# FYMCA SEM-II

# AWT Lab

# JOURNAL

**LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER APPLICATION**

**ISO 9001-2008 CERTIFIED**

**S.N. 341, Next to New English School, Govt. Colony,**
**Bandra (East), Mumbai – 400051,**
**Tel: 91-22-26570892/3181**

**Date:**

# CERTIFICATE

**This is to certify that Mr. Saurabh Bhausaheb Gawali**

**Roll No. 2021105 is a student of MCA of 1st year Semester-II has completed successfully full-semester practical/assignments of subject AWT Lab for the academic year 2021 – 22**

**Subject In-Charge** **External Examiner**

**Director**

**TABLE OF CONTENTS**


| No | Practical Title | Date |
|---|---|---|
| 1 | Design Applications using Classes and Objects | 28-April-2022 |
| 2 | Design Applications using Inheritance and Abstract Classes | 28-April-2022 |
| 3 | Design UI based applications using basic Windows forms Controls | 03-May-2022 |
| 4 | Design UI based applications using basic Windows forms Controls | 03-May-2022 |
| 5 | Design UI based applications using basic Windows forms Controls | 10-May-2022 |
| 6 | Design Applications using Inheritance and Abstract Classes | 12-May-2022 |
| 7 | Design a Web Application for an Organization with Registration forms and advanced controls | 14-May-2022 |
| 8 | Create website using master page concept. | 02-June-2022 |
| 9 | Build an angular web application. | 07-June-2022 |
| 10 | Design a webpage to demonstrate a connection oriented architecture. | 09-June-2022 |
| 11 | Design a webpage to demonstrate a disconnected architecture. | 14-June-2022 |
| 12 | Create a webpage that demonstrates the use of data bound controls of ASP.NET. | 16-June-2022 |
| 13 | Design a webpage to demonstrate the working of a simple stored procedure. | 21-June-2022 |
| 14 | Design a webpage to demonstrate the working of parameterized stored procedure. | 23-June-2022 |
| 15 | Design a webpage to display the use of LINQ. | 28-June-2022 |
| 16 | Build websites to demonstrate the working of entity framework in dot net. | 05-July-2022 |

| Practical No.1 | | Date : 28/04/2022 |
|---|---|---|
| **Aim:** | To understand the class in c-sharp & inheritance in c-sharp | |

Write program that will create class Person. Which has following data member
1. Name
2. Age
Implements instance method, class method and constructor (overloading), properties of that class
Instance methods are
1. void getData()
2. void showData()

**Source Code :** #person.cs

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

namespace practical1
{
    internal class Person
    {
        //data members
        private string name;
        private int age;

        //Properties(getter and setter method)
        public string Name { get { return name; } set { name = value; } }
        public int Age { get { return age; } set { age = value; } }

        // non-parameterized constructor
        public Person()
        {
            name = "aaa";
            age = 23;
        }

        //parameterized constructor
        public Person(String nam,int age)
        {
            this.name = nam;
            this.age = age;
        }

        //instance method1
        public void getdata()
        {
            Console.WriteLine("Enter the Name:");
            name =Console.ReadLine();
            Console.WriteLine("Enter Age:");
            age = Convert.ToInt16(Console.ReadLine());
        }
        public void showdata()
        {
            Console.WriteLine("Name = "+ name);
            Console.WriteLine("Age ="+ age);

        }
    }
}
```

*#Pragram.cs*

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

namespace practical1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Person p1 = new Person();//initialize with default constructor
            Person p2=new Person("bbb",25);//initialized with parm constructor
            Person p3=new Person();// instance method
            Person p4=new Person();//parameterized method

            p4.Name = "ddd";
            p4.Age = 35;

            p3.getdata();

            p1.showdata();
            p2.showdata();
            p3.showdata();
            p4.showdata();

            Console.Read();
        }
    }
}
```

Output:

```
C:\Users\HP\Desktop\AWT\practical1\prac1\prac1\bin\Debug\prac1.exe
Enter the name:
Atul
Enter the age:
22
Name: raju
Age: 21
Name: yash
Age: 20
Name: Atul
Age: 22
Name: sidhu
Age: 19
```

| Practical No.2 | | Date : 28/04/2022 |
|---|---|---|
| **Aim:** | To understand the inheritance in C-Sharp. Implementation of Multilevel inheritance. | |

Write program that will create class Student which is inherited from person class. Which has following data member

1. rollnumber
2. percentage

Implements instance method and constructor, properties of that class

Instance methods are

1. void getData()
2. void showData()
3. inherit the base class constructor initialization
4. inherit the base instance method

**Source Code :** #person.cs

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

namespace practical1
{
    internal class Person
    {
//data members private
string name;private int age;

//Properties(getter and setter method)
public string Name { get { return name; } set { name = value; } }public
int Age { get { return age; } set { age = value; } }

// non-parameterized constructorpublic
Person()
{
            name = "aaa";
            age = 23;
}

//parameterized constructor public
Person(String nam,int age)
{
            this.name = nam;
            this.age = age;
}

//instance method1 public
void getdata()
{
            Console.WriteLine("Enter the Name:");
            name =Console.ReadLine();
            Console.WriteLine("Enter Age:");
            age = Convert.ToInt16(Console.ReadLine());
}
public void showdata()
{
            Console.WriteLine("Name = "+ name);
            Console.WriteLine("Age ="+ age);

}
    }
}
```

*#Pragram.cs*

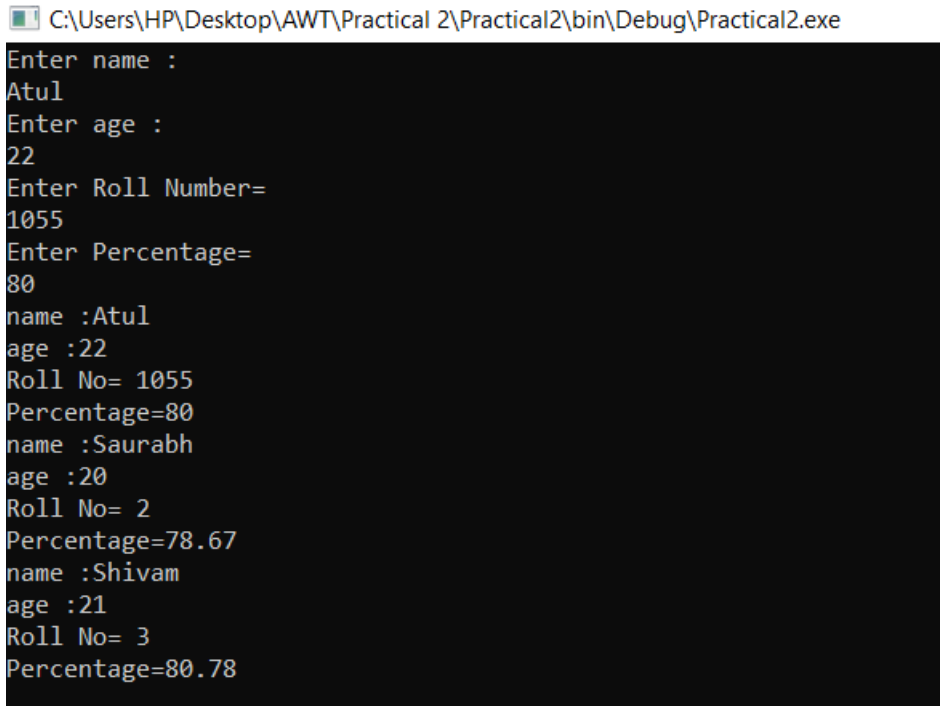
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace practical1
{
    internal class Program
    {
static void Main(string[] args)
{
            student s = new student();//initializes through default constructor
            s.showdata();
            Console.WriteLine();

            student s1 = new student(2,89.20,"Ram",20);//initialize throught
parameterized constructor
            s1.showdata();
            Console.WriteLine();

            student s2 = new student();//initialize throught properties
            s2.Name = "Radhika";
            s2.Age = 23;
            s2.Rollno = 3;
            s2.Percentage = 75.25;

            s2.showdata();

            Console.Read();
}
    }
}
```

*#student.cs*

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

namespace practical1
{
    internal class student:Person
    {
//data memebers private int
rollno;
private double percentage;

// properties
public int Rollno { get { return rollno; } set { rollno = value; } } public
double Percentage { get { return percentage; } set { percentage
= value; } }

//default constructor public
student():base()
{
            rollno = 1;
            percentage = 99.99;
}

// parameterized constructor
public student(int rn,double per,string name,int age):base(name,age)
{
            this.rollno = rn;
            this.percentage = per;
```

```
}

//instance method1 public
void getdata()
{
            base.getdata();
            Console.WriteLine("Enter your rollno:");
            rollno = Convert.ToInt16(Console.ReadLine());
            Console.WriteLine("Enter your percentage:");
            percentage = Convert.ToDouble(Console.ReadLine());
}

//instance method2 public void
showdata()
{
            base.showdata();
            Console.WriteLine("Roll No. is:"+rollno);
            Console.WriteLine("percentage is:"+percentage);
}
    }
}
```

**Output:**



| Practical No.3 | | Date : 03/05/2022 |
|---|---|---|
| **Aim:** | To understand file handling in C-Sharp | |

Write program that will create class Student. Which has following data member
1. rollno
2. name
3. bdate
4. percentage

Implements instance method and constructor, properties of that class Instance methods are
1. void getData()
2. void saveData()

saveData function(method) will generate student.csv file. Each object record will save in that file

**Source Code:** #person.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
namespace practical3
{
    internal class Person
    {
        //data members
        private string name;
        private int age;

        //Properties(getter and setter method)
        public string Name { get { return name; } set { name = value; } }
        public int Age { get { return age; } set { age = value; } }

        // non-parameterized constructor
        public Person()
        {
            name = "aaa";
            age = 23;
        }

        //parameterized constructor
        public Person(String nam,int age)
        {
            this.name = nam;
            this.age = age;
        }

        //instance method1
        public void getdata()
        {
            Console.WriteLine("Enter the Name:");
            name =Console.ReadLine();
            Console.WriteLine("Enter Age:");
            age = Convert.ToInt16(Console.ReadLine());
        }
        public void showdata()
        {
            Console.WriteLine("Name = "+ name);
            Console.WriteLine("Age ="+ age);

        }
    }
}
```

# Student.cs

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

namespace practical3
{
    internal class Student : Person
    {
            private int rollno;
            private double percentage;

            public Student() : base()
            {
                rollno = 1;
                percentage = 99.99;
            }
            public Student(int rn, double per, String nam, int ag) : base(nam,
ag)
            {
                this.rollno = rn;
                this.percentage = per;
            }
            public int Rollno { get { return rollno; } set { rollno = value; }
}
            public double Percentage { get { return percentage; } set {
percentage = value; } }

            public void getData()
            {
                base.getdata();
                Console.WriteLine("Enter the rollno");
                rollno = Convert.ToInt16(Console.ReadLine());
                Console.WriteLine("Enter the percentage");
                percentage = Convert.ToDouble(Console.ReadLine());
            }
            public void showData()
            {
                base.showdata();
                Console.WriteLine("Rollno=" + rollno);
                Console.WriteLine("Percentage=" + percentage);

            }
            public void saveData()
            {
                string currentdirectory = Directory.GetCurrentDirectory();
                FileStream fs = new FileStream(currentdirectory + "data.csv",
FileMode.Append);
                string record = rollno + "," + Name+ "," + Age+ "," +
percentage;
                StreamWriter sw = new StreamWriter(fs);
                sw.WriteLine(record);
                sw.Close();
                fs.Close();
                Console.WriteLine("Record store successfully");
            }
        }
    }
```

#program.cs

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

namespace practical3
{
    internal class Program
    {
        static Student s = new Student();
        static void Main(string[] args)
        {
            String ans = "Y";
            do
            {
                Console.Clear();
                int choice;
                Console.WriteLine("1. Add Student");
                Console.WriteLine("2. Read File");
                Console.WriteLine("3. Exit");
                Console.WriteLine("Enter your choice(1..3)");
                choice = Convert.ToInt16(Console.ReadLine());
                switch (choice)
                {
                    case 1:
                        addStudent();
                        break;
                    case 2:
                        readFile();
                        break;
                    case 3:
                        break;
                    default:
                        Console.WriteLine("Wrong choice ");
                        break;
                }
                Console.Write("Do you wish to continue(y/n)");
                ans = Console.ReadLine();
            } while (ans.Equals("Y") || ans.Equals("y"));
        }
        static void addStudent()
        {
            s.getData();
            s.saveData();
        }
        static void readFile()
        {
            string currentdir = Directory.GetCurrentDirectory();
            FileStream fs = new FileStream(currentdir + "data.csv",
FileMode.Open);
            StreamReader sr = new StreamReader(fs);
            sr.BaseStream.Seek(0, SeekOrigin.Begin);
            string record = sr.ReadLine();
            while (record != null)
            {
                Console.WriteLine(record);
                record = sr.ReadLine();
            }

            sr.Close();
```

```
            fs.Close();
        }
    }
}
```

**Output:**

C:\Users\HP\Desktop\AWT\Practical 3\Practical 3\bin\Debug\Practical 3.exe

```
1. Add Student
2. Read File
3. Exit
Enter your choice(1..3)
1
Enter the name
Atul
Enter the Age
21
Enter the Roll number
1044
Enter the Percentage
85
currentdirectory
Record store successfully
Do you wish to continue(y/n)
```

C:\Users\HP\Desktop\AWT\Practical 3\Practical 3\bin\Debug\Practical 3.exe

```
1. Add Student
2. Read File
3. Exit
Enter your choice(1..3)
2
1,Saurabh,22,99.4
1,Shiv ,21,99.4
1,Viraj,6,99.4
1,Saurabh gawali,22,99.4
1,Atul,21,99.4
Do you wish to continue(y/n)
```

*Untitled - Notepad

File Edit View

```
1,Saurabh,22,99.4
2,Shiv,21,99.4
3,Viraj,8,99.4
3,Atul,21,99.4
```

| Practical No.4 | | Date:03/05/2022 |
|---|---|---|
| **Aim** | Create GUI using Windows form for accepting the input from the user | |

Write program that will create class Student. Which has following data member

1. rollno
2. name
3. bdate
4. percentage

Implements instance method and constructor, properties of that class Instance methods are

1. void getData()
2. void saveData()

saveData function(method) will generate student.csv file. Each object record will save in that file
Refer following word file as template for GUI

**Source Code:** #person.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
namespace practical4
{
    internal class Person
    {
        //data members
        private string name;
        private int age;

        //Properties(getter and setter method)
        public string Name { get { return name; } set { name = value; } }
        public int Age { get { return age; } set { age = value; } }

        // non-parameterized constructor
        public Person()
        {
            name = "aaa";
            age = 23;
        }

        //parameterized constructor
        public Person(String nam,int age)
        {
            this.name = nam;
            this.age = age;
        }

