

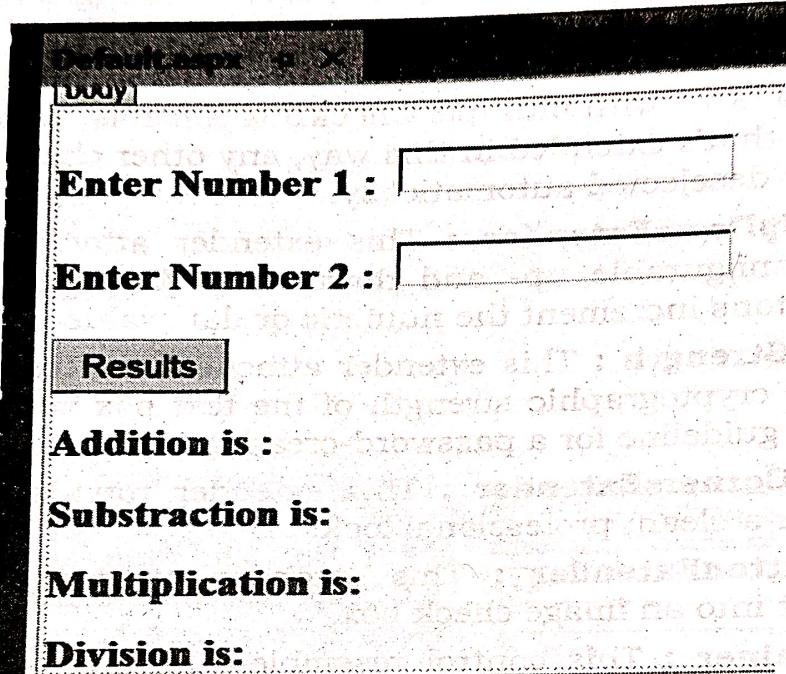
PRACTICALS

PRACTICAL - 1

Practical - 1 : Working with basic C# and ASP .NET

Practical - 1(a) : Create an application to print on screen the output of adding, subtracting, multiplying and dividing two numbers entered by the user in C#.

Default.aspx



Default.aspx.cs

```
public partial class _Default : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        int addition, subtraction, multiplication, division;
        addition = Convert.ToInt32(TextBox1.Text) + Convert.ToInt32(TextBox2.Text);
        subtraction = Convert.ToInt32(TextBox1.Text) - Convert.ToInt32(TextBox2.Text);
        multiplication = Convert.ToInt32(TextBox1.Text) * Convert.ToInt32(TextBox2.Text);
        division = Convert.ToInt32(TextBox1.Text) / Convert.ToInt32(TextBox2.Text);
        Label1.Text = "Addition of the number is: " + addition;
        Label2.Text = "Subtraction of the number is: " + subtraction;
        Label3.Text = "Multiplication of the number is: " + multiplication;
        Label4.Text = "Division of the number is: " + division;
    }
}
```

OUTPUT :

Enter Number 1 :

Enter Number 2 :
Results

Addition of the number is: 60

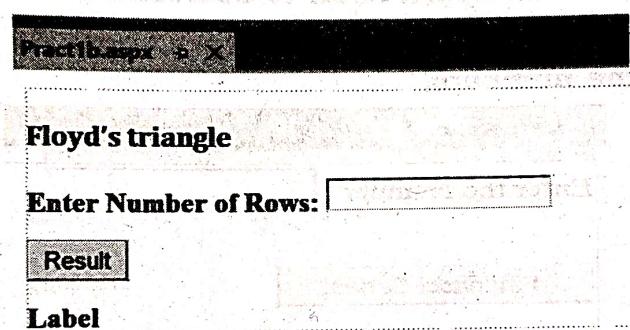
Subtraction of the number is: 30

Multiplication of the number is: 675

Division of the number is: 3

Practical - 1 (b) : Create an application to print Floyd's triangle till n rows in C#.

stropr.aspx



pract1b.aspx.cs

```
public partial class pract1b : System.Web.UI.Page
```

```
{
```

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

```
    int numOfRows = Convert.ToInt32(TextBox1.Text);
```

```
    int number = 1;
```

```
    StringBuilder sb = new StringBuilder();
```

```
    for (int i = 1; i <= numOfRows; i++)
```

```
{
```

```
        for (int j = 1; j <= i; j++)
```

```
{
```

```
            sb.Append(number + " ");
```

```
            number++;
```

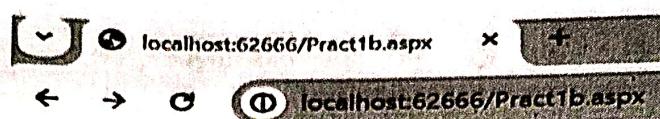
```
}
```

```
        sb.Append("<br />");
```

```
}
```

```
    Label1.Text = sb.ToString();
```

```
}
```

OUTPUT :**Floyd's triangle**Enter Number of Rows: **Result**

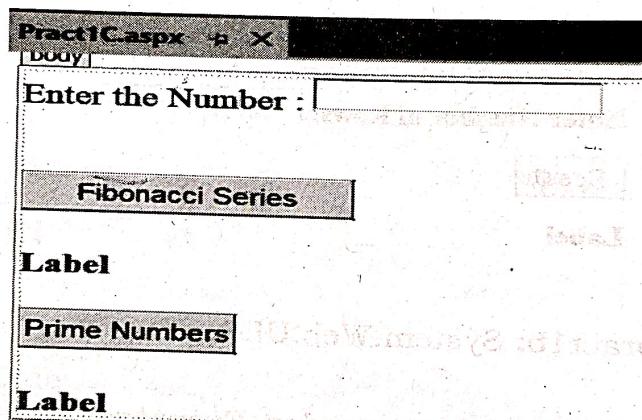
```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55

```

Practical - 1(c) : Create an application to demonstrate following operations

- i) Generate Fibonacci series.
- ii) Test for prime numbers.

**Pract1C.aspx.cs**

public partial class Pract1C : System.Web.UI.Page

{

```
protected void Button1_Click(object sender, EventArgs e)
{
```

```
    int a, b, c, i, n;
```

```
    a = 0;
```

```
    b = 1;
```

```
    Label1.Text = a.ToString() + b.ToString();
```

```
    n = Convert.ToInt32(textBox1.Text);
```

```
    for (i = 1; i <= n; ++i)
```

```
{
```

```
    c = a + b;
```

```
    Label1.Text = Label1.Text + c.ToString();
```

```
    a = b;
```

```
    b = c;
```

गुरुवार

```
}
```

```
protected void Button2_Click(object sender, EventArgs e)
```

```
{
```

```
    int n, i, s = 0;
```

```
    n = Convert.ToInt32(textBox1.Text);
```

```
    if (n == 0 || n == 1)
```

```
        s = 1;
```

```
    for (i = 2; i <= n / 2; ++i)
```

```
    {
```

```
        if (n % i == 0)
```

```
        {
```

```
            s = 1;
```

```
            break;
```

```
        }
```

```
    }
```

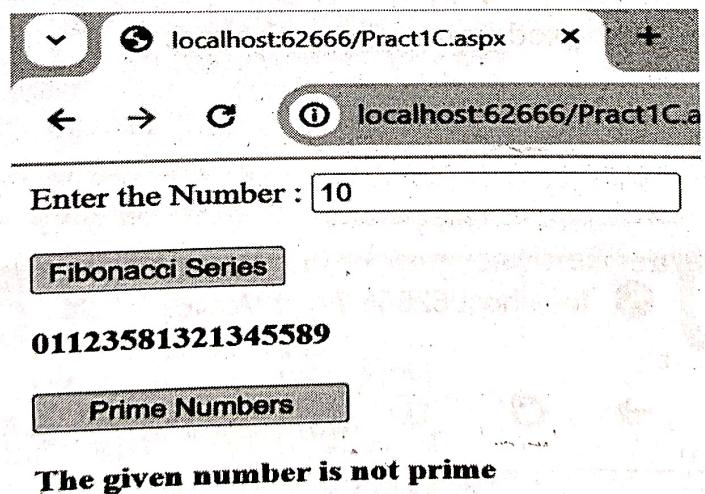
```
    if (s == 0)
```

```
        Label2.Text = "The given number is prime";
```

```
    else
```

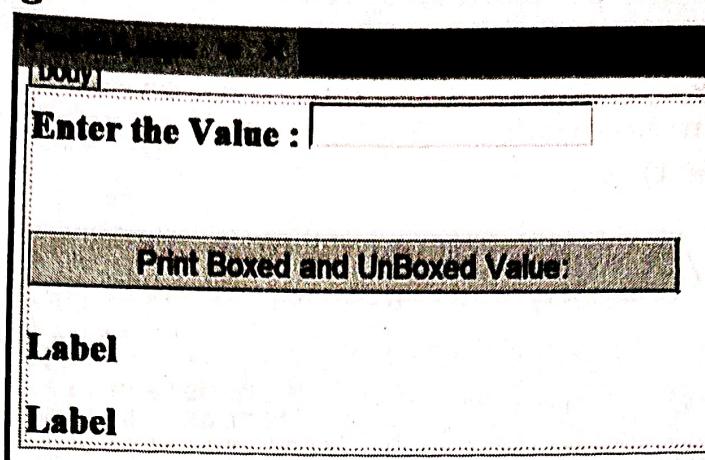
```
        Label2.Text = "The given number is not prime";
```

```
}
```

OUTPUT :

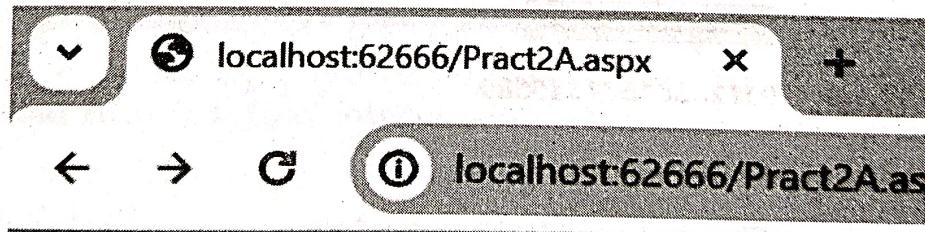
PRACTICAL - 2

Practical - 2(a) : Create a simple application to demonstrate the concepts of boxing and unboxing.

**Pract2a.aspx**

```
public partial class Pract2A : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        // Boxing: Converting a value type to an object type
        int valueType = Convert.ToInt32(textBox1.Text);
        object boxed = valueType; // The value type 'valueType' is boxed
        Label1.Text = "Boxed value: " + boxed;

        // Unboxing: Converting an object type back to a value type
        int unboxed = (int)boxed; // The object 'boxed' is unboxed
        Label2.Text = "Unboxed value: " + unboxed;
    }
}
```

OUTPUT :

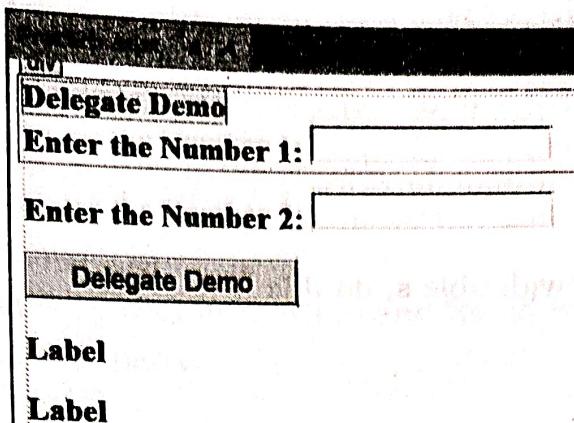
Enter the Value :

Print Boxed and UnBoxed Value:

Label

Label

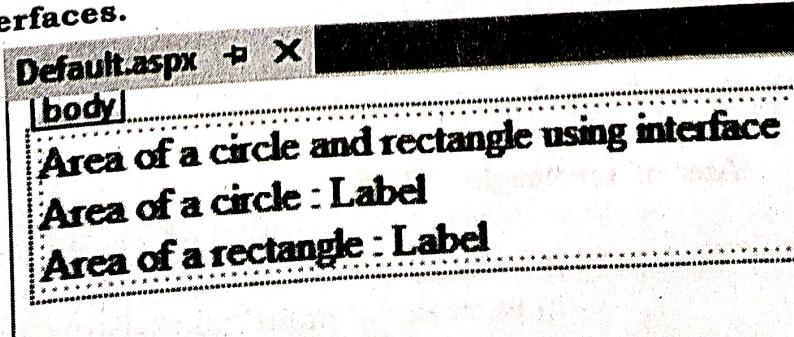
Practical - 2(b) : Create a simple application to perform addition and subtraction using delegate.



Pract2b.aspx.cs

```
public delegate int MathOperation(int a, int b);
public partial class Pract2b : System.Web.UI.Page
{
    int Add(int a, int b)
    {
        return a + b;
    }
    int Subtract(int a, int b)
    {
        return a - b;
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        int x = Convert.ToInt32(textBox1.Text);
        int y = Convert.ToInt32(textBox2.Text);
        MathOperation add = new MathOperation(Add);
        MathOperation subtract = new MathOperation(Subtract);
        Label1.Text = add(x,y).ToString();
        Label2.Text = subtract(x,y).ToString();
    }
}
```

Practical - 2(c) : Create a simple application to demonstrate use of the concepts of interfaces.



