

**SMT. K. L. TIWARI DEGREE COLLEGE
OF COMMERCE AND SCIENCE**

Department of Information Technology

Practical Journal

Name:

Class:

PRN NO:

Subject:

Course Code:

Certificate

Name:

Class:

Roll No:

Exam No:

Institution _____

This is certified to be the bonafide work of the student in the
_____ *Laboratory during the academic*
year 20 / 20

No. of practicals certified _____ out of _____ in the
subject of _____

.....
Teacher In-charge

.....
Examiner's Signature

.....
Principal

Date:

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

TABLE OF CONTENTS

Sr. No.	Description	Date	Signature
	PRACTICAL NO. 1		
1	A) Create an application that obtains 4 inputs and shows their product		
2	B) Create an application to demonstrate string operations		
3	C) Create an application to handle student data using an array of structures		
4	D1) Create an application to test for prime numbers		
5	D2) demonstrate a foreach loop with arrays		
	PRACTICAL NO. 2		
6	A1) Create applications for: Finding factorial value		
7	A2) Create applications for: Money conversion (Dollar to Rupee, Euro to Rupee, etc.)		
9	B1) Function Overloading (swap integers and floats)		
9	B2) Inheritance (Single, Multilevel Inheritance)		
10	B3) Constructor Overloading		
11	B4) Interfaces		
12	C1) Create simple application for using Delegates and events.		
13	C2) Create simple application to demonstrate Exception handling.		
	PRACTICAL NO. 3		
14	A) Create a web page with server controls and demonstrate AutoPostBack		
15	B1) Use Calendar control to display selected day and		
16	B2) Calculate the difference between two dates in a Calendar		
17	C) Use TreeView control to display a hierarchical structure		

Sr. No.	Description	Date	Signature
	PRACTICAL NO. 4		
18	A) Create a registration form with validation controls		
19	B) Create a Web Form demonstrating AdRotator control		
20	C) Demonstrate User Controls in a Web Form		
	PRACTICAL NO. 5		
21	A) Create a web application with Master Page, Styles, and Themes		
22	B) Demonstrate state management using Cookies		
	PRACTICAL NO. 6		
23	A) Bind data in a multiline textbox using a SQL query		
24	B) Display records from a database		
	PRACTICAL NO. 7		
25	A) Bind data to a DropDownList control		
26	B) Insert and delete records from a database using ExecuteNonQuery		

Practical No. - 1

A) Aim: Create an application that obtains 4 inputs and show product of the same.

Program:

```
using System;
namespace ConsoleApplication4
{
    class Program
    {
        public static void Main(string[] args)
        {
            int a, b, c, d, prod;
            Console.Write("Enter number 1: "); a =
                Int32.Parse(Console.ReadLine());
            Console.Write("Enter number 2: "); b =
                Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter number 3: "); c =
                Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter number 4: "); d =
                Convert.ToInt32(Console.ReadLine());
            prod = a * b * c * d;
            Console.WriteLine("Product of {0},{1},{2},{3} is {4}",a,b,c,d,prod);
            Console.ReadLine();
        }
    }
}
```

Output:

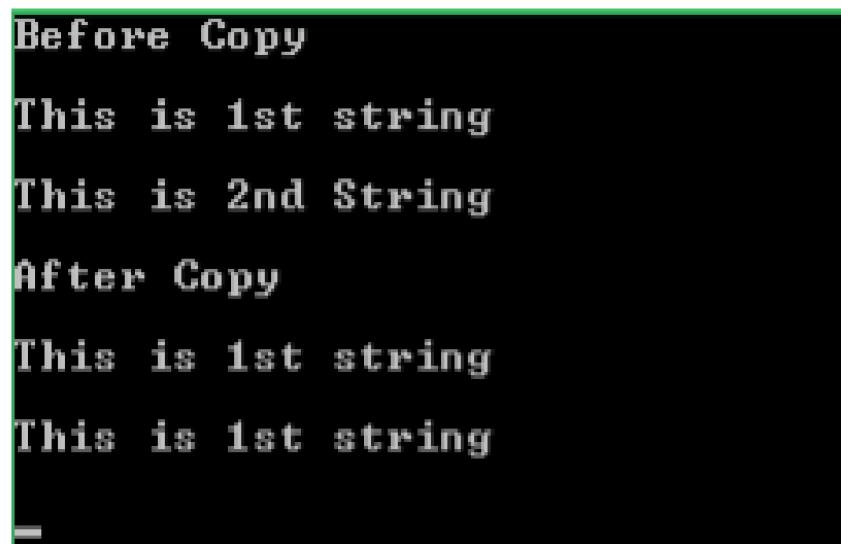
```
Enter number 1: 3
Enter number 2: 4
Enter number 3: 5
Enter number 4: 6
Product of 3,4,5,6 is 360
```

B) Aim: Create an application to demonstrate string operations.

Program:

```
using System; namespace  
ConsoleApplication4 {  
class Program {  
public static void Main(string[] args) {  
string str1 = "This is 1st string ";  
string str2 = "This is 2nd String";  
Console.WriteLine("Before Copy\n");  
Console.WriteLine(str1 + "\n");  
Console.WriteLine(str2 + "\n");  
str2 = string.Copy(str1); Console.WriteLine("After  
Copy\n");  
Console.WriteLine(str1 + "\n");  
Console.WriteLine(str2 + "\n");  
Console.ReadLine();  
}  
}  
}
```

Output:



```
Before Copy  
This is 1st string  
This is 2nd String  
After Copy  
This is 1st string  
This is 1st string  
=
```

C) Aim: Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.

Program:

Student Id:

Student Name:

Course Name:

Date of Birth:

The application should also display the information of all the students once the data is entered. Implement this using an Array of Structures.

using System;

```
namespace ConsoleApplication4 {
    class Program {
        struct Student {
            public string studentid, name, coursename;
            public int day, month, year;
        }
        static void Main(string[] args) {
            Student[] std = new Student[5];
            int i;
            for (i = 0; i < 2; i++)
            {
                Console.Write("Enter Student Id:"); std[i].studentid = Console.ReadLine();
                Console.Write("Enter Student name: ");
                std[i].name = Console.ReadLine(); Console.Write("Enter Course name: ");
                std[i].coursename = Console.ReadLine(); Console.Write("Enter date of birth\n Enter day:");
                std[i].day = Convert.ToInt32(Console.ReadLine());
                Console.Write("Enter month:");
                std[i].month = Convert.ToInt32(Console.ReadLine()); Console.Write("Enter year:");
            }
        }
    }
}
```

```

        std[i].year = Convert.ToInt32(Console.ReadLine());
    }

    Console.WriteLine("\n\nStudent's List\n");
    for (i = 0; i < 2; i++){
        Console.WriteLine("\nStudent ID: " + std[i].studentid);
        Console.WriteLine("\nStudent name: " + std[i].name);
        Console.WriteLine("\nCourse name: " + std[i].coursename);
        Console.WriteLine("\nDate of birth(dd/mm/yy): " + std[i].day + "/" + std[i].month + "/" +
        std[i].year);
    }

    Console.ReadLine();
}
}

```

Output:

```

Enter Student Id:1
Enter Student name : Amit
Enter Course name : Science
Enter date of birth
  Enter day:06
  Enter month:07
  Enter year:1994
Enter Student Id:2
Enter Student name : Dinesh
Enter Course name : Commerce
Enter date of birth
  Enter day:19
  Enter month:12
  Enter year:1993

Student's List

Student ID : 1
Student name : Amit
Course name : Science
Date of birth(dd/mm/yy) : 6/7/1994
Student ID : 2
Student name : Dinesh
Course name : Commerce
Date of birth(dd/mm/yy) : 19/12/1993

```

D) Aim: Create an application to demonstrate following operations.