        //instance method1
        public void getdata()
        {
            Console.WriteLine("Enter the Name:");
            name =Console.ReadLine();
            Console.WriteLine("Enter Age:");
            age = Convert.ToInt16(Console.ReadLine());
```

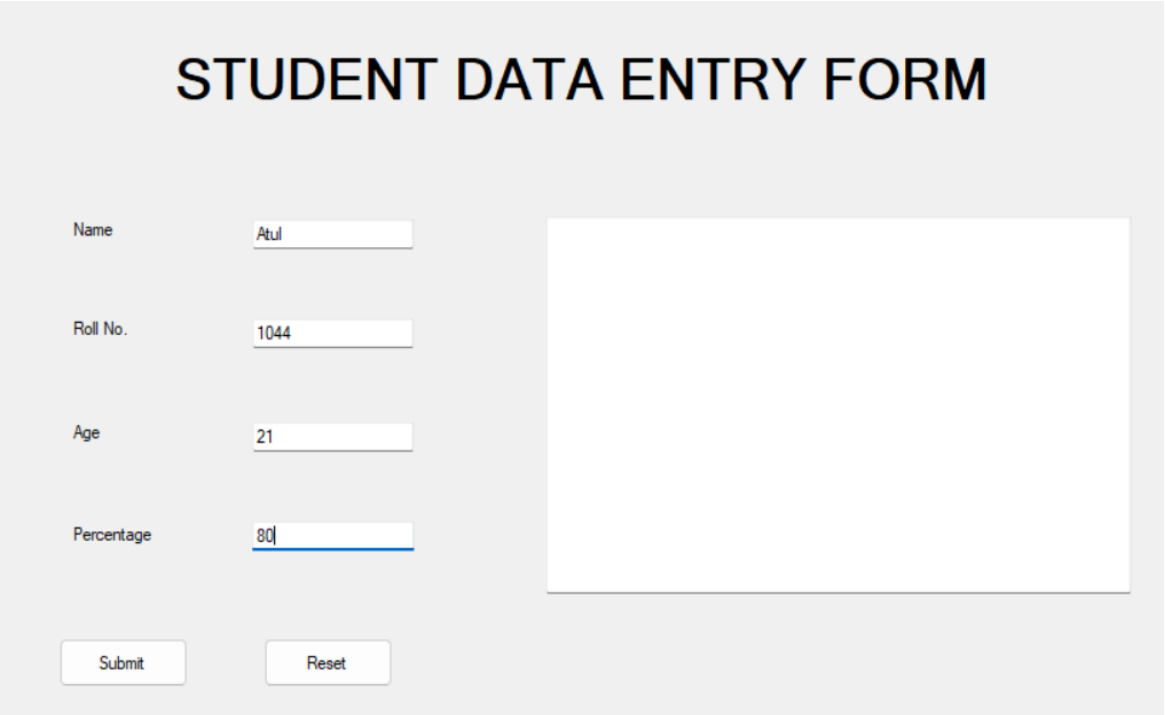
```
        }
        public void showdata()
        {
            Console.WriteLine("Name = "+ name);
            Console.WriteLine("Age ="+ age);

        }
    }
}
```

#Student.cs

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

namespace practical4
{
    internal class Student : Person
    {
            private int rollno;
            private double percentage;

            public Student() : base()
            {
                rollno = 1;
                percentage = 99.99;
            }
            public Student(int rn, double per, String nam, int ag) : base(nam,
ag)
            {
                this.rollno = rn;
                this.percentage = per;
            }
            public int Rollno { get { return rollno; } set { rollno = value; }
}
            public double Percentage { get { return percentage; } set {
percentage = value; } }

            public void getData()
            {
                base.getdata();
                Console.WriteLine("Enter the rollno");
                rollno = Convert.ToInt16(Console.ReadLine());
                Console.WriteLine("Enter the percentage");
                percentage = Convert.ToDouble(Console.ReadLine());
            }
            public void showData()
            {
                base.showdata();
                Console.WriteLine("Rollno=" + rollno);
                Console.WriteLine("Percentage=" + percentage);

            }
            public void saveData()
            {
                string currentdirectory = Directory.GetCurrentDirectory();
                FileStream fs = new FileStream(currentdirectory + "data.csv",
FileMode.Append);
                string record = rollno + "," + Name+ "," + Age+ "," +
percentage;
                StreamWriter sw = new StreamWriter(fs);
                sw.WriteLine(record);
```

```
                sw.Close();
                fs.Close();
                //Console.WriteLine("Record store successfully");
            }
        }
}
```

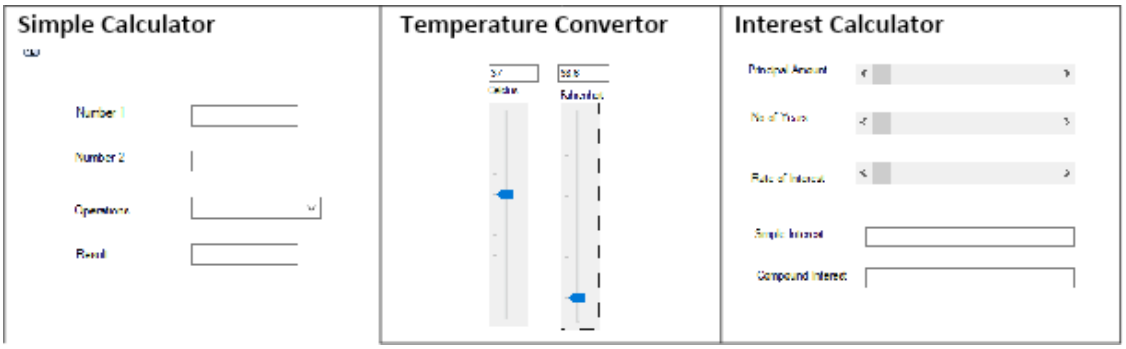
**Output:**



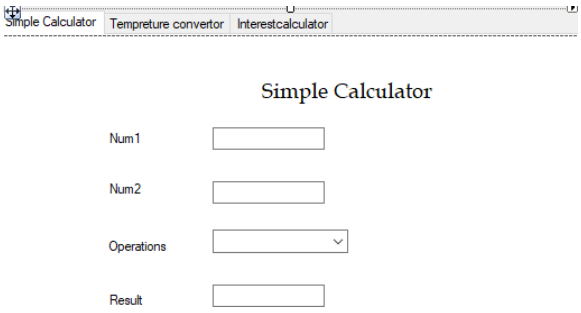
| Practical No.5 | | Date:10/05/2022 |
|---|---|---|
| **Aim:** | Understand the use of common controls in Windows Form | |

a. Create GUI using Windows form for simple calculator
b. Create GUI using Windows form for convert temperature from Celsius to Fahrenheit or visa-versa [use slider control]
c. Create GUI using Windows form to calculate simple interest & compound interest
d. Create 3 Tab where you can place above three practical in one Windows form
Refer following GUI for your practical and
use appropriate variable and events

Refer following word file as template for your practical use the GUI from it.




## Source Code:# (A)Simple Calculator

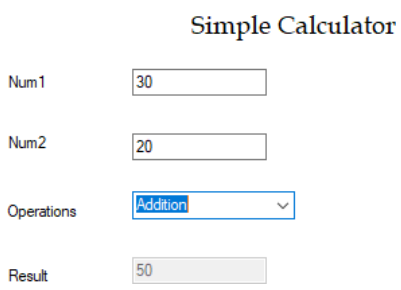



#Form.cs

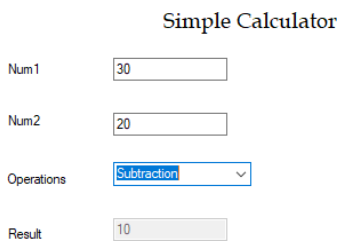
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace practical_5
{
    public partial class Form1 : Form
    {
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            int index = comboBox1.SelectedIndex;
            int num1, num2,result;
            num1 = Convert.ToInt16(txtnum1.Text);
            num2 = Convert.ToInt16(txtnum2.Text);

            switch(index)
            {
                case 0:
                    //add
                    result = num1 + num2;
                    txtresult.Text = result.ToString();
                    break;

                case 1:
                    //sub
                    result = num1 - num2;
                    txtresult.Text = result.ToString();
                    break;

                case 2:
                    //mul
                    result = num1 * num2;
                    txtresult.Text = result.ToString();
                    break;

                case 3:
                    //div
                    result = num1 / num2;
                    txtresult.Text = result.ToString();
                    break;
            }
}
}
}
```

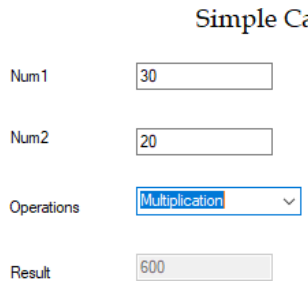
**Output:**



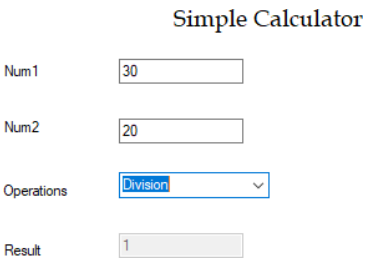

*Addition*




*Subtraction*

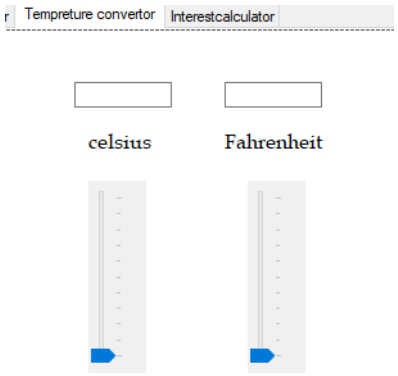



*Multiplication*



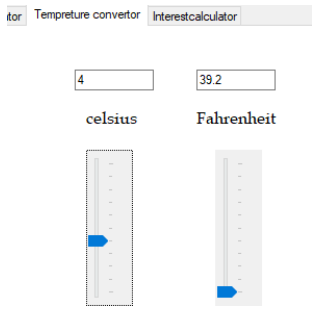

*Division*

#Form.cs [Design]
(B) Temperature Convertor

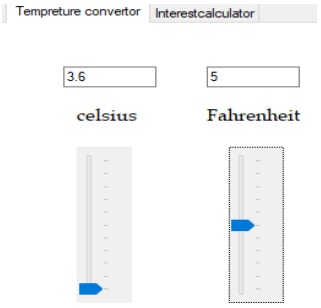



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace practical_5
{
    public partial class Form1 : Form
    {
      double celcious, fahrenheit, celciousToFahrenheit, farenheitToCelcious;
      private void tbCel_Scroll(object sender, EventArgs e)
        {
            celcious = tbCel.Value;
            txtCel.Text = celcious.ToString();

            celciousToFahrenheit = (celcious * 9) / 5 + 32;
            txtFah.Text =celciousToFahrenheit.ToString();
        }
        private void tbFah_Scroll(object sender, EventArgs e)
        {
            fahrenheit = tbFah.Value;
            txtFah.Text = fahrenheit.ToString();

            farenheitToCelcious = (fahrenheit - 32) / 5 + 9;
            txtCel.Text = farenheitToCelcious.ToString();
        }}
    }
```

#Output:

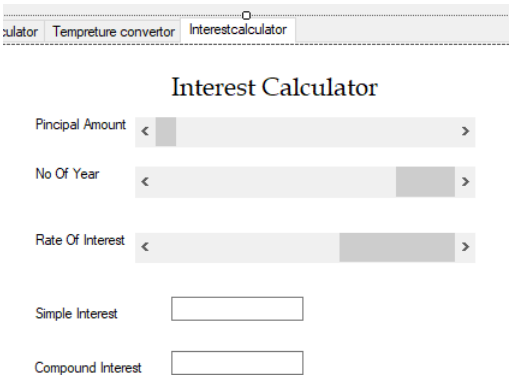



Celsius To Fahrenheit




Fahrenheit To Celsius

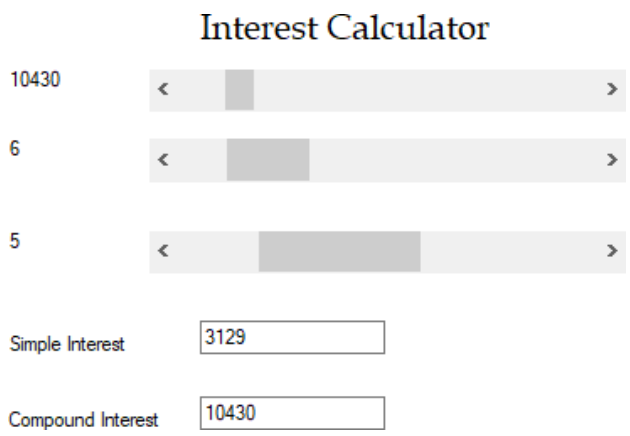
#Form.cs [Design]
Simple and Compound Interest




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace practical_5
{
    public partial class Form1: Form
    {
       int principalAmount, noofyear, interestRate, simpleinterest,
compoundinterest;

private void hScrollBar1_Scroll(object sender, ScrollEventArgs e)
        {
            principalAmount = hScrollBar1.Value;
            txtprincipal.Text = principalAmount.ToString();

            simpleinterest = ((principalAmount * noofyear * interestRate) /
100);
            txtSI.Text = simpleinterest.ToString();

            compoundinterest = (int)(principalAmount * Math.Pow(1 +
interestRate / 100, noofyear));
            txtCI.Text = compoundinterest.ToString();
        }
```

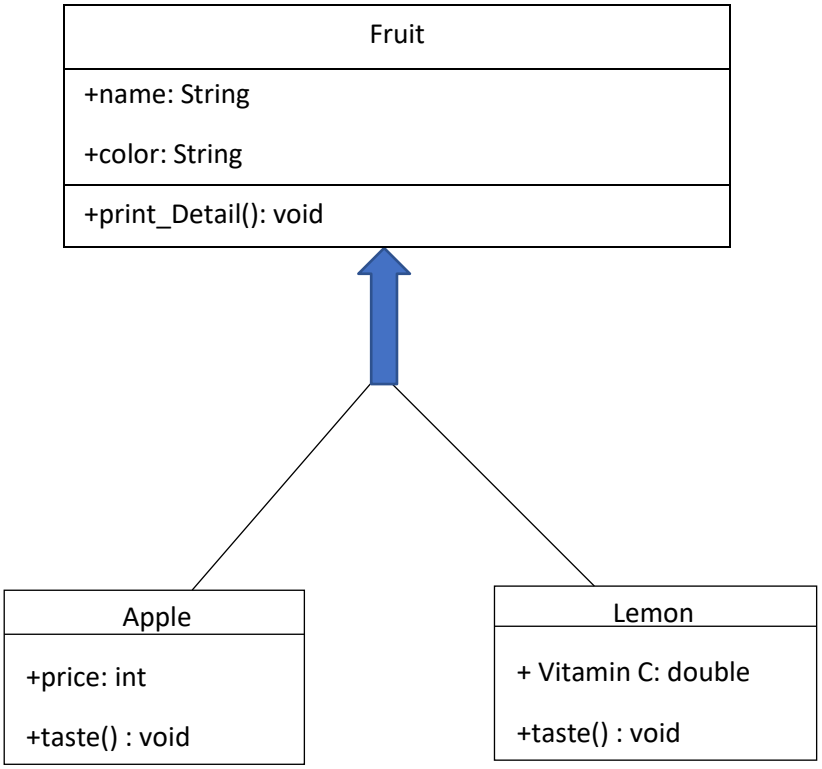
```
private void hScrollBar2_Scroll(object sender, ScrollEventArgs e)
        {
            noofyear = hScrollBar2.Value;
            txtyear.Text = noofyear.ToString();

            simpleinterest = ((principalAmount * noofyear * interestRate) /
100);
            txtSI.Text = simpleinterest.ToString();

            compoundinterest = (int)(principalAmount * Math.Pow(1 +
interestRate / 100, noofyear));
            txtCI.Text = compoundinterest.ToString();
        }

        private void hScrollBar3_Scroll(object sender, ScrollEventArgs e)
        {
            interestRate = hScrollBar3.Value;
            txtRate.Text = hScrollBar3.Value.ToString();

            simpleinterest = ((principalAmount * noofyear * interestRate) /
100);
            txtSI.Text = simpleinterest.ToString();

            compoundinterest = (int)(principalAmount * Math.Pow(1 +
interestRate / 100, noofyear));
            txtCI.Text = compoundinterest.ToString();

