TimeSpan	Represents a time interval.	TimeSpan timeSpan = new TimeSpan(1, 2, 3, 4);

Reference Types

Data Type	Description	Example
string	Represents a sequence of characters (immutable).	string name = "John Doe";
object	The base type for all data types in .NET.	object obj = "Hello World";
Array	Represents a fixed-size collection of elements of the same type.	int[] numbers = new int[5];
Class	Defines a reference type used for creating objects.	public class Person { public string Name; }
Delegate	Represents a reference type that can hold a reference to a method.	public delegate void MyDelegate();
Interface	Defines a contract that classes can implement.	public interface IShape { void

		Draw(); }

2. Variables in ASP.NET:

Variables store data in memory during the execution of a program.

Declaring Variables int age = 25; string name = "John"; bool isActive = true; double price = 99.99;

Types of Variables

Local Variables – Declared inside a function or block.

Global Variables – Declared at the class level and accessible throughout the class.

Static Variables – Shared among all instances of a class (static int count;).

Constant Variables – Value remains unchanged (const double PI = 3.14;).

ReadOnly Variables – Can be assigned only in the constructor (readonly int id;).

3. Operators in ASP.NET:

Operators perform operations on variables and values.

Operator	Category	Description	Example
+	Arithmetic	Adds two operands.	int result = 5 + 3;
-	Arithmetic	Subtracts the second operand from the first.	int result = 5 - 3;
*	Arithmetic	Multiplies two operands.	int result = 5 * 3;
/	Arithmetic	Divides the numerator by the denominator.	int result = 6 / 3;
%	Arithmetic	Returns the remainder of a division operation.	int result = 5 % 3;
==	Comparison	Compares if two values are equal.	if (x == y)
!=	Comparison	Compares if two values are not equal.	if (x != y)
>	Comparison	Compares if the left operand is greater than the right.	if (x > y)
<	Comparison	Compares if the left operand is less than the right.	if (x < y)
>=	Comparison	Compares if the left operand is greater than or equal to the right.	if (x >= y)
<=	Comparison	Compares if the left operand is less than or equal to the right.	if (x <= y)
&&	Logical	Returns true if both operands are true.	if (x > 5 && y < 10)
**		**	Logical
!	Logical	Reverses the logical state of its operand.	if (!(x > 5))
=	Assignment	Assigns the right operand to the left operand.	x = 5;
+=	Assignment	Adds the right operand to the left operand and assigns the result.	x += 5; // x = x + 5
-=	Assignment	Subtracts the right operand from the left operand and assigns the result.	x -= 3; // x = x - 3
*=	Assignment	Multiplies the left operand by the right operand and assigns the result.	x *= 2; // x = x * 2
/=	Assignment	Divides the left operand by the right operand and assigns the result.	x /= 2; // x = x / 2
% =	Assignment	Assigns the remainder of the division of the operands.	x %= 2; // x = x % 2
?:	Conditional (Ternary)	A shortcut for if-else statement.	int result = (x > 5) ? 1 : 0;

&	Bitwise	Performs a bitwise AND operation.	x & y
**	**	Bitwise	Performs a bitwise OR operation.
^	Bitwise	Performs a bitwise XOR operation.	x ^ y
~	Bitwise	Performs a bitwise NOT operation.	~x
<<	Bitwise	Shifts the bits of the left operand to the left by the number of positions specified by the right operand.	x << 2
Operator	Category	Description	Example
>>	Bitwise	Shifts the bits of the left operand to the right by the number of positions specified by the right operand.	x >> 2
??	Null-Coalescing	Returns the left operand if it is not null, otherwise returns the right operand.	int? result = x ?? 10;

4. ASP.NET Objects

ASP.NET provides several built-in objects:

a) Request Object

Used to get values from the user (form data, query strings, cookies).

```
string userName = Request.QueryString["name"]; string userEmail
= Request.Form["email"];
```

b) Response Object

Used to send data to the client (browser). Response.Write("Hello,

```
World!");
Response.Redirect("Home.aspx");
```

c) Session Object

Used to store user-specific data across multiple pages. Session["UserID"] =

```
123;
string userId = Session["UserID"].ToString();
```

d) Application Object

Used to store global data shared across all users. Application["SiteVisitors"] =

```
100;
```

e) Server Object:

Provides utility functions such as URL encoding and transferring requests. string

```
encodedUrl = Server.UrlEncode("http://example.com");
Server.Transfer("Home.aspx");
```

5. Basic server controls:

Following basic controls used to collect and display information:

Textbox: Allows the user to input text.

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

Button: Triggers events like submitting a form.

```
<asp:Button ID="btnSubmit" Text="Submit" runat="server" OnClick="btnSubmit_Click" />
```

Label: Displays static or dynamic text.

```
<asp:Label ID="lblMessage" runat="server" Text="Hello, World!"></asp:Label>
```

HyperLink: Creates a hyperlink.

<asp:HyperLink ID="lnkWebsite" runat="server" NavigateUrl="https://famt.ac.in">Visit Site</asp:HyperLink>

Image: Displays an image.

<asp:Image ID="imgLogo" runat="server“ ImageUrl="~/Images/logo.png" />

LinkButton: Displays a clickable link that triggers a server-side event.

<asp:LinkButton ID="lnkButton" runat="server" OnClick="lnkButton_Click">Click Me</asp:LinkButton>

ImageButton: Displays an image that acts as a button.

<asp:ImageButton ID="imgButton" ImageUrl="~/images/submit.png" runat="server“ OnClick="imgButton_Click" />

DropDownList: Displays a dropdown menu.