1. Aim: Create an application to test for prime numbers.

Program:

```
using System;
namespace ConsoleApplication4
{
    class Program
    {
        static void Main(string[] args)
        {
            int m1, i, m2 = 0, m3 = 0;
            Console.Write("Enter the Number to check ");
            m1 = int.Parse(Console.ReadLine());
            m2 = m1 / 2;
            for (i = 2; i <= m2; i++)
            {
                if (m1 % i == 0)
                {
                    Console.Write("Number is not Prime.");
                    m3 = 1;
                    break;
                }
            }
            if (m3 == 0)
                Console.Write("Number is Prime.");
            Console.ReadLine();
        }
    }
}
```

Output:

```
Enter the Number to check 19
Number is Prime.
```

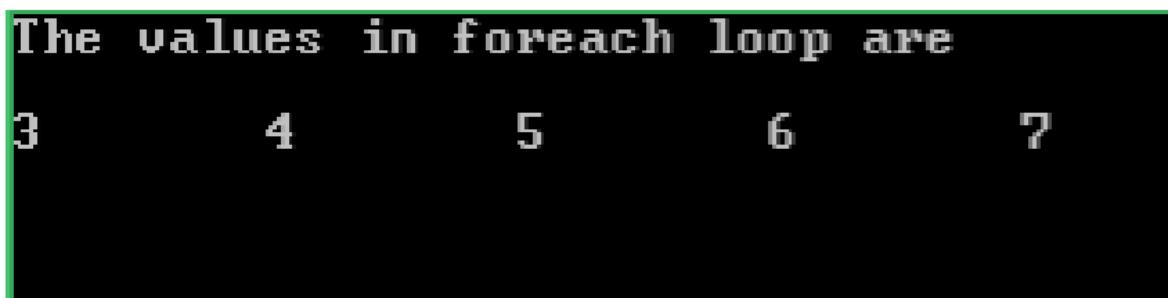
```
Enter the Number to check 4
Number is not Prime..
```

2. Aim: Create an application to demonstrate the use of foreach loop with arrays.

Program:

```
using System;
namespace ConsoleApplication4
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a = { 3, 4, 5, 6, 7 };
            Console.WriteLine("The values in foreach loop are \n");
            foreach (int i in a)
            {
                Console.Write(i+"\t");
            }
            Console.ReadLine();
        }
    }
}
```

Output:



```
The values in foreach loop are
3
4
5
6
7
```

Practical No. - 2

A) Aim: Create simple application to perform following operations

1. Aim: Create simple application for finding factorial value.

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication4
{
    class Program
    {
        public static void Main(string[] args)
        {
            int i, fact = 1, num1;
            Console.WriteLine("Enter the number");
            num1 = int.Parse(Console.ReadLine());
            for (i = 1; i <= num1; i++)
            {
                fact = fact * i;
            }
            Console.WriteLine("Factorial of number " + num1 + " is " + fact);
            Console.ReadLine();
        }
    }
}
```

Output:

```
Enter a Number:
5
The factorial of 5 is 120
```

2. Aim: Create simple application for Money Conversion.

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication4
{
    class Program
    {
        public static void Main(string[] args)
        {
            int choice;
            Console.WriteLine("Enter your choice:\n 1-Dollar to Rupee \n 2-Euro to
Rupee \n 3-Malaysian Ringgit to Rupee ");
            choice = int.Parse(Console.ReadLine());
            switch (choice)
            {
                case 1:
                    Double dollar, rupee, val1;
                    Console.WriteLine("Enter the dollar amount");
                    dollar = double.Parse(Console.ReadLine());
                    Console.WriteLine("Enter the dollar value");
                    val1 = double.Parse(Console.ReadLine());
                    rupee = dollar * val1;
                    Console.WriteLine("{0} Dollar Equals {1} Rupees", dollar, rupee);
                    break;
                case 2:
                    Double euro, rupee2, val2;
                    Console.WriteLine("Enter the Euro amount");
                    euro = double.Parse(Console.ReadLine());
```

```
Console.WriteLine("Enter the Euro value");
val2 = double.Parse(Console.ReadLine());
rupee2 = euro * val2;
Console.WriteLine("{0} Euro Equals {1} Rupees", euro, rupee2);
break;
case 3:
Double ringit, rupee3, val3;
Console.WriteLine("Enter the ringgit amount");
ringit = double.Parse(Console.ReadLine());
Console.WriteLine("Enter the ringgit value");
val3 = double.Parse(Console.ReadLine());
rupee3 = ringit * val3;
Console.WriteLine("{0} Malaysian Ringgit Equals {1} Rupees", ringit, rupee3);
break;
default:
Console.WriteLine("Invalid Choice");
break;
}
Console.ReadLine();
}
}
}
```

Output:

```
Enter your choice:  
1-Dollar to Rupee  
2-Euro to Rupee  
3-Malaysian Ringgit to Rupee  
1  
Enter the dollar amount  
2  
Enter the dollar value  
74  
2 Dollar Equals 148 Rupees  
Enter your choice:  
1-Dollar to Rupee  
2-Euro to Rupee  
3-Malaysian Ringgit to Rupee  
2  
Enter the Euro amount  
2  
Enter the Euro value  
88.47  
2 Euro Equals 176.94 Rupees
```

B) Aim: Create simple application to demonstrate use of following concepts:

1. Aim: Create simple application to demonstrate Function Overloading.

Program:

```
using System;
namespace ConsoleApplication4
{
    class Program
    {
        public void swap(ref int n, ref int m)
        {
            int t;
            t = n;
            n = m;
            m = t;
        }

        public void swap(ref float f1, ref float f2)
        {
            float f;
            f = f1;
            f1 = f2;
            f2 = f;
        }

        public static void Main(string[] args)
        {
            Program p1 = new Program();
            int n = 10, m = 20;
            Console.WriteLine("Before Swap " + "\tN=" + n + "\tM=" + m);
            p1.swap(ref n, ref m);
            Console.WriteLine("After Swap " + "\tN=" + n + "\tM=" + m);
            float f1 = 10.5f, f2 = 20.6f;
            Console.WriteLine("Before Swap " + "\tN=" + f1 + "\tM=" + f2);
```

```
p1.swap(ref f1, ref f2);  
Console.WriteLine("After Swap " + "\tN=" + f1 + "\tM=" + f2);  
Console.ReadLine();  
}  
}  
}
```

Output:

```
Before Swap      N=10      M=20  
After Swap      N=20      M=10  
Before Swap      N=10.5    M=20.6  
After Swap      N=20.6    M=10.5  
-
```

2. Aim: Create simple application to demonstrate Inheritance (all types).

a. Program for Single Inheritance:

```
using System;
namespace ConsoleApplication4
{
    public class base1
    {
        protected int a = 50;
        protected int b = 60;
    }
    public class base2: base1
    {
        public void show()
        {
            int c;
            c = a + b;
            Console.WriteLine("Example of Single inheritance with protected mode " + "\n\nSum is " +
c);
        }
    }
    class Program
    {
        public static void Main(string[] args)
        {
            base2 c2 = new base2();
            c2.show();
            Console.ReadLine();
        }
    }
}
```

Output:

```
Example of Single inheritance with protected mode
Sum is 110
```

b. Program for Multilevel Inheritance:

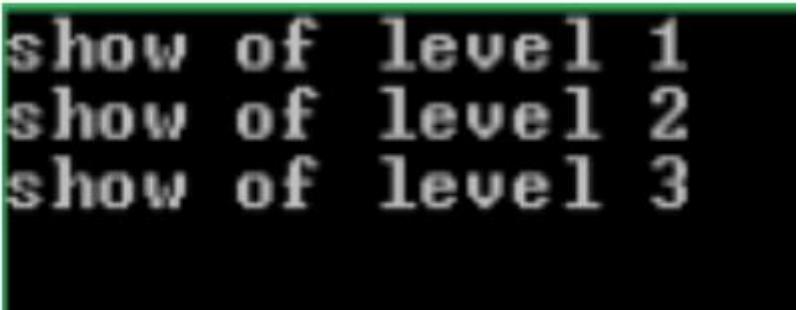
```
using System;
namespace ConsoleApplication4
{
    class test
    {
        public void show()
        {
            Console.WriteLine("show of level 1");
        }
    }

    class testme: test
    {
        public void showme()
        {
            base.show();
            Console.WriteLine("show of level 2");
        }
    }

    class Testus: testme
    {
        public void showus()
        {
            base.showme();
        }
    }
}
```

```
Console.WriteLine("show of level 3");
} }
public class program
{
public static void Main(string[] args)
{
Testus t1 = new Testus();
t1.showus();
Console.ReadLine();
}
}
}
```

Output:



```
show of level 1
show of level 2
show of level 3
```

3. Aim: Create simple application to demonstrate Constructor overloading.

Program:

```
using System;
namespace ConsoleApplication4
{
    class Add {
        int x, y;
        double f;
        string s;
        public Add(int a, double b)
        {
            x = a;
            f = b;
        }
        public Add(int a, string b)
        {
            y = a;
            s = b;
        }
        public void show()
        {
            Console.WriteLine("First Constructor (int+float): {0} ,(x+f));"
        }
        public void show1()
        {
            Console.WriteLine("Second Constructor (int+string): {0} ,(y + s));"
        }
    }
    class Program
    {
        public static void Main(string[] args)
        {
            Add g = new Add(10, 15.5);
        }
    }
}
```

```
g.show();  
Add m = new Add(10, " Name");  
m.show1();  
Console.ReadLine();  
}  
}  
}
```

Output:

```
First Constructor (int+float): 25.5
Second Constructor (int+string): 10 Name
```

4.Aim: Create simple application to demonstrate Interfaces.