        }
   }
}
```

#Output:



| **Practical No:6 A** | | **Date:**12/05/2022 |
|---|---|---|
| **Aim:** | Design Applications using Inheritance and Abstract Classes | |

6(a): Write a program to demonstrate single inheritance for the following design:

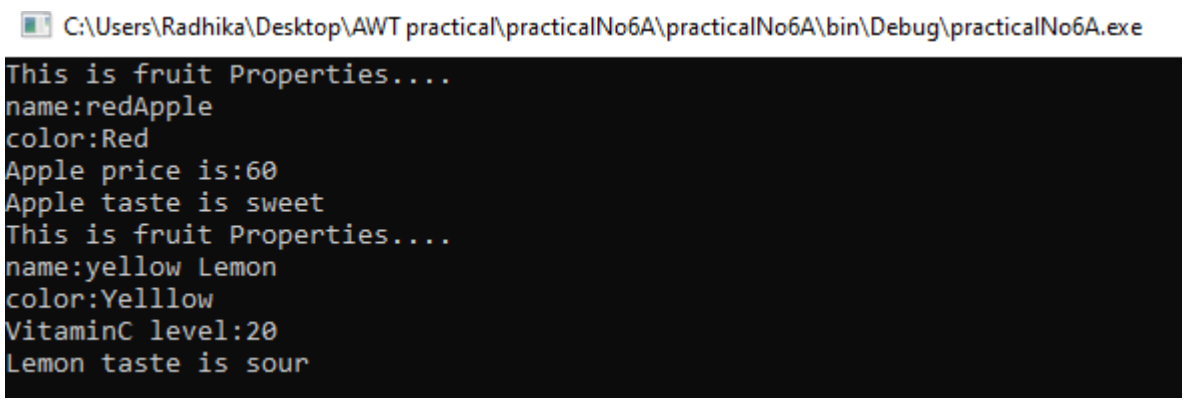



Source Code:

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

namespace practicalNo6A
{
    class Program
    {
        class fruit
        {
            public string name ;
            public string color;

            public void printDetail()
            {
                Console.WriteLine("This is fruit Properties ... ");
                Console.WriteLine("name:" + name);
                Console.WriteLine("color:" + color);
```

```
        }
    }
    class Apple:fruit
    {
        public int price;

        public void taste()
        {
            Console.WriteLine("Apple price is:" + price);
            Console.WriteLine("Apple taste is sweet");
        }
    }
    class Lemon :fruit
    {
        public float vitaminC;


        public void taste()
        {
            base.printDetail();
            Console.WriteLine("VitaminC level:"+vitaminC);
            Console.WriteLine("Lemon taste is sour");
        }
    }
    static void Main(string[] args)
    {
        Apple a = new Apple();
        a.name = "redApple";
        a.color = "Red";
        a.price = 60;
        a.printDetail();
        a.taste();

        Lemon l = new Lemon();
        l.name = "yellow Lemon";
        l.color = "Yelllow";
        l.vitaminC = 20 ;
        l.taste();
        Console.ReadLine();
    }
}
}
```

Output:



| **Practical No:6 B** | | **Date:**12/05/2022 |
|---|---|---|
| **Aim:** | Design Applications using Inheritance and Abstract Classes | |

Write a program to create abstract class account which has abstract
method deposit and withdraw. Create concrete
class Saving Account and Current Account which is inherited from abstract class. Show the polymorphism, method overriding into it.

a. Saving Account initial deposit is 1000 INR
b. Current Account initial deposit is 3000 INR

When account is open the initial deposit will automatically initialize. Create menu in Program.cs where user will create account, deposit and withdraw money from the options.
Withdraw amount will not cross minimum deposit amount. If user try to do withdraw amount
which is more than minimum deposit the appropriate message will display. There is one more
method you can create that will display the balance amount of account holder. (Name of method is display_Balance());

## Source Code:

*#Current_Account.cs*

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

namespace Practical_6B
{
    class Current_Account : Account
    {
        private decimal balance,minBalance;
        public Current_Account()
        {
            balance = 3000;
            minBalance = 3000;
        }
        public override void Deposit(decimal amount)
        {
            balance = balance + amount;
            DisplayBalance();
        }

        public override void Withdraw(decimal amount)
        {
            if (balance > amount && (balance-amount == minBalance || balance-
amount > minBalance))
            {
                balance = balance - amount;
                DisplayBalance();
            }
            else if(minBalance == amount)
            {
                DisplayBalance();
                Console.WriteLine("Oops!! Withdraw Failed!!!");
```

```
                Console.WriteLine("Sorry! Your Current balance is equal to the
amount required to keep account activate, so can't withdraw amount.");

            }
            else if(minBalance > amount)
            {
                DisplayBalance();
                Console.WriteLine("Oops!! Withdraw Failed!!!");
                Console.WriteLine("Sorry! Your Current balance is equal to the
amount required to keep account activate, so can't withdraw amount.");

            }
            else
            {
                Console.WriteLine("Insufficient Balance");
                DisplayBalance();
            }
        }

        public override void DisplayBalance()
        {
            Console.WriteLine("Your Balance is :" + balance);
        }
    }
}
```

*#Saving_Account*

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

namespace Practical_6B
{
    class Savings_Account : Account
    {
        private decimal balance,minBalance;

        public Savings_Account()
        {
            balance = 1000;
            minBalance = 1000;
        }

        public override void Deposit(decimal amount)
        {
            balance = balance + amount;
            DisplayBalance();
        }


        public override void Withdraw(decimal amount)
        {
            if(balance > amount && minBalance < amount)
            {
                balance = balance - amount;
                DisplayBalance();
            }
            else if(minBalance == amount)
            {
```

```
                DisplayBalance();
                Console.WriteLine("Oops!! Withdraw Failed!!!");
                Console.WriteLine("Sorry! Your Current balance is equal to the
amount required to keep account activate, so can't withdraw amount.");

            }
            else if(minBalance > amount)
            {
                DisplayBalance();
                Console.WriteLine("Oops!! Withdraw Failed!!!");
                Console.WriteLine("Sorry! Your Current balance is equal to the
amount required to keep account activate, so can't withdraw amount.");

            }
            else
            {
                Console.WriteLine("Insufficient Balance");
                DisplayBalance();
            }
        }

        public override void DisplayBalance()
        {
            Console.WriteLine("Your Current Balance is :" +balance);
        }
    }
}
```

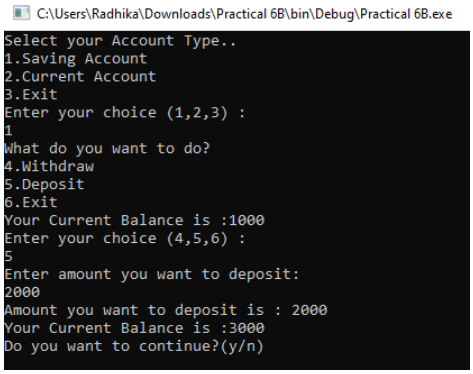
*#Program.cs*

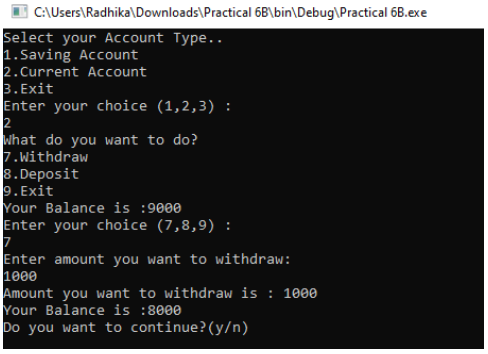
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Practical_6B
{
    class Program
    {
        static Account s = new Savings_Account();
        static Account c = new Current_Account();
        static void Main(string[] args)
        {
            String ans = "Y";
            do
            {
                Console.Clear();
                int choice, ch,withdrawAmount,depositAmount;
                Console.WriteLine("Select your Account Type..");
                Console.WriteLine("1.Saving Account");
                Console.WriteLine("2.Current Account");
                Console.WriteLine("3.Exit");
                Console.WriteLine("Enter your choice (1,2,3) :");
                choice = Convert.ToInt32(Console.ReadLine());

                switch (choice)
                {
                    case 1:
                        Console.WriteLine("What do you want to do?");
                        Console.WriteLine("4.Withdraw");
                        Console.WriteLine("5.Deposit");
                        Console.WriteLine("6.Exit");
                        s.DisplayBalance();
```

```
                        Console.WriteLine("Enter your choice (4,5,6) :");
                        ch = Convert.ToInt32(Console.ReadLine());

                        switch (ch)
                        {

                            case 4:
                                Console.WriteLine("Enter amount you want to
withdraw: ");
                                withdrawAmount =
Convert.ToInt32(Console.ReadLine());
                                Console.WriteLine("Amount you want to withdraw
is : " + withdrawAmount);

                                s.Withdraw(withdrawAmount);
                                break;
                            case 5:
                                Console.WriteLine("Enter amount you want to
deposit: ");
                                depositAmount =
Convert.ToInt32(Console.ReadLine());
                                Console.WriteLine("Amount you want to deposit
is : " + depositAmount);

                                s.Deposit(depositAmount);
                                break;
                            case 6:
                                break;
                            default:
                                Console.WriteLine("Wrong Choice...");
                                break;

                        }

                        break;


                    case 2:
                        Console.WriteLine("What do you want to do?");
                        Console.WriteLine("7.Withdraw");
                        Console.WriteLine("8.Deposit");
                        Console.WriteLine("9.Exit");
                        c.DisplayBalance();
                        Console.WriteLine("Enter your choice (7,8,9) :");
                        ch = Convert.ToInt32(Console.ReadLine());
                        switch (ch)
                        {

                            case 7:
                                Console.WriteLine("Enter amount you want to
withdraw: ");
                                withdrawAmount =
Convert.ToInt32(Console.ReadLine());
                                Console.WriteLine("Amount you want to withdraw
is : " + withdrawAmount);

                                c.Withdraw(withdrawAmount);
                                break;
                            case 8:
                                Console.WriteLine("Enter amount you want to
deposit: ");
                                depositAmount =
Convert.ToInt32(Console.ReadLine());
                                Console.WriteLine("Amount you want to deposit
is : " + depositAmount);

                                c.Deposit(depositAmount);
                                break;
                            case 9:
                                break;
```

```
                            default:
                                Console.WriteLine("Wrong Choice...");
                                break;

                        }
                        break;
                    case 3:
                        break;
                    default:
                        Console.WriteLine("Wrong Choice...");
                        break;
                }
                Console.Write("Do you want to continue?(y/n)");
                ans = Console.ReadLine();
            } while (ans.Equals("Y")||ans.Equals("y"));
        }
    }
}
```

## Output:



| **Practical No: 7 A** | | **Date:**14/05/2022 |
|---|---|---|
| Aim: | To Design a Web Application for an Organization with Registration forms and advanced controls | |

a. Create login form by using ASP.NET web form. Show appropriate message for valid and invalid user.

## Source Code:

#Default.aspx

```
    <%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="prac7._Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
    <h1>Practical - 7A: Design a login form</h1>
    <table style="width: 100%;">
        <tr>
            <td class="modal-sm" colspan="3" style="height: 38px">
                <asp:Literal ID="ltrMessage" runat="server"></asp:Literal>
            </td>
        </tr>
        <tr>
            <td class="modal-sm" style="width: 140px; height: 38px">
                <asp:Label ID="Label1" runat="server" Text="Username"></asp:Label>
            </td>
            <td style="width: 265px; height: 38px">
                <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox>
            </td>
            <td style="height: 38px"></td>
        </tr>
        <tr>
            <td class="modal-sm" style="width: 140px; height: 44px">
                <asp:Label ID="Label2" runat="server" Text="Password"></asp:Label>
            </td>
            <td style="width: 265px; height: 44px">
                <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox>
            </td>
            <td style="height: 44px"></td>
        </tr>
        <tr>
            <td class="modal-sm" colspan="3" style="height: 42px">
                <asp:Button ID="btnLogin" runat="server" OnClick="btnLogin_Click"
Text="Login" />
            </td>
        </tr>
    </table>
</asp:Content>
```

#Default.aspx.cs

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

namespace prac7
{
    public partial class _Default : Page
```

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

        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            if (txtUsername.Text.Equals("Admin") && txtPassword.Text.Equals("Admin"))
            {
                //ltrMessage.Text = "Valid User";
                Response.Redirect("success.aspx");
            }
            else
            {
                //ltrMessage.Text = "Invalid User";
                Response.Redirect("unsuccess.aspx");
            }
        }
    }
}
```

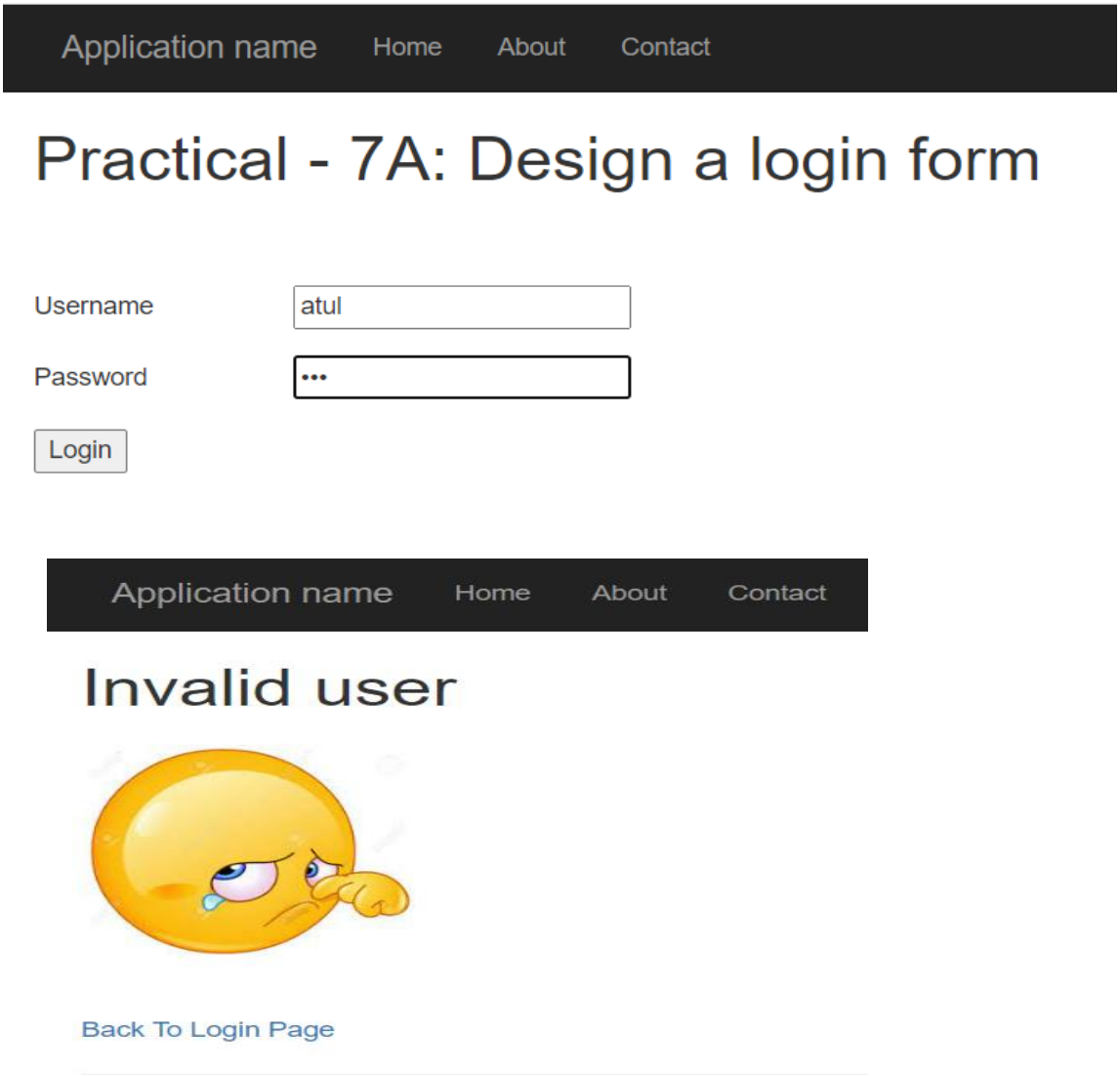
#success.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="success.aspx.cs" Inherits="prac7.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h1>It Is A Valid User</h1>
    <h1>
        <asp:Image ID="Image1" runat="server" Height="104px"
ImageUrl="~/image/success.png" Width="170px" />
    </h1>
    <p> </p>
    <p>
        <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Default.aspx">Back To Login Page</asp:HyperLink>
    </p>
</asp:Content>
```