<asp:DropDownList ID="ddlOptions" runat="server"> <asp:ListItem Text="Option 1" Value="1"></asp:ListItem> <asp:ListItem Text="Option 2" Value="2"></asp:ListItem> </asp:DropDownList>

ListBox: Displays a list of items where multiple selections can be made.

<asp:ListBox ID="lstItems" runat="server" SelectionMode="Multiple"> <asp:ListItem Text="Item 1" Value="1"></asp:ListItem> <asp:ListItem Text="Item 2" Value="2"></asp:ListItem> </asp:ListBox>

CheckBoxList: Displays a group of checkboxes.

<asp:CheckBoxList ID="cblOptions" runat="server"> <asp:ListItem Text="Option 1" Value="1"></asp:ListItem> <asp:ListItem Text="Option 2" Value="2"></asp:ListItem> </asp:CheckBoxList>

RadioButtonList: Displays a group of radio buttons.

<asp:RadioButtonList ID="rblOptions" runat="server"> <asp:ListItem Text="Option 1" Value="1"></asp:ListItem> <asp:ListItem Text="Option 2" Value="2"></asp:ListItem> </asp:RadioButtonList>

CheckBox: Allows the user to select a single option.

<asp:CheckBox ID="chkAgree" Text="I Agree" runat="server" />

RadioButton: Allows the user to select one option in a group.

<asp:RadioButton ID="rbOption1" Text="Option 1" GroupName="Options" runat="server" />

Panel: A container for grouping other controls with optional visibility settings.

<asp:Panel ID="pnlContainer" runat="server" Visible="true"> <asp:Label ID="lblInsidePanel" runat="server" Text="This is inside a panel"></asp:Label> </asp:Panel>

6. Postback, Cross Page Posting, and AutoPostBack in ASP.NET:

Feature	Postback	Cross Page Posting	AutoPostBack
Definition	Submits the page back to itself (same page).	Submits form data to another page.	Triggers a postback automatically when an event occurs.
Trigger	Clicking a button or any control that submits the form.	Clicking a button with PostBackUrl set.	Selection change in dropdowns, checkboxes, radio buttons, etc.

Usage	When the form needs to refresh with new data from the server.	When form data needs to be processed on a different page.	Reduces user action by submitting form elements automatically.
Control Example	<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />	<asp:Button ID="btnCrossPage" runat="server" Text="Submit" PostBackUrl="NextPage.aspx" />	<asp:DropDownList ID="ddlOptions" runat="server" AutoPostBack="true">
Data Retrieval	Uses Request.Form to get values from controls.	Uses PreviousPage property in the target page.	No explicit retrieval required; event fires automatically.
Performance	Causes a full page reload.	May cause a full page reload to another page.	Reduces manual interaction but can cause frequent postbacks.
Common Use Cases	Forms that update based on user input.	Multi-page forms, wizardstyle interfaces.	Dynamic dropdowns, validation, or filtering.

Exercise:

1. **Design asp.net web form to demonstrate various events in asp.net page life cycle.**

Program:

WFPageLifeCycle.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WFPageLifeCycle.aspx.cs" Inherits="Practical_No1.WFPageLifeCycle" %>
<!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="lblName" runat="server" ></asp:Label>
            <br/>
```

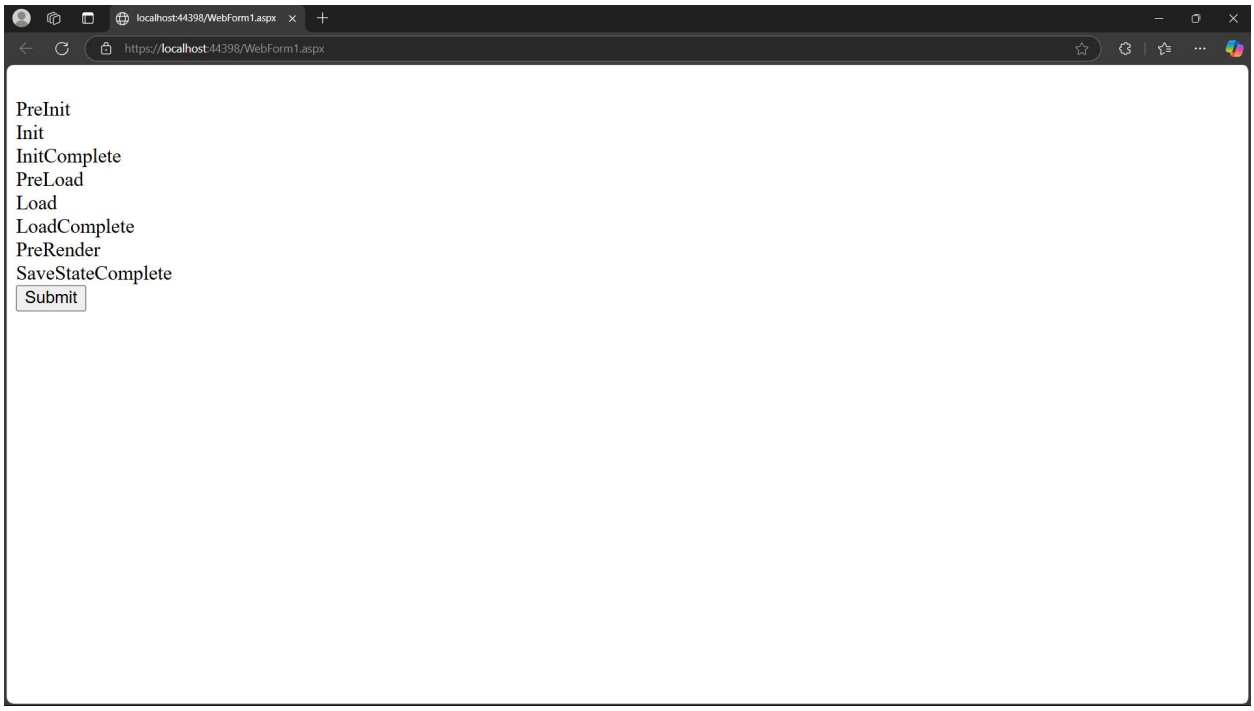
```
            <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click"  /><br />
            <asp:Label ID="lblButtonClick" runat="server" ></asp:Label>
        </div>
    </form>
</body>
</html>
```

WFPageLifeCycle.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical_No1
{
    public partial class WFPageLifeCycle : System.Web.UI.Page
    {
        protected void Page_PreInit(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "PreInit";
        }

        protected void Page_Init(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "Init";
        }
        protected void Page_InitComplete(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "InitComplete";
        }
        protected override void OnPreLoad(EventArgs e)
        {

            lblName.Text = lblName.Text + "<br/>" + "PreLoad";
        }
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
            lblName.Text = lblName.Text + "<br/>" + "Load";
        }

        protected void Page_LoadComplete(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "LoadComplete";
        }
        protected override void OnPreRender(EventArgs e)
        {
            System.Threading.Thread.Sleep(3000);
            lblName.Text = lblName.Text + "<br/>" + "PreRender";
        }
        protected override void OnSaveStateComplete(EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "SaveStateComplete";
        }
        protected void Page_UnLoad(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "UnLoad";
        }
        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            lblName.Text = lblName.Text + "<br/>" + "btnSubmit_Click";
        }
    }
}
```

