

INDEX

Sr.No.	Practicals	Date	Sign
1.	Working with basic C# and ASP .NET		
a.	Create an application that obtains four int values from the user and displays the product.		
b.	Create an application to demonstrate string operations.		
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.		
	Create an application to demonstrate following operations i. Generate Fibonacci series. ii. Test for vowels. iii. Use of foreach loop with arrays.		
2.	Working with Object Oriented C# and ASP .NET		
a.	Create simple application to perform following operation i. Finding factorial Value ii. Money Conversion iii. Quadratic Equation iv. Temperature Conversion		
b.	Create simple application to demonstrate use of following concepts i. Function Overloading ii. Inheritance (all types) . iii. Constructor overloading iv. Interfaces		
c.	Create simple application to demonstrate use of following concepts i. Using Delegates and events ii. Exception handling		
3.	Working with Web Forms and Controls		
a.	Create a simple web page with various sever controls to demonstrate setting and use of their properties. (Example : AutoPostBack)		
b.	Demonstrate the use of Calendar control to perform following operations. a) Display messages in a calendar control b) Display vacation in a calendar control c) Selected day in a calendar control using style		

	d) Difference between two calendar dates		
4.	Working with Form Controls		
a.	Create a Registration form to demonstrate use of various Validation controls.		
b.	Create Web Form to demonstrate use of Adrotator Control.		
c.	Create Web Form to demonstrate use User Controls.		
5.	Working with Navigation, Beautification and Master page.		
a.	Create Web Form to demonstrate use of Website Navigation controls and Site Map.		
6.	Working with Database		
a.	Create a web application bind data in a multiline textbox by querying in another textbox.		
b.	Create a web application to display records by using database.		
c.	Demonstrate the use of Datalist link control.		
7.	Working with Database		
a.	Create a web application to display Databinding using dropdownlist control.		
b.	Create a web application for to display the phone no of an author using database.		
c.	Create a web application for inserting and deleting record from a database. (Using Execute-Non Query).		
8.	Working with data controls		
a.	Create a web application to demonstrate various uses and properties of SqlDataSource.		
b.	Create a web application to demonstrate data binding using DetailsView and FormView Control.		
c.	Create a web application to display Using Disconnected Data Access and Databinding using GridView.		

PRACTICAL NO-1

1. Working with basic C# and ASP .NET

- a. Create an application that obtains four int values from the user and displays the product.
- b. Create an application to demonstrate string operations.
- 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.
- d. Create an application to demonstrate following operations
 - i. Generate Fibonacci series.
 - ii. Test for vowels.
 - iii. Use of foreach loop with arrays.
- a. Create an application that obtains four int values from the user and displays the product.

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace pract1_a
{
    class Product
    {
        static void Main(string[] args)
        {
            int a, b, c, d, product;
            Console.WriteLine("Enter four integers");
            a = Convert.ToInt32(Console.ReadLine());
            b = Convert.ToInt32(Console.ReadLine());
            c = Convert.ToInt32(Console.ReadLine());
            d = Convert.ToInt32(Console.ReadLine());
            product = a * b * c * d;
            Console.WriteLine("Product="+product);
            Console.ReadKey();
        }
    }
}
```

```
}
```

Step2:-

 file:///C:/Users/admin/Documents/Visual Studio 2012/Projects/pract1-a/pract1-a/bin/Debug/pract1-a.EXE

```
Enter four integers
```

```
2
```

```
3
```

```
1
```

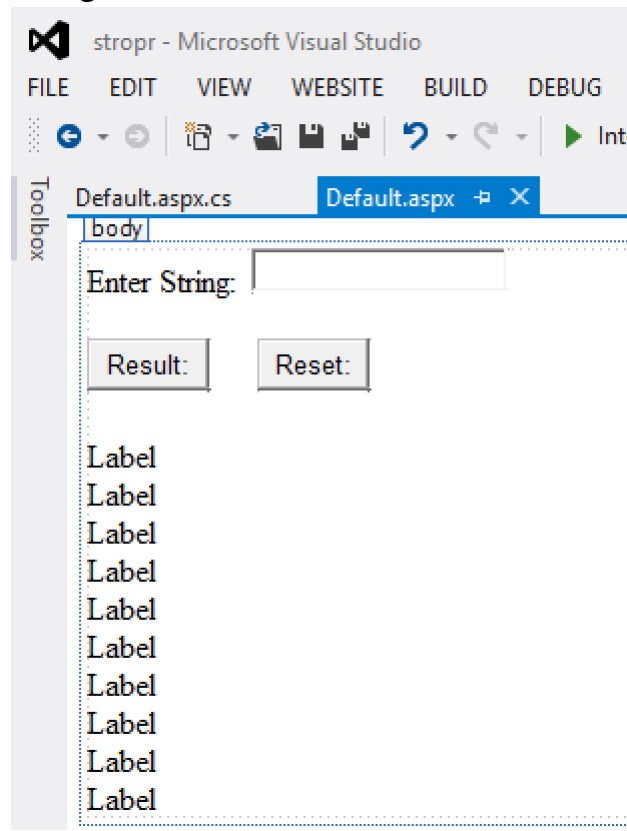
```
4
```

```
Product=24
```

- b. Create an application to demonstrate string operations.

Step1:-

Design View



Default.aspx

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

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

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

```
&nbsp;<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

```
<br />
```

```
<br />
```

```
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click"
```

```
Text="Result:" />
```

```
&nbsp;<asp:Button ID="Button2" runat="server" OnClick="Button2_Click"
Text="Reset:" />
<br />
<br />
<asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label5" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label6" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label7" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label8" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label9" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label10" runat="server" Text="Label"></asp:Label>
<br />
<asp:Label ID="Label11" runat="server" Text="Label"></asp:Label>

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

Default.aspx.cs

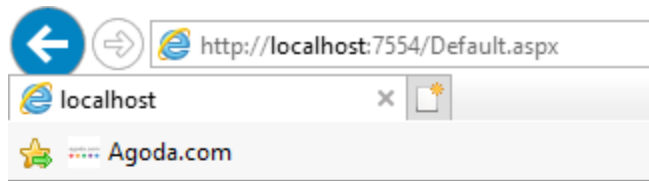
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

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

```
{
    string s = TextBox1.Text;
    Label2.Text = "String Length: "+s.Length;
    Label3.Text = "Substring: " + s.Substring(4,3);
    Label4.Text = "Upper String: " + s.ToUpper();
    Label5.Text = "Lower String: " + s.ToLower();
    string rev = "";
    for (int i = s.Length - 1; i >= 0; i--)
    {
        rev = rev + s[i];
    }
    Label6.Text = "Reverse String: " + rev.ToString();
    Label7.Text = "Replace 's' by 't' in String: " + s.Replace('s','t');
    Label8.Text = "Insert 'u' in String: " + s.Insert(3,"u");
    Label9.Text = "String Truncate: " + s.Trim();
    Label10.Text = "Remove String: " + s.Remove(4);
    Label11.Text = "Index of String: " + s.IndexOf('e');
}
protected void Button2_Click(object sender, EventArgs e)
{
    Label1.Text = "";
    Label2.Text = "";
    Label3.Text = "";
    Label4.Text = "";
    Label5.Text = "";
    Label6.Text = "";
    Label7.Text = "";
    Label8.Text = "";
    Label9.Text = "";
    Label10.Text = "";
    TextBox1.Text = "";
}
}
```

Output:-



Enter String:

String Length: 9
Substring: haw
Upper String: SHEKHAWAT
Lower String: shekhawat
Reverse String: tawahkehS
Replace 's' by 't' in String: Shekhawat
Insert 'u' in String: Sheukhawat
String Truncate: Shekhawat
Remove String: Shek
Index of String: 2

- 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.

Step1:-

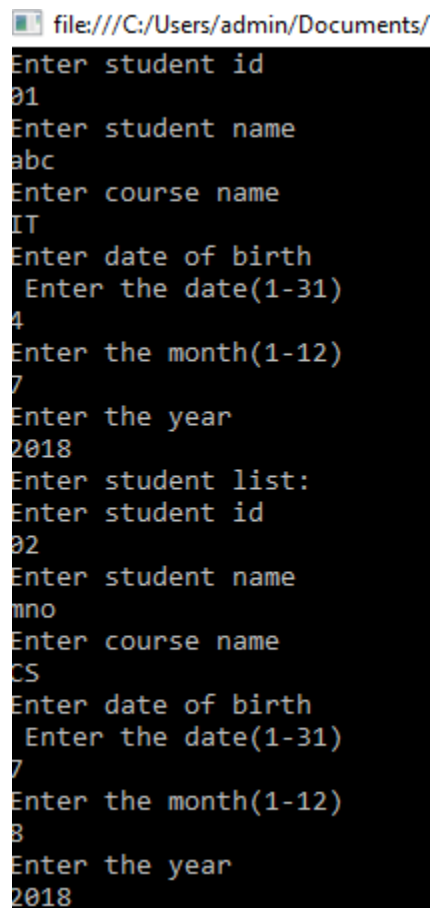
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace student
{

class Program
{
    struct stud
    {
        public String stud_name, stud_id, c_name;
        public int date, month, year;
    }
    static void Main(string[] args)
    {
        stud[] s = new stud[5];
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine("Enter student id");
            s[i].stud_id = Console.ReadLine();
            Console.WriteLine("Enter student name");
            s[i].stud_name = Console.ReadLine();
            Console.WriteLine("Enter course name");
            s[i].c_name = Console.ReadLine();
            Console.WriteLine("Enter date of birth \n Enter the date(1- 31)");
            s[i].date = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter the month(1-12)");
            s[i].month = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter the year");
            s[i].year = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter student list:");
        }
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine("Student ID=" + s[i].stud_id);
            Console.WriteLine("Student name=" + s[i].stud_name);
            Console.WriteLine("Course name=" + s[i].c_name);
        }
    }
}
```

```
        Console.WriteLine("Birth date=" + s[i].date + "-" + s[i].month + "-" +  
        s[i].year);  
    }  
    Console.ReadKey();  
}  
}
```

Step2:-



```
file:///C:/Users/admin/Documents/  
Enter student id  
01  
Enter student name  
abc  
Enter course name  
IT  
Enter date of birth  
Enter the date(1-31)  
4  
Enter the month(1-12)  
7  
Enter the year  
2018  
Enter student list:  
Enter student id  
02  
Enter student name  
mno  
Enter course name  
CS  
Enter date of birth  
Enter the date(1-31)  
7  
Enter the month(1-12)  
3  
Enter the year  
2018
```

```
Enter student list:
Enter student id
03
Enter student name
pqr
Enter course name
IT
Enter date of birth
  Enter the date(1-31)
5
Enter the month(1-12)
9
Enter the year
2018
Enter student list:
Enter student id
04
Enter student name
xyz
Enter course name
IT
Enter date of birth
  Enter the date(1-31)
5
Enter the month(1-12)
6
Enter the year
2018
```

```
Enter student list:
Enter student id
05
Enter student name
rst
Enter course name
IT
Enter date of birth
  Enter the date(1-31)
  7
Enter the month(1-12)
  3
Enter the year
  2018
Enter student list:
Student ID=01
Student name=abc
Course name=IT
Birth date=4-7-2018
Student ID=02
Student name=mno
Course name=CS
Birth date=7-8-2018
Student ID=03
Student name=pqr
Course name=IT
Birth date=5-9-2018
Student ID=04
Student name=xyz
Course name=IT
Birth date=5-6-2018
Student ID=05
Student name=rst
Course name=IT
Birth date=7-3-2018
```

d. Create an application to demonstrate following operations

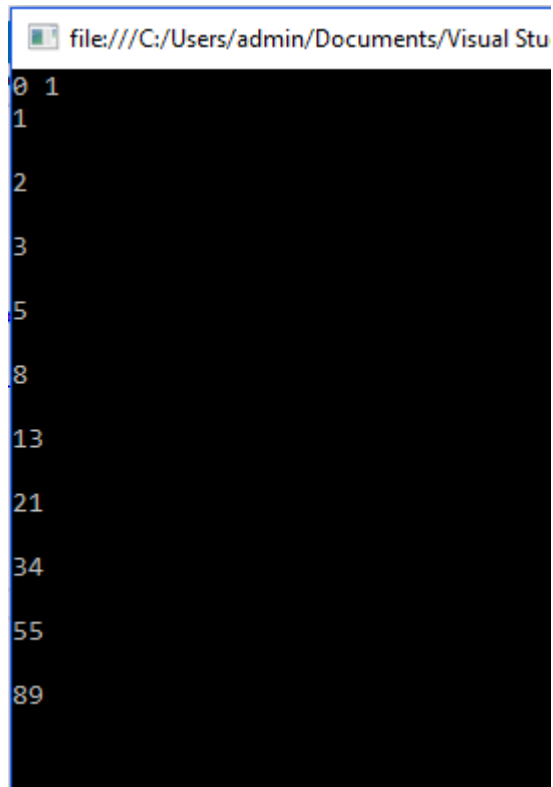
i. Generate Fibonacci series.