#unsuccessful.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="unsuccess.aspx.cs" Inherits="prac7.WebForm2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <h1>Invalid user</h1>
    <h1>
        <asp:Image ID="Image1" runat="server" Height="147px"
ImageUrl="~/image/unsuccess.jpg" Width="163px" />
    </h1>
    <p> </p>
    <p>
        <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Default.aspx">Back To Login Page</asp:HyperLink>
    </p>
</asp:Content>
```

Output :


Application name
Home
About
Contact
Practical - 7A: Design a login form
Username
atul
Password
Login
Application name
Home
About
Contact
Invalid user
Back To Login Page

| **Practical No: 7 B** | | **Date:**14/05/2022 |
|---|---|---|
| Aim: | To Design a Web Application for an Organization with Registration forms and advanced controls | |

b. Create registration form using ASP.NET web form. Which has following input

- Username
- Password
- Confirm password
- Email
- Date of Birth
- Image upload

Write appropriate validation messages [such as invalid email, required field , invalid password, image format and size etc] Submit form will place the data in same page by using crosspost and postback method And link of this page to your login form as sign up activity

**Source Code:**

**Default.aspx**

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="prac7b._Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

    <div class="jumbotron">
        <h1>Practical 7B - Registration Form</h1>
            <table style="width:100%;">
                <tr>
                    <td style="width: 318px; height: 45px">
                        <asp:Label ID="Label1" runat="server"
Text="Username"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 45px">
                        <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox>
                    </td>
                    <td style="height: 45px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 51px">
                        <asp:Label ID="Label2" runat="server"
Text="Password"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 51px">
                        <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox>
                    </td>
                    <td style="height: 51px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 48px">
                        <asp:Label ID="Label3" runat="server" Text="Confirm
Password"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 48px">
```

```
                        <asp:TextBox ID="txtCfPassword" runat="server"
TextMode="Password"></asp:TextBox>
                    </td>
                    <td style="height: 48px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 56px">
                        <asp:Label ID="Label4" runat="server"
Text="Email"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 56px">
                        <asp:TextBox ID="txtEmail" runat="server"
TextMode="Email"></asp:TextBox>
                    </td>
                    <td style="height: 56px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 52px">
                        <asp:Label ID="Label5" runat="server" Text="Date of
Birth"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 52px">
                        <asp:TextBox ID="txtDoB" runat="server"
TextMode="Date"></asp:TextBox>
                    </td>
                    <td style="height: 52px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 60px">
                        <asp:Label ID="Label6" runat="server"
Text="Image"></asp:Label>
                    </td>
                    <td style="width: 473px; height: 60px">
                        <asp:TextBox ID="txtImage" runat="server"></asp:TextBox>
                    </td>
                    <td style="height: 60px"></td>
                </tr>
                <tr>
                    <td style="width: 318px; height: 57px">
                        <asp:Button ID="Button1" runat="server" Text="Registration"
OnClick="Button1_Click" />
                    </td>
                    <td style="height: 57px; width: 473px"> </td>
                    <td style="height: 57px"> </td>
                </tr>
            </table>
    </div>


</asp:Content>
```

## Default.aspx.cs

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

namespace prac7b
```

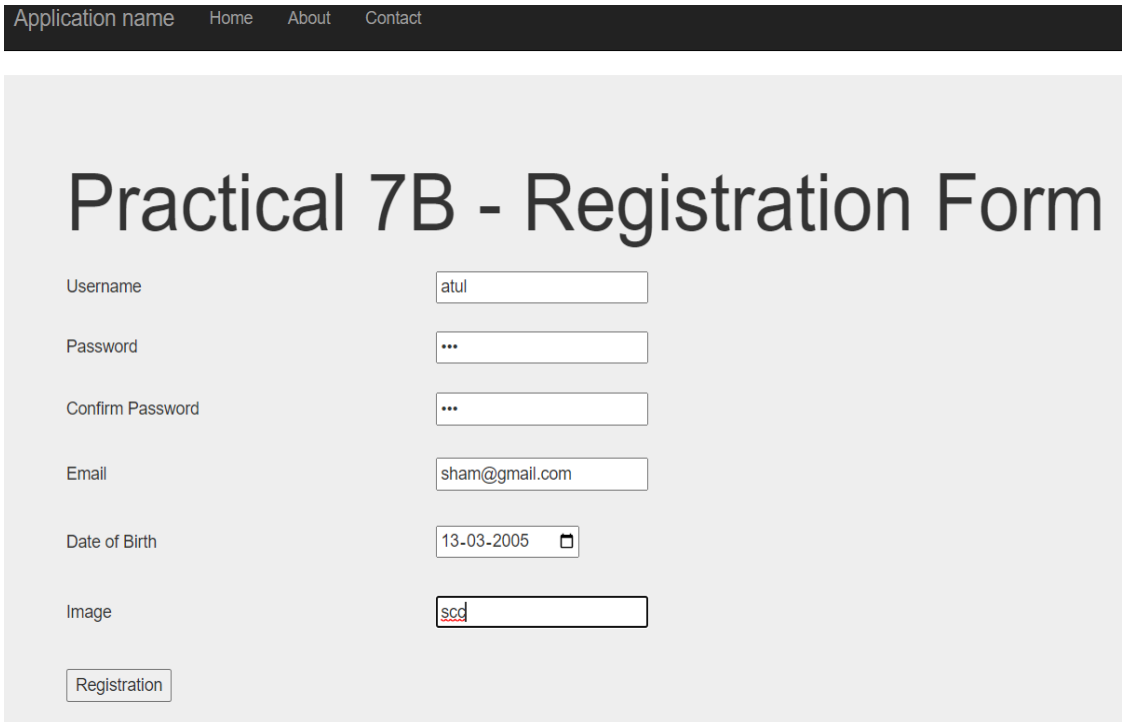
```
{
    public partial class _Default : Page
    {
        protected void Page_Load(object sender, EventArgs e)//what are the 2
parameters and what are the application
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {

Response.Redirect("output.aspx?txtUsername="+txtUsername.Text+"&txtPassword="+txtPassw
ord.Text+"&txtEmail="+txtEmail.Text+"&txtDoB="+txtDoB.Text+"&txtImage="+txtImage.Text)
;
        }
    }
}
```

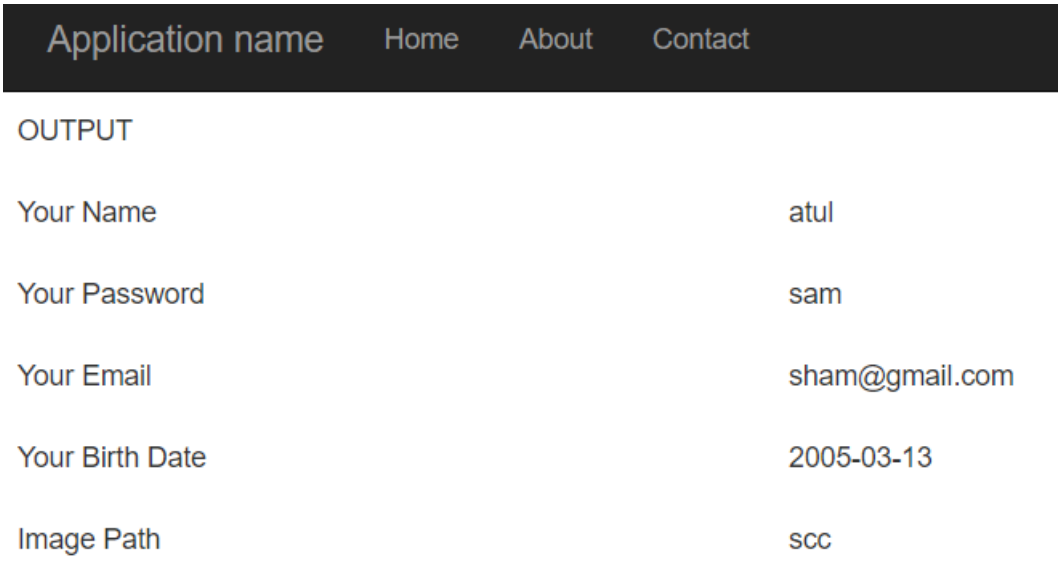
## Output.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="output.aspx.cs" Inherits="prac7b.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <table style="width: 100%; height: 245px;">
        <tr>
            <td colspan="3">OUTPUT</td>
        </tr>
        <tr>
            <td style="width: 385px">Your Name</td>
            <td style="width: 426px">
                <asp:Label ID="lblusername" runat="server"></asp:Label>
            </td>
            <td> </td>
        </tr>
        <tr>
            <td style="width: 385px">Your Password</td>
            <td style="width: 426px">
                <asp:Label ID="lblpassword" runat="server"></asp:Label>
            </td>
            <td> </td>
        </tr>
        <tr>
            <td style="width: 385px">Your Email</td>
            <td style="width: 426px">
                <asp:Label ID="lblemail" runat="server"></asp:Label>
            </td>
            <td> </td>
        </tr>
        <tr>
            <td style="width: 385px">Your Birth Date</td>
            <td style="width: 426px">
                <asp:Label ID="lbldob" runat="server"></asp:Label>
            </td>
            <td> </td>
        </tr>
        <tr>
            <td style="width: 385px">Image Path</td>
            <td style="width: 426px">
                <asp:Label ID="lblimg" runat="server"></asp:Label>
            </td>
            <td> </td>
        </tr>
    </table>
```

```
</asp:Content>
```

## Output



| **Practical No: 8** | | **Date:**02/06/2022 |
|---|---|---|
| Aim: | Create website using master page concept. | |

Create web application layout by using master page which has following pages

- Home
- About us
- Contact us
- Login

Create header and footer in master page and call it in each page. Write appropriate dummy content in each page.
[Refer following screen shot for Master page]

## Source Code:

## Default.aspx

```
      <%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="Practical_8._Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

    <div class="jumbotron">
        <h1>ASP.NET Practical-8 </h1>
        <p class="lead">ASP.NET Practicals </p>

    </div>

    <div class="row">
        <div class="col-md-4">
           <h1>Core-ASP.NET</h1>
            <a href="#">Class & Object</a> <br />
            <a href="#">Inheritance -Abstract Class</a> <br />
            <a href="#">Use of Window Form</a> <br />
            <a href="#">Use of file handling</a> <br />
            <a href="#">Use of C-Sharp controls</a> <br />
            <a href="#">Polymorphism & Abstract methods</a> <br />
            <a href="#">Registration Form-Asp net controls </a> <br />
        </div>
        <div class="col-md-4">
            <div>
               <h1>ADO.NET</h1>
                <a href="#">Searching</a> <br />
                <a href="#">Inserting/Updating/Deleting</a> <br />
                <a href="#">Use of store procedure</a> <br />
            </div>
            <div>
               <h1>Web Services</h1>
                <a href="#">Produce & consume web service</a> <br />
                <a href="#">Session Management</a> <br />

            </div>
        </div>
        <div class="col-md-4">
             <div>
               <h1>MVC</h1>
                 <a href="#">Create View</a> <br />
                 <a href="#">Update View</a> <br />
                 <a href="#">Detail View</a> <br />
```

```
                <a href="#">Delete View</a> <br />
            </div>
            <div>
                <h1>WCF Services</h1>
                <a href="#">Produce and Consume Web services</a> <br />
                <a href="#">Practical 2</a> <br />
            </div>
        </div>
    </div>

</asp:Content>
```

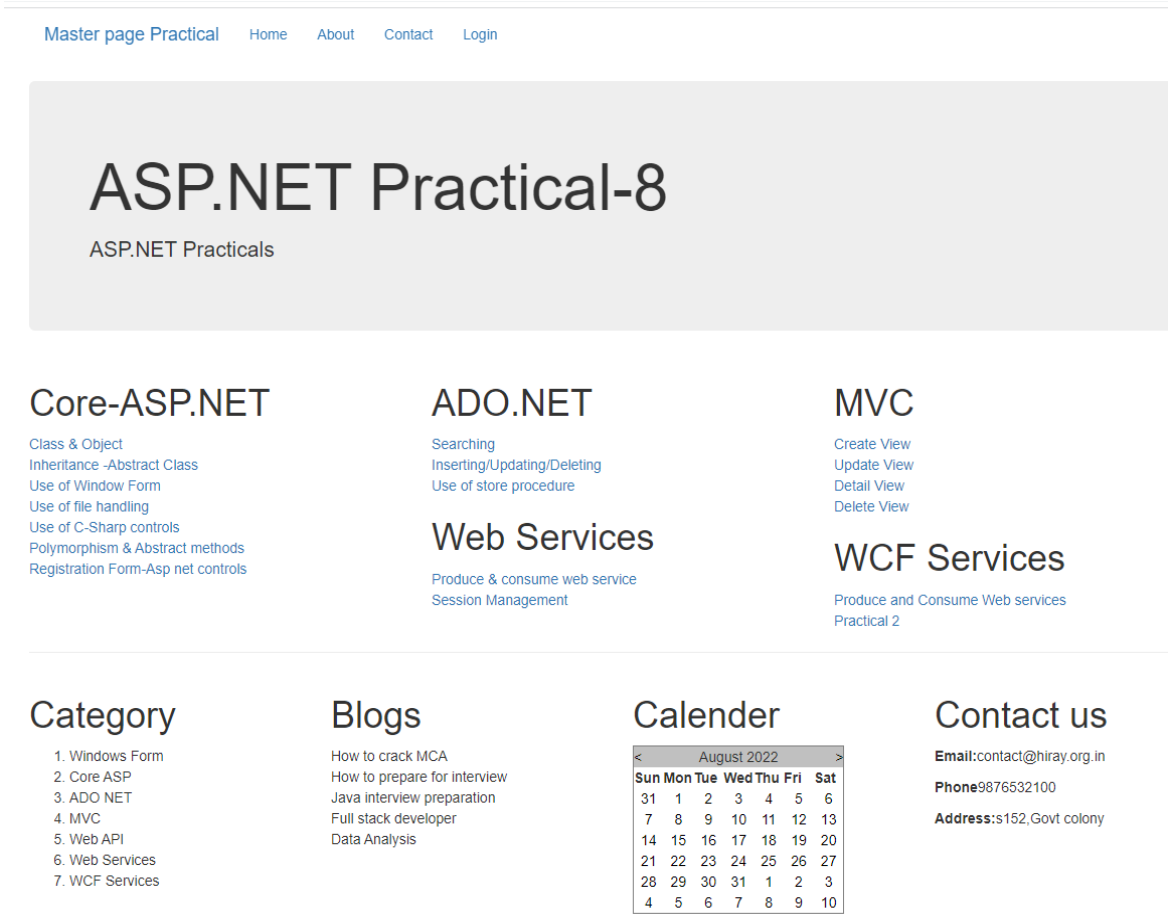
**Site Master**

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="Practical_8.SiteMaster" %>