Out Put:



2. Design asp.net web forms to demonstrate postback posting, cross-page posting and auto postback.

Program:

- **postback posting**

WFPostback.aspx

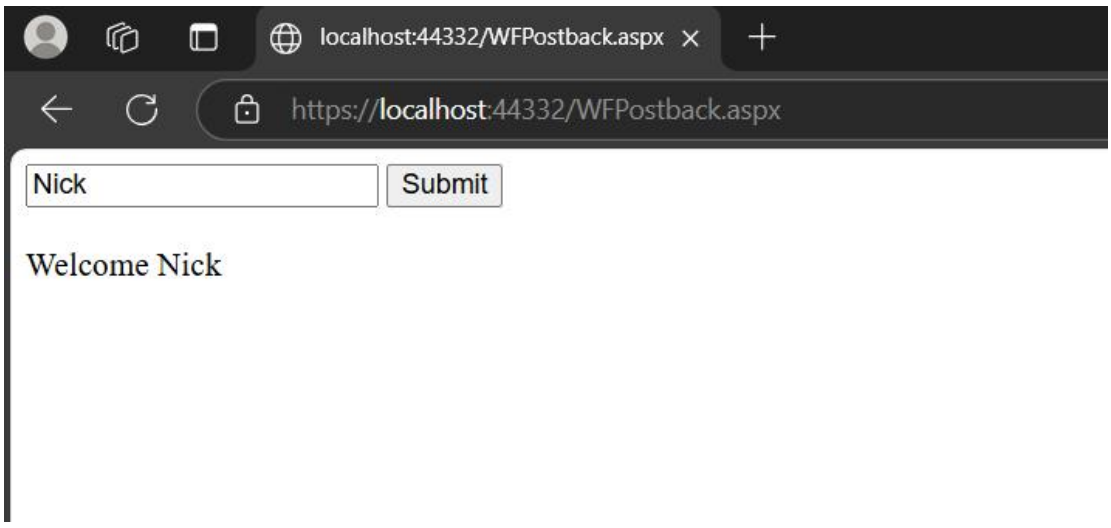
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WFPostback.aspx.cs"
Inherits="Practical_No1.WFPostback" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <asp:Button ID="btnSubmit" Text="Submit" runat="server"
OnClick="btnSubmit_Click" /><br /> <br />
            <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
        </div>
```

```
    </form>
</body>
</html>
```

WFPostback.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Practical_No1
{
    public partial class WFPostback : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            //txtName.Text = "Ravi";
            if (IsPostBack)
            {
                lblMessage.Text = "Welcome " + txtName.Text;
            }
        }
        protected void btnSubmit_Click(object sender, EventArgs e)
        {}
    }
}
```

OutPut:



- **cross-page posting**

WFCrosspage.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WFCrosspage.aspx.cs"
Inherits="Practical_No1.WFCrosspage" %>
<!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="Enter your Name : "></asp:Label>
         <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
         <asp:Button ID="btnSubmit" Text="Submit" runat="server"
PostBackUrl="~/WFCrosspage2.aspx" />
      </div>
   </form>
</body>
</html>
```

WFCrosspage2.aspx

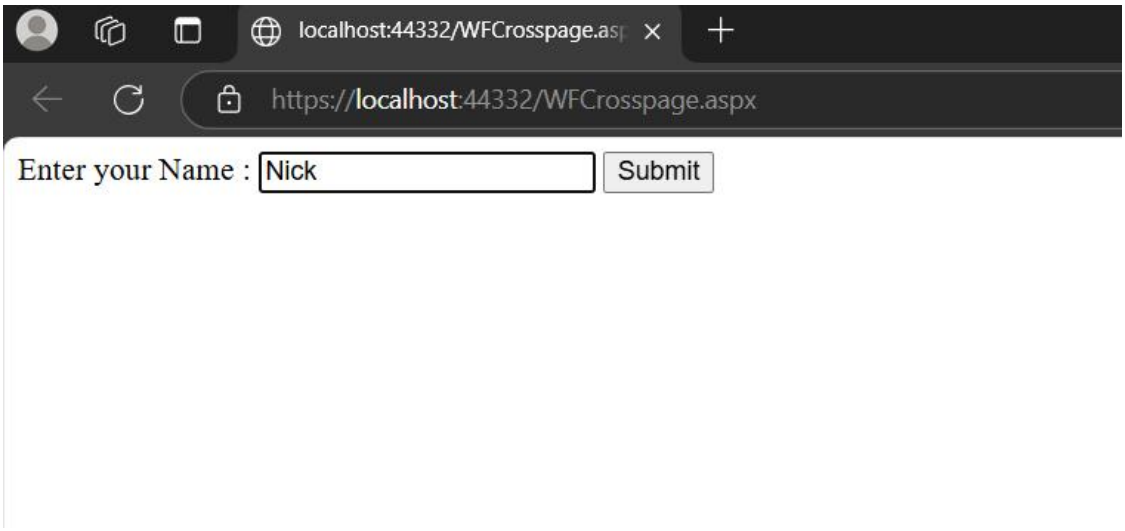
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WFCrosspage2.aspx.cs"
Inherits="Practical_No1.WFCrosspage2" %>

<!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="lblResult" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

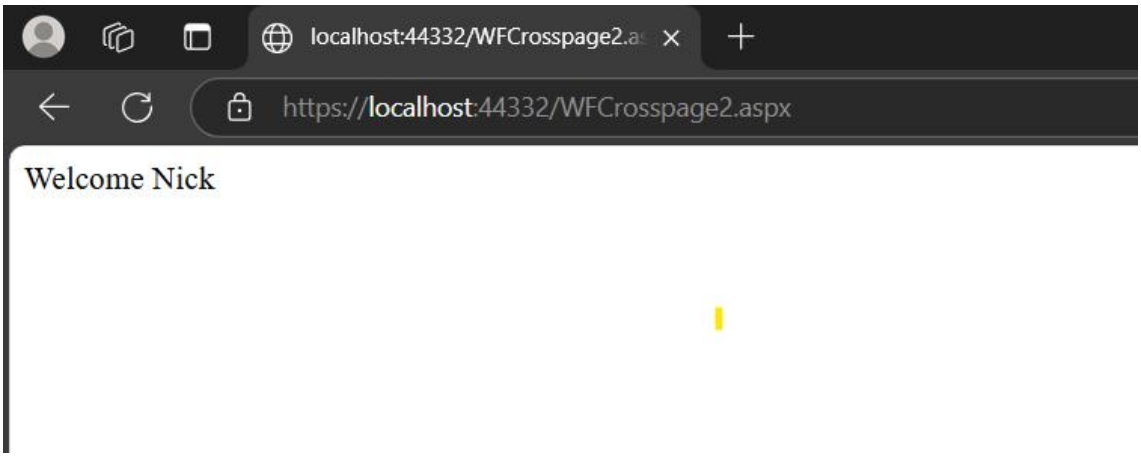
WFCrosspage2.aspx.cs

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Practical_No1
{
   public partial class WFCrosspage2 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (PreviousPage != null && PreviousPage.IsCrossPagePostBack)
         {
            TextBox txtName2 = (TextBox)PreviousPage.FindControl("txtName");
            if (txtName2 != null)
            {
               lblResult.Text = "Welcome " + txtName2.Text;
            }
         }
      }
   }
}
```

Out Put:





3. Design an asp.net web form to calculate employee salary. Accept Employee id, name, designation, department and basic salary. Consider dearness allowance 80% of basic salary and travelling allowance 10% of basic salary. Total Salary = (Basic + Dearness allowance + Travelling allowance) – (Income Tax + Professional Tax)

SalaryCalculator.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="SalaryCalculator.aspx.cs" Inherits="Practical_No1.SalaryCalculator" %>

<!DOCTYPE html>
<html>
<head>
   <title>Employee Salary Calculator</title>
 <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
   <div class="container">
      <h2>Employee Salary Calculator</h2>