Program:

```
using System;
namespace ConsoleApplication4{
public interface Class1
{
void draw();
}
public class Class1A: Class1
{
public void draw()
{
Console.WriteLine("Section A of Class 1");
}
}
public class Class1B: Class1
{
public void draw()
{
Console.WriteLine("Section B of Class 1 Interface");
}
}
public class TestInterface{
public static void Main() {
Class1 d;
d = new Class1A();
d.draw();
d = new Class1B();
d.draw();
Console.ReadLine();
}
}
}
```

Output:

```
Section A of Class 1
Section B of Class 1 Interface
```

```
-
```

C) Aim: Create simple application to demonstrate use of following concepts:

1. Aim: Create simple application for using Delegates and events.

Program:

```
using System;
namespace ConsoleApplication4
{
    delegate int NumberChanger(int n);
    class Program
    {
        static int num = 10;
        public static int AddNum(int p)
        {
            num += p;
            return num;
        }
        public static int MultNum(int q)
        {
            num *= q;
            return num;
        }
        public static int getNum()
        {
            return num;
        }
        public static void Main(string[] args)
        {
            NumberChanger nc1 = new NumberChanger(AddNum);
            NumberChanger nc2 = new NumberChanger(MultNum);
            //calling the methods using the delegate objects
            nc1(25);
            Console.WriteLine("Value of ADD Num: {0}", getNum());
```

```
nc2(5);

Console.WriteLine("Value of Product Num: {0}", getNum());

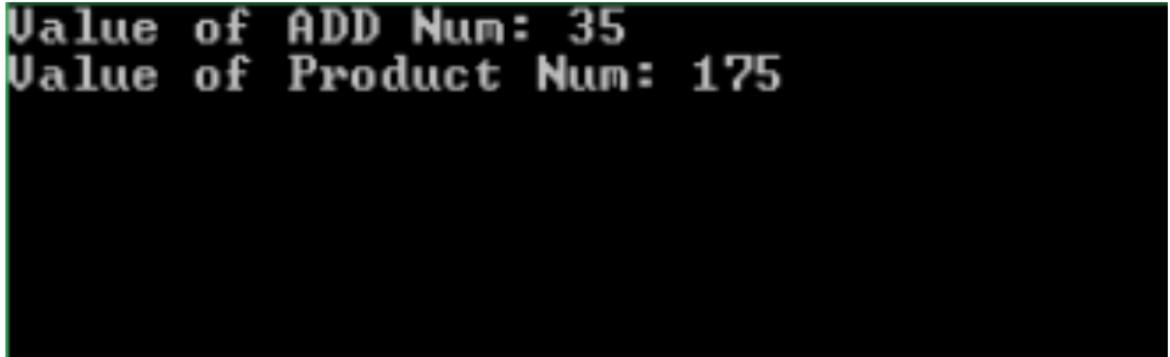
Console.ReadKey(); //Console.ReadLine();

}

}

}
```

Output:



```
Value of ADD Num: 35
Value of Product Num: 175
```

2. Aim: Create simple application to demonstrate Exception handling.

Program:

```
using System;
namespace ExceptionHandlingExample
{
    class NotEvenException: Exception
    {
        public NotEvenException(String msg)
        : base(msg) { }
    }
    class Program
    {
        public static void Main(string[] args)
        {
            int num;
            try
            {
                Console.WriteLine("Enter a Number: ");
                num = int.Parse(Console.ReadLine());
                if ((num % 2) != 0)
                {
                    throw new NotEvenException("Not an Even number");
                }
                else
                {
                    Console.WriteLine("It is an even Number");
                    Console.ReadLine();
                }
            }
            catch (NotEvenException e) { Console.WriteLine(e.Message); }
            Console.ReadLine();
        }
    }
}
```

```
}}}
```

Output:

```
Enter a Number:  
5  
Not an Even number
```

```
Enter a Number:  
6  
It is an even Number
```

Practical No. – 3

A) Aim: Create a simple web page with various sever controls to demonstrate settings and use of their properties. (Example: AutoPostBack)

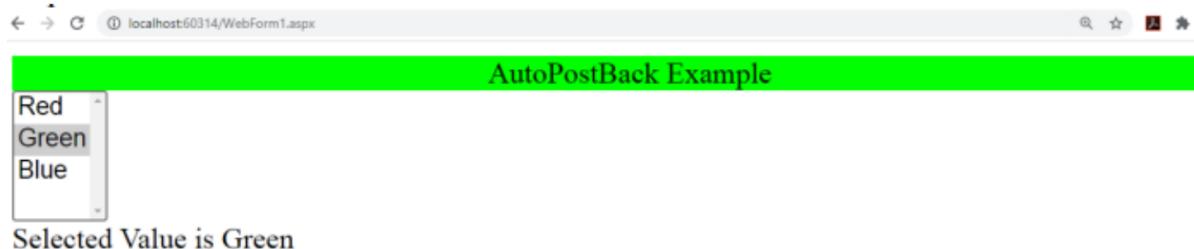
Program for Webform1.aspx page:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Pactical3.WebForm1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div align="center" style="background-color: #00FF00">
AutoPostBack Example
</div>
<div>
<asp:ListBox ID="ListBox1" runat="server" AutoPostBack="True"
onselectedindexchanged="ListBox1_SelectedIndexChanged">
<asp:ListItem>Red</asp:ListItem>
<asp:ListItem>Green</asp:ListItem>
<asp:ListItem>Blue</asp:ListItem>
</asp:ListBox>
<br />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

Program for Webform1.aspx.cs page:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Pactical3
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            Label1.Text = "Selected Value is " + ListBox1.Text;
        }
    }
}
```

Output:



B) Aim: Demonstrate the use of Calendar control to perform following operations:

1. Aim: Create program for displaying selected day in a calendar control using style

Program for Webform1.aspx page:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Pactical3.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

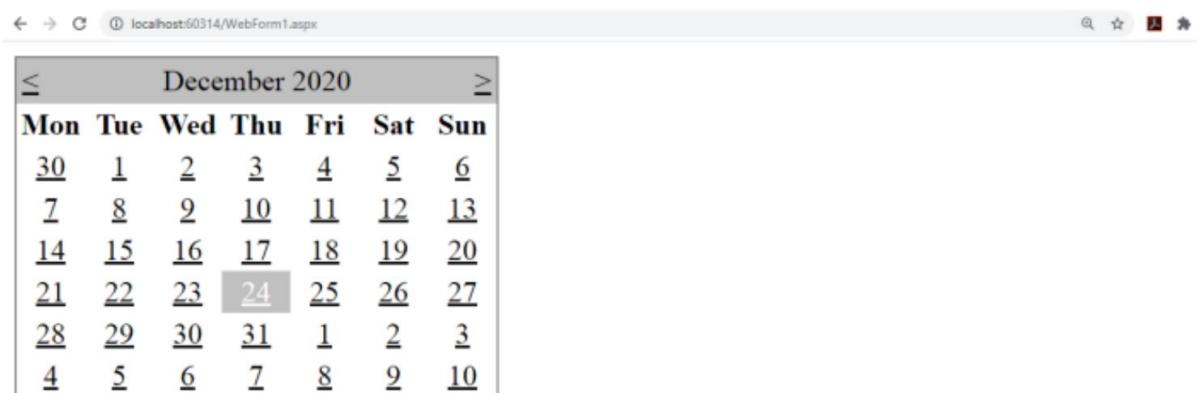
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Calendar ID="Calendar1" runat="server"
onselectionchanged="Calendar1_SelectionChanged"></asp:Calendar>
<br />
<asp:Label ID="ShowDate" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

Program for Webform1.aspx.cs page:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Pactical3
```

```
{  
public partial class WebForm1: System.Web.UI.Page  
{  
protected void Page_Load(object sender, EventArgs e){  
}  
protected void Calendar1_SelectionChanged(object sender, EventArgs e)  
{  
ShowDate.Text = " Selected Date is " + Calendar1.SelectedDate;  
}}}
```

