RAMNIRANJAN JHUNJHUNWALA COLLEGE GHATKOPAR (W), MUMBAI - 400 086

DEPARTMENT OF INFORMATION TECHNOLOGY

2023 - 2024

T.Y. B. Sc. (I .T.) SEM V

Paper RJSUIT503– Advanced Web Programming

Name : Rajbhar Sudesh Dinesh

Roll No. : 6417

Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE

Ghatkopar (W), Mumbai-400 086

Certificate

This is to certify that Mr./Ms.Rajbhar Sudesh Dinesh ,Roll No 6417 of T.Y.B.Sc.(I.T.) class has completed the required number of experiments in the subject of Advanced Web Programming in the Department of Information Technology during the academic year 2023-20 24 .

Professor In-Charge **Co-ordinator of IT Department**

Prof. Bharati Bhole

Prof. Archana Bhide

College Seal & Date **Examiner**

Index

		15/07/2023
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)	01/07/2023
	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 control	07/06/2023
	c) Demonstrate the use of Treeview control performs following operations. a. Treeview control and datalist b. Treeview Operations	07/06/2023
4	**Working with Navigation, Beautification and Master Page.**	
	a) Create a Registration form to demonstrate use of various Validation controls.	07/06/2023
	b) Create a Web Form to demonstrate use of Adrotator Control.	22/07/2023
	c) Create a Web Form to demonstrate use of User Controls.	22/07/2023
5	**Working with Navigation, Beautification and Master page.**	
	a) Create Web Form to demonstrate use of Website Navigation controls and Site Map. i.Menu Control. ii.Site Map Control	22/07/2023

	b) Create a web application to demonstrate use of Master page with applying Styles and Themes for page beautification.	15/07/2023
	c) Create a web application to demonstrate various states of ASP.NET Pages. i.View State ii.Query String	15/07/2023
6	**Working with Database**	
	a) Create a web application bind data in a multiline textbox by querying in another textbox.	12/08/2023
	b) Demonstrate the use of Data list link control	12/08/2023
7	**Working with Database**	
	a) Create a web application to display Data Binding using dropdown list control.	26/08/2023
	b) Create a web application to display the phone no of an author using a database.	26/08/2023
	c) Create a web application for inserting and deleting records from a database. (Using Execute-Non Query).	26/08/2023
8	**Working with Data Controls**	
	a) Create a web application to demonstrate data binding using DetailsView and FormView Control.	19/08/2023
	b) Create a web application to display Using Disconnected Data Access and Data binding using GridView.	19/08/2023
9	**Working with GridView control**	
	a) Create a web application to demonstrate use of GridView button column and GridView events.	26/08/2023

	b) Create a web application to demonstrate GridView paging and Create own table format using GridView.	26/08/2023
	c) Create a web application to demonstrate use of GridView control template and GridView hyperlink.	09/09/2023
10	**Working with AJAX and XML**	
	a) Create a web application to demonstrate reading and writing operation with XML.	02/09/2023
	b) Create a web application to demonstrate use of various Ajax controls.	09/09/2023
11	**Programs to create and use DLL**	02/09/2023

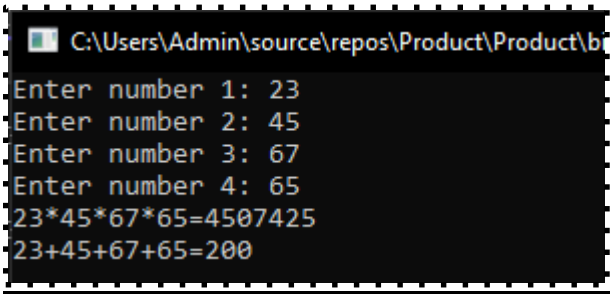
PRACTICAL NO 1: Working with basic C# and ASP .NET

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

Source code:

```
_using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int num1, num2, num3, num4, prod,sum;
         Console.Write("Enter number 1: ");
         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;
         sum= num1+ num2 +num3+num4;
         Console.WriteLine(num1 + "*" + num2 + "*" + num3 + "*" + num4 + "=" + prod);
         Console.WriteLine(num1 + "+" + num2 + "+" + num3 + "+" + num4 + "=" + sum);
         Console.ReadLine();       }
   }}
```

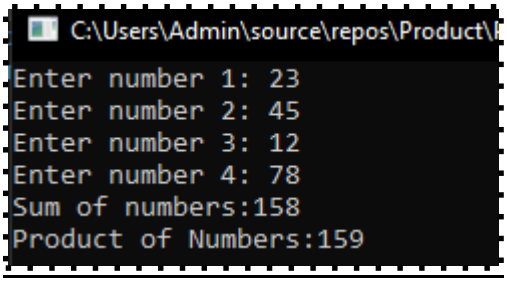
Output:



A) Using LOOP.

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {

         Console.Write("Enter number 1: ");
         int num1 = Int32.Parse(Console.ReadLine());
         Console.Write("Enter number 2: ");
         int num2 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 3: ");
         int num3 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 4: ");
         int num4 = Convert.ToInt32(Console.ReadLine());
         int[] ar = { num1, num2, num3, num4 };
         int sum = 0;
         int product = 1;
         for (int i = 0;i<4;i++)
         {
            sum += ar[i];
         }
         for (int i = 0; i < 4; i++)
         {
            product += ar[i];
         }
         Console.WriteLine("Sum of numbers:"+sum);
         Console.WriteLine("Product of Numbers:" + product);
         Console.ReadLine();       }
   }
}
```

Output:

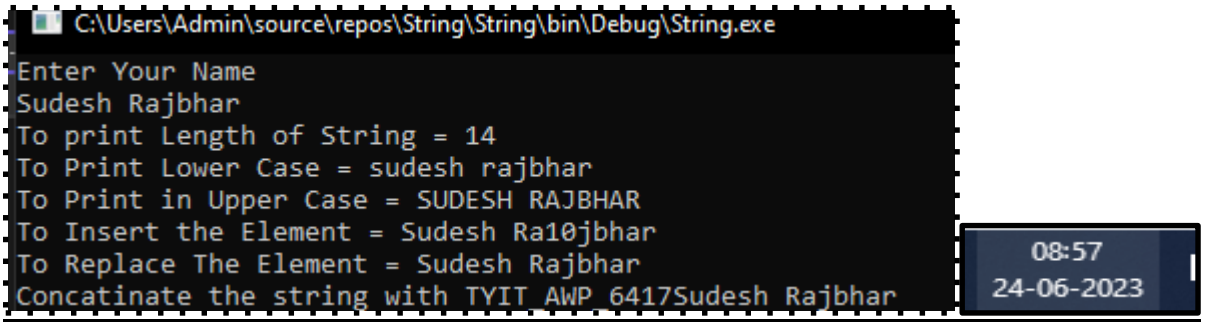


1)B) Create an application to demonstrate string operations.

Source code:

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

namespace ConsoleApp1
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Enter Your Name ");
         String s = Console.ReadLine();
         Console.WriteLine("To print Length of String = " + s.Length);
         Console.WriteLine("To Print Lower Case = " + s.ToLower());
         Console.WriteLine("To Print in Upper Case = " + s.ToUpper());
         Console.WriteLine("To Insert the Element = " + s.Insert(9, "10"));
         Console.WriteLine("To Replace The Element = " + s.Replace("y", "i"));
         String s2 = String.Concat("TYIT_AWP 6417");
         Console.WriteLine("Concatenate the string with TYIT_AWP_6417" + s);
         Console.ReadLine();
      }
   }
}
```

Output:

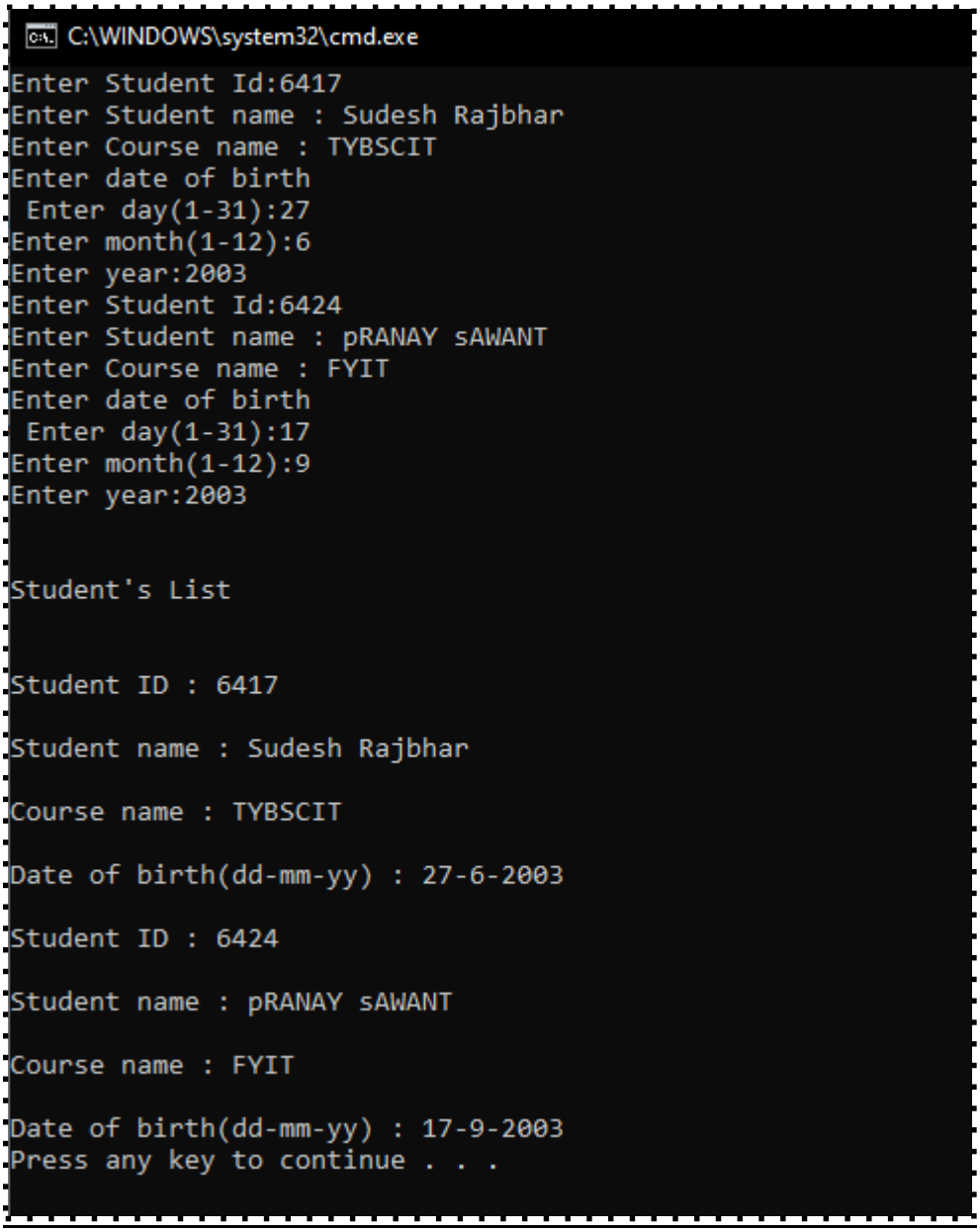


1) C) Create an application that receives the (Student Id, Student Name, Course Name, Date of Birth) information from a set of students. The application should also display the information of all the students once the data entered.

Source Code:

```
using System;
namespace ArrayOfStructs
{
 class Program
 {
 struct Student
 {
 public string studid, name, cname;
 public int day, month, year;
 }
 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);
```



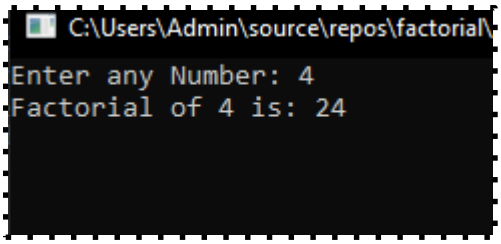
PRACTICAL NO 2: Working with Object Oriented C# and ASP .NET

2) A) i) Factorial

Source Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
public class Factorial
  {
    public static void Main(string[] args)
    {
      int i, fact=1,number;
      Console.Write("Enter any Number: ");
      number = int.Parse(Console.ReadLine());
      for (i = 1; i <= number; i++)
      {
         fact = fact * i;
      }
      Console.Write("Factorial of " + number + " is: " + fact);
      Console.ReadLine();
   }
 }
```

Output:



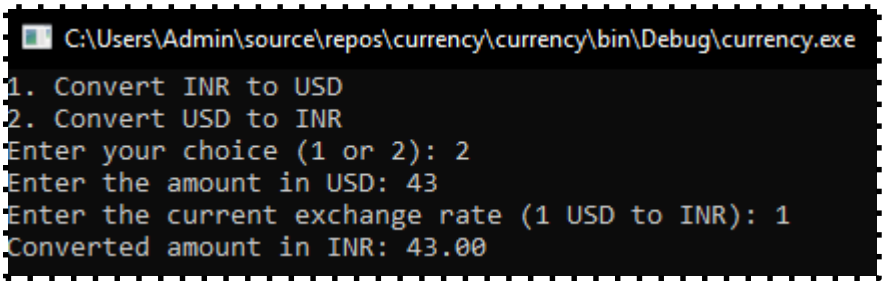
2) A) ii) Currency Converter

Source code:

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

namespace Currency Converter
{
   class Program
   {
      static void Main(string[] args)
      {
         double inr, usd, exchangeRate;

         Console.WriteLine("1. Convert INR to USD");
         Console.WriteLine("2. Convert USD to INR");
         Console.Write("Enter your choice (1 or 2): ");
         int choice = int.Parse(Console.ReadLine());

         if (choice == 1)
         {
            // Convert INR to USD
            Console.Write("Enter the amount in INR: ");
            inr = double.Parse(Console.ReadLine());

            Console.Write("Enter the current exchange rate (1 INR to USD): ");
            exchangeRate = double.Parse(Console.ReadLine());

            usd = inr / exchangeRate;
            Console.WriteLine("Converted amount in USD: " + usd.ToString("0.00"));
         }
         else if (choice == 2)
         {
            // Convert USD to INR
            Console.Write("Enter the amount in USD: ");
            usd = double.Parse(Console.ReadLine());
```

```
            Console.Write("Enter the current exchange rate (1 USD to INR): ");
            exchangeRate = double.Parse(Console.ReadLine());

            inr = usd * exchangeRate;
            Console.WriteLine("Converted amount in INR: " + inr.ToString("0.00"));
        }
        else
        {
            Console.WriteLine("Invalid choice!");
        }

        Console.ReadLine();
    }
  }
}
```

Output:



Source Code: (No 2)

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

namespace Currency
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("Enter the Currency in INR ");
         double currency = Double.Parse(Console.ReadLine());
         Console.WriteLine("1.USD " + "\n" + "2.EUR " + "\n");
          while(true)
         {
            double currency1;
            Console.WriteLine("Enter your choice");
            String Choice= Console.ReadLine();
            if(Choice=="USD")
            {
               currency1 = currency * 0.013;
               Console.WriteLine(currency1);
            }
            else if(Choice=="EURO")
            {
               currency1 = currency * 0.012;
               Console.WriteLine(currency1);
            }
            else if(Choice!="USD"||Choice!="EURO")
            {
               Console.WriteLine("hAVE YOU TYPED WRONG ,TYPE AGAIN:");
               continue;
            }
            Console.WriteLine("do you wish to continue");
            String s = Console.ReadLine();
            if (s=="yes"&& s=="YES")
            {
               continue;
```

```
        }
        else if(s=="no"&& s=="NO")
        {
           break;
        } } } }}
```

Output:

```
Enter the Currency in INR
50
1.USD
2.EUR

Enter your choice
USD
0.65
do you wish to continue
YES
Enter your choice
EURO
0.6
do you wish to continue
NO
```

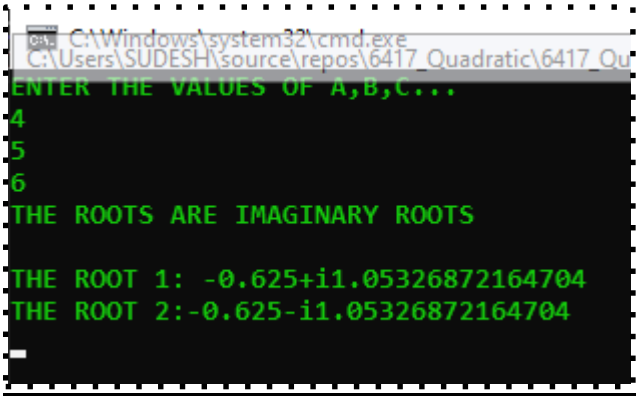
2) A) iii) Quadratic equation.

Source Code:

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

namespace ConsoleApp2
{
   internal class Program
   {

      static void Main(string[] args)
      {
         double a, b, c;
         double disc, deno, x1, x2;
         Console.WriteLine("ENTER THE VALUES OF A,B,C...");
         a = Convert.ToDouble(Console.ReadLine());
         b = Convert.ToDouble(Console.ReadLine());
         c = Convert.ToDouble(Console.ReadLine());
         if (a == 0)
         {
            x1 = -c / b;
            Console.WriteLine("The roots are Linear:", x1);
         }
         else
         {
            disc = (b * b) - (4 * a * c);
            deno = 2 * a;
            if (disc > 0)
            {
               Console.WriteLine("THE ROOTS ARE REAL AND DISTINCT ROOTS");
               x1 = (-b / deno) + (Math.Sqrt(disc) / deno);
               x2 = (-b / deno) - (Math.Sqrt(disc) / deno);
               Console.WriteLine("THE ROOTS ARE... " + x1 + " and " + x2);
            }
            else if (disc == 0)
            {
```

```
                Console.WriteLine("THE ROOTS ARE REPEATED ROOTS");
                x1 = -b / deno;
                Console.WriteLine("THE ROOT IS... : " + x1);
            }
            else
            {
                Console.WriteLine("THE ROOTS ARE IMAGINARY ROOTS\n");
                x1 = -b / deno;
                x2 = ((Math.Sqrt((4 * a * c) - (b * b))) / deno);
                Console.WriteLine("THE ROOT 1: " + x1 + "+i" + x2);
                Console.WriteLine("THE ROOT 2:" + x1 + "-i" + x2);
            }
        }
        Console.ReadLine();

    }
  }
}
```

Output:



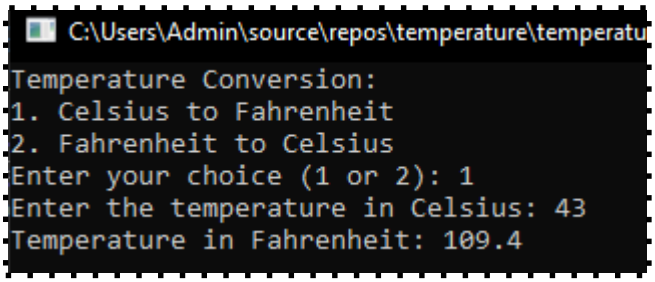
2) A) IV) Temperature Converter

Source Code:

```
using System;

class TemperatureConverter
{
   static void Main()
   {
      Console.WriteLine("Temperature Conversion:");
      Console.WriteLine("1. Celsius to Fahrenheit");
      Console.WriteLine("2. Fahrenheit to Celsius");
      Console.Write("Enter your choice (1 or 2): ");
      int choice = Convert.ToInt32(Console.ReadLine());

      if (choice == 1)
      {
         Console.Write("Enter the temperature in Celsius: ");
         double celsius = Convert.ToDouble(Console.ReadLine());
         double fahrenheit = CelsiusToFahrenheit(celsius);
         Console.WriteLine("Temperature in Fahrenheit: " + fahrenheit);
      }
      else if (choice == 2)
      {
         Console.Write("Enter the temperature in Fahrenheit: ");
         double fahrenheit = Convert.ToDouble(Console.ReadLine());
         double celsius = FahrenheitToCelsius(fahrenheit);
         Console.WriteLine("Temperature in Celsius: " + celsius);
      }
      else
      {
         Console.WriteLine("Invalid choice. Please enter 1 or 2.");
      }
      Console.ReadLine();
   }

   static double CelsiusToFahrenheit(double celsius)
   {
      return (celsius * 9 / 5) + 32;
   }
```

```
static double FahrenheitToCelsius(double fahrenheit)
{
   return (fahrenheit - 32) * 5 / 9;
}}
```

Output:



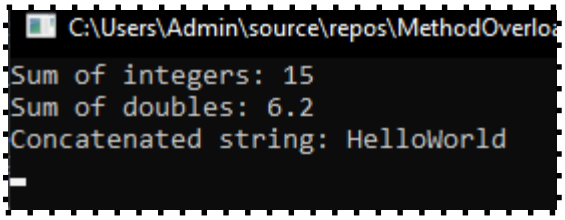
2) B) i) Method overloading.

Source Code:

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

public class Calculator
{
   public int Add(int num1, int num2)
   {
      return num1 + num2;
   }

   public double Add(double num1, double num2)
   {
      return num1 + num2;
   }

   public string Add(string str1, string str2)
   {
      return str1 + str2;
   }
}

public class Program
{
   public static void Main(string[] args)
   {
      Calculator calculator = new Calculator();

      int sumInt = calculator.Add(5, 10);
      Console.WriteLine("Sum of integers: " + sumInt);

      double sumDouble = calculator.Add(2.5, 3.7);
      Console.WriteLine("Sum of doubles: " + sumDouble);

      string concatenatedString = calculator.Add("Hello", "World");
```

```
        Console.WriteLine("Concatenated string: " + concatenatedString);
    }
}
```

Output:



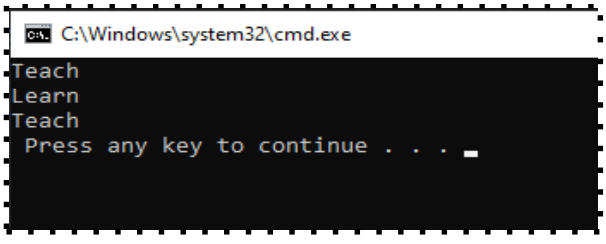
2) B) ii) INHERITANCES.

- Single inheritance:

Source Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Inheritance
{
   class Program
   {
      static void Main(string[] args)
      {
         Teacher d = new Teacher();
         d.Teach();
         Student s = new Student();
         s.Learn();
         s.Teach();
         Console.ReadKey();
      }
      class Teacher
      {
         public void Teach()
         {
            Console.WriteLine("Teach");
         }
      }
      class Student : Teacher
      {
         public void Learn()
         {
            Console.WriteLine("Learn");
         }
      }
   }
}
```

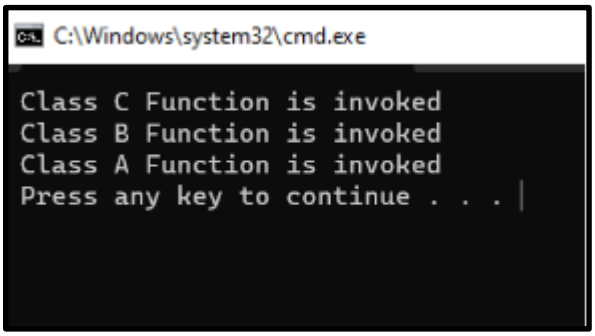
Output:



- Multilevel Inheritance.

```
using System;

namespace Practical2
{
    class A
    {
        public void funcA()
        {
            Console.WriteLine("Class A Function is invoked");
        }
    }

    class B : A
    {
        public void funcB()
        {
            Console.WriteLine("Class B Function is invoked");
        }
    }

    class C : B
    {
        public void funcC()
        {
            Console.WriteLine("Class C Function is invoked");
        }
    }

    internal class MultiLevelInheritance
    {
        public static void Main(string[] args)
        {
            C c = new C();
            c.funcC();
            c.funcB();
            c.funcA();
        }
```

```
  }
}
```


C:\Windows\system32\cmd.exe
Class C Function is invoked
Class B Function is invoked
Class A Function is invoked
Press any key to continue . . .

- Hierarchical Inheritance.

Source Code:

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

namespace ConsoleApp2
{   class A
    {
      public void a()
       {
          Console.WriteLine("Method a() from class A called..");
       }}
    class B : A
    {
      public void b()
       {
          a();
          Console.WriteLine("Method b() from class B called..");
       } }
    class C : A
    {
       public void c()
       {
          a();
          Console.WriteLine("Method c() from class C called..");
       } }
    internal class Program
    {
       static void Main(string[] args)
       {   C c_obj = new C();
          Console.WriteLine("hierarchical inheritance");
          c_obj.c();
          Console.ReadLine();
    }}}
```

```
hierarchical inheritance
Method a() from class A called..
Method c() from class C called..
```

- Multiple Inheritance. / Interface

Source code:

```
using System;
namespace MultipleInheritance
{class Program
   {
      static void Main(string[] args)
      {MultipleInheritanceTest obj = new MultipleInheritanceTest();
         obj.Test();

         Interface1 i1 = obj;
         i1.Show();

         ((Interface2)obj).Show();

         Console.ReadKey();
      } }
   public interface Interface1
   {
      void Test();
      void Show();
   }
   public interface Interface2
   {
      void Test();
      void Show(); }
   public class MultipleInheritanceTest : Interface1, Interface2
   {public void Test()
      {Console.WriteLine("Test Method is Implemented in Child Class"); }

      void Interface1.Show()
      {Console.WriteLine("Interface1 Show Method is Implemented in Child Class");
      }
      void Interface2.Show()
      {
         Console.WriteLine("Interface2 Show Method is Implemented in Child Class");
      } }}
```

```
Test Method is Implemented in Child Class
Interface1 Show Method is Implemented in Child Class
Interface2 Show Method is Implemented in Child Class
```

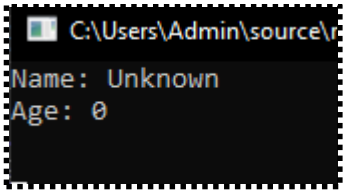
2) B) iii) Constructor Overloading.

Source Code:

```
using System;

public class Person
{
    private string name;
    private int age;

    public Person()
    {
        name = "Unknown";
        age = 0;
    }

    public Person(string name)
    {
        this.name = name;
        age = 0;
    }

    public Person(string name, int age)
    {
        this.name = name;
        this.age = age;
    }

    public void Display()
    {
        Console.WriteLine("Name: " + name);
        Console.WriteLine("Age: " + age);
        Console.WriteLine();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Person person1 = new Person();
```

```
        person1.Display();

        Person person2 = new Person("John");
        person2.Display();

        Person person3 = new Person("Jane", 25);
        person3.Display();
    }
}
```

Output:

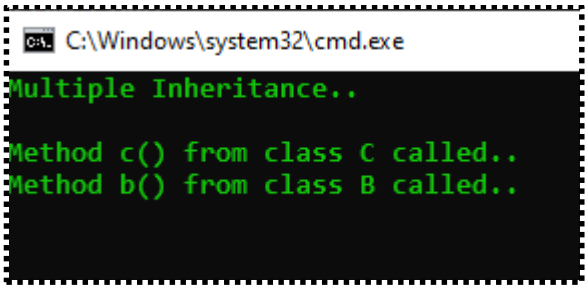


2) B) iv) Interfaces

Source Code:

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

namespace Inheritance
{
   internal class Program
   {
      class A
      {
         public void a()
          {
             Console.WriteLine("Method a() from class A called..");
          }

      }
      interface B
      {
          void b();


      }
      class C : A,B
      {
         public void c()
          {
             Console.WriteLine("Method c() from class C called..");
          }
          public void b()
          {
             Console.WriteLine("Method b() from class B called..");
          }
```

```
    }
    static void Main(string[] args)
    {
      C obj_c = new C();
       Console.WriteLine("Multiple Inheritance..\n");
       obj_c.c();
       obj_c.b();
       Console.ReadLine();
    }
  }
}
```



2) C) i) Using Delegates and Events.

Source Code:

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

namespace ConsoleApp4
{
   internal class Program
   {

      // "addnum" and "subnum" are two delegate names
      public delegate void addnum(int x,int y);
      public delegate void subnum(int x, int y);


      public void sum(int x, int y)
      {
         Console.Write("\n");
         Console.WriteLine(x + " + " + y + "=" + (x + y));
      }
      public void sub(int x, int y)
      {
         Console.WriteLine(x + " - " + y + "=" + (x - y));
      }

      static void Main(string[] args)
      {
        int num1, num2,div;
         Console.Write("Enter Number 1 : ");
         num1 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter Number 2 : ");
         num2 = Convert.ToInt32(Console.ReadLine());

         Program obj = new Program();

         // instantiating the delegates
```

```
        addnum objsum = new addnum(obj.sum);
        subnum objsub = new subnum(obj.sub);

        objsum(num1, num2);
        objsub(num1, num2);


        Console.ReadLine();
    }
  }
}
```

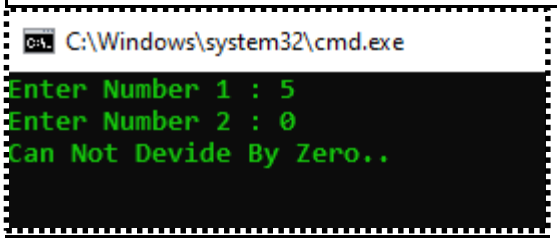
Output:

```
Enter Number 1 : 100
Enter Number 2 : 99

100 + 99=199
100 - 99=1
```

2) C) ii) Exception handling

Source Code

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

namespace ConsoleApp4
{
   internal class Program
   {

      static void Main(string[] args)
      {
        int num1, num2,div;
          Console.Write("Enter Number 1 : ");
          num1 = Convert.ToInt32(Console.ReadLine());
          Console.Write("Enter Number 2 : ");
          num2 = Convert.ToInt32(Console.ReadLine());
          try
          {
             div = num1 / num2;
             Console.Write("Division : "+div);

          }
          catch (ArithmeticException e)
          {
             Console.WriteLine("Can Not Devide By Zero..");
          }
          Console.ReadLine();
      }
   }
}
```



PRACTICAL NO 3: Working with Web Forms and Controls

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

Source code:

.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Prac3a.aspx.cs"
Inherits="prac3.webform1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body id="Name">
   <form id="form1" runat="server">
      <div>
         Student Registation Form_6417<br />
         <br />
         <br />
         <asp:Label ID="Label1" runat="server" Text="NAME"></asp:Label>
         <asp:TextBox ID="NAME_TB" runat="server"
OnTextChanged="TextBox1_TextChanged"></asp:TextBox>
         <br />
         <asp:Label ID="Label2" runat="server" Text="ROLL NO"></asp:Label>
         <asp:TextBox ID="ROLLNO_TB" runat="server"
OnTextChanged="TextBox2_TextChanged"></asp:TextBox>
         <br />
         <asp:Label ID="Label3" runat="server" Text="SECTION"></asp:Label>
         <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
         <br />
         <br />
         <br />
         Language Known<asp:RadioButtonList ID="lung_RBL" runat="server">
            <asp:ListItem>Java</asp:ListItem>
            <asp:ListItem>Python</asp:ListItem>
            <asp:ListItem>C++</asp:ListItem>
         </asp:RadioButtonList>
```

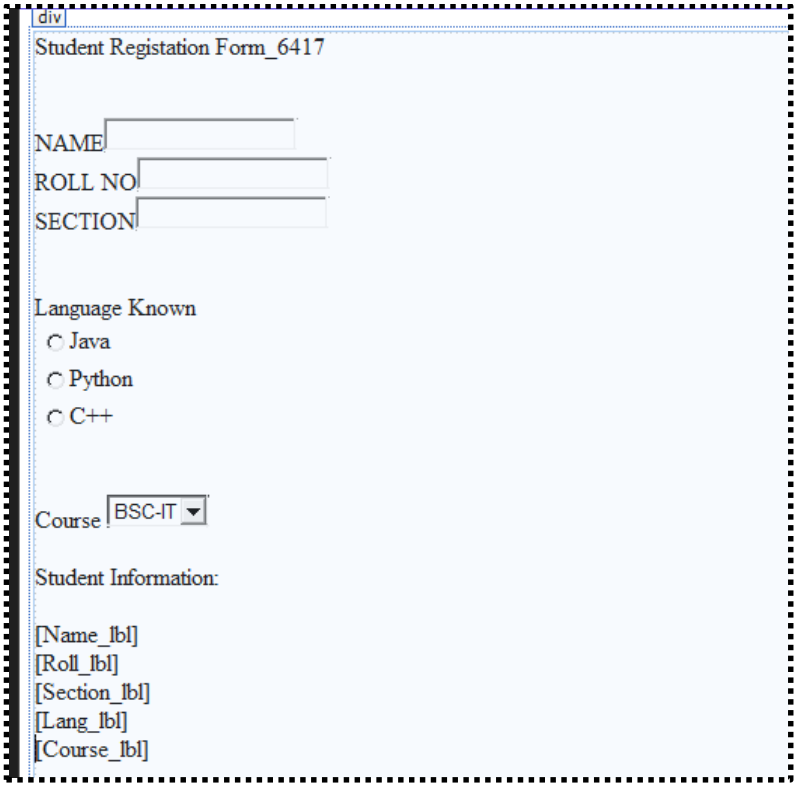
```
        <br />
        <br />
        Course <asp:DropDownList ID="DropDownList1" runat="server">
           <asp:ListItem>BSC-IT</asp:ListItem>
        </asp:DropDownList>
        <br />
        <br />
        Student Information:<br />
        <br />
        <asp:Label ID="Name_lbl" runat="server"></asp:Label>
        <br />
        <asp:Label ID="Roll_lbl" runat="server"></asp:Label>
        <br />
        <asp:Label ID="Section_lbl" runat="server"></asp:Label>
        <br />
        <asp:Label ID="Lang_lbl" runat="server"></asp:Label>
        <br />
        <asp:Label ID="Course_lbl" runat="server"></asp:Label>
        <br />
        <br />
        <br />
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Sumit" />
     </div>
   </form>
</body>
</html>
```

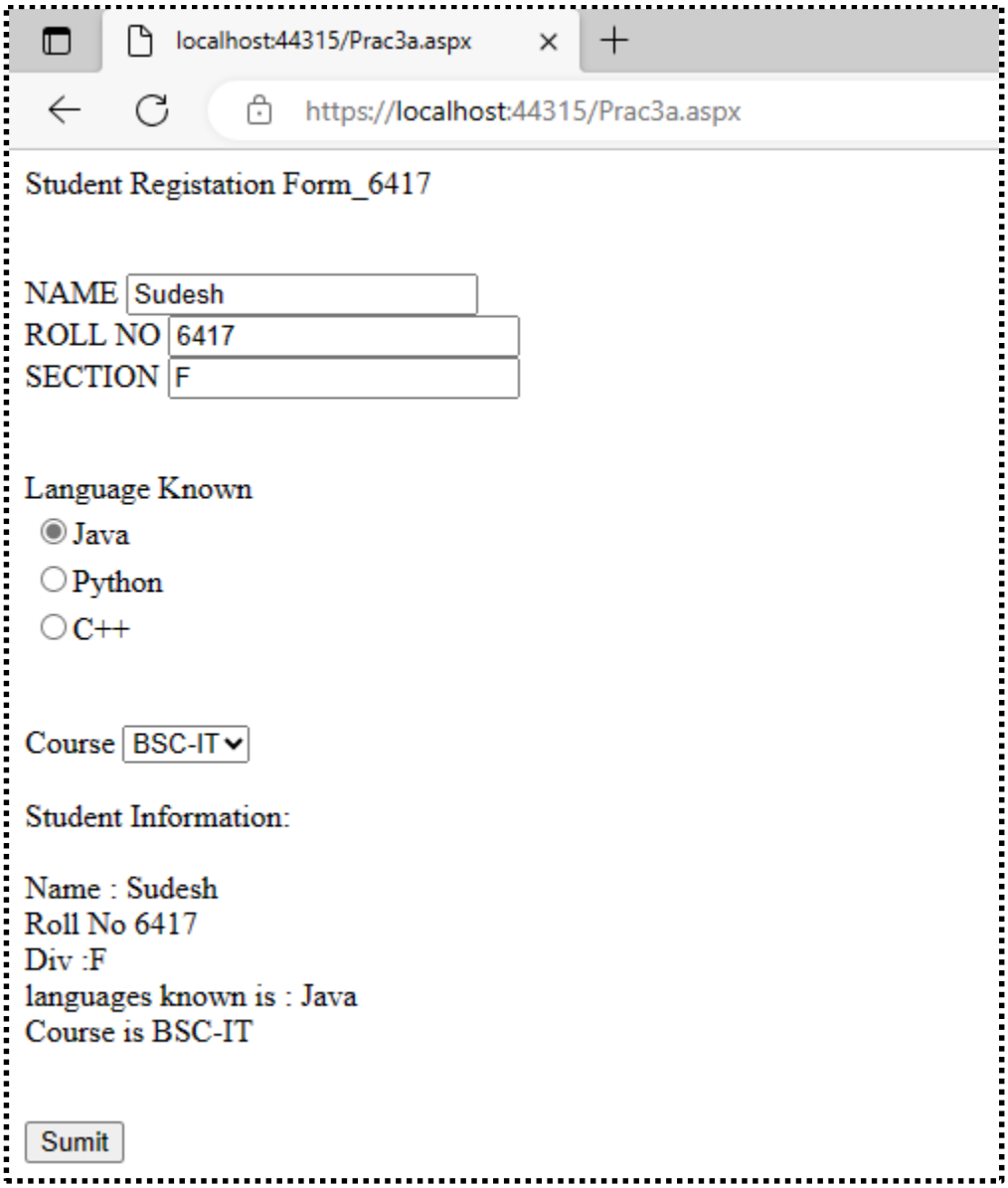
Aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace prac3
{
   public partial class webform1 : System.Web.UI.Page
   {
      protected void Button1_Click(object sender, EventArgs e)
      {
```

```
        Name_lbl.Text = "Name : " + NAME_TB.Text;
        Roll_lbl.Text = "Roll No " + ROLLNO_TB.Text;
        Section_lbl.Text = "Div :" + TextBox3.Text;
        Lang_lbl.Text = "languages known is :\t" + " " + lung_RBL.SelectedValue;
        Course_lbl.Text = "Course is " + DropDownList1.SelectedItem;

    }

    protected void TextBox1_TextChanged(object sender, EventArgs e)
    {

    }

    protected void TextBox2_TextChanged(object sender, EventArgs e)
    {

    }
  }
}
```