      <form id="form1" runat="server">
         <label>Employee ID:</label>
         <asp:TextBox ID="txtEmpID" runat="server"></asp:TextBox>
         <br />
         <label>Name:</label>
         <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
         <br />
         <label>Designation:</label>
         <asp:TextBox ID="txtDesignation" runat="server"></asp:TextBox>
```

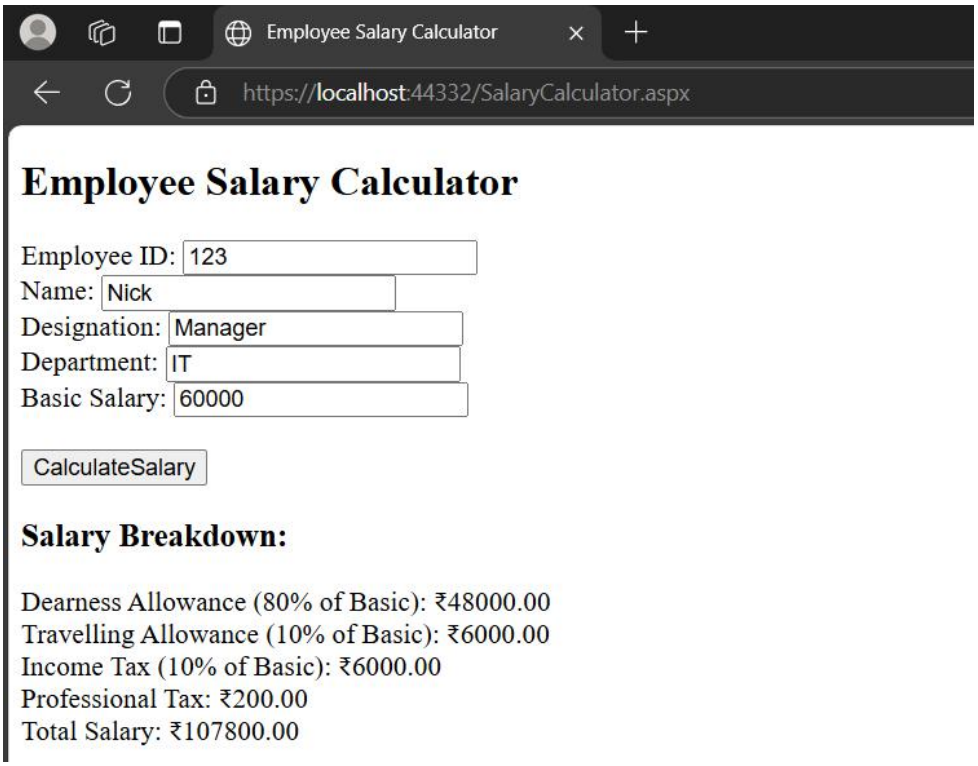
```
            <br />
            <label>Department:</label>
            <asp:TextBox ID="txtDepartment" runat="server"></asp:TextBox>
            <br />
            <label>Basic Salary:</label>
            <asp:TextBox ID="txtBasicSalary" runat="server"></asp:TextBox>
            <br /><br />
            <asp:Button ID="btnCalculate" runat="server" Text="CalculateSalary"
OnClick="btnCalculate_Click" />
            <br />
            <div class="result">
               <h3>Salary Breakdown:</h3>
               <asp:Label ID="lblDA" runat="server" CssClass="text-blue"></asp:Label><br />
               <asp:Label ID="lblTA" runat="server" CssClass="text-blue"></asp:Label><br />
               <asp:Label ID="lblIncomeTax" runat="server" CssClass="textred"></asp:Label><br
/>
               <asp:Label ID="lblProfessionalTax" runat="server"
CssClass="textred"></asp:Label><br />
              <asp:Label ID="lblTotalSalary" runat="server" CssClass="textgreen"></asp:Label><br
/>
            </div>
         </form>
      </div>
</body>
</html>
```

SalaryCalculator.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical_No1
{
   public partial class SalaryCalculator : System.Web.UI.Page
   {
      protected void btnCalculate_Click(object sender, EventArgs e)
      {
         double basicSalary;
         if (!double.TryParse(txtBasicSalary.Text, out basicSalary) || basicSalary < 0)
         {
            lblTotalSalary.Text = "Please enter a valid positive Basic Salary.";
```

```
            return;
        }
        double dearnessAllowance = 0.80 * basicSalary;
        double travellingAllowance = 0.10 * basicSalary;

        double incomeTax = 0.10 * basicSalary;
        double professionalTax = 200;

        double totalSalary = (basicSalary + dearnessAllowance + travellingAllowance) -
(incomeTax + professionalTax);

        lblDA.Text = $"Dearness Allowance (80% of Basic): ₹{dearnessAllowance:F2}";
        lblTA.Text = $"Travelling Allowance (10% of Basic): ₹{travellingAllowance:F2}";
        lblIncomeTax.Text = $"Income Tax (10% of Basic): ₹{incomeTax:F2}";
        lblProfessionalTax.Text = $"Professional Tax: ₹{professionalTax:F2}";
        lblTotalSalary.Text = $"Total Salary: ₹{totalSalary:F2}";
    }
  }
}
```

Output:



4. Design asp.net web form to calculate monthly electricity bill and show on another web form. Accept Consumer Number, Consumer Name, Address,

Previous meter reading, current meter reading, month and year of bill, total units consumed for month. Consider the following table for electricity bill calculation:

Sr. No	Condition	Per unit charge
1	total units consumed for month < = 100	2 Rs.
2	total units consumed for month >100 and <=200	4 Rs.
3	total units consumed for month >200	5 Rs.

ElectricityBillCalculator.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="ElectricityBillCalculator.aspx.cs"
Inherits="Practical_No1.ElectricityBillCalculator" %>

<!DOCTYPE html>
<html>
<head>
   <title>Electricity Bill Calculator</title>
      <link rel="stylesheet" type="text/css" href="style1.css" />
</head>
<body>
   <div class="container">
      <h2>Electricity Bill Calculator</h2>
      <form id="form1" runat="server">
         <label>Consumer Number:</label>
         <asp:TextBox ID="txtConsumerNumber" runat="server"></asp:TextBox>
         <br />
         <label>Consumer Name:</label>
         <asp:TextBox ID="txtConsumerName" runat="server"></asp:TextBox>
         <br />
         <label>Address:</label>
         <asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>
         <br />
         <label>Previous Meter Reading:</label>
         <asp:TextBox ID="txtPreviousReading" runat="server"></asp:TextBox>
         <br />
         <label>Current Meter Reading:</label>
         <asp:TextBox ID="txtCurrentReading" runat="server"></asp:TextBox>
         <br />
         <label>Month:</label>
         <asp:TextBox ID="txtMonth" runat="server"></asp:TextBox>
         <br />
         <label>Year:</label>
```