Output:



localhost:60314/WebForm1.aspx

December 2020						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
<u>30</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>	24	<u>25</u>	<u>26</u>	<u>27</u>
<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>

Selected Date is 24-12-2020 00:00:00

2. Aim: Create program to display difference between two calendar dates

Program for Webform1.aspx page:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication9.WebForm1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Calendar ID="Calendar2" runat="server"></asp:Calendar> &nbsp;<asp:Calendar
ID="Calendar3" runat="server"></asp:Calendar>
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
<br />
<br />
</div>
</form>
</body>
</html>
```

Program for Webform1.aspx.cs page:

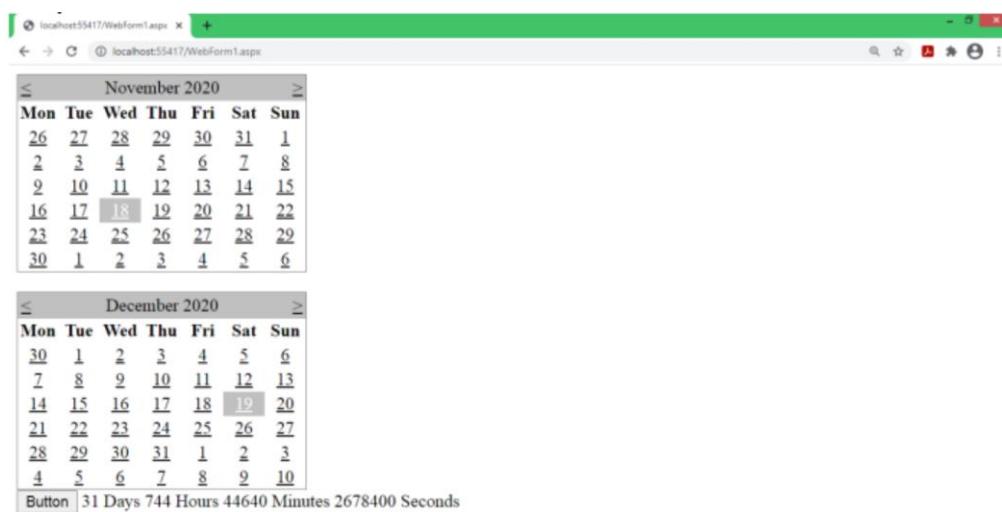
```
using System;
using System.Collections.Generic;
using System.Linq; using
System.Web; using
System.Web.UI;
using System.Web.UI.WebControls;
```

```

using System.Xml; using
System.IO; using System.Xml.Linq;
namespace WebApplication9
{
public partial class WebForm1: System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
}
protected void Button1_Click(object sender, EventArgs e)
{
DateTime dt1 = Calendar2.SelectedDate;
DateTime dt2 = Calendar3.SelectedDate;
TimeSpan ts = dt2 - dt1;
Label1.Text = ts.TotalDays.ToString() + " Days " + "" + ts.TotalHours.ToString() + " Hours " +
"" + ts.TotalMinutes.ToString() + " Minutes " + "" + ts.TotalSeconds.ToString() + " Seconds ";
}
}
}

```

Output:



C) Aim: Demonstrate the use of Tree view control perform operations.

Program:

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeBehind="Home.aspx.cs" Inherits="WebApplication10.WebForm1" %>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<title></title>  
</head>  
<body>  
<form id="form1" runat="server">  
<div>  
<h1>Welcome to Home Page</h1>  
<asp:TreeView ID="TreeView1" runat="server" ImageSet="Inbox"  
BackColor="#00CC99" BorderColor="#000099" Font-Bold="True">  
<HoverNodeStyle Font-Underline="True" BackColor="Red" />  
<LeafNodeStyle BackColor="Yellow" />  
<Nodes>  
<asp:TreeNode NavigateUrl="~/Home.aspx" Target="_blank"  
Text="Home"  
Value="Home">  
<asp:TreeNode NavigateUrl="~/Employer.aspx" Target="_blank"  
Text="Employer"  
Value="Employer">  
<asp:TreeNode NavigateUrl="~/Upload_Job.aspx"  
Target="_blank" Text="Upload Job"  
Value="Upload Job"></asp:TreeNode>  
<asp:TreeNode NavigateUrl="~/Edit_Job.aspx" Target="_blank"  
Text="Edit Job"  
Value="Edit Job"></asp:TreeNode>
```

```
<asp:TreeNode NavigateUrl="~/View_Job.aspx" Target="_blank"
Text="View Job"
Value="View Job"></asp:TreeNode>
</asp:TreeNode>
<asp:TreeNode NavigateUrl="~/Employee.aspx" Target="_blank"
Text="Employee"
Value="Employee">
<asp:TreeNode NavigateUrl="~/Upload_Resume.aspx"
Target="_blank"
Text="Upload Resume" Value="Upload
Resume"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/Edit_Resume.aspx"
Target="_blank"
Text="Edit Resume" Value="Edit Resume"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/View_Resume.aspx" Target="_blank"
Text="View_Resume" Value="View_Resume"></asp:TreeNode>
</asp:TreeNode>
<asp:TreeNode NavigateUrl="~/Admin.aspx" Target="_blank"
Text="Admin" Value="Admin">
<asp:TreeNode NavigateUrl="~/Add_User.aspx"
Target="_blank" Text="Add User" Value="Add User"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/Edit_User.aspx" Target="_blank" Text="Edit User"
Value="Edit User"></asp:TreeNode>
<asp:TreeNode NavigateUrl="~/View_User.aspx" Target="_blank" Text="View User"
Value="View User"></asp:TreeNode>
</asp:TreeNode>
</asp:TreeNode>
</Nodes>
<NodeStyle Font-Names="Verdana" Font-Size="8pt" ForeColor="Black"
HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />
<ParentNodeStyle Font-Bold="False" BackColor="#006600" BorderStyle="Dashed" />
```

```

<SelectedNodeStyle Font-Underline="True" HorizontalPadding="0px"
VerticalPadding="0px"

BackColor="Fuchsia" Font-Bold="True" />

</asp:TreeView>

</div>

</form>

</body>

</html>

```

Output:

The image consists of three vertically stacked screenshots of a web browser window. Each screenshot shows a navigation tree on the left and a main content area on the right.

Screenshot 1 (Top): The browser title bar says "localhost:54611/Home.aspx". The main content area displays "Welcome to Home Page". On the left, a navigation tree is shown with the following structure:

- Home
 - Employer
 - Upload Job
 - Edit Job
 - View Job
 - Employee
 - Upload Resume
 - Edit Resume
 - View_Resume
 - Admin
 - Add User
 - Edit User
 - View User

Screenshot 2 (Middle): The browser title bar says "localhost:54611/Home.aspx" and "localhost:54611/upload_job.aspx". The main content area displays "Welcome to Home Page". The "Upload Job" node in the navigation tree is highlighted with a red background.

Screenshot 3 (Bottom): The browser title bar says "localhost:54611/Home.aspx" and "localhost:54611/Edit_Resume.aspx". The main content area displays "Edit_resume". The "Edit Resume" node in the navigation tree is highlighted with a red background.