Design:




localhost:44315/Prac3a.aspx
https://localhost:44315/Prac3a.aspx
Student Registation Form_6417
NAME Sudesh
ROLL NO 6417
SECTION F
Language Known
Java
Python
C++
Course BSC-IT
Student Information:
Name : Sudesh
Roll No 6417
Div :F
languages known is : Java
Course is BSC-IT
Sumit

3) 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

webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication_CalendarControl.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:Calendar ID="Calendar1" runat="server"
            OnSelectionChanged="Calendar1_SelectionChanged"
OnDayRender="Calendar1_Render"></asp:Calendar>
      </div>
      <br />
      Today's Date: <asp:Label ID="Today_lb" runat="server" Text="Today_Label"></asp:Label>
      <br />
      Selected Date: <asp:Label ID="Selected_lb" runat="server"
Text="Selected_Label"></asp:Label>
      <br />
      Days Till Selected Date: <asp:Label ID="TillSelected_lb" runat="server"
Text="DaysTillSelected_Label"></asp:Label>
      <br />
      Diwali Date: <asp:Label ID="Diwali_lb" runat="server" Text="Label"></asp:Label>
      <br />
      Days till Diwali Date: <asp:Label ID="TillDiwali_lb" runat="server" Text="Label"></asp:Label>
   </form>
</body>
</html>
```

Webform.aspx.cs

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

namespace WebApplication_CalendarControl
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         Diwali_lb.Text = new DateTime(2023, 11, 10).ToString("dd/MM/yyyy");
         TillDiwali_lb.Text = new DateTime(2023, 11,
10).Subtract(DateTime.Today).Days.ToString();

      }

      protected void Calendar1_SelectionChanged(object sender, EventArgs e)
      {
         Today_lb.Text = DateTime.Today.ToString("dd/MM/yyyy");
         Selected_lb.Text = Calendar1.SelectedDate.ToString("dd/MM/yyyy");
         TillSelected_lb.Text =
Calendar1.SelectedDate.Subtract(DateTime.Today).Days.ToString();
      }

      protected void Calendar1_Render(object sender, DayRenderEventArgs e)
      {
         if (e.Day.IsSelected)
         {
            e.Cell.BackColor = Color.Cyan;
         }
      }
   }
}
```

≤	July 2023					≥
Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
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	5	6

Today's Date: 09-07-2023
Selected Date: 19-07-2023
Days Till Selected Date: 10
Diwali Date: 10-11-2023
Days till Diwali Date: 124

3) C) Demonstrate the use of Treeview control perform following operations.

i) Treeview control and datalist ii) Treeview operations

i) Treeview control and datalist

Treeview.aspx:

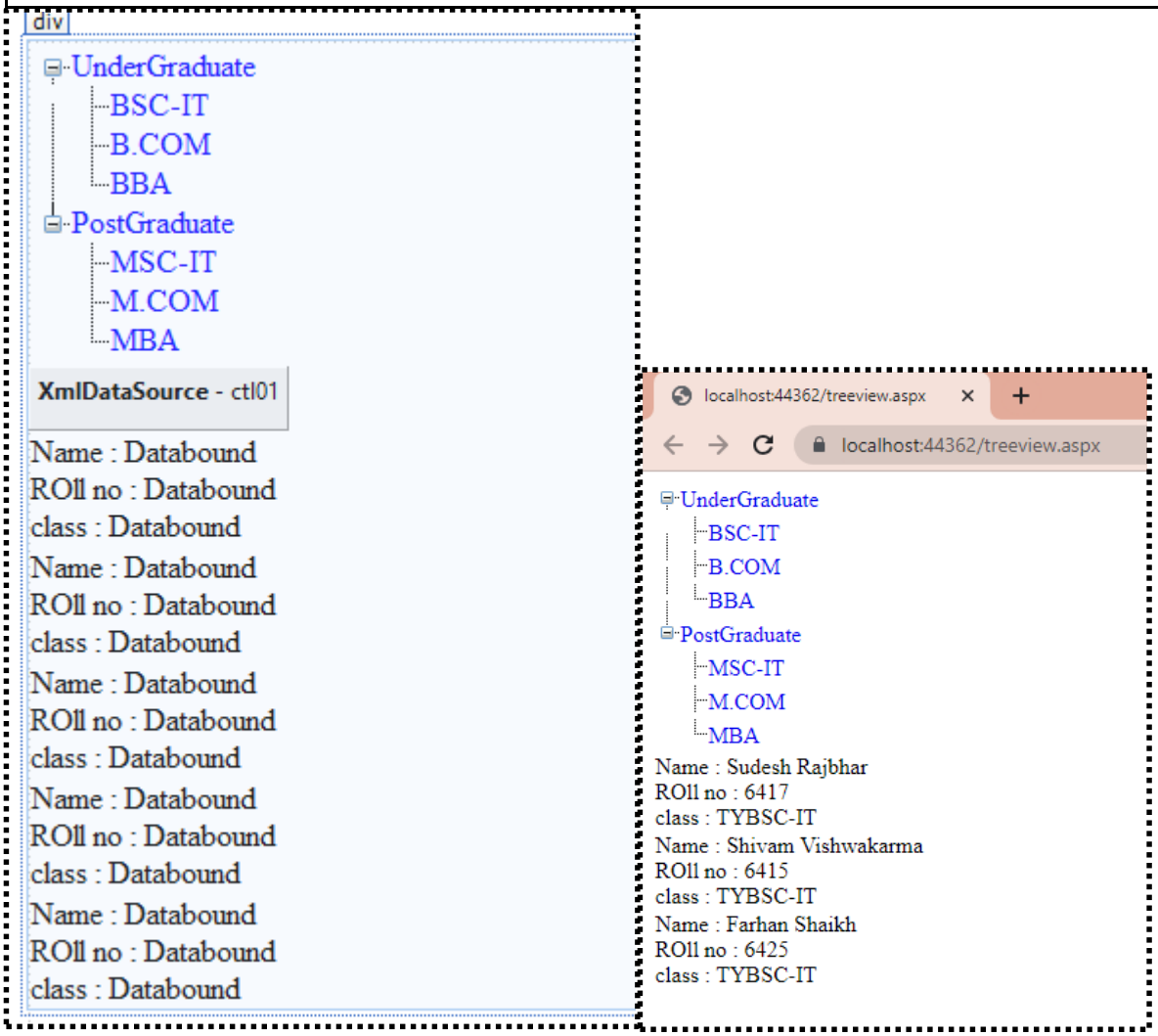
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="treeview.aspx.cs"
Inherits="prac3_3.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:TreeView ID="TreeView1" runat="server" ShowLines="True">
            <Nodes>
               <asp:TreeNode Text="UnderGraduate" Value="UnderGraduate">
                  <asp:TreeNode Text="BSC-IT" Value="BSC-IT"></asp:TreeNode>
                  <asp:TreeNode Text="B.COM" Value="B.COM"></asp:TreeNode>
                  <asp:TreeNode Text="BBA" Value="BBA"></asp:TreeNode>
               </asp:TreeNode>
               <asp:TreeNode Text="PostGraduate" Value="PostGraduate">
                  <asp:TreeNode Text="MSC-IT" Value="MSC-IT"></asp:TreeNode>
                  <asp:TreeNode Text="M.COM" Value="M.COM"></asp:TreeNode>
                  <asp:TreeNode Text="MBA" Value="MBA"></asp:TreeNode>
               </asp:TreeNode>
            </Nodes>
         </asp:TreeView>
         <asp:XmlDataSource runat="server" DataFile="~/XMLFile1.xml"
ID="ctl01"></asp:XmlDataSource>
         <asp:DataList ID="DataList1" runat="server">
            <ItemTemplate>
               Name : <%# Eval ("name") %> <br />
               ROll no : <%# Eval ("roll") %><br />
               class : <%# Eval ("class") %><br />
            </ItemTemplate>
         </asp:DataList>
```

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

Treeview.aspx.cs:

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

namespace prac3_3
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        BindData();
      }
    }
    protected void BindData()
    {
      DataSet ds = new DataSet();
      ds.ReadXml(Server.MapPath("XMLFile1.xml"));
      if (ds != null && ds.HasChanges())
      {
        DataList1.DataSource = ds;
        DataList1.DataBind();
      }
      else
      {
        DataList1.DataBind();
      }

    }
```

```
    }
}
```


div
UnderGraduate
BSC-IT
B.COM
BBA
PostGraduate
MSC-IT
M.COM
MBA
XmlDataSource - ctl01
Name : Databound
ROll no : Databound
class : Databound
Name : Databound
ROll no : Databound
class : Databound
Name : Databound
ROll no : Databound
class : Databound
Name : Databound
ROll no : Databound
class : Databound
Name : Databound
ROll no : Databound
class : Databound
localhost:44362/treeview.aspx
localhost:44362/treeview.aspx
UnderGraduate
BSC-IT
B.COM
BBA
PostGraduate
MSC-IT
M.COM
MBA
Name : Sudesh Rajbhar
ROll no : 6417
class : TYBSC-IT
Name : Shivam Vishwakarma
ROll no : 6415
class : TYBSC-IT
Name : Farhan Shaikh
ROll no : 6425
class : TYBSC-IT

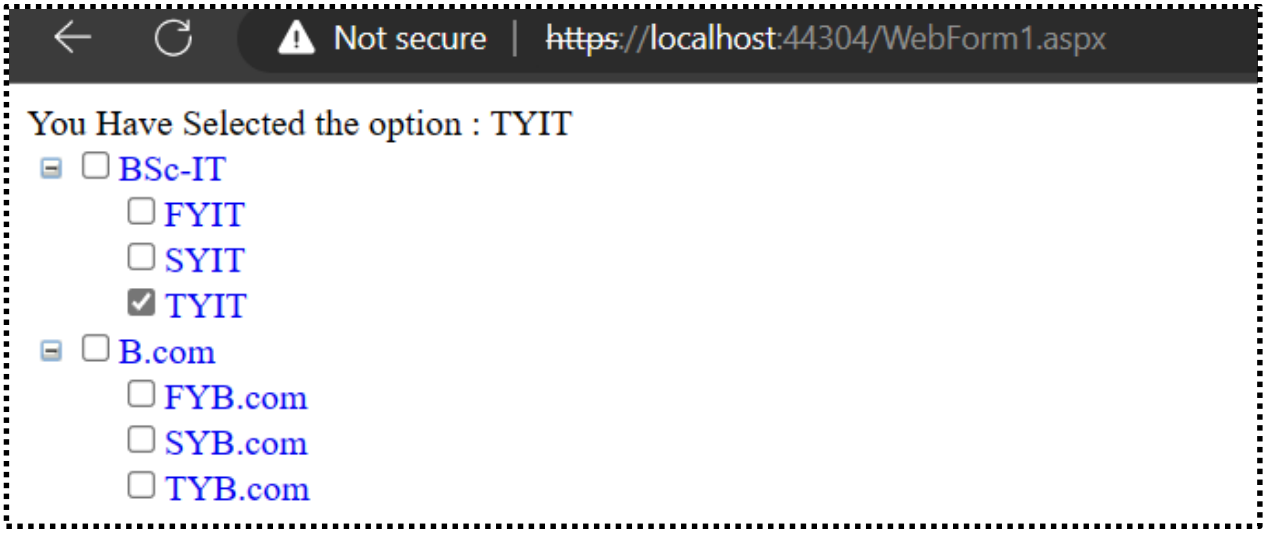
ii . Treeview Operations

WebForm1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication3.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:TreeView ID="TreeView1" runat="server"
OnSelectedNodeChanged="TreeView1_SelectedNodeChanged" OnTreeNodeCollapsed
="TreeView1_TreeNodeCollapsed">
            <Nodes>
               <asp:TreeNode Text="BSc-IT" Value="BSc-IT" ShowCheckBox="True">
                  <asp:TreeNode Text="FYIT" Value="FYIT"
ShowCheckBox="True"></asp:TreeNode>
                  <asp:TreeNode Text="SYIT" Value="SYIT"
ShowCheckBox="True"></asp:TreeNode>
                  <asp:TreeNode Text="TYIT" Value="TYIT"
ShowCheckBox="True"></asp:TreeNode>
               </asp:TreeNode>
               <asp:TreeNode Text="B.com" Value="B.com" ShowCheckBox="True">
                  <asp:TreeNode Text="FYB.com" Value="FYB.com"
ShowCheckBox="True"></asp:TreeNode>
                  <asp:TreeNode Text="SYB.com" Value="SYB.com"
ShowCheckBox="True"></asp:TreeNode>
                  <asp:TreeNode Text="TYB.com" Value="TYB.com"
ShowCheckBox="True"></asp:TreeNode>
               </asp:TreeNode>
            </Nodes>
         </asp:TreeView>
          <br />

      </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 WebApplication3
{
   public partial class WebForm1 : System.Web.UI.Page
   {

      protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
      {
         Response.Write("You Have Selected the option : " + TreeView1.SelectedValue);
      }
      protected void TreeView1_TreeNodeCollapsed(object sender, TreeNodeEventArgs e)
      {
         Response.Write("The Value Collapsed Was " + e.Node.Value);
      }

   }
}
```



PRACTICAL NO 4.Working with Form Controls

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

webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="prac4.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Name :
         <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
         <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="Name is required!! please enter name.."
ForeColor="Red"></asp:RequiredFieldValidator>
         <br />
         <br />
         Age :
         <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
         <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="Age is required!! please enter age.."
ForeColor="Red"></asp:RequiredFieldValidator>
         <asp:RangeValidator ID="RangeValidator1" runat="server" ErrorMessage="Enter
Valid Age !!" ForeColor="Red" MaximumValue="100" MinimumValue="15" Type="Integer"
ControlToValidate="TextBox2"></asp:RangeValidator>
         <br />
         <br />
         Email :
         <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
         <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
ControlToValidate="TextBox3" ErrorMessage="Email is required!! please enter Email.."
```

```
ForeColor="Red"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ControlToValidate="TextBox3" ErrorMessage="  Please Enter Valid
Email!! use [@, .]" ForeColor="Red" ValidationExpression="\w+([-+.']\w+)*@\w+([-
.]\w+)*\.\w+([-.]\w+)*"></asp:RegularExpressionValidator>
        <br />
        <br />
        Password :
        <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
ControlToValidate="TextBox4" ErrorMessage="Password is required!! " ForeColor="Red"
Type="Integer"></asp:RequiredFieldValidator>
        <br />
        <br />
        Confirm password :
        <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator5" runat="server"
ControlToValidate="TextBox5" ErrorMessage="Please Confirm the password!!!"
ForeColor="Red" Type="Integer"></asp:RequiredFieldValidator>
        <asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="TextBox4" ControlToValidate="TextBox5" ErrorMessage="Please
enter valid password !!check password again.." ForeColor="Red"
Type="Integer"></asp:CompareValidator>
        <br />
        <br />
        <asp:Button ID="Button1" runat="server"  Text="submit"
OnClick="Button1_Click" />
        <br />
        <asp:ValidationSummary ID="ValidationSummary1" runat="server" />
        <br />
        <asp:Label ID="Label1" runat="server"></asp:Label>
        <br />
    </div>
  </form>
</body>
</html>
```

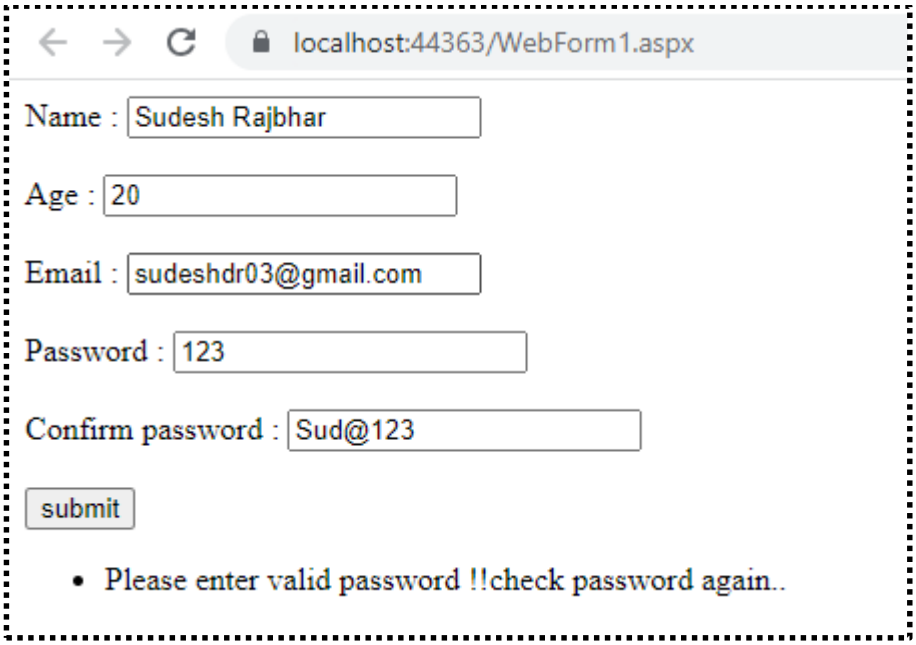
Webform1.aspx.cs:

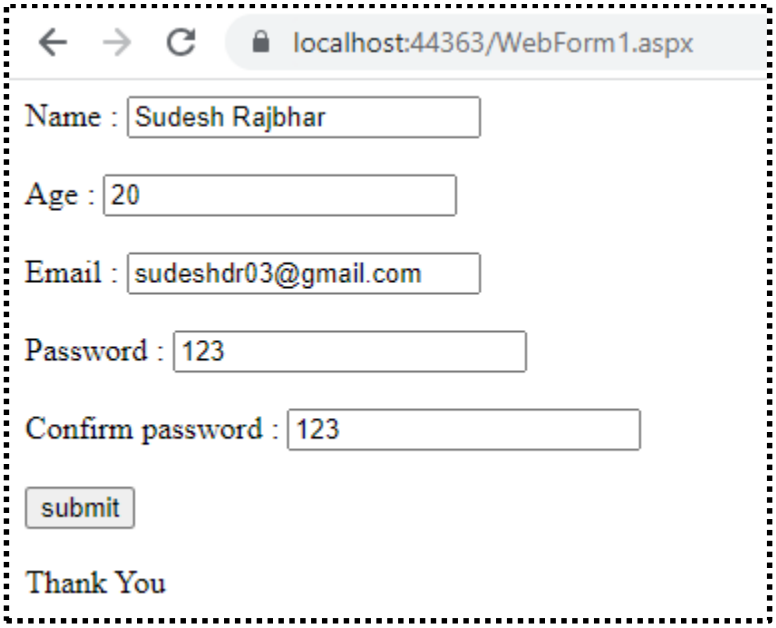
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Emit;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace prac4
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {

         if (Page.IsValid)
         {

            Label1.Text = "Thank You";
         }
         else
         {
            Label1.Text = "The text must be exactly 8 characters long!";
         }
      }
      void ServerValidation(object source, ServerValidateEventArgs e)
      {
         if (e.Value.Length == 8)
            e.IsValid = true;
         else
            e.IsValid = false;
      }
   }

}
```

Web.config:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
        <appSettings>
                <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
        </appSettings>

        <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
/define:_MYTYPE=\&amp;quot;Web\&amp;quot; /optionInfer+" />
    </compilers>
  </system.codedom>
</configuration>
```

Output:


localhost:44363/WebForm1.aspx
Name : Sudesh Rajbhar
Age : 20
Email : sudeshdr03@gmail.com
Password : 123
Confirm password : Sud@123
submit
Please enter valid password !!check password again..


localhost:44363/WebForm1.aspx
Name : Sudesh Rajbhar
Age : 20
Email : sudeshdr03@gmail.com
Password : 123
Confirm password : 123
submit
Thank You

4) B) Create a Web Form to demonstrate use of Adrotator Control.

XmlFile1.xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<Advertisements>
	<Ad>
		<ImageUrl>pikachu.jpg</ImageUrl>
		<NavigateUrl>https://www.pokemon.com/us/pokedex/pikachu</NavigateUrl>
		<AlternateText>
			Pikachu from Pokemon
		</AlternateText>
		<Impressions>1</Impressions>
		<Keyword>ASH KETCHUM</Keyword>
	</Ad>
	<Ad>
		<ImageUrl>eren.jpg</ImageUrl>
		<NavigateUrl>https://attackontitan.fandom.com/wiki/Eren_Yeager</NavigateUrl>
		<AlternateText>Eren yeager from attack on titan</AlternateText>
		<Impressions>2</Impressions>
		<Keyword>AOT</Keyword>
	</Ad>
	<Ad>
		<ImageUrl>tanjiro.jpg</ImageUrl>
		<NavigateUrl>https://kimetsu-no-
yaiba.fandom.com/wiki/Tanjiro_Kamado</NavigateUrl>
		<AlternateText>Tanjiro from demon slayer</AlternateText>
		<Impressions>3</Impressions>
		<Keyword>DEMON SLAYER</Keyword>
	</Ad>
</Advertisements>
```

Adrotator.aspx:

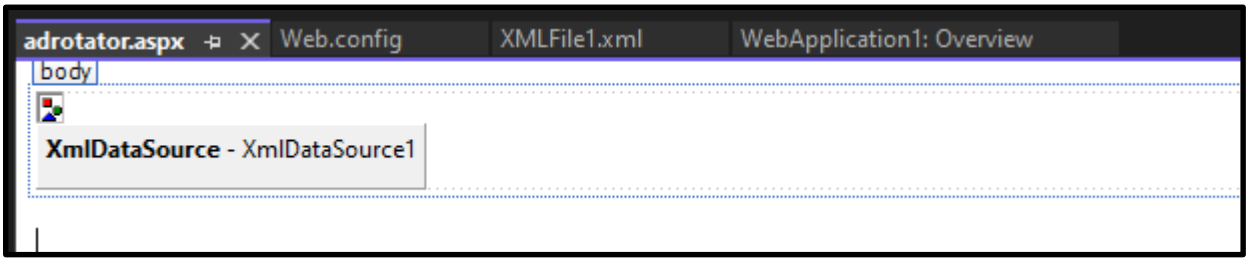
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="adrotator.aspx.cs"
Inherits="WebApplication1.adrotator" %>

<!DOCTYPE html>

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

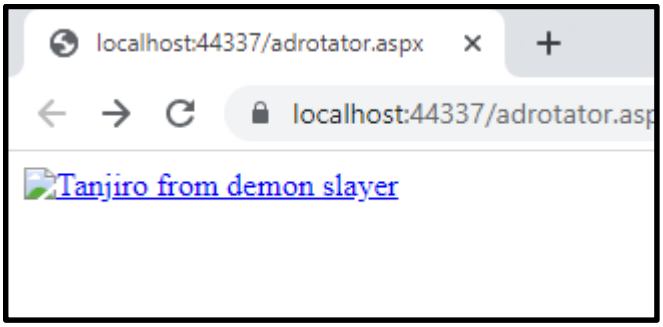
```
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:AdRotator ID="AdRotator1" runat="server" DataSourceID="XmlDataSource1" />
         <asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/XMLFile1.xml"></asp:XmlDataSource>
      </div>
   </form>
</body>
</html>
```

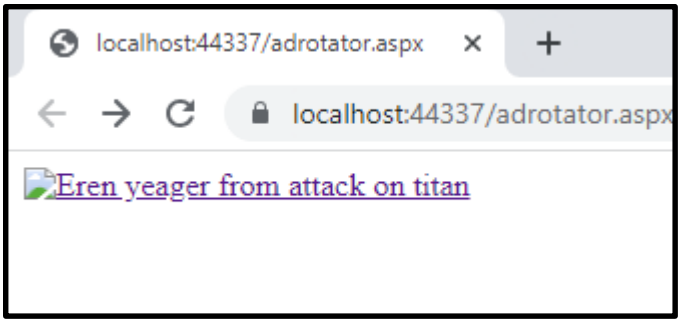
Web.config:

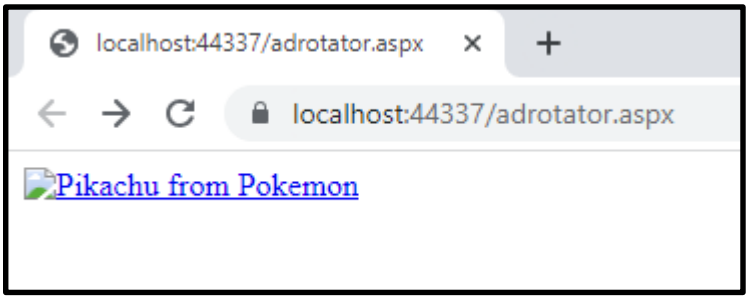
```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  https://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default
/nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" warningLevel="4" compilerOptions="/langversion:default
/nowarn:41008 /define:_MYTYPE=\&amp;quot;Web\&amp;quot; /optionInfer+" />
    </compilers>
  </system.codedom>
</configuration>
```

Output:


adrotator.aspx
Web.config
XMLFile1.xml
WebApplication1: Overview
body
XmlDataSource - XmlDataSource1


localhost:44337/adrotator.aspx
localhost:44337/adrotator.asp
Tanjiro from demon slayer


localhost:44337/adrotator.aspx
localhost:44337/adrotator.aspx
Eren yeager from attack on titan


localhost:44337/adrotator.aspx
localhost:44337/adrotator.aspx
Pikachu from Pokemon

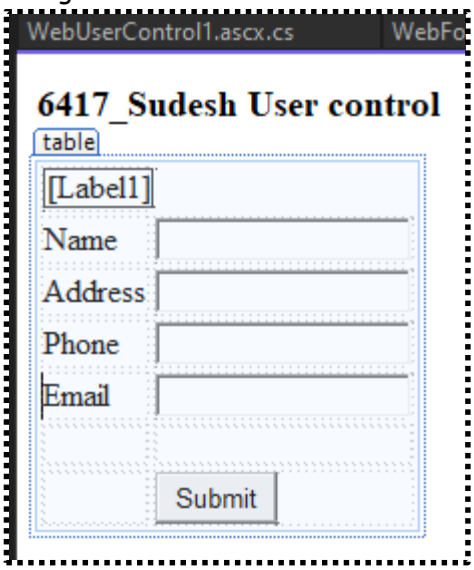
4(c) Create Web Form to demonstrate use of User Controls.

Web UserControl1.ascx:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="WebUserControl1.ascx.cs"
Inherits="practical4.WebUserControl1" %>
<h3>6417_Sudesh User control </h3>
<table>
  <tr>
    <td>
    <fieldset>
      <asp:Label ID="Label1" runat="server" ></asp:Label>
    </fieldset>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="Label2" runat="server" Text="Name"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="Label3" runat="server" Text="Address"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="Label4" runat="server" Text="Phone"></asp:Label>
    </td>
    <td>
      <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
    </td>
  </tr>
  <tr>
    <td>
      <asp:Label ID="Label5" runat="server" Text="Email"></asp:Label>

    </td>
    <td>
```

```
        <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td>
        <asp:Button ID="txtSubmit" runat="server" Text="Submit" OnClick="txtSubmit_Click" />
      </td>
</table>
```

Design:



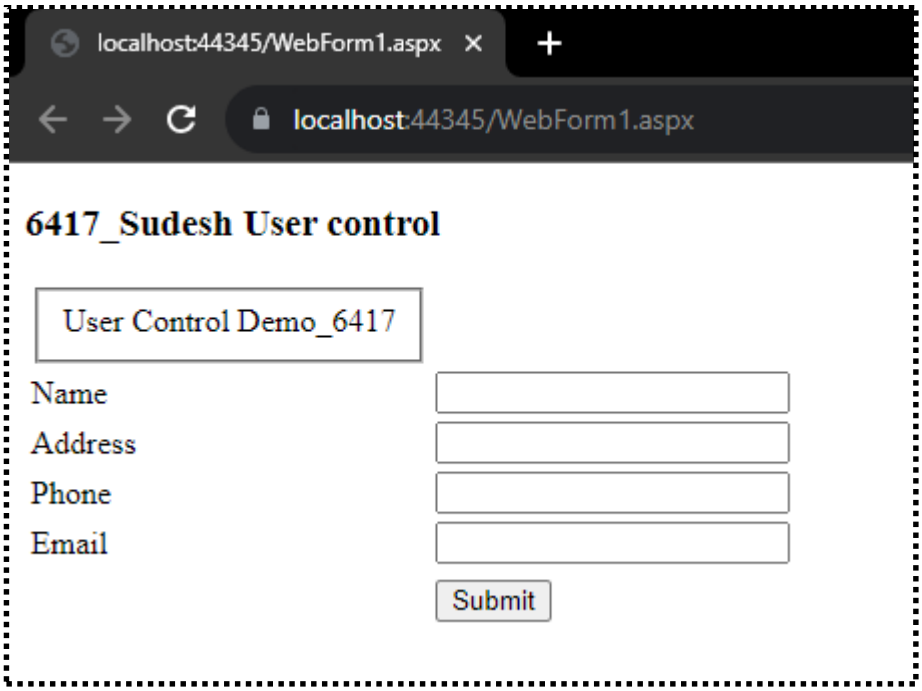
WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="practical4.WebForm1" %>

<%@ Register Src="~/WebUserControl1.ascx" TagPrefix="uc1" TagName="WebUserControl1" %>


<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <uc1:WebUserControl1 runat="server" id="WebUserControl1" Header="User Control
Demo_6417" />
      </div>
   </form>
</body>
</html>
```

WebUserControl1.aspx.cs:

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

namespace practical4
{
   public partial class WebUserControl1 : System.Web.UI.UserControl
   {
      private string _header;
      public string strURL;

      public string Header
      {
         get { return _header; }
```

```
        set { _header = value; }
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = _header;
    }

    protected void txtSubmit(object sender, EventArgs e)
    {

        response.Redirect();
    }
  }
}
```



PRACTICAL NO 5: Working with Navigation, Beautification and Master page.