<!DOCTYPE html>

<html lang="en">
<head runat="server">
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title><%: Page.Title %> - My ASP.NET Application</title>

    <asp:PlaceHolder runat="server">
        <%: Scripts.Render("~/bundles/modernizr") %>
    </asp:PlaceHolder>

    <webopt:bundlereference runat="server" path="~/Content/css" />
    <link href="~/favicon.ico" rel="shortcut icon" type="image/x-icon" />

</head>
<body>
    <form runat="server">
        <asp:ScriptManager runat="server">
            <Scripts>
                <%--To learn more about bundling scripts in ScriptManager see
https://go.microsoft.com/fwlink/?LinkID=301884 --%>
                <%--Framework Scripts--%>
                <asp:ScriptReference Name="MsAjaxBundle" />
                <asp:ScriptReference Name="jquery" />
                <asp:ScriptReference Name="bootstrap" />
                <asp:ScriptReference Name="WebForms.js" Assembly="System.Web"
Path="~/Scripts/WebForms/WebForms.js" />
                <asp:ScriptReference Name="WebUIValidation.js" Assembly="System.Web"
Path="~/Scripts/WebForms/WebUIValidation.js" />
                <asp:ScriptReference Name="MenuStandards.js" Assembly="System.Web"
Path="~/Scripts/WebForms/MenuStandards.js" />
                <asp:ScriptReference Name="GridView.js" Assembly="System.Web"
Path="~/Scripts/WebForms/GridView.js" />
                <asp:ScriptReference Name="DetailsView.js" Assembly="System.Web"
Path="~/Scripts/WebForms/DetailsView.js" />
                <asp:ScriptReference Name="TreeView.js" Assembly="System.Web"
Path="~/Scripts/WebForms/TreeView.js" />
                <asp:ScriptReference Name="WebParts.js" Assembly="System.Web"
Path="~/Scripts/WebForms/WebParts.js" />
                <asp:ScriptReference Name="Focus.js" Assembly="System.Web"
Path="~/Scripts/WebForms/Focus.js" />
                <asp:ScriptReference Name="WebFormsBundle" />
                <%--Site Scripts--%>
            </Scripts>
```

```
        </asp:ScriptManager>

        <div class="navbar navbar-fixed-top">
            <div class="container">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse" title="more options">
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                    </button>
                    <a class="navbar-brand" runat="server" href="~/">Master page
Practical</a>
                </div>
                <div class="navbar-collapse collapse">
                    <ul class="nav navbar-nav">
                        <li><a runat="server" href="~/">Home</a></li>
                        <li><a runat="server" href="~/About">About</a></li>
                        <li><a runat="server" href="~/Contact">Contact</a></li>
                        <li><a runat="server" href="#">Login</a></li>
                    </ul>
                </div>
            </div>
        </div>
        <div class="container body-content">
            <asp:ContentPlaceHolder ID="MainContent" runat="server">
            </asp:ContentPlaceHolder>
            <hr />
            <footer>
               <div class="row">
                    <div class="col-md-3">
                            <h1>Category</h1>
                            <ol>
                                <li>Windows Form</li>
                                <li>Core ASP</li>
                                <li>ADO NET</li>
                                <li>MVC</li>
                                <li>Web API</li>
                                <li>Web Services</li>
                                <li>WCF Services</li>
                            </ol>
                    </div>
                   <div class="col-md-3">
                            <h1>Blogs</h1>
                            <a href="#"></a>How to crack MCA<br />
                            <a href="#"></a>How to prepare for interview<br />
                            <a href="#"></a>Java interview preparation<br />
                            <a href="#"></a>Full stack developer<br />
                            <a href="#"></a>Data Analysis<br />
                    </div>
                   <div class="col-md-3">
                            <h1>Calender</h1>
                       <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
                    </div>
                   <div class="col-md-3">
                            <h1>Contact us</h1>
                            <p><b>Email:</b>contact@hiray.org.in</p>
                            <p><b>Phone</b>9876532100</p>
                            <p><b>Address:</b>s152,Govt colony</p>
                    </div>
               </div>
                <p>&copy; <%: DateTime.Now.Year %> - My ASP.NET Application</p>
```

```
            </footer>
        </div>

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

**#Output**



| **Practical No: 10** | | **Date:**09/06/2022 |
|---|---|---|
| Aim: | Design a webpage to demonstrate a connection oriented architecture.. | |

Create registration form using ASP.NET web form. Which has following input

- Username
- Password
- Confirm password
- Email
- Date of Birth
- Image upload

Write appropriate validation messages [such as invalid email, required field , invalid password, image
format and size etc] Submit form will place the data in the MS-SQL server

## Source Code:

### Default.aspx

```
      <%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="Practical10._Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">
<h1> Practical 10 - Understand the use of ADO.NET control</h1>
    <table class="table" style="width: 100%;">
        <tr>
            <td colspan="3">
                <asp:Literal ID="Literal1" runat="server"></asp:Literal></td>
        </tr>
        <tr>
            <td colspan="3" align="center"><h2>Registration-Form</h2></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label1" runat="server"
Text="Username"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Username is required"
ControlToValidate="txtUsername"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label2" runat="server"
Text="Password"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Password is required"
ControlToValidate="txtPassword"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td>
```

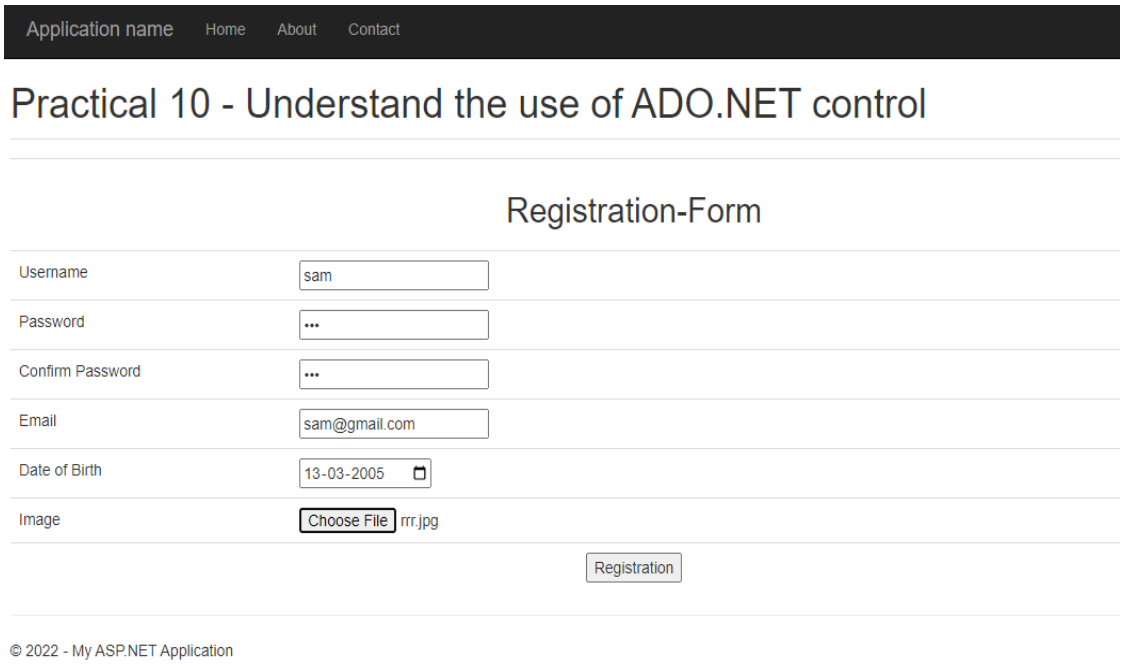
```
                <asp:Label ID="Label3" runat="server" Text="Confirm
Password"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtConfirm" runat="server"
TextMode="Password"></asp:TextBox></td>
            <td>
                <asp:CompareValidator ID="CompareValidator1" runat="server"
ErrorMessage="Password doesn't match" ControlToCompare="txtPassword"
ControlToValidate="txtConfirm"></asp:CompareValidator></td>
        </tr>
          <tr>
            <td>
                <asp:Label ID="Label4" runat="server" Text="Email"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtEmail" runat="server"
TextMode="SingleLine"></asp:TextBox></td>
            <td>
                <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ErrorMessage="Invalid Email" ControlToValidate="txtEmail"
ValidationExpression="^([\w-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([\w-
]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$"></asp:RegularExpressionValidator></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label5" runat="server" Text="Date of
Birth"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtDOB" runat="server" TextMode="Date"></asp:TextBox>
            </td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ErrorMessage="Birthdate is required"
ControlToValidate="txtDOB"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label6" runat="server" Text="Image"></asp:Label></td>
            <td>
                <asp:FileUpload ID="FileUpload1" runat="server" /></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Image is required" ValidateRequestMode="Enabled"
ControlToValidate="FileUpload1"></asp:RequiredFieldValidator></td>
        </tr>
         <tr>
            <td colspan="3" align="center">
                <asp:Button ID="Button1" runat="server" Text="Registration"
OnClick="Button1_Click" /></td>
         </tr>
    </table>
</asp:Content>
```

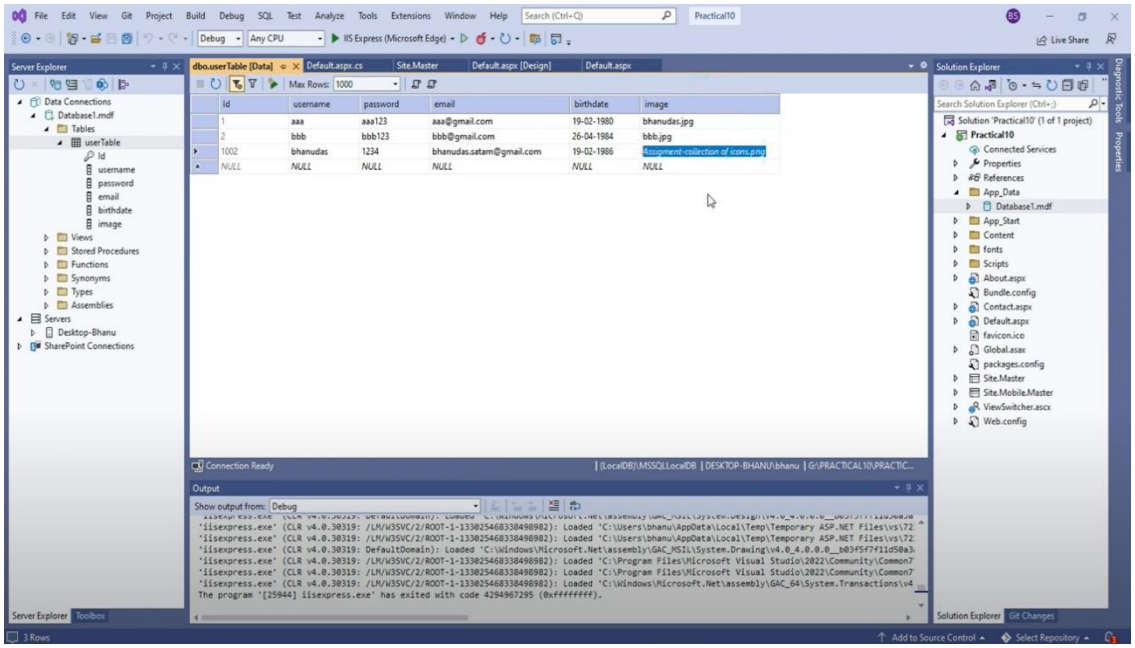
**Default.aspx.cs**

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

```
namespace Practical10
{
    public partial class _Default : Page
    {
        //Step 1 Declaration of Connection object,Command object, Dataadapter object
        SqlConnection conn;
        SqlDataAdapter adapter;
        SqlCommand cmd;

        protected void Page_Load(object sender, EventArgs e)
        {


        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            //step 1 Initializing the connection object
            conn = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=G:\\Practical10\\Practical10\\App_Data
\\Database1.mdf;Integrated Security=True");
            conn.Open();
            //step 2 Initializing the command object
            String insertquery = "insert into
usertable(username,password,email,birthdate,image)values(@uname,@pass,@email,@dob,@ima
ge)";
            cmd = new SqlCommand(insertquery,conn);
            //step 3 Initializing the DataAdapter
            adapter = new SqlDataAdapter(insertquery, conn);
            //step 4 Adding the parameters to query
            //adding the value of Username,password,email,dob,imagename
            adapter.InsertCommand = cmd;
            adapter.InsertCommand.Parameters.AddWithValue("@uname", txtUsername.Text);
            adapter.InsertCommand.Parameters.AddWithValue("@pass", txtPassword.Text);
            adapter.InsertCommand.Parameters.AddWithValue("@email", txtEmail.Text);
            adapter.InsertCommand.Parameters.AddWithValue("@dob", txtDOB.Text);
            adapter.InsertCommand.Parameters.AddWithValue("@image",
FileUpload1.FileName);
            adapter.InsertCommand.ExecuteNonQuery();
            Literal1.Text = "Record inserted successfully";
            conn.Close();
        }
    }
 }
```

## Output


Application name
Home
About
Contact
Practical 10 - Understand the use of ADO.NET control
Registration-Form
Username
sam
Password
Confirm Password
Email
sam@gmail.com
Date of Birth
13-03-2005
Image
Choose File
rrr.jpg
Registration
© 2022 - My ASP.NET Application



| **Practical No: 12** | | **Date:**16/06/2022 |
|---|---|---|
| Aim: | Create a webpage that demonstrates the use of data bound controls of ASP.NET. | |

Which has following structure

- Empid
- EmpName
- EmpPost
- EmpSal
- Dept

Create Add/Modify/Delete/Search functionality using ADO.NET controls

**Source Code:**

**Default.aspx**

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="Practical12_CRUD_operations._Default" %>

<asp:Content ID="BodyContent" ContentPlaceHolderID="MainContent" runat="server">

    <table class="table" style="width: 100%;">
        <tr>
            <th colspan="3" align="center"><h2>Employee Data Entry</h2></th>
        </tr>
        <tr>
            <td colspan="3">
                <asp:Literal ID="Literal1" runat="server"></asp:Literal></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label5" runat="server" Text="Employee
id"></asp:Label></td>
            <td colspan="2">
                <asp:DropDownList ID="DropDownList1" runat="server"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"
AutoPostBack="True"></asp:DropDownList></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label1" runat="server" Text="Employee
Name"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtEmployeeName" runat="server"
AutoPostBack="True"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Empname is required"
ControlToValidate="txtEmployeeName"></asp:RequiredFieldValidator></td>

        </tr>
        <tr>
            <td>
                <asp:Label ID="Label2" runat="server" Text="Employee
Post"></asp:Label></td>
            <td>
```

```
                <asp:TextBox ID="txtEmployeePost" runat="server"
AutoPostBack="True"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Employee Post is required"
ControlToValidate="txtEmployeePost"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label3" runat="server" Text="Employee
Salary"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtEmployeeSal" runat="server"
AutoPostBack="True"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ErrorMessage="Employee Salary is required"
ControlToValidate="txtEmployeeSal"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label4" runat="server"
Text="Department"></asp:Label></td>
            <td>
                <asp:TextBox ID="txtEmployeeDept" runat="server"
AutoPostBack="True"></asp:TextBox></td>
            <td>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Department is required"
ControlToValidate="txtEmployeeDept"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td colspan="3">
                <asp:Button ID="Button1" runat="server" Text="Insert"
OnClick="Button1_Click" /> 
                <asp:Button ID="Button2" runat="server" Text="Update"
OnClick="Button2_Click" /> 
                <asp:Button ID="Button3" runat="server" Text="Delete"
OnClick="Button3_Click" /> 
                <asp:Button ID="Button4" runat="server" Text="Detail" /> 