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace fibonacci
{
    class Program
    {
        static void Main(string[] args)
        {
            int num1 = 0;
            int num2 = 1;
            int num3;
            Console.WriteLine(num1 + " " + num2);
            for (int i = 0; i < 10; i++)
            {
                num3 = num1 + num2;
                num1 = num2;
                num2 = num3;
                Console.WriteLine(num3);
                Console.WriteLine(" ");
            }
            Console.ReadKey();
        }
    }
}
```

Step2:-



The image shows a screenshot of a Visual Studio Code window. The title bar at the top reads "file:///C:/Users/admin/Documents/Visual Stu". The main editor area has a dark background and displays a list of numbers in a light blue font. The numbers are: 0 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, and 89. These numbers are arranged in a single column, with "0 1" on the first line and the others following on subsequent lines.

ii. Test for vowels.

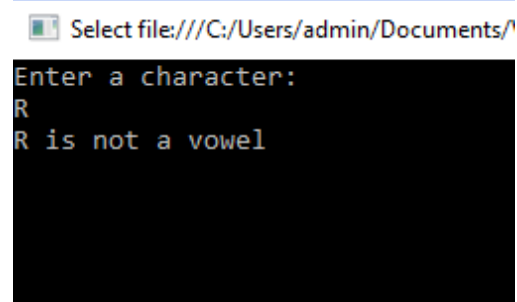
Step1:-

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace vowels  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            int a;
```

```
char ch;
Console.WriteLine("Enter a character:");
a = Console.Read();
ch = Convert.ToChar(a);
switch (ch)
{
    case 'a': Console.WriteLine(ch + " is a vowel");
              break;
    case 'e': Console.WriteLine(ch + " is a vowel");
              break;
    case 'i': Console.WriteLine(ch + " is a vowel");
              break;
    case 'o': Console.WriteLine(ch + " is a vowel");
              break;
    case 'u': Console.WriteLine(ch + " is a vowel");
              break;
    case 'A': Console.WriteLine(ch + " is a vowel");
              break;
    case 'E': Console.WriteLine(ch + " is a vowel");
              break;
    case 'I': Console.WriteLine(ch + " is a vowel");
              break;
    case 'O': Console.WriteLine(ch + " is a vowel");
              break;
    case 'U': Console.WriteLine(ch + " is a vowel");
              break;
    default: Console.WriteLine(ch + " is not a vowel");
             break;
}
Console.ReadKey();
}
```

Step2:-




iii. Use of foreach loop with arrays.

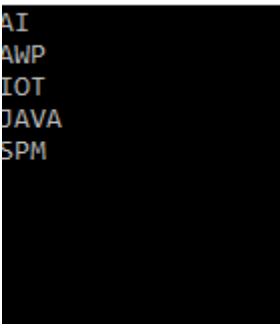
Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace foreachloop
{
    class Program
    {
        static void Main(string[] args)
        {
            String[] str = { "AI", "AWP", "IOT", "JAVA", "SPM" };
            foreach(String s in str)
            {
                Console.WriteLine(s);
            }
            Console.ReadKey();
        }
    }
}
```

Step2:-

 file:///C:/Users/admin,



```
AI
AWP
IOT
JAVA
SPM
```

PRACTICAL NO-2

2. Working with Object Oriented C# and ASP .NET

- a. Create simple application to perform following operations
 - i. Finding factorial Value
 - ii. Money Conversion
 - iii. Quadratic Equation
 - iv. Temperature Conversion

- b. Create simple application to demonstrate use of following concepts
 - i. Function Overloading
 - ii. Inheritance (all types)
 - iii. Constructor overloading
 - iv. Interfaces

- c. Create simple application to demonstrate use of following concepts
 - i. Using Delegates and events
 - ii. Exception handling

a. Create simple application to perform following operations

i. Finding factorial Value

Solution:-


Program.cs

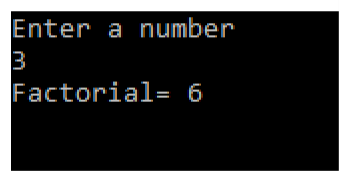
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace fact
{
    class Program
    {
        static void Main(string[] args)
        {
            int num, i, fact=1;
            Console.WriteLine("Enter a number");
            num = Convert.ToInt32(Console.ReadLine());

            for (i = 1; i <= num; i++)
            {
                fact = fact * i;
            }
            Console.WriteLine("Factorial= "+fact);
            Console.ReadKey();
        }
    }
}
```

Output:-

 file:///c:/users/admin/do



```
Enter a number
3
Factorial= 6
```

ii. Money Conversion

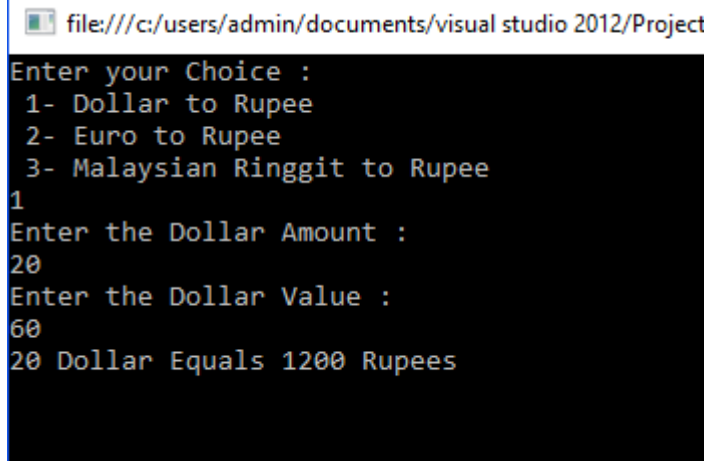
Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace moneyconversion
{
    class Program
    {
        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, val;
                    Console.WriteLine("Enter the Dollar Amount :");
                    dollar = Double.Parse(Console.ReadLine());
                    Console.WriteLine("Enter the Dollar Value :");
                    val = double.Parse(Console.ReadLine());
                    rupee = dollar * val;
                    Console.WriteLine("{0} Dollar Equals {1} Rupees", dollar, rupee);
                    break;
                case 2:
                    Double Euro, rupe, valu;
                    Console.WriteLine("Enter the Euro Amount :");
                    Euro = Double.Parse(Console.ReadLine());
                    Console.WriteLine("Enter the Euro Value :");
                    valu = double.Parse(Console.ReadLine());
                    rupe = Euro * valu;
                    Console.WriteLine("{0} Euro Equals {1} Rupees", Euro, rupe);
                    break;
                case 3:
                    Double ringit, rup, value;
                    Console.WriteLine("Enter the Ringgit Amount :");
                    ringit = Double.Parse(Console.ReadLine());
                    Console.WriteLine("Enter the Ringgit Value :");
                    value = double.Parse(Console.ReadLine());
                    rup = ringit * value;
```

```
        Console.WriteLine("{0} Malaysian Ringgit Equals {1} Rupees", ringit,
rup);
        break;
    }
    Console.ReadLine();
}
}
```

Output:-



The screenshot shows a Visual Studio console window with the following text:

```
file:///c:/users/admin/documents/visual studio 2012/Project
Enter your Choice :
1- Dollar to Rupee
2- Euro to Rupee
3- Malaysian Ringgit to Rupee
1
Enter the Dollar Amount :
20
Enter the Dollar Value :
60
20 Dollar Equals 1200 Rupees
```

iii. Quadratic Equation

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace quadequation
{
    class Quadraticroots
    {
        double a, b, c;

        public void read()
        {
            Console.WriteLine(" \n To find the roots of a quadratic equation of the
form  $a*x*x + b*x + c = 0$ ");
            Console.Write("\n Enter value for a : ");
            a = double.Parse(Console.ReadLine());
            Console.Write("\n Enter value for b : ");
            b = double.Parse(Console.ReadLine());
            Console.Write("\n Enter value for c : ");
            c = double.Parse(Console.ReadLine());
        }

        public void compute()
        {
            int m;
            double r1, r2, d1;
            d1 = b * b - 4 * a * c;
            if (a == 0)
                m = 1;
            else if (d1 > 0)
                m = 2;
            else if (d1 == 0)
                m = 3;
            else
                m = 4;
            switch (m)
            {
                case 1: Console.WriteLine("\n Not a Quadratic equation, Linear
equation");
                    Console.ReadLine();
                    break;
                case 2: Console.WriteLine("\n Roots are Real and Distinct");
                    r1 = (-b + Math.Sqrt(d1)) / (2 * a);
```

```
        r2 = (-b - Math.Sqrt(d1)) / (2 * a);
        Console.WriteLine("\n First root is {0:###.###}", r1);
        Console.WriteLine("\n Second root is {0:###.###}", r2);
        Console.ReadLine();
        break;
    case 3: Console.WriteLine("\n Roots are Real and Equal");
        r1 = r2 = (-b) / (2 * a);
        Console.WriteLine("\n First root is {0:###.###}", r1);
        Console.WriteLine("\n Second root is {0:###.###}", r2);
        Console.ReadLine();
        break;
    case 4: Console.WriteLine("\n Roots are Imaginary");
        r1 = (-b) / (2 * a);
        r2 = Math.Sqrt(-d1) / (2 * a);
        Console.WriteLine("\n First root is {0:###.###} + i {1:###.###}", r1, r2);
        Console.WriteLine("\n Second root is {0:###.###} - i {1:###.###}", r1, r2);
        Console.ReadLine();
        break;
    }
}
}
}
class Roots
{
    static void Main(string[] args)
    {
        Quadraticroots qr = new Quadraticroots();
        qr.read();
        qr.compute();
    }
}
}
```

Output:-

file:///c:/users/admin/documents/visual studio 2012/Projects/quadequation/quadequation/bin/Debug/quadequation.exe

```
To find the roots of a quadratic equation of the form  $a*x*x + b*x + c = 0$ 
Enter value for a : 3.5
Enter value for b : 2.5
Enter value for c : 1.0
Roots are Imaginary
First root is  $-.36 + i .4$ 
Second root is  $-.36 - i .4$ 
```

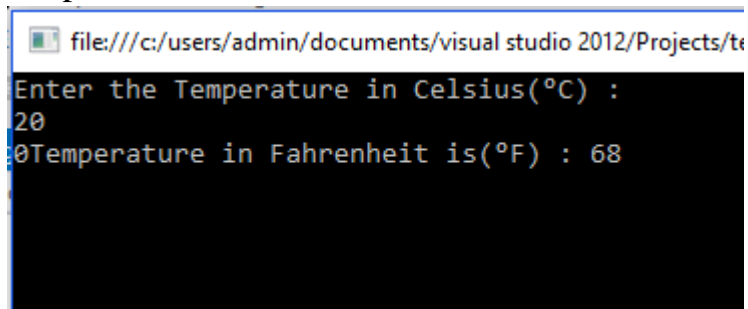
iv. Temperature Conversion

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace temperature
{
    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("Temperature in Fahrenheit is(°F) : " + faren);
            Console.ReadLine();
        }
    }
}
```


Output:-



```
file:///c:/users/admin/documents/visual studio 2012/Projects/te
Enter the Temperature in Celsius(°C) : 20
Temperature in Fahrenheit is(°F) : 68
```

b. Create simple application to demonstrate use of following concepts

i. Function Overloading

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

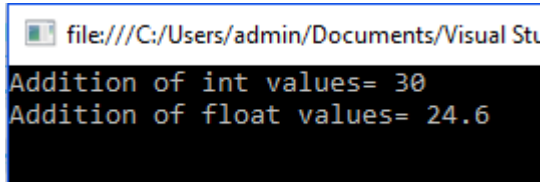
namespace methooverloading
{
    class Program
    {
        public void addition(int a, int b)
        {
            Console.WriteLine("Addition of int values= "+(a+b));
        }
        public void addition(float a, float b)
        {
            Console.WriteLine("Addition of float values= "+(a+b));
        }
        static void Main(string[] args)
        {
            Program p=new Program();
            p.addition(10,20);
        }
    }
}
```

```

        p.addition(11.2f,13.4f);
        Console.ReadKey();
    }
}

```

Step2:-



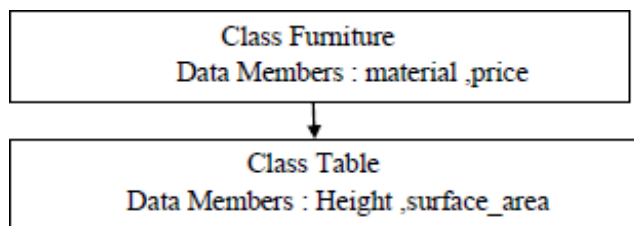
```

file:///C:/Users/admin/Documents/Visual Stu
Addition of int values= 30
Addition of float values= 24.6