Default.aspx.cs**interface Area**

```

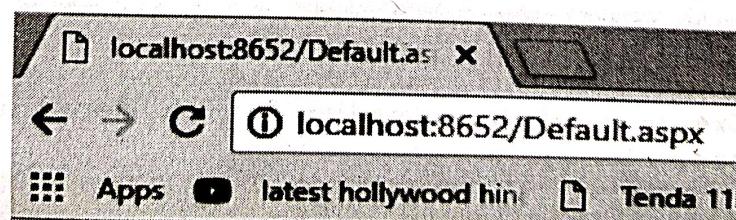
{
    double show(double s, double t);
}

class Rect : Area
{
    public double show(double s, double t)
    {
        return s * t;
    }
}

class Circle : Area
{
    public double show(double s, double t)
    {
        return (3.14 * s * s);
    }
}

public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Rect r1 = new Rect();
        double x = r1.show(3, 4);
        Circle c1 = new Circle();
        double y = c1.show(3, 4);
        Label1.Text = x.ToString();
        Label2.Text = y.ToString();
    }
}

```

OUTPUT :

Area of a circle and rectangle using interface

Area of a circle : 12

Area of a rectangle : 28.26

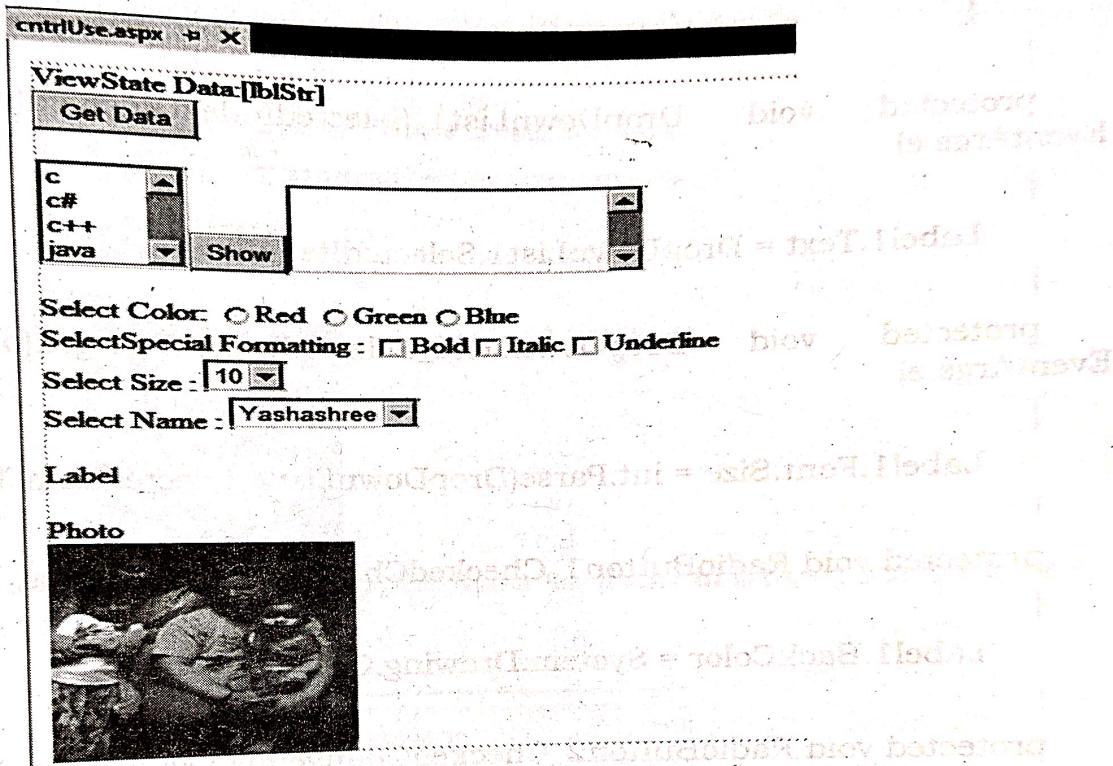
PRACTICAL - 3

9°9°9°

Practical - 3(a) : Create a simple web page with various server controls to demonstrate setting and use of their properties. (Example: AutoPostBack)

1. On click of a button control display the selected items from the list in a textbox. Also in the same webpage display the name of the selected item from the DropDownList1 in a label. Also change the font size of the same label according to the font size selected from the dropdownlist2.
2. Image control for photo.
3. Check Boxes provides special formatting and Radio Buttons provides colour.
4. Use of AutoPostBack property.

cntrlUse.aspx



cntrlUse.aspx.cs

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            string str = "Vedshree Sambare";
            if (ViewState["name"] == null)
            {
                ViewState["name"] = str;
            }
        }
    }

    protected void btn_Click(object sender, EventArgs e)
    {
        lblStr.Text = ViewState["name"].ToString();
    }
}
```

```

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        TextBox1.Text = "";
        for (int i = 0; i < ListBox1.Items.Count; i++)
        {
            if (ListBox1.Items[i].Selected == true)
            {
                TextBox1.Text = TextBox1.Text + "" + ListBox1.Items[i].Text + "\n";
            }
        }
    }

    protected void DropDownList1_SelectedIndexChanged(object sender,
                                                EventArgs e)
    {
        Label1.Text = DropDownList1.SelectedItem.Text;
    }

    protected void DropDownList2_SelectedIndexChanged(object sender,
                                                EventArgs e)
    {
        Label1.Font.Size = int.Parse(DropDownList2.SelectedItem.Text);
    }

    protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
    {
        Label1.BackColor = System.Drawing.Color.Red;
    }

    protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
    {
        Label1.BackColor = System.Drawing.Color.Green;
    }

    protected void RadioButton3_CheckedChanged(object sender, EventArgs e)
    {
        Label1.BackColor = System.Drawing.Color.Blue;
    }

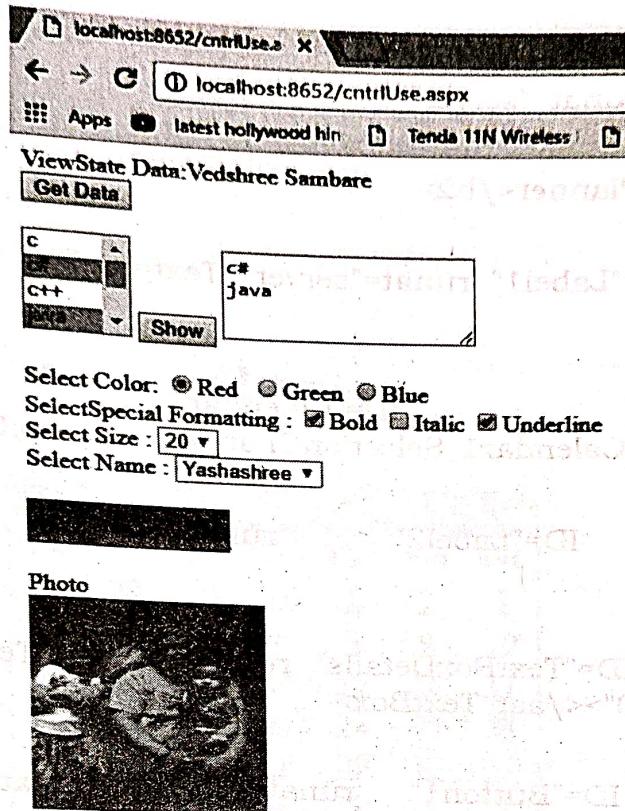
    protected void CheckBox1_CheckedChanged(object sender, EventArgs e)
    {
        Label1.Font.Bold = true;
    }

    protected void CheckBox2_CheckedChanged(object sender, EventArgs e)
    {
        Label1.Font.Italic = true;
    }

    protected void CheckBox3_CheckedChanged(object sender, EventArgs e)
    {
}

```

```
Label1.Font.Underline = true;
}
```

OUTPUT :

Practical – 3(b) : Create a simple application to demonstrate your vacation using calendar control.

calndrCtrl.aspx

Vacation Planner

Select your vacation start date:

July 2024						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Vacation Details:

Save Vacation

[LabelResult]

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>
<!DOCTYPE html>
```

```

Q'Q'Q'
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Vacation Planner</h2>

            <asp:Label ID="Label1" runat="server" Text="Select your vacation start date:"></asp:Label>
            <br />
            <asp:Calendar ID="Calendar1" runat="server" OnSelectionChanged="Calendar1_SelectionChanged" ></asp:Calendar>
            <br /><br />
            <asp:Label ID="Label2" runat="server" Text="Vacation Details:"></asp:Label>
            <br />
            <asp:TextBox ID="TextBoxDetails" runat="server" TextMode="MultiLine" Rows="5" Columns="30"></asp:TextBox>
            <br /><br />
            <asp:Button ID="Button1" runat="server" Text="Save Vacation" OnClick="Button1_Click" />
            <br /><br />
            <asp:Label ID="LabelResult" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>

```

calndrCtrl.aspx.cs

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        LabelResult.Text = "You selected: " + Calendar1.SelectedDate.ToString();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        string vacationDate = Calendar1.SelectedDate.ToString();
        string vacationDetails = TextBoxDetails.Text;
    }
}

```

```

practicals
if (string.IsNullOrEmpty(vacationDetails))
{
    LabelResult.Text = "Please enter vacation details.";
}
else
{
    LabelResult.Text = $"Vacation on {vacationDate}: {vacationDetails}";
}
}
}

```

OUTPUT :

Vacation Planner

Select your vacation start date:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>
<u>25</u>	<u>26</u>	27	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>
<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>

Vacation Details:

Yashashree's Birthday

Save Vacation

Vacation on 27-08-2024: Yashashree's Birthday

Practical - 3(c) : Demonstrate the use of Treeview operations on the web form.

Add XML File

Website -> Add -> XML File and Name it 'stdetail'.

stdetail.xml

```

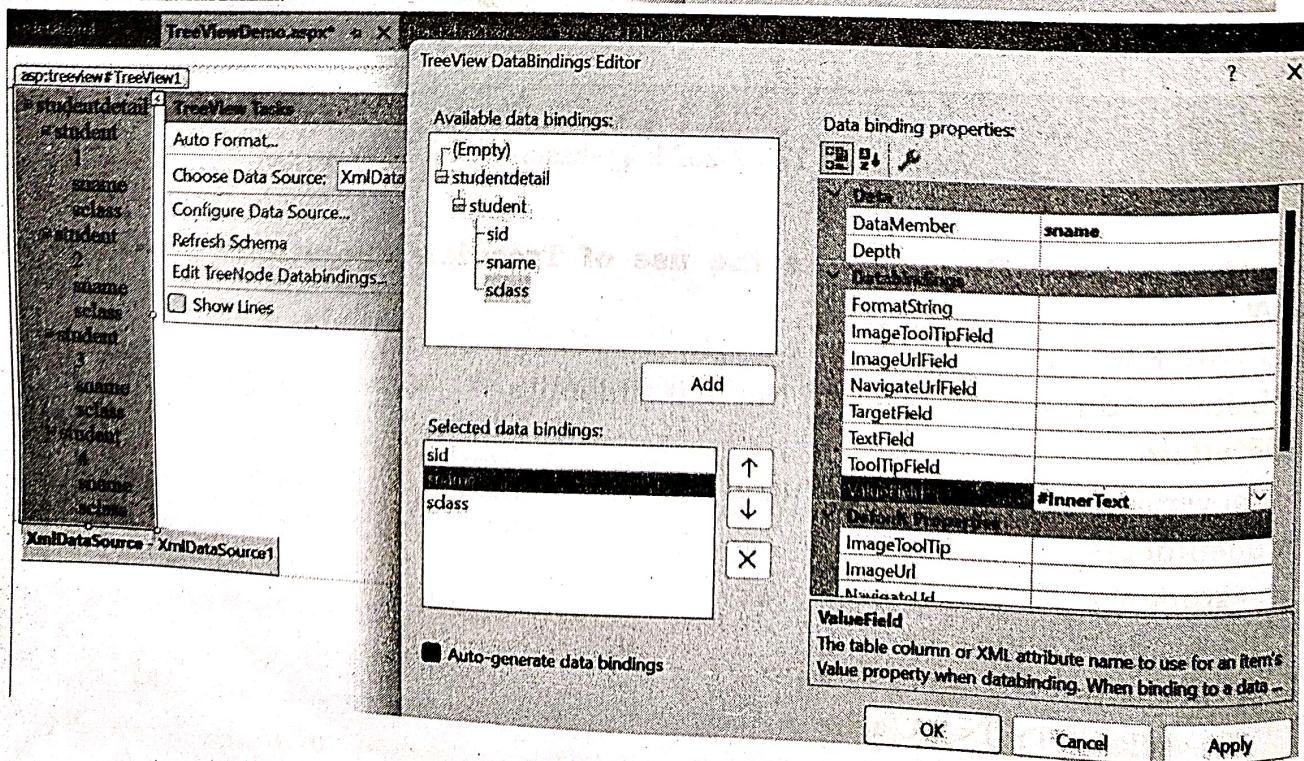
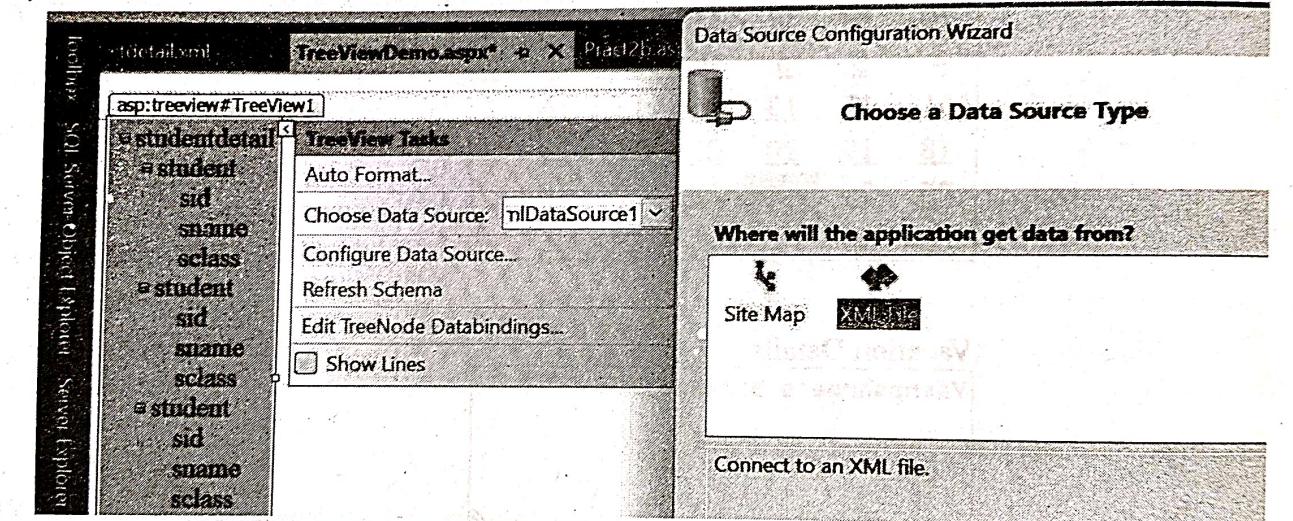
<?xml version="1.0" encoding="utf-8" ?>
<studentdetail>
    <student>
        <sid>1</sid>
        <sname>Tushar</sname>
        <sclass>TYIT</sclass>
    </student>

```

```

<student>
  <sid>2</sid>
  <sname>Sonali</sname>
  <sclass>TYCS</sclass>
</student>
<student>
  <sid>3</sid>
  <sname>Yashashree</sname>
  <sclass>TYIT</sclass>
</student>
<student>
  <sid>4</sid>
  <sname>Vedshree</sname>
  <sclass>TYCS</sclass>
</student>
</studentdetail>

```



TreeViewDemo.aspx

गोप्य

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="TreeViewDemo.aspx.cs" Inherits="TreeViewDemo" %>
<!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:TreeView ID="TreeView1" runat="server"
DataSourceID="XmlDataSource1">
            <DataBindings>
                <asp:TreeNodeBinding DataMember="sid" ValueField="#InnerText" />
                <asp:TreeNodeBinding DataMember="sname" ValueField="#InnerText" />
                <asp:TreeNodeBinding DataMember="sclass" ValueField="#InnerText" />
            </DataBindings>
        </asp:TreeView>
        <asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/stdetail.xml"></asp:XmlDataSource>
    </form>
</body>
</html>

```

← → ⌂ localhost:62666/TreeViewDemo.aspx

```

    └─ studentdetail
        └─ student
            1
            Tushar
            TYIT
        └─ student
            2
            Sonali
            TYCS
        └─ student
            3
            Yashashree
            TYIT
        └─ student
            4
            Vedshree
            TYCS

```

PRACTICAL - 4

Practical - 4(a) : Create a Registration form to demonstrate use of various Validation controls.

