TOLANI COLLEGE OF COMMERCE

(AUTONOMOUS)

150-151, Sher-E-Punjab Society Guru Gobind Singh Road,
Andheri East, Mumbai, Maharashtra 400 093

Department of B.Sc. (Information Technology)

CERTIFICATE

This is to certify that Mr. / Ms. ____bearing Roll No ______ have completed the practical in the Course of __________________________in accordance with the syllabus of B.Sc. (Information Technology) Programme of Semester______as prescribed by the Tolani College of Commerce (Autonomous) in the academic year 2024-2025.

____

Internal Examiner

____

Programme Coordinator

____

External Examiner

Date: ____________

College Seal

Index

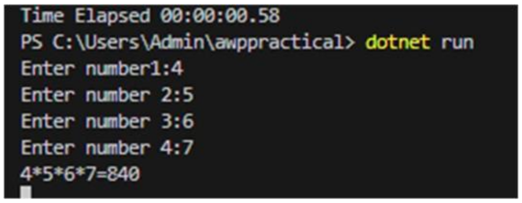
Practical No.: 1

AIM: Working with basic C# and ASP .NET

A) Create an application that obtains four int values from the user and displays the product.

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace  ConsoleApplication1

{

  class Program

  {

     static void Main(string[] args){

      int num1,num2,num3,num4,prod;

      Console.Write("Enter  number1:");

      num1 = Int32.Parse(Console.ReadLine());

      Console.Write("Enter number 2:");

      num2= Convert.ToInt32(Console.ReadLine());

      Console.Write("Enter number 3:");

      num3= Convert.ToInt32(Console.ReadLine());

      Console.Write("Enter number 4:");

      num4 = Convert.ToInt32(Console.ReadLine());

      prod = num1*num2*num3*num4;

      Console.WriteLine(num1+"*"+num2+"*" + num3 +"*"+num4+"=" +prod);

      Console.ReadKey();

    }}}
```

Output:

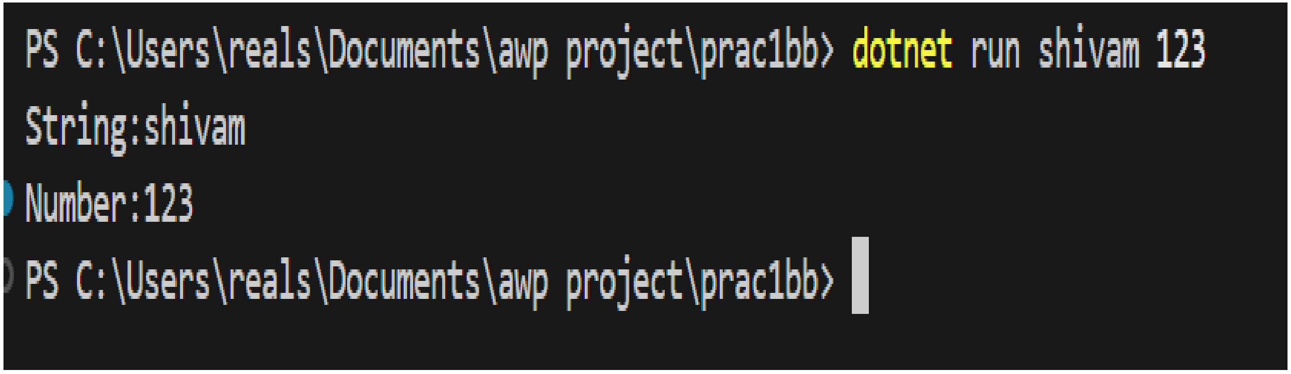


```
Time Elapsed 00:00:00.58
PS C:\Users\Admin\awppractical> dotnet run
Enter number1:4
Enter number 2:5
Enter number 3:6
Enter number 4:7
4*5*6*7=840
```

B) Create an application to demonstrate string operations.

```
using System;
namespace cmdLineArgs
{
class Program
{
static void Main(string[] args)
{
string str = args[0];
int n = Convert.ToInt32(args[1]);
Console.WriteLine("String:" + str);
Console.WriteLine("Number:" + n);
}
}
}
```

Output:

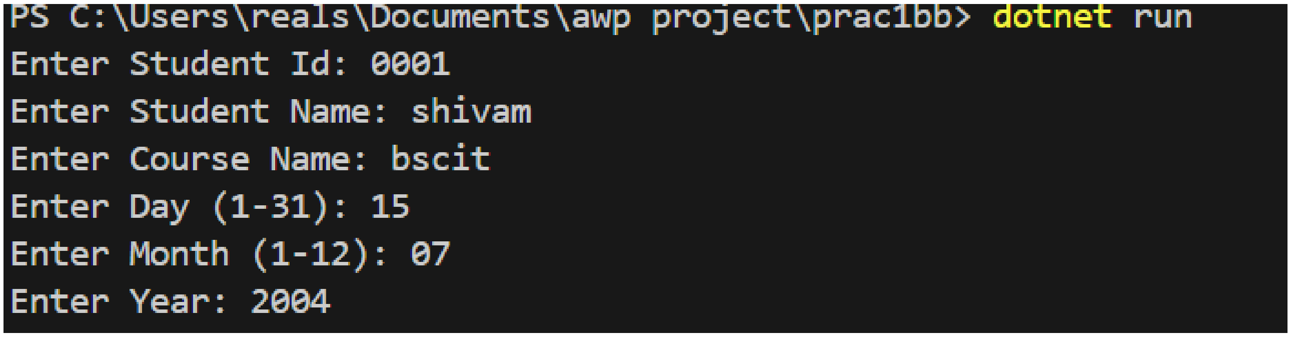


```
PS C:\Users\reals\Documents\awp project\prac1bb> dotnet run shivam 123
String:shivam
Number:123
PS C:\Users\reals\Documents\awp project\prac1bb>
```

C) 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.

```
using System;

namespace ArrayOfStructs

{

class Program

{

struct Student

{

public string studid, name, cname;

public int day, month, year;1

}

static void Main(string[] args)

{

Student[] s = new Student[5];

int i;

for (i = 0; i < 5; i++)

{

Console.Write("Enter Student Id:");

s[i].studid = Console.ReadLine();

Console.Write("Enter Student name : ");

s[i].name = Console.ReadLine();

Console.Write("Enter Course name : ");

s[i].cname = Console.ReadLine();

Console.Write("Enter date of birth\n Enter day(1-31):");

s[i].day = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter month(1-12):");

s[i].month = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter year:");

s[i].year = Convert.ToInt32(Console.ReadLine());
}
Console.WriteLine("\n\nStudent's List\n");
for (i = 0; i < 5; i++)
{
Console.WriteLine("\nStudent ID : " + s[i].studid);
Console.WriteLine("\nStudent name : " + s[i].name);
Console.WriteLine("\nCourse name : " + s[i].cname);
Console.WriteLine("\nDate of birth(dd-mm-yy) : " + s[i].day + "-" +
s[i].month +
"-" + s[i].year);
} } } }
```

Output:



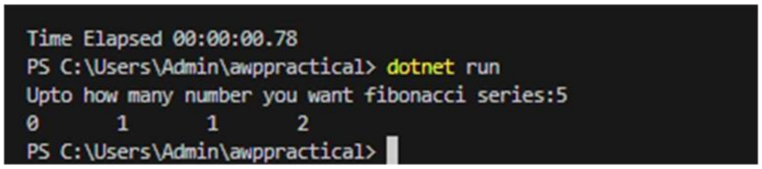
```
PS C:\Users\reals\Documents\awp project\prac1bb> dotnet run
Enter Student Id: 0001
Enter Student Name: shivam
Enter Course Name: bscit
Enter Day (1-31): 15
Enter Month (1-12): 07
Enter Year: 2004
```

D) Create an application to demonstrate following operations

[i] Fibonacci Series

```
using System;
namespace ConsoleApplication3
{
class Program
{
static void Main(string[] args)
{
int num1=0,num2=1,num3,num4,num,counter;
Console.Write ("Upto how many number you want fibonacci series:");
num=int.Parse(Console.ReadLine());
counter=3;
Console.Write(num1+"\t"+num2);
while(counter<=num)
{
num3 = num1 + num2;
if (counter >= num)
break;
Console.Write("\t" + num3);
num1 = num2;
num2 = num3;
counter++;
}}}}
```

Output:-



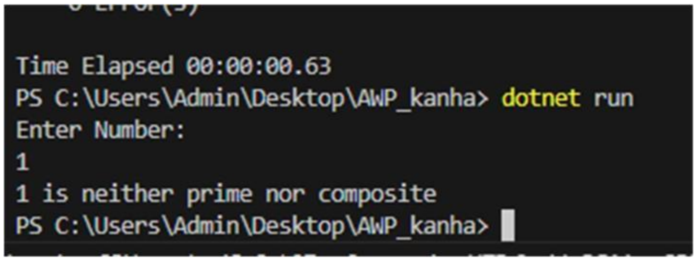
```
Time Elapsed 00:00:00.78
PS C:\Users\Admin\awppractical> dotnet run
Upto how many number you want fibonacci series:5
0       1       1       2
PS C:\Users\Admin\awppractical>
```

[ii] Test for prime numbers.