```

ii. Inheritance (all types)

a. Single Inheritance



Step1:-

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace singleinherit
{
    class furniture
    {
        string material;
        float price;
        public void getdata()
        {
            Console.WriteLine("Enter material name:");
            material = Console.ReadLine();
            Console.WriteLine("Enter price:");
            price = float.Parse(Console.ReadLine());
        }
    }
}

```

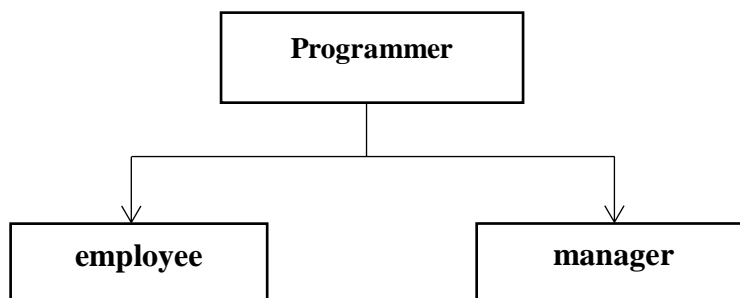
```
        public void displaydata()
        {
            Console.WriteLine("\nMaterial=" + material);
            Console.WriteLine("\nPrice=" + price);
        }
    }
    class table : furniture
    {
        int height, surface_area;
        public void accept()
        {
            base.getdata();
            Console.WriteLine("Enter height:");
            height = Int32.Parse(Console.ReadLine());
            Console.WriteLine("Enter surface area");
            surface_area = Int32.Parse(Console.ReadLine());
        }
        public void display()
        {
            base.displaydata();
            Console.WriteLine("\nHeight=" + height);
            Console.WriteLine("Surface area=" + surface_area);
        }
    }
    class sample
    {
        static void Main(string[] args)
        {
            table obj = new table();
            obj.accept();
            obj.display();
            Console.ReadKey();
        }
    }
}
```

Step2:-

```
file:///C:/Users/admin/Document
Enter material name:
chair
Enter price:
45000
Enter height:
12
Enter surface area
10

Material=chair
Price=45000
Height=12
Surface area=10
```

b. Hierarchical Inheritance



Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace hierarchicalinherit
{
    class employee
    {
        public void display()
        {
            Console.WriteLine("This is Employee class");
        }
    }
    class programmer : employee
    {
        public void display()
        {

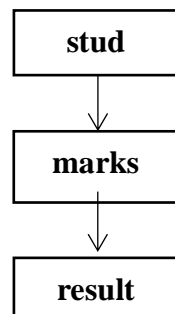
```

```
        Console.WriteLine("This is Programmer class");
    }
}
class manager : employee
{
    public void display()
    {
        Console.WriteLine("This is Manager class");
    }
}
class sample
{
    static void Main(string[] args)
    {
        Console.WriteLine("Whose information you want to see: \n 1.Programmer
\n 2.Manager");
        int ch = Int32.Parse(Console.ReadLine());
        if (ch == 1)
        {
            programmer p=new programmer();
            p.display();
        }
        else if (ch == 2)
        {
            manager m=new manager();
            m.display();
        }
        else
        {
            Console.WriteLine("Enter correct choice");
        }
        Console.ReadKey();
    }
}
```

Output:-

```
This is Employee class  
This is Employee class  
Whose information you want to see: 1.programmer 2.manager  
1  
This is Programmer class
```

d. Multilevel Inheritance



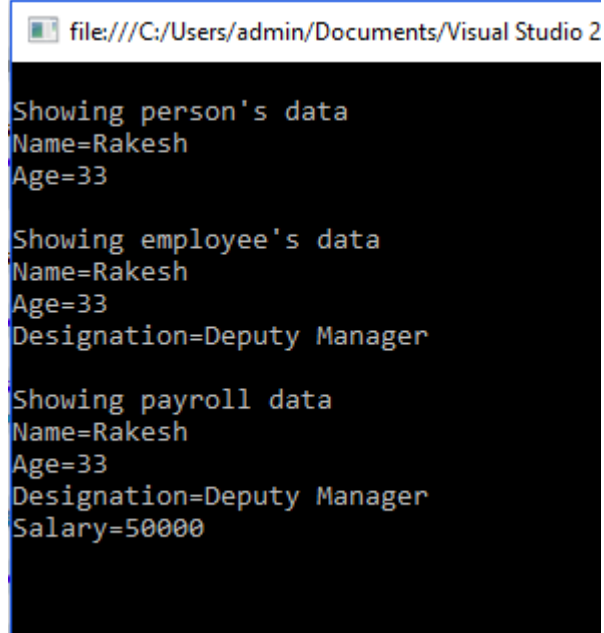
Step1:-

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace multilevel  
{  
    class person  
    {  
        int age;  
        String name;  
        public person(String s, int a)  
        {  
            name = s;  
            age = a;  
        }  
        public void show()  
        {  
            Console.WriteLine("Name=" + name);  
        }  
    }  
}
```

```
        Console.WriteLine("Age=" + age);
    }
}
class employee : person
{
    String designation;
    public employee(String s, int a, String d)
        : base(s, a)
    {
        designation = d;
    }
    public void show()
    {
        base.show();
        Console.WriteLine("Designation=" + designation);
    }
}
class payroll : employee
{
    int salary;
    public payroll(String s, int a, String d, int sal)
        : base(s, a, d)
    {
        salary = sal;
    }
    public void show()
    {
        base.show();
        Console.WriteLine("Salary=" + salary);
    }
}
class program
{
    static void Main(string[] args)
    {
        Console.WriteLine("\nShowing person's data");
        person p = new person("Rakesh",33);
        p.show();
        Console.WriteLine("\nShowing employee's data");
        employee e = new employee("Rakesh",33,"Deputy Manager");
        e.show();
        Console.WriteLine("\nShowing payroll data");
        payroll p1=new payroll("Rakesh",33,"Deputy Manager",50000);
        p1.show();
        Console.ReadKey();
    }
}
```

```
}
```

Step2:-



```
file:///C:/Users/admin/Documents/Visual Studio 2012/Projects/1/1/obj/Debug/1.exe
Showing person's data
Name=Rakesh
Age=33

Showing employee's data
Name=Rakesh
Age=33
Designation=Deputy Manager

Showing payroll data
Name=Rakesh
Age=33
Designation=Deputy Manager
Salary=50000
```

iii. Constructor overloading

Step1:-

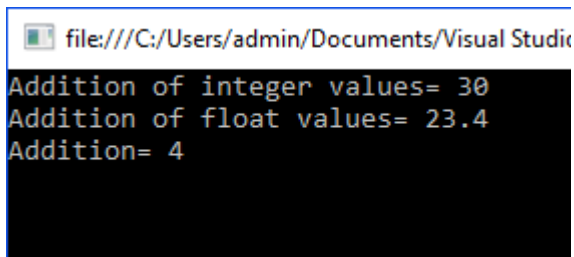
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace constructoverloading
{
    class construct
    {
        int x;
        public construct(int a, int b)
        {
```



```
        Console.WriteLine("Addition of integer values= "+(a+b));
    }
    public construct(float a, float b)
    {
        Console.WriteLine("Addition of float values= "+(a+b));
    }
    public construct()
    {
        x = 2;
        Console.WriteLine("Addition= "+(x+x));
    }
}
class program
{
    static void Main(string[] args)
    {
        construct obj1 = new construct(10,20);
        construct obj2 = new construct(11.2f,12.2f);
        construct obj3 = new construct();
        Console.ReadKey();
    }
}
```

Step2:-

A screenshot of a console window with a black background and white text. The window title bar shows the file path: file:///C:/Users/admin/Documents/Visual Studio... The output text is as follows:
Addition of integer values= 30
Addition of float values= 23.4
Addition= 4

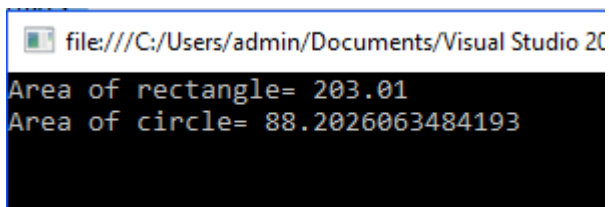
iv. Interfaces

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace interfaces
{
    interface rect
    {
        void calculate1(float x, float y);
    }
    class circle
    {
        public void calculate2(float a)
        {
            Console.WriteLine("Area of circle= " + 3.14 * a * a);
        }
    }
    class shape : circle, rect
    {
        public void calculate1(float x, float y)
        {
            Console.WriteLine("Area of rectangle= " + x * y);
        }
    }
    class sample
    {
        static void Main(string[] args)
        {
            shape obj = new shape();
            obj.calculate1(10.1f, 20.1f);
            obj.calculate2(5.3f);
            Console.ReadKey();
        }
    }
}
```

Step2:-



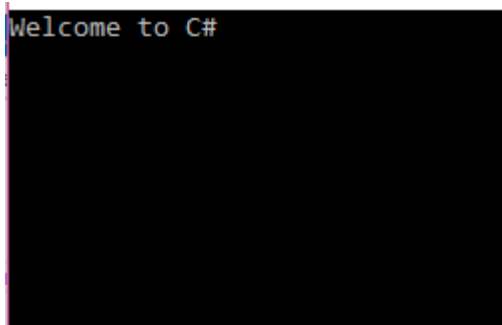
```
file:///C:/Users/admin/Documents/Visual Studio 20...
Area of rectangle= 203.01
Area of circle= 88.2026063484193
```

- c. Create simple application to demonstrate use of following concepts
i. Using Delegates and events

Solution:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace simple
{
    public delegate void dele();
    class Program
    {
        public static void display()
        {
            Console.WriteLine("Welcome to C#");
        }
        static void Main(string[] args)
        {
            dele d1 = new dele(display);
            d1();
            Console.ReadKey();
        }
    }
}
```

OUTPUT:

A screenshot of a console application window with a black background. The text "Welcome to C#" is displayed in a light blue or cyan monospaced font at the top left of the window.

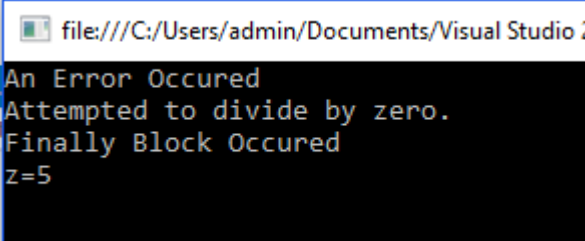
ii. Exception handling

Step1:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace except
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 10, y = 0, z;
            try
            {
                z = x / y;
            }
            catch (Exception e)
            {
                Console.WriteLine("An Error Occured\n");
                Console.WriteLine(e.Message);
            }
            finally
            {
                Console.WriteLine("\nFinally Block Occured");
                y = 2;
                z = x / y;
                Console.WriteLine("\nz=" + z);
            }
            Console.ReadKey();
        }
    }
}
```

Step2:-



```
file:///C:/Users/admin/Documents/Visual Studio 2
An Error Occured
Attempted to divide by zero.
Finally Block Occured
z=5
```

PRACTICAL NO-3

3. Working with Web Forms and Controls

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

b. Demonstrate the use of Calendar control to perform following operations.

- a) Display messages in a calendar control
- b) Display vacation in a calendar control
- c) Selected day in a calendar control using style
- d) Difference between two calendar dates

c. Demonstrate the use of Treeview control perform following operations.

- a) Treeview control and datalist
- b) Treeview operations

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

Step1:-

Design View

Name:

Address:

Gender:

☐ Male ☐ Female

Subjects:

☐ AI

☐ IOT

☐ SPM

☐ JAVA

☐ AWP

Vehicles:

☐ BUS

☐ CAR

☐ AUTO

Fruits:

Apple ▼

Display

Result:

Default.aspx

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

<!DOCTYPE html>

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

[illegible]

```

        <br />
        <asp:Button ID="Button1" runat="server" Text="Display"
OnClick="Button1_Click1" />
        <br />

        <asp:Label ID="lblresult" runat="server" Text="Result: "></asp:Label>
        <br />
    </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;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
        lblresult.Text += "Name:" + txtname.Text + "<br>" + "Address:" +
txtaddress.Text + "<br>";

        if (rbmale.Checked == true)
            lblresult.Text += "Gender:" + rbmale.Text + "<br>";
        else
            lblresult.Text += "Gender:" + rbfemale.Text + "<br>";

        for (int i = 0; i < RadioButtonList1.Items.Count; i++)
        {
            if (RadioButtonList1.Items[i].Selected)
                lblresult.Text += "Subjects:" + RadioButtonList1.Text + "<br>";
        }

        for (int i = 0; i < CheckBoxList1.Items.Count; i++)
        {
            if (CheckBoxList1.Items[i].Selected)

```