5) A) Create Web Form to demonstrate use of Website Navigation controls and Site Map.

web.sitemap:

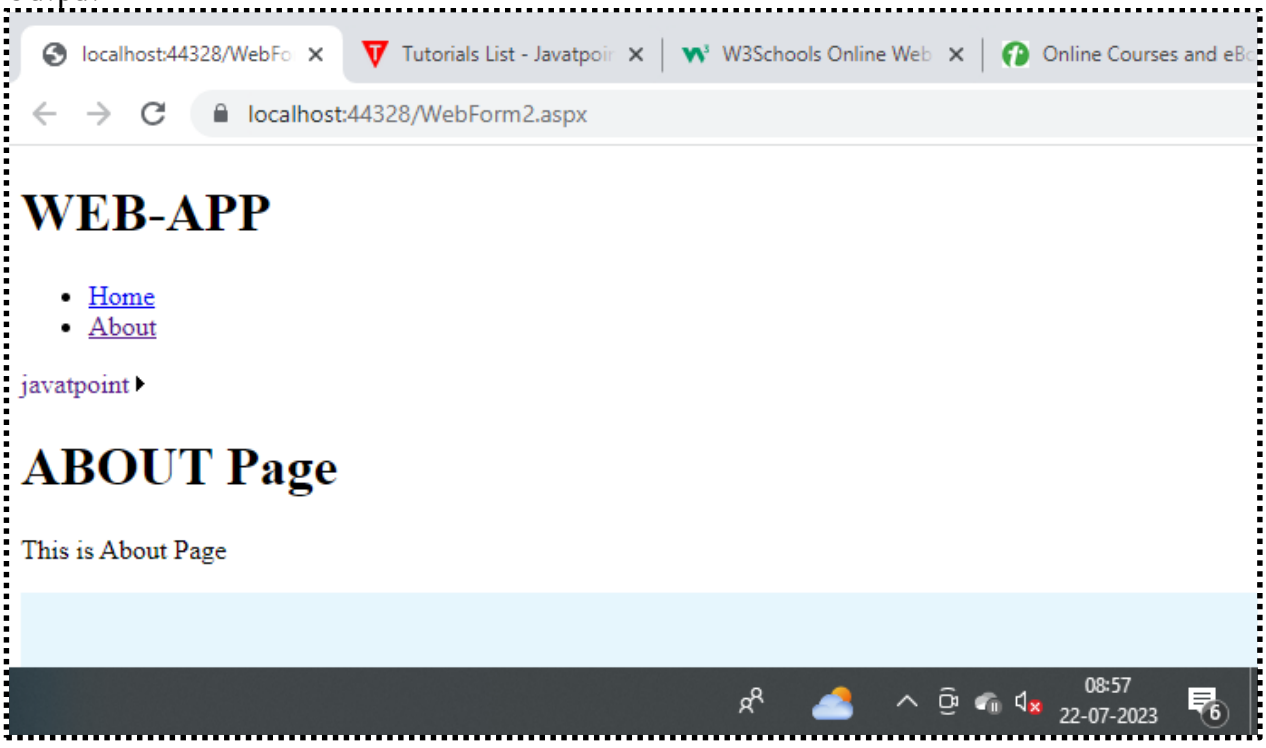
```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="practical5.WebForm2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
   <h1>ABOUT Page</h1>
   <p>
      <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
      This is About Page</p>
   <img src="img/About.png" />

</asp:Content>
```

Master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.Master.cs"
Inherits="practical5.Site1" %>
<!DOCTYPE html>
<html>
<head runat="server">
   <title></title>
   <asp:ContentPlaceHolder ID="head" runat="server">
   </asp:ContentPlaceHolder>
</head>
<body>
   <header>
      <h1>WEB-APP</h1>
      <ul>
         <li><a href="WebForm1.aspx">Home</a></li>
         <li><a href="WebForm2.aspx">About</a></li>
```

```
    </ul>
  </header>
  <form id="form1" runat="server">
    <div>
      <asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" />
  <asp:Menu ID="Menu1" runat="server"
DataSourceID="SiteMapDataSource1"></asp:Menu>

      <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

        <h1>This is header page</h1>
        <p>About</p>
        <image>
          <img src="img/About.png" /></image>
      </asp:ContentPlaceHolder>
    </div>

  </form>
  <footer>&copy; 6417_Sudesh</footer>
</body>
</html>
```

Output:



5) B) Create Web Form to demonstrate use of Website Navigation controls and Site Map.

Site1.Master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.Master.cs"
Inherits="practical5.Site1" %>

<!DOCTYPE html>

<html>
<head runat="server">
   <title></title>
   <asp:ContentPlaceHolder ID="head" runat="server">
   </asp:ContentPlaceHolder>
</head>
<body>

   <header>
      <h1>WEB-APP</h1>

      <ul>

         <li><a href="WebForm1.aspx">Home</a></li>
         <li><a href="WebForm2.aspx">About</a></li>

      </ul>

   </header>

   <form id="form1" runat="server">
      <div>
         <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

            <h1>This is header page</h1>
            <p>About</p>
            <image></image>
         </asp:ContentPlaceHolder>
      </div>
   </form>
   <footer>&copy; 6417_Sudesh</footer>

</body>
</html>
```

WebForm1.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="practical5.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">

   <h1>Home Page</h1>
   <p>This is Home Page</p>
   <img src="img/home.jpg" />

</asp:Content>
```

WebForm2.aspx:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="practical5.WebForm2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
   <h1>ABOUT Page</h1>
   <p>This is About Page</p>
   <img src="img/About.png" />
</asp:Content>
```

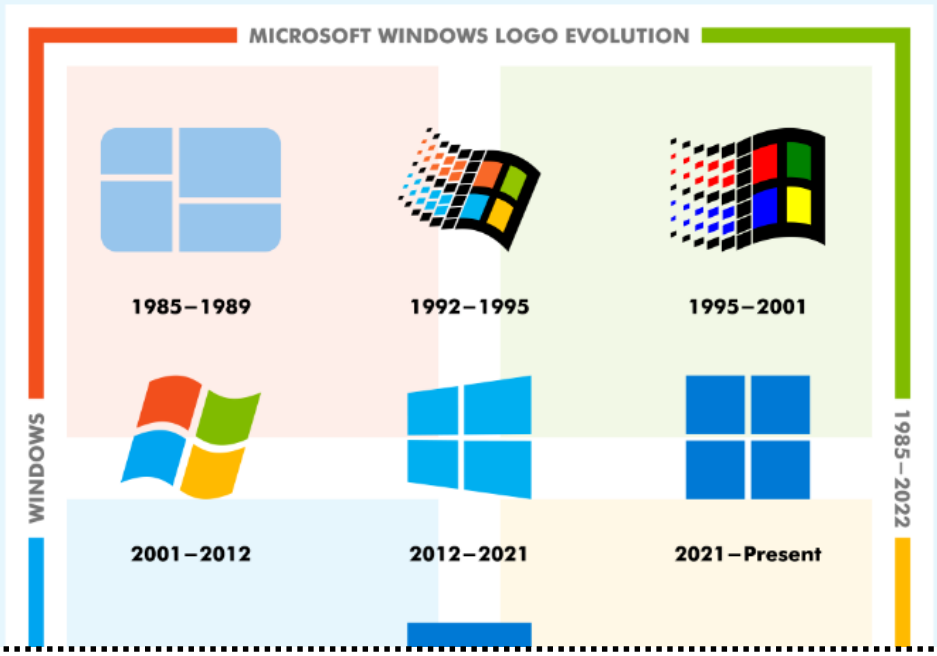
Output:

WEB-APP

- Home
- About

ABOUT Page

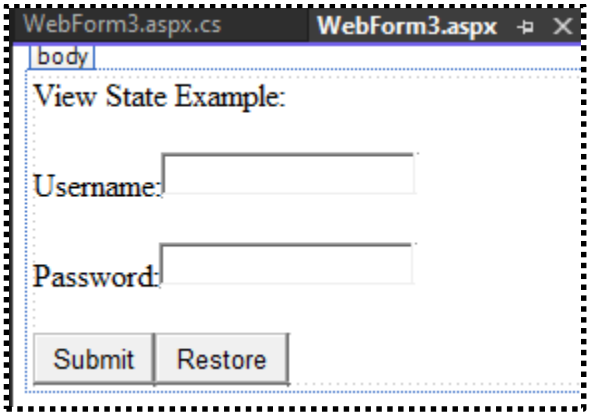
This is About Page



5) C) Create a web application to demonstrate various states of ASP.NET Pages.

1) Viewstate

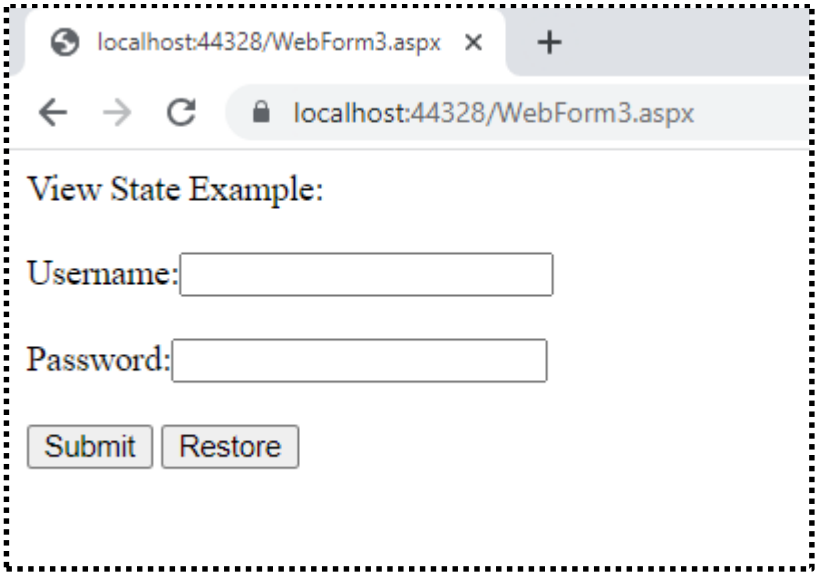
WebForm3.aspx:

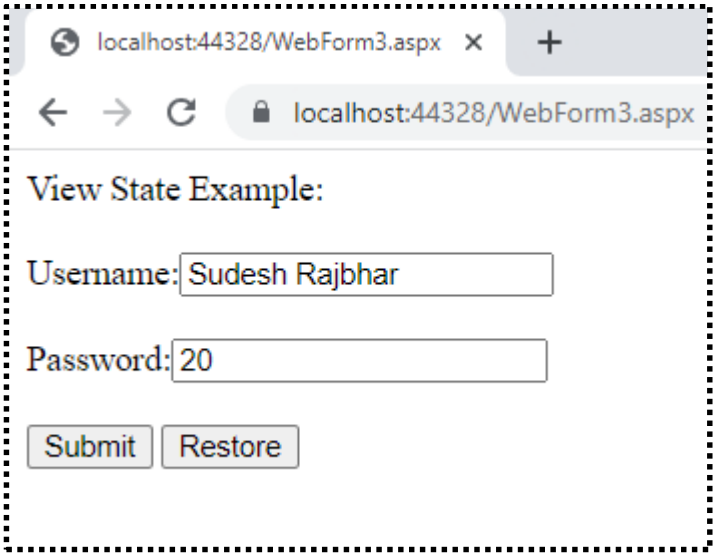


WebForm.aspx.cs:

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

namespace practical5
{
   public partial class WebForm3 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         ViewState["name"] = TextBox1.Text;
         ViewState["password"] = TextBox2.Text;
         TextBox1.Text = TextBox2.Text = string.Empty;
      }

      protected void Button2_Click(object sender, EventArgs e)
      {
         if (ViewState["name"] != null)
         {
            TextBox1.Text = ViewState["name"].ToString();
```

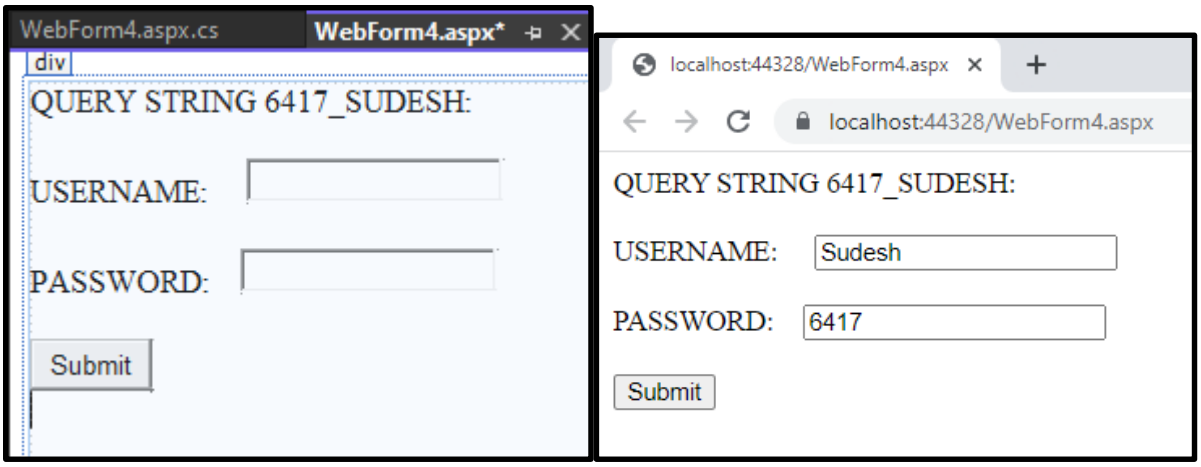
```
        }
        if (ViewState["pass"] != null)
        {
            TextBox2.Text = ViewState["pass"].ToString();
        }

    }
  }
}
```





2) QueryString.

WebForm4.aspx:



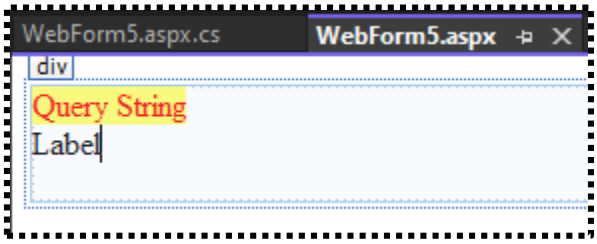
WebForm4.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm4.aspx.cs"
Inherits="practical5.WebForm4" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         QUERY STRING 6417_SUDESH:<br />
         <br />
         USERNAME:    
         <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
         <br />
         <br />
         PASSWORD:   
         <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
         <br />
         <br />
         <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Submit" />
         <br />
         <br />
      </div>
   </form>
</body>
</html>
```

WebForm.aspx.cs:

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

namespace practical5
{
   public partial class WebForm4 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         Response.Redirect("WebForm5.aspx?UserName" + TextBox1.Text + "&Password" +
TextBox2.Text);
      }
   }
}
```

WebForm5.aspx:



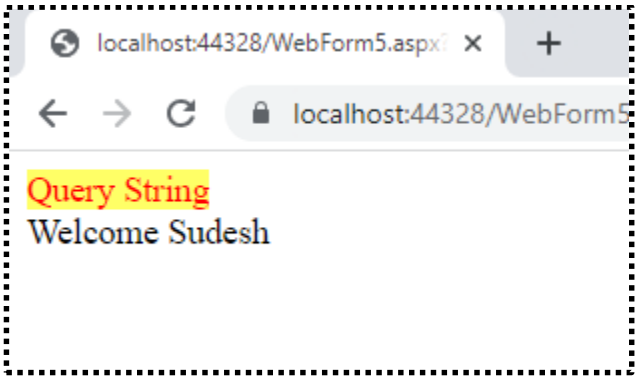
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm5.aspx.cs"
Inherits="practical5.WebForm5" %>

<!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" BackColor="#FFFF66" ForeColor="Red"
Text="Query String"></asp:Label>
         <br />
         <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
         <br />
      </div>
   </form>
</body>
</html>
```

Webform.aspx.cs:

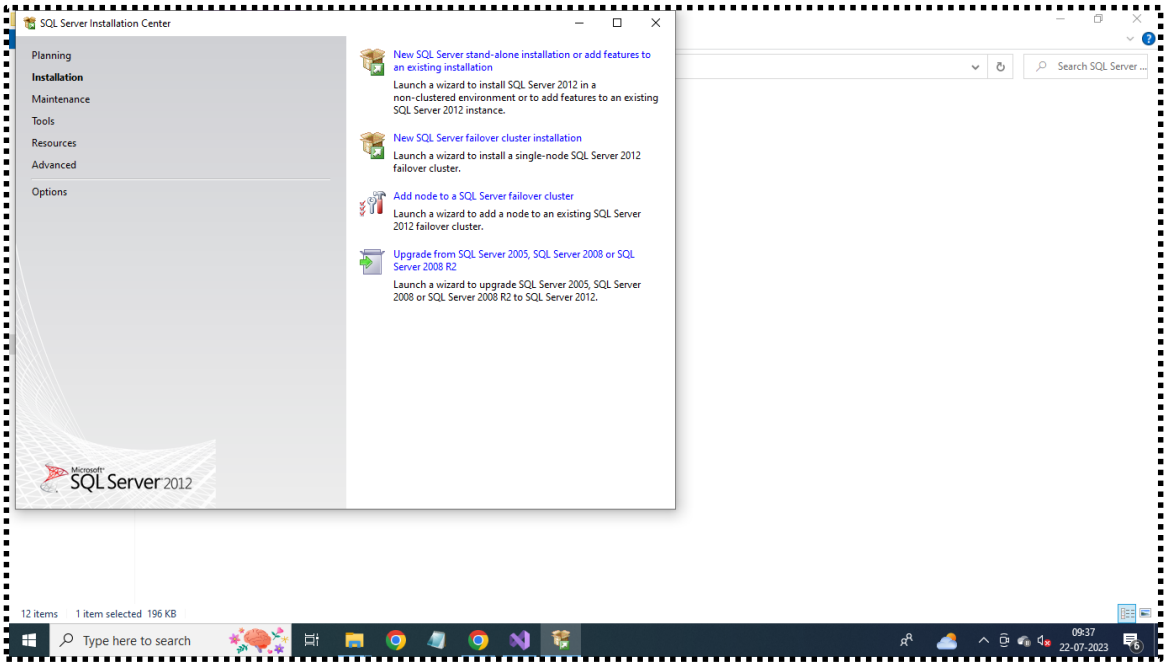
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace practical5
{
   public partial class WebForm5 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         String Username;
         Username=Request.QueryString["username"];
         String Password;
         Password =Request.QueryString["password"];