Practical No. - 4

A) Aim: Create a registration form to demonstrate use of various Validation controls.

Program:

Home.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
```

CodeBehind="Home.aspx.cs"

Inherits="WebApplication10.WebForm1" %>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

```
<head runat="server">
```

<title></title>

</head>

<body>

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

<h1>Validation Example</h1>


```
<asp:Label ID="Label1" runat="server" Text="Candidate Name">
```

</asp:Label>

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

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
```

ControlToValidate="cname" ErrorMessage="*">>

```
</asp:RequiredFieldValidator>
```



```
<asp:Label ID="Label2" runat="server" Text="Address">
```

```
</asp:Label>
```



```
ControlToValidate="Age" ErrorMessage="Age must be
>18 and age <30" MaximumValue="30"
MinimumValue="18"
Type="Integer"></asp:RangeValidator>
<br />
 <asp:Label ID="Label5" runat="server"
Text="Contact No."></asp:Label>
 &nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp;
<asp:TextBox ID="contact" runat="server"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator7"
runat="server"
ControlToValidate="contact"
ErrorMessage="*></asp:RequiredFieldValidator>
<asp:RegularExpressionValidator
ID="RegularExpressionValidator2" runat="server"
ControlToValidate="contact" ErrorMessage="Enter 10 digits"
ValidationExpression="\d{10}"></asp:RegularExpressionValidator>
<br />
<asp:Label ID="Label6" runat="server"
Text="Gender"></asp:Label>
 &nbsp;<asp:RadioButtonList ID="Gender" runat="server"
AutoPostBack="True">
<asp:ListItem>Male</asp:ListItem>
<asp:ListItem>Female</asp:ListItem>
<asp:ListItem>Other</asp:ListItem>
<asp:ListItem>Rather not to say</asp:ListItem>
</asp:RadioButtonList>
<asp:RequiredFieldValidator ID="RequiredFieldValidator8"
runat="server"
ControlToValidate="Gender"
ErrorMessage="*></asp:RequiredFieldValidator>
<br />
```

```
<asp:Label ID="Label7" runat="server" Text="ID  
Proof"></asp:Label>  
&nbsp;<asp:CheckBoxList ID="ID_Proof" runat="server">  
<asp:ListItem>Aadhar Card</asp:ListItem>  
<asp:ListItem>Driving License</asp:ListItem>  
<asp:ListItem>Voter Card</asp:ListItem>  
<asp:ListItem>Passport</asp:ListItem>  
</asp:CheckBoxList>  
<asp:Button ID="Button1" runat="server" Text="Submit"  
onclick="Button1_Click" />  
<br />  
<br />  
</div>  
</form>  
</body></html>
```

Output:

localhost:64611/Home.aspx

Validation Example

Candidate Name *

Address *

Email *

Password *

Confirm Password *

Age * Age must be >18 and age <30
Enter 10 digits

Contact No.

Gender

Male

Female

Other

Rather not to say

ID Proof

Aadhar Card

Driving License

Voter Card

Passport

localhost:64611/Home.aspx

Validation Example

Candidate Name

Address

Email

Password

Confirm Password

Age

Contact No.

Gender

Male

Female

Other

Rather not to say

ID Proof

Aadhar Card

Driving License

Voter Card

Passport

localhost:64611/Employee.aspx

Employee Page

B) Aim: Create Web Form to demonstrate use of Adrotator Control.

Add Home.aspx

- Create a new folder named as images
- Add images to the folder
- Add the .xml file to project.
- Insert advertisement info in the xml file named AdRotator.xml
- Select AdRotator Control on the Design Page
- Select xml datasource control on the design page
- Select the data source file named AdRotator.xml

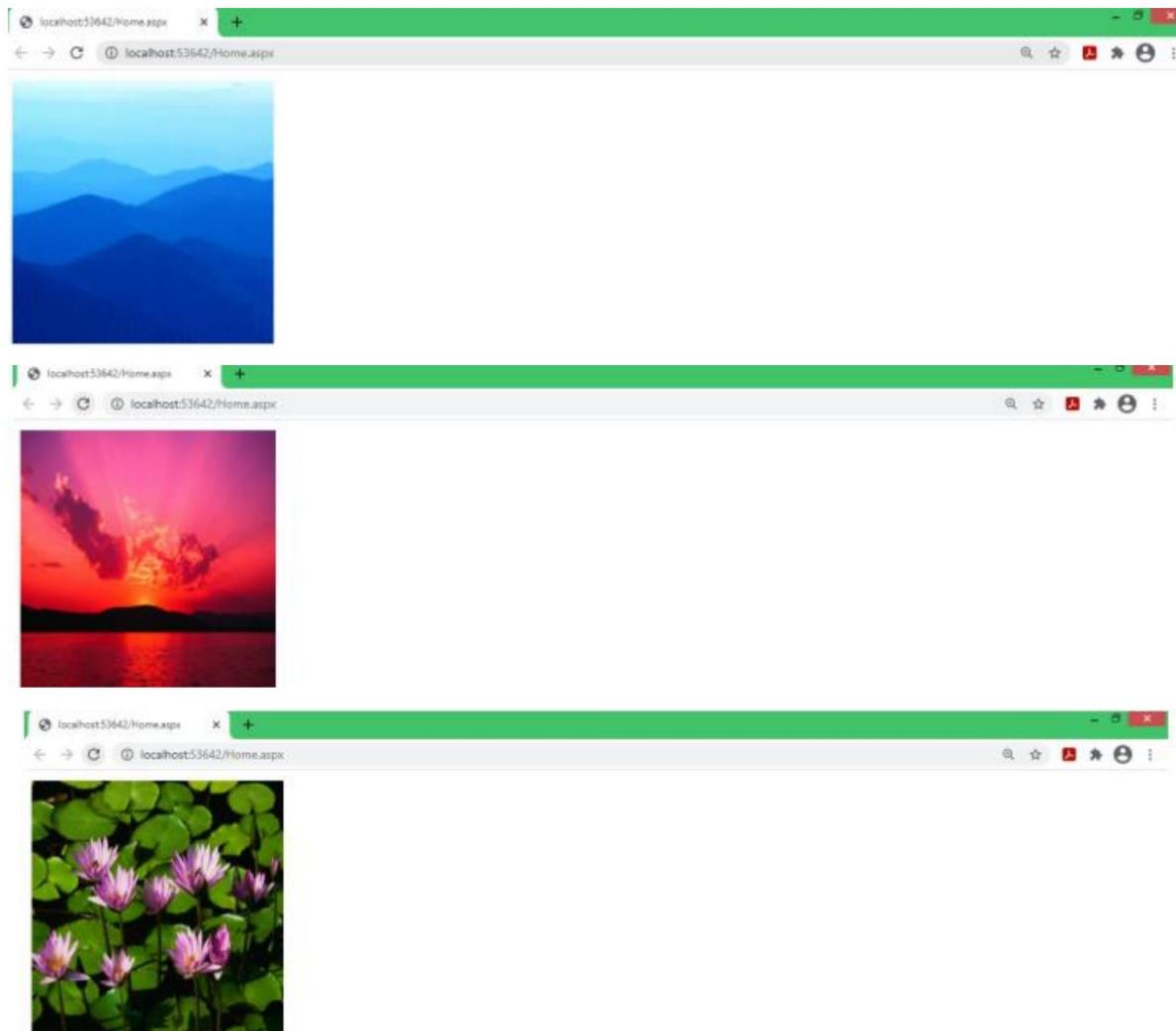
Program for Home.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Home.aspx.cs"
Inherits="AdRotatorExample.Home" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1"
Target="_self" Width="200px" Height="200px" />
<br />
<asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/Adrotator.xml"></asp:XmlDataSource>
</div>
</form>
</body>
</html>
```

Program for AdRotator.xml page:

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
<Ad>
<ImageUrl>~\images\Bluehills.jpg</ImageUrl>
<NavigateUrl>Ad1.aspx</NavigateUrl>
<AlternateText>page1</AlternateText>
<Keyword>B</Keyword>
<Impressions>2</Impressions>
<Caption>This is the caption for Ad#1</Caption>
</Ad>
<Ad>
<ImageUrl>~\images\Sunset.jpg</ImageUrl>
<NavigateUrl>Ad2.aspx</NavigateUrl>
<AlternateText>page2</AlternateText>
<Keyword>S</Keyword>
<Impressions>3</Impressions>
<Caption>This is the caption for Ad#2</Caption>
</Ad>
<Ad>
<ImageUrl>~\images\Waterlilies.jpg</ImageUrl>
<NavigateUrl>Ad3.aspx</NavigateUrl>
<AlternateText>page3</AlternateText>
<Keyword>W</Keyword>
<Impressions>4</Impressions>
<Caption>This is the caption for Ad#2</Caption>
</Ad>
</Advertisements>
```

Output:



C) Aim: Create Web Form to demonstrate use User Controls.