```
        lblresult.Text += "Vehicles:" + CheckBoxList1.Text + "</br>";  
    }  
  
    for (int i = 0; i < DropDownList1.Items.Count; i++)  
    {  
        if (DropDownList1.Items[i].Selected)  
            lblresult.Text += "Fruits:" + DropDownList1.Text + "</br>";  
    }  
}
```

Output:-

Name:

Address:

Gender:
☐ Male ☒ Female

Subjects:
☒ AI
☐ IOT
☐ SPM
☐ JAVA
☐ AWP

Vehicles:
☐ BUS
☒ CAR
☐ AUTO

Fruits:

Result: Name:Rishika
Address:Bhiwandi
Gender:Female
Subjects:AI
Vehicles:CAR
Fruits:Mango

b. Demonstrate the use of Calendar control to perform following operations.

- a) Display messages in a calendar control
- b) Display vacation in a calendar control
- c) Selected day in a calendar control using style
- d) Difference between two calendar dates

Step1:-

Design View

div

October 2018						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Label

Label

Label

Label

Label

Result:

Reset:

Default.aspx

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

```
<!DOCTYPE html>
```

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

```

</head>
<body>
  <form id="form1" runat="server">
    <div>

      <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
      <br />
      &nbsp;<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label3" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label4" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Label ID="Label5" runat="server" Text="Label"></asp:Label>
      <br />
      <br />
      <asp:Button ID="btnresult" runat="server" OnClick="btn_result_Click"
Text="Result:" />
      <br />
      <br />
      <br />
      <asp:Button ID="btnreset" runat="server" OnClick="btnreset_Click"
Text="Reset:" />
      <br />

    </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;

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

```

```

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

        }

        protected void btn_result_Click(object sender, EventArgs e)
        {
            Calendar1.Caption = "T.Y.IT";
            Calendar1.FirstDayOfWeek = FirstDayOfWeek.Sunday;
            Calendar1.NextPrevFormat = NextPrevFormat.ShortMonth;
            Calendar1.TitleFormat = TitleFormat.Month;

            Label2.Text = "Todays Date" + Calendar1.TodaysDate.ToShortDateString();
            Label3.Text = "Ganpati Vacation Start: 10-13-2021";
            TimeSpan d = new DateTime(2021, 10, 13) - DateTime.Now;
            Label4.Text = "Days Remaining For Ganpati Vacation:" + d.Days.ToString();
            TimeSpan d1 = new DateTime(2021, 12, 31) - DateTime.Now;
            Label5.Text = "Days Remaining for New Year:" + d1.Days.ToString();
            if (Calendar1.SelectedDate.ToShortDateString() == "10-13-2021")
                Label3.Text = "<b>Ganpati Festival Start</b>";
            if (Calendar1.SelectedDate.ToShortDateString() == "10-23-2021")
                Label3.Text = "<b>Ganpati Festival End</b>";
        }

        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)
    {
        Label11.Text = "your selected date " + Calendar1.SelectedDate.Date.ToString();
    }

    protected void Calendar1_DayRender(object sender, 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);
        }

        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);  
    }  
}
```

Output:-

[←](#) [→](#) [↻](#) [i](#) localhost:2326/Default.aspx

Rishika

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

Welcome to Calandar

Today's date26/10/2018

Ganpati Vacation Start: 9-13-2018

Days remaining for ganpati vacation:-43

Days remaining for new year:65

PRACTICAL NO-4

4. Working with Form Controls

- a. Create a Registration form to demonstrate use of various Validation controls.
- b. Create Web Form to demonstrate use of Adrotator Control.
- c. Create Web Form to demonstrate use User Controls.

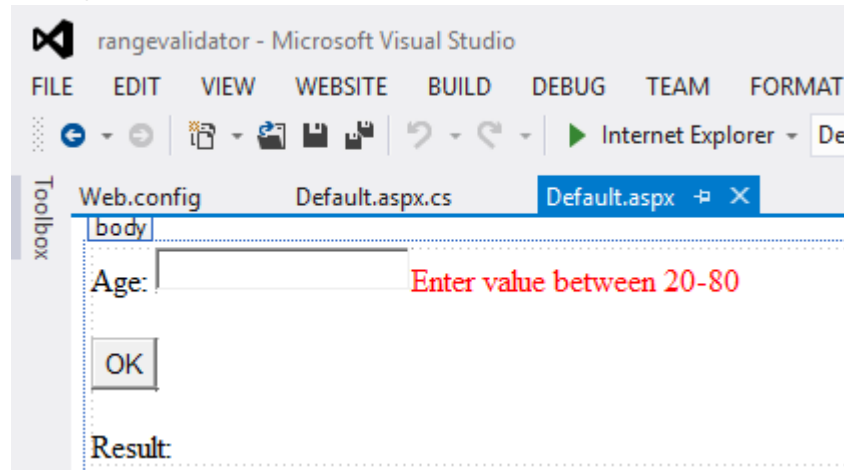
a. Create a Registration form to demonstrate use of various Validation controls.

Step1:-

1 RangeValidator

Solution:-

Design View



Default.aspx

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

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

```
<asp:Label ID="Label1" runat="server" Text="Age: "></asp:Label>
```

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

```
<asp:RangeValidator ID="RangeValidator1" runat="server"
```

```
ControlToValidate="TextBox1" ErrorMessage="Enter value between 20-80"
```

```
ForeColor="Red" MaximumValue="80" MinimumValue="20"></asp:RangeValidator>
```

```
<br />
```

```
<br />
```

```
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="OK" />
```

```
<br />
```

```
<br />
```

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



```
</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;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

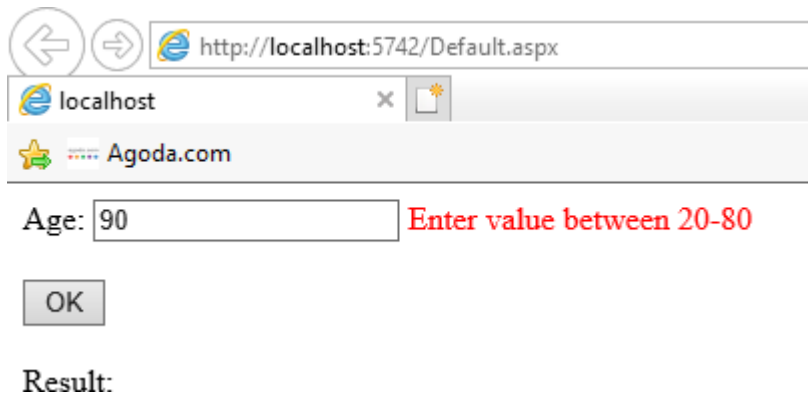
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label2.Text = TextBox1.Text;
    }
}
```

Web.config

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.5"/>
    <httpRuntime targetFramework="4.5"/>
  </system.web>
</configuration>
```

Output:-

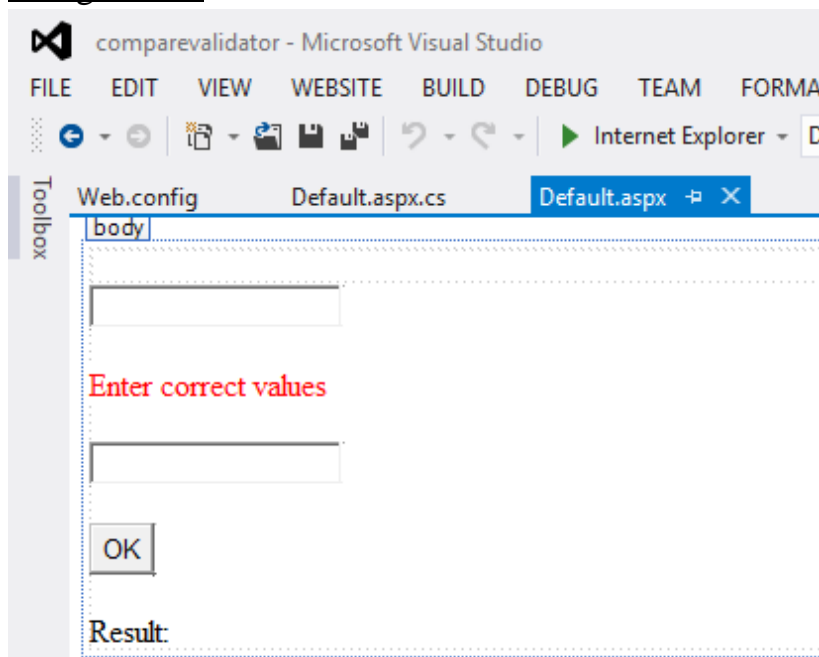


A screenshot of a web browser window. The address bar shows 'http://localhost:5742/Default.aspx'. The browser has a single tab titled 'localhost'. Below the address bar, there is a search bar with 'Agoda.com' entered. The main content area displays a form with the label 'Age:' followed by a text input field containing the value '90'. To the right of the input field, a red error message reads 'Enter value between 20-80'. Below the input field is an 'OK' button. At the bottom of the form, the text 'Result:' is visible.

2 Compare Validator

Solution:-

Design View



Default.aspx

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

<!DOCTYPE html>

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

            </div>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:CompareValidator ID="CompareValidator1" runat="server"
BackColor="White" ControlToCompare="TextBox1" ControlToValidate="TextBox2"
ErrorMessage="Enter correct values" ForeColor="Red"></asp:CompareValidator>
            <br />
            <br />
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="OK" />
            <br />
            <br />
            <asp:Label ID="Label1" runat="server" Text="Result:"></asp:Label>
        </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;

public partial class _Default : System.Web.UI.Page
{
```

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

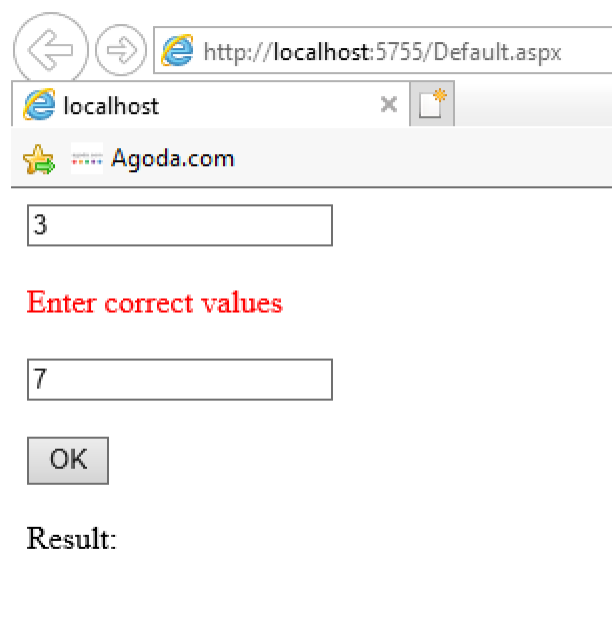
}

protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = TextBox2.Text;
}
```

Web.config

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.5"/>
    <httpRuntime targetFramework="4.5"/>
  </system.web>
</configuration>
```

Output:-



3

Enter correct values

7

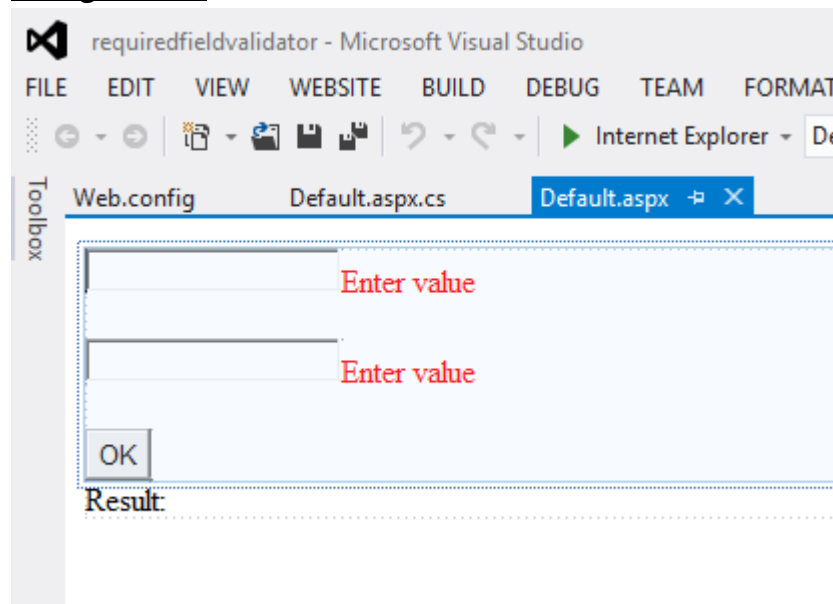
OK

Result:

3 Required Field Validator

Solution:-

Design View



Default.aspx

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

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

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

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