```
            <asp:TextBox ID="txtYear" runat="server"></asp:TextBox>
            <br /><br />
            <asp:Button ID="btnCalculate" runat="server" Text="Calculate Bill"
OnClick="btnCalculate_Click" CssClass="aspNetButton" />
        </form>
    </div>
</body>
</html>
```

ElectricityBillCalculator.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical_No1
{
    public partial class ElectricityBillCalculator : System.Web.UI.Page
    {
        protected void btnCalculate_Click(object sender, EventArgs e)
        {
            string consumerNumber = txtConsumerNumber.Text;
            string consumerName = txtConsumerName.Text;
            string address = txtAddress.Text;
            string month = txtMonth.Text;
            string year = txtYear.Text;
            int prevReading, currReading;
            if (!int.TryParse(txtPreviousReading.Text, out prevReading)
|| !int.TryParse(txtCurrentReading.Text, out currReading) || currReading < prevReading)
            {
                Response.Write("<script>alert('Invalid meter readings!');</script>");
                return;
            }
            int totalUnits = currReading - prevReading;
            double billAmount = 0;
            if (totalUnits <= 100)
                billAmount = totalUnits * 2;
            else if (totalUnits <= 200)
                billAmount = (100 * 2) + ((totalUnits - 100) * 4);
            else
                billAmount = (100 * 2) + (100 * 4) + ((totalUnits - 200) * 5);
            string url =
$"BillSummary.aspx?consumerNumber={consumerNumber}&consumerName={consumerNam
```

```
e}&address={address}&month={month}&year={year}&totalUnits={totalUnits}&billAmount=
{billAmount}";
            Response.Redirect(url);
        }
    }
}
```

BillSummary.aspx

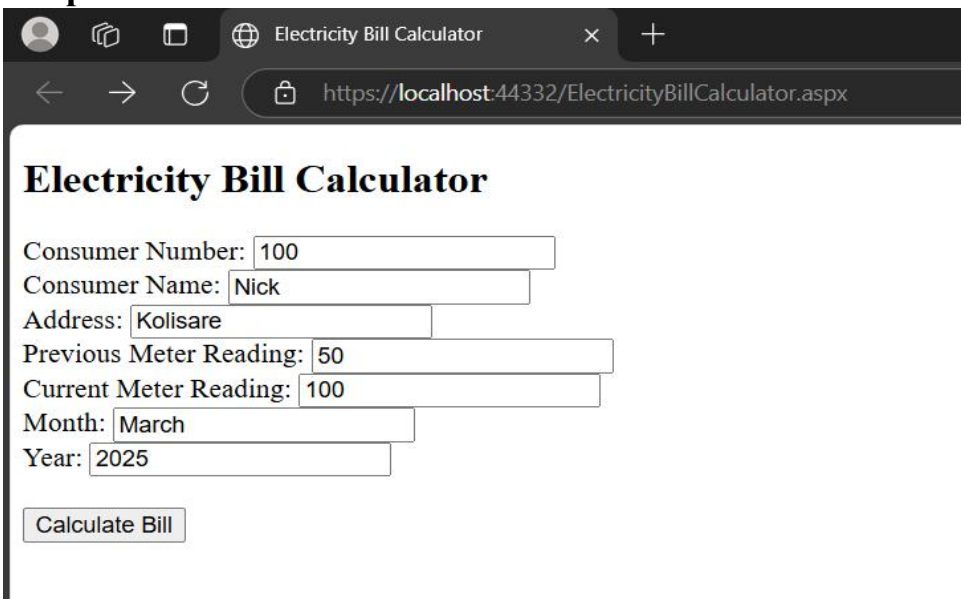
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="BillSummary.aspx.cs"
Inherits="Practical_No1.BillSummary" %>

<!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="lblSummary" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

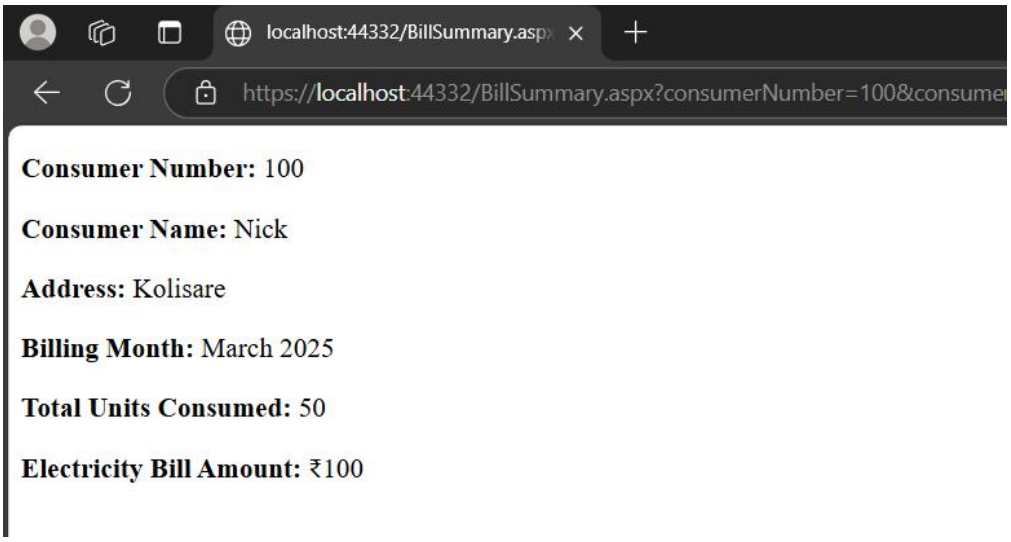
BillSummary.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Practical_No1
{
    public partial class BillSummary : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {

                string consumerNumber = Request.QueryString["consumerNumber"];
```

```
            string consumerName = Request.QueryString["consumerName"];
            string address = Request.QueryString["address"];
            string month = Request.QueryString["month"];
            string year = Request.QueryString["year"];
            string totalUnits = Request.QueryString["totalUnits"];
            string billAmount = Request.QueryString["billAmount"];

            lblSummary.Text = $@"
            <div class='result'>
                <p><strong>Consumer Number:</strong> {consumerNumber}</p>
                <p><strong>Consumer Name:</strong> {consumerName}</p>
                <p><strong>Address:</strong> {address}</p>
                <p><strong>Billing Month:</strong> {month} {year}</p>
                <p><strong>Total Units Consumed:</strong> {totalUnits}</p>
                <p><strong>Electricity Bill Amount:</strong> ₹{billAmount}</p>
            </div>";
        }
    }
  }
}
```

Output:





5. Design asp.net web form to accept friend information like name, city, address, date of birth (Use TextBox only), gender, hobbies and show the same details on another webform. Use appropriate controls.

FriendInfo.aspx

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

<!DOCTYPE html>
<html>
<head>
   <title>Friend Information Form</title>
</head>
<body>
   <h2>Friend Information Form</h2>
   <form id="form1" runat="server">
      <table>
         <tr>
            <td>Name:</td>
            <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
         </tr>
         <tr>
            <td>City:</td>
            <td><asp:TextBox ID="txtCity" runat="server"></asp:TextBox></td>
         </tr>
         <tr>
            <td>Address:</td>
            <td><asp:TextBox ID="txtAddress" runat="server"></asp:TextBox></td>
         </tr>
         <tr>
```

```
            <td>Date of Birth:</td>
            <td><asp:TextBox ID="txtDOB" runat="server"
TextMode="Date"></asp:TextBox></td>
        </tr>
        <tr>
            <td>Gender:</td>
            <td>
                <asp:RadioButtonList ID="rblGender" runat="server">
                    <asp:ListItem Text="Male" Value="Male"></asp:ListItem>
                    <asp:ListItem Text="Female" Value="Female"></asp:ListItem>
                    <asp:ListItem Text="Other" Value="Other"></asp:ListItem>
                </asp:RadioButtonList>
            </td>
        </tr>
        <tr>
            <td>Hobbies:</td>
            <td>
                <asp:CheckBoxList ID="chkHobbies" runat="server">
                    <asp:ListItem Text="Reading" Value="Reading"></asp:ListItem>
                    <asp:ListItem Text="Music" Value="Music"></asp:ListItem>
                    <asp:ListItem Text="Sports" Value="Sports"></asp:ListItem>
                    <asp:ListItem Text="Gaming" Value="Gaming"></asp:ListItem>
                </asp:CheckBoxList>
            </td>
        </tr>
        <tr>
            <td colspan="2">
<asp:ButtonID="btnSubmit"runat="server"Text="Submit"OnClick="btnSubmit_Click" />
            </td>
        </tr>
    </table>
  </form>
</body>
</html>
```

FriendInfo.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebFormsDemo
{
```