        </tr>
    </table>


</asp:Content>
```

## Default.aspx.cs

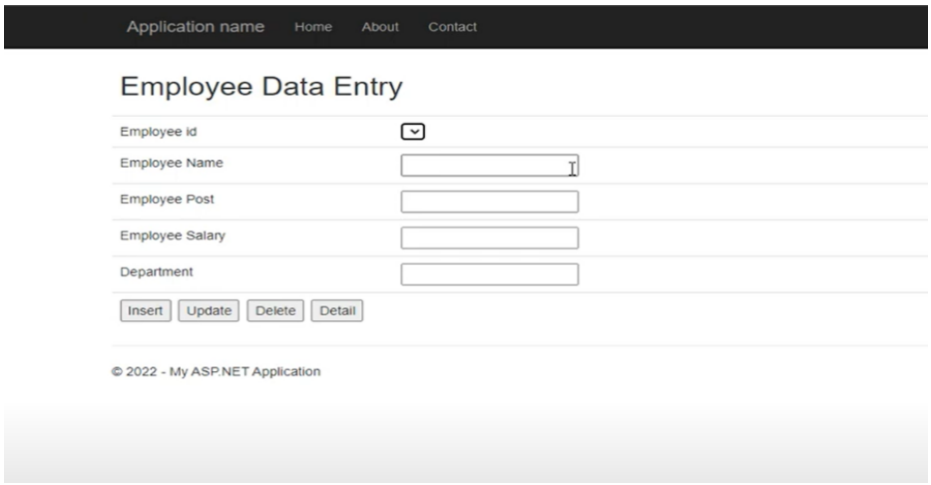
```
       using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Practical12_CRUD_operations
{
    public partial class _Default : Page
    {
```

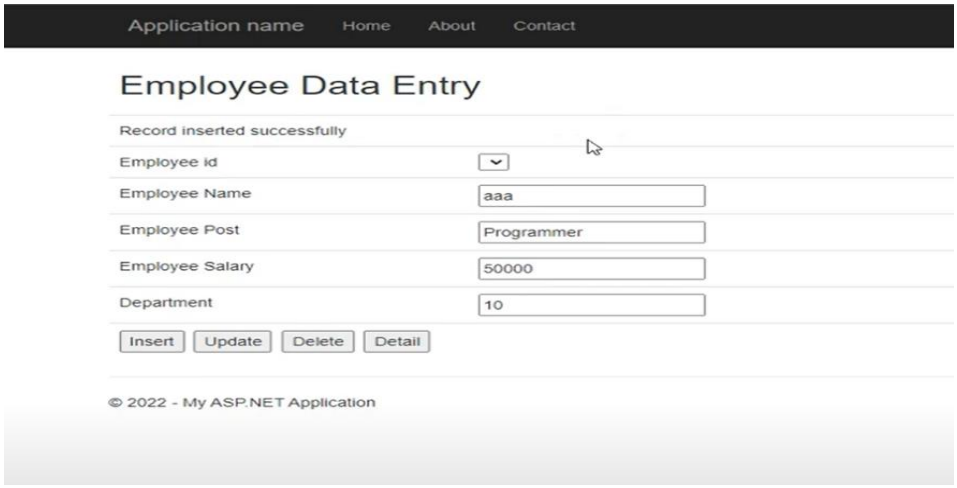
```
        String constr;
        SqlConnection conn;
        SqlDataAdapter adapter;
        DataSet ds;
        protected void Page_Load(object sender, EventArgs e)
        {
            //Code for inserting record
            if(!IsPostBack)
            {
                constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integra
ted Security=True";
                conn = new SqlConnection(constr);
                adapter = new SqlDataAdapter("select * from employee", conn);
                ds = new DataSet();
                adapter.Fill(ds, "tempEmployee");
                for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables[0].Rows[i].ItemArray[0].ToString());
                }
                txtEmployeeName.Text = ds.Tables[0].Rows[0].ItemArray[1].ToString();
                txtEmployeePost.Text = ds.Tables[0].Rows[0].ItemArray[2].ToString();
                txtEmployeeSal.Text = ds.Tables[0].Rows[0].ItemArray[3].ToString();
                txtEmployeeDept.Text = ds.Tables[0].Rows[0].ItemArray[4].ToString();
            }


        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            //Code for inserting record
                constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integra
ted Security=True";
            //step 1  Create connection
                conn = new SqlConnection(constr);
            //step 2  Create Data Adapter
                adapter = new SqlDataAdapter("select * from employee", conn);
                SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(adapter);
            //step 3  Create Data Set object
                ds = new DataSet();
            //step 4 Filling Query result  result into data set
                adapter.Fill(ds, "tempEmployee");
            //step 5  Create a new Row
            DataRow myrow = ds.Tables["tempEmployee"].NewRow();
            //step 6 Adding the textbox content to respective row fields
            myrow["empname"] = txtEmployeeName.Text;
            myrow["emppost"] = txtEmployeePost.Text;
            myrow["empsal"] = Convert.ToInt32(txtEmployeeSal.Text);
            myrow["deptid"] = Convert.ToInt32(txtEmployeeDept.Text);

            //step 7 Adding the row to data set
            ds.Tables["tempEmployee"].Rows.Add(myrow);

            //step 8 updating database using data adapter

            adapter.Update(ds, "tempEmployee");
            Literal1.Text = "Record inserted successfully";
```

```
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            //code for updating record

            constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integra
ted Security=True";
            //step 1  Create connection
            conn = new SqlConnection(constr);
            int id = Convert.ToInt16(DropDownList1.SelectedValue);
            //step 2  Create Data Adapter
            adapter = new SqlDataAdapter("select * from employee where empid="+id,
conn);
            SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(adapter);
            //step 3  Create Data Set object
            ds = new DataSet();
            //step 4 Filling Query result  result into data set
            adapter.Fill(ds, "tempEmployee");
            //step 5 Create DataRow object
            DataRow dr = ds.Tables["tempEmployee"].Rows[0];
            dr["empname"] = txtEmployeeName.Text;
            dr["emppost"] = txtEmployeePost.Text;
            dr["empsal"] = txtEmployeeSal.Text;
            dr["deptid"] = txtEmployeeDept.Text;
            //step 6 update the record to database
            adapter.Update(ds, "tempEmployee");
            Literal1.Text = "Record updated successfully";
        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {
            constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integra
ted Security=True";
            conn = new SqlConnection(constr);
            String id = DropDownList1.SelectedValue;
            adapter = new SqlDataAdapter("select * from employee where empid="+id,
conn);
            ds = new DataSet();
            adapter.Fill(ds, "tempEmployee");
            txtEmployeeName.Text = ds.Tables[0].Rows[0].ItemArray[1].ToString();
            txtEmployeePost.Text= ds.Tables[0].Rows[0].ItemArray[2].ToString();
            txtEmployeeSal.Text= ds.Tables[0].Rows[0].ItemArray[3].ToString();
            txtEmployeeDept.Text= ds.Tables[0].Rows[0].ItemArray[4].ToString();
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            //code for deleting record

            constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integra
ted Security=True";
            //step 1  Create connection
            conn = new SqlConnection(constr);
            int id = Convert.ToInt16(DropDownList1.SelectedValue);
            //step 2  Create Data Adapter
            adapter = new SqlDataAdapter("select * from employee where empid=" + id,
conn);
```

```
            SqlCommandBuilder sqlCommandBuilder = new SqlCommandBuilder(adapter);
            //step 3  Create Data Set object
            ds = new DataSet();
            //step 4 Filling Query result  result into data set
            adapter.Fill(ds, "tempEmployee");
            //step 5 deleting row from dataset
            ds.Tables["tempEmployee"].Rows[0].Delete();
            //step 6 updating the database
            adapter.Update(ds, "tempEmployee");
            Literal1.Text = "Record Deleted successfully";
        }
    }
}
```

## Output :



| **Practical No.: 13** | | **Date: 21/06/2022** |
|---|---|---|
| Aim: | Design a webpage to demonstrate the working of a simple stored procedure. | |

Create CRUD web Application by using following Model [Class Employee]
Which has following structure

- Empid
- EmpName
- EmpPost
- EmpSal
- Dept

Create Add/Modify/Delete/Search functionality using store procedure

**Source:**

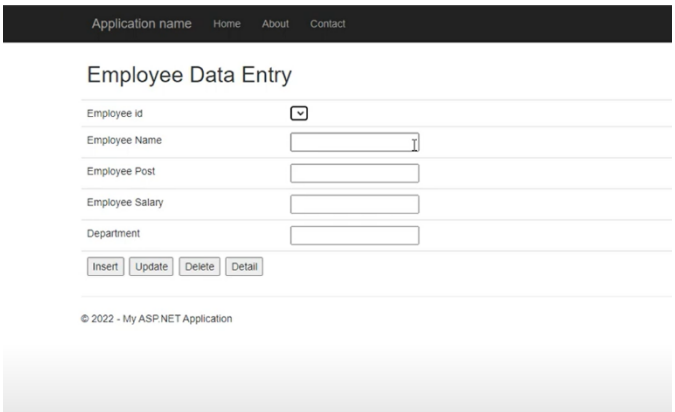
```
//Default.aspx
<%@PageTitle="Home
Page"Language="C#"MasterPageFile="~/Site.Master"AutoEventWireup="true"CodeBehind="De
fault.aspx.cs"Inherits="Practical_13_CRUD_operations_using_Store_Procedure._Default"
%>

<asp:ContentID="BodyContent"ContentPlaceHolderID="MainContent"runat="server">
<tableclass="table"style="width: 100%;">
<tr>
<thcolspan="3"align="center"><h2>Employee Data Entry</h2></th>
</tr>
<tr>
<tdcolspan="3">
<asp:LiteralID="Literal1"runat="server"></asp:Literal></td>
</tr>
<tr>
<td>
<asp:LabelID="Label5"runat="server"Text="Employee id"></asp:Label></td>
<tdcolspan="2">
<asp:DropDownListID="DropDownList1"runat="server"OnSelectedIndexChanged="DropDownLis
t1_SelectedIndexChanged"AutoPostBack="True"></asp:DropDownList></td>
</tr>
<tr>
<td>
<asp:LabelID="Label1"runat="server"Text="Employee Name"></asp:Label></td>
<td>
<asp:TextBoxID="txtEmployeeName"runat="server"AutoPostBack="True"></asp:TextBox></td
>
<td>
<asp:RequiredFieldValidatorID="RequiredFieldValidator1"runat="server"ErrorMessage="E
mpname is
required"ControlToValidate="txtEmployeeName"></asp:RequiredFieldValidator></td>

</tr>
<tr>
<td>
<asp:LabelID="Label2"runat="server"Text="Employee Post"></asp:Label></td>
<td>
<asp:TextBoxID="txtEmployeePost"runat="server"AutoPostBack="True"></asp:TextBox></td
>
<td>
```

```
<asp:RequiredFieldValidatorID="RequiredFieldValidator2"runat="server"ErrorMessage="E
mployee Post is
required"ControlToValidate="txtEmployeePost"></asp:RequiredFieldValidator></td>
</tr>
<tr>
<td>
<asp:LabelID="Label3"runat="server"Text="Employee Salary"></asp:Label></td>
<td>
<asp:TextBoxID="txtEmployeeSal"runat="server"AutoPostBack="True"></asp:TextBox></td>
<td>
<asp:RequiredFieldValidatorID="RequiredFieldValidator3"runat="server"ErrorMessage="E
mployee Salary is
required"ControlToValidate="txtEmployeeSal"></asp:RequiredFieldValidator></td>
</tr>
<tr>
<td>
<asp:LabelID="Label4"runat="server"Text="Department"></asp:Label></td>
<td>
<asp:TextBoxID="txtEmployeeDept"runat="server"AutoPostBack="True"></asp:TextBox></td
>
<td>
<asp:RequiredFieldValidatorID="RequiredFieldValidator4"runat="server"ErrorMessage="D
epartment is
required"ControlToValidate="txtEmployeeDept"></asp:RequiredFieldValidator></td>
</tr>
<tr>
<tdcolspan="3">
<asp:ButtonID="Button1"runat="server"Text="Insert"OnClick="Button1_Click"/> 
<asp:ButtonID="Button2"runat="server"Text="Update"OnClick="Button2_Click"/> 
<asp:ButtonID="Button3"runat="server"Text="Delete"OnClick="Button3_Click"/> 
<asp:ButtonID="Button4"runat="server"Text="Detail"/> 

</tr>
</table>

</asp:Content>

//Default.aspx.cs
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Data;
usingSystem.Data.SqlClient;
namespace Practical_13_CRUD_operations_using_Store_Procedure
{
publicpartialclass_Default : Page
    {
        String constr;
SqlConnection conn;
SqlDataAdapter adapter;
DataSet ds;
SqlCommandcmd;
protectedvoidPage_Load(object sender, EventArgs e)
        {
//Code for inserting record
if (!IsPostBack)
            {
```

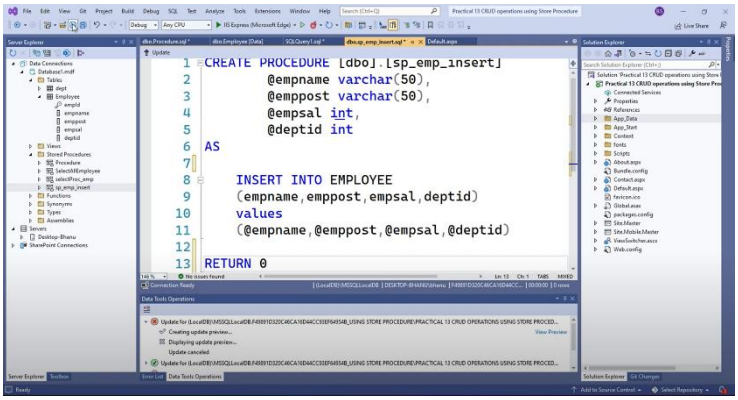
```
constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integ
rated Security=True";
                conn = newSqlConnection(constr);
cmd = newSqlCommand("sp_emp_selectAll",conn);
                adapter = newSqlDataAdapter(cmd.CommandText, conn);
                ds = newDataSet();
adapter.Fill(ds, "tempEmployee");
for (inti = 0; i<ds.Tables[0].Rows.Count; i++)
                {

DropDownList1.Items.Add(ds.Tables[0].Rows[i].ItemArray[0].ToString());
                }
txtEmployeeName.Text = ds.Tables[0].Rows[0].ItemArray[1].ToString();
txtEmployeePost.Text = ds.Tables[0].Rows[0].ItemArray[2].ToString();
txtEmployeeSal.Text = ds.Tables[0].Rows[0].ItemArray[3].ToString();
txtEmployeeDept.Text = ds.Tables[0].Rows[0].ItemArray[4].ToString();
            }
        }

protectedvoid DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {
constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integ
rated Security=True";
            conn = newSqlConnection(constr);
cmd = newSqlCommand();
            String id = DropDownList1.SelectedValue;
            adapter = newSqlDataAdapter("select * from employee where empid=" + id,
conn);
            ds = newDataSet();
adapter.Fill(ds, "tempEmployee");
txtEmployeeName.Text = ds.Tables[0].Rows[0].ItemArray[1].ToString();
txtEmployeePost.Text = ds.Tables[0].Rows[0].ItemArray[2].ToString();
txtEmployeeSal.Text = ds.Tables[0].Rows[0].ItemArray[3].ToString();
txtEmployeeDept.Text = ds.Tables[0].Rows[0].ItemArray[4].ToString();
        }

protectedvoid Button1_Click(object sender, EventArgs e)
        {
//Code for inserting record
constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integ
rated Security=True";
//step 1  Create connection
            conn = newSqlConnection(constr);
//step 2  Create Command object
cmd = newSqlCommand("sp_emp_insert",conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@empname", txtEmployeeName.Text);
cmd.Parameters.AddWithValue("@emppost", txtEmployeePost.Text);
cmd.Parameters.AddWithValue("@empsal", txtEmployeeSal.Text);
cmd.Parameters.AddWithValue("@deptid", txtEmployeeDept.Text);
conn.Open();
cmd.ExecuteNonQuery();
conn.Close();
            Literal1.Text = "Record inserted successfully";
        }

protectedvoid Button2_Click(object sender, EventArgs e)
        {
//updating record
```