```
ControlToValidate="TextBox1" ErrorMessage="Enter value"
```

```
ForeColor="Red"></asp:RequiredFieldValidator>
```

```
<br />
```

```
<br />
```

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

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="Enter value"
ForeColor="Red"></asp:RequiredFieldValidator>
<br />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="OK" />
<br />

</div>
<asp:Label ID="Label1" runat="server" Text="Result:"></asp:Label>
</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;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        Label1.Text = TextBox2.Text;
    }
}
```

Web.config

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
```

```

</appSettings>
<system.web>
  <compilation debug="true" targetFramework="4.5"/>
  <httpRuntime targetFramework="4.5"/>
</system.web>
</configuration>

```

Output:-

17

Enter value

OK

Result:

4 Regular Expression Validator

Solution:-

Design View

regularexpvalidator - Microsoft Visual Studio

FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM FORMAT

Internet Explorer Debug

Web.config Default.aspx.cs Default.aspx

body

Enter 10 digit mobile number

OK

Result:

Default.aspx

[illegible]

Default.aspx.cs

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

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
```

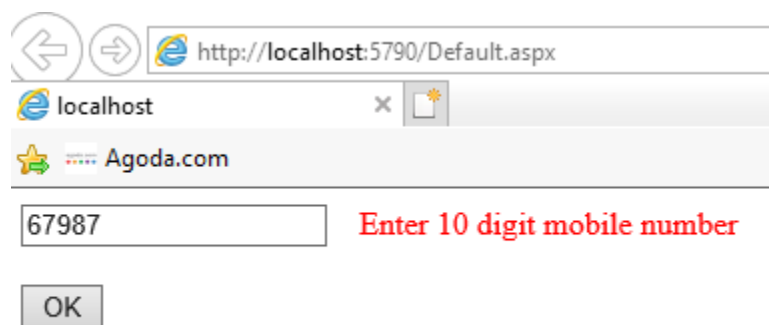


```
{  
  
}  
protected void Button1_Click(object sender, EventArgs e)  
{  
    Label1.Text = TextBox1.Text;  
}  
}
```

Web.config

```
<?xml version="1.0"?>  
<!--  
    For more information on how to configure your ASP.NET application, please visit  
    http://go.microsoft.com/fwlink/?LinkId=169433  
-->  
<configuration>  
  <appSettings>  
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>  
  </appSettings>  
  <system.web>  
    <compilation debug="true" targetFramework="4.5"/>  
    <httpRuntime targetFramework="4.5"/>  
  </system.web>  
</configuration>
```

Output:-

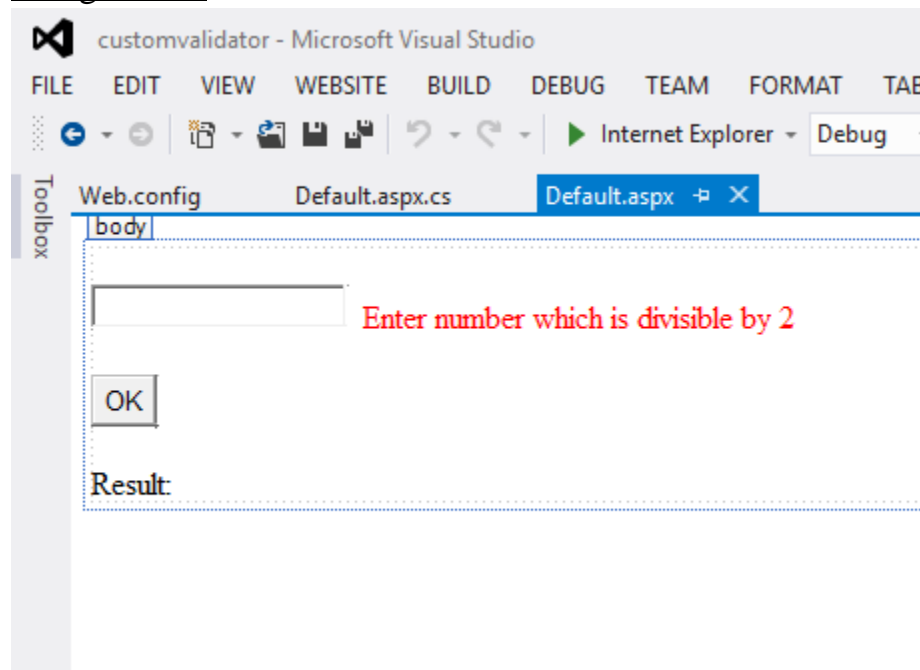


Result:

5 Custom Validator

Solution:-

Design View



Default.aspx

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

<!DOCTYPE html>

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

            &nbsp;&nbsp;&nbsp;</div>
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
&nbsp;&nbsp;&nbsp;<br />
            <asp:CustomValidator ID="CustomValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="Enter number which is divisible by 2"
ForeColor="Red"
OnServerValidate="CustomValidator1_ServletValidator"></asp:CustomValidator>
        <br />
        <br />
    </form>

```

```
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="OK" />
<br />
<br />
<asp:Label ID="Label1" runat="server" Text="Result:"></asp:Label>
</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;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = TextBox1.Text;
    }

    protected void CustomValidator1_ServerValidator(object source,
ServerValidateEventArgs args)
    {
        int num = int.Parse(TextBox1.Text);
        if (num % 2 == 0)
        {
            args.IsValid = true;
        }
        else
        {
            args.IsValid = false;
        }
    }
}
```

Web.config

```
<?xml version="1.0"?>
```

```
<!--
```

For more information on how to configure your ASP.NET application, please visit
<http://go.microsoft.com/fwlink/?LinkId=169433>

```
-->
```

```
<configuration>
```

```
<appSettings>
```

```
<add key="ValidationSettings"/>
```

```
</appSettings>
```

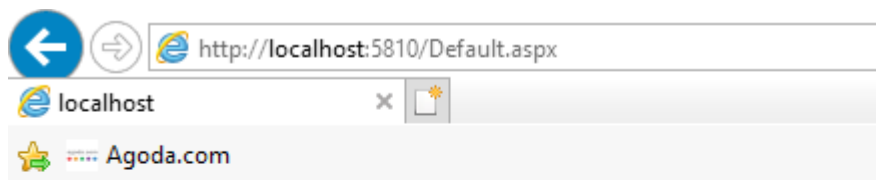
```
<system.web>
```

```
<compilation debug="false" targetFramework="4.5" />
```

```
<httpRuntime targetFramework="4.5" />
```

```
</system.web>
```

```
</configuration>
```

Output:-

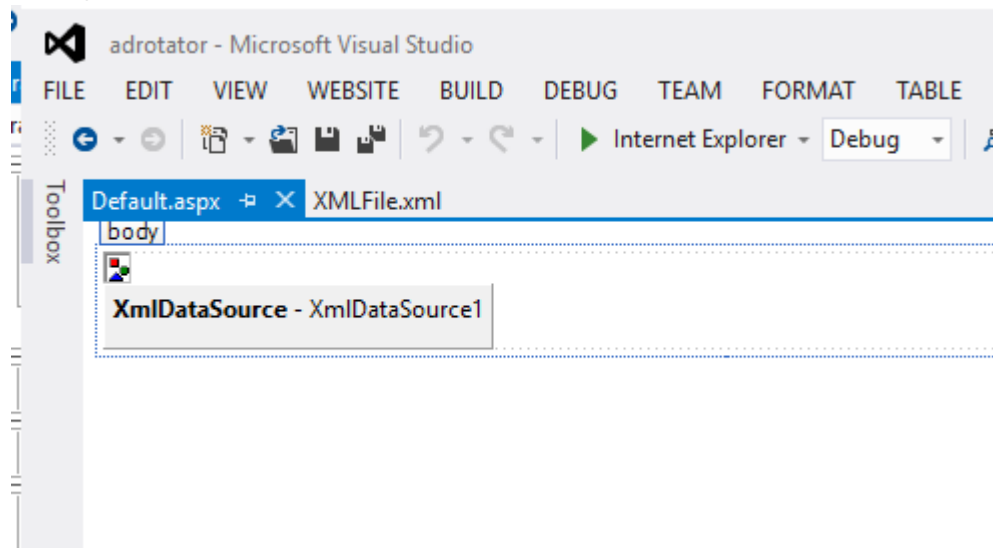
Enter number which is divisible by 2

3

b. Create Web Form to demonstrate use of Adrotator Control.

Step1:-

Design View



Default.aspx

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

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

```
<asp:AdRotator ID="AdRotator1" runat="server"
DataSourceID="XmlDataSource1" />
```

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

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

XML File.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
  <Ad>
    <ImageUrl>beauty-wallpaper.jpg</ImageUrl>
    <NavigateUrl>http://www.google.com</NavigateUrl>
    <AlternateText>Google Site</AlternateText>
  </Ad>
  <Ad>
    <ImageUrl>planet blue ice space.jpg</ImageUrl>
    <NavigateUrl>http://www.facebook.com</NavigateUrl>
    <AlternateText>Facebook Site</AlternateText>
  </Ad>
  <Ad>
    <ImageUrl>rose.jpg</ImageUrl>
    <NavigateUrl>http://www.gmail.com</NavigateUrl>
    <AlternateText>Gmail Site</AlternateText>
  </Ad>
</Advertisements>
```

Output:-

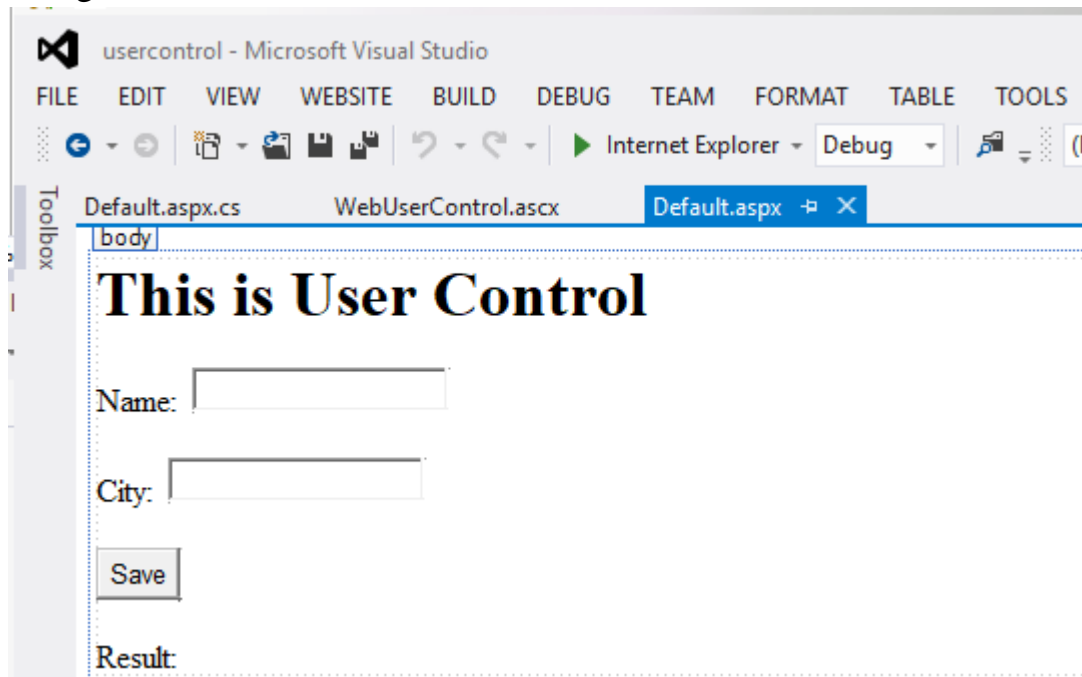




- c. Create Web Form to demonstrate use User Controls.