```
using System;

namespace TestPrime
{
  class Program
  {
    static void Main(string[] args)
    {
      int num, counter, result;
      Console.WriteLine("Enter Number:");
      num = int.Parse(Console.ReadLine());

      if (num == 1)
      {
        Console.WriteLine(num + " is neither prime nor composite");
      }
      else
      {
        // Initialize result to a non-zero value
        result = 0;

        for (counter = 2; counter < num; counter++)
        {
          if (num % counter == 0)
          {
            result = 0; // Not a prime number if any divisor is found
            break; // No need to check further
          }
        }
        if (result == 0)
        {
```

```
                    Console.WriteLine(num + " is not a prime number.");
                }
                else
                {
                    Console.WriteLine(num + " is a prime number.");
                }
            }
        }
    }
}
```

Output:-

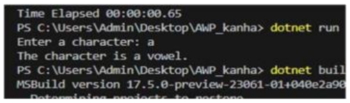


```
    0 Error(s)

Time Elapsed 00:00:00.63
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet run
Enter Number:
1
1 is neither prime nor composite
PS C:\Users\Admin\Desktop\AWP_kanha>
```

[iii] Test for vowels.

```
using System;
namespace vowels
{
  class Program
  {
    static void Main(string[] args)
    {
      char ch;
      Console.Write("Enter a character: ");
      ch = (char)Console.Read();
      switch(ch)
      {
        case 'a':
        case 'A':
        case 'e':
        case 'E':
        case 'i':
        case 'I':
        case 'o':
        case 'O':
        case 'u':
        case 'U':
          Console.WriteLine("The character is a vowel.");
          break;
        default:
          Console.WriteLine("The character is not a vowel.");
          break; }}}}
```

Output:-

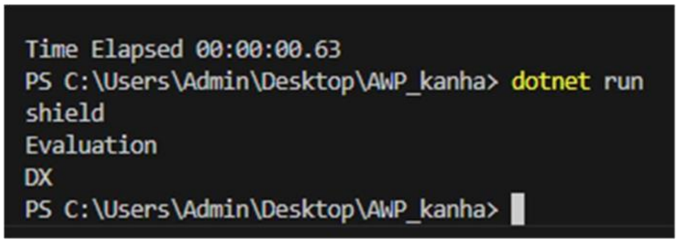


```
Time Elapsed 00:00:00.65
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet run
Enter a character: a
The character is a vowel.
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet buil
MSBuild version 17.5.0-preview-23061-01+040e2a90
  Determining projects to restore
```

[iv] Use of foreach loop with arrays.

```
using System;


class ExampleForEach
{
    public static void Main()
    {
        string[] str = {"shield", "Evaluation", "DX"};
        foreach(String s in str)
        {
            Console.WriteLine(s);
        }
    }
}
```

Output:-

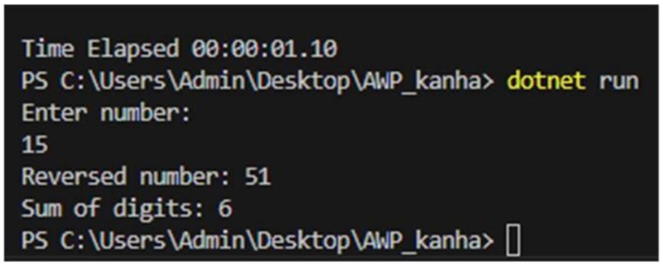


```
Time Elapsed 00:00:00.63
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet run
shield
Evaluation
DX
PS C:\Users\Admin\Desktop\AWP_kanha>
```

[v] Reverse a number and find sum of digits of a number.

```
using System;


namespace reverseNumber
{
    class Program
    {
        static void Main(string[] args)
        {
           int num,actualnumber,revnum=0,digit,sumDigits=0;
           Console.WriteLine("Enter number:");
           num = int.Parse(Console.ReadLine());
           actualnumber = num;
           while(num > 0)
           {
            digit = num % 10;
            revnum = (revnum * 10) + digit;
            sumDigits = sumDigits+digit;
            num = num / 10;
           }
            Console.WriteLine("Reversed number: " + revnum);
            Console.WriteLine("Sum of digits: " + sumDigits);
    }
}
}
```

Output:-



```
Time Elapsed 00:00:01.10
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet run
Enter number:
15
Reversed number: 51
Sum of digits: 6
PS C:\Users\Admin\Desktop\AWP_kanha> 
```

Practical No.: 2

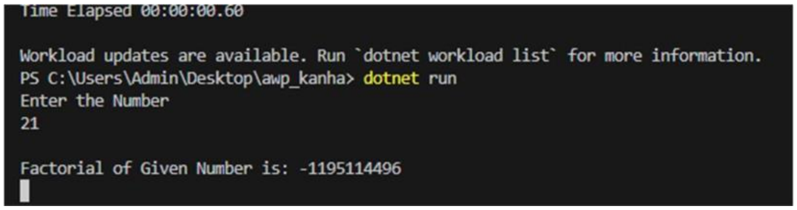
AIM: Working with Object Oriented C# and ASP .NET

A) **Create simple application to perform following operations.**

[i] Finding Factorial Value

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace factorial
{
class Program
{
static void Main(string[] args)
{
int i, number, fact;
Console.WriteLine("Enter the Number");
number = int.Parse(Console.ReadLine());
fact = number;
for (i = number - 1; i >= 1; i--)
{
fact = fact * i;
}
Console.WriteLine("\nFactorial of Given Number is: "+fact);
Console.ReadLine();
}
}
}
```

Output:-



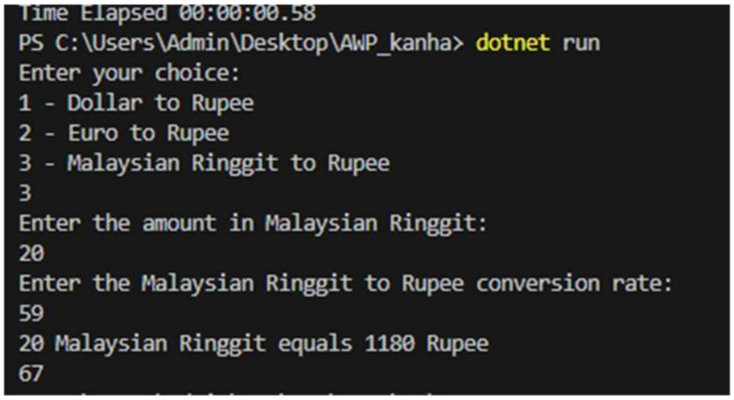
```
Time Elapsed 00:00:00.60

Workload updates are available. Run `dotnet workload list` for more information.
PS C:\Users\Admin\Desktop\awp_kanha> dotnet run
Enter the Number
21

Factorial of Given Number is: -1195114496
```