         Label2.Text = "Welcome" + Username;
```

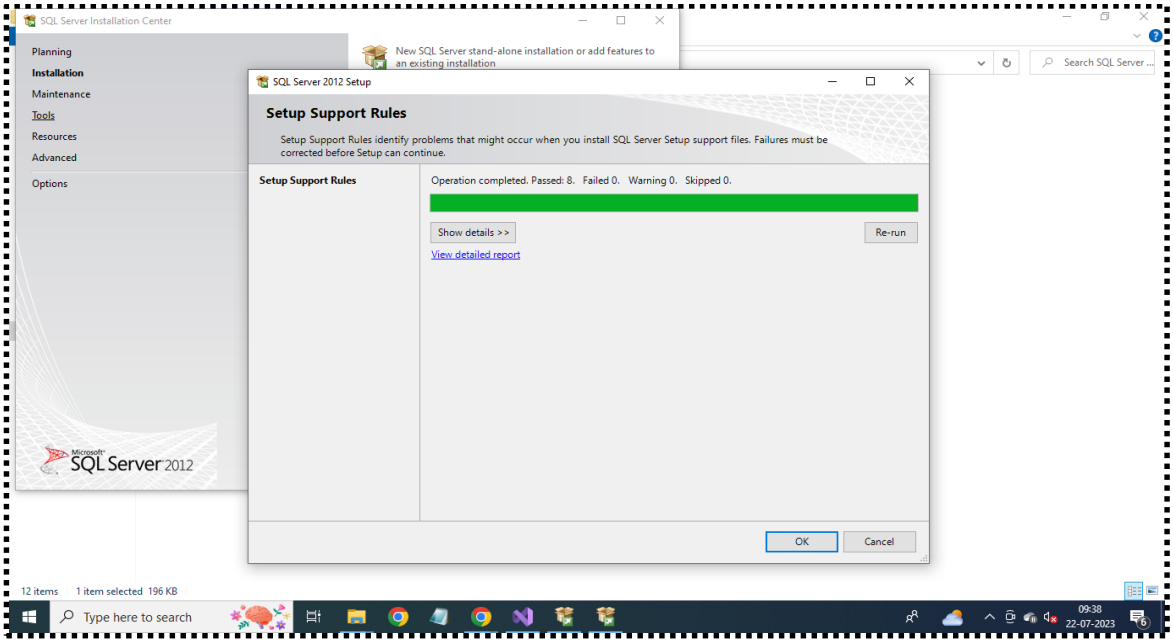
```
    }
  }
}
```

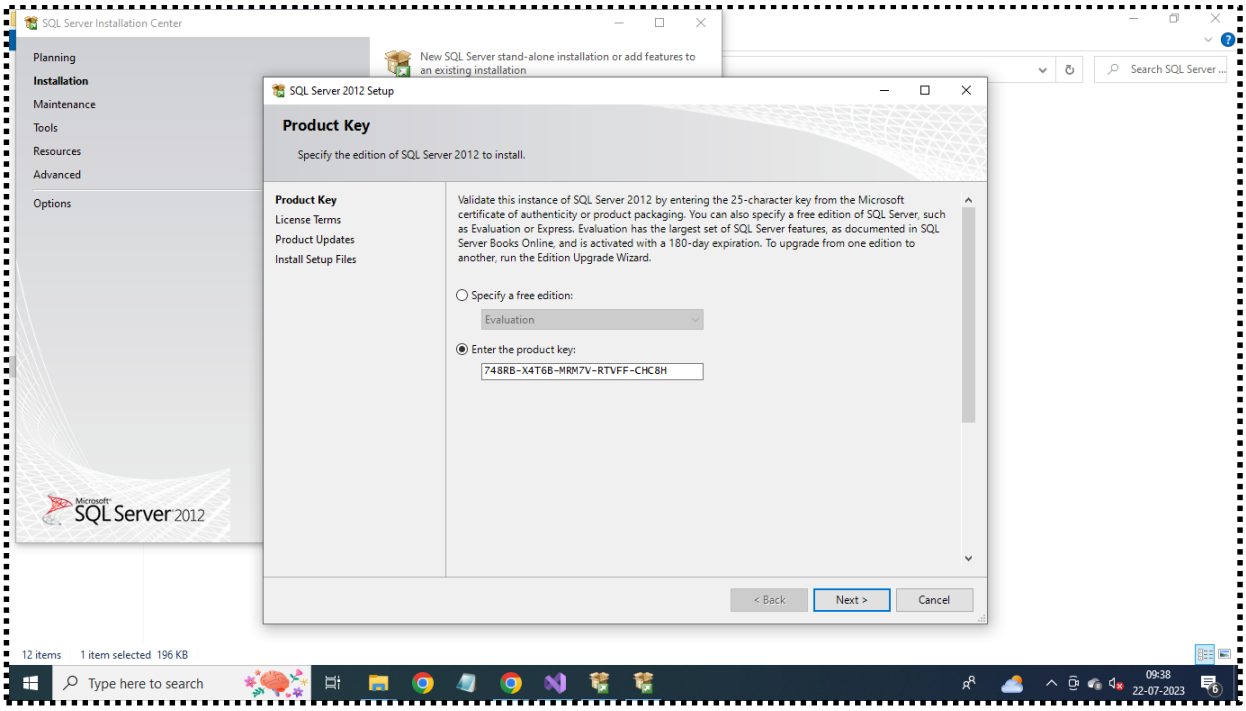
Output:

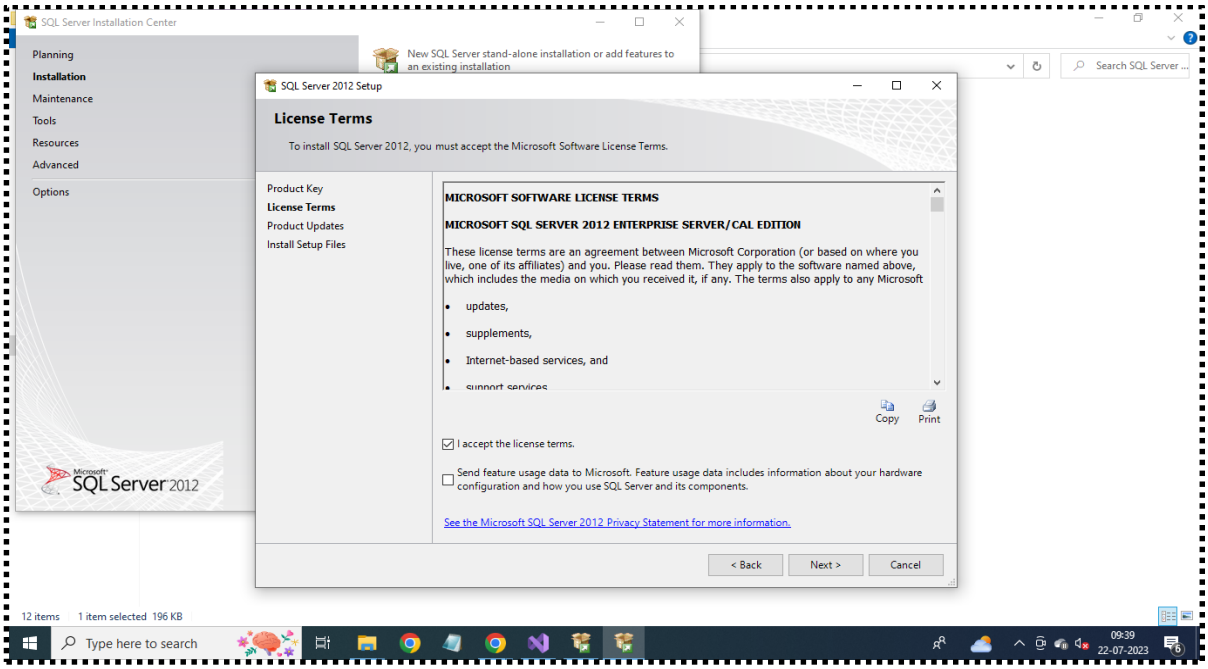

localhost:44328/WebForm5.aspx
localhost:44328/WebForm5
Query String
Welcome Sudesh

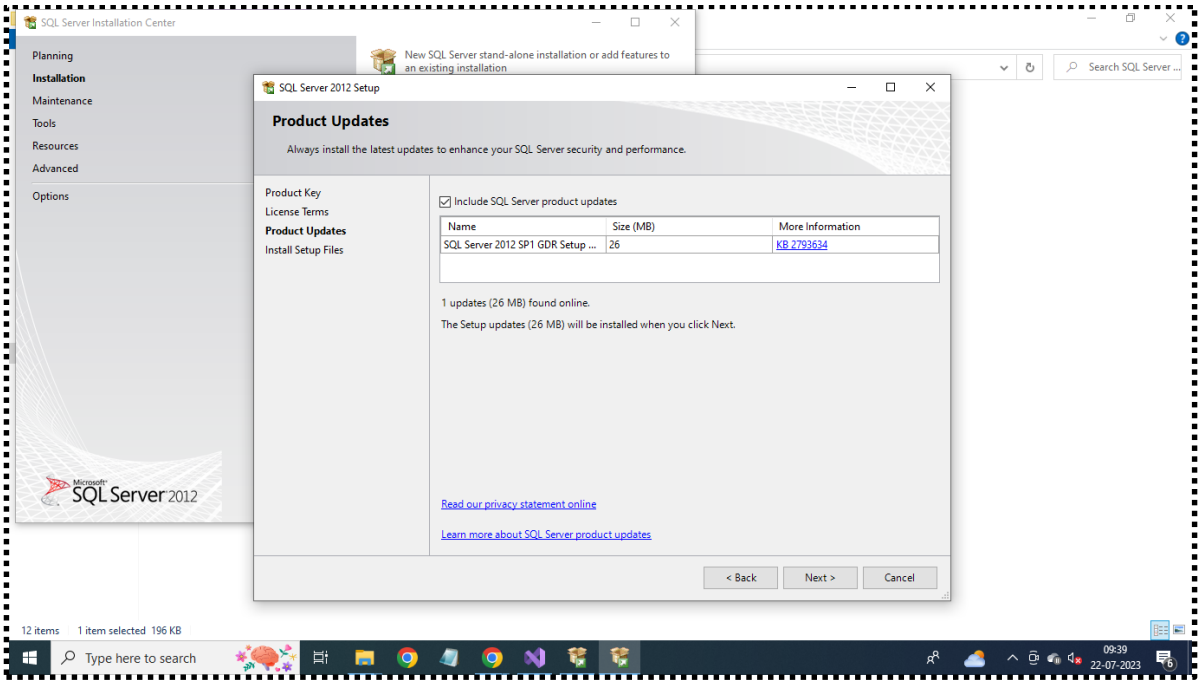
PRACTICAL NO: 6 Working with Database.

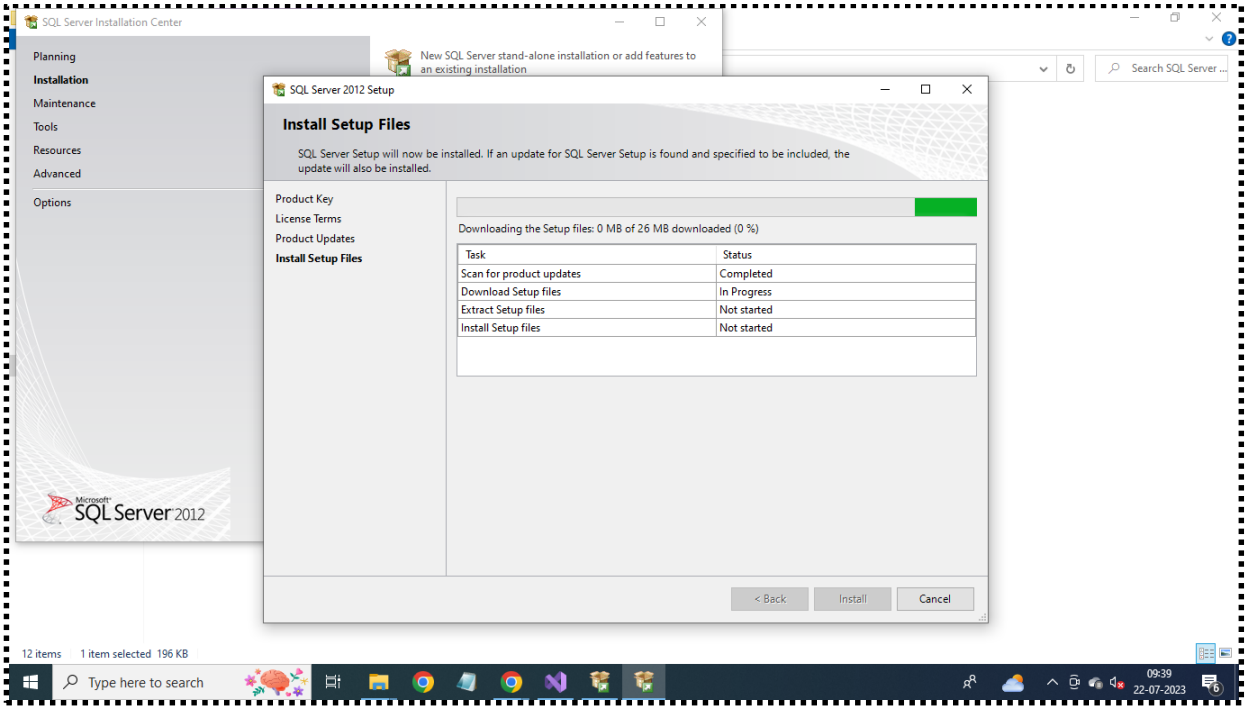


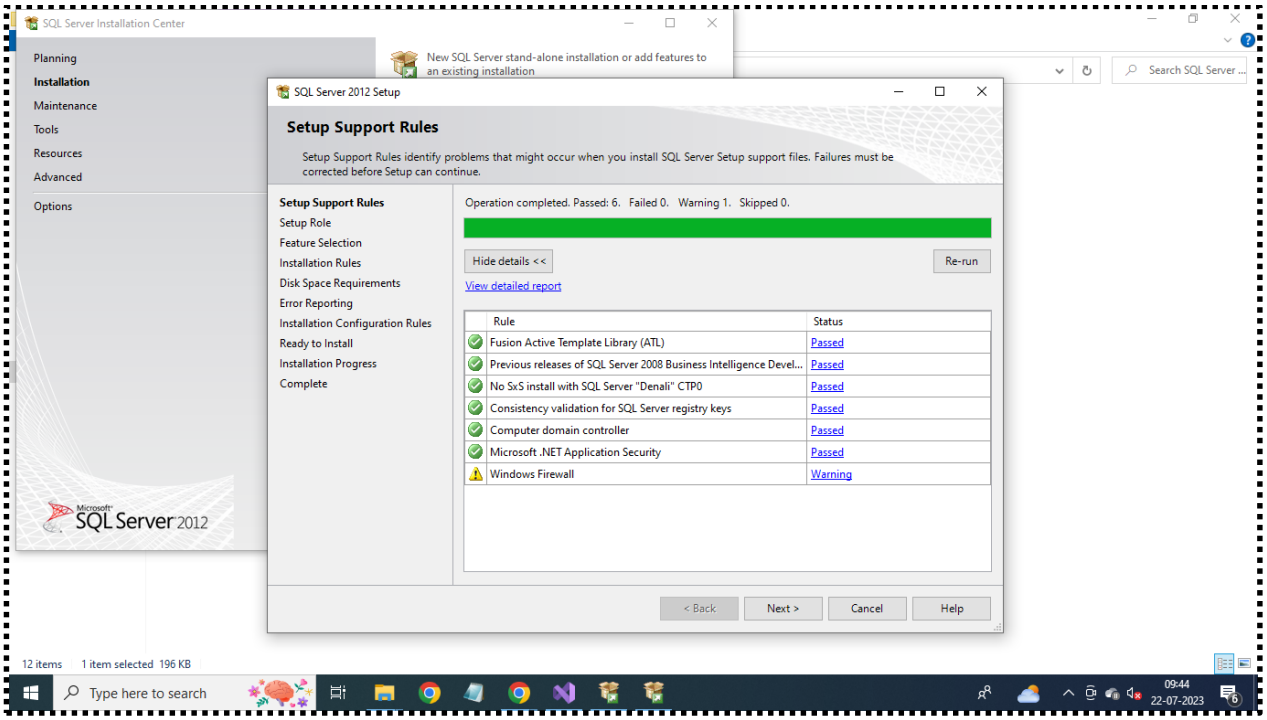


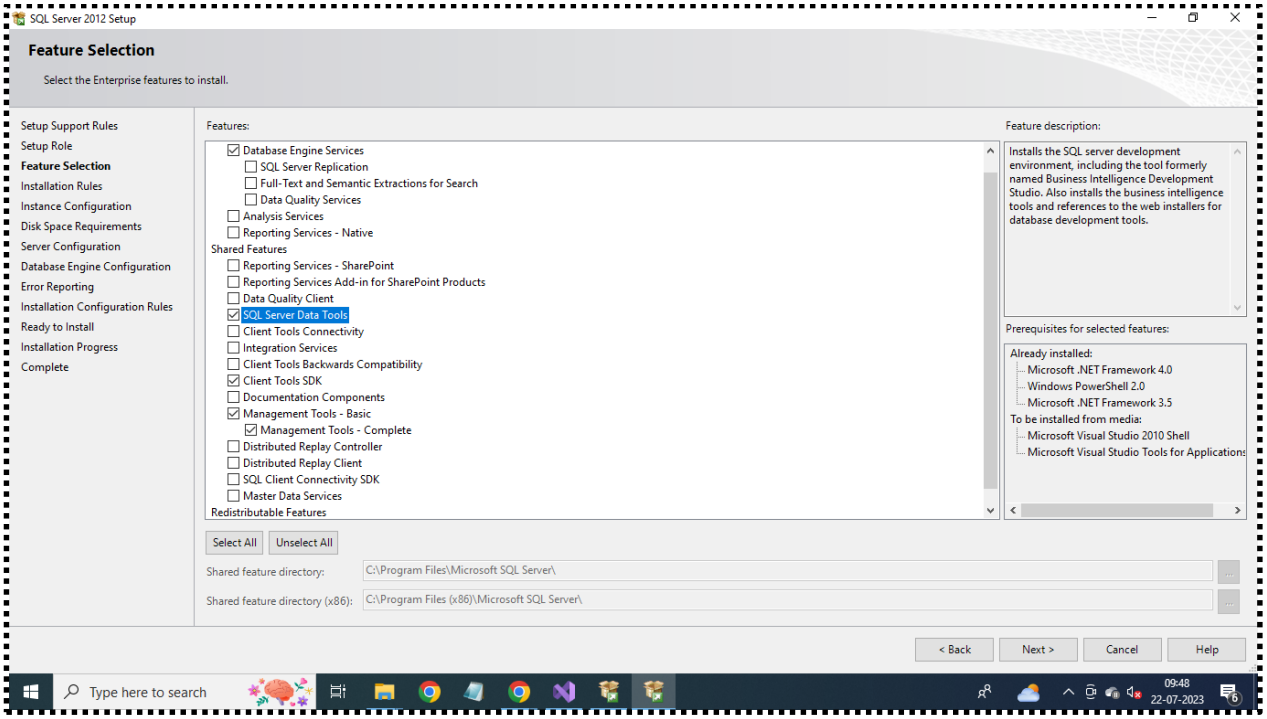

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server stand-alone installation or add features to an existing installation
SQL Server 2012 Setup
Product Key
Specify the edition of SQL Server 2012 to install.
Product Key
License Terms
Product Updates
Install Setup Files
Validate this instance of SQL Server 2012 by entering the 25-character key from the Microsoft certificate of authenticity or product packaging. You can also specify a free edition of SQL Server, such as Evaluation or Express. Evaluation has the largest set of SQL Server features, as documented in SQL Server Books Online, and is activated with a 180-day expiration. To upgrade from one edition to another, run the Edition Upgrade Wizard.
Specify a free edition:
Evaluation
Enter the product key:
748RB-X4T6B-MRM7V-RTVFF-CHC8H
< Back
Next >
Cancel
Microsoft SQL Server 2012
12 items
1 item selected 196 KB
Type here to search
09:38
22-07-2023

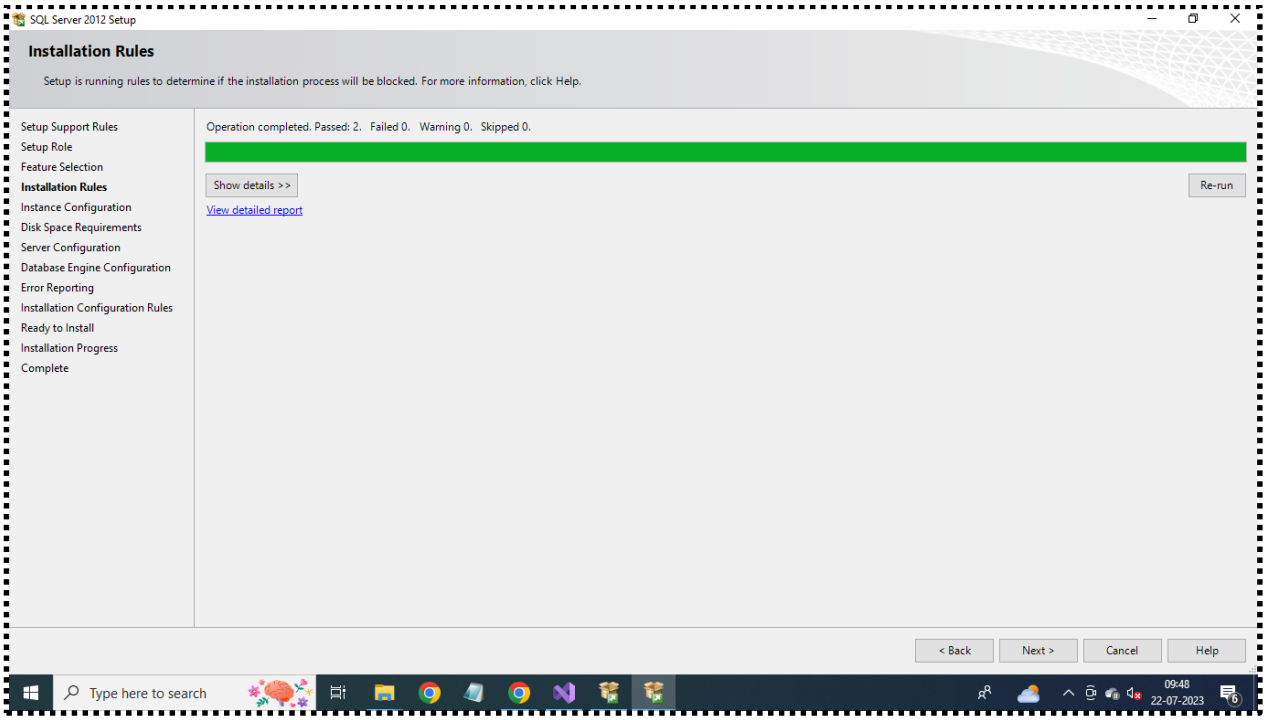

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server stand-alone installation or add features to an existing installation
SQL Server 2012 Setup
License Terms
To install SQL Server 2012, you must accept the Microsoft Software License Terms.
Product Key
License Terms
Product Updates
Install Setup Files
MICROSOFT SOFTWARE LICENSE TERMS
MICROSOFT SQL SERVER 2012 ENTERPRISE SERVER/CAL EDITION
These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft
updates,
supplements,
Internet-based services, and
support services
Copy
Print
I accept the license terms.
Send feature usage data to Microsoft. Feature usage data includes information about your hardware configuration and how you use SQL Server and its components.
See the Microsoft SQL Server 2012 Privacy Statement for more information.
< Back
Next >
Cancel
Microsoft SQL Server 2012
12 items
1 item selected 196 KB
Type here to search
09:39
22-07-2023

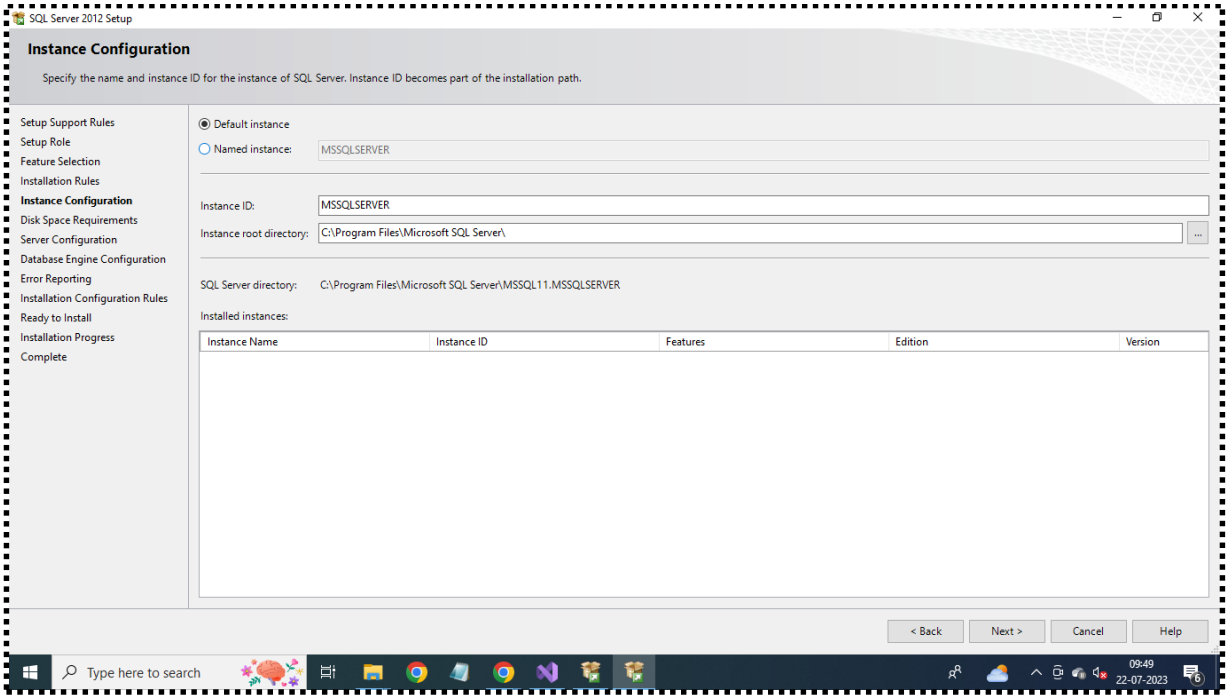

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server stand-alone installation or add features to an existing installation
SQL Server 2012 Setup
Product Updates
Always install the latest updates to enhance your SQL Server security and performance.
Product Key
License Terms
Product Updates
Install Setup Files
Include SQL Server product updates
Name
Size (MB)
More Information
SQL Server 2012 SP1 GDR Setup ...
26
KB 2793634
1 updates (26 MB) found online.
The Setup updates (26 MB) will be installed when you click Next.
Read our privacy statement online
Learn more about SQL Server product updates
< Back
Next >
Cancel
SQL Server 2012
12 items
1 item selected 196 KB
Type here to search
09:39
22-07-2023

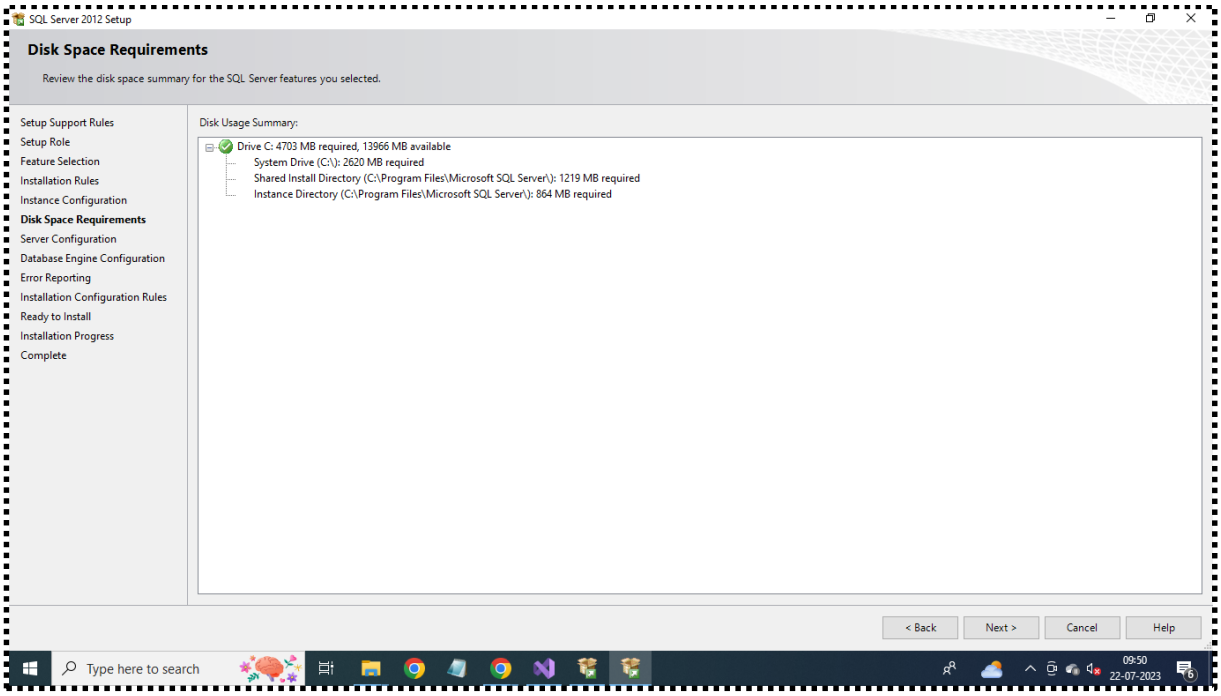

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server stand-alone installation or add features to an existing installation
SQL Server 2012 Setup
Install Setup Files
SQL Server Setup will now be installed. If an update for SQL Server Setup is found and specified to be included, the update will also be installed.
Product Key
License Terms
Product Updates
Install Setup Files
Downloading the Setup files: 0 MB of 26 MB downloaded (0 %)
Task
Status
Scan for product updates
Completed
Download Setup files
In Progress
Extract Setup files
Not started
Install Setup files
Not started
< Back
Install
Cancel
SQL Server 2012
12 items
1 item selected 196 KB
Type here to search
09:39
22-07-2023

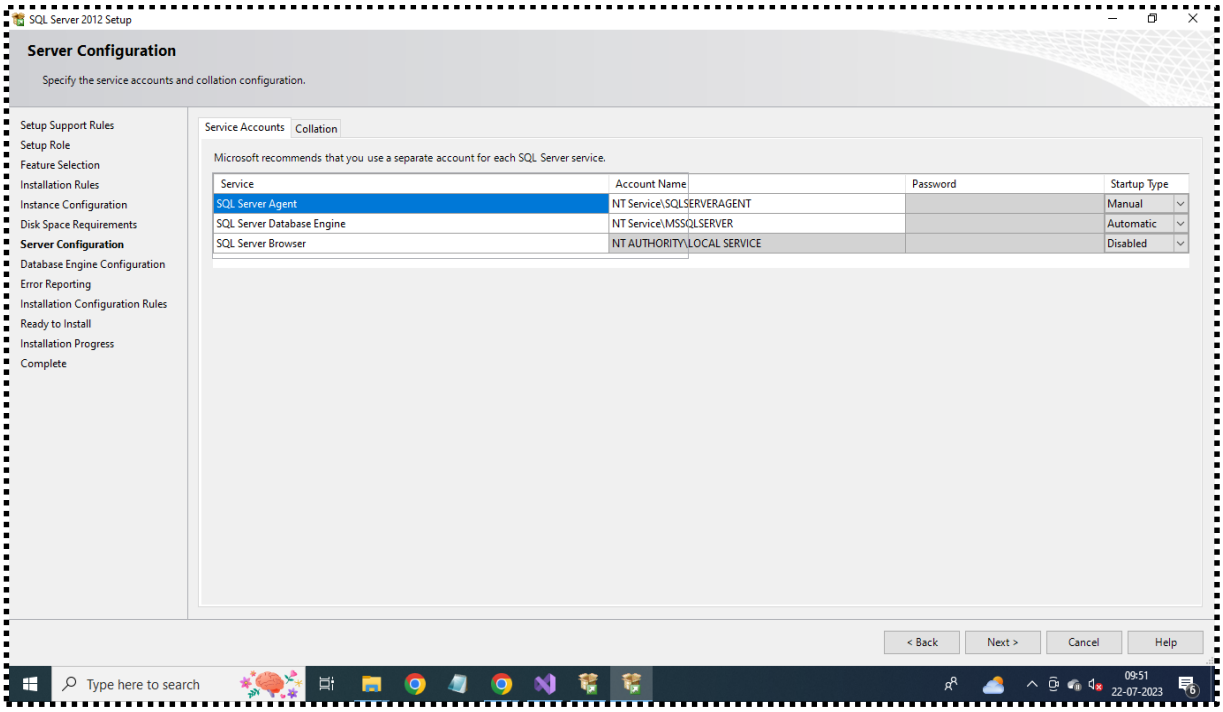

SQL Server Installation Center
Planning
Installation
Maintenance
Tools
Resources
Advanced
Options
New SQL Server stand-alone installation or add features to an existing installation
SQL Server 2012 Setup
Setup Support Rules
Setup Support Rules identify problems that might occur when you install SQL Server Setup support files. Failures must be corrected before Setup can continue.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Disk Space Requirements
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Operation completed. Passed: 6. Failed 0. Warning 1. Skipped 0.
Hide details <<
Re-run
View detailed report
Rule
Status
Fusion Active Template Library (ATL)
Passed
Previous releases of SQL Server 2008 Business Intelligence Devel...
Passed
No SxS install with SQL Server "Denali" CTP0
Passed
Consistency validation for SQL Server registry keys
Passed
Computer domain controller
Passed
Microsoft .NET Application Security
Passed
Windows Firewall
Warning
< Back
Next >
Cancel
Help
Search SQL Server ...
12 items
1 item selected 196 KB
Type here to search
09:44
22-07-2023

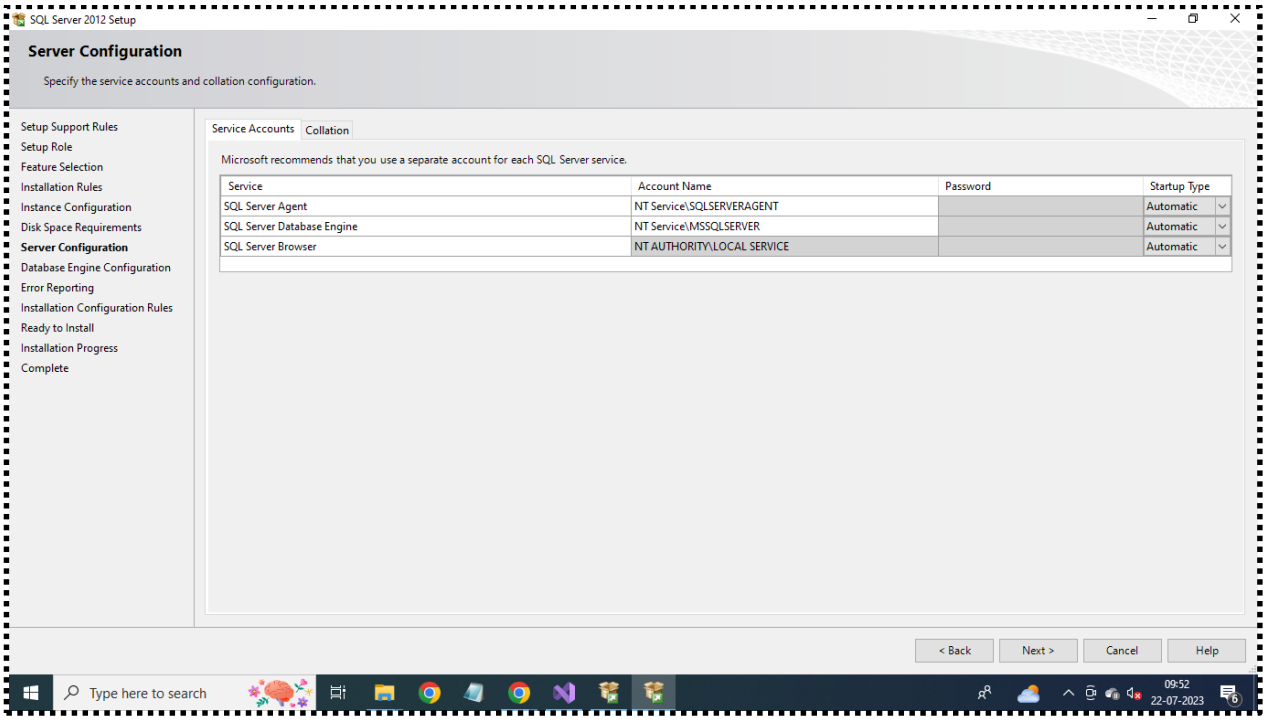

SQL Server 2012 Setup
Feature Selection
Select the Enterprise features to install.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Features:
Database Engine Services
SQL Server Replication
Full-Text and Semantic Extractions for Search
Data Quality Services
Analysis Services
Reporting Services - Native
Shared Features
Reporting Services - SharePoint
Reporting Services Add-in for SharePoint Products
Data Quality Client
SQL Server Data Tools
Client Tools Connectivity
Integration Services
Client Tools Backwards Compatibility
Client Tools SDK
Documentation Components
Management Tools - Basic
Management Tools - Complete
Distributed Replay Controller
Distributed Replay Client
SQL Client Connectivity SDK
Master Data Services
Redistributable Features
Feature description:
Installs the SQL server development environment, including the tool formerly named Business Intelligence Development Studio. Also installs the business intelligence tools and references to the web installers for database development tools.
Prerequisites for selected features:
Already installed:
Microsoft .NET Framework 4.0
Windows PowerShell 2.0
Microsoft .NET Framework 3.5
To be installed from media:
Microsoft Visual Studio 2010 Shell
Microsoft Visual Studio Tools for Applications
Select All
Unselect All
Shared feature directory:
C:\Program Files\Microsoft SQL Server\
Shared feature directory (x86):
C:\Program Files (x86)\Microsoft SQL Server\
< Back
Next >
Cancel
Help
Type here to search
09:48
22-07-2023

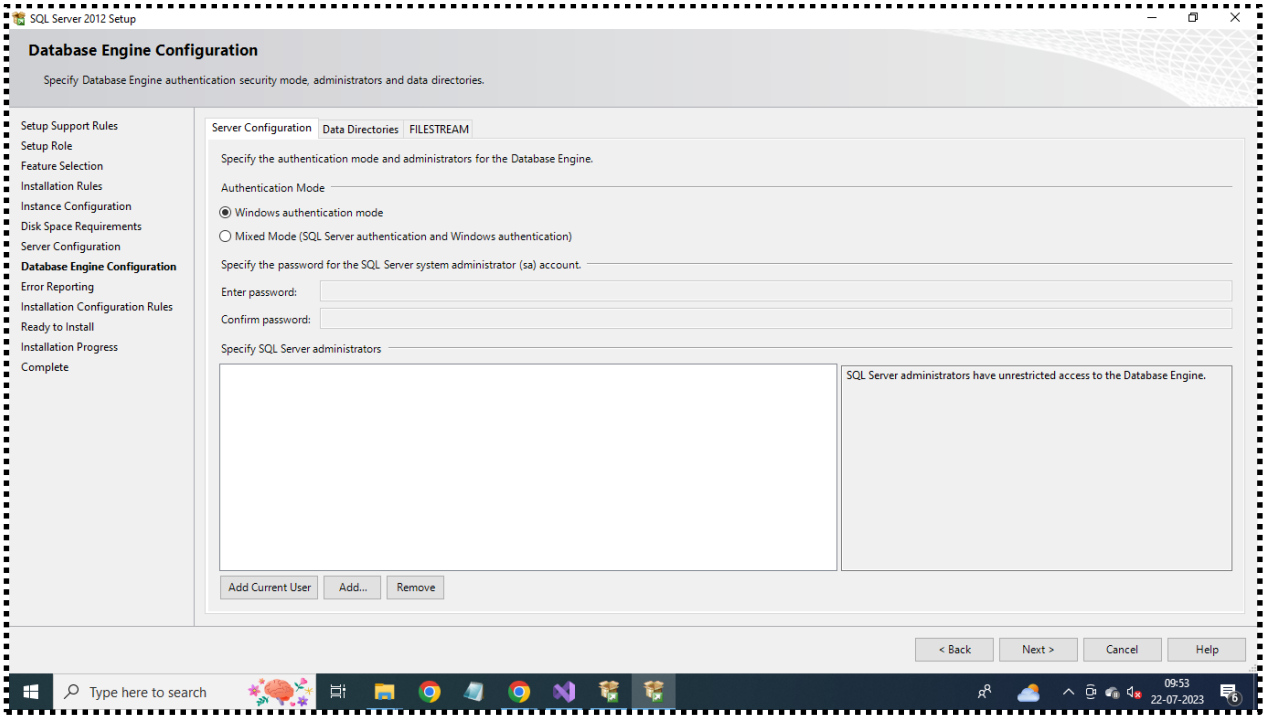

SQL Server 2012 Setup
Installation Rules
Setup is running rules to determine if the installation process will be blocked. For more information, click Help.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Operation completed. Passed: 2. Failed 0. Warning 0. Skipped 0.
Show details >>
Re-run
View detailed report
< Back
Next >
Cancel
Help
Type here to search
09:48
22-07-2023

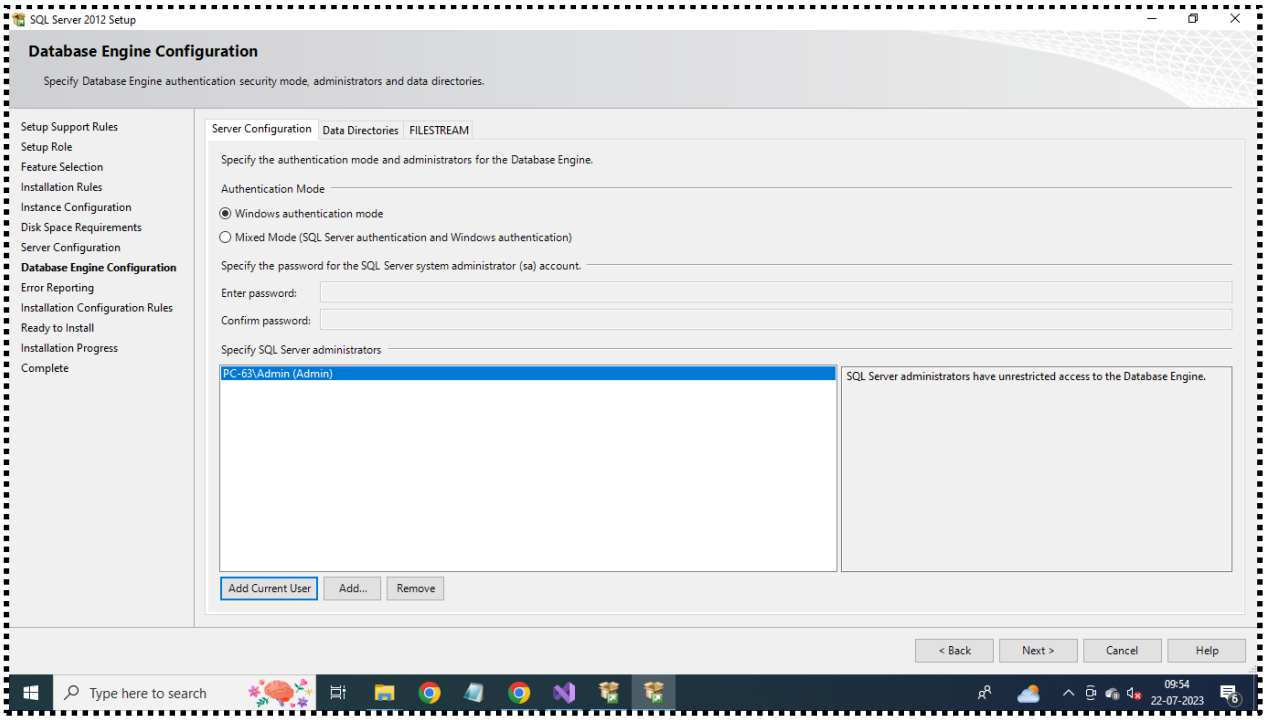

SQL Server 2012 Setup
Instance Configuration
Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Default instance
Named instance:
MSSQLSERVER
Instance ID:
MSSQLSERVER
Instance root directory:
C:\Program Files\Microsoft SQL Server\
SQL Server directory:
C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER
Installed instances:
Instance Name
Instance ID
Features
Edition
Version
< Back
Next >
Cancel
Help
Type here to search
09:49
22-07-2023

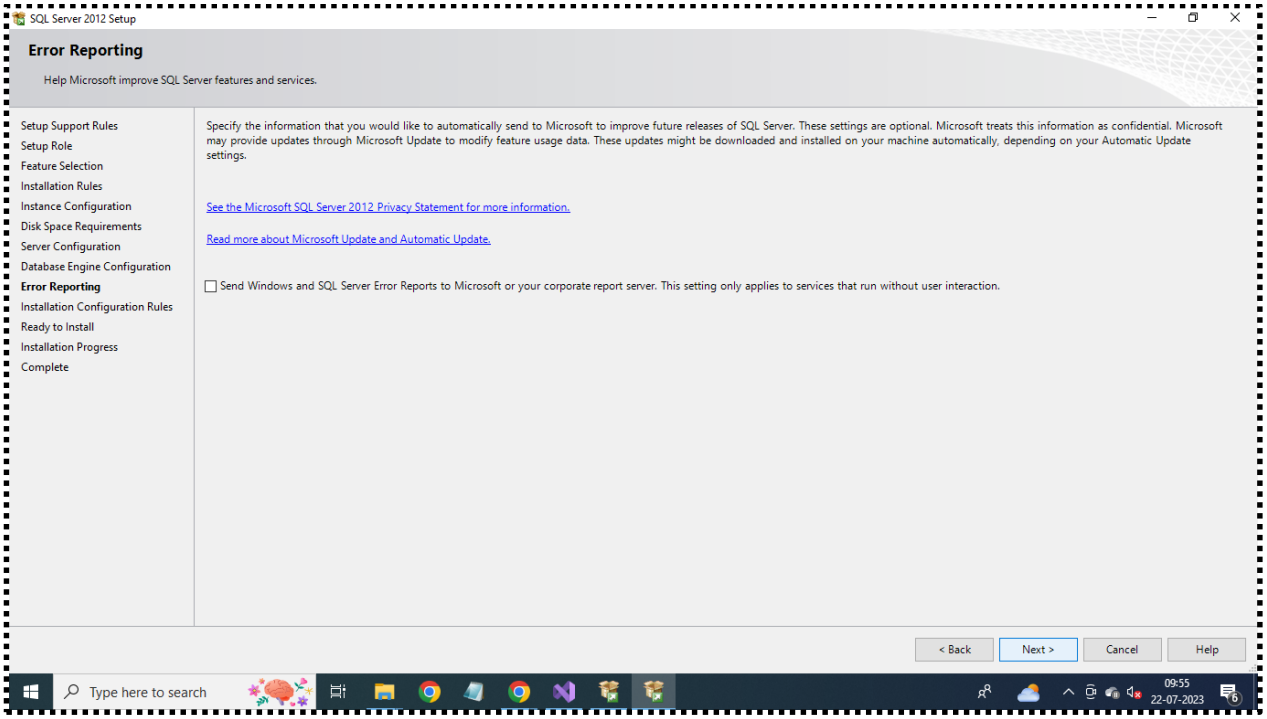

SQL Server 2012 Setup
Disk Space Requirements
Review the disk space summary for the SQL Server features you selected.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Disk Usage Summary:
Drive C: 4703 MB required, 13966 MB available
System Drive (C:\): 2620 MB required
Shared Install Directory (C:\Program Files\Microsoft SQL Server\): 1219 MB required
Instance Directory (C:\Program Files\Microsoft SQL Server\): 864 MB required
< Back
Next >
Cancel
Help
Type here to search
09:50
22-07-2023

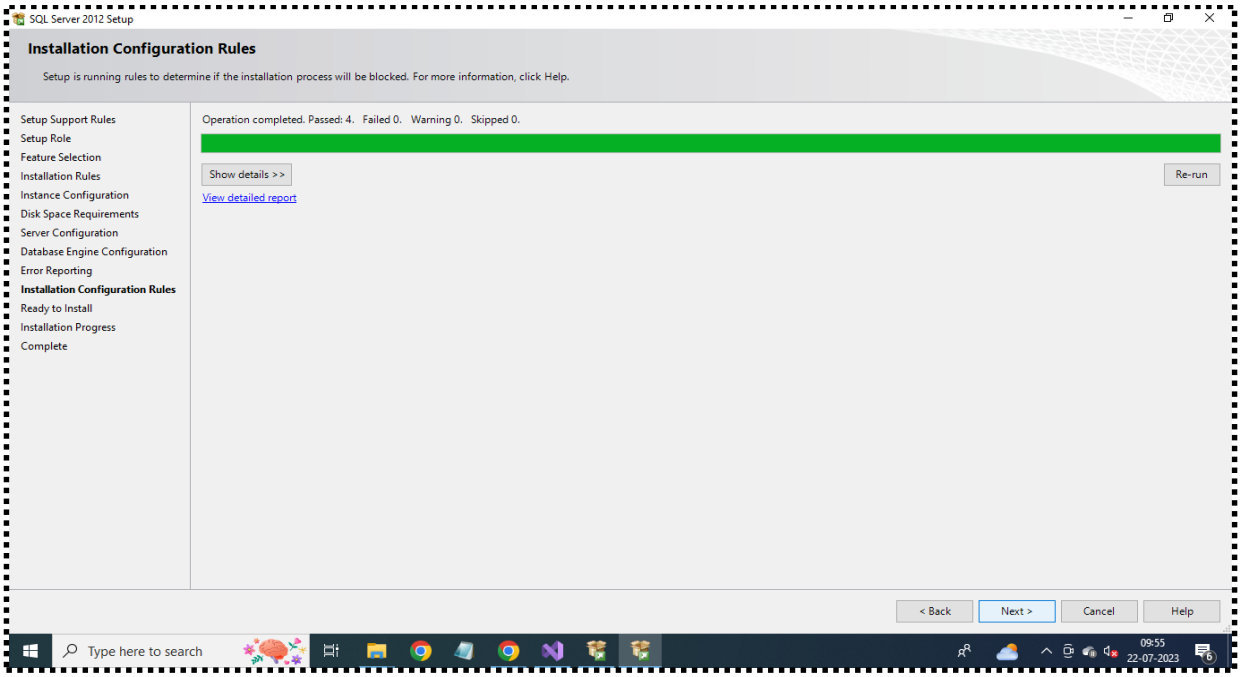

SQL Server 2012 Setup
Server Configuration
Specify the service accounts and collation configuration.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Service Accounts
Collation
Microsoft recommends that you use a separate account for each SQL Server service.
Service
Account Name
Password
Startup Type
SQL Server Agent
NT Service\SQLSERVERAGENT
Manual
SQL Server Database Engine
NT Service\MSSQLSERVER
Automatic
SQL Server Browser
NT AUTHORITY\LOCAL SERVICE
Disabled
< Back
Next >
Cancel
Help
Type here to search
09:51
22-07-2023

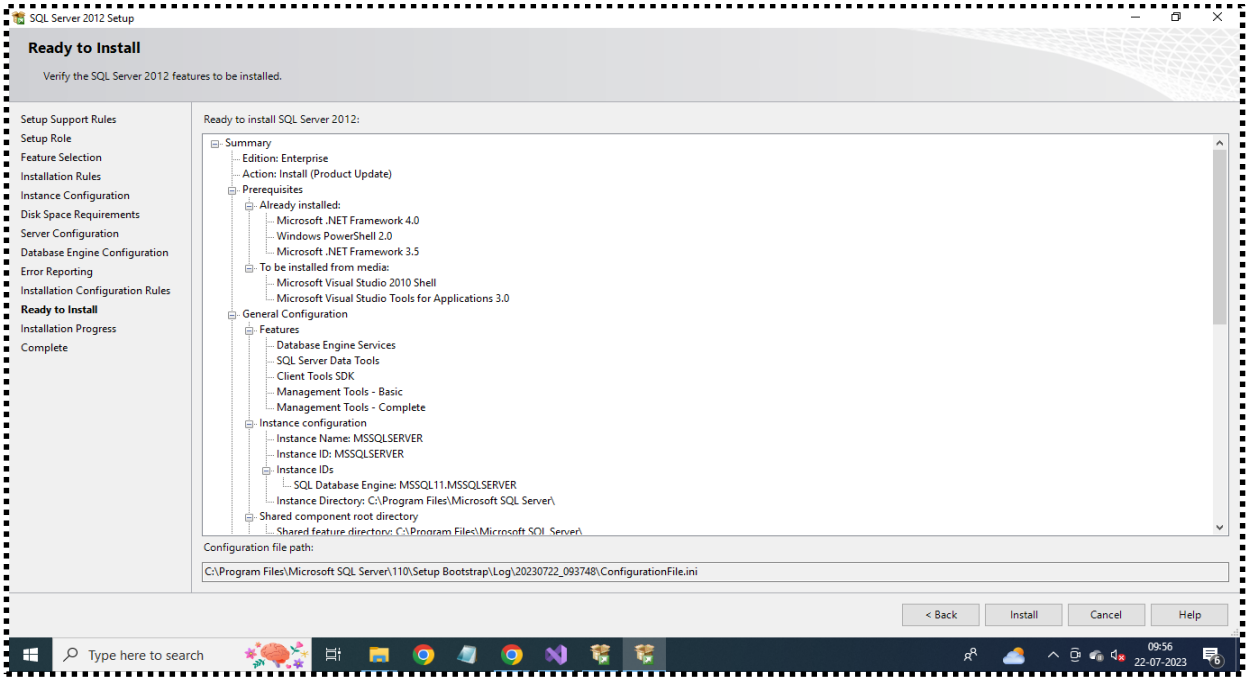

SQL Server 2012 Setup
Server Configuration
Specify the service accounts and collation configuration.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Service Accounts
Collation
Microsoft recommends that you use a separate account for each SQL Server service.
Service
Account Name
Password
Startup Type
SQL Server Agent
NT Service\SQLSERVERAGENT
Automatic
SQL Server Database Engine
NT Service\MSSQLSERVER
Automatic
SQL Server Browser
NT AUTHORITY\LOCAL SERVICE
Automatic
< Back
Next >
Cancel
Help
Type here to search
09:52
22-07-2023

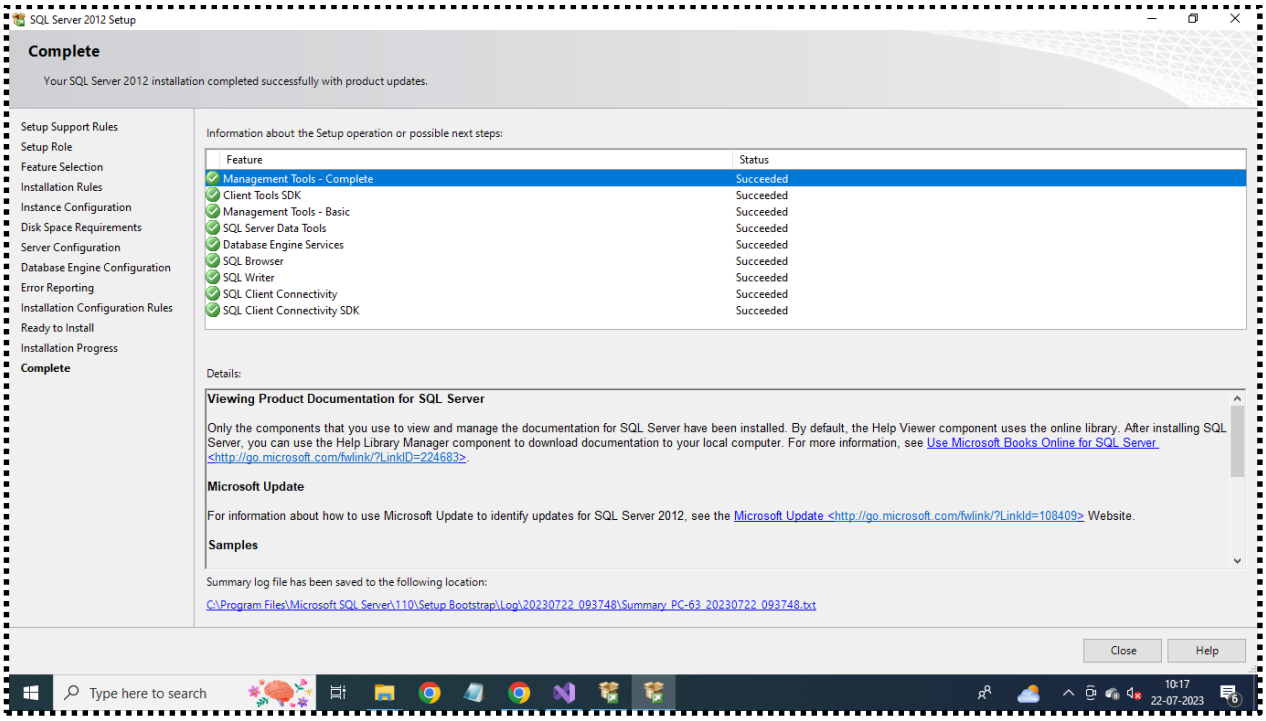

SQL Server 2012 Setup
Database Engine Configuration
Specify Database Engine authentication security mode, administrators and data directories.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Server Configuration
Data Directories
FILESTREAM
Specify the authentication mode and administrators for the Database Engine.
Authentication Mode
Windows authentication mode
Mixed Mode (SQL Server authentication and Windows authentication)
Specify the password for the SQL Server system administrator (sa) account.
Enter password:
Confirm password:
Specify SQL Server administrators
SQL Server administrators have unrestricted access to the Database Engine.
Add Current User
Add...
Remove
< Back
Next >
Cancel
Help
Type here to search
09:53
22-07-2023


SQL Server 2012 Setup
Database Engine Configuration
Specify Database Engine authentication security mode, administrators and data directories.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Server Configuration
Data Directories
FILESTREAM
Specify the authentication mode and administrators for the Database Engine.
Authentication Mode
Windows authentication mode
Mixed Mode (SQL Server authentication and Windows authentication)
Specify the password for the SQL Server system administrator (sa) account.
Enter password:
Confirm password:
Specify SQL Server administrators
PC-63\Admin (Admin)
SQL Server administrators have unrestricted access to the Database Engine.
Add Current User
Add...
Remove
< Back
Next >
Cancel
Help
Type here to search
09:54
22-07-2023


SQL Server 2012 Setup
Error Reporting
Help Microsoft improve SQL Server features and services.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Specify the information that you would like to automatically send to Microsoft to improve future releases of SQL Server. These settings are optional. Microsoft treats this information as confidential. Microsoft may provide updates through Microsoft Update to modify feature usage data. These updates might be downloaded and installed on your machine automatically, depending on your Automatic Update settings.
See the Microsoft SQL Server 2012 Privacy Statement for more information.
Read more about Microsoft Update and Automatic Update.
Send Windows and SQL Server Error Reports to Microsoft or your corporate report server. This setting only applies to services that run without user interaction.
< Back
Next >
Cancel
Help
Type here to search
09:55
22-07-2023


SQL Server 2012 Setup
Installation Configuration Rules
Setup is running rules to determine if the installation process will be blocked. For more information, click Help.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Operation completed. Passed: 4. Failed 0. Warning 0. Skipped 0.
Show details >>
Re-run
View detailed report
< Back
Next >
Cancel
Help
Type here to search
09:55
22-07-2023


SQL Server 2012 Setup
Ready to Install
Verify the SQL Server 2012 features to be installed.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Ready to install SQL Server 2012:
Summary
Edition: Enterprise
Action: Install (Product Update)
Prerequisites
Already installed:
Microsoft .NET Framework 4.0
Windows PowerShell 2.0
Microsoft .NET Framework 3.5
To be installed from media:
Microsoft Visual Studio 2010 Shell
Microsoft Visual Studio Tools for Applications 3.0
General Configuration
Features
Database Engine Services
SQL Server Data Tools
Client Tools SDK
Management Tools - Basic
Management Tools - Complete
Instance configuration
Instance Name: MSSQLSERVER
Instance ID: MSSQLSERVER
Instance IDs
SQL Database Engine: MSSQL11.MSSQLSERVER
Instance Directory: C:\Program Files\Microsoft SQL Server\
Shared component root directory
Configuration file path:
C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20230722_093748\ConfigurationFile.ini
< Back
Install
Cancel
Help
Type here to search
09:56
22-07-2023


SQL Server 2012 Setup
Complete
Your SQL Server 2012 installation completed successfully with product updates.
Setup Support Rules
Setup Role
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Ready to Install
Installation Progress
Complete
Information about the Setup operation or possible next steps:
Feature
Status
Management Tools - Complete
Succeeded
Client Tools SDK
Succeeded
Management Tools - Basic
Succeeded
SQL Server Data Tools
Succeeded
Database Engine Services
Succeeded
SQL Browser
Succeeded
SQL Writer
Succeeded
SQL Client Connectivity
Succeeded
SQL Client Connectivity SDK
Succeeded
Details:
Viewing Product Documentation for SQL Server
Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server. <http://go.microsoft.com/fwlink/?LinkID=224683>.
Microsoft Update
For information about how to use Microsoft Update to identify updates for SQL Server 2012, see the Microsoft Update <http://go.microsoft.com/fwlink/?LinkId=108409> Website.
Samples
Summary log file has been saved to the following location:
C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20230722_093748\Summary_PC-63_20230722_093748.txt
Close
Help
Type here to search
10:17
22-07-2023

6) A) Create a web application that binds data in a multiline textbox by querying in another textbox.

WebForm2.aspx:

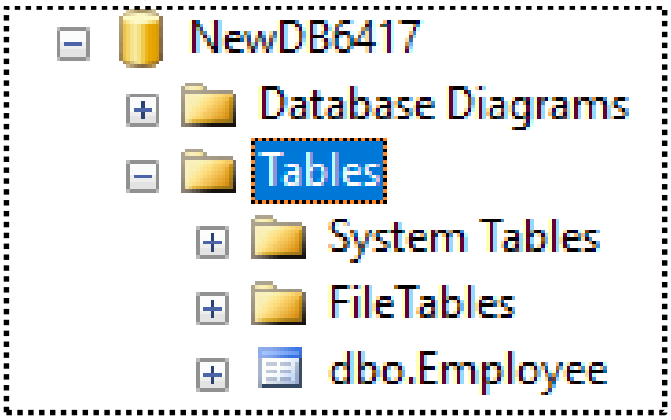
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="Practical6.WebForm2" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
      <div>
         <asp:Label ID="name" runat="server" Text="Name"></asp:Label>
         <asp:TextBox ID="name_TB" runat="server"></asp:TextBox>
         <br />
         <br />
         <asp:Button ID="show_button" runat="server" Text="show" OnClick="show_button_Click"
/>
         <br />
         <br />
         <asp:TextBox ID="output" runat="server" TextMode="MultiLine" Height="45px"
Width="100px"></asp:TextBox>
      </div>
   </form>
</body>
</html>
```