Step1:-

Design view:-



Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
<%@ Register Src="~/WebUserControl.ascx" TagPrefix="UC" TagName="student" %>
```

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

```
<UC:student ID="Studentcontrol" runat="server"/>
```

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

```
&nbsp;
```

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

```
<br />
```

```
<br />
```

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

```
&nbsp;
```

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



```
<br />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Save"
/>
<br />
<br />
<asp:Label ID="Label3" runat="server" Text="Result:"></asp:Label>

</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;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label3.Text += "Your name is " + TextBox1.Text + " and you are from " +
        TextBox2.Text;
    }
}
```

WebUserControl.aspx.cs

```
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="WebUserControl.ascx.cs" Inherits="WebUserControl" %>
<h1>This is User Control</h1>
```

Output:-



The screenshot shows a web browser window with the address bar displaying `http://localhost:5822/Default.aspx`. The browser's tab is labeled `localhost` and the address bar includes a search icon and the text `Agoda.com`. The main content area of the browser displays the following:

This is User Control

Name:

City:

Result: Your name is Rishika and you are from Bhiwandi

PRACTICAL NO-5

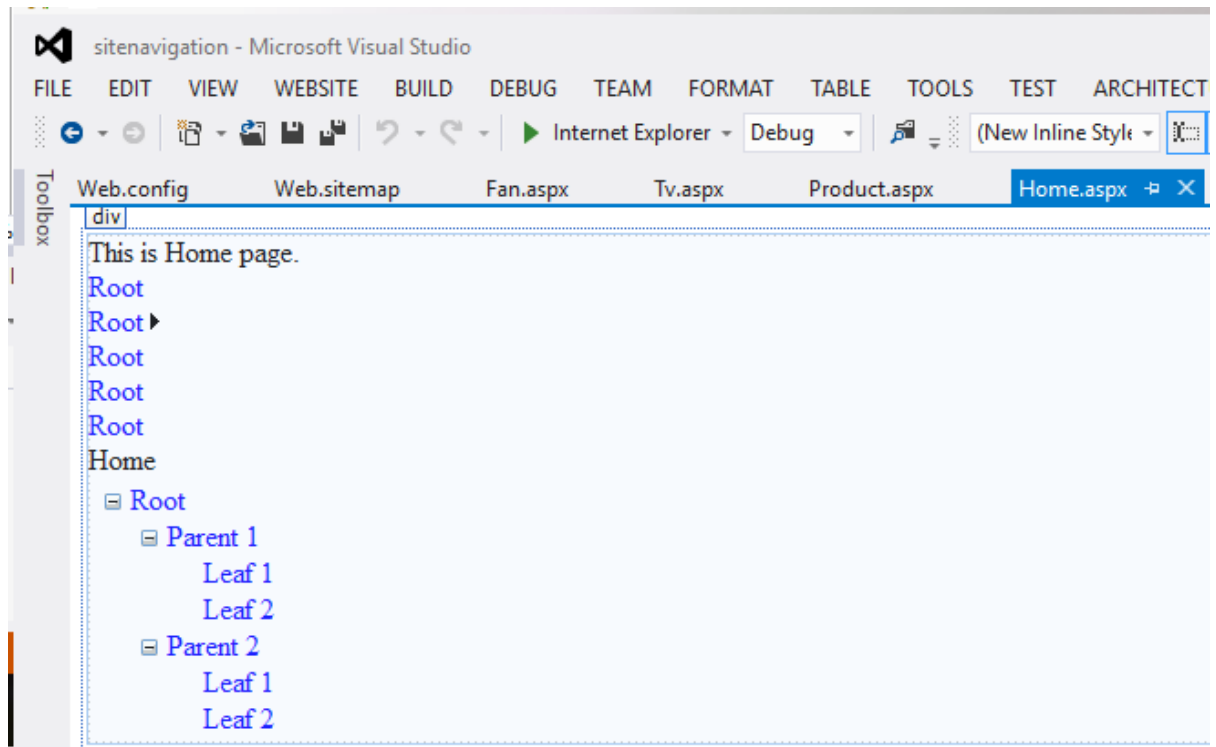
5. Working with Navigation, Beautification and Master page.

a. Create Web Form to demonstrate use of Website Navigation controls and Site Map.

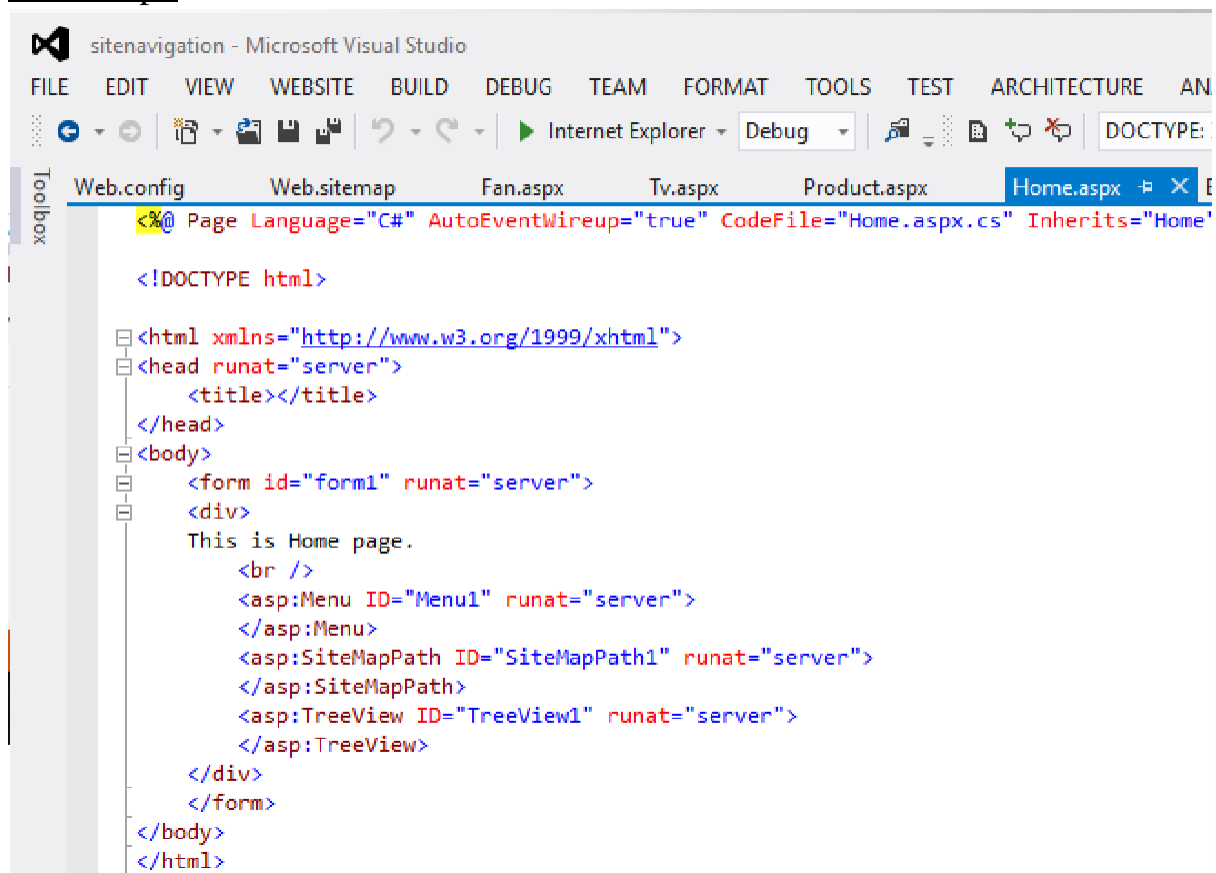
a. Create Web Form to demonstrate use of Website Navigation controls and Site Map.

Step1:-

Design view:-



Home.aspx



Web.sitemap

```

<?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 url="Product.aspx" title="Product" description="" >

      <siteMapNode url="Electronics.aspx" title="Electronics" description="" >
        <siteMapNode url="Tv.aspx" title="Tv" description="" />
        <siteMapNode url="Fan.aspx" title="Fan" description="" />
      </siteMapNode>

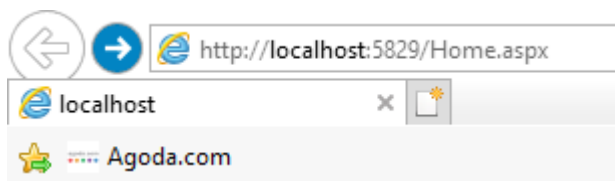
      <siteMapNode url="Fashion.aspx" title="Fashion" description="" >
        <siteMapNode url="Cloths.aspx" title="Cloths" description="" />
      </siteMapNode>

      <siteMapNode url="Shopping.aspx" title="Shopping" description="" >
        <siteMapNode url="Bags" title="Bags" description="" />
        <siteMapNode url="Shoes.aspx" title="Shoes" description="" />
      </siteMapNode>

    </siteMapNode>
  </siteMapNode>
</siteMap>

```

Output:-



This is Home page.

Home > Product > Electronics > Tv
 Home Fashion Fan
 Shopping >

PRACTICAL NO-6

6. Working with Database

- a. Create a web application bind data in a multiline textbox by querying in another textbox.
- b. Create a web application to display records by using database.
- c. Demonstrate the use of Datalist link control.

a) Create a web application bind data in a multiline textbox by querying in another textbox.

Code:

Default.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>
```

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

```
<asp:TextBox ID="TextBox1"runat="server"Height="48px"TextMode="MultiLine"Width="206px"></asp:TextBox>
```

```
<br/>
```

```
<br/>
```

```
<asp:Button ID="Button1"runat="server"OnClick="Button1_Click"Text="ok"/>
```

```
<br/>
```

```
<br/>
```

```
<asp:TextBox ID="TextBox2"runat="server"Height="130px"TextMode="MultiLine"Width="275px"></asp:TextBox>
```

```
</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;
```

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
using System.Configuration;
```

```
public partial class _Default :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 sdr = cmd.ExecuteReader();
    TextBox2.Text = "";
    while (sdr.Read())
    {
        TextBox2.Text += Environment.NewLine;
        for (int i = 0; i < sdr.FieldCount; i++)
            TextBox2.Text += sdr[i].ToString().PadLeft(15);
    }
    sdr.Close();
    con.Close();
}
}
```

Web.config

```
<?xml version="1.0"?>
<!--
    For more information on how to configure your ASP.NET application, please visit
    http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5"/>
    <httpRuntime targetFramework="4.5"/>
  </system.web>
  <connectionStrings>
    <add name="connstr" connectionString="Data Source=.;InitialCatalog=db;Integrated
    Security=True"/>
  </connectionStrings>
</configuration>
```

OUTPUT:

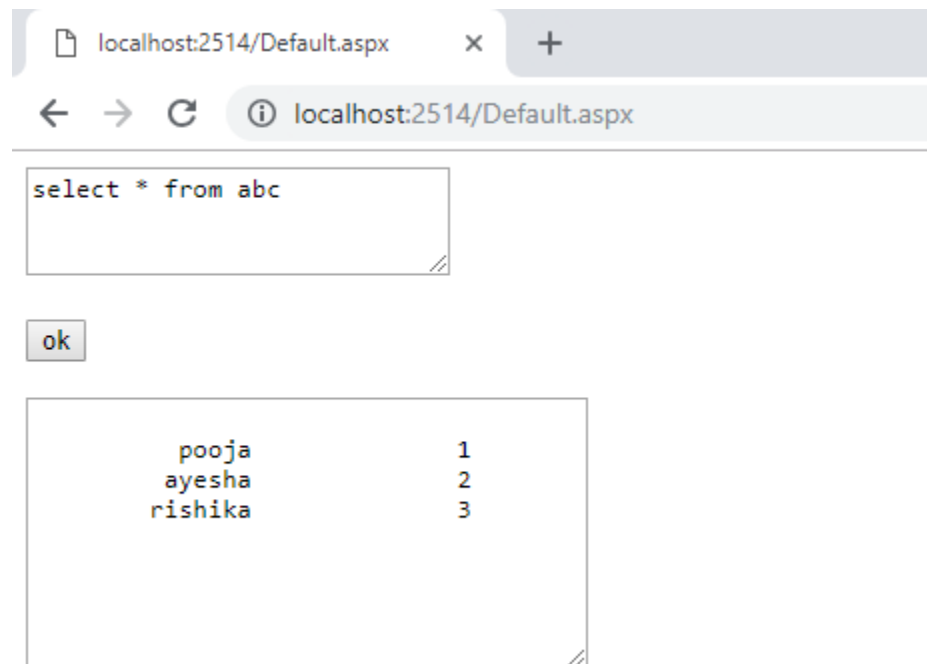
The screenshot shows a web browser window with the title 'Default.aspx*'. The page content consists of two text input fields stacked vertically. The first input field has a label 'body' positioned to its left. Below the first input field is a button labeled 'ok'. The second input field is located below the 'ok' button. Both input fields have a small vertical scrollbar on their right side. The browser's address bar is empty.

The screenshot displays the Microsoft SQL Server Enterprise Edition interface. On the left, the Object Explorer shows the server hierarchy: Desktop-VCAKQPJ\SQLEXPRESS (SQL Server) > Databases > db. The main window shows a SQL query in the SQL Query1.sql file:

```
use db
create table abc
(
    name varchar(20),
    id int
)
insert into abc values('rishika',3)
select * from abc
```

Below the query editor, the Results tab is active, showing the output of the query:

	name	id
1	pooja	1
2	ayesha	2
3	rishika	3



The screenshot shows a web browser window with a single tab titled 'localhost:2514/Default.aspx'. The address bar also displays 'localhost:2514/Default.aspx'. Below the address bar, there is a text input field containing the SQL query 'select * from abc'. Below the input field is a button labeled 'ok'. Below the button is a table displaying the results of the query.

pooja	1
ayesha	2
rishika	3

b) Create a web application to display records by using database.

CODE:

Default.aspx.cs