[ii] Money Conversion

```
using System;

namespace CurrencyConversion

{

   class Program

   {

      static void Main(string[] args)

      {

         int choice;

         Console.WriteLine("Enter your choice:\n1 - Dollar to Rupee\n2 - Euro to Rupee\n3 -
Malaysian Ringgit to Rupee");

         if (!int.TryParse(Console.ReadLine(), out choice) || choice < 1 || choice > 3)

         {

            Console.WriteLine("Invalid choice. Please select a valid option (1, 2, or 3).");

            return;

         }

         switch (choice)

         {

            case 1:

               ConvertCurrency("Dollar", "Rupee");

               break;

            case 2:

               ConvertCurrency("Euro", "Rupee");

               break;

            case 3:

               ConvertCurrency("Malaysian Ringgit", "Rupee");

               break;

            default:

               Console.WriteLine("Unexpected error. Please select a valid option.");

               break;

         }

         Console.ReadLine();

      }
```

```
        static void ConvertCurrency(string fromCurrency, string toCurrency)
        {
            double amount, value, convertedAmount;
            Console.WriteLine($"Enter the amount in {fromCurrency}: ");
            if (!double.TryParse(Console.ReadLine(), out amount) || amount < 0)
            {
                Console.WriteLine("Invalid amount. Please enter a positive number.");
                return;
            }
            Console.WriteLine($"Enter the {fromCurrency} to {toCurrency} conversion rate: ");
            if (!double.TryParse(Console.ReadLine(), out value) || value <= 0)
            {
                Console.WriteLine("Invalid conversion rate. Please enter a positive number.");
                return;
            }
            convertedAmount = amount * value;
            Console.WriteLine($"{amount} {fromCurrency} equals {convertedAmount} {toCurrency}");
        }
    }
}
```

Output:

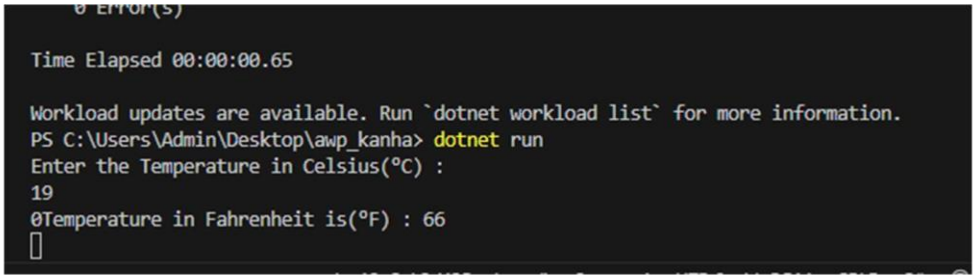


```
Time Elapsed 00:00:00.58
PS C:\Users\Admin\Desktop\AWP_kanha> dotnet run
Enter your choice:
1 - Dollar to Rupee
2 - Euro to Rupee
3 - Malaysian Ringgit to Rupee
3
Enter the amount in Malaysian Ringgit:
20
Enter the Malaysian Ringgit to Rupee conversion rate:
59
20 Malaysian Ringgit equals 1180 Rupee
67
```

[iii] Temperature Conversion

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace temperatureconversion
{
class Program
{
static void Main(string[] args)
{
int celsius, faren;
Console.WriteLine("Enter the Temperature in Celsius(°C) : ");
celsius = int.Parse(Console.ReadLine());
faren = (celsius * 9) / 5 + 32;
Console.WriteLine("0Temperature in Fahrenheit is(°F) : " + faren);
Console.ReadLine();
}
}
}
```

Output:-



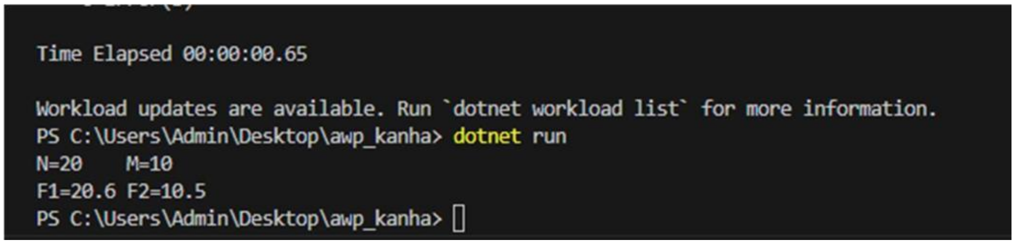
```
    0 Error(s)

Time Elapsed 00:00:00.65

Workload updates are available. Run `dotnet workload list` for more information.
PS C:\Users\Admin\Desktop\awp_kanha> dotnet run
Enter the Temperature in Celsius(°C) :
19
0Temperature in Fahrenheit is(°F) : 66
```

B) Create simple application to demonstrate use of following concepts.

[i] Function Overloading

```
using System;
namespace swap
{
class Overloading
{
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;
}
}
class program
{
static void Main(string[] args)
{
Overloading objOverloading = new Overloading();
int n = 10, m = 20;
objOverloading.swap(ref n, ref m);
Console.WriteLine("N=" + n + "\tM=" + m);
float f1 = 10.5f, f2 = 20.6f;
objOverloading.swap(ref f1, ref f2);
Console.WriteLine("F1=" + f1 + "\tF2=" + f2);

}

}

}
```

Output:-



```
    0 Error(s)

Time Elapsed 00:00:00.65

Workload updates are available. Run `dotnet workload list` for more information.
PS C:\Users\Admin\Desktop\awp_kanha> dotnet run
N=20    M=10
F1=20.6 F2=10.5
PS C:\Users\Admin\Desktop\awp_kanha> 
```

[ii] Inheritance

(a) Single Inheritance

Write a program to implement single inheritance from following figure. Accept and display data for one table.

Furniture.cs

```
using System;
namespace SingleInheritance
{
class Furniture
{
string material;
float price;
public void getdata()
{
Console.Write("Enter material : ");
material = Console.ReadLine();
Console.Write("Enter price : ");
price = float.Parse(Console.ReadLine());
}
public void showdata()
{Console.WriteLine("Material : " + material);
Console.WriteLine("Price : " + price);
} } }
```

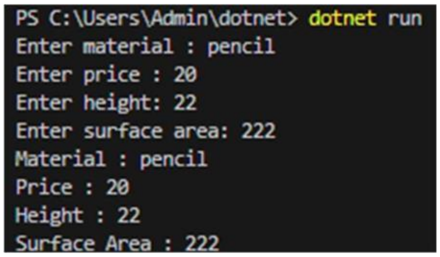
Table.cs

```
using System;
namespace SingleInheritance
{
class Table:Furniture
{
int height, surface_area;
public void getdata()
{
base.getdata();
Console.Write("Enter height: ");
height = int.Parse(Console.ReadLine());
Console.Write("Enter surface area: ");
surface_area = int.Parse(Console.ReadLine());
}
public void showdata()
{
base.showdata();
Console.WriteLine("Height : " + height);
Console.WriteLine("Surface Area : " + surface_area);
} } }
```

Program.cs

```
using System;
namespace SingleInheritance
{
class Program
{
static void Main(string[] args)
{
Table t1 = new Table();
t1.getdata();
t1.showdata();
} } }
```

Output:-



```
PS C:\Users\Admin\dotnet> dotnet run
Enter material : pencil
Enter price : 20
Enter height: 22
Enter surface area: 222
Material : pencil
Price : 20
Height : 22
Surface Area : 222
```

[ii](b) Multiple inheritance

Gross.cs

```
using System;
namespace MultipleInheritance
{
interface Gross
{
int ta
{
get;
set;
}
int da
{
get;
set;
}
int GrossSal();
} }
```

Employee.cs

```
using System;
namespace MultipleInheritance
{
class Employee
{
string name;
public Employee(string name)
{ this.name = name; }
public int BasicSal(int basicSal)
{ return basicSal; }
public void ShowData()
{
Console.WriteLine("Name : " + name);
} } }
```

Salary.cs

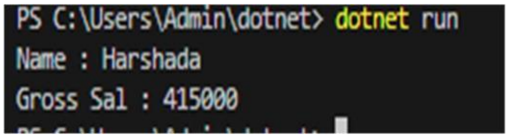
```
using System;
namespace MultipleInheritance
{
class Salary:employee,Gross
{
int hra;
public Salary(string name, int hra):base(name)
{ this.hra = hra; }
public int ta
{
get {return S_ta; }
set { S_ta = value; }
}
private int S_ta;
public int da
{
```

```
get { return S_da; }
set { S_da = value; }
}
private int S_da;
public int GrossSal()
{
int gSal;
gSal = hra + ta + da + BasicSal(15000);
return gSal;
}
public void dispSal()
{ base.ShowData();
Console.WriteLine("Gross Sal : " + GrossSal());
} } }
```

Program.cs

```
using System;
namespace MultipleInheritance
{
class Program
{

static void Main(string[] args)

{

Salary s = new Salary("Prachit", 35000);

s.da = 20000;

s.ta = 30000;

s.dispSal();

} } }
```

Output:



```
PS C:\Users\Admin\dotnet> dotnet run
Name : Harshada
Gross Sal : 415000
```

(ii)[c] Heirarchical Inheritance

Employee.cs

```
using System;
namespace HeirarchicalInheritance
{
class employee
{
public virtual void display()
{
Console.WriteLine("Display of employee class called ");
} } }
```

Programmer.cs

```
using System;
namespace HeirarchicalInheritance
{
class Programmer:employee
{
public void display()
{
Console.WriteLine(" Display of Programmer class called ");
} } }
```

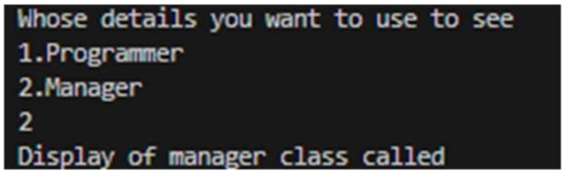
Manager.cs

```
using System;
namespace HeirarchicalInheritance
{
class Manager
{
public void display()
{
Console.WriteLine("Display of manager class called ");
} } }
```

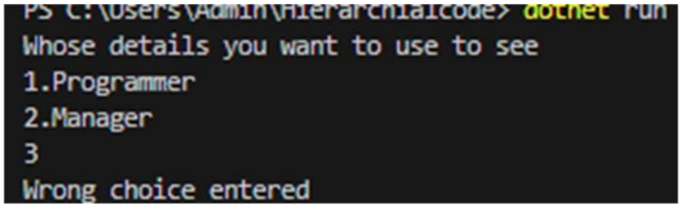
Program.cs

```
using System;
namespace HeirarchicalInheritance
{
class Program
{
static void Main(string[] args)
{
Programmer objProgrammer;
Manager objManager;
Console.Write("Whose details you want to use to see \n1.Programmer \n2.Manager");
int choice=int.Parse(Console.ReadLine());
if(choice==1)
{
objProgrammer=new Programmer();
objProgrammer.display();
}
else if(choice==2)
{
objManager=new Manager();
objManager.display();
}
else
{
Console.WriteLine("Wrong choice entered");
} } } }
```

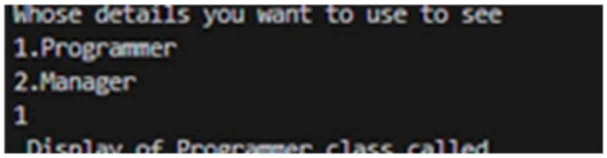
Output:



```
Whose details you want to use to see
1.Programmer
2.Manager
2
Display of manager class called
```



```
PS C:\Users\Admin\Hierarchialcode> dotnet run
Whose details you want to use to see
1.Programmer
2.Manager
3
Wrong choice entered
```



```
Whose details you want to use to see
1.Programmer
2.Manager
1
 Display of Programmer class called
```

(ii)[d] Multilevel Inheritance

Result.cs

```
using System;
namespace multilevelinheritance
{
class Result:Test
{
int total;
public Result(int roll_no, string name, int marks1, int marks2)
: base(roll_no, name, marks1, marks2)
{
total = getMarks1() + getMarks2();
}
public void display()
{
base.display();
Console.WriteLine("Total: " + total);
} } }
```

Test.cs

```
using System;
namespace multilevelinheritance
{
class Test:student
{
int marks1, marks2;
public Test(int roll_no, string name, int marks1, int marks2)
: base(roll_no, name)
{
this.marks1 = marks1;
this.marks2 = marks2;
}
public int getMarks1()
{
return marks1;
}
public int getMarks2()
{
return marks2;
}
public void dispaly()
{
base.display();
Console.WriteLine("Marks1: " + marks1);
Console.WriteLine("Marks2: " + marks2);
} } }
```

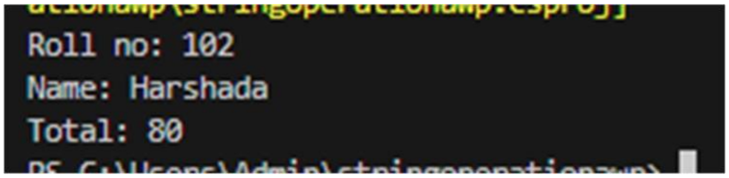
Student.cs

```
using System;
namespace multilevelinheritance
{
class student
{
int roll_no;
string name;
public student(int roll_no, string name)
{
this.roll_no = roll_no;
this.name = name;
}
public student() { }
public void display()
{
Console.WriteLine("Roll no: " + roll_no);
Console.WriteLine("Name: " + name);
} } }
```

Program.cs

```
using System;

namespace multilevelinheritance

{

class Program

{

static void Main(string[] args)

{

Result r1 = new Result(101, "Prachit", 50, 70);

r1.display();

} } }
```

Output:



```
ationawp\stringoperationawp.csproj]
Roll no: 102
Name: Harshada
Total: 80
PS C:\Users\Admin\stringoperationawp>
```

B.[iii] Constructor Overloading

Salary.cs

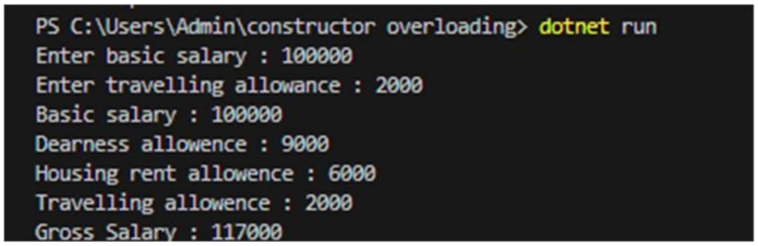
```
using System;
namespace SalaryConstructure
{
class Salary
{
int basic, ta, da, hra;
public Salary()
{
da = 9000;
hra = 6000;
}
public void getdata()
{
Console.Write("Enter basic salary : ");
basic = int.Parse(Console.ReadLine());
Console.Write("Enter travelling allowance : ");
ta = int.Parse(Console.ReadLine());
}
public void showdata()
{
Console.WriteLine("Basic salary : " + basic);
Console.WriteLine("Dearness allowence : " + da);
Console.WriteLine("Housing rent allowence : " + hra);
Console.WriteLine("Travelling allowence : " + ta);
Console.WriteLine("Gross Salary : " + (basic + da + hra + ta));
} } }
```

Program.cs

```
using System;
namespace SalaryConstructure
{
```

```
class Program

{

static void Main(string[] args)

{

Salary s = new Salary();

s.getdata();

s.showdata();

} } }
```

Output:-



```
PS C:\Users\Admin\constructor overloading> dotnet run
Enter basic salary : 100000
Enter travelling allowance : 2000
Basic salary : 100000
Dearness allowence : 9000
Housing rent allowence : 6000
Travelling allowence : 2000
Gross Salary : 117000
```

(C) Create simple application to demonstrate use of following concepts.

[i] Using Delegates and events

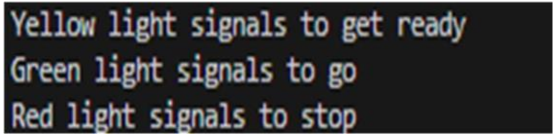
TrafficSignal.cs

```
using System;
namespace TrafficDelegateExample
{
public delegate void TrafficDel();
class TrafficSignal
{
public static void Yellow()
{
Console.WriteLine("Yellow light signals to get ready");
}
public static void Green()
{
Console.WriteLine("Green light signals to go");
}
public static void Red()
{
Console.WriteLine("Red light signals to stop");

}

TrafficDel[] td = new TrafficDel[3];

public void IdentifySignal()
{
td[0] = new TrafficDel(Yellow);
td[1] = new TrafficDel(Green);
td[2] = new TrafficDel(Red);
}
public void display()
{
td[0]();
td[1]();
td[2]();
}
} }
```

Program.cs

```
using System;
namespace TrafficDelegateExample
{
class Program
{
static void Main(string[] args)
{
TrafficSignal ts = new TrafficSignal();
ts.IdentifySignal();
ts.display();
} } }
```

Output:



```
Yellow light signals to get ready
Green light signals to go
Red light signals to stop
```

Practical No.: 3

AIM: Working with Web Forms and Controls.

A) Demonstrate the use of Treeview control perform following operations.
i. Treeview control and datalist

Webform1.aspx

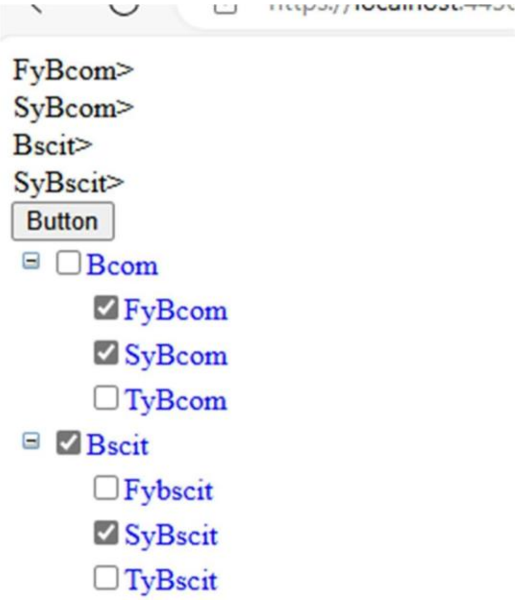
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="prac3_a.WebForm1 %>

<!DOCTYPE html>

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

<head runat="server">

  <title></title>

</head>

<body>

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

    <div>

      <asp:DataList ID="DataList1" runat="server">

        <ItemTemplate>

          <%# Eval("text") %>>

        </ItemTemplate>

      </asp:DataList>

      <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Button" />

      <asp:TreeView ID="TreeView3" runat="server">

        <Nodes>

          <asp:TreeNode Text="Bcom" Value="Bcom" Checked="True" ShowCheckBox="True">

            <asp:TreeNode Text="FyBcom" Value="FyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

            <asp:TreeNode Text="SyBcom" Value="SyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

            <asp:TreeNode Text="TyBcom" Value="TyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

          </asp:TreeNode>

          <asp:TreeNode Text="Bscit" Value="Bscit" Checked="True" ShowCheckBox="True">

            <asp:TreeNode Text="Fybscit" Value="Fybscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>
```

```
                        <asp:TreeNode Text="SyBscit" Value="SyBscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>
                        <asp:TreeNode Text="TyBscit" Value="TyBscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>
                    </asp:TreeNode>
                </Nodes>
            </asp:TreeView>
        </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace prac3_a
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button2_Click(object sender, EventArgs e)
        {
            {
                {
                    TreeNodeCollection  T;
                    T = TreeView3.CheckedNodes;
                    DataList1.DataSource = T;
                    DataList1.DataBind();
                    DataList1.Visible = true;
                }
            }
        }
    }
}
```

Output:-



```
FyBcom>
SyBcom>
Bscit>
SyBscit>
[Button]
⊟ □ Bcom
      ☑ FyBcom
      ☑ SyBcom
      □ TyBcom
⊟ ☑ Bscit
      □ Fybscit
      ☑ SyBscit
      □ TyBscit
```

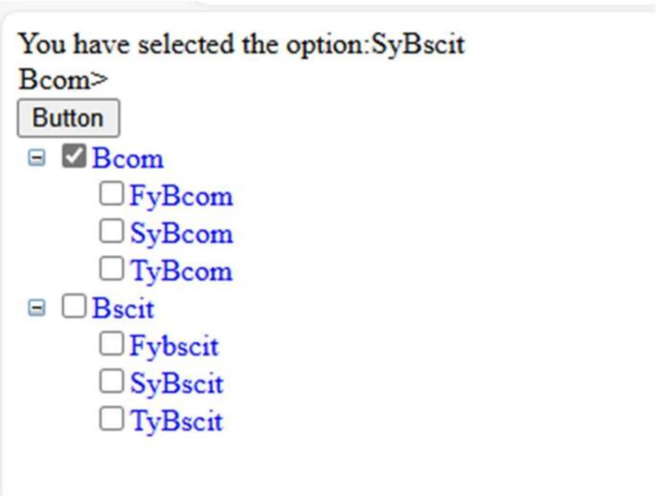
ii. TreeView Operations

WebForm1.apsx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="prac3_a.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

  <title></title>

</head>

<body>

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

    <div>

      <asp:DataList ID="DataList1" runat="server">

        <ItemTemplate>

          <%# Eval("text") %>>

        </ItemTemplate>

      </asp:DataList>

      <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Button" />

      <asp:TreeView ID="TreeView3" runat="server"
OnSelectedNodeChanged="TreeView3_SelectedNodeChanged"
OnTreeNodeCollapsed="TreeView3_TreeNodeCollapsed"> //Just Changed this
“OnTreeNodeCollapsed”

        <Nodes>

          <asp:TreeNode Text="Bcom" Value="Bcom" Checked="True" ShowCheckBox="True">

            <asp:TreeNode Text="FyBcom" Value="FyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

            <asp:TreeNode Text="SyBcom" Value="SyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

            <asp:TreeNode Text="TyBcom" Value="TyBcom" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

          </asp:TreeNode>

          <asp:TreeNode Text="Bscit" Value="Bscit" Checked="True" ShowCheckBox="True">

            <asp:TreeNode Text="Fybscit" Value="Fybscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

            <asp:TreeNode Text="SyBscit" Value="SyBscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>
```

```
                  <asp:TreeNode Text="TyBscit" Value="TyBscit" Checked="True"
ShowCheckBox="True"></asp:TreeNode>

               </asp:TreeNode>

            </Nodes>

         </asp:TreeView>

      </div>

   </form>

</body>

</html>
```

Output:-



You have selected the option:SyBscit
Bcom>

```
⊟ ☑ Bcom
      ☐ FyBcom
      ☐ SyBcom
      ☐ TyBcom
⊟ ☐ Bscit
      ☐ Fybscit
      ☐ SyBscit
      ☐ TyBscit
```

B) Demonstrate the use of Calendar control to perform followingoperations.

a) **Display messages in a calendar control**

b) **Display vacation in a calendarcontrol**

c) **Selected day in a calendar control using style**

d) **Difference between two calendardates**

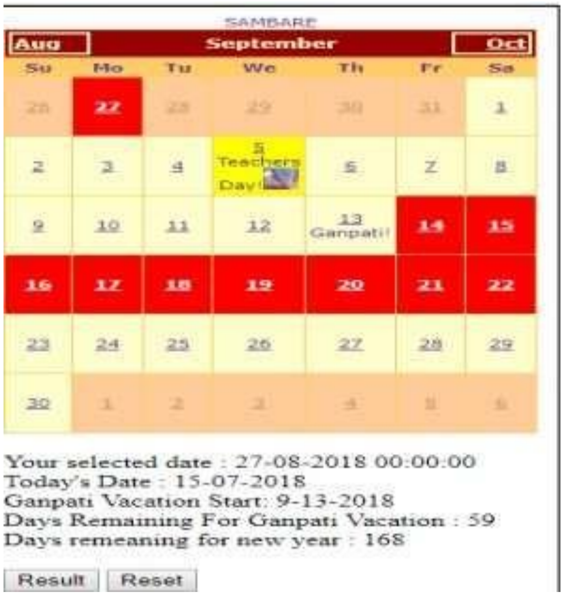
Calender properties set for this example:

```
<asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC"
BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399"
Height="200px"
NextPrevFormat="ShortMonth" OnDayRender="Calendar1_DayRender"
ShowGridLines="True" Width="300px"
OnSelectionChanged="Calendar1_SelectionChanged" >
<DayHeaderStyle  BackColor="#FFCC66"  Font-Bold="True"  Height="1px"  />
<NextPrevStyle BorderStyle="Solid" BorderWidth="2px" Font-Size="9pt"
ForeColor="#FFFFCC" />
<OtherMonthDayStyle  BackColor="#FFCC99"  BorderStyle="Solid"

ForeColor="#CC9966"  />
<SelectedDayStyle  BackColor="Red"  Font-Bold="True"  />
<SelectorStyle  BackColor="#FFCC66"  />
<TitleStyle  BackColor="#990000"  Font-Bold="True"  Font-Size="9pt"

ForeColor="#FFFFCC" />

<TodayDayStyle  BackColor="#FFCC66"  ForeColor="White"  />
<WeekendDayStyle  Height="50px"  />
</asp:Calendar>
```

calndrCtrl.aspx.cs

```
protected void btnResult_Click(object sender, EventArgs e)
{
Calendar1.Caption = "SAMBARE";
Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;
Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;
Calendar1.TitleFormat = TitleFormat.Month;


Label2.Text = "Todays Date"+Calendar1.TodaysDate.ToShortDateString();
Label3.Text = "Ganpati Vacation Start: 9-13-2018";
TimeSpan d = new DateTime(2018, 9, 13) - DateTime.Now;
Label4.Text = "Days Remaining For Ganpati Vacation:"+d.Days.ToString();
TimeSpan d1 = new DateTime(2018, 12, 31) - DateTime.Now; Label5.Text = "Days Remaining for
New Year:"+d1.Days.ToString(); if (Calendar1.SelectedDate.ToShortDateString() == "9-13-2018")
Label3.Text = "<b>Ganpati Festival Start</b>";
if (Calendar1.SelectedDate.ToShortDateString() == "9-23-2018")
Label3.Text = "<b>Ganpati Festival End</b>";
}
protected void Calendar1_DayRender(object sender,
System.Web.UI.WebControls.DayRenderEventArgs e)
{
if (e.Day.Date.Day == 5 && e.Day.Date.Month == 9)
{
e.Cell.BackColor = System.Drawing.Color.Yellow; Label lbl = new Label(); lbl.Text = "<br>Teachers
Day!"; e.Cell.Controls.Add(lbl); Image g1 = new Image(); g1.ImageUrl = "td.jpg"; g1.Height = 20;
g1.Width = 20;
e.Cell.Controls.Add(g1);


}
if (e.Day.Date.Day == 13 && e.Day.Date.Month == 9)
{
Calendar1.SelectedDate = new DateTime(2018, 9, 12);
Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate,
Calendar1.SelectedDate.AddDays(10)); Label lbl1 = new Label(); lbl1.Text = "<br>Ganpati!";
e.Cell.Controls.Add(lbl1);
}
}
protected void btnReset_Click(object sender, EventArgs e)
{
Label1.Text = "";
Label2.Text = "";
Label3.Text = "";
Label4.Text = "";
Label5.Text = "";
Calendar1.SelectedDates.Clear();
}
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
Label1.Text = "Your Selected Date:" +
Calendar1.SelectedDate.Date.ToString();
}
```

OUTPUT:-



Aug			September			Oct
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5 Teachers Day!	6	7	8
9	10	11	12	13 Ganpati!!	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

```
Your selected date : 27-08-2018 00:00:00
Today's Date : 15-07-2018
Ganpati Vacation Start: 9-13-2018
Days Remaining For Ganpati Vacation : 59
Days remeaning for new year : 168
```

Practical No.: 4

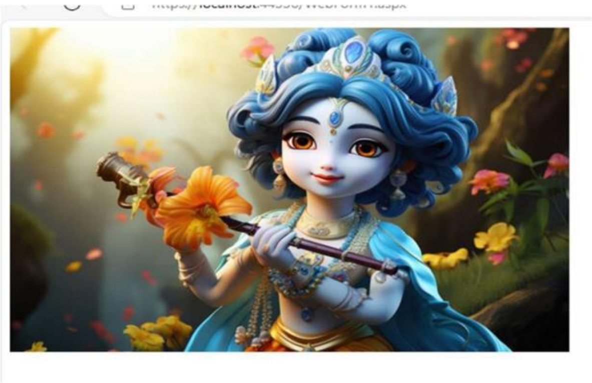
AIM: Working with form controls

A) Create a web form to demonstrate the Adrotator Control.

XML File

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
 <Ad>
  <ImageUrl>~/Images/rk.jpg</ImageUrl>
  <Navigateurl></Navigateurl>
  <AlternateText> Radha</AlternateText>
  <Impressions>50</Impressions>
  <Keyword>Radha</Keyword>
 </Ad>
 <Ad>
  <ImageUrl>~/Images/madhav.jpg</ImageUrl>
  <Navigateurl></Navigateurl>
  <AlternateText> madhav</AlternateText>
  <Impressions>50</Impressions>
  <Keyword>Madhav</Keyword>
 </Ad>
 <Ad>
  <ImageUrl>~/Images/ganesha.jpg</ImageUrl>
  <Navigateurl></Navigateurl>
  <AlternateText> madhav</AlternateText>
  <Impressions>50</Impressions>
  <Keyword>Madhav</Keyword>
 </Ad>
 <Ad>
  <ImageUrl>~/Images/mahadev.jpg</ImageUrl>
  <Navigateurl></Navigateurl>
  <AlternateText> madhav</AlternateText>
  <Impressions>50</Impressions>
  <Keyword>Madhav</Keyword>
 </Ad>
 <Ad>
  <ImageUrl>~/Images/madhav.jpg</ImageUrl>
  <Navigateurl></Navigateurl>
  <AlternateText> madhav</AlternateText>
  <Impressions>50</Impressions>
  <Keyword>Madhav</Keyword>
 </Ad>
</Advertisements>
```

Output:-



B) Create web form to demonstrate use User Controls.

Webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication5.WebForm1" %>
<%@ Register src="~/WebUserControl1.ascx" TagName="WebUserControl" TagPrefix="uc1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      <uc1:WebUserControl runat="server" id="WebUserControl1" />
    </div>
  </form>
</body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication5
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
```

```
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            TextBox2.Text = "heelo guest" + TextBox1.Text;
        }
    }
}
```

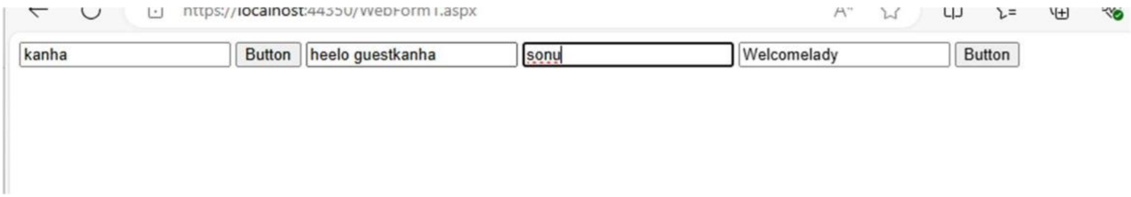
WebUserControl.ascx

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="WebUserControl1.ascx.cs"
Inherits="WebApplication5.WebUserControl1" %>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
```

WebUserControl.ascx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication5
{
    public partial class WebUserControl1 : System.Web.UI.UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            TextBox2.Text = "Welcome" + TextBox1.Text;
        }
    }
}
```

Output:-



https://localhost:44350/WebForm1.aspx

kanha

heelo guestkanha

sonu

Welcomelady

C) Create web form to demonstrate use Validation Controls.

WebForm1.aspx

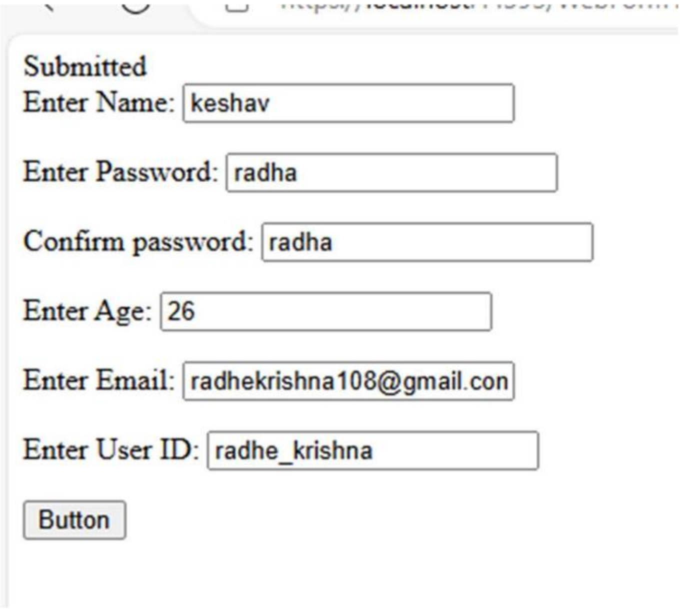
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication6.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

  <title></title>

</head>

<body>

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

    <div>

      <asp:Label ID="Label1" runat="server" Text="Enter Name:"></asp:Label>

      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

      <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="*Name Required"
ForeColor="#CC3300"></asp:RequiredFieldValidator>

    </div>

    <p>

      <asp:Label ID="Label2" runat="server" Text="Enter Password:"></asp:Label>

      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

      <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="*Password Required"
ForeColor="#CC3300"></asp:RequiredFieldValidator>

    </p>

    <p>

      <asp:Label ID="Label3" runat="server" Text="Confirm password:"></asp:Label>

      <asp:TextBox ID="TextBox3" runat="server"
OnTextChanged="TextBox3_TextChanged"></asp:TextBox>

      <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
ControlToValidate="TextBox3" ErrorMessage="*Password required"
ForeColor="#CC3300"></asp:RequiredFieldValidator>

      <asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="TextBox2" ControlToValidate="TextBox3" ErrorMessage="*Enter same
password" ForeColor="#CC3300"></asp:CompareValidator>

    </p>
```

```
        <p>
            <asp:Label ID="Label4" runat="server" Text="Enter Age:"></asp:Label>
            <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
ControlToValidate="TextBox4" ErrorMessage="*Enter Age"
ForeColor="#CC3300"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="TextBox4"
ErrorMessage="*Age required should be between 21 to 30" ForeColor="#CC3300"
MaximumValue="30" MinimumValue="21" Type="Integer"></asp:RangeValidator>
        </p>
        <p>
            <asp:Label ID="Label5" runat="server" Text="Enter Email:"></asp:Label>
            <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
ControlToValidate="TextBox5" ErrorMessage="*Email required"
ForeColor="#CC3300"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
ControlToValidate="TextBox5" ErrorMessage="*Email should be proper" ForeColor="#CC3300"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
        </p>
        <p>
            <asp:Label ID="Label6" runat="server" Text="Enter User ID:"></asp:Label>
            <asp:TextBox ID="TextBox6" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ControlToValidate="TextBox6" ErrorMessage="*User ID required"
ForeColor="#CC3300"></asp:RequiredFieldValidator>
        </p>
        <p>
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Button" />
        </p>
    </form>
</body>
</html>
```

WebForm1.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebApplication6
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void TextBox3_TextChanged(object sender, EventArgs e)
      {
      }
      protected void Button1_Click(object sender, EventArgs e)
      {
         Response.Write("Submitted");
      }
   }
}
```

Output:-



Submitted
Enter Name: keshav
Enter Password: radha
Confirm password: radha
Enter Age: 26
Enter Email: radhekrishna108@gmail.con
Enter User ID: radhe_krishna

Practical No.: 5

AIM: Working with Navigation, Beautification and Master page

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

Master1.master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs"
Inherits="prac5b.Site1" %>
<!DOCTYPE html>
<html>
<head runat="server">
  <title></title>
  <asp:ContentPlaceHolder ID="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <link href="StyleSheet1.css" rel="stylesheet" type="text/css" />
  <form id="form1" runat="server">
    <div>
      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
      </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>
```

WebForm1.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="prac5b.WebForm1" Theme ="Skin1"%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
  <asp:Label ID="Label1" runat="server" SkinId="lbl" Text="Select The date"></asp:Label>
  <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
  <br />
```

```
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/WebForm2.aspx">Next</asp:HyperLink>

</asp:Content>
```

WebForm2.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="prac5b.WebForm2" Theme="Skin1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

    <p>

        <br />

        <asp:Label ID="Label1" runat="server" Text="Label" SkinId="lbl"></asp:Label>

    </p>

    <p>

        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

    </p>

    </asp:Content>
```

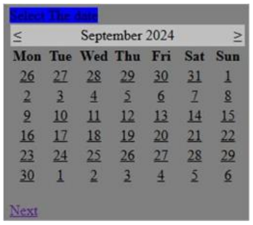
Skin1.skin

```
<asp:Label runat="server" SkinId="lbl" backcolor="blue"/>
```

StyleSheet1.css

```
body {

    background-color: gray;

    font:italic;

}
```

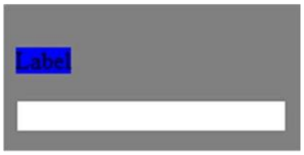
Output:



Select The date

<	September 2024					>
Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	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	1	2	3	4	5	6

Next



Label

B) Create Web Application to demonstrate various states of ASP.NET pages

i.View State

Default.aspx

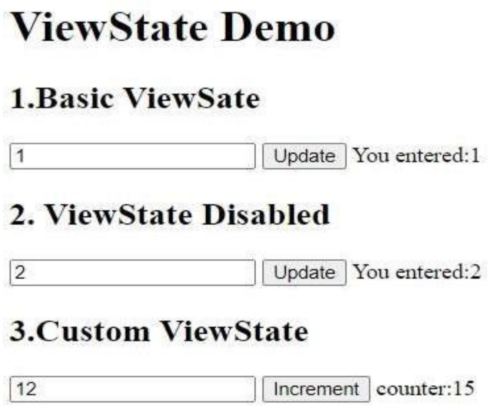
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="ViewStateDemo.Default" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>ViewState Demo</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h1>ViewState Demo</h1>

         <h2>1. Basic ViewState</h2>
         <asp:TextBox ID="txtBasic" runat="server"></asp:TextBox>
         <asp:Button ID="btnBasic" runat="server" Text="Update" OnClick="btnBasic_Click" />
         <asp:Label ID="lblBasic" runat="server"></asp:Label>

         <h2>2. ViewState Disabled</h2>
         <asp:TextBox ID="txtDisabled" runat="server" EnableViewState="false"></asp:TextBox>
         <asp:Button ID="btnDisabled" runat="server" Text="Update"
OnClick="btnDisabled_Click"/>
         <asp:Label ID="lblDisabled" runat="server"></asp:Label>

         <h2>3. Custom ViewState</h2>
         <asp:TextBox ID="txtCustom" runat="server"></asp:TextBox>
         <asp:Button ID="btnCustom" runat="server" Text="Increment"
OnClick="btnCustom_Click"/>
         <asp:Label ID="lblCustom" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

Default.aspx.cs

```
using System;

namespace ViewStateDemo
{
   public partial class Default : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            ViewState["Counter"] = 0;
         }
      }

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

```
        {
            lblBasic.Text = $"You entered: {txtBasic.Text}";
        }

        protected void btnDisabled_Click(object sender, EventArgs e)
        {
            lblDisabled.Text = $"You entered: {txtDisabled.Text}";
        }

        protected void btnCustom_Click(object sender, EventArgs e)
        {
            int counter = (int)ViewState["Counter"];
            counter++;
            ViewState["Counter"] = counter;
            lblCustom.Text = $"Counter: {counter}";
        }
    }
}
```

Output:



ViewState Demo

1.Basic ViewSate

1 You entered:1

2. ViewState Disabled

2 You entered:2

3.Custom ViewState

12 counter:15

ii.Session State

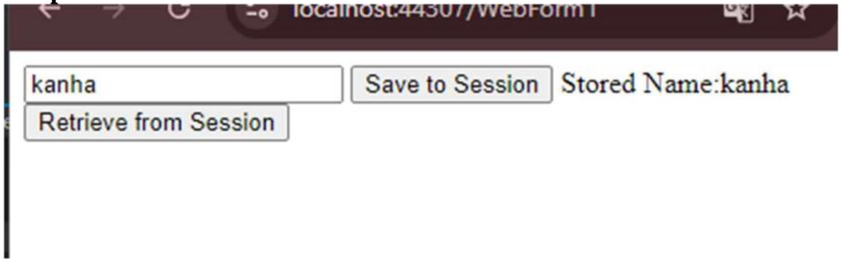
Default.aspx

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

<!DOCTYPE html>

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

<head runat="server">

  <title></title>

</head>

<body>

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

    <div>

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

      <asp:Button ID="btnSaveSession" runat="server" Text="Save to Session"
OnClick="btnSaveSession_Click" />

      <asp:Label ID="lblSessionResult" runat="server"></asp:Label>

      <asp:Button ID="btnRetrieveSession" runat="server" Text="Retrieve from Session"
OnClick="btnRetrieveSession_Click" />

    </div>

  </form>

</body>

</html>
```

Default.aspx.cs

```
using System;

namespace prac_c

{

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

  {

    protected void btnSaveSession_Click(object sender, EventArgs e)

    {

      Session["UserName"] = txtName.Text;

      lblSessionResult.Text = "Name saved to session!";

    }

    protected void btnRetrieveSession_Click(object sender, EventArgs e)
    {
      if(Session["UserName"]!=null)
      {
        lblSessionResult.Text = "Stored Name:" + Session["UserName"].ToString();
      }
      else
      {
        lblSessionResult.Text = "No name found in session";
      }
    }
  }
}
```