WebForm2.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.SqlClient;
using System.Data;
using System.Data.Sql;

namespace prac6
{
   public partial class WebForm2 : System.Web.UI.Page
```

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

      }

      protected void show_button_Click(object sender, EventArgs e)
      {
          SqlConnection con = new SqlConnection("Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True");
          con.Open();
          String str = "Select * from [Employee] where Name='"+ name_TB.Text+"'";
          SqlCommand cmd = new SqlCommand(str, con);
          SqlDataReader rdr = cmd.ExecuteReader();
          while(rdr.Read())
          {
              output.Text = "ID:" + rdr[0].ToString() + "\n Name" + rdr[1].ToString() + "\n Salary:" +
rdr[2].ToString();
          }
      }

      protected void output_TextChanged(object sender, EventArgs e)
      {

      }

      protected void name_TB_TextChanged(object sender, EventArgs e)
      {

      }
  }
}
```

SQL Management Studio:

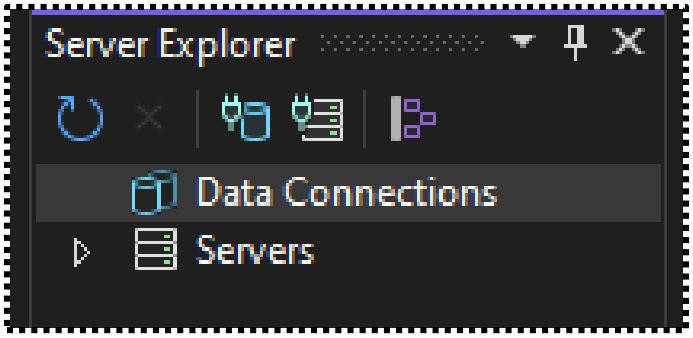


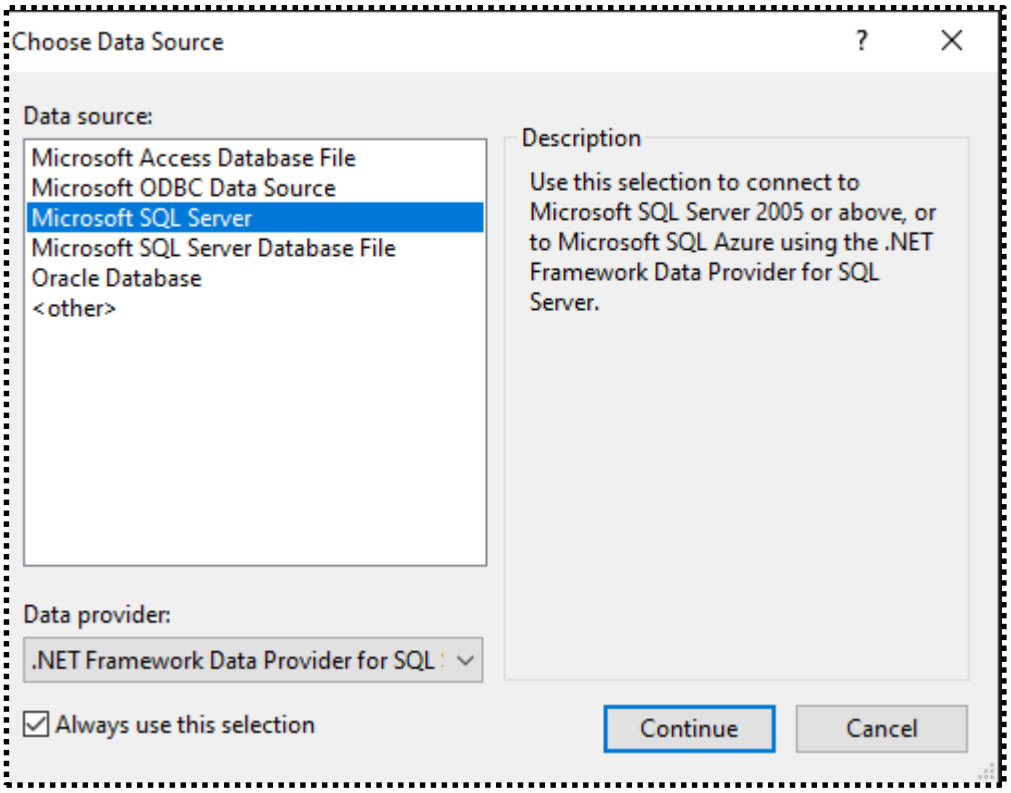
PC-63.NewDB6417 -...17 - dbo.Employee

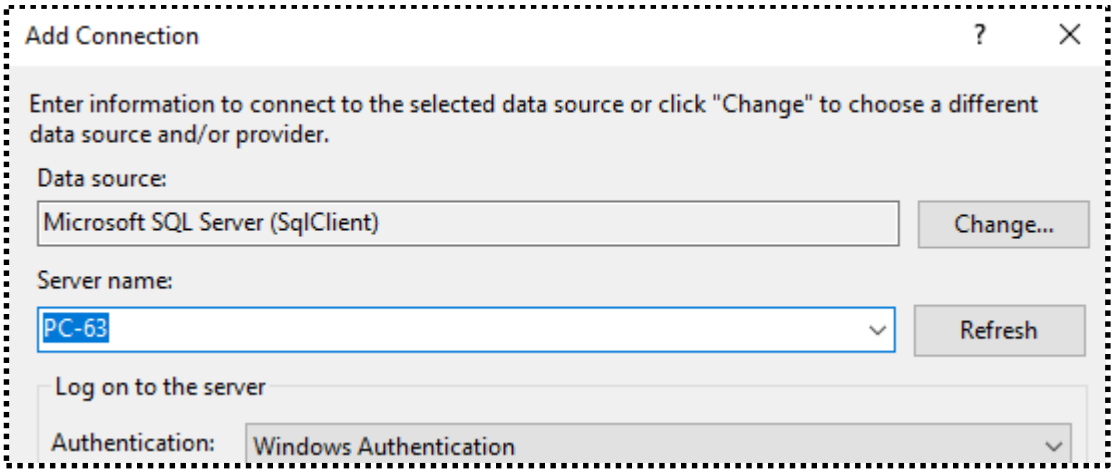
Column Name	Data Type	Allow Nulls
ID	int	☑
Name	varchar(50)	☑
Salary	varchar(50)	☑
		☐

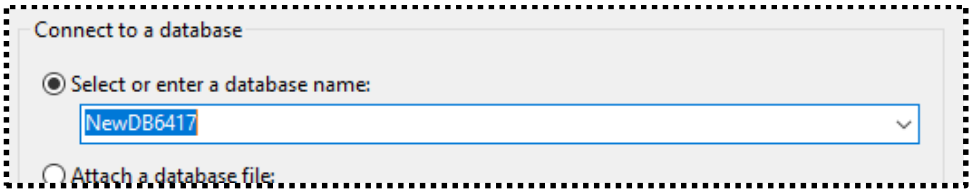
PC-63.NewDB6417 - dbo.Employee | PC-63.NewDB6417 -...1

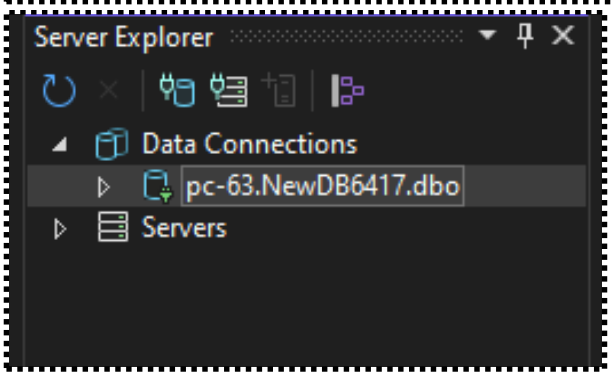
ID	Name	Salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6415	Shivam	35000
NULL	*NULL*	*NULL*


Server Explorer
Data Connections
Servers

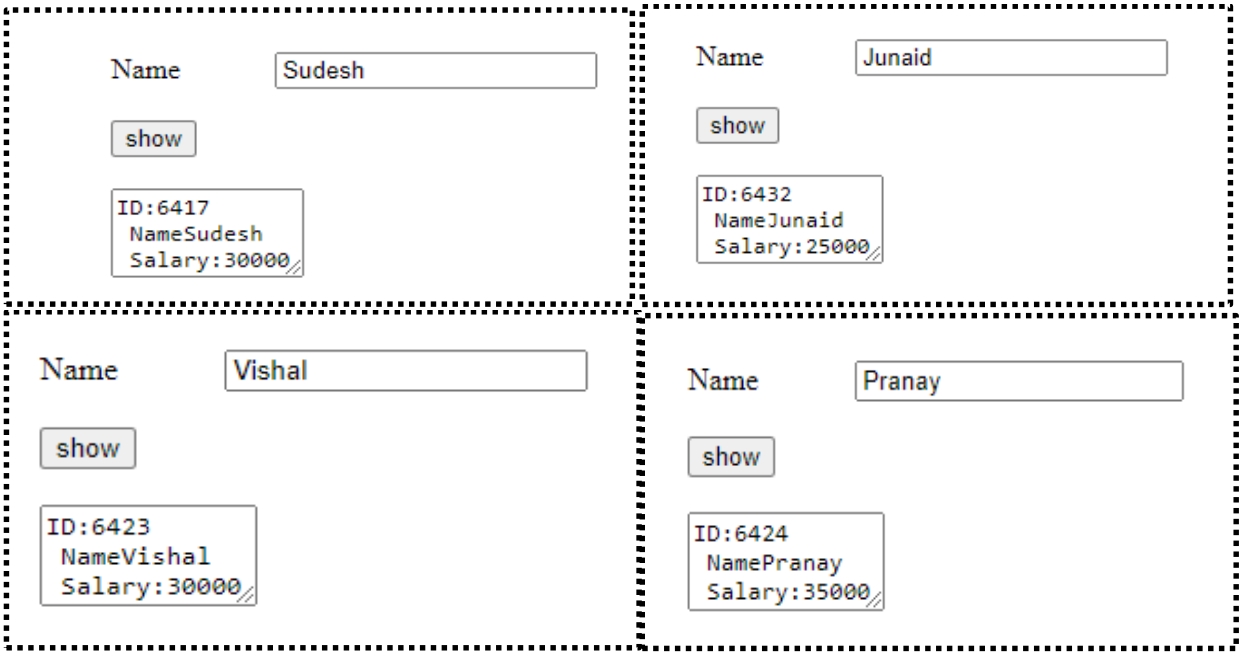

Choose Data Source
Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Database File
Oracle Database
<other>
Description
Use this selection to connect to Microsoft SQL Server 2005 or above, or to Microsoft SQL Azure using the .NET Framework Data Provider for SQL Server.
Data provider:
.NET Framework Data Provider for SQL
Always use this selection
Continue
Cancel


Add Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
PC-63
Refresh
Log on to the server
Authentication:
Windows Authentication


Connect to a database
Select or enter a database name:
NewDB6417
Attach a database file:


Server Explorer
Data Connections
pc-63.NewDB6417.dbo
Servers

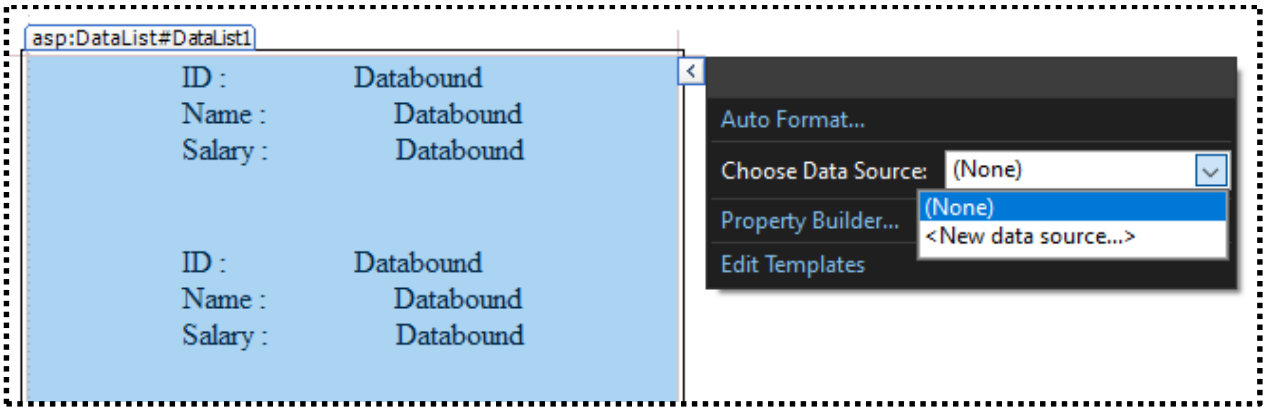
OUTPUT:

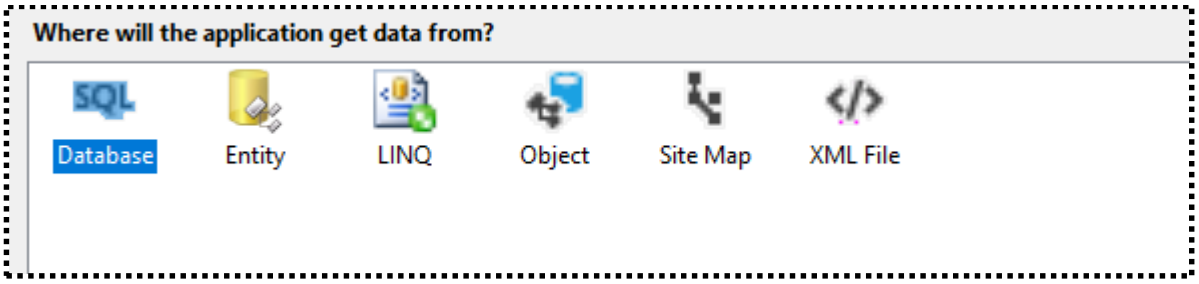

Name
Sudesh
show
ID:6417
NameSudesh
Salary:30000
Name
Junaid
show
ID:6432
NameJunaid
Salary:25000
Name
Vishal
show
ID:6423
NameVishal
Salary:30000
Name
Pranay
show
ID:6424
NamePranay
Salary:35000

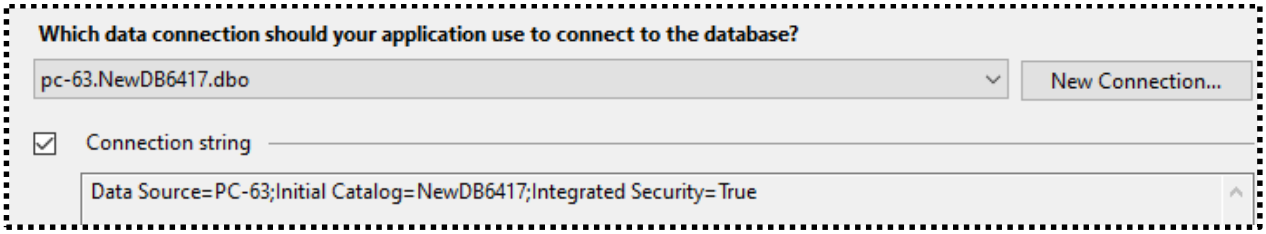
6) B) Demonstrate the use of Data list link control.

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Practical6.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>
               ID :
            <asp:Label ID="ID_Label" runat="server" Text='<%# Eval("ID") %>' />
               <br />

               Name :
            <asp:Label ID="name_Label" runat="server" Text='<%# Eval("Name") %>' />
               <br />

               Salary :
            <asp:Label ID="salary_Label" runat="server" Text='<%# Eval("Salary") %>' />
               <br />
               <br />
            </ItemTemplate>
         </asp:DataList>
      </div>
   </form>
</body>
</html>
```


asp:DataList#DataList1
ID : Databound
Name : Databound
Salary : Databound
ID : Databound
Name : Databound
Salary : Databound
Auto Format...
Choose Data Source: (None)
Property Builder...
Edit Templates
(None)
<New data source...>


Where will the application get data from?
Database
Entity
LINQ
Object
Site Map
XML File

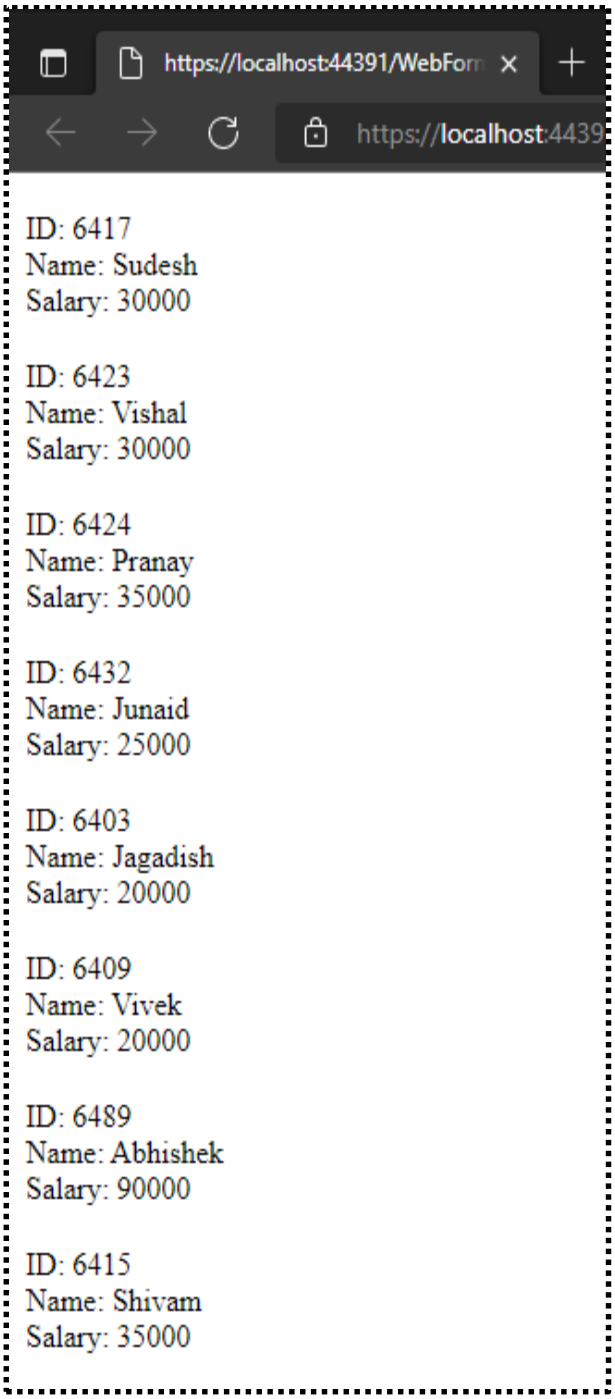

Which data connection should your application use to connect to the database?
pc-63.NewDB6417.dbo
New Connection...
Connection string
Data Source=PC-63;Initial Catalog=NewDB6417;Integrated Security=True


Name:
Employee
Columns:
*
ID
Name
Salary

ID	Name	Salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6415	Shivam	35000

SELECT statement:

```
SELECT * FROM [Employee]
```



PRACTICAL NO 7 - Working with Database

7) A) Create a web application to display Data Binding using dropdown list control.

Source code:

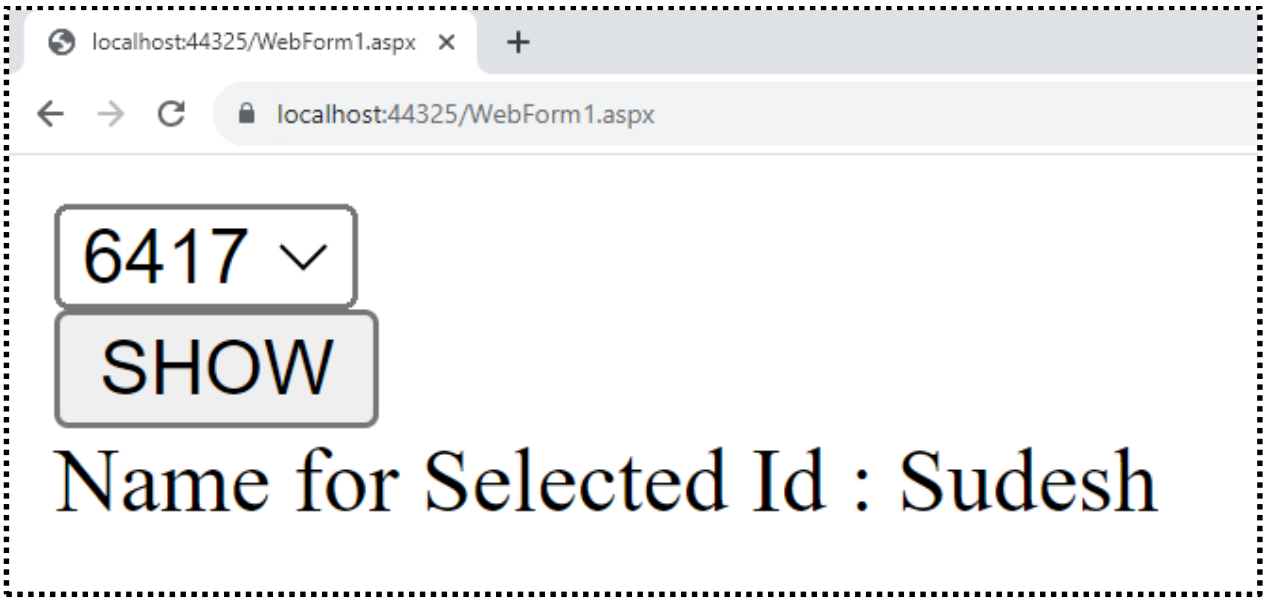
webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="PRAC7.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:DropDownList ID="DropDownList1" runat="server">
         </asp:DropDownList>
         <br />
         <asp:Button ID="Button1" runat="server" Text="SHOW" OnClick="Button1_Click" />

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

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

webform.aspx.cs

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

namespace PRAC7
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {

            SqlConnection con = new SqlConnection(" Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True");
            con.Open();
            string str = "select * from Employee";
            SqlCommand cmd = new SqlCommand(str, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            DropDownList1.DataSource = rdr;
            DropDownList1.DataTextField = "id";
            DropDownList1.DataValueField = "name";


            DropDownList1.DataBind();
         }
      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         Label1.Text = "Name for Selected Id : " + DropDownList1.SelectedValue;
      }
   }
}
```

Output:


localhost:44325/WebForm1.aspx
localhost:44325/WebForm1.aspx
6417
SHOW
Name for Selected Id : Sudesh

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

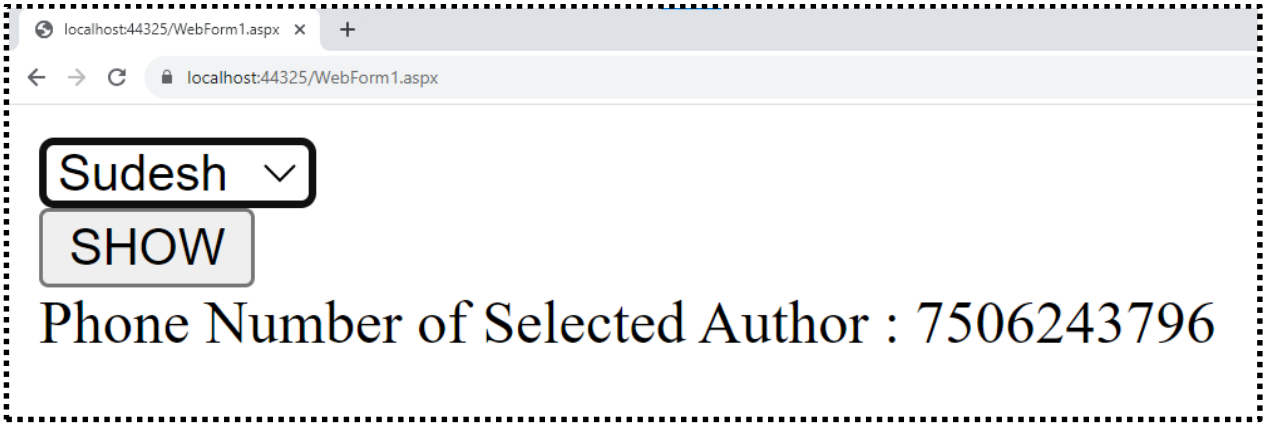
Source code

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="PRAC7.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:DropDownList ID="DropDownList1" runat="server">
         </asp:DropDownList>
         <br />
         <asp:Button ID="Button1" runat="server" Text="SHOW" OnClick="Button1_Click" />

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

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

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

```
namespace PRAC7
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {

            SqlConnection con = new SqlConnection(" Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True");
            con.Open();
            string str = "select * from Authors ";
            SqlCommand cmd = new SqlCommand(str, con);
            SqlDataReader rdr = cmd.ExecuteReader();
            DropDownList1.DataSource = rdr;
            DropDownList1.DataTextField = "Author";
            DropDownList1.DataValueField = "Phone";


            DropDownList1.DataBind();
         }
      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         Label1.Text = "Phone Number of Selected Author : " + DropDownList1.SelectedValue;
      }
   }
}
```