- Add .ascx file
- Insert some controls to the file
- Add .aspx file
- Insert Register Directive to Implementation.aspx which includes tag name and tag prefix.

Program for CustomUserControl.ascx page:

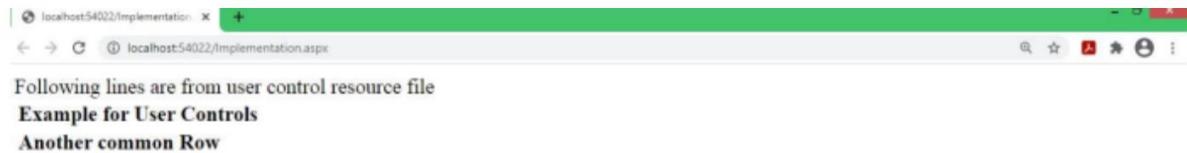
```
<%@ Control Language="C#" AutoEventWireup="true"  
CodeBehind="CustomUserControl.ascx.cs"  
Inherits="UserControls1.CustomButton" %>  
<table>  
<tr>  
<td><b>Example for User Controls</b></td>  
</tr>  
<tr>  
<td><b>Another common Row</b></td>  
</tr>  
</table>
```

Program for Implementation.aspx page:

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeBehind="Implementation.aspx.cs" Inherits="UserControls1.Implementation" %>  
<%@ Register Src="~/CustomUserControl.ascx" TagName="WebControl"  
TagPrefix="TWeb" %>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<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="Following lines are from user control
resource file"></asp:Label>
<TWeb:WebControl ID="Header" runat="server"/>
</div>
</form>
</body>
</html>
```

Output:



Practical No. - 5

A) Aim: Create a web application to demonstrate use of Master Page with applying Styles and Themes for page beautification.

Master Page

- Add a new project as web application.
- Master page by right clicking on solution and select master page with name as site.master
- Add Styles to master page.
- Add content pages in project and link them to site.master
- Add custom content in other content pages.

Program for Site.Master page:

```
<%@ Master Language="C#" AutoEventWireup="true"  
CodeBehind="Site.master.cs" Inherits="WebApplication8.Site" %>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<title></title>  
<asp:ContentPlaceHolder ID="head" runat="server">  
</asp:ContentPlaceHolder>  
<style type="text/css">  
.style1  
{ width: 485px; }  
.style2  
{  
width: 395px;  
}  
</style>  
</head>
```

```
<body>
<form id="form1" runat="server">
<div style="background-color: #00FF00; height: 73px;">
<asp:Menu ID="Menu1" runat="server"
DataSourceID="SiteMapDataSource1"
StaticSubMenuIndent="16px">
<DynamicHoverStyle BackColor="Aqua" BorderColor="Fuchsia" />
<DynamicMenuItemStyle BackColor="#0000CC" />
<DynamicMenuStyle BackColor="Red" />
<DynamicItemTemplate>
<%# Eval("Text") %>
</DynamicItemTemplate>
</asp:Menu>
<asp:SiteMapPath ID="SiteMapPath1" runat="server"
ToolTip="navgation breadcrumbs">
<NodeStyle BorderColor="#660066" />
</asp:SiteMapPath>
</div>
<div>
<table style="width: 100%; height: 139px;">
<tr> <td class="style2">
<asp:TreeView ID="TreeView1" runat="server"
DataSourceID="SiteMapDataSource1"
ImageSet="WindowsHelp" MaxDataBindDepth="2">
<HoverNodeStyle Font-Underline="True" ForeColor="#6666AA" />
<NodeStyle Font-Names="Tahoma" Font-Size="8pt"
ForeColor="Black"
HorizontalPadding="5px" NodeSpacing="0px"
VerticalPadding="1px" />
<ParentNodeStyle Font-Bold="False" />
<SelectedNodeStyle BackColor="#B5B5B5" FontUnderline="False"
HorizontalPadding="0px" VerticalPadding="0px" />
```

```

</asp:TreeView></td>
<td class="style1">
<asp:ContentPlaceHolder ID="ContentPlaceHolder1"
runat="server">
</asp:ContentPlaceHolder></td>
<td>&nbsp;</td> </tr>
</table></div>
<div style="background-color: #FFFF00; height: 123px;">
<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server"/>
</div>
</form>
</body>
</html>

```

Program for Home.aspx page:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Home.aspx.cs"
Inherits="WebApplication8.Home" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" runat="server"
contentplaceholderid="ContentPlaceHolder1">
&nbsp;&nbsp; This is Home page
</asp:Content>

```

Program for About.aspx page:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="About.aspx.cs"
Inherits="WebApplication8.About" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" runat="server"
contentplaceholderid="ContentPlaceHolder1">

```

This is about us page</asp:Content>

Program for Contact.aspx page:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Contact.aspx.cs"
Inherits="WebApplication8.Contact" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server"></asp:Content>
<asp:Content ID="Content2" runat="server"
contentplaceholderid="ContentPlaceHolder1">
This is contact us page
</asp:Content>
```

Program for Web.Sitemap page (For Adding Navigation):

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
<siteMapNode url "~/Home.aspx" title="Home" description="">
<siteMapNode title="Login" url="">
<siteMapNode url "~/About.aspx" title="About" description="">
<siteMapNode url "~/Contact.aspx" title="Contact" target="_blank"/>
</siteMapNode>
</siteMapNode>
</siteMap>
```

Output:



B) Aim: Create a web application to demonstrate use State management 1) Cookie State

Program for Default.aspx page:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="CookieExample.Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<asp:Label ID="Label1" runat="server" Text="Select Brand Preferences"></asp:Label>
<br />
<br />
<asp:CheckBox ID="apple" runat="server" Text="Apple" />
<br />
<asp:CheckBox ID="dell" runat="server" Text="Dell" />
<br />
<asp:CheckBox ID="lenevo" runat="server" Text="Lenevo" />
<br />
<asp:CheckBox ID="acer" runat="server" Text="Acer" />
<br />
<asp:CheckBox ID="sony" runat="server" Text="Sony" />
<br />
<asp:CheckBox ID="wipro" runat="server" Text="Wipro" />
<br />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Submit" />
<p>
<asp:Label ID="Label2" runat="server"></asp:Label>
```

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

Program for Default.aspx.cs page:

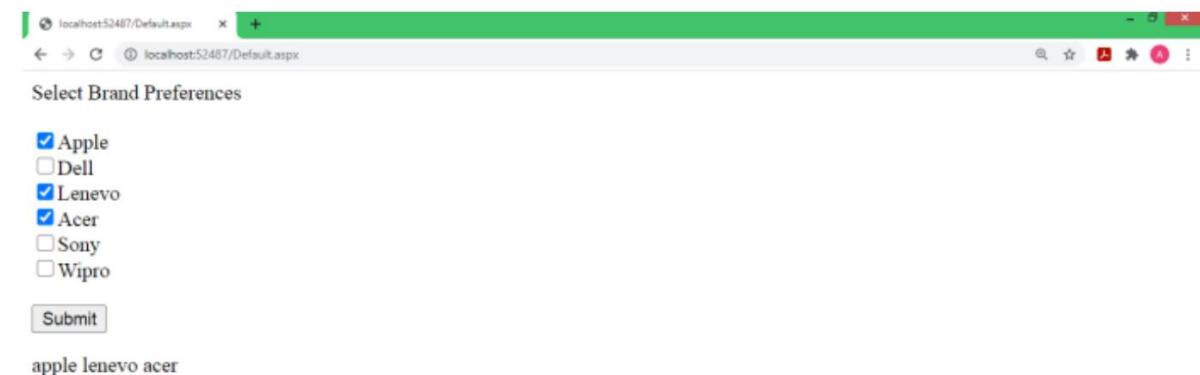
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace CookieExample
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Response.Cookies["computer"].Expires = DateTime.Now.AddDays(-1);
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Label2.Text = "";
            // ----- Adding Cookies -----
            if (apple.Checked)
                Response.Cookies["computer"]["apple"] = "apple";
            if (dell.Checked)
                Response.Cookies["computer"]["dell"] = "dell";
            if (lenevo.Checked)
                Response.Cookies["computer"]["lenevo"] = "lenevo";
            if (acer.Checked)
                Response.Cookies["computer"]["acer"] = "acer";
            if (sony.Checked)
```