ValidationPract.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ValidationPract.aspx.cs" Inherits="ValidationPract" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <script runat="server">
        void ValidateBtn_OnClick(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                lbl1.Text = "Thank You!";
            }
            else
            {
                lbl1.Text = "The text must be exactly 8 characters long!";
            }
        }
        void ServerValidation (object source, ServerValidateEventArgs e)
        {
            if (e.Value.Length == 8)
                e.IsValid = true;
            else
                e.IsValid = false;
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Your name:<br />
            <asp:TextBox runat="server" id="txtName" />
            <asp:RequiredFieldValidator runat="server" id="reqName" ForeColor="Red"
controltovalidate="txtName" errormessage="Please enter your name!" />
            <br /><br />
            Enter age:<br />
            <asp:TextBox runat="server" id="txt1" />
            <asp:RangeValidator ID="RangeValidator2" Type="Integer" runat="server"
ForeColor="Red" ControlToValidate="txt1" MinimumValue="18"
MaximumValue="100" ErrorMessage="Not valid age">Not valid
age</asp:RangeValidator>
        </div>
    </form>
</body>
```

8'8'8'

```

<br /><br />
password:<br />
<asp:TextBox runat="server" id="txt11" TextMode="Password" /><br />
ReEnter Password:<br />
<asp:TextBox runat="server" id="txt12" TextMode="Password" /><br />
<asp:CompareValidator runat="server" id="cmpNumbers" ForeColor="Red"
controltovalidate="txt12" controltocompare="txt11" operator="LessThan"
type="Integer" errormessage="Password should match!" >Password should
match!</asp:CompareValidator><br />
<br />
Email ID:<br />
<asp:TextBox runat="server" id="txtNumber" />
<asp:RegularExpressionValidator runat="server" ForeColor="Red" id="rexNumber"
controltovalidate="txtNumber" errormessage="Please enter valid email address!"
ValidationExpression="\w+([-.\'])\w+@\w+([-.\'])\w+\.\w+([-.\'])\w+*" >Please
enter valid email address!</asp:RegularExpressionValidator>
<br />
Custom text:<br />
<asp:TextBox runat="server" id="txtCustom" />
<asp:CustomValidator id="CustomValidator1" ControlToValidate="txtCustom"
ClientValidationFunction="ServerValidation" Display="Static" ForeColor="Red"
runat="server"/>
<asp:Label id="lbl1" runat="server" Font-Name="Verdana" Font-Size="10pt" Font-
Names="Verdana" ForeColor="Red" /><br />
<asp:Button id="Button1" Text="Validate" OnClick="ValidateBtn_OnClick"
runat="server"/>

```

```

<asp:ValidationSummary ID="ValidationSummary1" runat="server" Style="top:
452px; left: 24px; position: absolute; height: 38px; width: 625px; right: 487px;" />

```

</div>

</form>

</body>

</html>

Web.config

```

<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
  </system.web>
</configuration>

```

OUTPUT :

localhost:3863/Validation X

localhost:3863/ValidationPract.aspx

Apps latest hollywood hinh Tenda 11N Wireless Introduction to th

Your name: Please enter your name!

Enter age: 1 Not valid age

Password:

ReEnter Password:

Password should match!

Email ID: slyv Please enter valid email address!

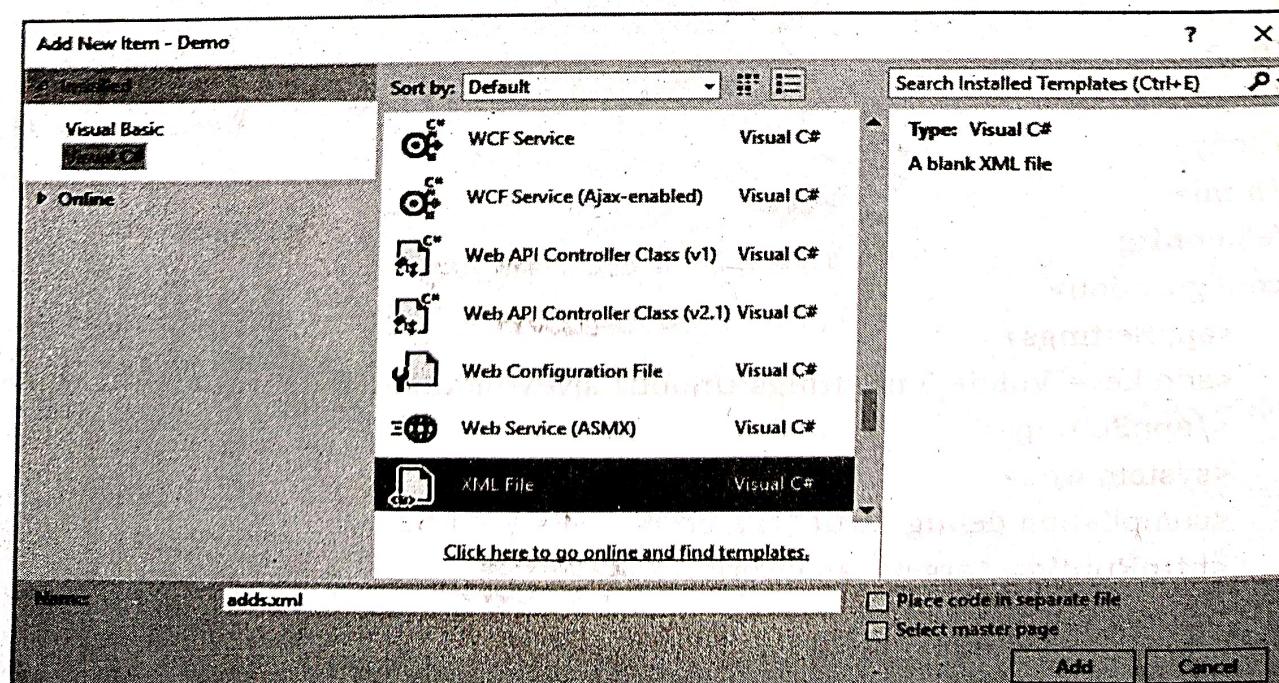
Custom text: yv The text must be exactly 8 characters long!

Validate

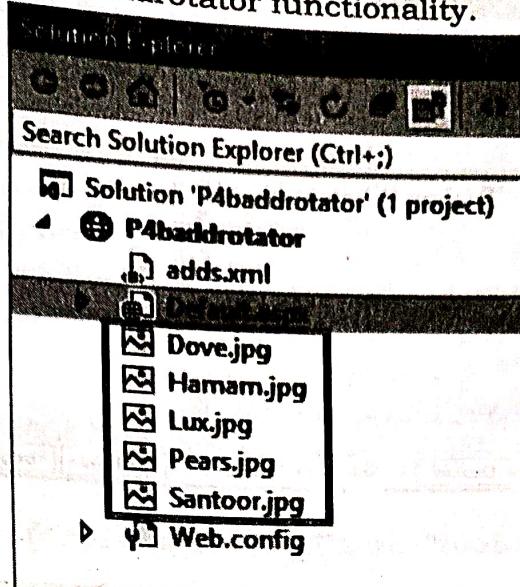
- Please enter your name!
- Not valid age
- Password should match!
- Please enter valid email address!

Practical - 4(b) :Create Web Form to demonstrate use of Adrotator Control.

Add an XML file, name it "adds.xml"



Add images to test out the Adrotator functionality.

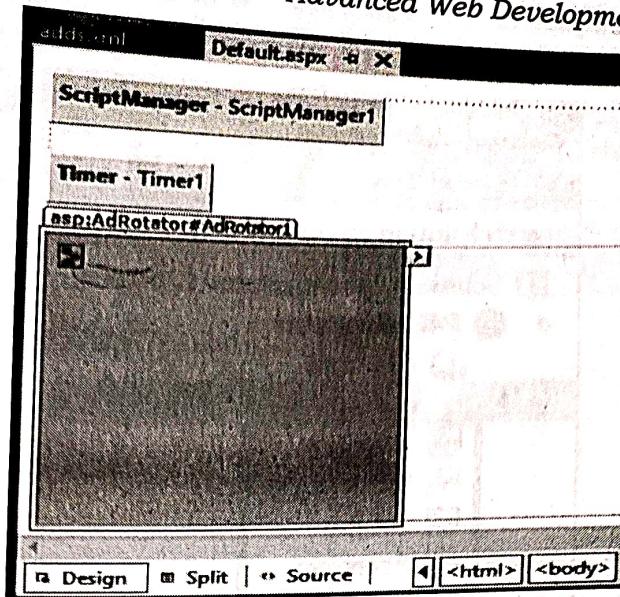


Default.aspx

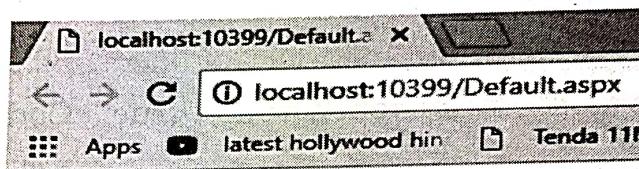
```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <br />
            <asp:Timer ID="Timer1" interval="2000" runat="server">
            </asp:Timer>
            <br />
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <Triggers>
                    <asp:AsyncPostBackTrigger ControlID="Timer1" EventName="Tick" />
                </Triggers>
                <ContentTemplate>
                    <asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile="~/adds.xml"
Height="200px" Width="200px" />
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
        </form>
    </body>
</html>

```



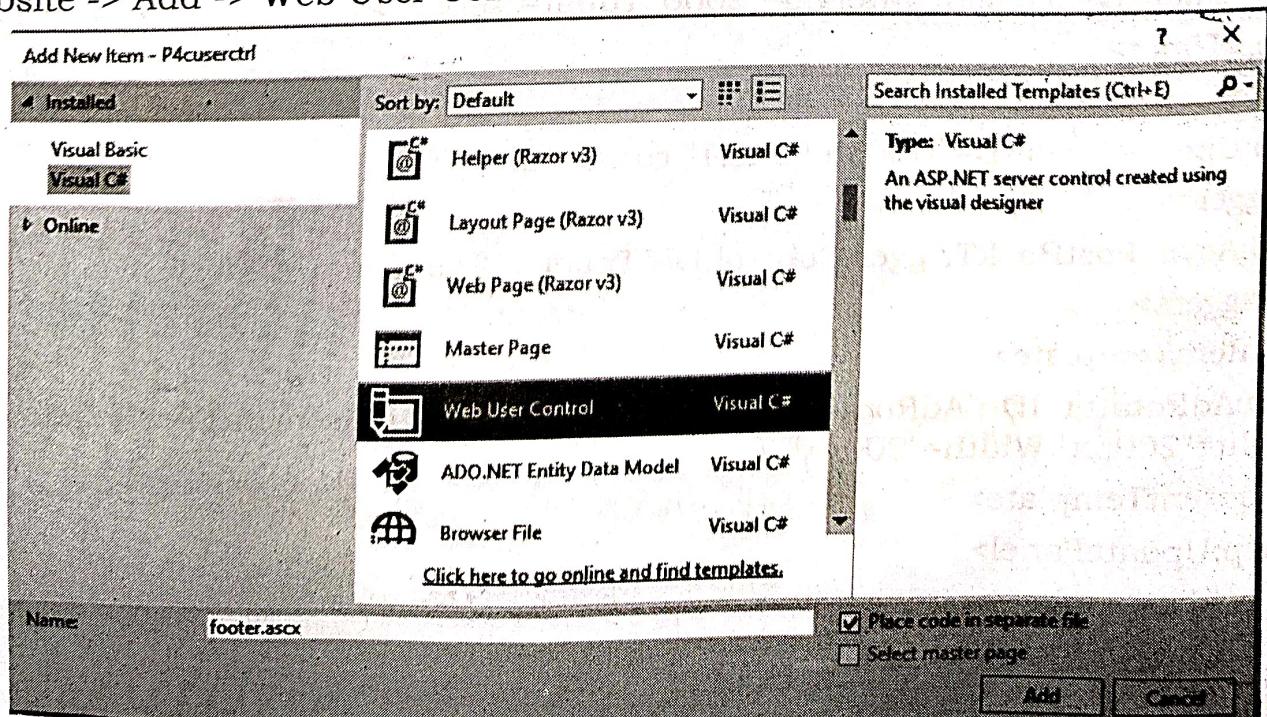
OUTPUT :



Practical – 4(c) : Create Web Form to demonstrate use User Controls.

Add Web User Control

Website -> Add -> Web User Control and Name it 'footer'.



ગુજરાતી

footer.ascx

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="footer.ascx.cs"
Inherits="footer" %>

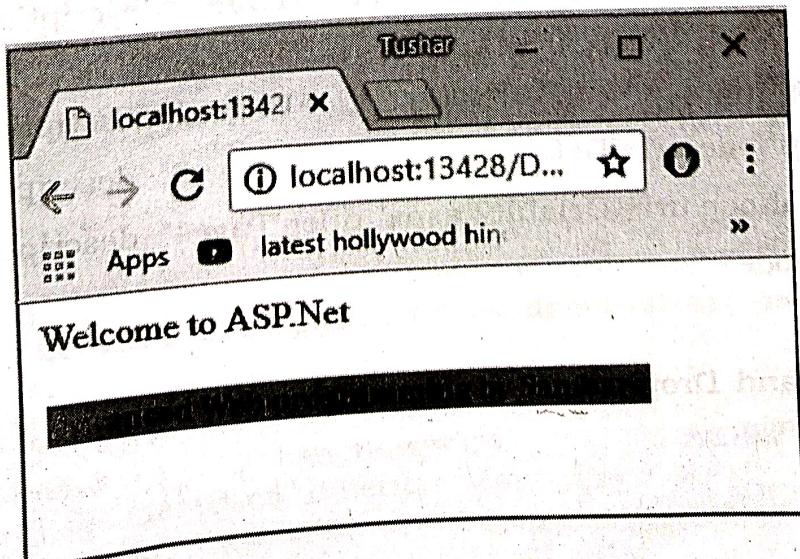

|                                     |
|-------------------------------------|
| Advanced Web programming by Sambare |
|-------------------------------------|


```

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<%@ Register Src="~/footer.ascx" TagName="footer" TagPrefix="STfooter"
%>
<!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="Welcome to ASP.Net ">
            </asp:Label><br /> <br />
        </div>
        <STfooter:footer ID="footer1" runat="server" />
    </form>
</body>
</html>
```

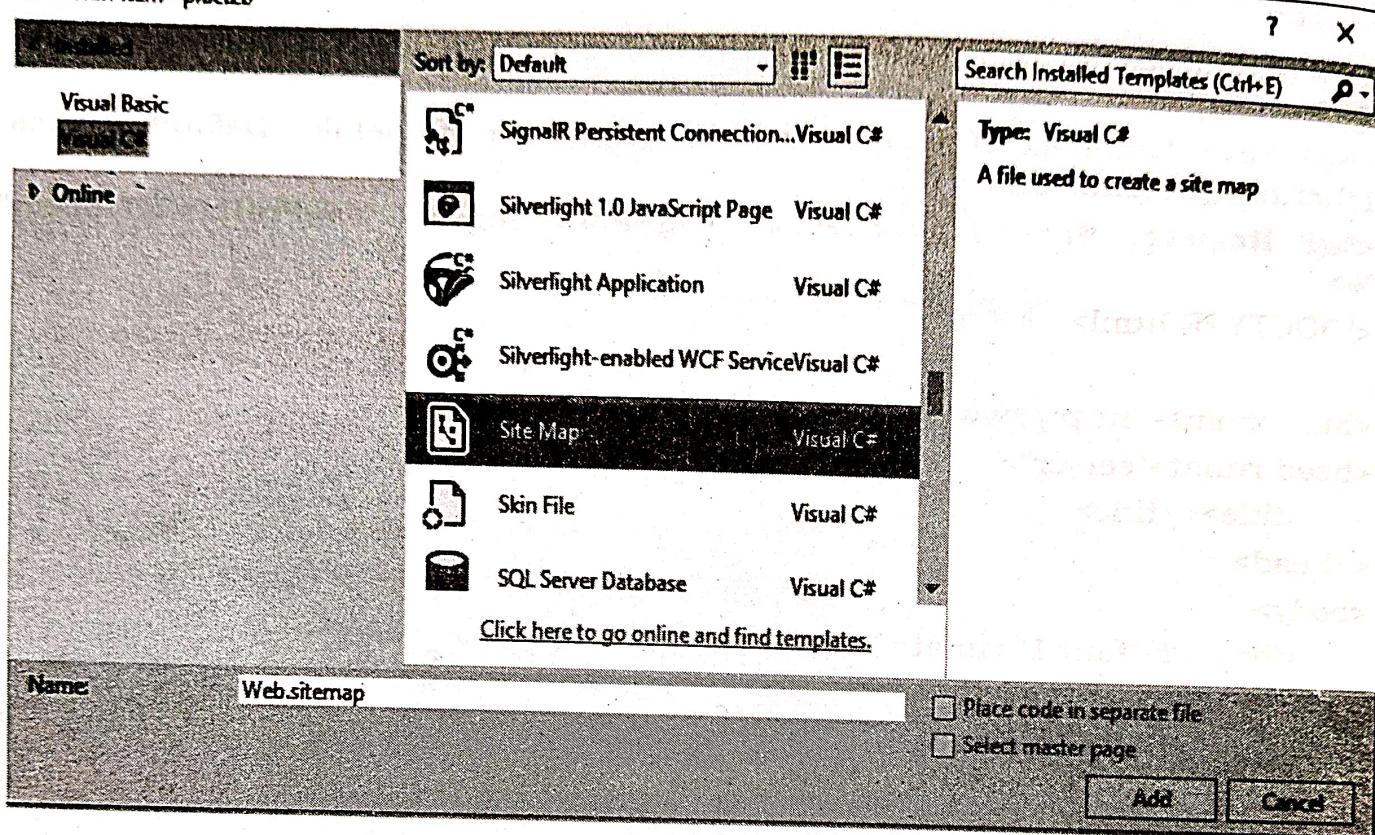
OUTPUT :

PRACTICAL - 5**Practical - 5(a) : Create Web Form to demonstrate use of Website Navigation controls and Site Map.**

Add Site Map File

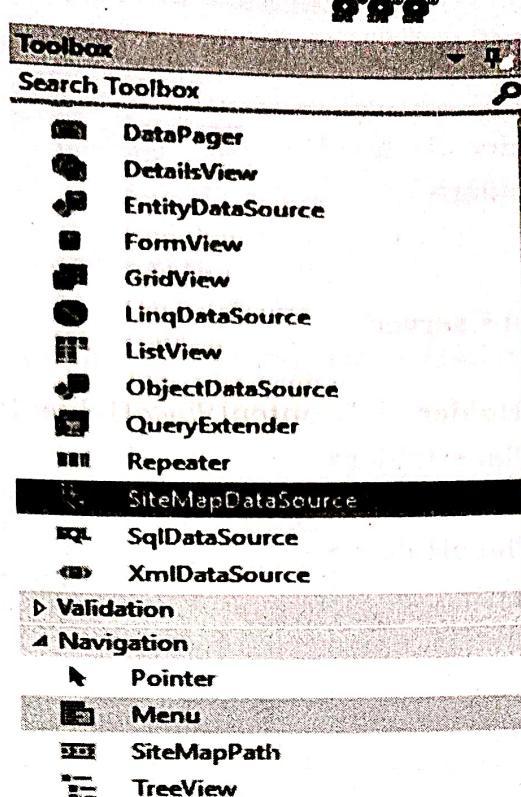
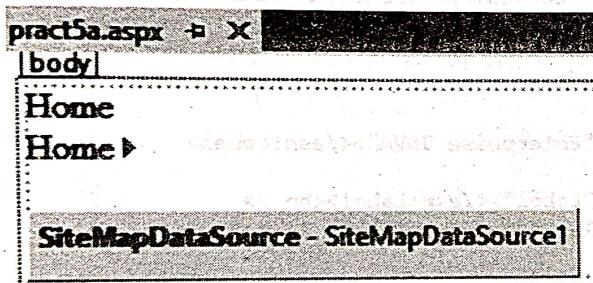
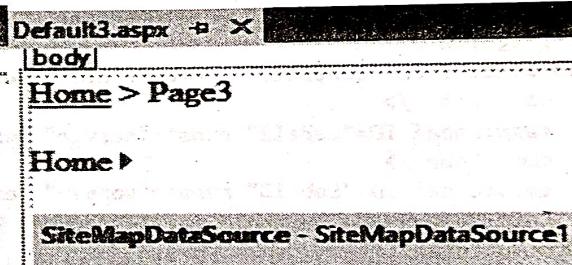
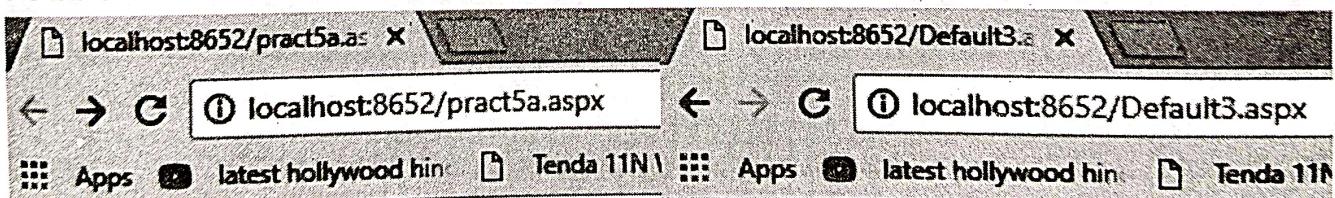
Website -> Add -> Site Map and Name it 'Web.Sitemap'.

Add New Item - pract2b

**Web.Sitemap**

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
    <siteMapNode url="pract5a.aspx" title="Home" description="Home page of our web site">
        <siteMapNode url="clsProp.aspx" title="Page2" description="Page2" />
        <siteMapNode url="Default3.aspx" title="Page3" description="Page3" />
        <siteMapNode url="Default2.aspx" title="Page4" description="Page4" />
    </siteMapNode>
</siteMap>
```

Now, Drag and Drop a SiteMapDataSource control and Menu control from Toolbox to the page.

**Pract5a.aspx****Default3.aspx****OUTPUT :**

Home
Home > Page2
 Page3
 Page4

Home > Page3
Home >

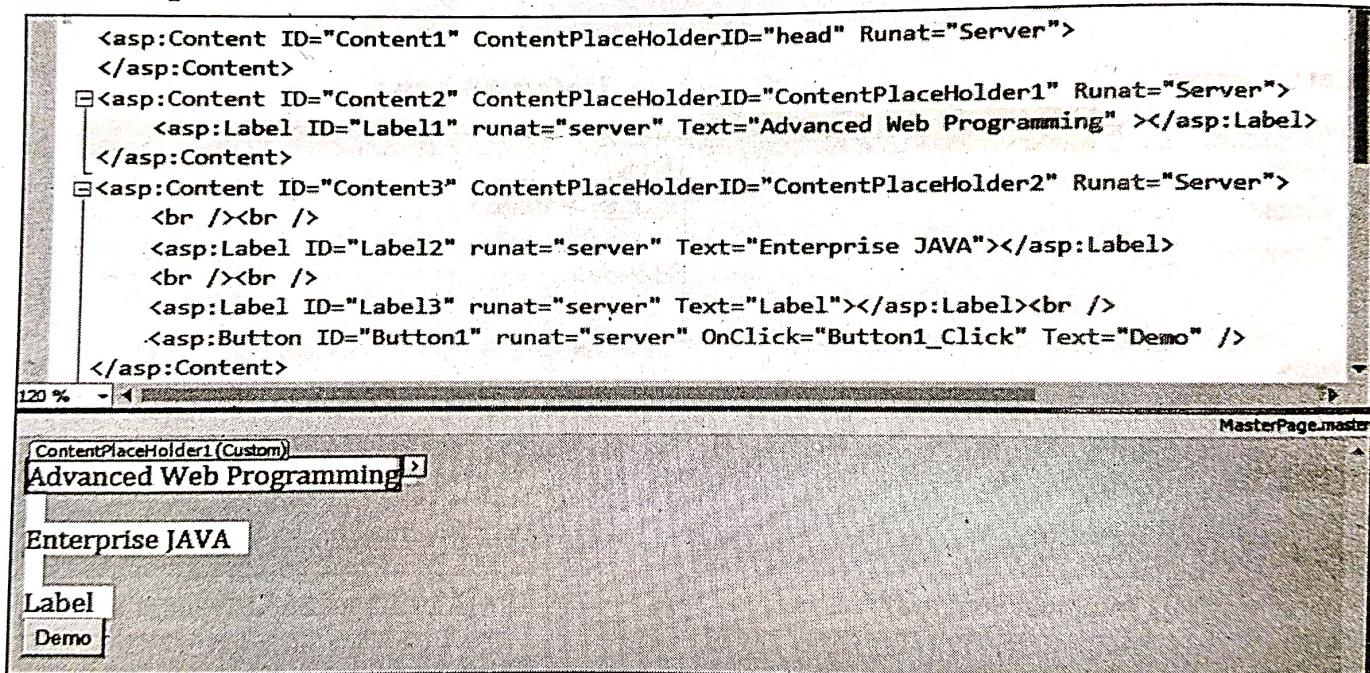
Practical – 5(b) : Create a web application to demonstrate use of Master Page and content page.**MasterPage.master**

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="MasterPage.master.cs" Inherits="MasterPage" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

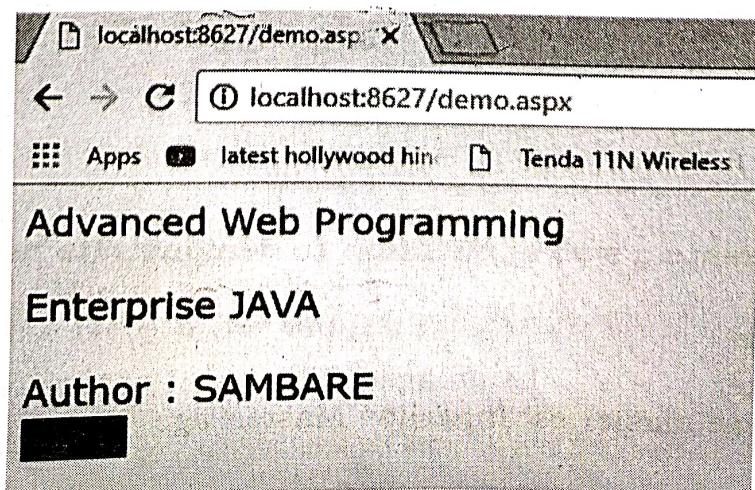
```

<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
                </asp:ContentPlaceHolder>
            <asp:ContentPlaceHolder id="ContentPlaceHolder2" runat="server">
                </asp:ContentPlaceHolder>
            </div>
        </form>
    </body>
</html>
demo.aspx

```



OUTPUT :



Practical - 5(c) : Create a web application to demonstrate various States of ASP.NET Pages.