Output



7) C) Create a web application for inserting and deleting records from a database. (Using Execute-Non Query).

webform1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="PRAC7.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="Insert & Delete Operation on Database"
Font-Bold="True" Font-Size="16pt" ForeColor="#333333"></asp:Label>
        <br />
        <asp:Label ID="Label2" runat="server" Text="ID : " Font-Size="14pt"></asp:Label>
        <asp:TextBox ID="id_TB" runat="server"></asp:TextBox>
        <br />
        <asp:Label ID="Label3" runat="server" Text="Name : " Font-Size="14pt"></asp:Label>
        <asp:TextBox ID="name_TB" runat="server"></asp:TextBox>
        <br />
        <asp:Label ID="Label4" runat="server" Text="Salary : " Font-Size="14pt"></asp:Label>
        <asp:TextBox ID="salary_TB" runat="server"></asp:TextBox>
        <br />
        <br />
        <asp:Button ID="insert_Btn" runat="server" Text="Insert" OnClick="insert_Btn_Click"
Font-Bold="True" Font-Size="14pt" ForeColor="#33CC33" />
            
        <asp:Button ID="delete_Btn" runat="server" Text="Delete" OnClick="delete_Btn_Click"
Font-Bold="True" Font-Size="14pt" ForeColor="Red" />

        <br />
        <br />
        <asp:Label ID="message" runat="server" Font-Size="13pt"></asp:Label>
        <br />
         <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataSourceID="SqlDataSource1">
            <Columns>
               <asp:BoundField DataField="ID" HeaderText="id" SortExpression="id" />
               <asp:BoundField DataField="Name" HeaderText="name" SortExpression="name" />
```

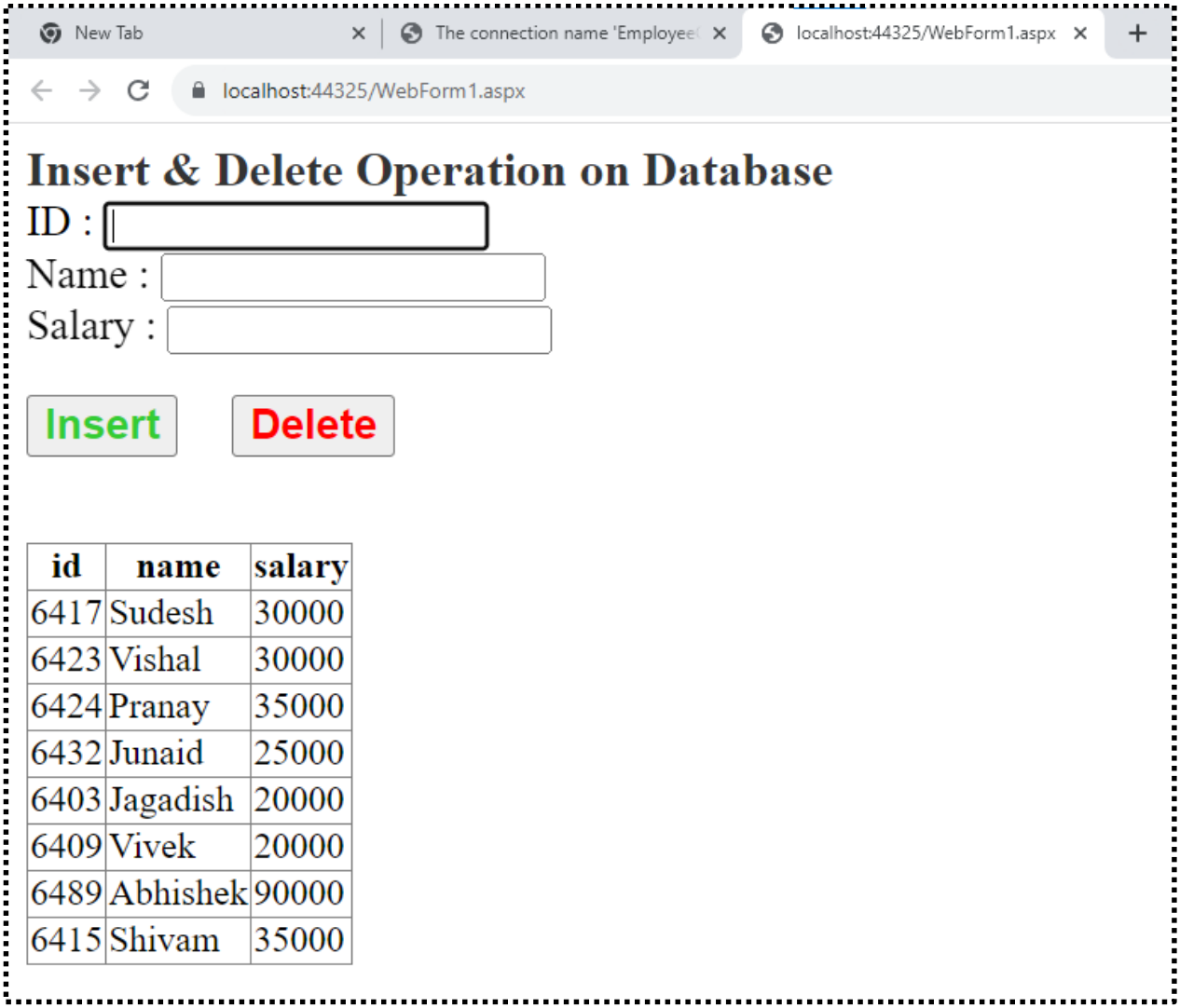
```
            <asp:BoundField DataField="Salary" HeaderText="salary" SortExpression="salary" />
        </Columns>
    </asp:GridView>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="Data
Source=PC-63;Initial Catalog=NewDB6417;Integrated Security=True" SelectCommand="SELECT
* FROM [Employee]" ProviderName="System.Data.SqlClient"></asp:SqlDataSource>

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

webform1.aspx.cs

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

namespace PRAC7
{
  public partial class WebForm1 : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {

        SqlConnection con = new SqlConnection("Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True ");
        con.Open();
        string str = "select * from Employee";
        SqlCommand cmd = new SqlCommand(str, con);
```

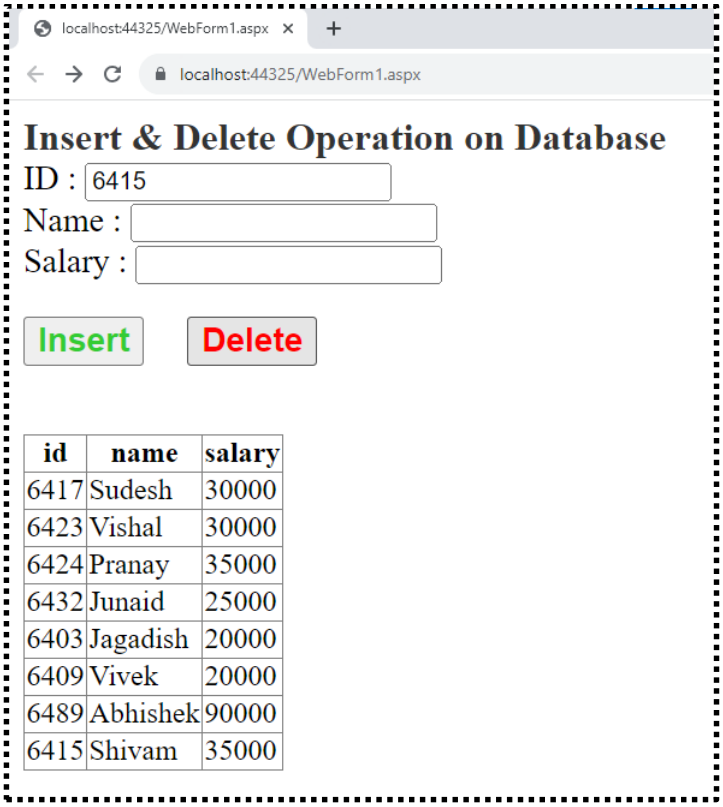
```
            SqlDataReader rdr = cmd.ExecuteReader();
            GridView1.DataBind();
            rdr.Close();
            con.Close();

        }
    }

    protected void insert_Btn_Click(object sender, EventArgs e)
    {
        string id = id_TB.Text;
        string name = name_TB.Text;
        string salary = salary_TB.Text;
        SqlConnection con = new SqlConnection(" Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True");
        con.Open();

        string str = "INSERT INTO [ ] VALUES(" + id + ", '" + name + "' , " + salary + ")";
        SqlCommand cmd = new SqlCommand(str, con);
        cmd.ExecuteNonQuery();

        message.Text = "Data Inserted Successfully";
        GridView1.DataBind();
        con.Close();



    }

    protected void delete_Btn_Click(object sender, EventArgs e)
    {
        string id = id_TB.Text;
        SqlConnection con = new SqlConnection(" ");
        con.Open();
        string str = "delete from where id = (" + id + ")";
        SqlCommand cmd = new SqlCommand(str, con);
        cmd.ExecuteNonQuery();
        message.Text = "Data Deleted Successfully";
        GridView1.DataBind();
        con.Close();

    }
  }
}
```

Output



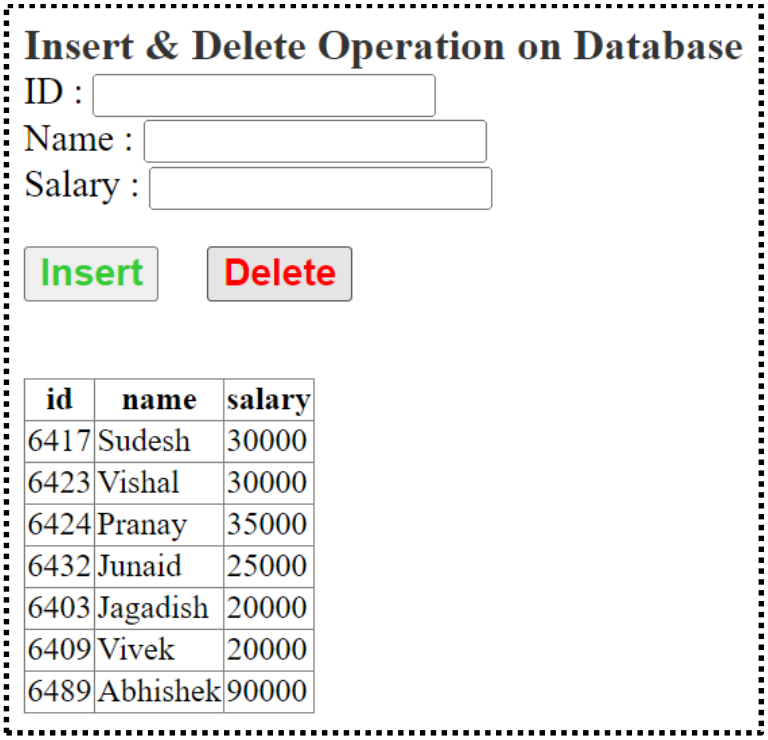
id	name	salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6415	Shivam	35000

Deleting the 6514 record:



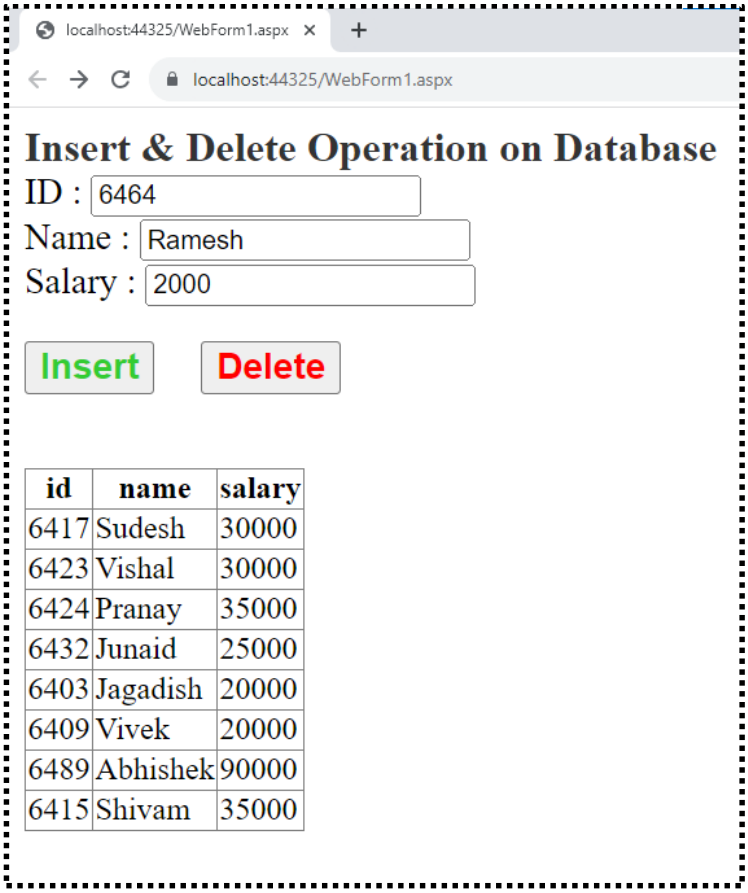
id	name	salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6415	Shivam	35000

6415 record deleted:

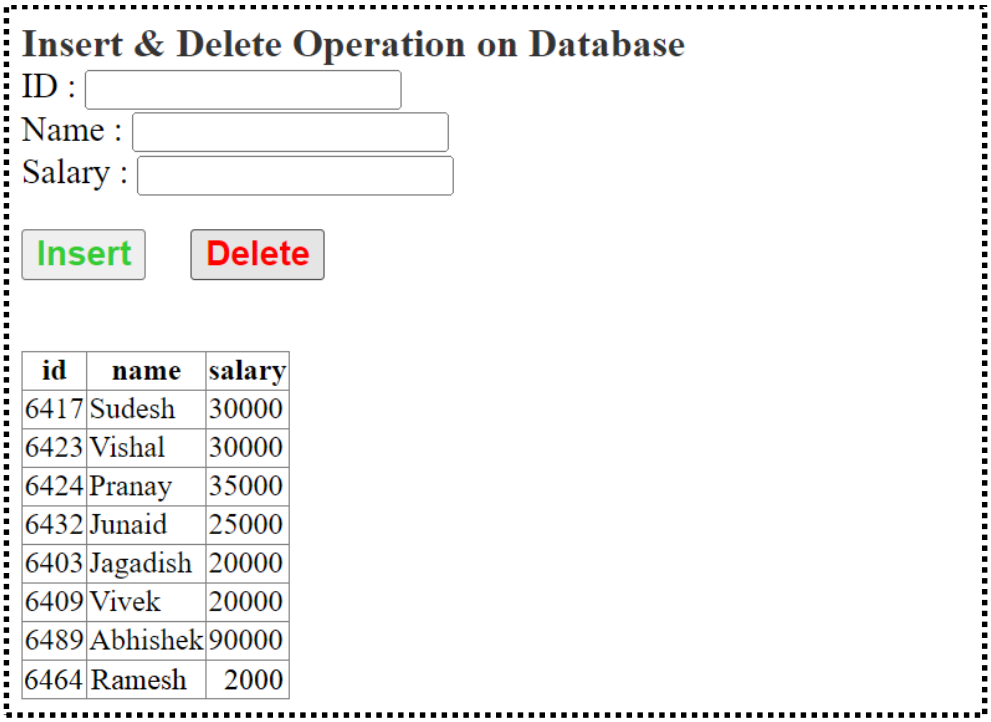


id	name	salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000

Inserting A record in the database:



id	name	salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6415	Shivam	35000



id	name	salary
6417	Sudesh	30000
6423	Vishal	30000
6424	Pranay	35000
6432	Junaid	25000
6403	Jagadish	20000
6409	Vivek	20000
6489	Abhishek	90000
6464	Ramesh	2000

PRACTICAL NO 8 - Working with Data Controls

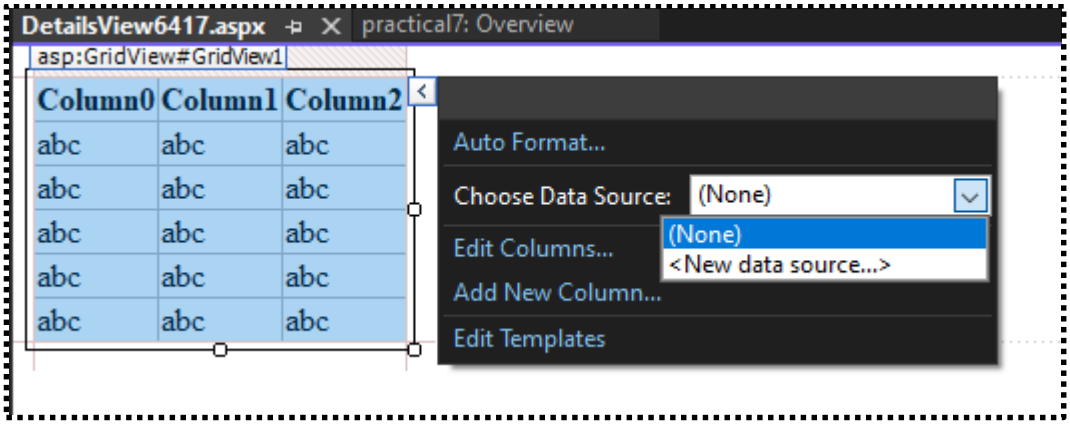
8) A) Create a web application to demonstrate data binding using DetailsView and FormView Control.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DetailsView6417.aspx.cs"
Inherits="practical7.DetailsView6417" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
BackColor="#DEBA84" BorderColor="#DEBA84" BorderStyle="None" BorderWidth="1px"
CellPadding="3" CellSpacing="2" DataKeyNames="ID" DataSourceID="SqlDataSource1">
            <Columns>
               <asp:CommandField ShowSelectButton="True" />
               <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
               <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
            </Columns>
            <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
            <HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
            <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
            <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
            <SelectedRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
            <SortedAscendingCellStyle BackColor="#FFF1D4" />
            <SortedAscendingHeaderStyle BackColor="#B95C30" />
            <SortedDescendingCellStyle BackColor="#F1E5CE" />
            <SortedDescendingHeaderStyle BackColor="#93451F" />
         </asp:GridView>
         <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$
ConnectionStrings:NewDB6417ConnectionString %>" SelectCommand="SELECT [ID], [Name]
FROM [Employee]"></asp:SqlDataSource>
      </div>
      <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
BackColor="#DEBA84" BorderColor="#DEBA84" BorderStyle="None" BorderWidth="1px"
CellPadding="3" CellSpacing="2" DataSourceID="SqlDataSource2" Height="50px" Width="125px">
         <EditRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
```

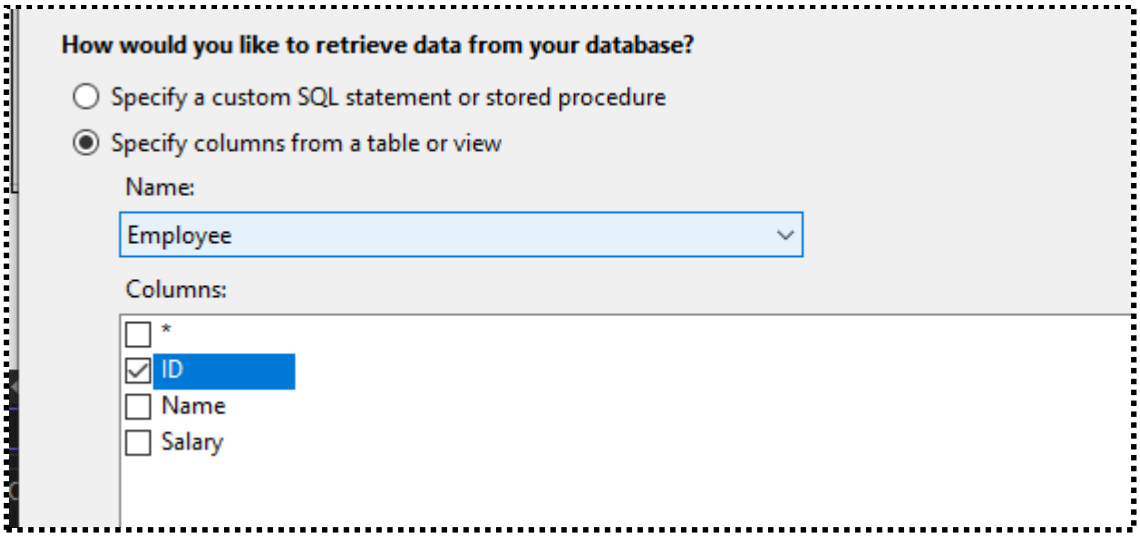
```
        <Fields>
            <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
            <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
            <asp:BoundField DataField="Salary" HeaderText="Salary" SortExpression="Salary" />
        </Fields>
        <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
        <HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
        <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
        <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
    </asp:DetailsView>
    <asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%$
ConnectionStrings:NewDB6417ConnectionString %>" SelectCommand="SELECT * FROM
[Employee] WHERE ([ID] = @ID)">
        <SelectParameters>
            <asp:ControlParameter ControlID="GridView1" PropertyName="SelectedValue"
Name="ID" Type="Int32" />
        </SelectParameters>
    </asp:SqlDataSource>
    <br />
    <asp:FormView ID="FormView1" runat="server" BackColor="#DEBA84"
BorderColor="#DEBA84" BorderStyle="None" BorderWidth="1px" CellPadding="3"
CellSpacing="2" DataSourceID="SqlDataSource1" GridLines="Both">
        <EditItemTemplate>
            ID:
            <asp:TextBox ID="IDTextBox" runat="server" Text='<%# Bind("ID") %>' />
            <br />
            Name:
            <asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />
            <br />
            <asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True"
CommandName="Update" Text="Update" />
             <asp:LinkButton ID="UpdateCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel" Text="Cancel" />
        </EditItemTemplate>
        <EditRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
        <FooterStyle BackColor="#F7DFB5" Font-Names="sans-serif" ForeColor="#8C4510" />
        <FooterTemplate>
            <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
BackColor="#DEBA84" BorderColor="#DEBA84" BorderStyle="None" BorderWidth="1px"
CellPadding="3" CellSpacing="2" DataSourceID="SqlDataSource2" Height="50px" Width="125px">
        <EditRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
        <Fields>
            <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
            <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
            <asp:BoundField DataField="Salary" HeaderText="Salary" SortExpression="Salary" />
```

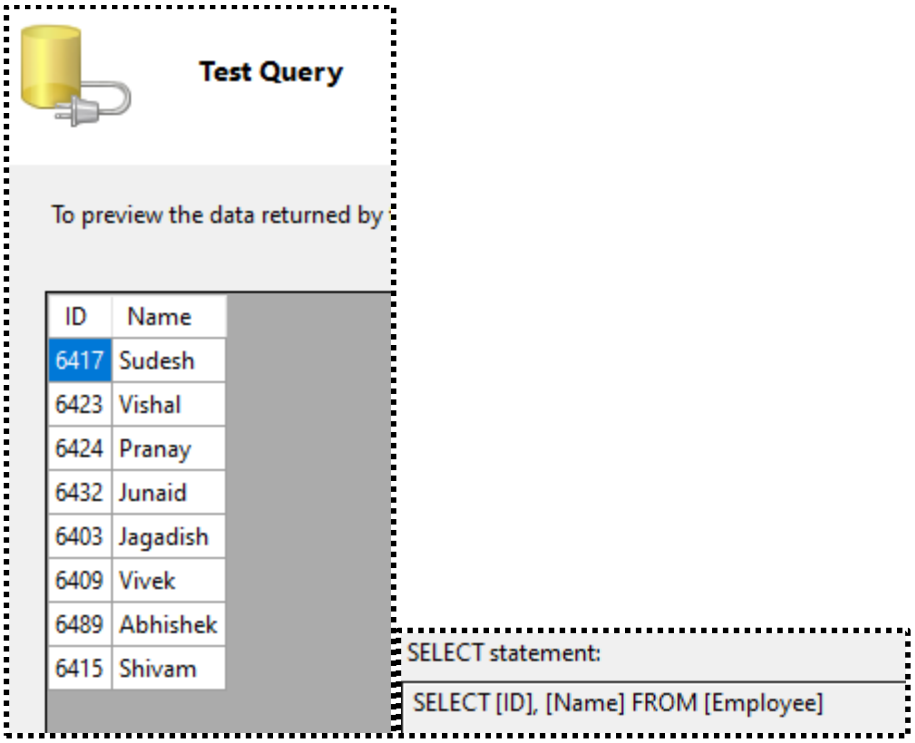
```
        </Fields>
        <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
        <HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
        <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
        <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
      </asp:DetailsView>
            <br />
            THANK YOU....
        </FooterTemplate>
        <HeaderStyle ForeColor="White" BackColor="#A55129" Font-Bold="True" />
        <HeaderTemplate>
          Student Information
        </HeaderTemplate>
        <InsertItemTemplate>
            ID:
            <asp:TextBox ID="IDTextBox" runat="server" Text='<%# Bind("ID") %>' />
            <br />
            Name:
            <asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />
            <br />
            <asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True"
CommandName="Insert" Text="Insert" />
             <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 />
            Name:
            <asp:Label ID="NameLabel" runat="server" Text='<%# Bind("Name") %>' />
            <br />
        </ItemTemplate>
        <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
        <PagerTemplate>1</PagerTemplate>
        <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
      </asp:FormView>
    </form>
</body>
</html>
```

Select a GridView :

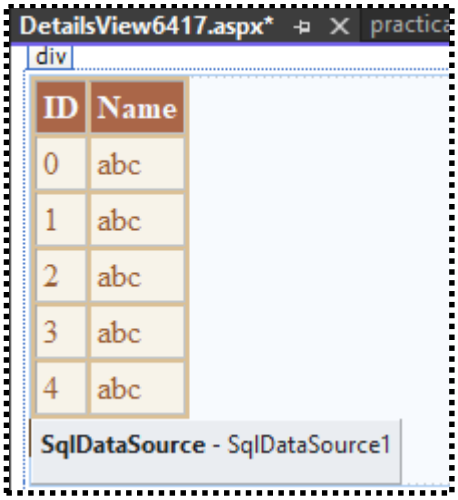


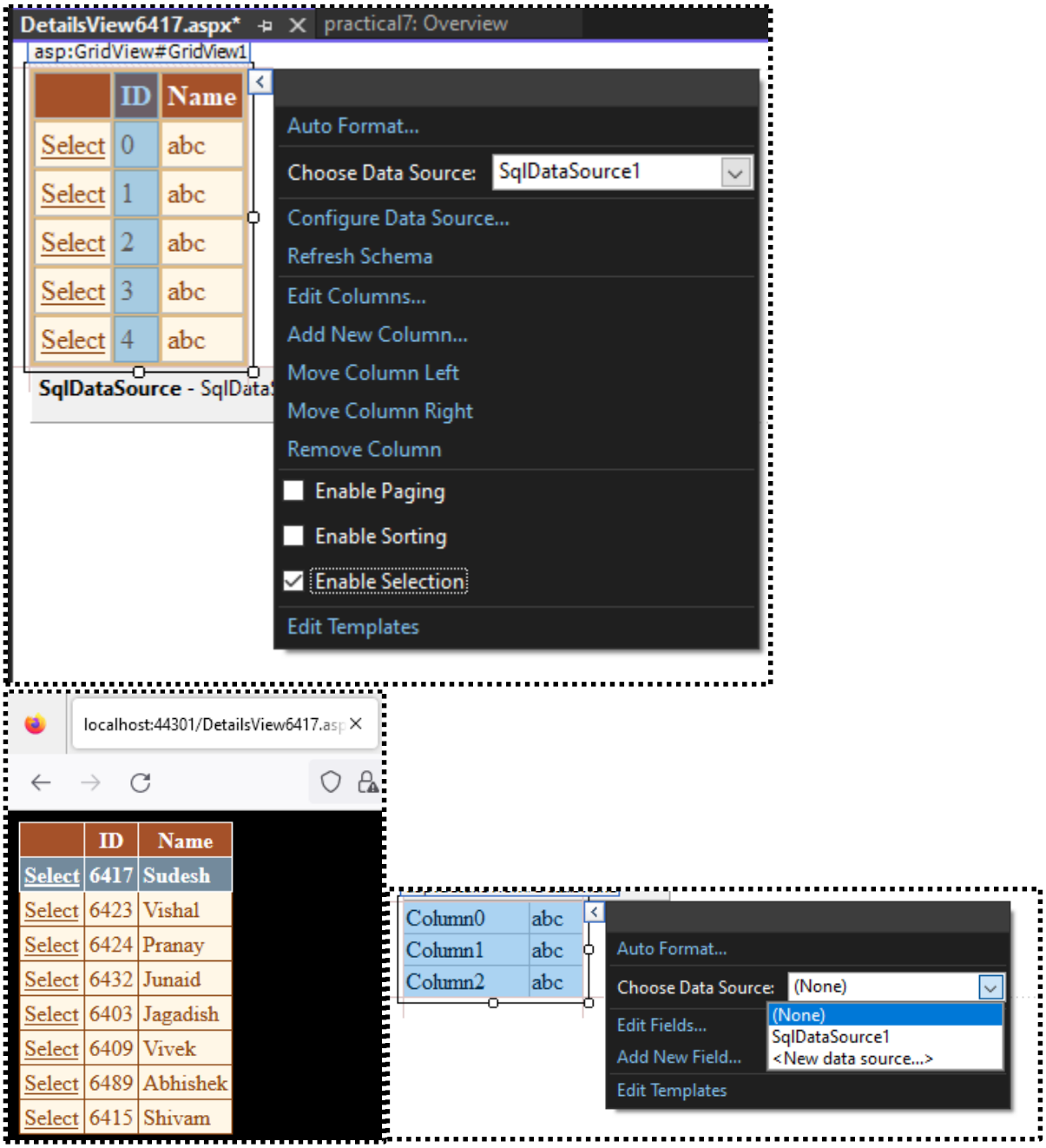
Configure:

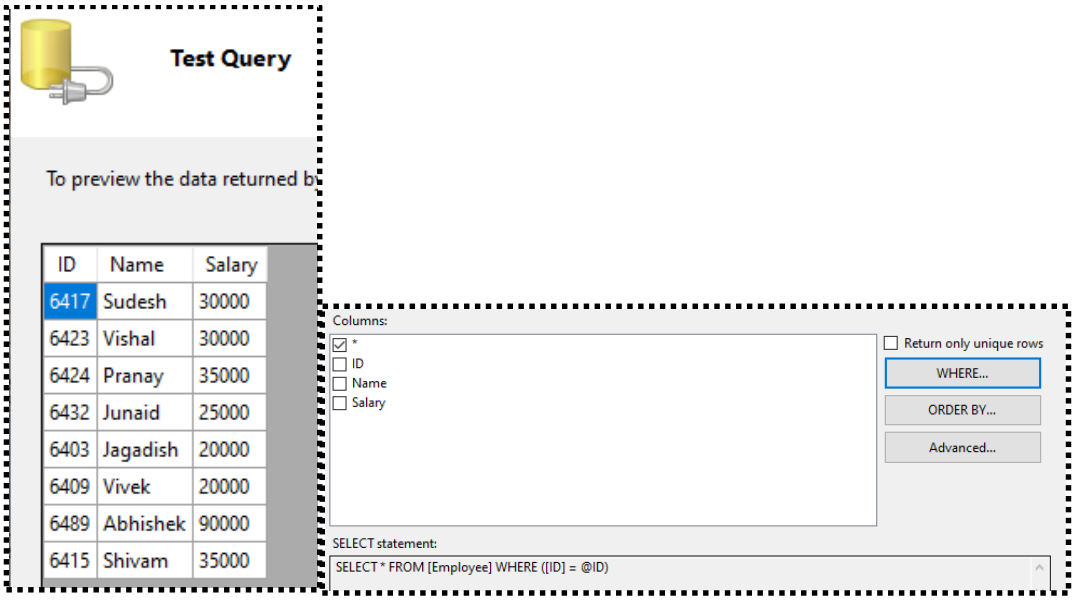




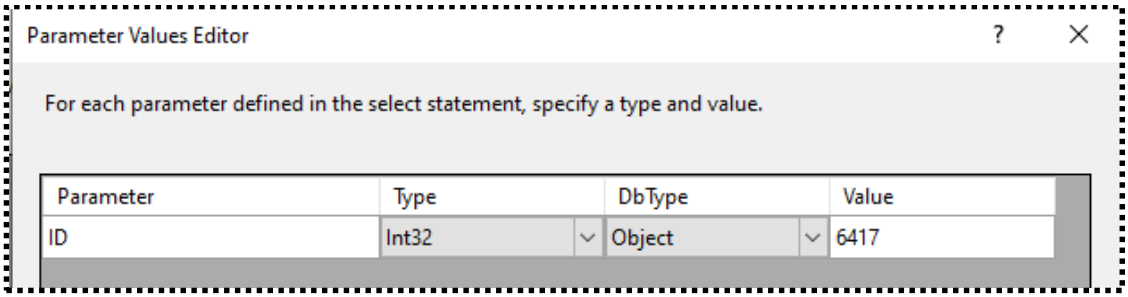
Add a DetailsView:

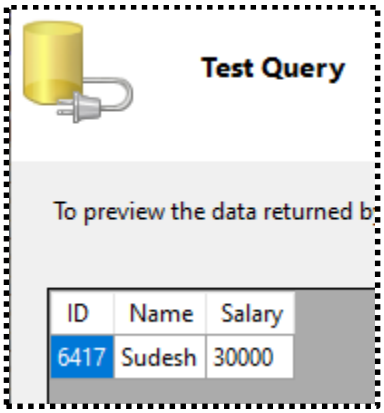



DetailsView6417.aspx*
practical7: Overview
asp:GridView#GridView1
ID
Name
Select 0 abc
Select 1 abc
Select 2 abc
Select 3 abc
Select 4 abc
SqlDataSource - SqlData
Auto Format...
Choose Data Source: SqlDataSource1
Configure Data Source...
Refresh Schema
Edit Columns...
Add New Column...
Move Column Left
Move Column Right
Remove Column
Enable Paging
Enable Sorting
Enable Selection
Edit Templates
localhost:44301/DetailsView6417.asp
ID
Name
Select 6417 Sudesh
Select 6423 Vishal
Select 6424 Pranay
Select 6432 Junaid
Select 6403 Jagadish
Select 6409 Vivek
Select 6489 Abhishek
Select 6415 Shivam
Column0 abc
Column1 abc
Column2 abc
Auto Format...
Choose Data Source: (None)
Edit Fields...
Add New Field...
Edit Templates
(None)
SqlDataSource1
<New data source...>

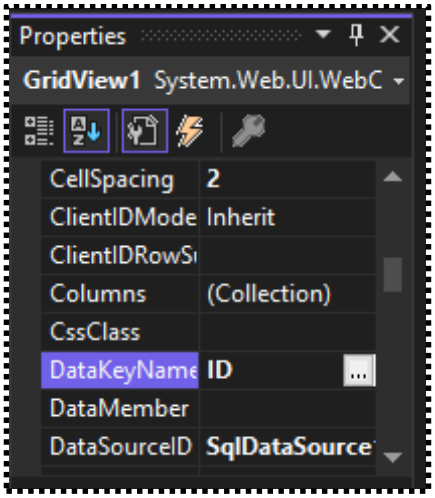


Configure using where:





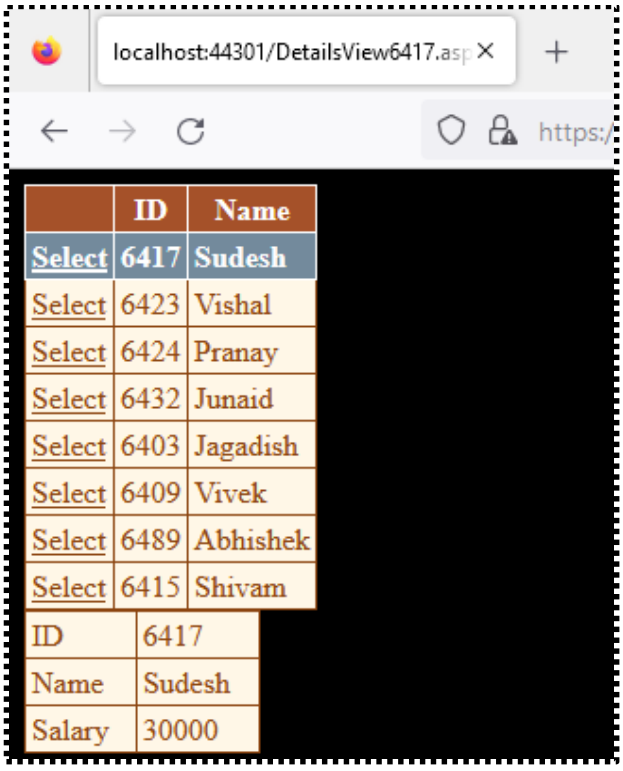
Change Properties of GridView:



Change code of SqlDataSource2:

```
 <asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%$
ConnectionStrings:NewDB6417ConnectionString %>" SelectCommand="SELECT * FROM
[Employee] WHERE ([ID] = @ID)">
        <SelectParameters>
          <asp:ControlParameter ControlID="GridView1" PropertyName="SelectedValue"
Name="ID" Type="Int32" />
        </SelectParameters>
      </asp:SqlDataSource>
```

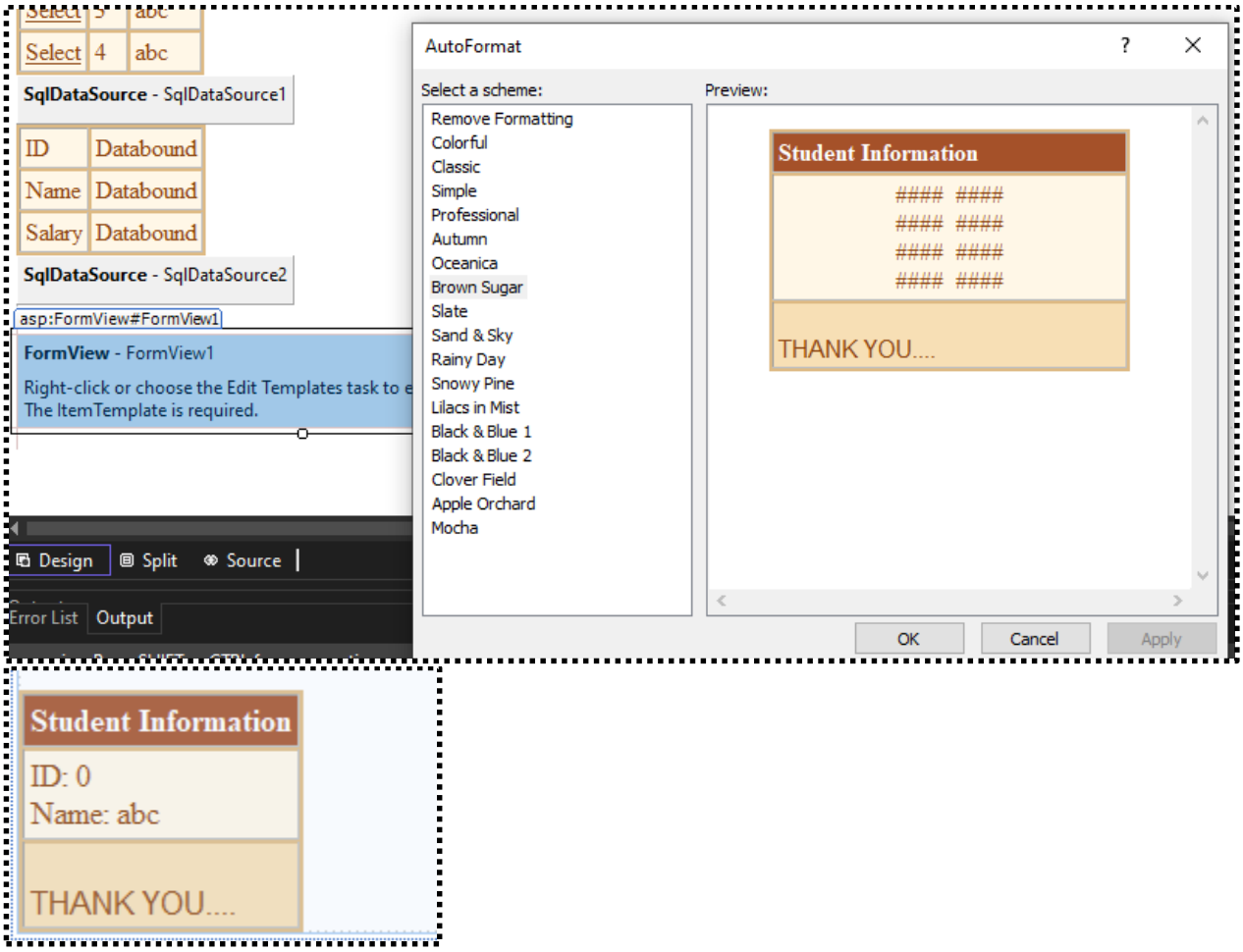
OUTPUT:



	ID	Name
Select	6417	Sudesh
Select	6423	Vishal
Select	6424	Pranay
Select	6432	Junaid
Select	6403	Jagadish
Select	6409	Vivek
Select	6489	Abhishek
Select	6415	Shivam

ID	6417
Name	Sudesh
Salary	30000

Add FormView:



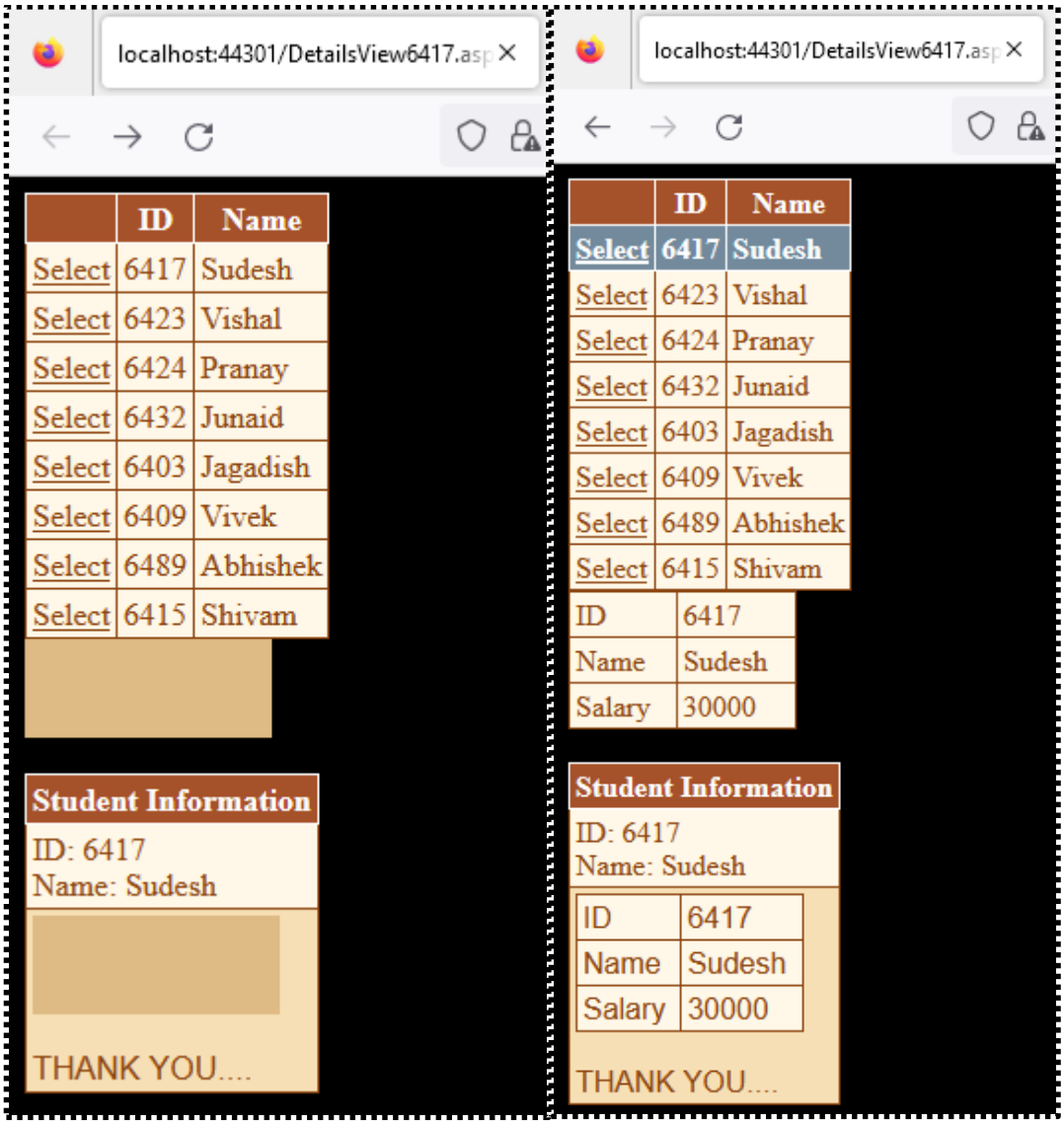
Code change in formview:

```
<asp:FormView ID="FormView1" runat="server" BackColor="#DEBA84" BorderColor="#DEBA84"
BorderStyle="None" BorderWidth="1px" CellPadding="3" CellSpacing="2"
DataSourceID="SqlDataSource1" GridLines="Both">

        <EditItemTemplate>
           ID:
           <asp:TextBox ID="IDTextBox" runat="server" Text='<%# Bind("ID") %>' />
           <br />
           Name:
           <asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />
           <br />
           <asp:LinkButton ID="UpdateButton" runat="server" CausesValidation="True"
CommandName="Update" Text="Update" />
            <asp:LinkButton ID="UpdateCancelButton" runat="server"
CausesValidation="False" CommandName="Cancel" Text="Cancel" />
        </EditItemTemplate>
        <EditRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
        <FooterStyle BackColor="#F7DFB5" Font-Names="sans-serif" ForeColor="#8C4510" />
        <FooterTemplate>

           <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
BackColor="#DEBA84" BorderColor="#DEBA84" BorderStyle="None" BorderWidth="1px"
CellPadding="3" CellSpacing="2" DataSourceID="SqlDataSource2" Height="50px" Width="125px">
        <EditRowStyle BackColor="#738A9C" Font-Bold="True" ForeColor="White" />
        <Fields>
           <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
           <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />

           <asp:BoundField DataField="Salary" HeaderText="Salary" SortExpression="Salary" />
        </Fields>
        <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
        <HeaderStyle BackColor="#A55129" Font-Bold="True" ForeColor="White" />
        <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
        <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
     </asp:DetailsView>

           <br />
           THANK YOU....
        </FooterTemplate>
        <HeaderStyle ForeColor="White" BackColor="#A55129" Font-Bold="True" />
        <HeaderTemplate>
          Student Information
        </HeaderTemplate>
```

```
        <InsertItemTemplate>

           ID:
           <asp:TextBox ID="IDTextBox" runat="server" Text='<%# Bind("ID") %>' />
           <br />
           Name:
           <asp:TextBox ID="NameTextBox" runat="server" Text='<%# Bind("Name") %>' />
           <br />
           <asp:LinkButton ID="InsertButton" runat="server" CausesValidation="True"
CommandName="Insert" Text="Insert" />

            <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 />
           Name:
           <asp:Label ID="NameLabel" runat="server" Text='<%# Bind("Name") %>' />
           <br />
        </ItemTemplate>

        <PagerStyle ForeColor="#8C4510" HorizontalAlign="Center" />
        <PagerTemplate>1</PagerTemplate>

        <RowStyle BackColor="#FFF7E7" ForeColor="#8C4510" />
     </asp:FormView>
```

Output:

BEFORE SELECTION/AFTER SELECTION:



8) B) Create a web application to display Using Disconnected Data Access and Data binding using GridView.

DisconnectedDB.aspx:

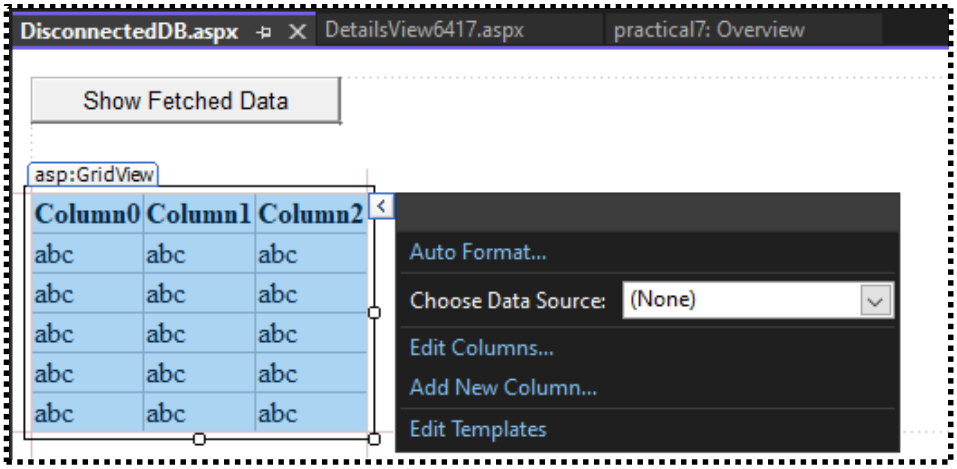
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DisconnectedDB.aspx.cs"
Inherits="practical7.DisconnectedDB" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:Button ID="Button1" runat="server" Text="Show Fetched Data" />
         <br />
         <br />
         <br />
         <asp:GridView runat="server" AutoGenerateColumns="False" DataKeyNames="ID"
DataSourceID="SqlDataSource1"
OnSelectedIndexChanged="Unnamed1_SelectedIndexChanged">
            <Columns>
               <asp:BoundField DataField="ID" HeaderText="ID" SortExpression="ID" />
               <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
               <asp:BoundField DataField="Salary" HeaderText="Salary" SortExpression="Salary"
/>
            </Columns>

         </asp:GridView>
        <!-- <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$
ConnectionStrings:NewDB6417ConnectionString %>" SelectCommand="SELECT * FROM
[Employee]"></asp:SqlDataSource>-->
      </div>
   </form>
</body>
</html>
```

Disconnected.aspx.cs:

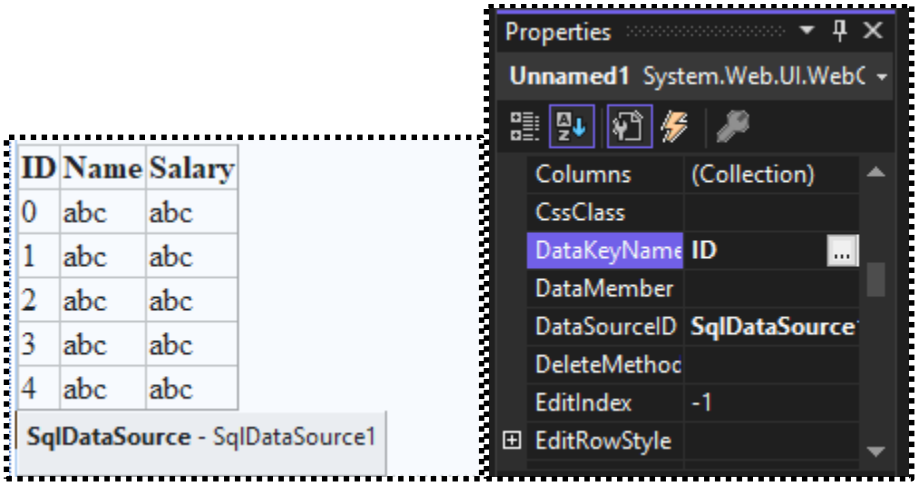
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Data.Sql;
namespace practical7
{
   public partial class DisconnectedDB : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Unnamed1_SelectedIndexChanged(object sender, EventArgs e)
      {
      }
      protected void Button1_Click(object sender, EventArgs e)
      {
         SqlConnection con = new SqlConnection("Data Source=PC-63;Initial
Catalog=NewDB6417;Integrated Security=True");
        GridView1.Visible = true;
         con.Open();
         String str = "Select * from [Employees]";
         SqlDataAdapter ad = new SqlDataAdapter(str,con);
         DataSet ds = new DataSet();
         ad.Fill(ds,"Employees");
         GridView1.DataSource = ds;
         GridView1.DataBind();
         con.Close();
      }
   }
```

Output:

Disconnected DB:

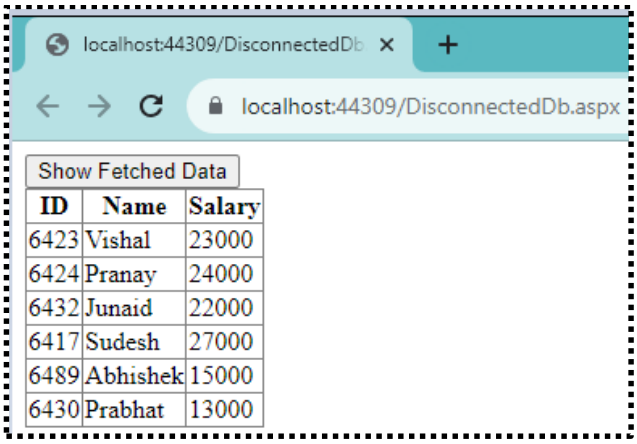


Configure the datasource AND Keyname:



Comment the line: & remove the datasource:

```
<!-- <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:NewDB6417ConnectionString %>"
SelectCommand="SELECT * FROM [Employee]"></asp:SqlDataSource>-->
```



PRACTICAL NO 9 - Working with GridView control

9) A) Create a web application to demonstrate use of GridView button column and GridView events.

Source code

Webform.aspx

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="false"
OnRowCommand="GridView1_RowCommand" CellPadding="4" ForeColor="#333333"
GridLines="None">
         <AlternatingRowStyle BackColor="White" />
         <Columns>
            <asp:TemplateField HeaderText="Name" ItemStyle-Width="150">
               <ItemTemplate>
                  <asp:TextBox ID="TextBox1" runat="server" Text='<%# Eval("Name") %>'>

                  </asp:TextBox>
               </ItemTemplate>

               <ItemStyle Width="150px"></ItemStyle>
            </asp:TemplateField>
            <asp:BoundField DataField="Country" HeaderText="Country" />
            <asp:TemplateField>
               <ItemTemplate>
```

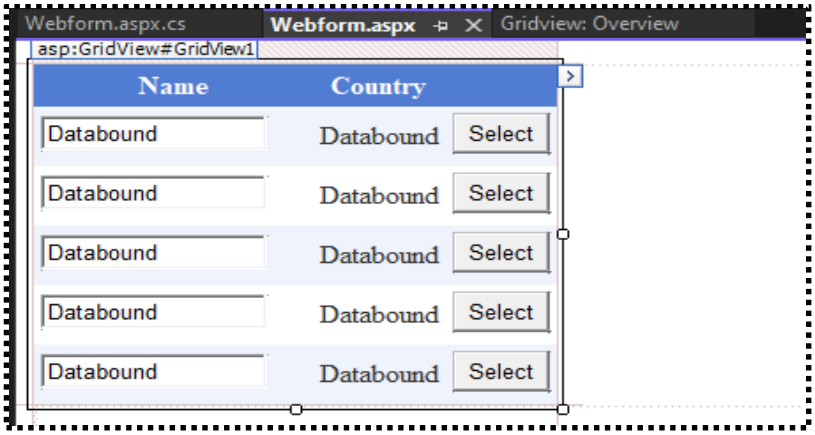
```
                <asp:Button ID="Button1" runat="server" Text="Select" CommandName="Select"
CommandArgument="<%# Container.DataItemIndex %>" />
              </ItemTemplate>
            </asp:TemplateField>
        </Columns>
        <EditRowStyle BackColor="#2461BF" />
        <FooterStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
        <HeaderStyle BackColor="#507CD1" Font-Bold="True" ForeColor="White" />
        <PagerStyle BackColor="#2461BF" ForeColor="White" HorizontalAlign="Center" />
        <RowStyle BackColor="#EFF3FB" />
        <SelectedRowStyle BackColor="#D1DDF1" Font-Bold="True" ForeColor="#333333" />
        <SortedAscendingCellStyle BackColor="#F5F7FB" />
        <SortedAscendingHeaderStyle BackColor="#6D95E1" />
        <SortedDescendingCellStyle BackColor="#E9EBEF" />
        <SortedDescendingHeaderStyle BackColor="#4870BE" />
      </asp:GridView>
      <div>
      </div>
   </form>
</body>
</html>
```

Webform.aspx.cs

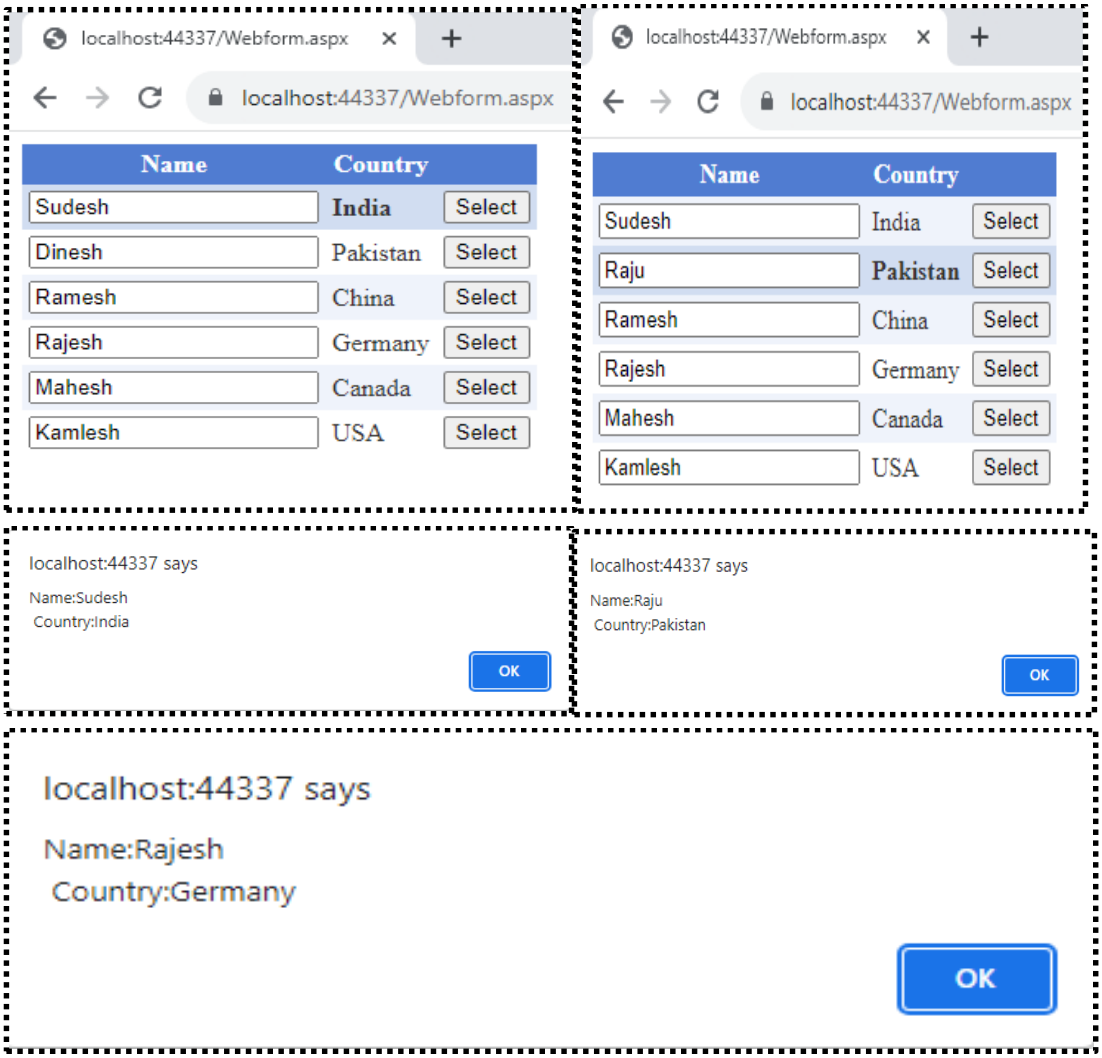
```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

```
    {
        if(!this.IsPostBack)
        {
            DataTable dt =new DataTable();
            dt.Columns.AddRange(new DataColumn[2] { new DataColumn("Name"), new
DataColumn("Country") });
            dt.Rows.Add("Sudesh", "India");
            dt.Rows.Add("Dinesh", "Pakistan");
            dt.Rows.Add("Ramesh", "China");
            dt.Rows.Add("Rajesh", "Germany");
            dt.Rows.Add("Mahesh", "Canada");
            dt.Rows.Add("Kamlesh", "USA");
            GridView1.DataSource = dt;
            GridView1.DataBind();

        }
    }

    protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
    {
        if(e.CommandName=="Select")
        {
            int RowIndex = Convert.ToInt32(e.CommandArgument);
            GridViewRow row = GridView1.Rows[RowIndex];
            string name = (row.FindControl("TextBox1") as TextBox).Text;
            string country = row.Cells[1].Text;
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('Name:" + name +
"\\n Country:"+country + "')", true);
        }
    }
  }
}
```

Design:



Output



9) B) Create a web application to demonstrate GridView paging and Create own table format using GridView.

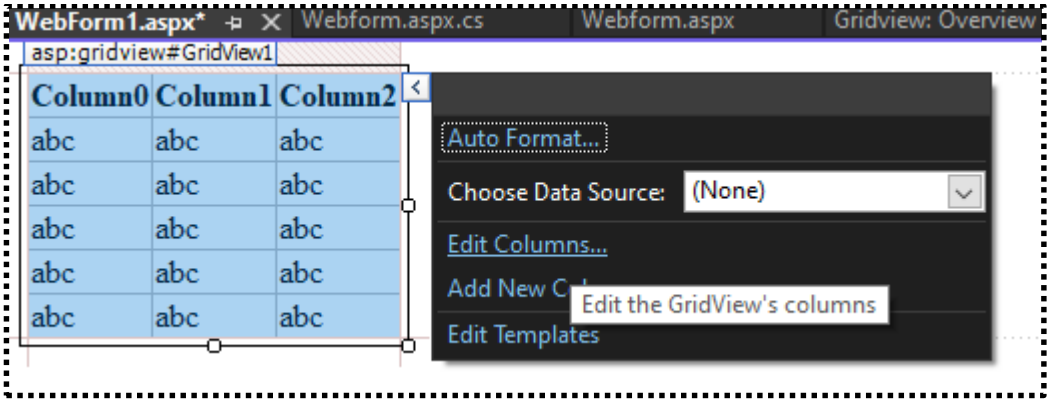
WebForm1.aspx.

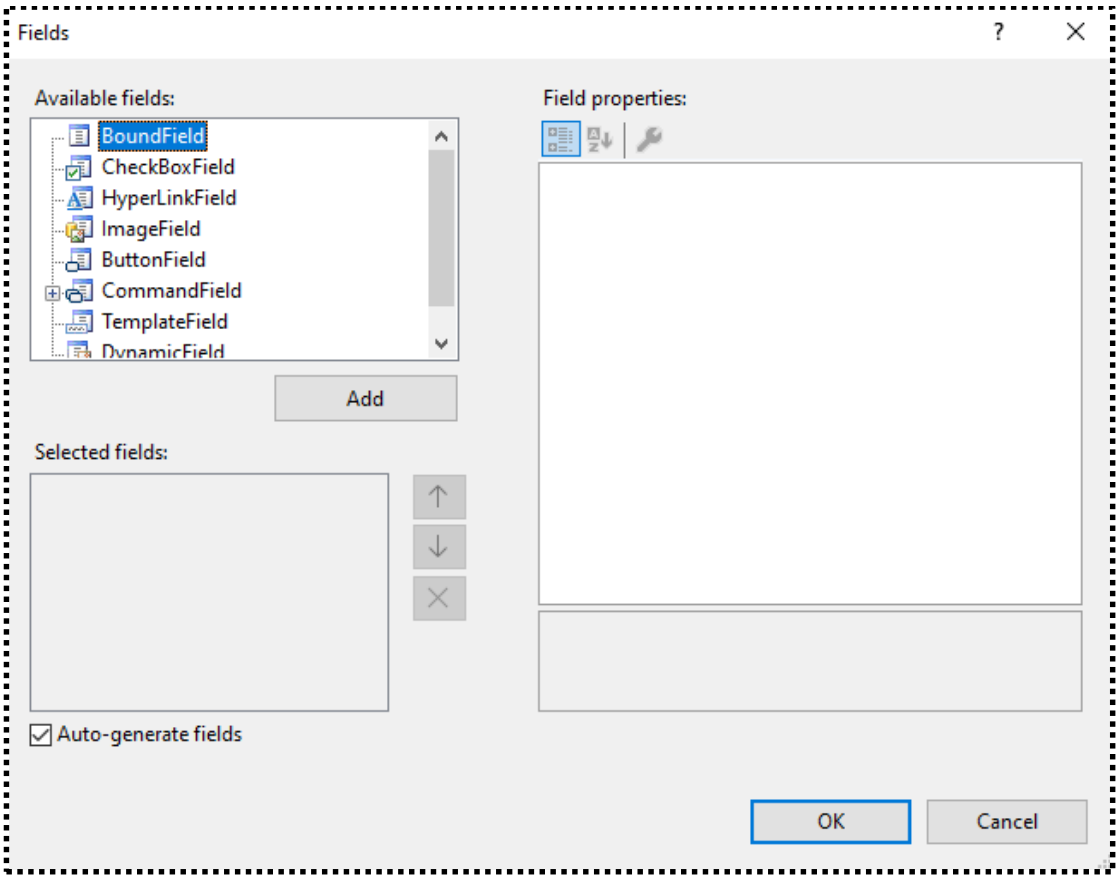
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="Gridview.WebForm1" %>

<!DOCTYPE html>

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

         <asp:GridView ID="GridView1" runat="server" AllowPaging="True"
OnPageIndexChanging="GridView1_PageIndexChanging" AutoGenerateColumns="False" Font-
Names="Comic Sans MS">
            <Columns>
               <asp:BoundField DataField="ID" HeaderText="ID" />
               <asp:BoundField DataField="Name" HeaderText="Name" />
            </Columns>
         </asp:GridView>

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

WebForm1.aspx.cs

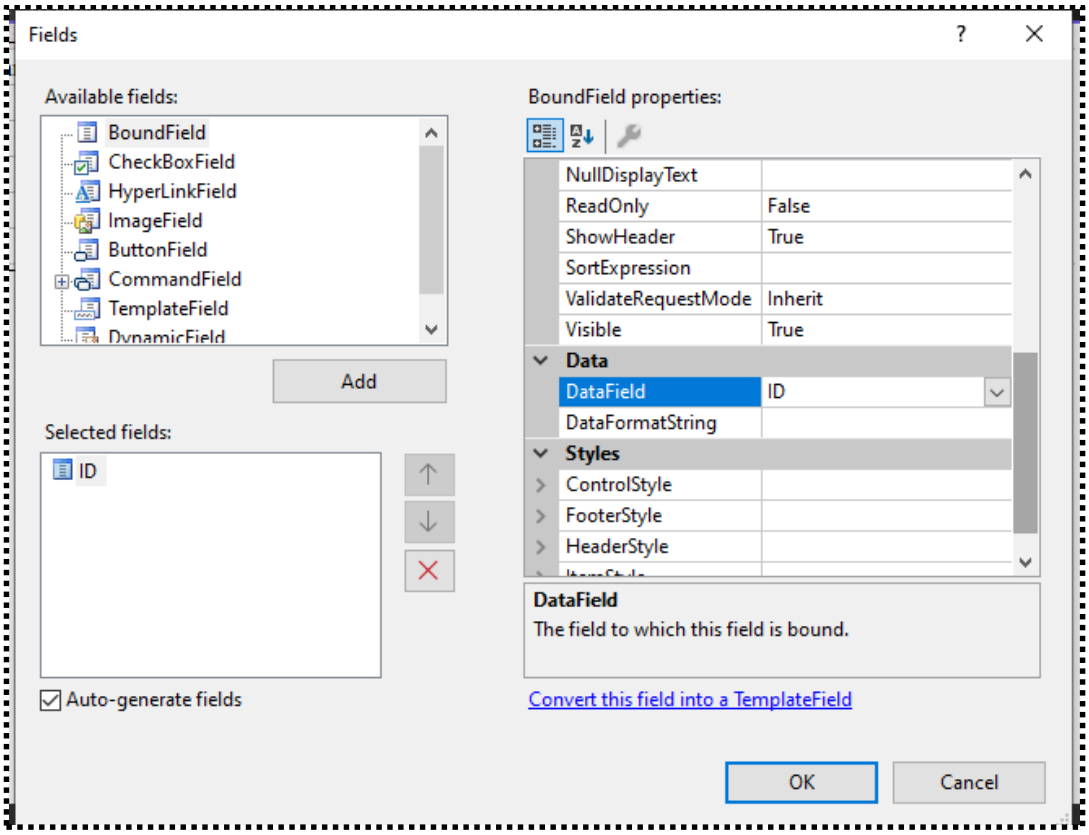
```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Gridview
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
            LoadGridData();
      }
      private void LoadGridData()
      {
         DataTable dt =new DataTable();
         dt.Columns.Add("ID");
         dt.Columns.Add("Name");
         for (int i=0; i < 10; i++)
         {
            DataRow dr= dt.NewRow();
            dr["ID"] = i +1;
            dr["Name"] = "Student" + (i + 1);
            dt.Rows.Add(dr);
         }
         GridView1.DataSource = dt;
         GridView1.DataBind();

      }
       protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
      {
```