```

Response.Cookies["computer"]["sony"] = "sony";
if (wipro.Checked)
    Response.Cookies["computer"]["wipro"] = "wipro";
// ----- Fetching Cookies -----
if (Request.Cookies["computer"].Values.ToString() != null)
{
    if (Request.Cookies["computer"]["apple"] != null)
        Label2.Text += Request.Cookies["computer"]["apple"] + " ";
    if (Request.Cookies["computer"]["dell"] != null)
        Label2.Text += Request.Cookies["computer"]["dell"] + " ";
    if (Request.Cookies["computer"]["lenevo"] != null)
        Label2.Text += Request.Cookies["computer"]["lenevo"] + " ";
    if (Request.Cookies["computer"]["acer"] != null)
        Label2.Text += Request.Cookies["computer"]["acer"] + " ";
    if (Request.Cookies["computer"]["sony"] != null)
        Label2.Text += Request.Cookies["computer"]["sony"] + " ";
    if (Request.Cookies["computer"]["wipro"] != null)
        Label2.Text += Request.Cookies["computer"]["wipro"] + " ";
}
else Label2.Text = "Please select your choice";
Response.Cookies["computer"].Expires = DateTime.Now.AddDays(-1);
} } }

```

Output:



Select Brand Preferences

Apple
 Dell
 Lenevo
 Acer
 Sony
 Wipro

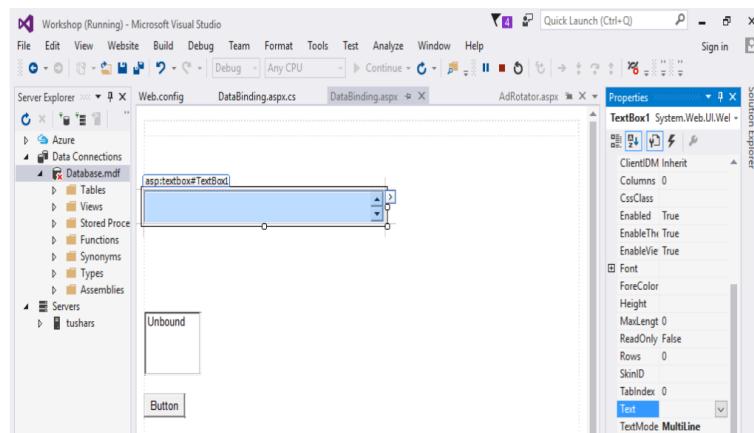
Submit

apple lenevo acer

Practical No. – 6

A) Aim: Create a web application to bind data in a multiline textbox by querying in another textbox.

1. Create a webpage with one Button, one Multiline TextBox and one list box with setting TextMode Property of text box to Multiline as shown below.

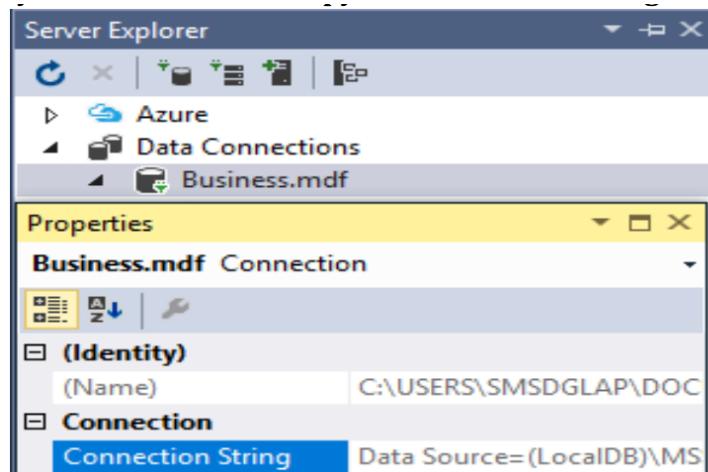


2. Write the Database related code in code behind C# file as given below.

Note: The users must use their own system connection string in place of connection string given in following code.

The connection string is available in Server Explorer (Right click on Database Name and Select

Properties) as displayed below. User can copy this connection string and can use in code.



3. Add this string to configuration file (web.config) as given below.

Web.config

```
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5.2" />
    <httpRuntime targetFramework="4.5.2" />
  </system.web>
  <connectionStrings>
    <add name="connStr" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\Users\tushars\Documents\Visua
1 Studio
2015\WebSites\Workshop\App_Data\Database.mdf;Integrated Security=True" />
  </connectionStrings>
</configuration>
```

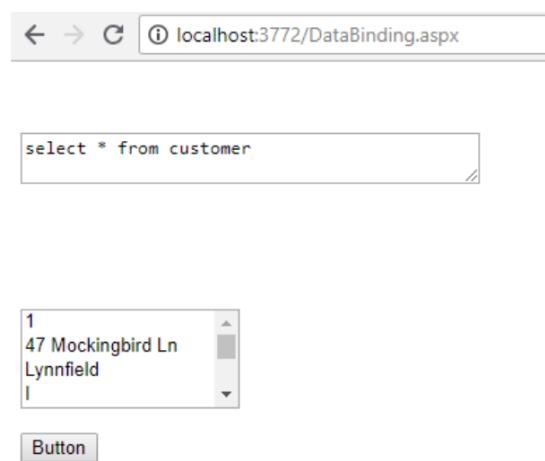
4. Now use the following code C# in Default.aspx.cs (Note : First write following using statements at the top of file

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class DataBinding : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
    string connStr =
      ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
```

```
SqlConnection con = new SqlConnection(connStr);
con.Open();
SqlCommand cmd = new SqlCommand(TextBox1.Text, con);
SqlDataReader reader = cmd.ExecuteReader();
ListBox1.Items.Clear();
while (reader.Read())
{
//To add new blank line in the text area

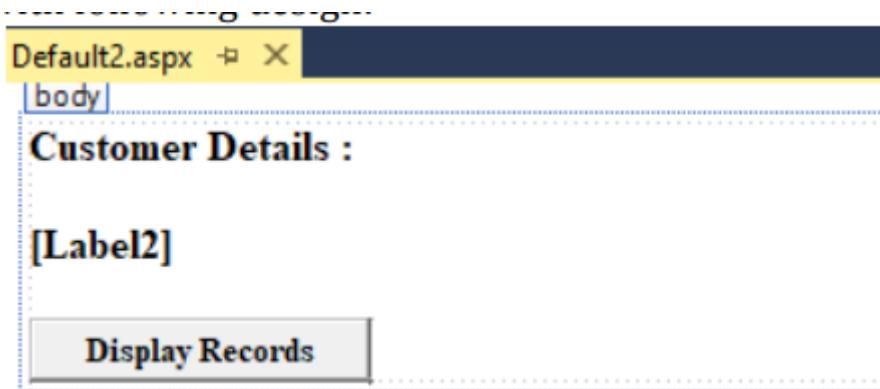
for (int i = 0; i < reader.FieldCount - 1; i++)
{
    ListBox1.Items.Add(reader[i].ToString());
}
}
reader.Close();
con.Close();
}
```

Output:



B) Aim: Create a web application to display records by using database.