```
public partial class FriendInfo : System.Web.UI.Page
{
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        string name = txtName.Text;
        string city = txtCity.Text;
        string address = txtAddress.Text;
        string dob = txtDOB.Text;
        string gender = rblGender.SelectedValue;
        string hobbies = "";
        foreach (var item in chkHobbies.Items)
        {
            if (((System.Web.UI.WebControls.ListItem)item).Selected)
            {
                hobbies += ((System.Web.UI.WebControls.ListItem)item).Value + ", ";
            }
        }
        hobbies = hobbies.TrimEnd(',', ' ');
        string url =
$"FriendDetails.aspx?name={HttpUtility.UrlEncode(name)}&city={HttpUtility.UrlEncode(city)}
&address={HttpUtility.UrlEncode(address)}&dob={HttpUtility.UrlEncode(dob)}&gender={Htt
pUtility.UrlEncode(gender)}&hobbies={HttpUtility.UrlEncode(hobbies)}";
        Response.Redirect(url);
    }
}
}
```

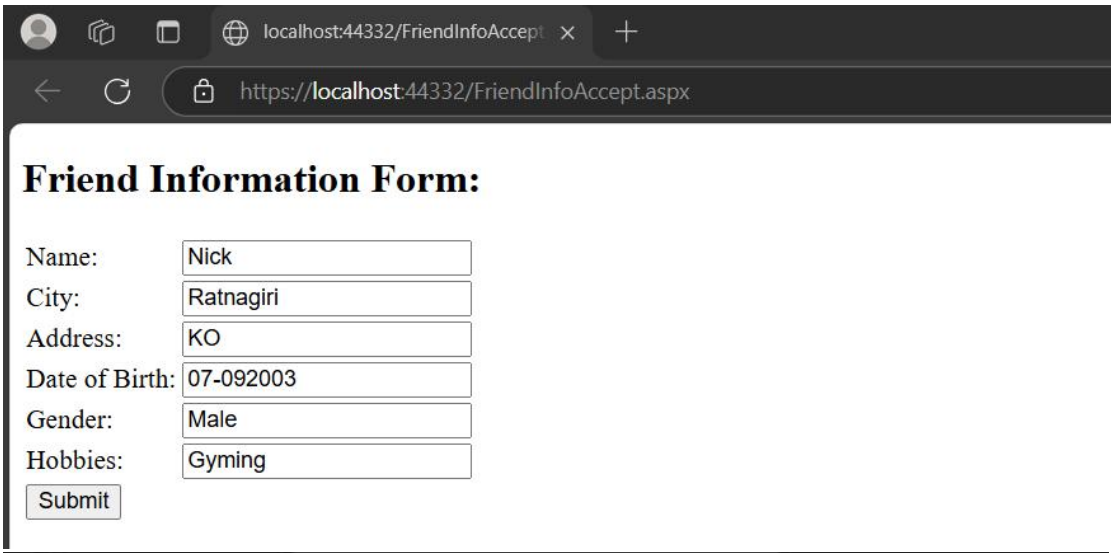
FriendDetails.aspx

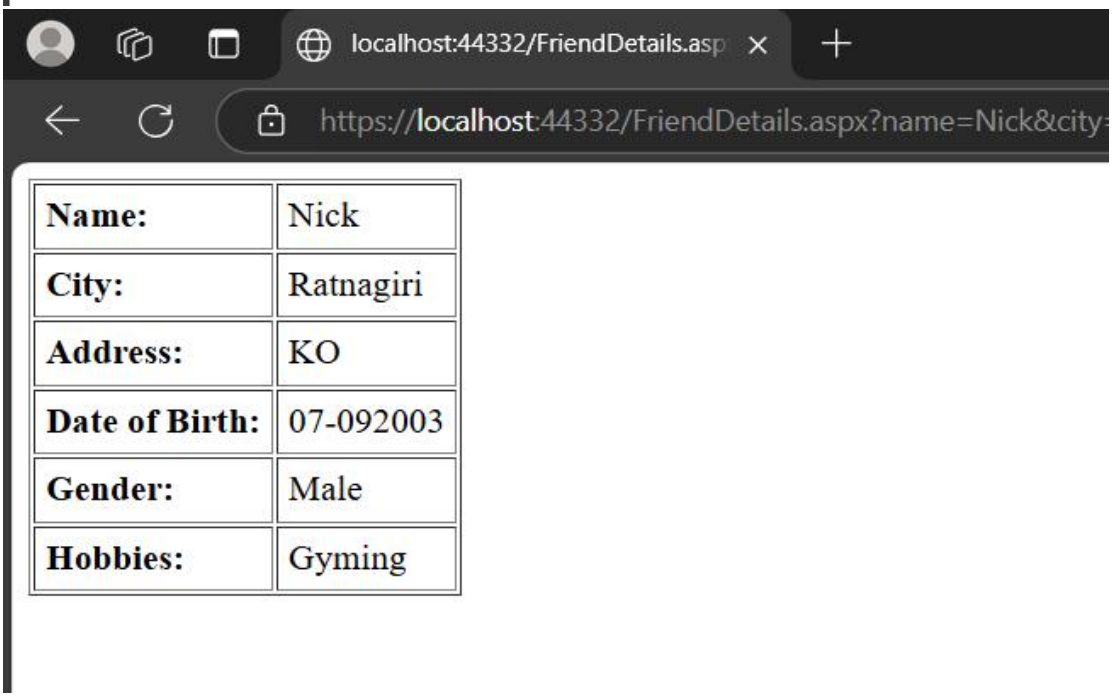
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="FriendDetails.aspx.cs"
Inherits="WebFormsDemo.FriendDetails" %>

<!DOCTYPE html>
<html>
<head>
    <title>Friend Details</title>
</head>
<body>
    <h2>Friend Details</h2>
    <form id="form1" runat="server">
        <asp:Label ID="lblDetails" runat="server"></asp:Label>
    </form>
</body>
</html>
```

FriendDetails.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebFormsDemo
{
   public partial class FriendDetails : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            string name = Request.QueryString["name"];
            string city = Request.QueryString["city"];
            string address = Request.QueryString["address"];
            string dob = Request.QueryString["dob"];
            string gender = Request.QueryString["gender"];
            string hobbies = Request.QueryString["hobbies"];

            lblDetails.Text = $@"
            <table border='1' cellpadding='5'>
               <tr><td><strong>Name:</strong></td><td>{name}</td></tr>
               <tr><td><strong>City:</strong></td><td>{city}</td></tr>
               <tr><td><strong>Address:</strong></td><td>{address}</td></tr>
               <tr><td><strong>Date of Birth:</strong></td><td>{dob}</td></tr>
               <tr><td><strong>Gender:</strong></td><td>{gender}</td></tr>
               <tr><td><strong>Hobbies:</strong></td><td>{hobbies}</td></tr>
            </table>";
         }
      }
   }
}
```

Output:


localhost:44332/FriendInfoAccept
https://localhost:44332/FriendInfoAccept.aspx
Friend Information Form:
Name: Nick
City: Ratnagiri
Address: KO
Date of Birth: 07-092003
Gender: Male
Hobbies: Gyming
Submit


localhost:44332/FriendDetails.asp
https://localhost:44332/FriendDetails.aspx?name=Nick&city
Name: Nick
City: Ratnagiri
Address: KO
Date of Birth: 07-092003
Gender: Male
Hobbies: Gyming