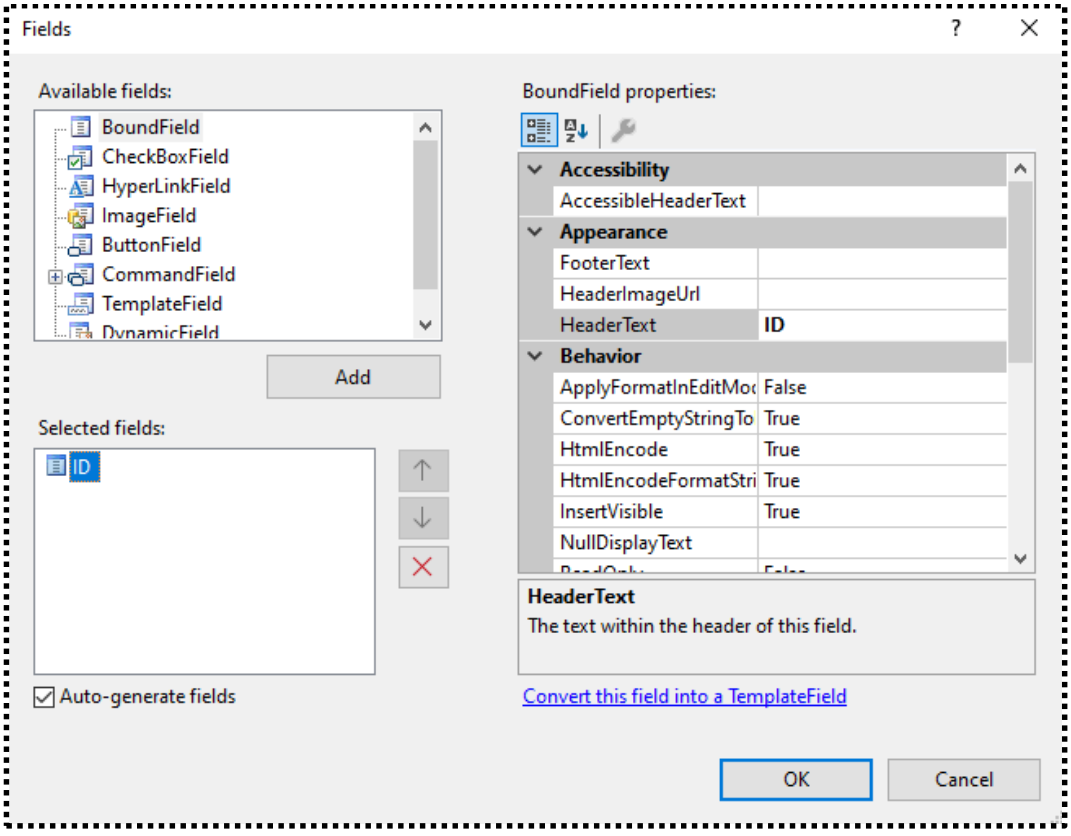
```
        GridView1.PageIndex= e.NewPageIndex;
        LoadGridData();
    }
  }
}
```

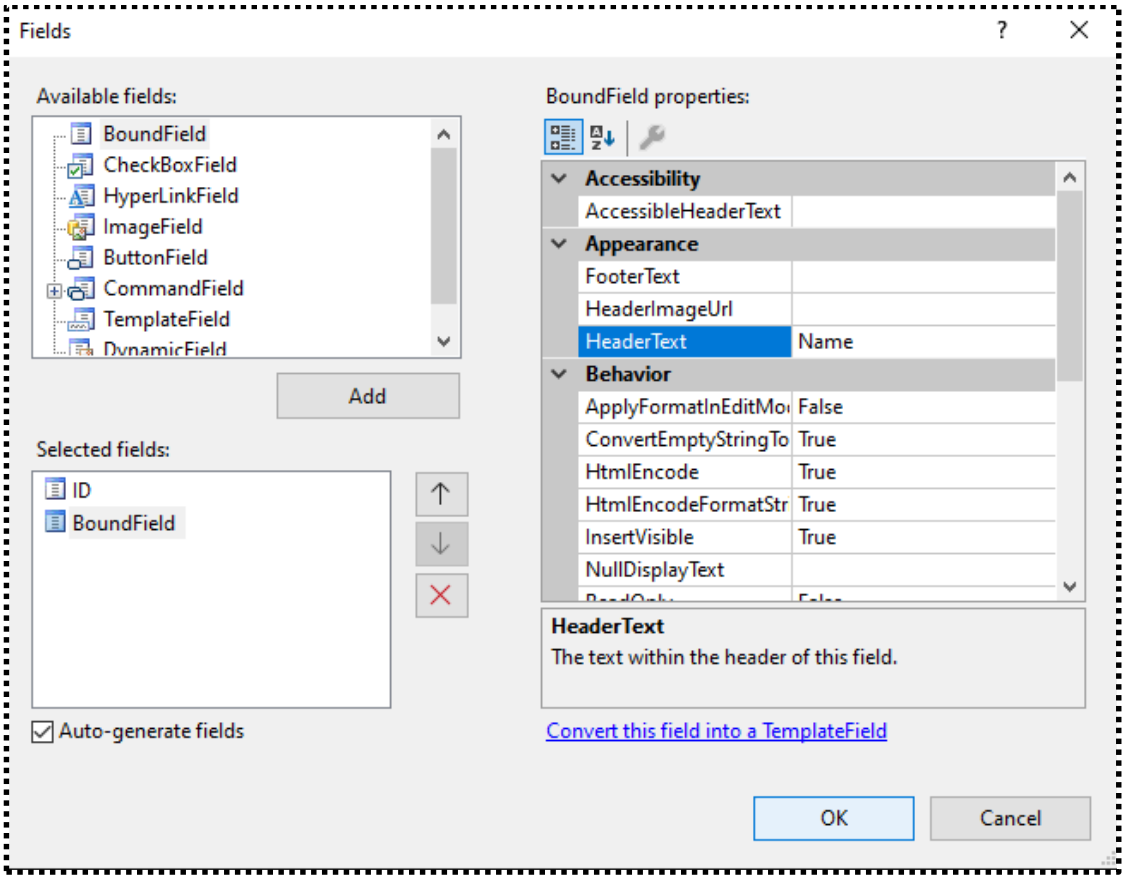




HeaderText & datafield:






Fields
Available fields:
BoundField
CheckBoxField
HyperLinkField
ImageField
ButtonField
CommandField
TemplateField
DynamicField
Add
Selected fields:
ID
BoundField
Auto-generate fields
BoundField properties:
Accessibility
AccessibleHeaderText
Appearance
FooterText
HeaderImageUrl
HeaderText
Name
Behavior
ApplyFormatInEditMo
False
ConvertEmptyStringTo
True
HtmlEncode
True
HtmlEncodeFormatStr
True
InsertVisible
True
NullDisplayText
HeaderText
The text within the header of this field.
Convert this field into a TemplateField
OK
Cancel

Output


localhost:44337/WebForm1.aspx
ID Name
1 Student1
2 Student2
3 Student3
4 Student4
5 Student5
6 Student6
7 Student7
8 Student8
9 Student9
10 Student10
localhost:44337/WebForm1.aspx
ID Name
1 Student1
2 Student2
3 Student3
4 Student4
5 Student5
6 Student6
7 Student7
8 Student8
9 Student9
10 Student10
localhost:44337/WebForm1.aspx
ID Name
1 6411
2 6412
3 6413
4 6414
5 6415
6 6416
7 6417
8 6418
9 6419
10 64110

9) C) Create a web application to demonstrate use of GridView control template and GridView hyperlink.

Source code

PRAC9C.PRAC:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="PRAC9.aspx.cs"
Inherits="PRAC9C.PRAC9" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowCommand="GridView1_RowCommand">
            <Columns>
               <asp:HyperLinkField DataNavigateUrlFields="ID" DataTextField="Name"
HeaderText="Name" DataNavigateUrlFormatString="~\WebForm2.aspx?ID={0}" />
               <asp:BoundField DataField="ID" HeaderText="ID" />
               <asp:BoundField DataField="Country" HeaderText="Country" />
               <asp:TemplateField>
                  <ItemTemplate>
                     <asp:Button ID="Button1" runat="server" Text="Select"
CommandName="Select" CommandArgument="<%#Container.DataItemIndex %>" />
                  </ItemTemplate>
               </asp:TemplateField>
            </Columns>
         </asp:GridView>
      </div>
   </form>
</body>
</html>
```

PRAC9.aspx.cs

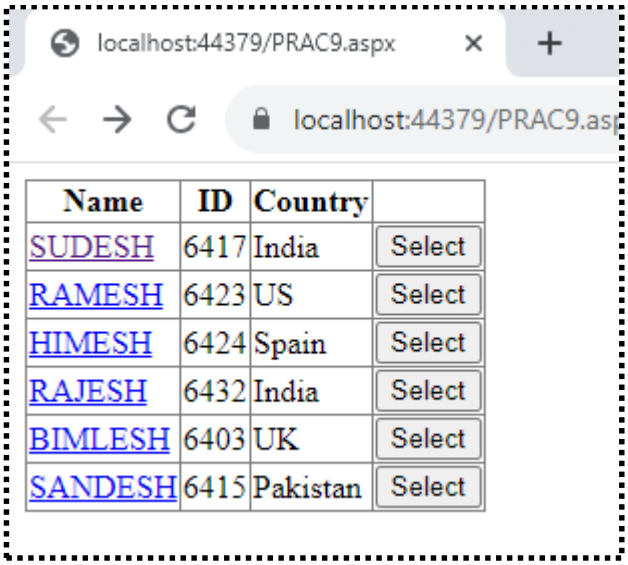
```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace PRAC9C
{
   public partial class PRAC9 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!this.IsPostBack)
         {
            DataTable dt = new DataTable();
            dt.Columns.AddRange(new DataColumn[3] { new DataColumn("ID"), new
DataColumn("Name"), new DataColumn("Country") });
            dt.Rows.Add("6417", "SUDESH", "India");
            dt.Rows.Add("6423", "RAMESH", "US");
            dt.Rows.Add("6424", "HIMESH", "Spain");
            dt.Rows.Add("6432", "RAJESH", "India");
            dt.Rows.Add("6403", "BIMLESH", "UK");
            dt.Rows.Add("6415", "SANDESH", "Pakistan");
            GridView1.DataSource = dt;
            GridView1.DataBind();
         }


      }

      protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
      {
         if (e.CommandName == "Select")
         {
            int rowIndex = Convert.ToInt32(e.CommandArgument);
```

```
            GridViewRow row = GridView1.Rows[rowIndex];
            string ID = row.Cells[1].Text;

            string Country = row.Cells[2].Text;
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('ID: " + ID + "
\\nCountry : " + Country + "')", true);
        }

    }
  }
}
```

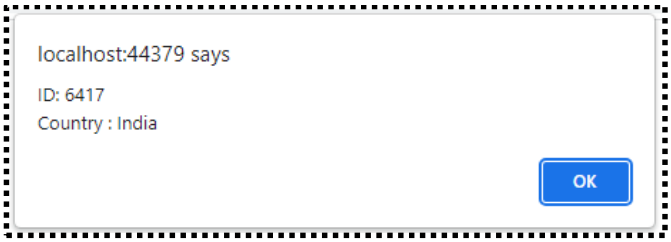
Webform2.aspx:

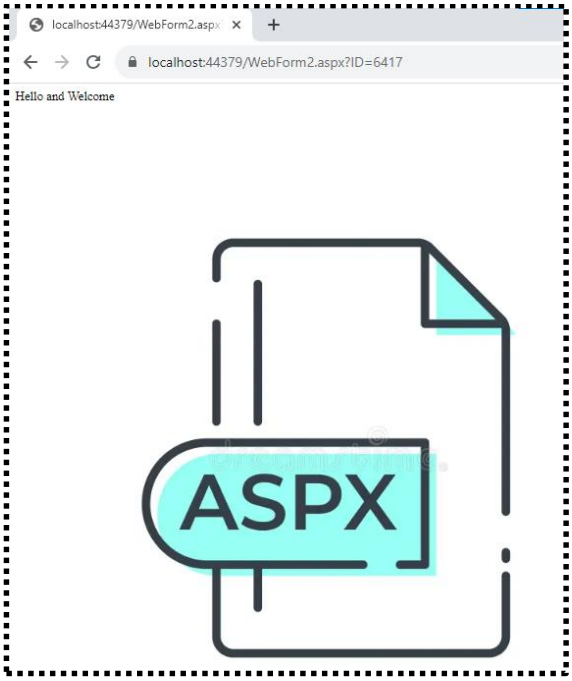
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="PRAC9C.WebForm2" %>

<!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="Hello and Welcome"></asp:Label><br /><br
/><br />

      <img src="https://thumbs.dreamstime.com/b/aspx-file-format-icon-extension-line-
189463972.jpg" />

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

Output:



Name	ID	Country	
SUDESH	6417	India	Select
RAMESH	6423	US	Select
HIMESH	6424	Spain	Select
RAJESH	6432	India	Select
BIMLESH	6403	UK	Select
SANDESH	6415	Pakistan	Select





PRACTICAL NO 10 - Working with AJAX and XML

10) A) Create a web application to demonstrate reading and writing operations with XML.

Source code

6417_Prac10a.aspx

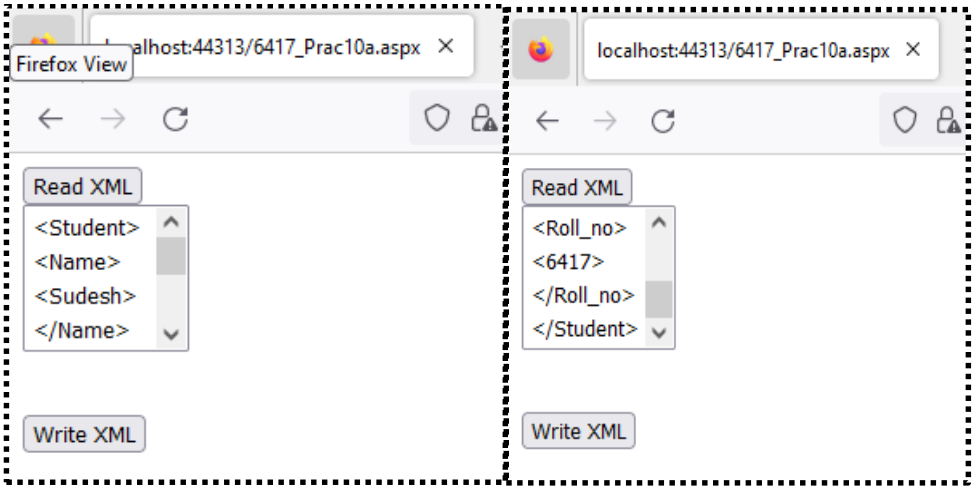
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="6417_Prac10a.aspx.cs"
Inherits="WebApplication2._6417_Prac10a" %>

<!DOCTYPE html>

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



         <asp:Button ID="read_btn" runat="server" Text="Read XML" OnClick="read_btn_Click"
/>
         <br />

         <asp:ListBox ID="ListBox1" runat="server"></asp:ListBox>
          <asp:XmlDataSource ID="XmlDataSource1" runat="server"
DataFile="~/myFile.xml"></asp:XmlDataSource>
          <br/> <br /><br/>
         <asp:Button ID="write_btn" runat="server" Text="Write XML" OnClick="write_btn_Click"
/>



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

6417_Prac10a.aspx.cs

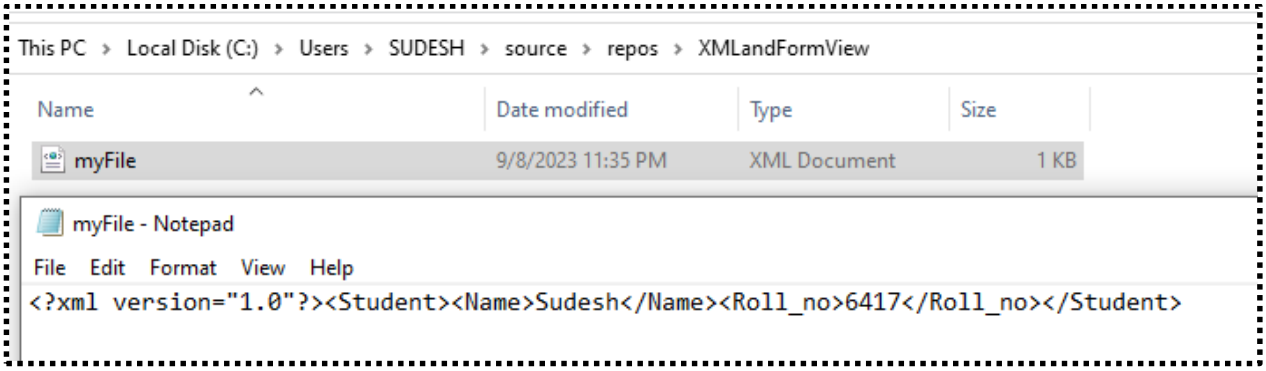
```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Xml;

namespace WebApplication2
{
   public partial class _6417_Prac10a : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }

      protected void read_btn_Click(object sender, EventArgs e)
      {
         string xml = "C:\\Users\\SUDESH\\source\\repos\\XMLandFormView\\myFile.xml";
         XmlReader rd = XmlReader.Create(xml);
         while (rd.Read())
         {
            switch (rd.NodeType)
            {
               case XmlNodeType.Element:
                  ListBox1.Items.Add("<" + rd.Name + ">");
                  break;
               case XmlNodeType.Text:
                  ListBox1.Items.Add("<" + rd.Value + ">");
                  break;
               case XmlNodeType.EndElement:
                  ListBox1.Items.Add("</" + rd.Name + ">");
                  break;
            }         }
         rd.Close();
      }
```

```
    protected void write_btn_Click(object sender, EventArgs e)
    {
        XmlTextWriter tw = new
XmlTextWriter("C:\\Users\\SUDESH\\source\\repos\\XMLandFormView\\myFile.xml", null);
        tw.WriteStartDocument();
        tw.WriteStartElement("Student");
        tw.WriteStartElement("Name", "");
        tw.WriteString("Sudesh");
        tw.WriteEndElement();
        tw.WriteStartElement("Roll_no", "");
        tw.WriteString("6417");
        tw.WriteEndElement();
        tw.WriteEndElement();
        tw.WriteEndDocument();
        tw.Close();
    }  }}
```

output



myFile.xml:



10) B) Create a web application to demonstrate use of various Ajax controls.

1. Update panel

Source code

UpdatePanel.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="6417_Updatepanel.aspx.cs"
Inherits="_6417_prac10b._6417_Updatepanel" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>6417_Sudesh Rajbhar</title>

</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:ScriptManager ID="ScriptManager1" runat="server" />
         <br />

         <fieldset>
           6417_FIRST PANEL:<asp:UpdatePanel runat="server" ID="UpdatePanel"
UpdateMode="Conditional">
            <Triggers>
               <asp:AsyncPostBackTrigger ControlID="UpdateBtn2" EventName="Click" />
            </Triggers>
            <ContentTemplate>
               <asp:Label runat="server" ID="Label1" />
               <asp:Button runat="server" ID="UpdateBtn1" OnClick="UpdateBtn_Click"
Text="Update" />
               <br />

               <br />
            </ContentTemplate>
         </asp:UpdatePanel>
         </fieldset>

         <fieldset>
```

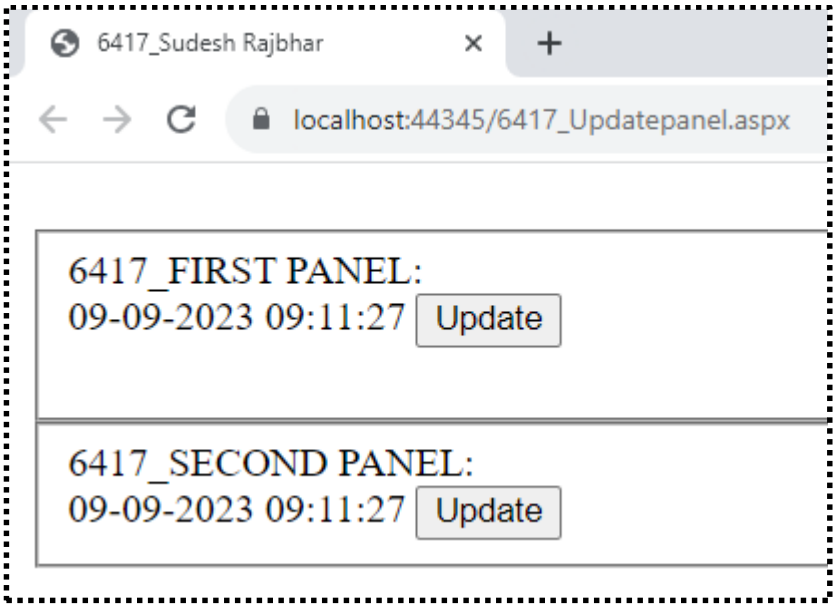
```
            6417_SECOND PANEL:

        <!--Second panel-->
        <asp:UpdatePanel runat="server" ID="UpdatePanel1" UpdateMode="Conditional">
           <ContentTemplate>
              <asp:Label runat="server" ID="Label2" />
              <asp:Button runat="server" ID="UpdateBtn2" OnClick="UpdateBtn_Click"
Text="Update" />
           </ContentTemplate>
        </asp:UpdatePanel>
        </fieldset>



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

Updatepanel.aspx.cs

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

namespace _6417_prac10b
{
   public partial class _6417_Updatepanel : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }
```

```
    protected void UpdateBtn_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
        Label2.Text = DateTime.Now.ToString();
    }
  }
}
```

Output:



2. Timer Control.

Timer.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Timer.aspx.cs"
Inherits="_6417_prac10b.Timer" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
         <asp:UpdatePanel ID="UpdatePanel1" runat="server">
            <ContentTemplate>
               <asp:Panel ID="Panel1" runat="server" BackColor="#FFFF99"
                  BorderStyle="Ridge" BorderColor="#FF3300" BorderWidth="5">
                  <center>
                     <asp:Label ID="label2" runat="server" BackColor="#CCFFFF"
                        Text="Label" Font-Bold="False" Font-Names="Algerian"
ForeColor="#FF3300"></asp:Label>
                     <asp:Timer ID="timer1" runat="server" Interval="1000"></asp:Timer>
                  </center>
               </asp:Panel>
            </ContentTemplate>
         </asp:UpdatePanel>
      </div>
   </form>
</body>
</html>
```

Timer.aspx.cs

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

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

namespace _6417_prac10b
{
   public partial class Timer : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         label2.Text = "6417_Page Refreshed at:" + DateTime.Today.ToLongDateString() + "Time: "
+ DateTime.Now.ToLongTimeString();
      }
   }
}
```

OUTPUT:

3. Update Progress Timer Control.

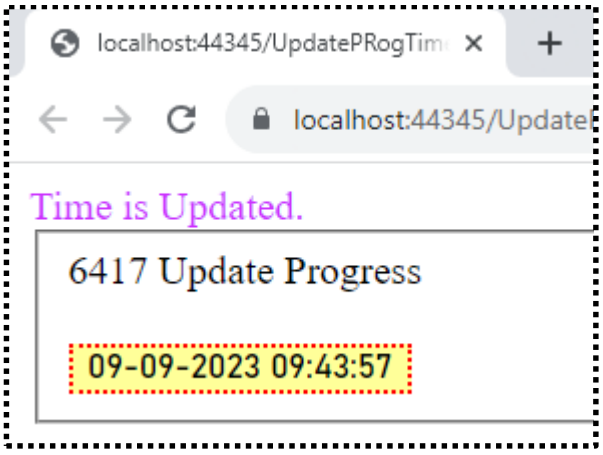
UpdatePRogTimer.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="UpdatePRogTimer.aspx.cs"
Inherits="_6417_prac10b.UpdatePRogTimer" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
         <asp:UpdateProgress ID="UpdateProgress1" runat="server">
            <ProgressTemplate>
               <asp:Label ID="Label2" runat="server" Text="Time is Updated."
                  ForeColor="#CC33FF"></asp:Label>
            </ProgressTemplate>
         </asp:UpdateProgress>
         <fieldset>
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
            <ContentTemplate>
               6417 Update Progress
               <br />
               <asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Names="Bahnschrift"
ForeColor="#FFFF99" ></asp:Label>
               <br />
               <asp:Button ID="Button1" runat="server" Text="Time to Refresh "
BackColor="#FFFF99" BorderColor="Red" BorderStyle="Dotted" Font-Names="Bahnschrift" />
            </ContentTemplate>
         </asp:UpdatePanel>
         </fieldset>

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

UpdatePRogTimer.aspx.cs:

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

namespace _6417_prac10b
{
   public partial class UpdatePRogTimer : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         System.Threading.Thread.Sleep(10000);
         Button1.Text = DateTime.Now.ToString();
      }
   }
}
```

Output:



PRACTICAL NO 11 - Programs to create and use DLL

11) A) Create a web application to create a calculator using dll.

6417_dll.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="6417_dll.aspx.cs"
Inherits="_6417_Sudesh_dll._6417_dll" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>CALCULATOR</title>
   <style>
      button {
         width: 100%;
         box-sizing: border-box; /* Ensure padding and borders don't affect the width */
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <fieldset>
         <table>
            <tr>
               <td>
                  <asp:Label ID="Label1" runat="server" Text="Calculator"></asp:Label>
               </td>
            </tr>
            <tr>
               <td>
                  <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
               </td>
               <td>
                  <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
               </td>
            </tr>
```

```
            <tr>
              <td>
                <asp:Button ID="add_btn" runat="server" Text="ADD" OnClick="add_btn_Click"
/>
              </td>

              <td>
                <asp:Button ID="sub_btn" runat="server" Text="SUB" OnClick="sub_btn_Click"
/>
              </td>
            </tr>
            <tr>
              <td>
                <asp:Button ID="mul_btn" runat="server" Text="MUL" OnClick="mul_btn_Click"
/>
              </td>

              <td>
                <asp:Button ID="div_btn" runat="server" Text="DIV" OnClick="div_btn_Click"
/>
              </td>
            </tr>

          <tr>
            <td>
              <asp:Label ID="out_lbl" runat="server" Text=""></asp:Label>
            </td>
          </tr>


        </table>
</fieldset>

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

6417_dll.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using prac11_dll;


namespace _6417_Sudesh_dll
{
   public partial class _6417_dll : System.Web.UI.Page
   {

      double a, b, output;
      Class1 cf = new Class1();

      protected void sub_btn_Click(object sender, EventArgs e)
      {
         a = Convert.ToDouble(TextBox1.Text);
         b = Convert.ToDouble(TextBox2.Text);


         output = cf.sub(a, b);
         out_lbl.Text = "Answer : " + output;
      }

      protected void mul_btn_Click(object sender, EventArgs e)
      {
         a = Convert.ToDouble(TextBox1.Text);
         b = Convert.ToDouble(TextBox2.Text);


         output = cf.mul(a, b);
         out_lbl.Text = "Answer : " + output;
      }

      protected void div_btn_Click(object sender, EventArgs e)
      {
         a = Convert.ToDouble(TextBox1.Text);
         b = Convert.ToDouble(TextBox2.Text);
```

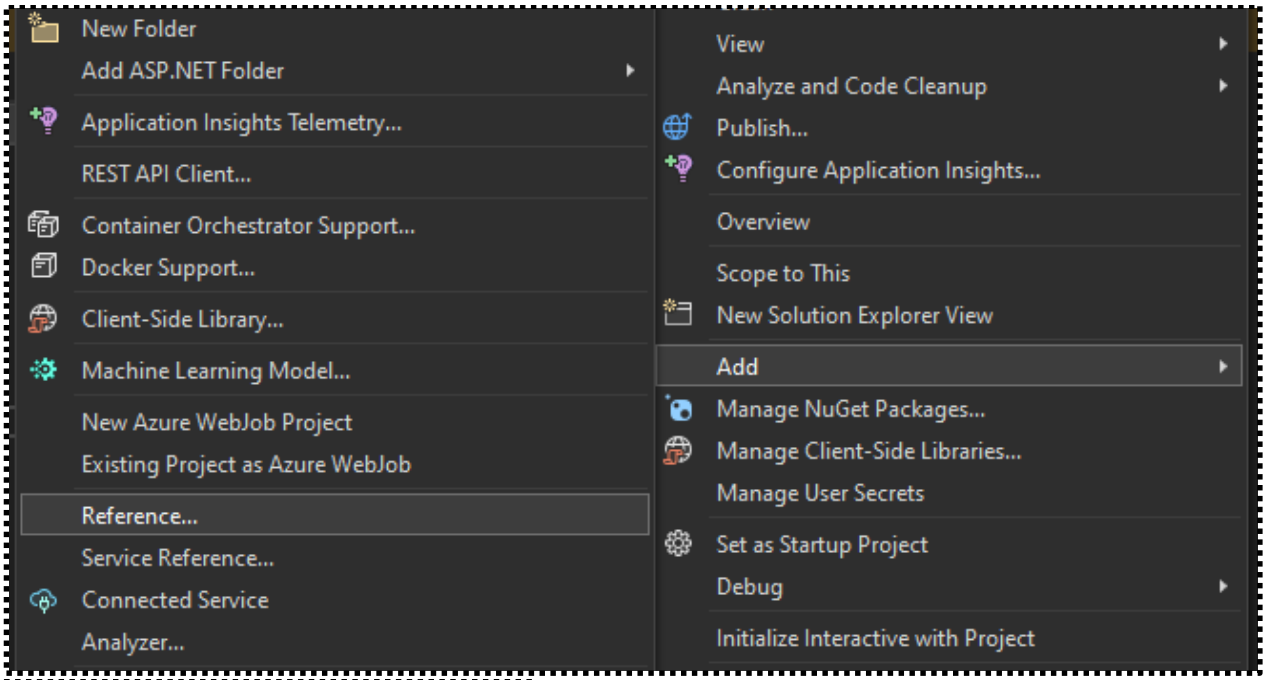
```
        output = cf.div(a, b);
        out_lbl.Text = "Answer : " + output;
    }

    protected void add_btn_Click(object sender, EventArgs e)
    {
        a = Convert.ToDouble(TextBox1.Text);
        b = Convert.ToDouble(TextBox2.Text);


        output = cf.add(a, b);
        out_lbl.Text = "Answer : " + output;
    }
    protected void Page_Load(object sender, EventArgs e)
    {
    }
  }
}
```

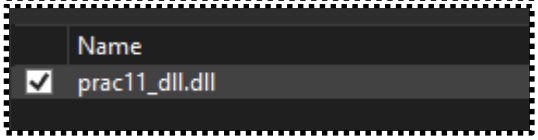
Reference dll file:

Class1.cs:

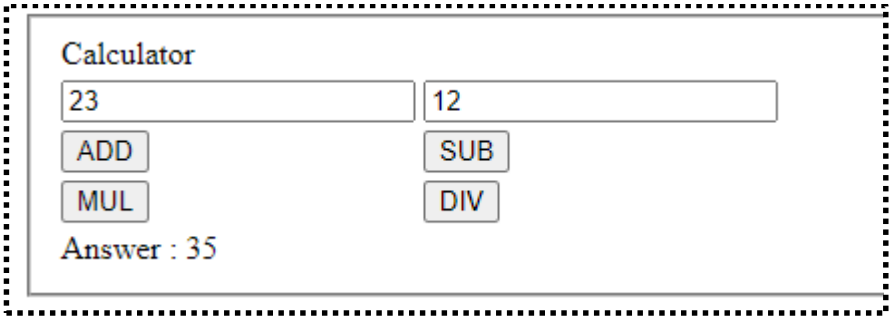
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace prac11_dll
{
  public class Class1
  {
    public double add(double a, double b)
    {
      return a + b;
    }
    public double sub(double a, double b)
    {
      return a - b;
    }
    public double mul(double a, double b)
    {
      return a * b;
```

```
    }
    public double div(double a, double b)
    {
        return a / b;
    }
  }
}
```

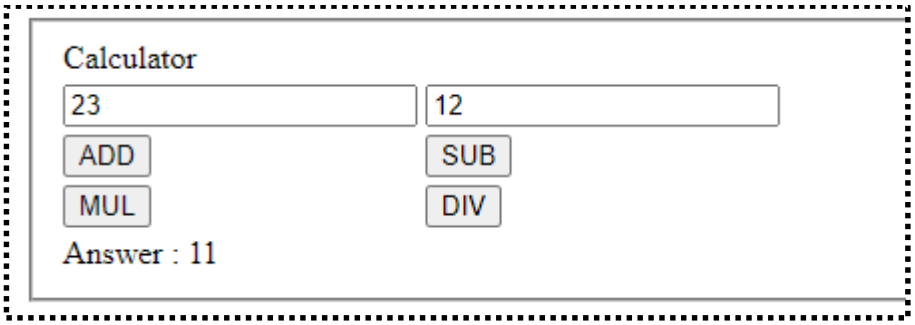




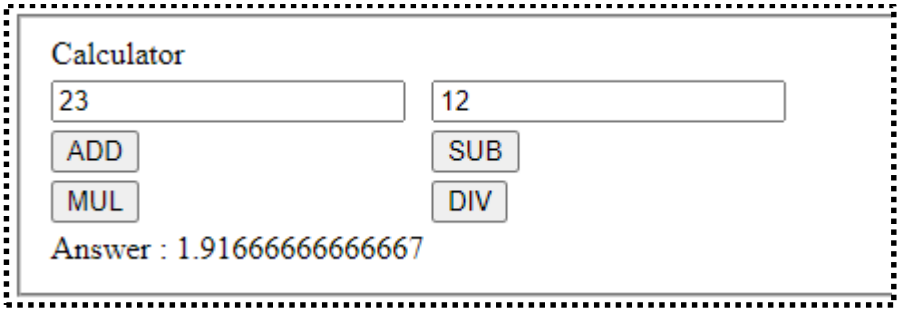
ADD:



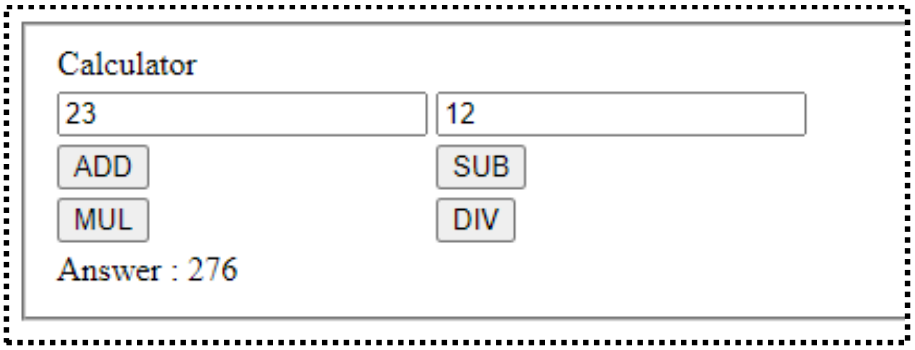
SUB:


Calculator
23
12
ADD
SUB
MUL
DIV
Answer : 11

DIV:


Calculator
23
12
ADD
SUB
MUL
DIV
Answer : 1.91666666666667

MUL:


Calculator
23
12
ADD
SUB
MUL
DIV
Answer : 276