Create a web page with following design:



Add the following code on Button click event in C# Code behind file.

```
protected void Button1_Click(object sender, EventArgs e)
{
    string connStr =
        ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
    SqlConnection con = new SqlConnection(connStr);
    SqlCommand cmd = new SqlCommand("Select City, State from Customer", con);
    con.Open();
    SqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        Label1.Text += reader["City"].ToString() + " " + reader["State"].ToString() +
        "<br>";
    }
    reader.Close();
    con.Close();
}
```

Output:



← → C | ⓘ localhost:3772/DisplayRecords.aspx

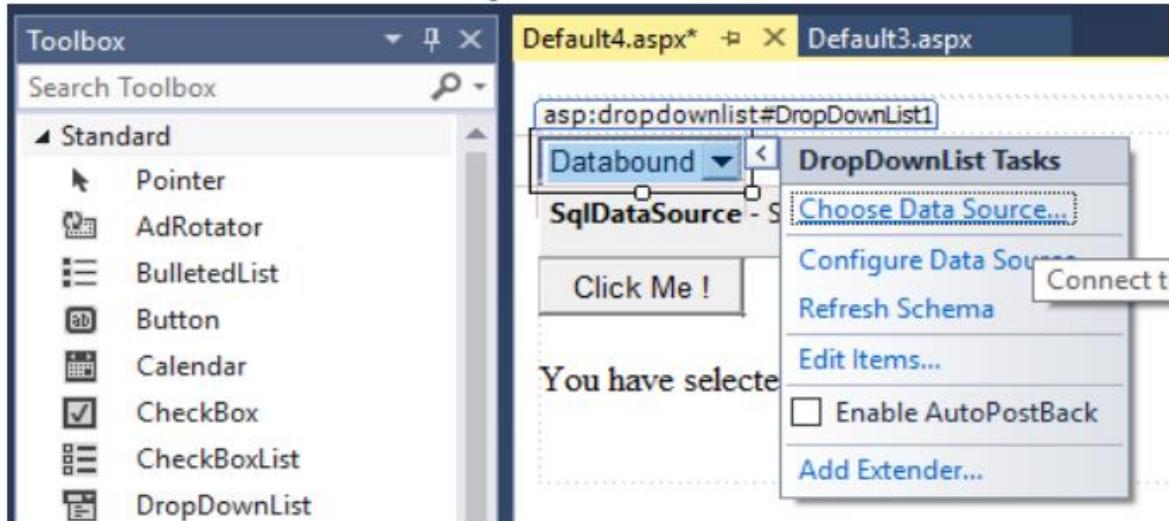
Customer Details

LabelLynnfield MA
Woburn MA
Quincy MA
Waltham MA
Display I
Waltham MA
Wilmington MA
Salem NH
Newton MA
Salem NH
Wilmington MA
Salem NH
Quincy MA

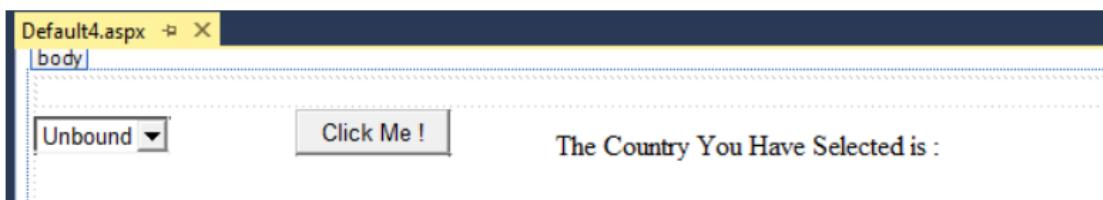
Practical No. – 7

A) Aim: Create a web application to display Databinding using Dropdownlist control.

1. Create a web page with DropDownList control, one Button and one Label control.
2. Use code to bind the data to DropDownList.



Or with Code also we can achieve the same thing.

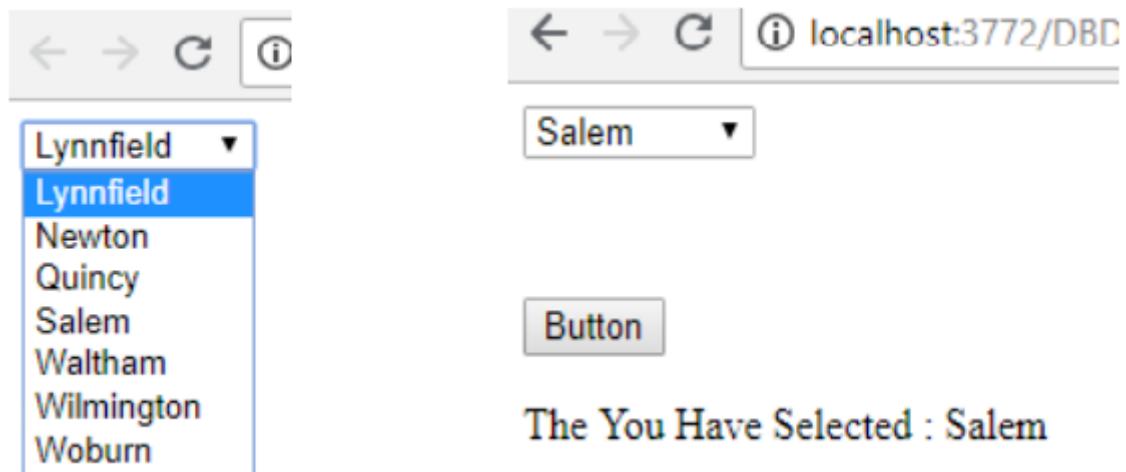


Code of C# Code behind file

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

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
public partial class DBDropDown : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack == false)
        {
            string connStr =
ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
            SqlConnection con = new SqlConnection(connStr);
            SqlCommand cmd = new SqlCommand("Select Distinct City from Customer", con);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            DropDownList1.DataSource = reader;
            DropDownList1.DataTextField = "City";
            DropDownList1.DataBind();
            reader.Close();
            con.Close();
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = "The You Have Selected : " + DropDownList1.SelectedValue;
    }
}
```

Output:



The screenshot shows a web browser window with the URL `localhost:3772/DBD` in the address bar. On the left, there is a dropdown menu with the following options:

- Lynnfield
- Newton
- Quincy
- Salem
- Waltham
- Wilmington
- Woburn

The option "Lynnfield" is currently selected, highlighted with a blue background. On the right, there is another dropdown menu with the value "Salem" selected. Below these dropdowns is a button labeled "Button". To the right of the button, the text "The You Have Selected : Salem" is displayed.

The You Have Selected : Salem

B) Aim: Create a web application for inserting and deleting record 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.

The image shows a Windows Form with a light gray background. It contains five text boxes arranged vertically. The first text box is labeled 'Bank Address'. The second is labeled 'Bank City'. The third is labeled 'Bank Branch Name'. The fourth is labeled 'State'. The fifth is labeled 'ZIP Code'. Below these five text boxes are two buttons: a blue 'Insert' button on the left and a blue 'Delete' button on the right. The entire form is enclosed in a dashed gray border.

Code of C# Code behind file

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class ExecuteNonQuery : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        string connStr = ConfigurationManager.ConnectionStrings["connStr"].ConnectionString;
        SqlConnection con = new SqlConnection(connStr);
```

```
string InsertQuery = "insert into BRANCH values(@ADDRESS, @CITY, @NAME,  
@STATE,  
@ZIP_CODE);  
SqlCommand cmd = new SqlCommand(InsertQuery, con);  
cmd.Parameters.AddWithValue("@ADDRESS", TextBox1.Text);  
cmd.Parameters.AddWithValue("@CITY", TextBox2.Text);  
cmd.Parameters.AddWithValue("@NAME", TextBox3.Text);  
cmd.Parameters.AddWithValue("@STATE", TextBox4.Text);  
cmd.Parameters.AddWithValue("@ZIP_CODE", 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 branch where NAME=@NAME";  
SqlCommand cmd = new SqlCommand(InsertQuery, con);  
cmd.Parameters.AddWithValue("@NAME", TextBox1.Text);  
con.Open();  
cmd.ExecuteNonQuery();  
Label1.Text = "Record Deleted Successfully.";  
con.Close();  
}  
}
```