1. View State:

p3a.aspx

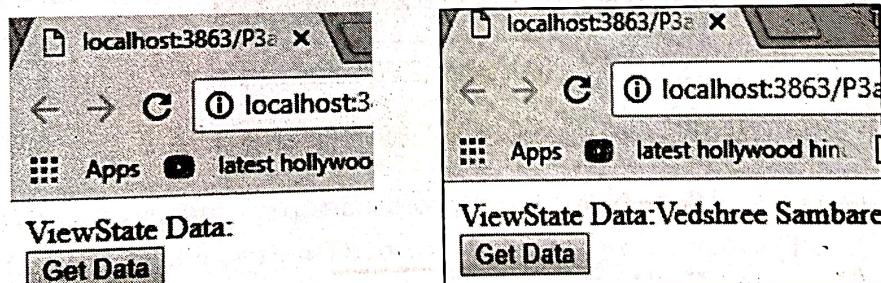
p3a.aspx.cs

public partial class P3a : System.Web.UI.Page

```
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            string str = "Vedshree Sambare";
            if (ViewState["nam"] == null)
            {
                ViewState["nam"] = str;
            }
        }
    }

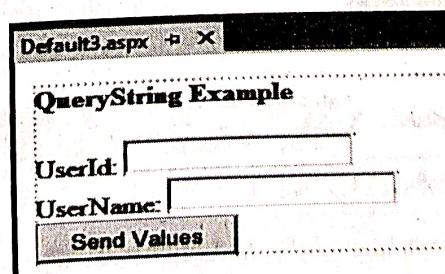
    protected void btn_Click(object sender, EventArgs e)
    {
        lblStr.Text = ViewState["nam"].ToString();
    }
}
```

OUTPUT :



2. Query String

Default3.aspx



Default3.aspx.cs

protected void btnSend_Click(object sender, EventArgs e)

```
{
    Response.Redirect("Default4.aspx?UserId=" + txtUserId.Text +
        "&UserName=" + txtUserName.Text);
}
```

Default4.aspx**QueryString parameter Values in Default3.aspx Page****UserId:[lblUserId]****UserName:[lblUserName]****Default4.aspx.cs**

```
protected void Page_Load(object sender, EventArgs e)
```

{

if (!IsPostBack)

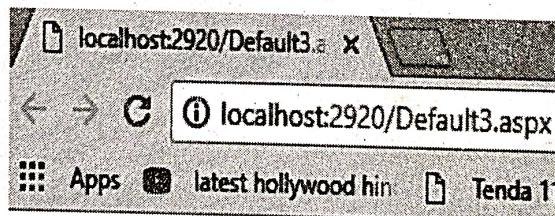
{

lblUserId.Text = Request.QueryString["UserId"];

lblUserName.Text = Request.QueryString["UserName"];

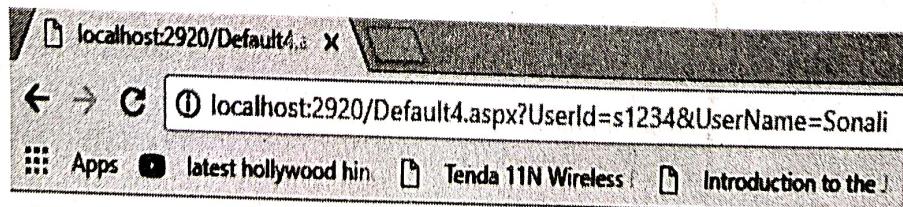
}

}

OUTPUT :**QueryString Example**

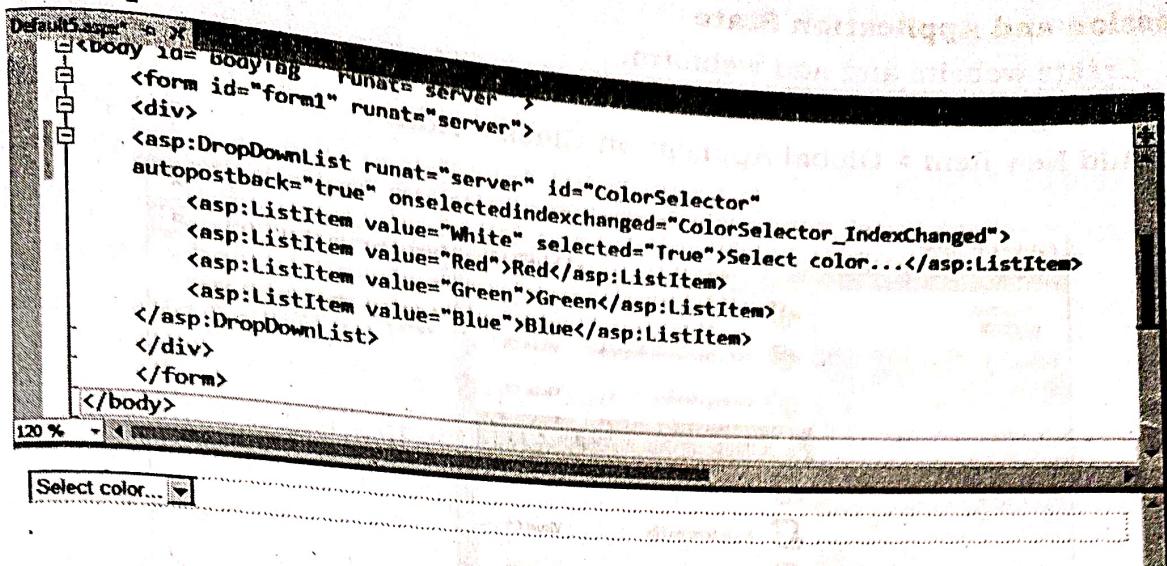
UserId: s1234

UserName: Sonali

Send Values**QueryString parameter Values in Default3.aspx Page****UserId:s1234****UserName:Sonali**

3. Cookies

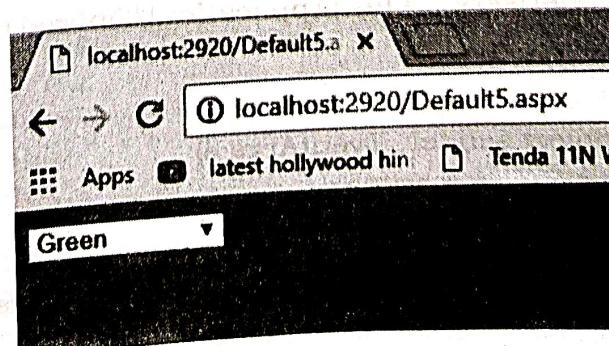
Default5.aspx



Default5.aspx.cs

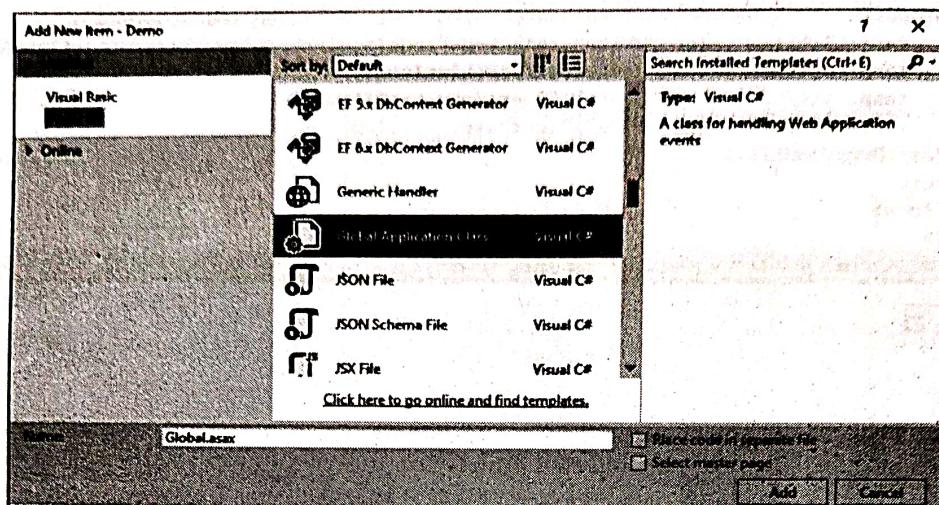
```
using System.Data;
public partial class Default5 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request.Cookies["BackgroundColor"] != null)
        {
            ColorSelector.SelectedValue = Request.Cookies["BackgroundColor"].Value;
            BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
        }
    }
    protected void ColorSelector_SelectedIndexChanged(object sender, EventArgs e)
    {
        BodyTag.Style["background-color"] = ColorSelector.SelectedValue;
        HttpCookie cookie = new HttpCookie("BackgroundColor");
        cookie.Value = ColorSelector.SelectedValue;
        cookie.Expires = DateTime.Now.AddMilliseconds(20);
        Response.SetCookie(cookie);
    }
}
```

OUTPUT :



4. Session and Application State

1. Create website and add webform.
2. Adding a Global.asax to web application.
3. Add New Item > Global Application Class > Add.



Global.asax

```
<%@ Application Language="C#" %>
<script runat="server">

    void Application_Start(object sender, EventArgs e)
    {
        Application["OnlineUsers"] = 0; //application variable
    }

    void Application_End(object sender, EventArgs e)
    {
        // Code that runs on application shutdown
    }

    void Application_Error(object sender, EventArgs e)
    {
        // Code that runs when an unhandled error occurs
    }

    void Session_Start(object sender, EventArgs e)
    {
        Application.Lock();
        Application["OnlineUsers"] = (int)Application["OnlineUsers"] + 1;
        Application.UnLock();
    }

    void Session_End(object sender, EventArgs e)
    {
        Application.Lock();
        Application["OnlineUsers"] = (int)Application["OnlineUsers"] - 1;
        Application.UnLock();
    }
}
```

```

        }
</script>

```

Web.config

```

<?xml version="1.0"?>
<configuration>
    <system.web>
        <sessionState mode="InProc" cookieless="false" timeout="1"/>
        <compilation debug="true" targetFramework="4.5.2" />
        <httpRuntime targetFramework="4.5.2" />
    </system.web>
</configuration>

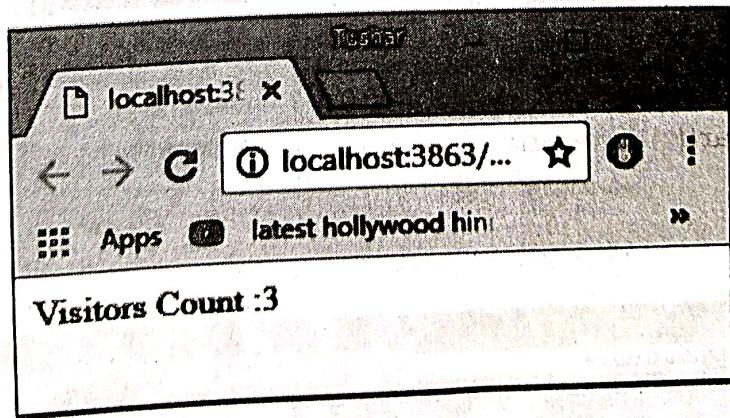
```

Default.aspx

```

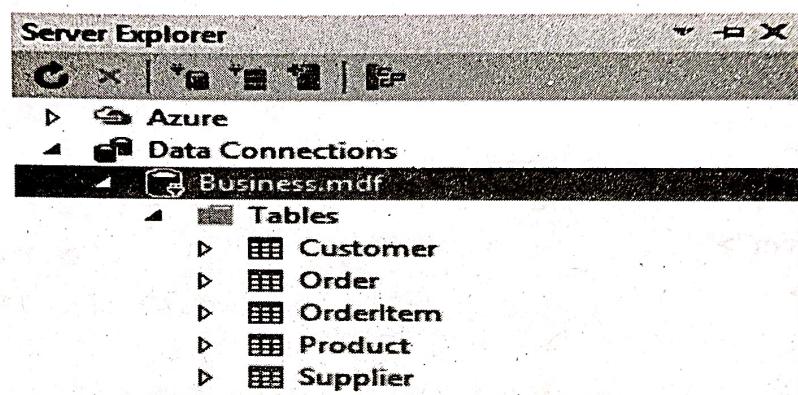
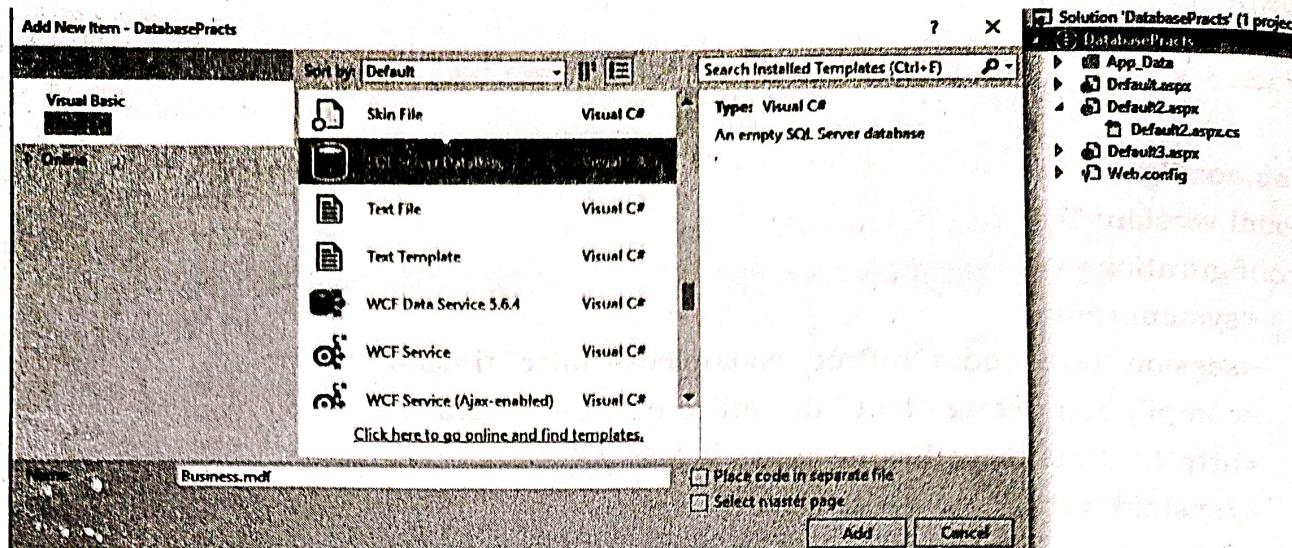
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Visitors Count :<%=Application["OnlineUsers"].ToString( ) %>
        </div>
    </form>
</body>
</html>

```

**** Database Prerequisite**

For Database practical's we have used SQL Server 2014 version.

Note: For Database practical's we have used SQL Server 2014 version.
Here we to add new database in our website, as shown below. Add this database inside App_Data folder.



The structure of the tables are displayed below.

Customer Table

dbo.Customer [Design]

	Name	Data Type	Allow Nulls	Default
Id		int	<input type="checkbox"/>	
FirstName		nvarchar(40)	<input type="checkbox"/>	
LastName		nvarchar(40)	<input type="checkbox"/>	
City		nvarchar(40)	<input checked="" type="checkbox"/>	
Country		nvarchar(40)	<input checked="" type="checkbox"/>	
Phone		nvarchar(20)	<input checked="" type="checkbox"/>	

Keys (1)
PK_CUSTOMER (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (1)
IndexCustomerName (LastName, FirstName)
Foreign Keys (0)
Triggers (0)

Order Table

dbo.Order [Design]

	Name	Data Type	Allow Nulls	Default
Id		int	<input type="checkbox"/>	
OrderDate		datetime	<input type="checkbox"/>	(getdate())
OrderNumber		nvarchar(10)	<input checked="" type="checkbox"/>	
CustomerId		int	<input type="checkbox"/>	
TotalAmount		decimal(12,2)	<input checked="" type="checkbox"/>	((0))

Keys (1)
PK_ORDER (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (2)
IndexOrderCustomerId (CustomerId)
IndexOrderOrderDate (OrderDate)
Foreign Keys (1)
FK_ORDER_REFERENCE_CUSTOMER (Id)
Triggers (0)

गोपनीय

OrderItem Table

dbo.OrderItem [Design] ➔ X

Update Script File: **dbo.OrderItem.sql**

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
OrderId	int	<input type="checkbox"/>	
ProductId	int	<input type="checkbox"/>	
UnitPrice	decimal(12,2)	<input type="checkbox"/>	((0))
Quantity	int	<input type="checkbox"/>	((1))

- ▲ Keys (1)
 - PK_ORDERITEM (Primary Key, Clustered: Id)
- Check Constraints (0)
- Indexes (2)
 - IndexOrderItemOrderId (OrderId)
 - IndexOrderItemProductId (ProductId)
- Foreign Keys (2)
 - FK_ORDERITE_REFERENCE_ORDER (Id)
 - FK_ORDERITE_REFERENCE_PRODUCT (Id)
- Triggers (0)

Product Table

dbo.Product [Design] ➔ X

Update Script File: **dbo.Product.sql**

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
ProductName	nvarchar(50)	<input type="checkbox"/>	
SupplierId	int	<input type="checkbox"/>	
UnitPrice	decimal(12,2)	<input checked="" type="checkbox"/>	((0))
Package	nvarchar(30)	<input checked="" type="checkbox"/>	
IsDiscontinued	bit	<input type="checkbox"/>	((0))

- ▲ Keys (1)
 - PK_PRODUCT (Primary Key, Clustered: Id)
- Check Constraints (0)
- Indexes (2)
 - IndexProductSupplierId (SupplierId)
 - IndexProductName (ProductName)
- Foreign Keys (1)
 - FK_PRODUCT_REFERENCE_SUPPLIER (Id)
- Triggers (0)

Supplier Table

dbo.Supplier [Design] ➔ X

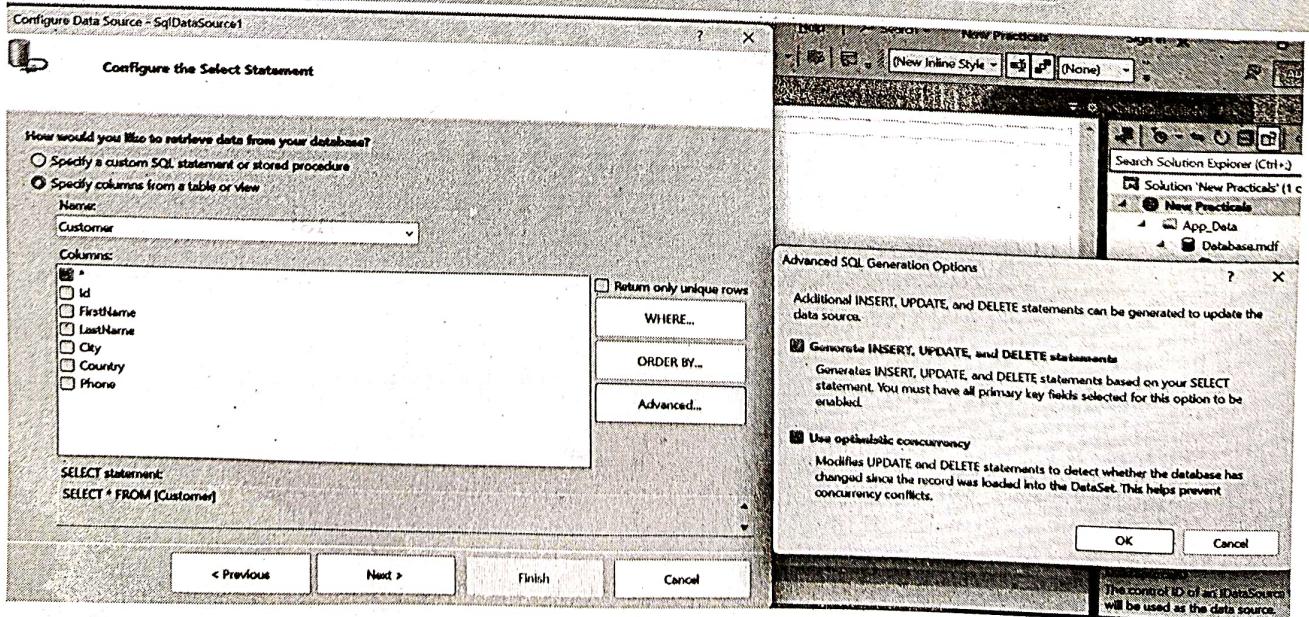
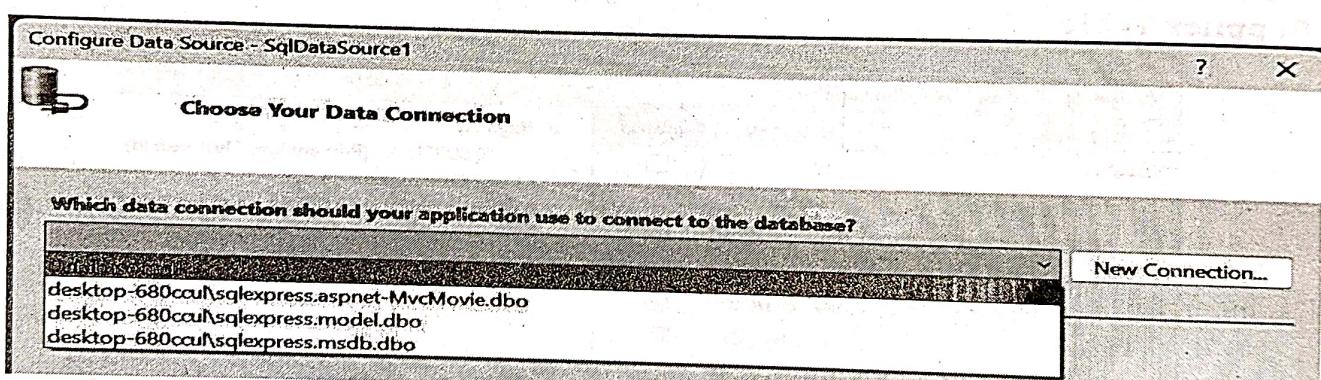
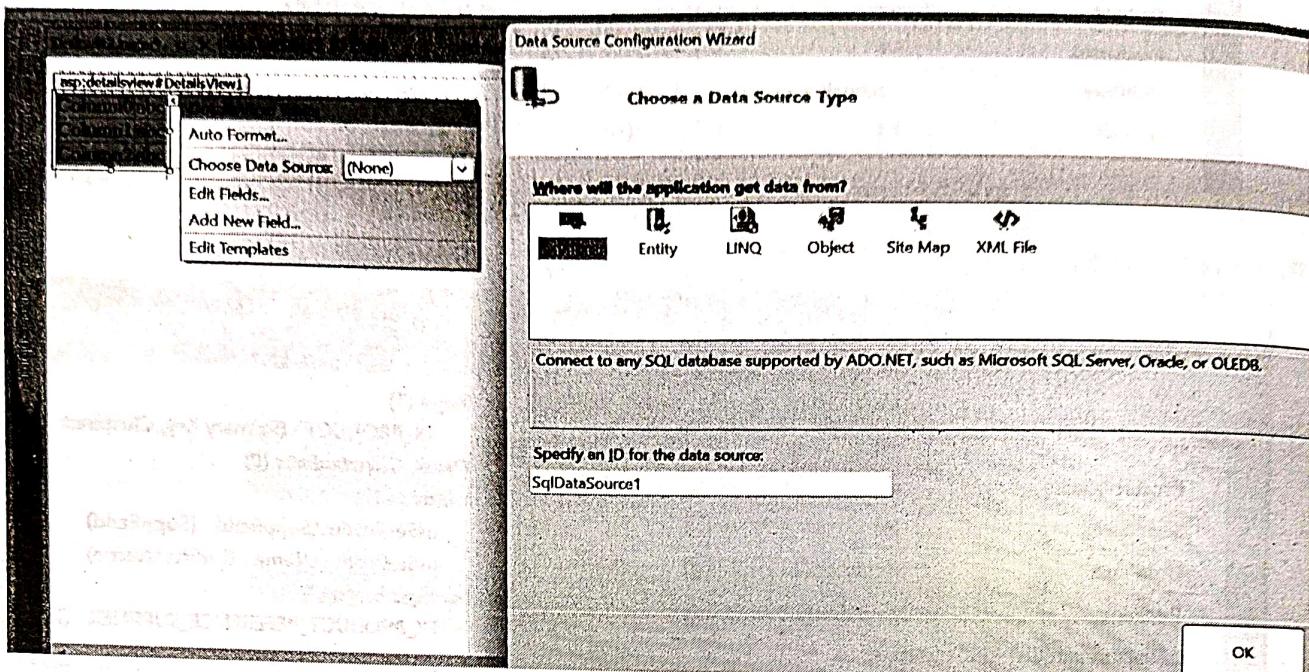
Update Script File: **dbo.Supplier.sql**

Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
CompanyName	nvarchar(40)	<input type="checkbox"/>
ContactName	nvarchar(50)	<input checked="" type="checkbox"/>
ContactTitle	nvarchar(40)	<input checked="" type="checkbox"/>
City	nvarchar(40)	<input checked="" type="checkbox"/>
Country	nvarchar(40)	<input checked="" type="checkbox"/>
Phone	nvarchar(30)	<input checked="" type="checkbox"/>
Fax	nvarchar(30)	<input checked="" type="checkbox"/>

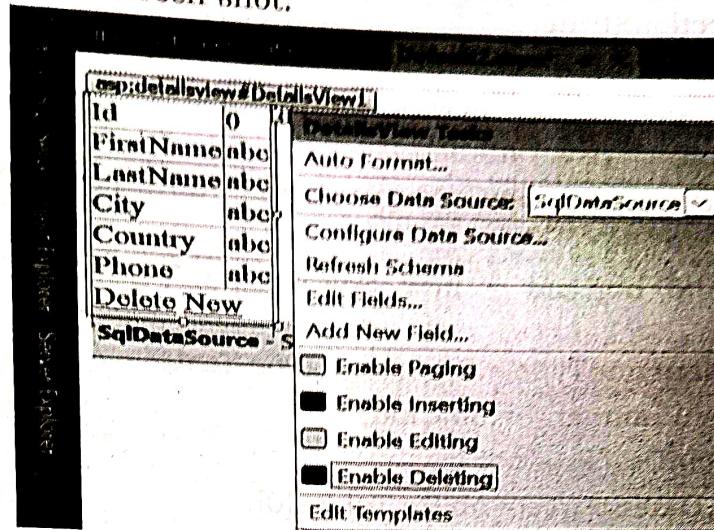
- ▲ Keys (1)
 - PK_SUPPLIER (Primary Key, Clustered: Id)
- Check Constraints (0)
- Indexes (2)
 - IndexSupplierName (CompanyName)
 - IndexSupplierCountry (Country)
- Foreign Keys (0)
- Triggers (0)

PRACTICAL – 6**Practical – 6 (a) : Create a web application for inserting and deleting records from a database.**

- Drag the details view on the web page and follow the steps shown as per shown in the screenshots.



- Select Advanced, in next window Select both the options as shown above and click ok.
- In next window Select Finish.
- Once again go to Design mode of aspx web page and select the options shown below in screen shot.



- Now Run the application and follow the steps shown in output screens.

OUTPUT :

Id	1
FirstName	Tushar
LastName	Sambare
City	Mumbai
Country	India
Phone	9988776654
Delete New	

Id	5
FirstName	Mohoj
LastName	Mendis
City	Colombo
Country	Srilanka
Phone	7784576574
Insert Cancel	

Id	1
FirstName	Tushar
LastName	Sambare
City	Mumbai
Country	India
Phone	9988776654
Delete New	

Practical - 6 (b) : Create a web application to display Using Disconnected Data Access and Databinding using GridView

- Create a webpage using Button and GridView Control.
- Modify the display setting of GridView using Auto Format option.
- Change the Text of Button.
- Write the code provided below in C# code behind file inside the Button click event code segment.
- Then run the web application.

Column0	Column1	Column2
abc	abc	abc

Code of C# Code behind file

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
    string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    SqlConnection con = new SqlConnection(connStr);
    SqlDataAdapter objDa = new SqlDataAdapter();
    DataSet objDs = new DataSet();
    using (SqlConnection objConn = new SqlConnection(connStr))
    {
        SqlCommand objCmd = new SqlCommand("Select * from Customer", objConn);
        objCmd.CommandType = CommandType.Text;
        objDa.SelectCommand = objCmd;
        objDa.Fill(objDs, "Product");
        GridView1.DataSource = objDs.Tables[0];
        GridView1.DataBind();
    }
}
```

OUTPUT :

	Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321	
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729	
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932	
4	Thomas	Hardy	London	UK	(171) 555-7788	
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65	

PRACTICAL - 7

Practical - 7 (a) : Create a web application to demonstrate the use of different types of Cookies.

**Select the Company**

- Apple
- Dell
- Lenevo
- Acer
- Sony
- Wipro

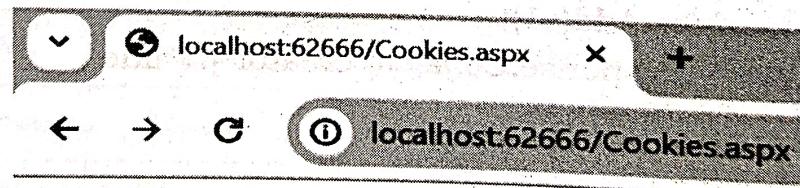
Submit**[Label3]****Cookies.aspx.cs :**

```
public partial class Cookies : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Creating HttpCookie instance by name "creator"
        HttpCookie c1 = new HttpCookie("creator");
        // Assigning value to the created cookie
        c1.Value = "Dr. Tushar Sambare";
        // Adding Cookie to the response instance
        Response.Cookies.Add(c1);
        String author = Response.Cookies["creator"].Value;
        Label2.Text = author;
        //Second Cookie
        Response.Cookies["comp"].Expires = DateTime.Now.AddDays(-1);
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label3.Text = "";
        // ----- Adding Cookies -----
        if (apple.Checked)
            Response.Cookies["comp"]["apple"] = "apple";
        if (dell.Checked)
            Response.Cookies["comp"]["dell"] = "dell";
        if (lenevo.Checked)
            Response.Cookies["comp"]["lenevo"] = "lenevo";
        if (acer.Checked)
            Response.Cookies["comp"]["acer"] = "acer";
        if (sony.Checked)
            Response.Cookies["comp"]["sony"] = "sony";
    }
}
```

```

Response.Cookies["comp"]["sony"] = "sony";
if (wipro.Checked)
    Response.Cookies["comp"]["wipro"] = "wipro";
// ----- Fetching Cookies -----
if (Request.Cookies["comp"].Values.ToString() != null)
{
    if (Request.Cookies["comp"]["apple"] != null)
        Label3.Text += Request.Cookies["comp"]["apple"] + " ";
    if (Request.Cookies["comp"]["dell"] != null)
        Label3.Text += Request.Cookies["comp"]["dell"] + " ";
    if (Request.Cookies["comp"]["lenevo"] != null)
        Label3.Text += Request.Cookies["comp"]["lenevo"] + " ";
    if (Request.Cookies["comp"]["acer"] != null)
        Label3.Text += Request.Cookies["comp"]["acer"] + " ";
    if (Request.Cookies["comp"]["sony"] != null)
        Label3.Text += Request.Cookies["comp"]["sony"] + " ";
    if (Request.Cookies["comp"]["wipro"] != null)
        Label3.Text += Request.Cookies["comp"]["wipro"] + " ";
}
else Label3.Text = "Please select your choice";
Response.Cookies["comp"].Expires = DateTime.Now.AddDays(-1);
}
}

```

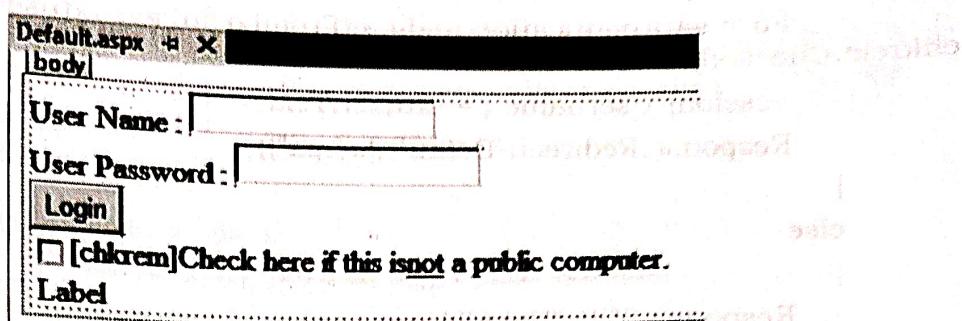
OUTPUT :**Dr. Tushar Sambare****Select the Company**

- Apple
- Dell
- Lenevo
- Acer
- Sony
- Wipro

Submit**apple lenevo**

Practical - 7 (b) : Create a web application to demonstrate Form Security and Windows Security with proper Authentication and Authorization

Default.aspx



Default.aspx.cs

```

using System;
using System.Web.Security;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Check if the user is already authenticated
        if (User.Identity.IsAuthenticated)
        {
            Response.Redirect("Welcome.aspx");
        }
    }

    protected bool authenticate(String uname, String pass)
    {
        if (uname == "Yash")
        {
            if (pass == "yash123")
                return true;
        }

        if (uname == "Ved")
        {
            if (pass == "ved123")
                return true;
        }

        if (uname == "Sam")
        {
            if (pass == "sam123")
                return true;
        }

        return false;
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        // Check if the user has checked the checkbox
        if (chkrem.Checked)
        {
            // Set the cookie to remember the user
            FormsAuthentication.SetAuthCookie(uname, true);
        }
        else
        {
            // Clear the cookie
            FormsAuthentication.SignOut();
        }

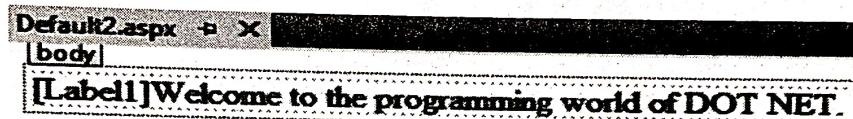
        // Redirect to the welcome page
        Response.Redirect("Welcome.aspx");
    }
}

```

```

    {
        if (authenticate(txtuser.Text, txtpwd.Text))
        {
            FormsAuthentication.RedirectFromLoginPage(txtuser.Text,
            chkrem.Checked);
            Session["Username"] = txtuser.Text;
            Response.Redirect("Default2.aspx");
        }
        else
        {
            Response.Write("Invalid user name or password");
        }
    }
}

```

Default2.aspx**Default2.aspx.cs**

```

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["Username"] != null)
    {
        //Response.Write(Session["Username"].ToString());
        Label1.Text = Session["Username"].ToString();
    }
}

```

Web.config

```

<configuration>
    <system.web>
        <authentication mode="Forms">
            <forms loginUrl ="Default.aspx"/>
        </authentication>
        <authorization>
            <deny users="?" />
        </authorization>
        <compilation debug="true" targetFramework="4.5.2" />
        <httpRuntime targetFramework="4.5.2" />
    </system.web>
</configuration>

```

Practicals

OUTPUT :

localhost:1644/Default.aspx

User Name : Yash
User Password : yash123

Check here if this is not a public computer.

localhost:1644/Default2.aspx

Yash Welcome to the programming world of DOT NET.

PRACTICAL - 8

Practical - 8 (a) : Create a web application for inserting and deleting records from a database. (Using Execute-Non Query).

Create a web page with TextBox, and Two Button and one Label control as shown below.

And follow the database related steps same as it is in previous examples.

Code of C# Code behind file

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
    string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    SqlConnection con = new SqlConnection(connStr);
    string InsertQuery = "insert into customer values(@fname, @lname, @city, @country, @phone)";
    SqlCommand cmd = new SqlCommand(InsertQuery, con);
    cmd.Parameters.AddWithValue("@fname", TextBox1.Text);
    cmd.Parameters.AddWithValue("@lname", TextBox2.Text);
    cmd.Parameters.AddWithValue("@city", TextBox3.Text);
    cmd.Parameters.AddWithValue("@country", TextBox4.Text);
    cmd.Parameters.AddWithValue("@phone", TextBox5.Text);
    con.Open();
    cmd.ExecuteNonQuery();
    Label1.Text = "Record Inserted Successfully.";
    con.Close();
}
```

```
protected void Button2_Click(object sender, EventArgs e)
```

```
{
    string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    SqlConnection con = new SqlConnection(connStr);
    string InsertQuery = "delete from customer where FirstName=@fname";
    SqlCommand cmd = new SqlCommand(InsertQuery, con);
    cmd.Parameters.AddWithValue("@fname", TextBox1.Text);
    con.Open();
```

```

    cmd.ExecuteNonQuery();
    Label1.Text = "Record Deleted Successfully.";
    con.Close();
}

```

OUTPUT :

localhost:54415/Default5.aspx

Enter Customer First Name :	Tushar
Enter Customer Last Name :	Sambare
Enter Customer City :	Mumbai
Enter Customer Country :	India
Enter Customer Phone :	1212121212

Insert Record **Delete Record**

Record Inserted Successfully.

localhost:54415/Default5.aspx

Enter Customer First Name :	Demo
Enter Customer Last Name :	
Enter Customer City :	
Enter Customer Country :	
Enter Customer Phone :	

Insert Record **Delete Record**

Record Deleted Successfully.

Practical - 8 (b) : Create a web application for user defined exception handling.

using System;

```

class UserDefinedException : Exception
{
    public MyException(string str)
    {
        Console.WriteLine("User defined exception");
    }
}

```

class HandledException

```

public static void Main()
{
    try
    {
        throw new UserDefinedException ("New User Defined Exception");
    }
    catch(Exception e)
    {
        Label1.Text="Exception caught here" + e.ToString();
    }
    Label2.Text ="Final Statement that is executed ";
}

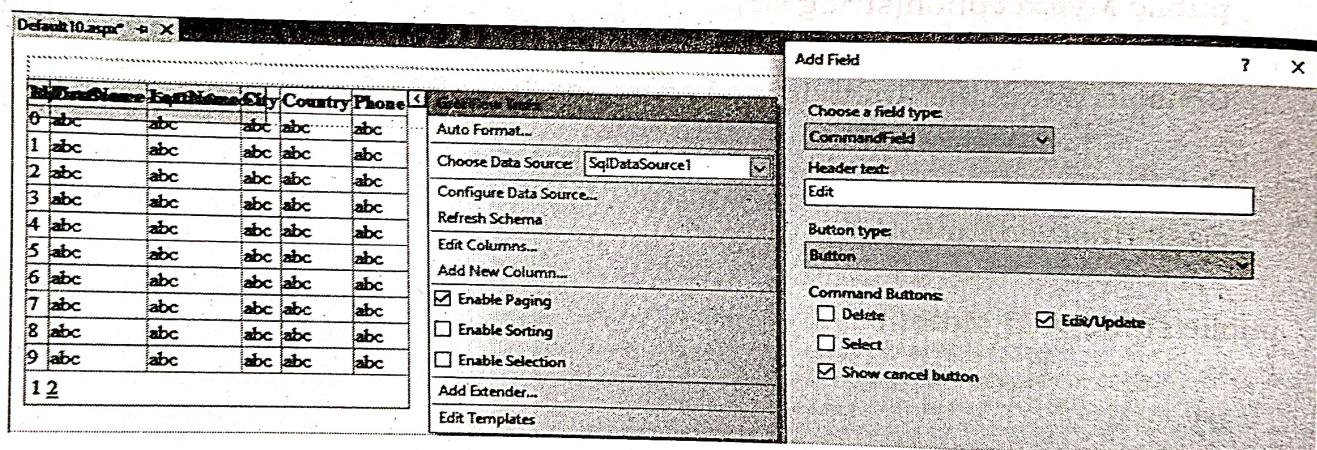
```

PRACTICAL - 9

Practical - 9 (a) : Create a web application to demonstrate use of GridView button column and GridView events along with paging and sorting.

Note : This practical intends to demonstrate the capabilities of GridView Control with various options. Following steps and code demonstrates the **Edit** and **Update** Capability of GridView with some additional features in aspx page code, so there is no need of C# code.

1. Add GridView Control to web page.
2. Select **Choose Data Source-> New Data Source** Option from **GridView Tasks**.
3. Configure the new SQLDataSource as explained in **Practical 6(a)**.
4. Select **Add New column** Option from **GridView Tasks**.
5. In **Add Field** Window Choose field type as **CommandField**, Header Text as **Edit**, Button type as **Button**, also select **Edit/Update** CheckBox with **Show Cancel Button** CheckBox.
6. Again Select **Add New column** Option from **GridView Tasks**.
7. In **Add Field** Window Choose field type as **CommandField**, Header Text as **Delete**, Button type as **Button**, also select **Delete** CheckBox.
8. Select **Enable Paging** Checkbox Option from **GridView Tasks**.
9. From **Auto Format** option select the appropriate design for GridView.



The GridView Code of aspx page.

Note : Code that user has to write additionally in SqlDataSource is shown in bold letters.

```
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AutoGenerateColumns="False" CssClass="auto-style1" DataKeyNames="Id"
DataSourceID="SqlDataSource1" style="z-index: 1" BackColor="White"
BorderColor="#CCCCCC" BorderStyle="None" BorderWidth="1px" CellPadding="4"
ForeColor="Black" GridLines="Horizontal">

    <Columns>
        <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
ReadOnly="True" SortExpression="Id" />
        <asp:BoundField DataField="FirstName" HeaderText="FirstName"
SortExpression="FirstName" />
        <asp:BoundField DataField="LastName" HeaderText="LastName"
SortExpression="LastName" />
    </Columns>
</asp:GridView>
```

```

<asp:BoundField DataField="City" HeaderText="City" SortExpression="City" />
<asp:BoundField DataField="Country" HeaderText="Country" SortExpression="Country" />
<asp:BoundField DataField="Phone" HeaderText="Phone" SortExpression="Phone" />
<asp:CommandField ButtonType="Button" HeaderText="Edit" ShowEditButton="True" ShowHeader="True"></asp:CommandField>
</Columns>
<FooterStyle BackColor="#CCCC99" ForeColor="Black" />
<HeaderStyle BackColor="#333333" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="White" ForeColor="Black" HorizontalAlign="Right" />
<SelectedRowStyle BackColor="#CC3333" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F7F7F7" />
<SortedAscendingHeaderStyle BackColor="#4B4B4B" />
<SortedDescendingCellStyle BackColor="#E5E5E5" />
<SortedDescendingHeaderStyle BackColor="#242121" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\Business.mdf;Integrated Security=True" ProviderName="System.Data.SqlClient" SelectCommand="SELECT * FROM [Customer]" UpdateCommand="UPDATE Customer SET FirstName=@FirstName, LastName=@LastName, City=@City, Country=@Country, Phone=@Phone Where Id=@Id" >
<UpdateParameters>
<asp:Parameter Name="Id" Type="Int32" />
<asp:Parameter Name="FirstName" Type="String" />
<asp:Parameter Name="LastName" Type="String" />
<asp:Parameter Name="City" Type="String" />
<asp:Parameter Name="Country" Type="String" />
<asp:Parameter Name="Phone" Type="String" />
</UpdateParameters>
</asp:SqlDataSource>

```

OUTPUT :

localhost:54415/Default10.aspx

Id	FirstName	LastName	City	Country	Phone	Edit
1	Tushar	Anders	Berlin	Germany	030-0074321	Edit
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729	Edit



← → C ① localhost:54415/Default10.aspx

	Id	FirstName	LastName	City	Country	Phone	Edit
1	Tushar	Sambare	Mumbai	India	22222222	(5) 555-4729	
2	Ana	Trujillo	México D.F.	Mexico		(5) 555-4729	

← → C ① localhost:54415/Default10.aspx

	Id	FirstName	LastName	City	Country	Phone	Edit
1	Tushar	Sambare	Mumbai	India	22222222	(5) 555-4729	

Practical – 9 (b) : Create a web application to demonstrate data binding using DetailsView and FormView Control

1. Add new web page with **DetailsView** and **FormView** Controls. These controls are available under Data controls in Toolbox.
2. After adding these controls, configure the Data Source property of these controls in similar manner as explained in **Practical 6 (a)**.
3. After finishing the Data Source work, we can also use the Auto Format option and Edit template option to configure the display setting of the controls.
4. And now just run the application.

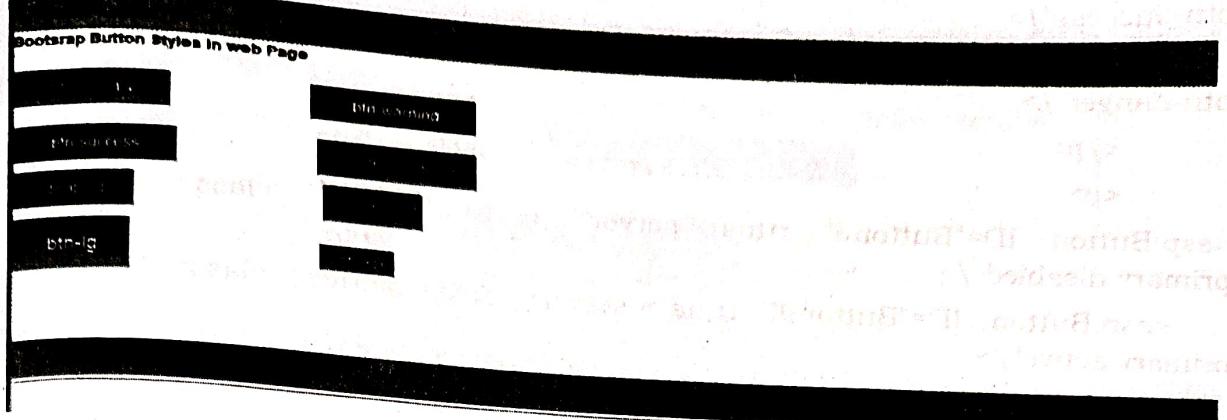
OUTPUT :

← → C ① localhost:54415/Default7.aspx

Id: 1 FirstName: Maria LastName: Anders City: Berlin Country: Germany Phone: 030-0074321	<table border="1"> <tbody> <tr><td>Id</td><td>1</td></tr> <tr><td>ProductName</td><td>Chai</td></tr> <tr><td>SupplierId</td><td>1</td></tr> <tr><td>UnitPrice</td><td>18.00</td></tr> <tr><td>Package</td><td>10 boxes x 20 bags</td></tr> <tr><td>IsDiscontinued</td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>	Id	1	ProductName	Chai	SupplierId	1	UnitPrice	18.00	Package	10 boxes x 20 bags	IsDiscontinued	<input checked="" type="checkbox"/>
Id	1												
ProductName	Chai												
SupplierId	1												
UnitPrice	18.00												
Package	10 boxes x 20 bags												
IsDiscontinued	<input checked="" type="checkbox"/>												

PRACTICAL - 10

Practical - 10 (a) : Create a web application to demonstrate JS Bootstrap Button.

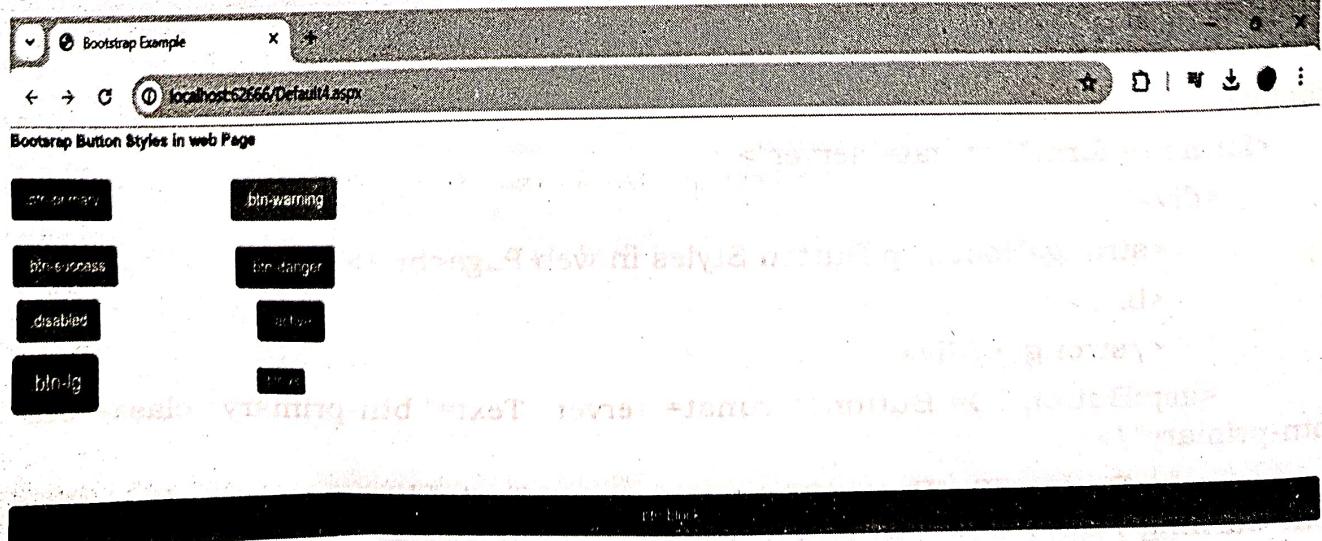
**Aspx page code**

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default4.aspx.cs" Inherits="Default4" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">
    </script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <strong>Bootstrap Button Styles in web Page<br />
            <br />
            </strong></div>
            <asp:Button ID="Button1" runat="server" Text=".btn-primary" class="btn btn-primary"/>
            <asp:Button ID="Button2" runat="server" Text=".btn-warning" class="btn btn-warning"/>
            <br />
            <br />
    </form>
</body>
```

```

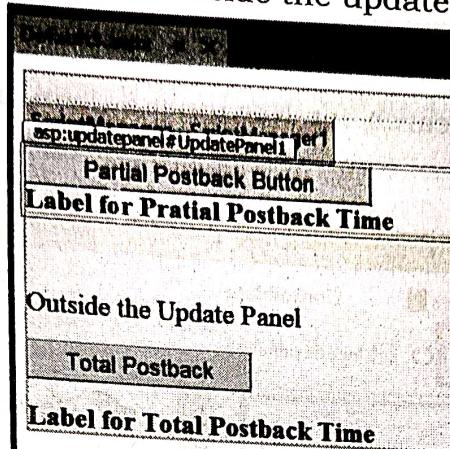
<p>
    <asp:Button ID="Button3" runat="server" Text=".btn-success" class="btn
    btn-success"/>
    <asp:Button ID="Button4" runat="server" Text=".btn-danger" class="btn
    btn-danger"/>
</p>
<p>
    <asp:Button ID="Button6" runat="server" Text=".disabled" class="btn
    btn-primary disabled"/>
    <asp:Button ID="Button9" runat="server" Text=".active" class="btn
    btn-primary active"/>
</p>
<p>
    <asp:Button ID="Button7" runat="server" Text=".btn-lg" class="btn
    btn-primary btn-lg"/>
    <asp:Button ID="Button8" runat="server" Text=".btn-xs" class="btn
    btn-primary btn-xs"/>
</p>
<p>
    <asp:Button ID="Button5" runat="server" Text=".btn-block" class="btn
    btn-primary btn-block"/>
</form>
</body>
</html>

```

OUTPUT :

Practical - 10 (b) : Create a web application to demonstrate use of various Ajax controls.

1. Add the Script Manager Control on your Page.
2. Add Update Panel control, Under the Update Panel control as the Button and label as shown in the screen shot.
3. Add one more button and label outside the update panel control.



Default.aspx

```
public partial class Default3 : System.Web.UI.Page
```

```
{
```

```
    protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

```
        string time = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
```

```
        Label1.Text = "Showing time from panel" + time;
```

```
        Label2.Text = "Showing time from outside" + time;
```

```
}
```

```
    protected void Button2_Click(object sender, EventArgs e)
```

```
{
```

```
        string time = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
```

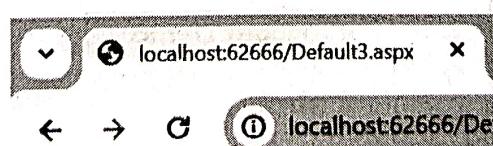
```
        Label1.Text = "Showing time from panel" + time;
```

```
        Label2.Text = "Showing time from outside" + time;
```

```
}
```

```
}
```

OUTPUT :



```
Partial Postback Button  
Showing time from panel16:46:40
```

```
Outside the Update Panel
```

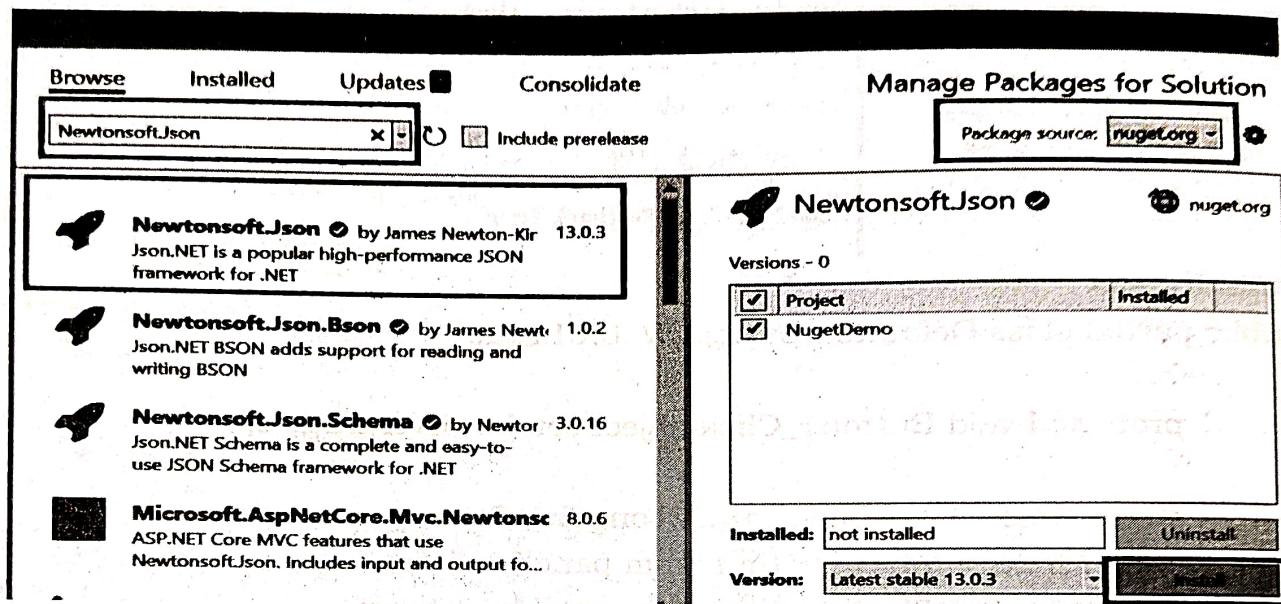
```
Total Postback
```

```
Showing time from outside16:46:38
```

Practical - 10 (c) : Create a web application to demonstrate Installation and use of NuGet package.

To install the Newtonsoft.Json package in Visual Studio using the NuGet Package Manager, follow these steps :

1. Navigate to Project > Tools > Nuget Package Manager > Manage NuGet Packages.
2. In the NuGet Package Manager page, designate nuget.org as the Package source.
3. On the Browse tab, search for Newtonsoft.Json, select it from the list, then click Install.

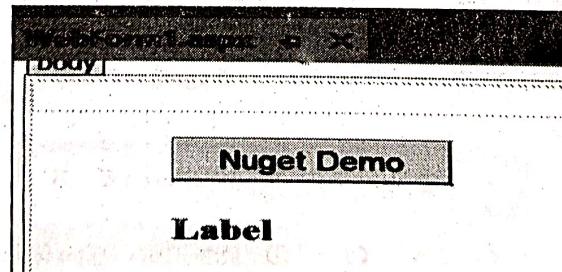


4. If prompted to verify the installation, click OK.

Use the Newtonsoft.Json API in the Web Application

To utilize the JsonConvert.SerializeObject method from the Newtonsoft.Json package in the project, follow these steps:

1. In Solution Explorer, right-click on the Project Name, then select Add, New item, and choose Web Form from the Item window.
2. On the web form in design view, add a button and a label.



3. Open the code-behind file for the WebForm (WebForm1.aspx.cs), or double-click on the button created in the WebForm, and use the following code :

WebForm1Aspx.cs

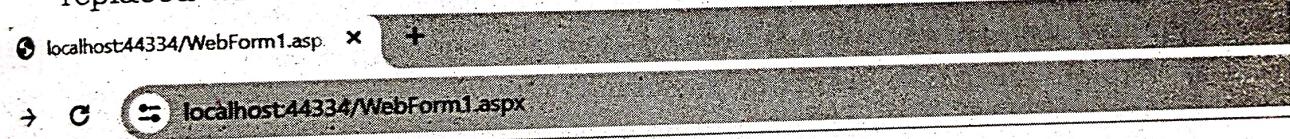
```
using System;
using Newtonsoft.Json;
namespace NugetDemo
{
    public partial class WebForm1 : System.Web.UI.Page
```

```

{
    protected void Button1_Click(object sender, EventArgs e)
    {
        Account account = new Account
        {
            Name = "Tushar Sambare",
            Email = "tushars@microsoft.com",
            DOB = new DateTime(1982, 12, 13, 0, 0, 0, DateTimeKind.Local),
        };
        string json =
            JsonConvert.SerializeObject(account,
Newtonsoft.Json.Formatting.Indented);
        Label1.Text = json;
    }
}

```

1. To launch and run the application, press F5 or navigate to Debug > Start Debugging.
2. Click the button in the output window to observe the Label object's contents replaced with JSON text.



Nugget Demo

```
{"Name": "Tushar Sambare", "Email": "tushars@microsoft.com", "DOB": "1982-12-13T00:00:00+05:30"}
```

SAMPLE QUESTION PAPER**ADVANCED WEB PROGRAMMING****T.Y.B.Sc. (I.T.) (Semester - V)****All questions are compulsory :****1. Attempt any three of the following :****[15]**

- What is .NET Framework? What is in the .NET Framework?
- What is the CTS, and how does it relate to the CLS?
- Write a note on Type Conversion.
- Explain array memory representation with an example.
- How to write properties in class? Give example.
- What is constructor overloading? Give example.

2. Attempt any three of the following :**[15]**

- Explain any five common properties of web server controls.
- Explain Anatomy of a Webform.
- Write a note on Page class.
- What is use of autopostback and runat properties.
- Explain TreeView and Menu site navigation controls.
- Give the Life Cycle of a Web Page.

3. Attempt any three of the following :**[15]**

- Explain exception handling mechanism in C#.
- What is custom exception and how to raise it in C#?
- How to Manage State in ASP.Net?
- Explain cross page posting with an example.
- Explain Cookies with example.
- Give the uses of Master Pages. How Master Pages Work?

4. Attempt any three of the following :**[15]**

- What is Database? What are its uses?
- Give the example of filling ListBox with SQL Database table entries.
- Write short note on Data Source Controls
- How to set Parameter Values in Code?
- Write short note on selecting a GridView Row.
- What is connected and disconnected database in ADO .NET?

5. Attempt any three of the following :**[15]**

- What is NuGet Package? What is it used for?
- What is bootstrap? What are its key features?
- Explain Windows Authentication.
- How to Test and Deploy Security Settings in ASP .NET?
- Explain AJAX with its advantages and Disadvantages.
- Write short note on Accordion control with appropriate properties.