```
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 _Default : 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("select * from abc", con);
        SqlDataReader sdr = cmd.ExecuteReader();

        while (sdr.Read())
        {
            Label1.Text += sdr["name"].ToString()+" " +sdr["id"].ToString()+"<br>" ;
        }
        sdr.Close();
        con.Close();
    }
}
```

Default.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>

<!DOCTYPE.html>

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

```
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

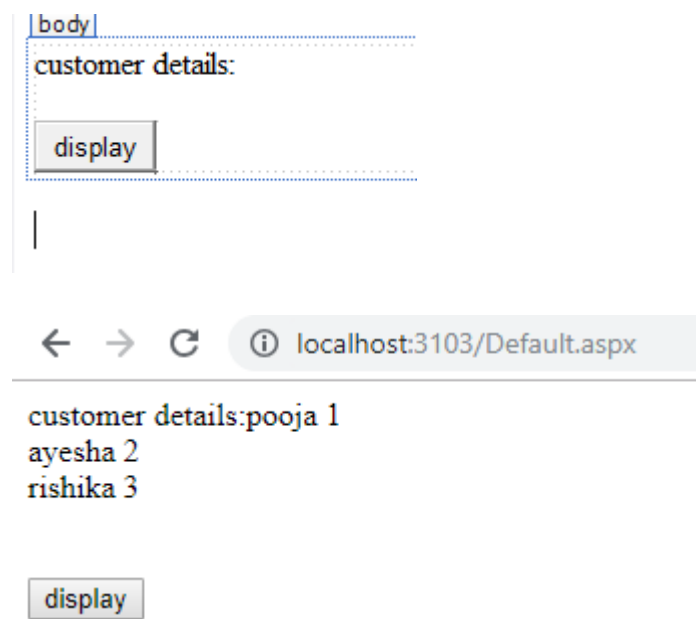
<asp:Label ID="Label1" runat="server" Text="customer details:"></asp:Label>
<br/>
<br/>
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="display"/>

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

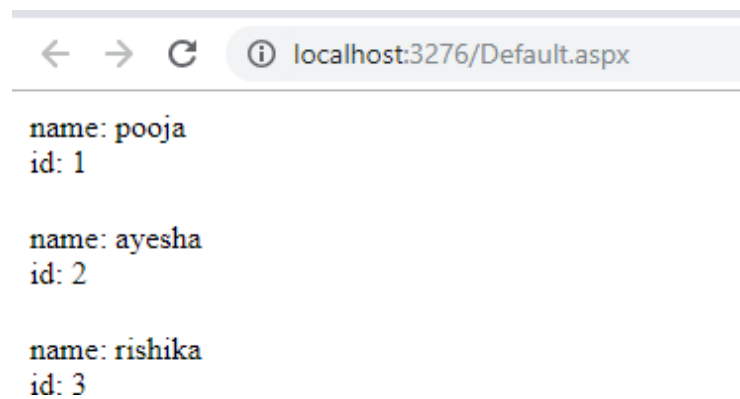
Web.config

```
<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
<system.web>
<compilation debug="true" targetFramework="4.5"/>
<httpRuntime targetFramework="4.5"/>
</system.web>
<connectionStrings>
<add name="connstr" connectionString="Data Source=.;Initial Catalog=db; Integrated
Security=True"/>
</connectionStrings>

</configuration>
```

OUTPUT:

c) Demonstrate the use of Datalist link control.

OUTPUT:

PRACTICAL NO-7

7. Working with Database

- a. Create a web application to display Databinding using dropdownlist control.
- b. Create a web application for to display the phone no of an author using database.
- c. Create a web application for inserting and deleting record from a database. (Using Execute-Non Query).

a) Create a web application to display Databinding using dropdownlist control.

CODE:

Default.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>
```

```
<!DOCTYPEhtml>
```

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

```
<headrunat="server">
```

```
<title></title>
```

```
</head>
```

```
<body>
```

```
<formid="form1"runat="server">
```

```
<div>
```

```
<asp:DropDownListID="DropDownList1"runat="server"
```

```
DataSourceID="SqlDataSource1"DataTextField="cname"DataValueField="country">
```

```
</asp:DropDownList>
```

```
<asp:SqlDataSourceID="SqlDataSource1"runat="server"ConnectionString="<%$
```

```
ConnectionStrings:dbConnectionString4 %>"SelectCommand="SELECT * FROM  
[customer]"></asp:SqlDataSource>
```

```
<br/>
```

```
<br/>
```

```
<asp:ButtonID="Button1"runat="server"OnClick="Button1_Click"Text="OK"/>
```

```
<br/>
```

```
<br/>
```

```
<asp:LabelID="Label1"runat="server"Text="you have to selected:"></asp:Label>
```

```
</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;
```

```
using System.Data;
```

```
using System.Data.SqlClient;
```

```
using System.Configuration;
public partial class _Default :System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    Protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = "The country you have selected is:" + DropDownList1.SelectedValue;
        if(IsPostBack==false)
        {
            string connstr=ConfigurationManager.ConnectionStrings["connstr"].ConnectionString;
            SqlConnection con=new SqlConnection(connstr);
            SqlCommand cmd = new SqlCommand("Select Distinct country from customer",con );
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            DropDownList1.DataSource = reader;
            DropDownList1.DataTextField = "country";
            DropDownList1.DataBind();
            reader.Close();
            con.Close();
        }
    }
}
```


OUTPUT:

SQLQuery3.sql - DE...KQPJ\Sonali G (54))* X

```
use db
create table customer
(
  cid int,
  cname varchar(20),
  country varchar(30)
)
insert into customer values(3,'pqr','china')
select * from customer
```

100 %

Results Messages

	cid	cname	country
1	1	abc	india
2	2	xyz	india
3	3	pqr	china

Web.config Default.aspx.cs Default.aspx X

body

Databound

SqlDataSource - SqlDataSource1

OK

you have to selected:

|

← → ↻ ⓘ localhost:3383/Default.aspx

pqr ▼

OK

The country you have selected is:china

b) Create a web application for to display the phone no of an author using database.

CODE:

OUTPUT:

```
use db
create table cust
(
  cname varchar(30),
  cid int,
  phoneno int
)
insert into cust values('pqr',3,22114)
select * from cust
```

100 % <

Results Messages

	cname	cid	phoneno
1	abc	1	23134
2	xyz	2	22234
3	pqr	3	22114

c) Create a web application for inserting and deleting record from a database.
(Using Execute-Non Query).

CODE:

```
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 _Default : 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);
        string insertquery="insert into person values(@firstname, @lastname, @city,
        @phoneno, @cno)";
        SqlCommand cmd=new SqlCommand(insertquery,con);
        cmd.Parameters.AddWithValue("@firstname",TextBox1.Text);
        cmd.Parameters.AddWithValue("@lastname",TextBox2.Text);
        cmd.Parameters.AddWithValue("@city",TextBox3.Text);
        cmd.Parameters.AddWithValue("@phoneno",TextBox4.Text);
        cmd.Parameters.AddWithValue("@cno",TextBox5.Text);
        con.Open();
        cmd.ExecuteNonQuery();
        Label1.Text="RecordsInsertedSuccessfully...!";
        con.Close();
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        string connstr=ConfigurationManager.ConnectionStrings["connstr"].ConnectionString;
        SqlConnection con=new SqlConnection(connstr);
        string updatequery = "update person set
        firstname=@firstname,lastname=@lastname,city=@city,phoneno=@phoneno,cno=@cno
        where cno=@cno";
        SqlCommand cmd=new SqlCommand(updatequery,con);
        cmd.Parameters.AddWithValue("@cno", TextBox1.Text);
```

```
cmd.Parameters.AddWithValue("@firstname", TextBox2.Text);
cmd.Parameters.AddWithValue("@lastname", TextBox3.Text);
cmd.Parameters.AddWithValue("@city", TextBox4.Text);
cmd.Parameters.AddWithValue("@phoneno", TextBox5.Text);

con.Open();
cmd.ExecuteNonQuery();
Label1.Text="Records Updated Successfully...!";
con.Close();

}
protected void Button3_Click(object sender, EventArgs e)
{
string connstr = ConfigurationManager.ConnectionStrings["connstr"].ConnectionString;
SqlConnection con = new SqlConnection(connstr);
string deletequery = "delete from person where firstname=@firstname";
SqlCommand cmd = new SqlCommand(deletequery, con);
cmd.Parameters.AddWithValue("@firstname", TextBox1.Text);
con.Open();
cmd.ExecuteNonQuery();
    Label1.Text = "RecordsDeletedSuccessfully...!"; con.Close();

}
protected void Button4_Click(object sender, EventArgs e)
{
    TextBox1.Text = "";
    TextBox2.Text = "";
    TextBox3.Text = "";
    TextBox4.Text = "";
    TextBox5.Text = "";
    Label1.Text = "";

}
protected void Button5_Click(object sender, EventArgs e)
{
    GridView1.Visible = true;
}

}
```

Default.aspx

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="Default.aspx.cs"Inherits="_Default"%>
```

<!DOCTYPEhtml>

[illegible]

```
<br/>
<br/>
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataSourceID="SqlDataSource1">
<Columns>
<asp:BoundField DataField="cno" HeaderText="cno" SortExpression="cno"/>
<asp:BoundField DataField="firstname" HeaderText="firstname"
SortExpression="firstname"/>
<asp:BoundField DataField="lastname" HeaderText="lastname"
SortExpression="lastname"/>
<asp:BoundField DataField="city" HeaderText="city" SortExpression="city"/>
<asp:BoundField DataField="phoneno" HeaderText="phoneno"
SortExpression="phoneno"/>
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%"$
ConnectionStrings:dbConnectionString%">" SelectCommand="SELECT * FROM
[person]"></asp:SqlDataSource>

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

OUTPUT:

FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM SQL FORMAT

Google Chrome Debug

Default.aspx.cs Web.config Default.aspx

Server Explorer Toolbox

cno:

fname:

lname:

city:

phoneno:

insert update delete clear show

asp:gridview#GridView1

cno	firstname	lastname	city	phoneno
0	abc	abc	abc	0
1	abc	abc	abc	1
2	abc	abc	abc	2
3	abc	abc	abc	3
4	abc	abc	abc	4

SqlDataSource - SqlDataSource1

SQLQuery6.sql - DE...KQPJ\Sonali G (54))* X

```
use db
create table person
(
  cno int,
  firstname varchar(20),
  lastname varchar(30),
  city varchar(20),
  phoneno int
)
insert into person values(2,'ayesha','ansari','bhiwandi',11311)
select * from person
```

100 %

Results Messages

	cno	firstname	lastname	city	phoneno
1	1	pooja	bhoir	bhiwandi	21311
2	2	ayesha	ansari	bhiwandi	11311

← → ↻ ⓘ localhost:1649/Default.aspx

RecordsInsertedSuccessfully...!

fname

lname:


city

phoneno:

cno	firstname	lastname	city	phoneno
1	pooja	bhoir	bhiwandi	21311
2	ayesha	ansari	bhiwandi	11311



RecordsDeletedSuccessfully...! fname lname: city phoneno:

cno	firstname	lastname	city	phoneno
1	pooja	bhoir	bhiwandi	21311
2	ayesha	ansari	bhiwandi	11311
3	kajal	yadav	bhiwandi	12345

 localhost:1649/Default.aspxRecords Updated Successfully...! fname lname: city phoneno:

cno	firstname	lastname	city	phoneno
1	pooja	bhoir	bhiwandi	21311
2	ayesha	ansari	bhiwandi	11311
3	kajal	yadav	bhiwandi	12345

100 % <

 Results  Messages

	cno	firstname	lastname	city	phoneno
1	1	nishika	shikavat	bhiwandi	23456
2	2	ayasha	ansari	bhiwandi	11311
3	3	kajal	yadav	bhiwandi	12345

PRACTICAL NO-8

8. Working with Data Controls

- a. Create a web application to demonstrate various uses and properties of SqlDataSource.
- b. Create a web application to demonstrate data binding using DetailsView and FormView Control.
- c. Create a web application to display Using Disconnected Data Access and Databinding using GridView.

a) Create a web application to demonstrate various uses and properties of SqlDataSource.

CODE:

```
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 _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlDataSource1.SelectCommand= "select * from customer where city= '" +
        DropDownList1.SelectedValue + "'";
    }
}
```

Default.aspx

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

```
<!DOCTYPE html>
```

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

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

```
<title></title>
```

```
</head>
```

```
<body>
```

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

```
<div>
```

```
</div>
```

```
<asp:DropDownList ID="DropDownList1" runat="server"
```

```
DataSourceID="SqlDataSource1" DataTextField="city" DataValueField="city">
```