```
//Code for updating record
constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integ
rated Security=True";
//step 1  Create connection
            conn = newSqlConnection(constr);
//step 2  Create Command object
cmd = newSqlCommand("sp_emp_update", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@empname", txtEmployeeName.Text);
cmd.Parameters.AddWithValue("@emppost", txtEmployeePost.Text);
cmd.Parameters.AddWithValue("@empsal", txtEmployeeSal.Text);
cmd.Parameters.AddWithValue("@deptid", txtEmployeeDept.Text);
cmd.Parameters.AddWithValue("@empid", DropDownList1.SelectedValue);
conn.Open();
cmd.ExecuteNonQuery();
conn.Close();
            Literal1.Text = "Record Updated successfully";
        }

protectedvoid Button3_Click(object sender, EventArgs e)
        {
//deleting record
//Code for deleting record
constr = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\Database1.mdf;Integ
rated Security=True";
//step 1  Create connection
            conn = newSqlConnection(constr);
//step 2  Create Command object
cmd = newSqlCommand("sp_emp_delete", conn);
cmd.CommandType = CommandType.StoredProcedure;
cmd.Parameters.AddWithValue("@empid", DropDownList1.SelectedValue);
conn.Open();
cmd.ExecuteNonQuery();
conn.Close();
            Literal1.Text = "Record deleted successfully";
        }
    }
}
```

**Output:**

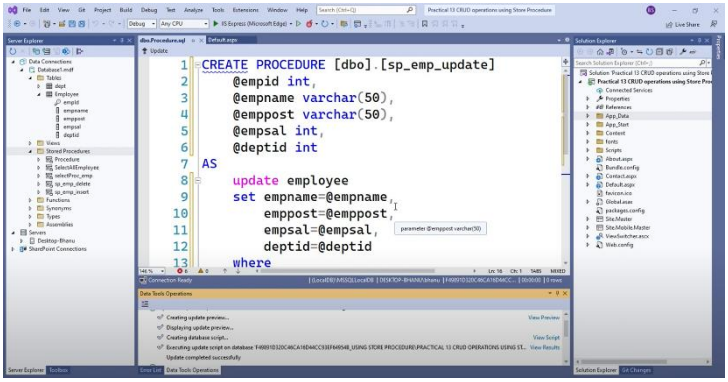


Insertion Stored procedure:

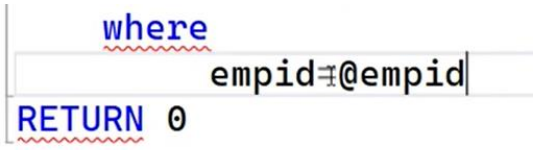


```
CREATE PROCEDURE [dbo].[sp_emp_insert]
        @empname varchar(50),
        @emppost varchar(50),
        @empsal int,
        @deptid int
AS

    INSERT INTO EMPLOYEE
    (empname,emppost,empsal,deptid)
    values
    (@empname,@emppost,@empsal,@deptid)

RETURN 0
```

Updation Stored Procedure:

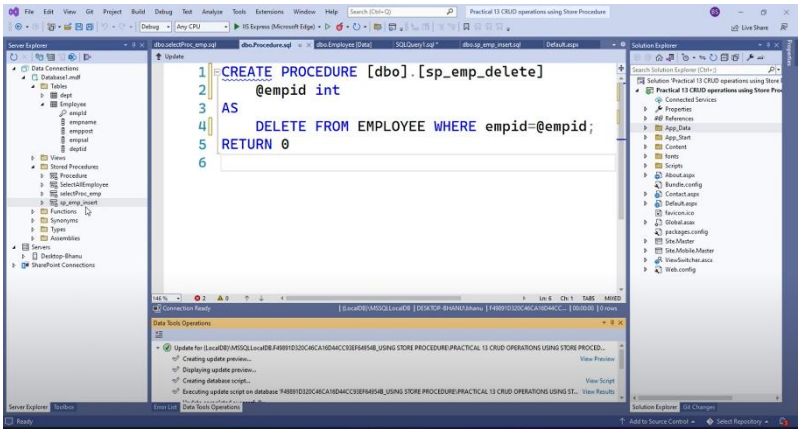


```
CREATE PROCEDURE [dbo].[sp_emp_update]
    @empid int,
    @empname varchar(50),
    @emppost varchar(50),
    @empsal int,
    @deptid int
AS
    update employee
    set empname=@empname,
        emppost=@emppost,
        empsal=@empsal,
        deptid=@deptid
    where
```



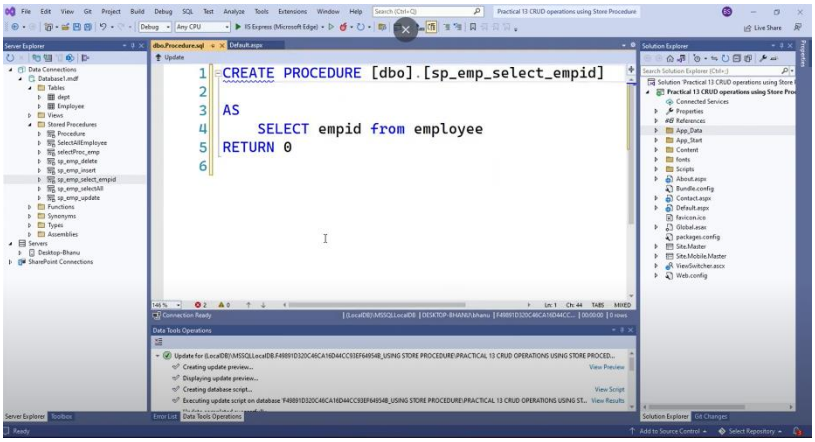
```
    where
        empid=@empid
RETURN 0
```

Deletion Stored Procedure

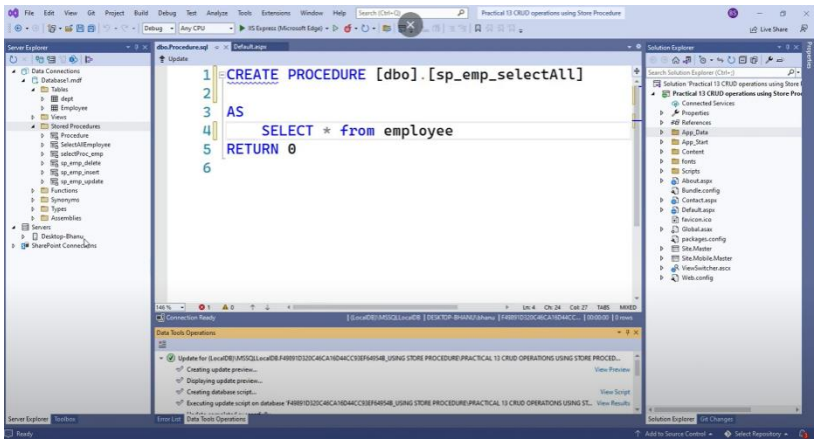


```
CREATE PROCEDURE [dbo].[sp_emp_delete]
    @empid int
AS
    DELETE FROM EMPLOYEE WHERE empid=@empid;
RETURN 0
```

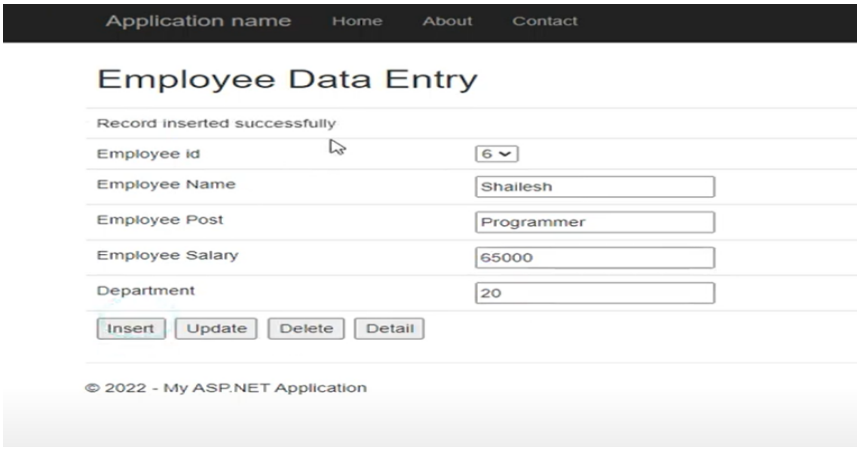
Selection of one details stored procedure:



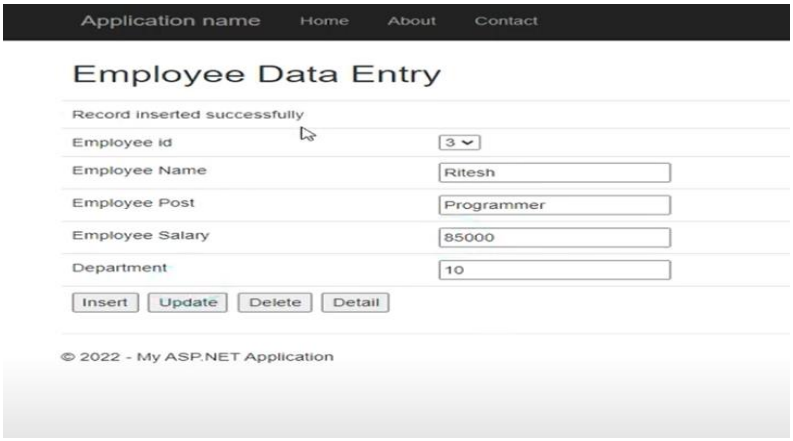

Selection of all details stored procedure:



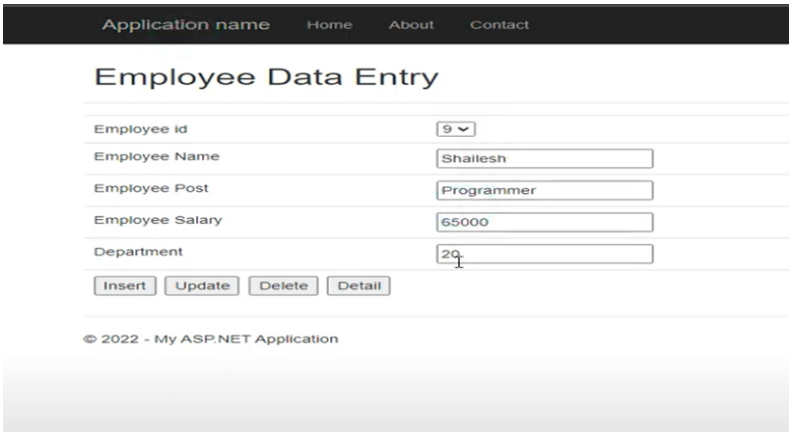

Insertion Output:



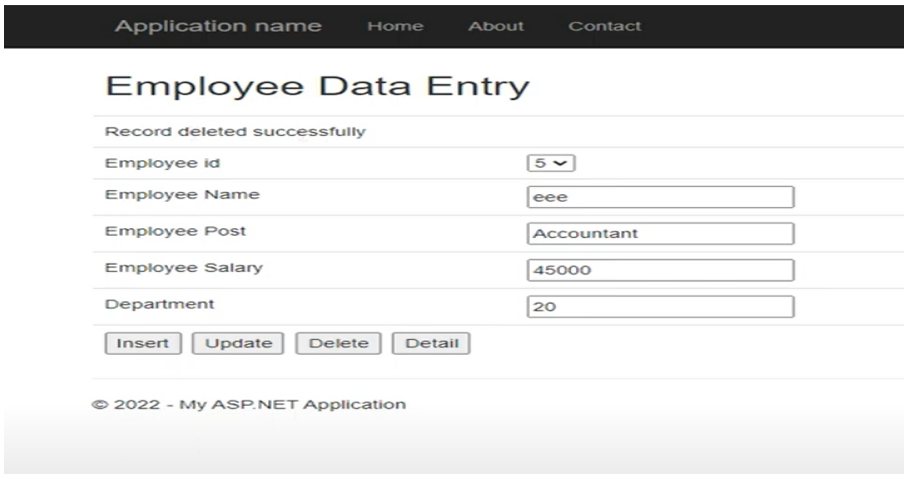
Updation Output:




Selection Output:




Deletion Output:



| **Practical No.: 16** | | **Date: 05/07/2022** |
|---|---|---|
| Aim: | Build websites to demonstrate the working of entity framework in dot net. | |

Create CRUD web Application by using following Model [Class Employee]
Which has following structure

- Empid
- EmpName
- EmpPost
- EmpSal
- Dept

Create Entity for each module and implement Add/Modify/Delete/Search functionality

**Source:**

Models

Model1.cs

```
using System;
usingSystem.ComponentModel.DataAnnotations.Schema;
usingSystem.Data.Entity;
usingSystem.Linq;

namespace Practical_14.Models
{
publicpartialclassModel1 : DbContext
    {
publicModel1()
            : base("name=Model1")
        {
        }

publicvirtualDbSet<Employee> Employees { get; set; }

protectedoverridevoidOnModelCreating(DbModelBuildermodelBuilder)
        {
modelBuilder.Entity<Employee>()
                .Property(e =>e.Empname)
                .IsUnicode(false);

modelBuilder.Entity<Employee>()
                .Property(e =>e.Emppost)
                .IsUnicode(false);

modelBuilder.Entity<Employee>()
                .Property(e =>e.Dept)
                .IsFixedLength();
        }
    }
}
```

Employee.cs

```
namespace Practical_14.Models
{
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel.DataAnnotations;
usingSystem.ComponentModel.DataAnnotations.Schema;
usingSystem.Data.Entity.Spatial;
```

```
    [Table("Employee")]
publicpartialclassEmployee
    {
        [Key]
publicintEmpId { get; set; }

        [Required]
        [StringLength(50)]
publicstringEmpname { get; set; }

        [Required]
        [StringLength(50)]
publicstringEmppost { get; set; }

publicintEmpsalary { get; set; }

        [Required]
        [StringLength(10)]
publicstring Dept { get; set; }
    }
}
```

View

Index.cshtml

```
@model IEnumerable<Practical_14.Models.Employee>

@{
ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
@Html.ActionLink("Create New", "Create")
</p>
<tableclass="table">
<tr>
<th>
@Html.DisplayNameFor(model =>model.Empname)
</th>
<th>
@Html.DisplayNameFor(model =>model.Emppost)
</th>
<th>
@Html.DisplayNameFor(model =>model.Empsalary)
</th>
<th>
@Html.DisplayNameFor(model =>model.Dept)
</th>
<th></th>
</tr>

@foreach (var item in Model) {
<tr>
<td>
@Html.DisplayFor(modelItem =>item.Empname)
</td>
<td>
@Html.DisplayFor(modelItem =>item.Emppost)
</td>
```

```
<td>
@Html.DisplayFor(modelItem =>item.Empsalary)
</td>
<td>
@Html.DisplayFor(modelItem =>item.Dept)
</td>
<td>
@Html.ActionLink("Edit", "Edit", new { id=item.EmpId }) |
@Html.ActionLink("Details", "Details", new { id=item.EmpId }) |
@Html.ActionLink("Delete", "Delete", new { id=item.EmpId })
</td>
</tr>
}

</table>
```

Create.cshtml

```
@model Practical_14.Models.Employee

@{
ViewBag.Title = "Create";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
@Html.AntiForgeryToken()

<divclass="form-horizontal">
<h4>Employee</h4>
<hr/>
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
<divclass="form-group">
@Html.LabelFor(model =>model.Empname, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Empname, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Empname, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Emppost, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Emppost, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Emppost, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Empsalary, htmlAttributes: new { @class = "control-
label col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Empsalary, new { htmlAttributes = new { @class =
"form-control" } })
```

```
@Html.ValidationMessageFor(model =>model.Empsalary, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Dept, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Dept, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Dept, "", new { @class = "text-danger" })
</div>
</div>

<divclass="form-group">
<divclass="col-md-offset-2 col-md-10">
<inputtype="submit"value="Create"class="btnbtn-default"/>
</div>
</div>
</div>
}

<div>
@Html.ActionLink("Back to List", "Index")
</div>
```

Edit.cshtml

```
@model Practical_14.Models.Employee

@{
ViewBag.Title = "Edit";
}

<h2>Edit</h2>

@using (Html.BeginForm())
{
@Html.AntiForgeryToken()

<divclass="form-horizontal">
<h4>Employee</h4>
<hr/>
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
@Html.HiddenFor(model =>model.EmpId)

<divclass="form-group">
@Html.LabelFor(model =>model.Empname, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Empname, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Empname, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Emppost, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
```

```
@Html.EditorFor(model =>model.Emppost, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Emppost, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Empsalary, htmlAttributes: new { @class = "control-
label col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Empsalary, new { htmlAttributes = new { @class =
"form-control" } })
@Html.ValidationMessageFor(model =>model.Empsalary, "", new { @class = "text-danger"
})
</div>
</div>

<divclass="form-group">
@Html.LabelFor(model =>model.Dept, htmlAttributes: new { @class = "control-label
col-md-2" })
<divclass="col-md-10">
@Html.EditorFor(model =>model.Dept, new { htmlAttributes = new { @class = "form-
control" } })
@Html.ValidationMessageFor(model =>model.Dept, "", new { @class = "text-danger" })
</div>
</div>

<divclass="form-group">
<divclass="col-md-offset-2 col-md-10">
<inputtype="submit"value="Save"class="btnbtn-default"/>
</div>
</div>
</div>
}

<div>
@Html.ActionLink("Back to List", "Index")
</div>
```

Details.cshtml

```
@model Practical_14.Models.Employee

@{
ViewBag.Title = "Details";
}

<h2>Details</h2>

<div>
<h4>Employee</h4>
<hr/>
<dlclass="dl-horizontal">
<dt>
@Html.DisplayNameFor(model =>model.Empname)
</dt>

<dd>
@Html.DisplayFor(model =>model.Empname)
</dd>

<dt>
```

```
@Html.DisplayNameFor(model =>model.Emppost)
</dt>

<dd>
@Html.DisplayFor(model =>model.Emppost)
</dd>

<dt>
@Html.DisplayNameFor(model =>model.Empsalary)
</dt>

<dd>
@Html.DisplayFor(model =>model.Empsalary)
</dd>

<dt>
@Html.DisplayNameFor(model =>model.Dept)
</dt>

<dd>
@Html.DisplayFor(model =>model.Dept)
</dd>

</dl>
</div>
<p>
@Html.ActionLink("Edit", "Edit", new { id = Model.EmpId }) |
@Html.ActionLink("Back to List", "Index")
</p>
```

Delete.cshtml

```
@model Practical_14.Models.Employee

@{
ViewBag.Title = "Delete";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
<h4>Employee</h4>
<hr/>
<dlclass="dl-horizontal">
<dt>
@Html.DisplayNameFor(model =>model.Empname)
</dt>

<dd>
@Html.DisplayFor(model =>model.Empname)
</dd>

<dt>
@Html.DisplayNameFor(model =>model.Emppost)
</dt>

<dd>
@Html.DisplayFor(model =>model.Emppost)
</dd>

<dt>
@Html.DisplayNameFor(model =>model.Empsalary)
```

```
</dt>

<dd>
@Html.DisplayFor(model =>model.Empsalary)
</dd>

<dt>
@Html.DisplayNameFor(model =>model.Dept)
</dt>

<dd>
@Html.DisplayFor(model =>model.Dept)
</dd>

</dl>

@using (Html.BeginForm()) {
@Html.AntiForgeryToken()

<divclass="form-actions no-color">
<inputtype="submit"value="Delete"class="btnbtn-default"/> |
@Html.ActionLink("Back to List", "Index")
</div>
    }
</div>
```

Controller

EmployeeController.cs

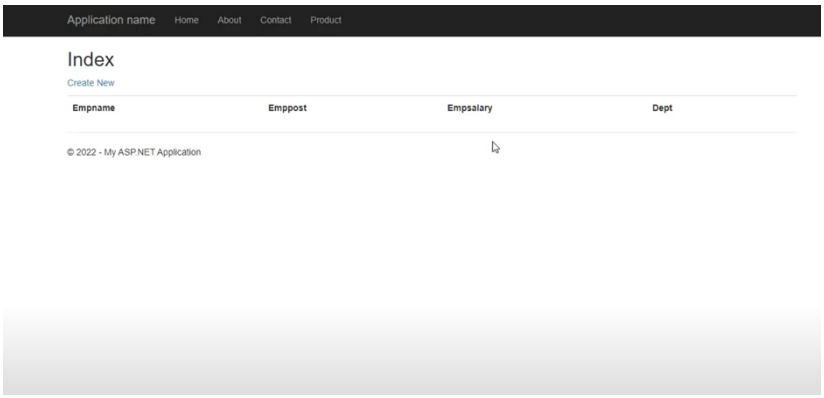
```
using System;
usingSystem.Collections.Generic;
usingSystem.Data;
usingSystem.Data.Entity;
usingSystem.Linq;
using System.Net;
usingSystem.Web;
usingSystem.Web.Mvc;
using Practical_14.Models;

namespace Practical_14.Controllers
{
publicclassEmployeesController : Controller
    {
private Model1 db = new Model1();

// GET: Employees
publicActionResult Index()
        {
return View(db.Employees.ToList());
        }

// GET: Employees/Details/5
publicActionResult Details(int? id)
        {
if (id == null)
            {
returnnewHttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Employee employee = db.Employees.Find(id);
if (employee == null)
            {
```

```
returnHttpNotFound();
            }
return View(employee);
        }

// GET: Employees/Create
publicActionResult Create()
        {
return View();
        }

// POST: Employees/Create
// To protect from overposting attacks, enable the specific properties you want to
bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
publicActionResult Create([Bind(Include = "EmpId,Empname,Emppost,Empsalary,Dept")]
Employee employee)
        {
if (ModelState.IsValid)
            {
db.Employees.Add(employee);
db.SaveChanges();
returnRedirectToAction("Index");
            }

return View(employee);
        }

// GET: Employees/Edit/5
publicActionResult Edit(int? id)
        {
if (id == null)
            {
returnnewHttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Employee employee = db.Employees.Find(id);
if (employee == null)
            {
returnHttpNotFound();
            }
return View(employee);
        }

// POST: Employees/Edit/5
// To protect from overposting attacks, enable the specific properties you want to
bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
publicActionResult Edit([Bind(Include = "EmpId,Empname,Emppost,Empsalary,Dept")]
Employee employee)
        {
if (ModelState.IsValid)
            {
db.Entry(employee).State = EntityState.Modified;
db.SaveChanges();
returnRedirectToAction("Index");
            }
return View(employee);
        }
```

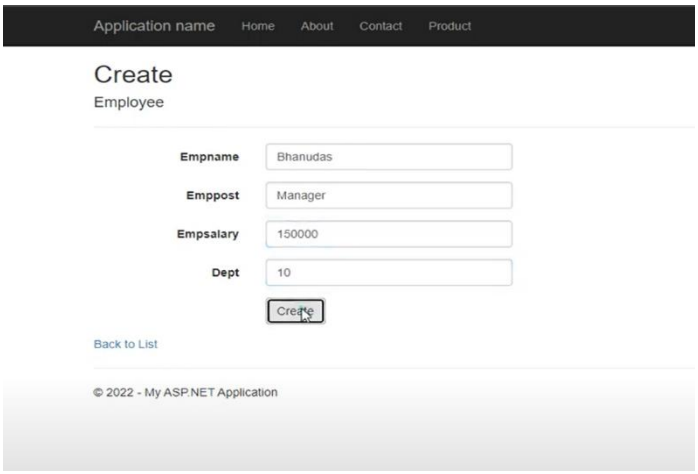
```
// GET: Employees/Delete/5
publicActionResult Delete(int? id)
        {
if (id == null)
            {
returnnewHttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Employee employee = db.Employees.Find(id);
if (employee == null)
            {
returnHttpNotFound();
            }
return View(employee);
        }

// POST: Employees/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
publicActionResultDeleteConfirmed(int id)
        {
            Employee employee = db.Employees.Find(id);
db.Employees.Remove(employee);
db.SaveChanges();
returnRedirectToAction("Index");
        }

protectedoverridevoid Dispose(bool disposing)
        {
if (disposing)
            {
db.Dispose();
            }
base.Dispose(disposing);
        }
    }
}
```

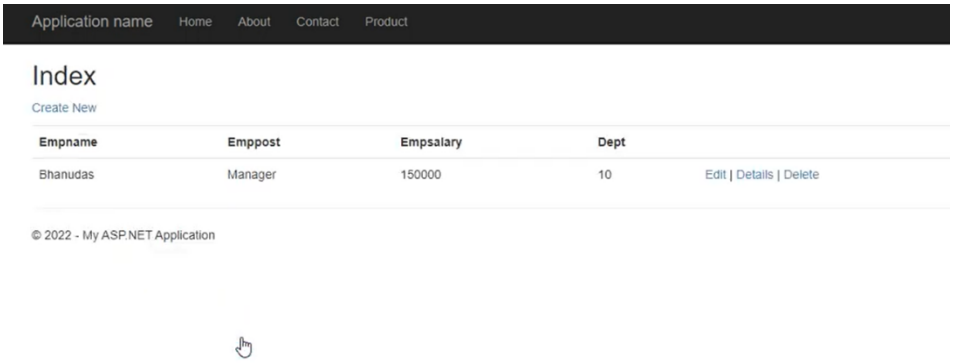
| | | |
|---|---|---|
| **Output:** | | |

Index page output

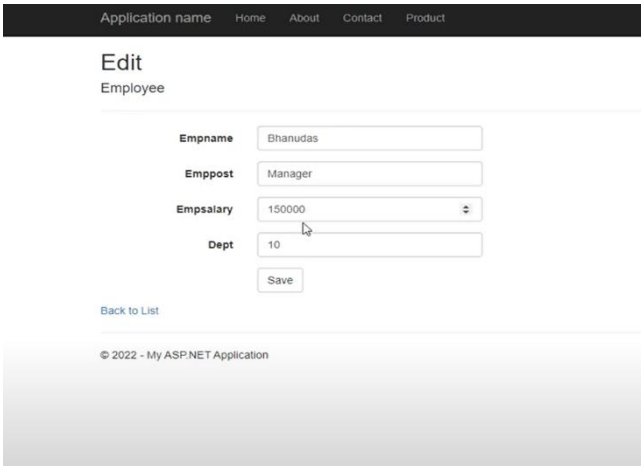


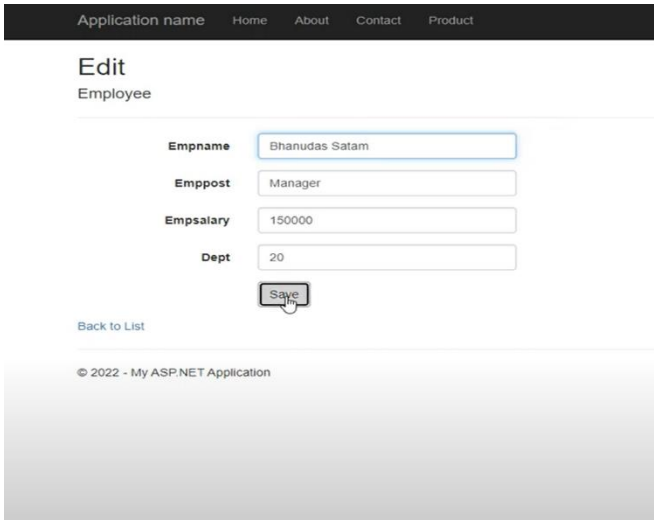

Employee Creation Output:

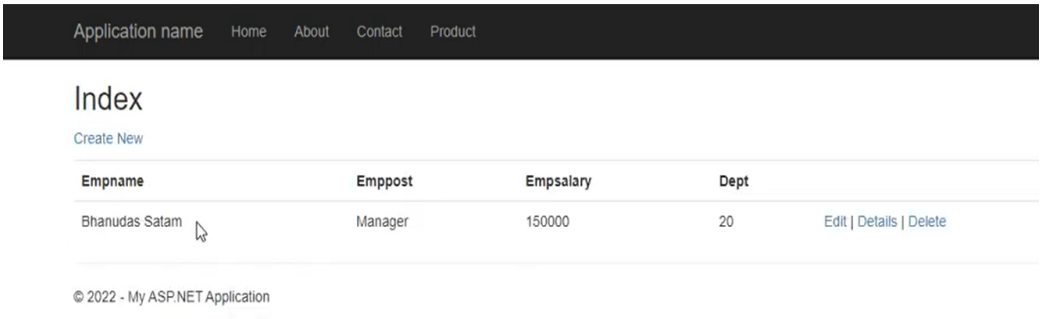


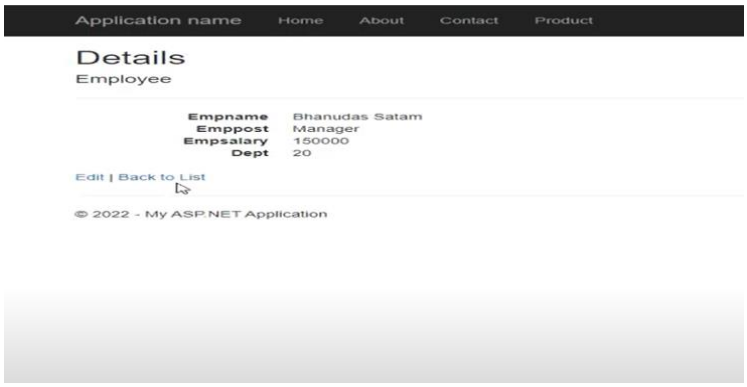
Edit Output:


Application name
Home
About
Contact
Product
Edit
Employee
Empname
Bhanudas
Emppost
Manager
Empsalary
150000
Dept
10
Save
Back to List
© 2022 - My ASP.NET Application


Application name
Home
About
Contact
Product
Edit
Employee
Empname
Bhanudas Satam
Emppost
Manager
Empsalary
150000
Dept
20
Save
Back to List
© 2022 - My ASP.NET Application


Application name
Home
About
Contact
Product
Index
Create New
Empname
Emppost
Empsalary
Dept
Bhanudas Satam
Manager
150000
20
Edit | Details | Delete
© 2022 - My ASP.NET Application

Details Output:




Deletion Output