Output:-



kanha

Stored Name:kanha

Practical No.:6

AIM: Demonstrate the use of DataList link Control

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="prac6_c.WebForm1" %>

<!DOCTYPE html>

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

<head runat="server">

  <title></title>

</head>

<body>

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

    <div>

      <h2>Book List</h2>

      <asp:DataList ID="dlBooks" runat="server" RepeatColumns="2">

        <ItemTemplate>

          <div style="margin-bottom: 10px; padding: 10px; border:1px solid #ccc;">

            <h3><%#Eval("Title") %></h3>

            <p>Author: <%#Eval("Author") %></p>

            <p>Price: $<%# Eval("Price", "{0:F2}") %></p>

            <p>Price: $<%# Eval("Price", "{0:F2}") %></p>

          </div>

        </ItemTemplate>

      </asp:DataList>

    </div>
```

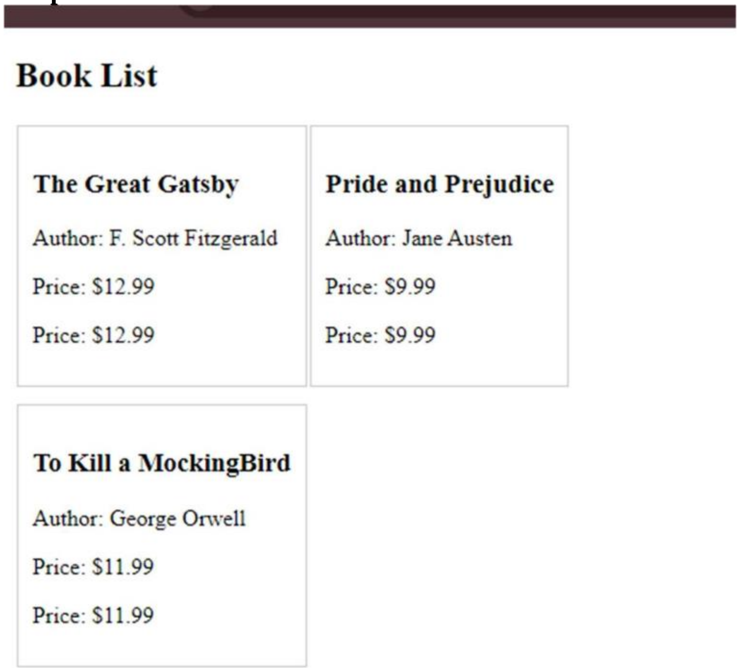
Default.aspx.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;
```

```
namespace prac6_c
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindDataList();
            }
        }
        private void BindDataList()
        {
            List<Book> books = new List<Book>
            {
                new Book { Title = "The Great Gatsby", Author = "F. Scott Fitzgerald", Price = 12.99m},
                new Book { Title = "To Kill a MockingBird", Author = "George Orwell", Price = 11.99m},
                new Book { Title = "Pride and Prejudice", Author = "Jane Austen", Price = 9.99m}
            };
            dlBooks.DataSource = books;
            dlBooks.DataBind();
        }
    }
    public class Book
    {
        public string Title { get; set; }
        public string Author { get; set; }
        public decimal Price { get; set; }
    }
}
    </form>
</body>
```

```
</html>
```

Output:



Book List

The Great Gatsby

Author: F. Scott Fitzgerald

Price: $12.99

Price: $12.99

Pride and Prejudice

Author: Jane Austen

Price: $9.99

Price: $9.99

To Kill a MockingBird

Author: George Orwell

Price: $11.99

Price: $11.99

Practical No. – 7

AIM : Create A Web Application For Inserting and Deleting Record From A Database(Using Execute-Non Query)

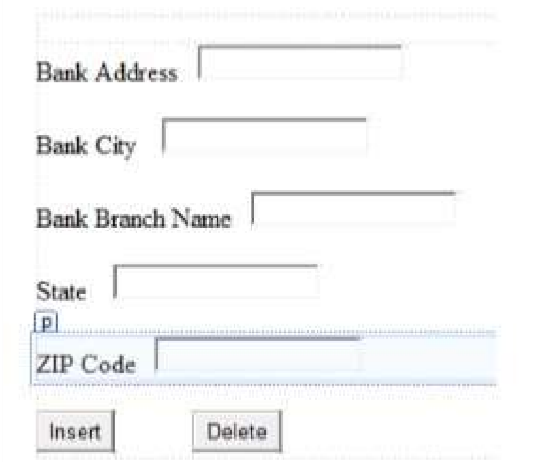
Default.aspx

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

<!DOCTYPE html>

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

<head runat="server">

  <title>Database Operations</title>

</head>

<body>

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

    <div>

      <h2>Insert Record</h2>

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

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

      <asp:Button ID="btnInsert" runat="server" Text="Insert" OnClick="btnInsert_Click" />

      <h2>Delete Record</h2>

      <asp:TextBox ID="txtId" runat="server" placeholder="ID"></asp:TextBox>

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

      <h2>Records</h2>

      <asp:GridView ID="gvRecords" runat="server"
AutoGenerateColumns="true"></asp:GridView>

    </div>

  </form>

</body>

</html>
```

Default.aspx.cs

```
using System;

using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Web.UI;

namespace DatabaseWebApp
{
    public partial class Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGridView();
            }
        }
        protected void btnInsert_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;
            string email = txtEmail.Text;

            string query = "INSERT INTO Users (Name, Email) VALUES (@Name, @Email)";
            ExecuteNonQuery(query, new SqlParameter("@Name", name), new SqlParameter("@Email",
email));
            BindGridView();
            ClearInputs();
        }

        protected void btnDelete_Click(object sender, EventArgs e)
        {
            int id;
            if (int.TryParse(txtId.Text, out id))
            {
                string query = "DELETE FROM Users WHERE Id = @Id";
                ExecuteNonQuery(query, new SqlParameter("@Id", id));

                BindGridView();
                ClearInputs();
            }
        }
        private void ExecuteNonQuery(string query, params SqlParameter[] parameters)
        {
            string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddRange(parameters);
                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }
        }
        private void BindGridView()
        {
            string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
            string query = "SELECT * FROM Users";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    connection.Open();
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dt = new DataTable();
                    adapter.Fill(dt);
                    gvRecords.DataSource = dt;
                    gvRecords.DataBind();
                }
            }
        }
        private void ClearInputs()
        {
            txtName.Text  =  string.Empty;
            txtEmail.Text = string.Empty;
            txtId.Text = string.Empty;
        }
    }
}
```

Output:



Practical No 8

Aim: Create a web application to demonstrate the use of Ajax Controls.

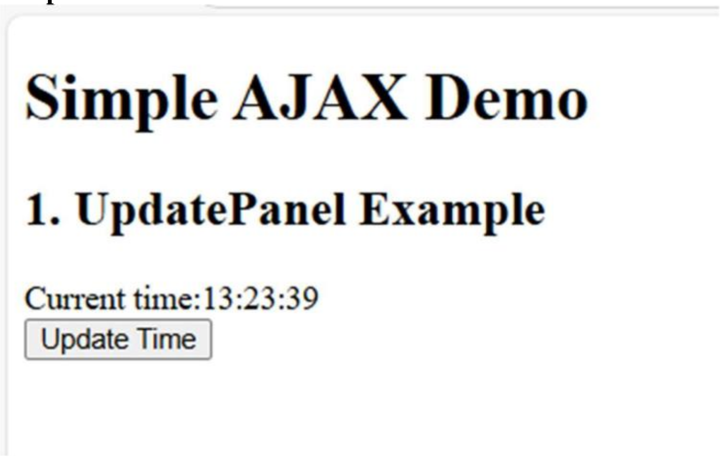
Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication7.WebForm1" %>
<!DOCTYPE html>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Simple AJAX Demo</title>
</head>
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
    <div>
      <h1>Simple AJAX Demo</h1>
      <h2>1. UpdatePanel Example</h2>
      <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
          <asp:Label ID="lblTime" runat="server" Text=""></asp:Label><br />
          <asp:Button ID="btnUpdateTime" runat="server" Text="Update Time"
OnClick="btnUpdateTime_Click" />
        </ContentTemplate>
      </asp:UpdatePanel>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication7

{

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

  {

    protected void Page_Load(object sender, EventArgs e)

    {

    }

    protected void btnUpdateTime_Click(object sender, EventArgs e)

    {

      lblTime.Text = "Current time:" + DateTime.Now.ToString("HH:mm:ss");

    }

    }
```

Output:



Simple AJAX Demo

1. UpdatePanel Example

Current time:13:23:39