```
</asp:DropDownList>
```

```
<br />
```

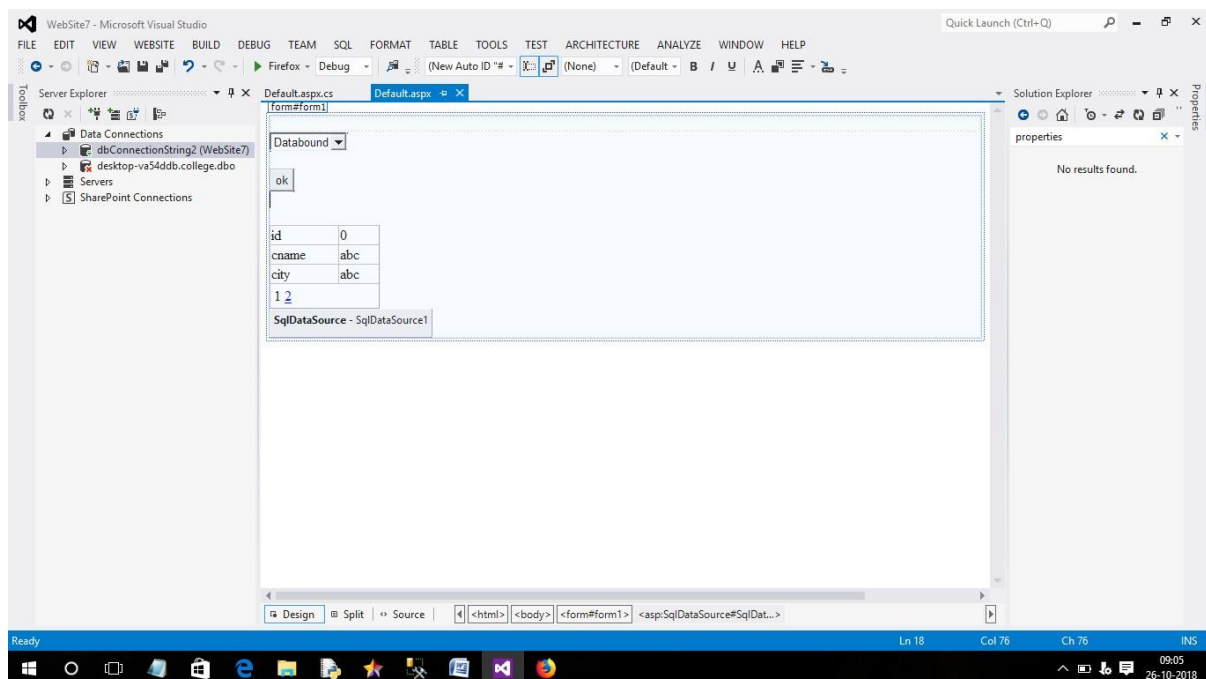
```
<br />
```

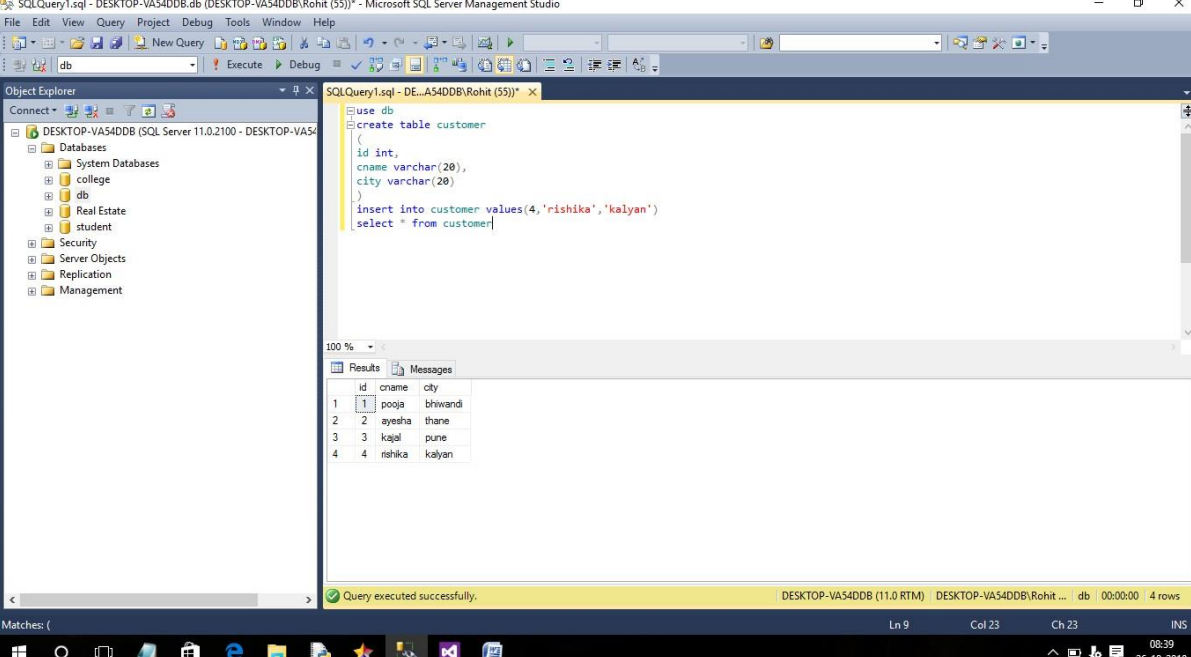
```

<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="ok" />
<br />
<br />
<asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False" DataSourceID="SqlDataSource1" Height="50px"
Width="125px">
    <Fields>
        <asp:BoundField DataField="id" HeaderText="id" SortExpression="id" />
        <asp:BoundField DataField="cname" HeaderText="cname"
SortExpression="cname" />
        <asp:BoundField DataField="city" HeaderText="city" SortExpression="city" />
    </Fields>
</asp:DetailsView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%=
ConnectionStrings:dbConnectionString2 %>" SelectCommand="SELECT * FROM
[customer]"></asp:SqlDataSource>
</form>
</body>
</html>

```

OUTPUT:





The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'db' database. The central query window shows the following SQL code:

```
use db
create table customer
(
    id int,
    cname varchar(20),
    city varchar(20)
)
insert into customer values(4,'rishika','kalyan')
select * from customer
```

The Results pane at the bottom shows the output of the query, which is a table with 4 rows and 3 columns (id, cname, city):

	id	cname	city
1	1	pooja	bhiwandi
2	2	ayasha	thane
3	3	kajal	pune
4	4	rishika	kalyan

The status bar at the bottom indicates that the query was executed successfully, returning 4 rows. The taskbar at the bottom shows the Windows Start button and several open applications.

b) Create a web application to demonstrate data binding using DetailsView and FormView Control

CODE:

Default.aspx

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

<!DOCTYPE html>

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

            <asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False" DataSourceID="SqlDataSource1" Height="50px"
Width="125px">
                <Fields>
                    <asp:BoundField DataField="id" HeaderText="id" SortExpression="id" />
                    <asp:BoundField DataField="cname" HeaderText="cname"
SortExpression="cname" />
                    <asp:BoundField DataField="city" HeaderText="city" SortExpression="city" />
                </Fields>
            </asp:DetailsView>
            <br />
            <asp:FormView ID="FormView1" runat="server" AllowPaging="True"
DataSourceID="SqlDataSource1">
                <EditItemTemplate>
                    id:
                    <asp:TextBox ID="idTextBox" runat="server" Text='<%# Bind("id") %>' />
                    <br />
                    cname:
                    <asp:TextBox ID="cnameTextBox" runat="server" Text='<%# Bind("cname") %>' />
                    <br />
                    city:
                    <asp:TextBox ID="cityTextBox" runat="server" Text='<%# Bind("city") %>' />
                    <br />
                    <asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True"
CommandName="Update" Text="Update" />
                    &nbsp;<asp:LinkButton ID="UpdateCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel" Text="Cancel" />
                </EditItemTemplate>
            </asp:FormView>
        </div>
    </form>
</body>
</html>
```

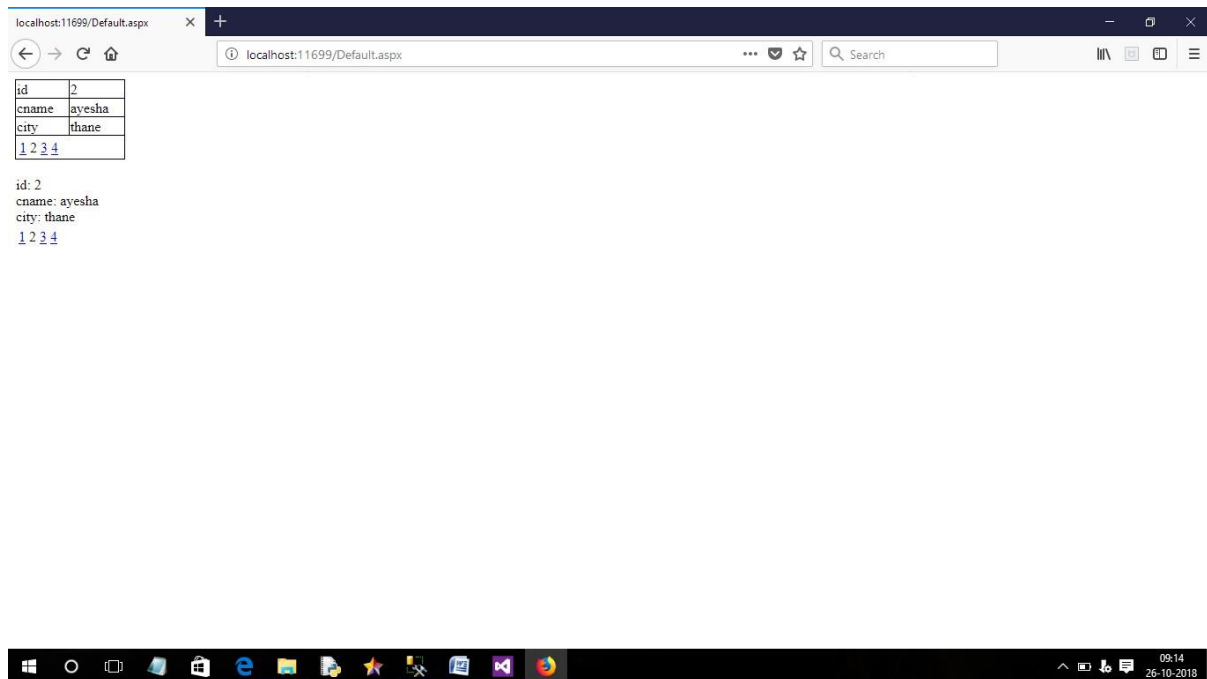
```

</EditItemTemplate>
<InsertItemTemplate>
    id:
    <asp:TextBox ID="idTextBox" runat="server" Text='<%# Bind("id") %>' />
    <br />
    cname:
    <asp:TextBox ID="cnameTextBox" runat="server" Text='<%# Bind("cname") %>' />
    <br />
    city:
    <asp:TextBox ID="cityTextBox" runat="server" Text='<%# Bind("city") %>' />
    <br />
    <asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True"
CommandName="Insert" Text="Insert" />
    &nbsp;<asp:LinkButton ID="InsertCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel" Text="Cancel" />
</InsertItemTemplate>
<ItemTemplate>
    id:
    <asp:Label ID="idLabel" runat="server" Text='<%# Bind("id") %>' />
    <br />
    cname:
    <asp:Label ID="cnameLabel" runat="server" Text='<%# Bind("cname") %>' />
    <br />
    city:
    <asp:Label ID="cityLabel" runat="server" Text='<%# Bind("city") %>' />
    <br />

</ItemTemplate>
</asp:FormView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString='<%$
ConnectionStrings:dbConnectionString %>' SelectCommand="SELECT * FROM
[customer]"></asp:SqlDataSource>

</div>
</form>
</body>
</html>

```


OUTPUT:

C) Create a web application to display Using Disconnected Data Access and Databinding using GridView

CODE:

Default.aspx.cs

```
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 _Default : 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);
        SqlDataAdapter sda = new SqlDataAdapter();
        DataSet ds = new DataSet();
        using (SqlConnection conn = new SqlConnection(connstr))
        {
            SqlCommand cmd = new SqlCommand("select * from customer", conn);
            cmd.CommandType = CommandType.Text;
            sda.SelectCommand = cmd;
            sda.Fill(ds, "country");
            GridView1.DataSource = ds.Tables[0];
            GridView1.DataBind();
        }
    }
}
```

Default.aspx

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
```

```

<system.web>
  <compilation debug="true" targetFramework="4.5"/>
  <httpRuntime targetFramework="4.5"/>
</system.web>
<connectionStrings>
  <add name="connstr" connectionString="Data Source=.;Initial Catalog=db;Integrated
Security=True"/>
</connectionStrings>
</configuration>

```

OUTPUT:

