PRACTICAL NO. : 01(A)

AIM: Write a console application that obtains four int values from the user and displays the product.

Hint: you may recall that the Convert.ToDouble() command was used to convert the input from the console to a double; the equivalent command to convert from a string to an int is Convert.ToInt32().

CODE:

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int num1, num2,num3,num4,prod;
         Console.Write("Enter number 1: ");
         num1 = Int32.Parse(Console.ReadLine());
         Console.Write("Enter number 2: ");
         num2 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 3: ");
         num3 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter number 4: ");
         num4 = Convert.ToInt32(Console.ReadLine());
         prod = num1 * num2 * num3 * num4;
         Console.WriteLine(num1 + "*" + num2 + "*" + num3 + "*" + num4 + "=" + prod);
      }
   }
}
```

OUTPUT:

```
Enter number 1: 6
Enter number 2: 5
Enter number 3: 4
Enter number 4: 3
6*5*4*3=360
```

PRACTICAL NO. : 01(B)

AIM: If you have two integers stored in variables var1 and var2, what Boolean test can you perform to see if one or the other (but not both) is greater than 10?

CODE:

```
using System;
namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int var1, var2;
         Console.Write("Enter number 1: ");
         var1 = Int32.Parse(Console.ReadLine());
         Console.Write("Enter number 2: ");
         var2 = Convert.ToInt32(Console.ReadLine());
         if ((var1 > 10 && var2 <= 10) || (var2 > 10 && var1 <= 10))
         {
            Console.WriteLine("Boolean test succedded \n Both number are not >10");
         }
      }
   }
}
```

OUTPUT:

```
Enter number 1: 5
Enter number 2: 11
Boolean test succedded
 Both number are not >10
```

PRACTICAL NO. : 01(C)

AIM: Write an application that includes the logic from Exercise 1, obtains two numbers from the user, and displays them, but rejects any input where both numbers are greater than 10 and asks for two new numbers.

CODE:

```
using System;
namespace ConsoleApplication2
{
   class Program
   {
      static void Main(string[] args)
      {
         int var1, var2;
         label1:
         Console.Write("Enter number 1: ");
         var1 = Int32.Parse(Console.ReadLine());
         Console.Write("Enter number 2: ");
         var2 = Convert.ToInt32(Console.ReadLine());
         if ((var1 > 10 && var2 > 10) )
         {
            Console.WriteLine("Both No are greater than 10 are not allowed");
            goto label1;
         }
         else
         {
            Console.WriteLine("Number 1: "+var1);
            Console.WriteLine("Number 2 :"+var2);
         }
      }
   }
}
```

OUTPUT:

Enter number 1:15
Enter number 2: 16
Both no. are greater than 10 are not allowed
Enter number 1:5
Enter number 2: 15
Number 1: 5
Number 2 :15

PRACTICAL NO. : 01(D)

AIM: Write a console application that places double quotation marks around each word in a string .

CODE:

```
using System;
namespace ConsoleApplication3
{
   class Program
   {
      static void Main(string[] args)
      {
         string str1;
         Console.Write("Enter string 1: ");
         str1 = Console.ReadLine();
         string[] words = str1.Split(' ');
         for (int i = 0; i < words.Length; i++)
         {
            Console.Write("\" " + words[i] + "\" ");
         }
      }
   }
}
```

OUTPUT:

Enter string 1: we can and we will
“we” “can” “and” “we” “will”

PRACTICAL NO. : 01(E)

AIM: Write an application that uses two command-line arguments to place values into a string and an integer variable, respectively. Then display these values.

CODE:

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

OUTPUT:

String : Roman
Number : 10

PRACTICAL NO. : 01(F)

AIM: Write an application that receives the following information from a set of students:
Student Id:
Student Name:
Course Name:
Date of Birth:
The application should also display the information of all the students once the data is Entered. Implement this using an Array of Structures.

CODE:

```
using System;
namespace ArrayOfStructs
{
   class Program
   {
      struct Student
      {
         public string studid, name, cname;
         public int day, month, year;
      }
      static void Main(string[] args)
      {
         Student[] s = new Student[5];
         int i;
         for (i = 0; i < 5; i++)
         {
            Console.Write("Enter Student Id:");
            s[i].studid = Console.ReadLine();
            Console.Write("Enter Student name : ");
            s[i].name = Console.ReadLine();
            Console.Write("Enter Course name : ");
            s[i].cname = Console.ReadLine();
            Console.Write("Enter date of birth\n Enter day(1-31):");
            s[i].day = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter month(1-12):");
            s[i].month = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter year:");
            s[i].year = Convert.ToInt32(Console.ReadLine());
         }
         Console.WriteLine("\n\nStudent's List\n");
         for (i = 0; i < 5; i++)
         {
            Console.WriteLine("\nStudent ID : " + s[i].studid);
            Console.WriteLine("\nStudent name : " + s[i].name);
            Console.WriteLine("\nCourse name : " + s[i].cname);
            Console.WriteLine("\nDate of birth(dd-mm-yy) : " + s[i].day + "-" + s[i].month +
"-" + s[i].year);
         } } } }
```

OUTPUT:

```
Enter Student Id:0001
Enter Student name : Prachit
Enter Course name : MSCit
Enter date of birth
 Enter day(1-31):29
Enter month(1-12):9
Enter year:1995
Enter Student Id:0002
Enter Student name : Aniket
Enter Course name : Bscit
Enter date of birth
 Enter day(1-31):4
Enter month(1-12):3
Enter year:1996
Enter Student Id:0003
Enter Student name : Prathamesh
Enter Course name : BMS
Enter date of birth
 Enter day(1-31):9
Enter month(1-12):8
Enter year:2000
Enter Student Id:0004
Enter Student name : Sumit
Enter Course name :MScet
Enter date of birth
 Enter day(1-31):25
Enter month(1-12):5
Enter year:1994
Enter Student Id : 0005
Enter Student name : Zaid
Enter Course name : BCOM
Enter date of birth
 Enter day(1-31):6
Enter month(1-12):7
Enter year:1993

Student's List

Student ID : 0001
Student name : Prachit
Course name : MSCit
Date of birth(dd-mm-yy) : 29-9-1995
Student ID : 0002
Student name : Aniket
Course name : Bscit
Date of birth(dd-mm-yy) : 4-3-1996
Student ID : 0003
Student name : Prathamesh
Course name : BMS
Date of birth(dd-mm-yy) : 9-8-2000
```

Student ID : 0004
Student name : Sumit
Course name : MScet
Date of birth(dd-mm-yy) : 25-5-1994
Student ID : 0005
Student name : Zaid
Course name : BCOM
Date of birth(dd-mm-yy) : 6-7-1993

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
I) Generate Fibonacci series.

CODE:

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

         Console.Write(num1+"\t"+num2);
         while(counter<=num)
         {
            num3 = num1 + num2;
                        if (counter >= num)
               break;
           Console.Write("\t" + num3);
            num1 = num2;
            num2 = num3;
            counter++;
         }
      }
   }
}
```

OUTPUT:

```
Upto how many number you want fibonacci series:5
0       1       1       2       3
```

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
II) Generate various patterns (triangles, diamond and other patterns) with numbers.

CODE -1:

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int row, col;
        for (row = 1; row <= 5; row++)
         {
            for (col = 1; col <= row; col++)
               Console.Write(col);
            Console.WriteLine();
         }
      }
   }
}
```

OUTPUT:

```
1
12
123
1234
12345
```

CODE -2:

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int row, sp, col;
         for (row = 1; row <= 5; row++)
         {
            for (sp = 1; sp <= 5 - row; sp++)
            {
               Console.Write(' ');
            }
            for (col = 1; col <= row; col++)
            {
               Console.Write(col);
            }
               Console.WriteLine();
            }
      } } } }
```

OUTPUT:

```
    1
   12
  123
 1234
12345
```

CODE -3:

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int row, sp, col,revcol;
         for (row = 1; row <= 5; row++)
         {
            for (sp = 1; sp <= 5 - row; sp++)
            {
               Console.Write(' ');
            }
            for (col = 1; col <= row; col++)
            {
               Console.Write(col);
            }
            for (revcol = col - 2; revcol >= 1; revcol--)
            {
               Console.Write(revcol);
            }
               Console.WriteLine();
         }
      }
   }
}
```

OUTPUT:

```
    1
   121
  12321
 1234321
123454321
```

CODE-4:

```
using System;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {
         int row, sp, col, revcol;
for (row = 1; row <= 5; row++)                    {
            for (sp = 1; sp <= 5 - row; sp++)
            {
               Console.Write(' ');
            }
            for (col = 1; col <= row; col++)
            {
               Console.Write(col);
            }
            for (revcol = col - 2; revcol >= 1; revcol--)
            { Console.Write(revcol); }
            Console.WriteLine();
         }
         for (row = 4; row >= 1; row--)  {
            for (sp = 1; sp <= 5 - row; sp++)
            {
               Console.Write(' ');
            }
            for (col = 1; col <= row; col++)
            {
               Console.Write(col);
            }
            for (revcol = col - 2; revcol >= 1; revcol--)
            { Console.Write(revcol); }
            Console.WriteLine();
         }  } } }
```

OUTPUT:

```
    1
   121
  12321
 1234321
123454321
 1234321
  12321
   121
    1
```

CODE-5:

```
using System;
namespace pattern
{
  class Program
   {
      static void Main(string[] args)
      {
         int row, col,sp,reverse;
           for (row = 1; row <= 5; row++)
         {
            for (sp = 1; sp <= 5 - row; sp++)
               Console.Write(" ");
            for (col = 1; col <= row; col++)
               if (col == 1)
                  Console.Write("*");
               else
                  Console.Write(" ");
            for (reverse = col - 2; reverse >= 1; reverse--)
               if (reverse == 1)
                  Console.Write("*");
               else
                  Console.Write(" ");
            Console.WriteLine();
         }
         for (row = 4; row >=1; row--)
         {
            for (sp = 1; sp <= 5 - row; sp++)
               Console.Write(" ");
            for (col = 1; col <= row; col++)
               if (col == 1)
                  Console.Write("*");
               else
                  Console.Write(" ");
            for (reverse = col - 2; reverse >= 1; reverse--)
               if (reverse == 1)
                  Console.Write("*");
               else
                  Console.Write(" ");
            Console.WriteLine();
         }
      } } }
```

OUTPUT:

```
    *
   * *
  *   *
  *   *
  *   *
  *   *
  *   *
   * *
    *
```

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
III) Test for prime numbers.

CODE:

```
using System;
namespace testprime
{
   class Program
   {
      static void Main(string[] args)
      {
         int num, counter;
         Console.Write("Enter number:");
         num = int.Parse(Console.ReadLine());
         for (counter = 2; counter <= num / 2; counter++)
         {
            if ((num % counter) == 0)
               break;
         }
         if (num == 1)
            Console.WriteLine(num + "is neither prime nor composite");
         else if(counter<(num/2))
         Console.WriteLine(num+"is not prime number");
         else
         Console.WriteLine(num+"is prime number");
      }
   }
}
```

OUTPUT:

(1st attempt)
Enter number:3
3 is prime number
(2nd)
Enter number:1
1 is neither prime nor composite
(3rd)
Enter number:4
4 is not prime number

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
IV) Generate prime numbers.

CODE:

```
using System;
namespace testprime
{
   class Program
   {
      static void Main(string[] args)
      {
         int counter, lowerlimit, upperlimit, limitCounter;
         Console.Write("Enter lowerlimit:"); lowerlimit =
         int.Parse(Console.ReadLine());
         Console.Write("Enter upperlimit:"); upperlimit =
         int.Parse(Console.ReadLine());
         Console.WriteLine("Prime number between " + lowerlimit + "and " + upperlimit + "
are ");
         for (limitCounter = lowerlimit; limitCounter <= upperlimit; limitCounter++)
         {
            for (counter = 2; counter <= limitCounter / 2; counter++)
            {
               if ((limitCounter % counter) == 0)
                  break;
            }
            if (limitCounter == 1)
               Console.WriteLine(limitCounter + "is neither prime nor composite");
            else if (counter >= (limitCounter / 2))
               Console.WriteLine(limitCounter + "\t");
              }
          Console.WriteLine();
      }}}
```

OUTPUT:

```
Enter lowerlimit:1
Enter upperlimit:15
Prime number between 1and 15 are
1is neither prime nor composite
2
3
4
5
7
11
13
```

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
V) Reverse a number and find sum of digits of a number.

CODE:

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

OUTPUT:

```
Enter number:15
Reverse of 15=51
Sum of its digits:6
```

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
V) Test for vowels.

CODE:

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

OUTPUT:

```
Enter a character : a
a is vowel

Enter a character : p
p is not a vowel
```

PRACTICAL NO. : 01(G)

AIM: Write programs using conditional statements and loops:
VII) Use of foreach loop with arrays.

CODE:

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

OUTPUT:

```
Shield
Evaluation
DX
```

PRACTICAL NO. : 02(1)

AIM: Write a program to declare a class ‘staff’ having data members as name and post.accept this data 5for 5 staffs and display names of staff who are HOD.

CODE:

```
using System;
namespace staff
{
   class staff
   {
      string name, post;
      public void getdata()
      {
         Console.Write("Enter name and post:");
         name = Console.ReadLine();
         post = Console.ReadLine();
      }
      public void display()
      {
         Console.WriteLine(name + "\t\t" + post);
      }
      public string getPost()
      {
         return post;
      }
   }
   class program
   {
     static void Main(string[] args)
      {
         staff[] objStaff = new staff[5];
         int i;
         for (i = 0; i < 5; i++)
         {
            objStaff[i] = new staff();
            objStaff[i].getdata();
         }
         Console.WriteLine("Name \t\t Post");
         for (i = 0; i < 5; i++)
         {
            if (objStaff[i].getPost() == "HOD")
               objStaff[i].display();
         }
      }
   }
}
```

OUTPUT:

```
Enter name and post:Prachit
HOD
Enter name and post:Sumit
PM
Enter name and post:Aniket
HOD
Enter name and post:Prathamesh
PM
Enter name and post:Zaid
CA
Name          Post
Prachit         HOD
Aniket          HOD
```

PRACTICAL NO. : 02(2)

AIM: Write a program to declare class 'Distance' have data members dist1,dist2 ,dist3. Initialize the two data members using constructor and store their addition in third data member using function and display addition.

CODE:

```
using System;
namespace distanceclass
{
   class Distance
   {
      int dist1,dist2,dist3;
      public Distance(int dist1,int dist2)
      {
         this.dist1=dist1;
         this.dist2=dist2;
      }
      public void addition()
      {
         dist3=dist1+dist2;
      }
      public void display()
      {
         Console.WriteLine("Distance1:"+ dist1);
         Console.WriteLine("Distance1:"+ dist2);
         Console.WriteLine("Distance1:"+ dist3);
      } }
     class program
     {
      static void Main(string[] args)
      {
         Distance objDistance = new Distance(10, 20);
         objDistance.addition();
         objDistance.display();
      } } }
```

OUTPUT:

Distance1:10
Distance1:20
Distance1:30

PRACTICAL NO. : 02(3)

AIM: Write a program using function overloading to swap two integer numbers and swap two float numbers.

CODE:

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

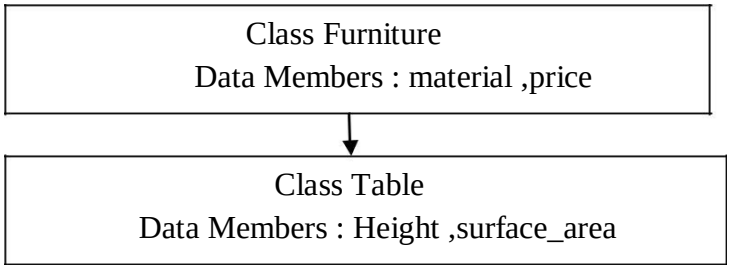
OUTPUT:

N=20 M=10

F1=20.6 F2=10.5

PRACTICAL NO. : 02(4)

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



CODE:

Furniture.cs

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

Table.cs

```
using System;
namespace SingleInheritance
{
   class Table:Furniture
   {
      int height, surface_area;
      public void getdata()
      {
         base.getdata();
         Console.Write("Enter height: ");

         height = int.Parse(Console.ReadLine());
         Console.Write("Enter surface area: ");
```

```
        surface_area = int.Parse(Console.ReadLine());
    }
    public void showdata()
    {
        base.showdata();
        Console.WriteLine("Height : " + height);
        Console.WriteLine("Surface Area : " + surface_area);
    } } }
```

Program.cs

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

OUTPUT:

Enter material : wood
Enter price : 1220
Enter height: 35
Enter surface area: 26
Material : wood
Price : 1220
Height : 35
Surface Area : 26

PRACTICAL NO. : 02(5)

AIM: Define a class 'salary' which will contain member variable Basic, TA, DA, HRA. Write a program using Constructor with default values for DA and HRA and calculate the salary of employee.

CODE:

Salary.cs

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

Program.cs

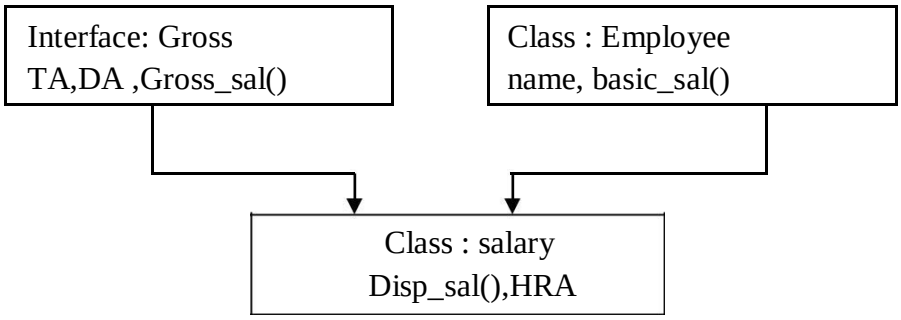
```
using System;
namespace SalaryConstructure
{
   class Program
   {
      static void Main(string[] args)
      {
         Salary s = new Salary();
         s.getdata();
         s.showdata();
      } } }
```

OUTPUT:

Enter basic salary : 52000
Enter travelling allowance : 3000
Basic salary : 52000
Dearness allowence : 9000
Housing rent allowence : 6000
Travelling allowence : 3000
Gross Salary : 70000

PRACTICAL NO. : 02(6)

AIM: Program to implement the following multiple inheritance using interface.



CODE:

Gross.cs

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

Employee.cs

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

Salary.cs

```
using System;
namespace MultipleInheritance
{
   class Salary:employee,Gross
   {
      int hra;
      public Salary(string name, int hra):base(name)
      { this.hra = hra; }
      public int ta
      {
         get {return S_ta; }
         set { S_ta = value; }
      }
      private int S_ta;
      public int da
      {
         get { return S_da; }
         set { S_da = value; }
      }
      private int S_da;
      public int GrossSal()
      {
         int gSal;
         gSal = hra + ta + da +
         BasicSal(15000); return gSal;
      }
      public void dispSal()
      { base.ShowData();
           Console.WriteLine("Gross Sal : " + GrossSal());
      } } }
```

Program.cs

```
using System;
namespace MultipleInheritance
{
   class Program
   {
      static void Main(string[] args)
      {
         Salary s = new Salary("Prachit", 35000);
         s.da = 20000;
         s.ta = 30000;
         s.dispSal();
      } } }
```

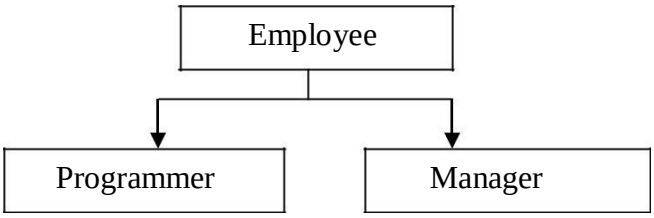
OUTPUT:

Name :Prachit

Gross Sal :100000

PRACTICAL NO. : 02(7)

AIM: Write a program for above class hierarchy for the Employee where the base class is Employee and derived class and Programmer and Manager. Here make display function virtual which is common for all and which will display information of Programmer and Manager interactively.



CODE:

Employee.cs

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

Programmer.cs

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

Manager.cs

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

Program.cs

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

OUTPUT:

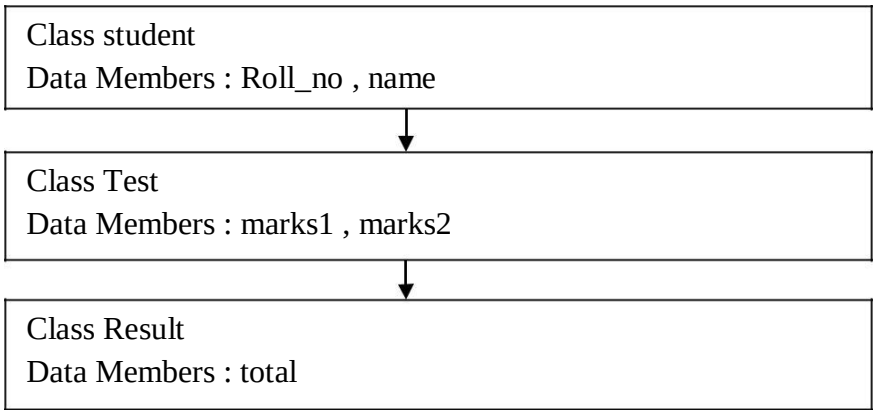
Whose details you want to use to see
1.Programmer
2.Manager1
Display of Programmer class called

Whose details you want to use to see
1.Programmer
2.Manager2
Display of manager class called

Whose details you want to use to see
1.Programmer
2.Manager6
Wrong choice entered

PRACTICAL NO. : 02(8)

AIM: Write a program to implement multilevel inheritance from following figure. Accept and display data for one student.



CODE:

Result.cs

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

Test.cs

```
using System;
namespace multilevelinheritance
{
   class Test:student
   {
      int marks1, marks2;
      public Test(int roll_no, string name, int marks1, int marks2)
         : base(roll_no, name)
      {
         this.marks1 = marks1;
         this.marks2 = marks2;
```

```
    }
    public int getMarks1()
    {
       return marks1;
    }
    public int getMarks2()
    {
       return marks2;
    }
    public void dispaly()
    {
       base.display();
       Console.WriteLine("Marks1: " + marks1);
       Console.WriteLine("Marks2: " + marks2);
    } } }
```

Student.cs

```
using System;
namespace multilevelinheritance
{
   class student
   {
      int roll_no;
      string name;

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

Program.cs

```
using System;
namespace multilevelinheritance
{
   class Program
   {
      static void Main(string[] args)
      {
         Result r1 = new Result(101, "Prachit", 50, 70);
         r1.display();
      } } }
```

OUTPUT:

Roll no: 101
Name: Prachit
Marks1: 50
Marks2: 70
Total: 120

PRACTICAL NO. : 02(9)

AIM: Write a program to create a delegate called TrafficDel and a class called TrafficSignal with the following delegate methods.

Public static void Yellow()
{
Console.WriteLine("Yellow Light Signal To Get Ready");
}

Public static void Green()
{
Console.WriteLine("Green Light Signal To Go");
}

Public static void Red()
{
Console.WriteLine("Red Light Signal To Stop");
}

Also include a method IdentifySignal() to initialize an array of delegate with the above methods and a method show() to invoke members of the above array.

CODE:

TrafficSignal.cs

```
using System;
namespace TrafficDelegateExample
{
   public delegate void TrafficDel();
   class TrafficSignal
   {
      public static void Yellow()
      {
         Console.WriteLine("Yellow light signals to get ready");
      }
       public static void Green()
      {
         Console.WriteLine("Green light signals to go");
      }
       public static void Red()
      {
         Console.WriteLine("Red light signals to stop");
      }
       TrafficDel[] td = new TrafficDel[3];
       public void IdentifySignal()
       {
          td[0] = new TrafficDel(Yellow);
          td[1] = new TrafficDel(Green);

          td[2] = new TrafficDel(Red);
```

```
        }
        public void display()
        {
            td[0]();
            td[1]();
            td[2]();
        }
    } }
```

Program.cs

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

OUTPUT:

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

PRACTICAL NO. : 02(10)

AIM: Write a program to accept a number from the user and throw an exception if the number is not an even number.

CODE:

NotEvenException.cs

```
using System;
namespace ExceptionHandlingExample
{
   class NotEvenException:Exception
   {
      public NotEvenException(string msg)
         : base(msg)
      {

      }
} }
```

Program.cs

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

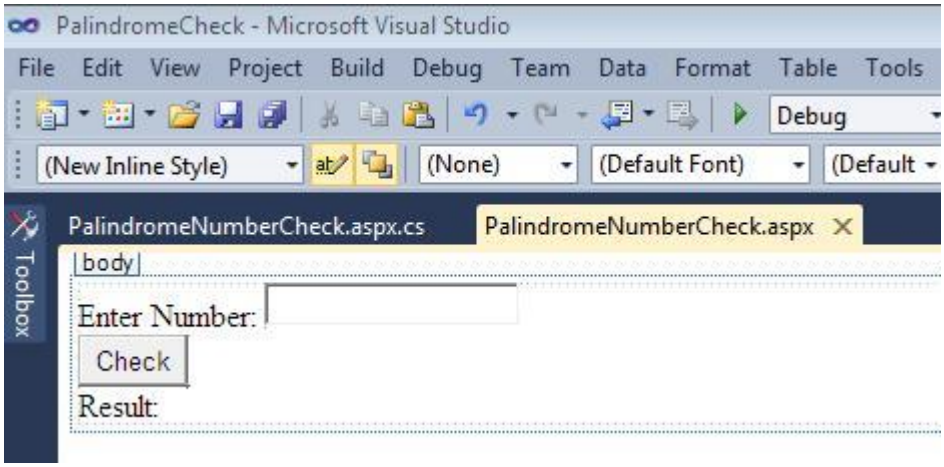
OUTPUT:

```
Enter a number: 5
Not an even number

Enter a number: 6
Its even number
```

PRACTICAL NO. : 03(1)

AIM: Create an application that allows the user to enter a number in the textbox named 'getnum'. Check whether the number in the textbox 'getnum' is palindrome or not. Print the message accordingly in the label control named lbldisplay when the user clicks on the button 'check'.

DESIGN:



PROPERTIES TABLE:

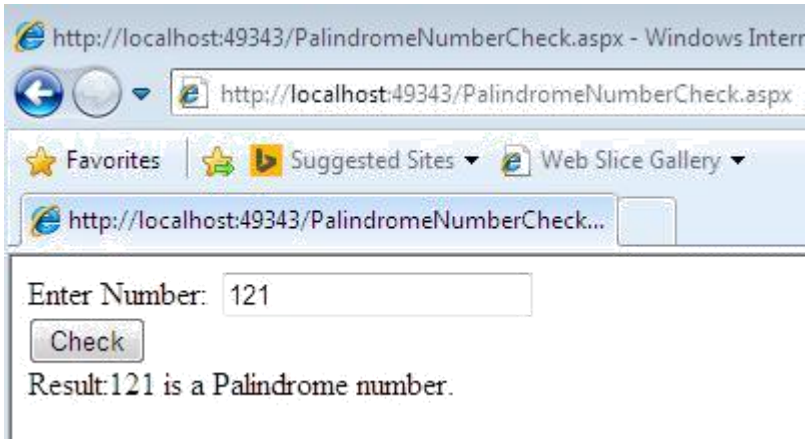
Control	Property	Value
Label1	Text	Enter Number
	ID	lblnum1
TextBox	ID	getNum
Button	Text	Check
	ID	btncheck
Label2	Text	Result
	ID	lblnum2

CODE:

```
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace PalindromeCheck
{
   public partial class PalindromeNumberCheck : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void btncheck_Click(object sender, EventArgs e)
      {
         int num = int.Parse(getNum.Text);
         int n, rev = 0, d;
```

```
        n = num;
        while (n > 0)
        {
            d = n % 10;
            n = n / 10;
            rev = rev * 10 + d;
        }
        if (rev == num)

          lblnum2.Text = lblnum2.Text + num + " is a Palindrome
        number."; else
            lblnum2.Text = lblnum2.Text + num + " is not a Palindrome number.";
    } } }
```

BROWSER OUTPUT:



PRACTICAL NO. : 03(2)

AIM: Create an application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and check boxes for selection , the user can make the label text bold ,underlined or italic and change its color . include buttons to display the message in the label, clear the text boxes and label and exit.

DESIGN:

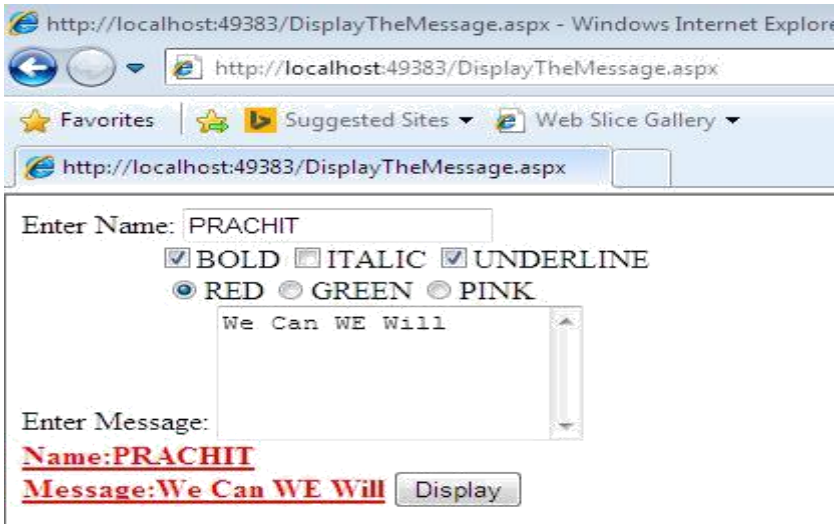
PROPERTIES TABLE:

Control	Property	Value
Label1	ID	lbl1
	Text	Enter Name
Checkbox1	ID	chkbold
	Text	BOLD
Checkbox2	ID	chkitalic
	Text	ITALIC
Checkbox3	ID	chkunderline
	Text	UNDERLINE
RadioButton1	ID	rbred
	Text	RED
RadioButton2	ID	rbgreen
	Text	GREEN
RadioButton3	ID	rbpink
	Text	PINK
Label2	ID	txtmessage
	Text	Enter Message
Button	ID	btndisplay
	Text	Display
Label3	ID	lblDisplay
	Text	Label3

CODE:

```
using System;
namespace DisplayMessage
{
   public partial class DisplayTheMessage : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void btndisplay_Click(object sender, EventArgs e)
      {
         if (chkbold.Checked == true)
           lblDisplay.Font.Bold = true;
         else
            lblDisplay.Font.Bold = false;

          if (chkitalic.Checked == true)
           lblDisplay.Font.Italic = true;
         else

            lblDisplay.Font.Italic = false;

         if (chkunderline.Checked == true)
           lblDisplay.Font.Underline = true;
         else
            lblDisplay.Font.Underline = false;
         if (rbred.Checked == true)
            lblDisplay.ForeColor = System.Drawing.Color.Red;
         else if(rbgreen.Checked == true)
            lblDisplay.ForeColor = System.Drawing.Color.Green;
         else if (rbpink.Checked == true)
            lblDisplay.ForeColor = System.Drawing.Color.Pink;
         lblDisplay.Text = "Name:" + txtName.Text + "<br/>" + "Message:" +
txtMessage.Text;
      } } }
```

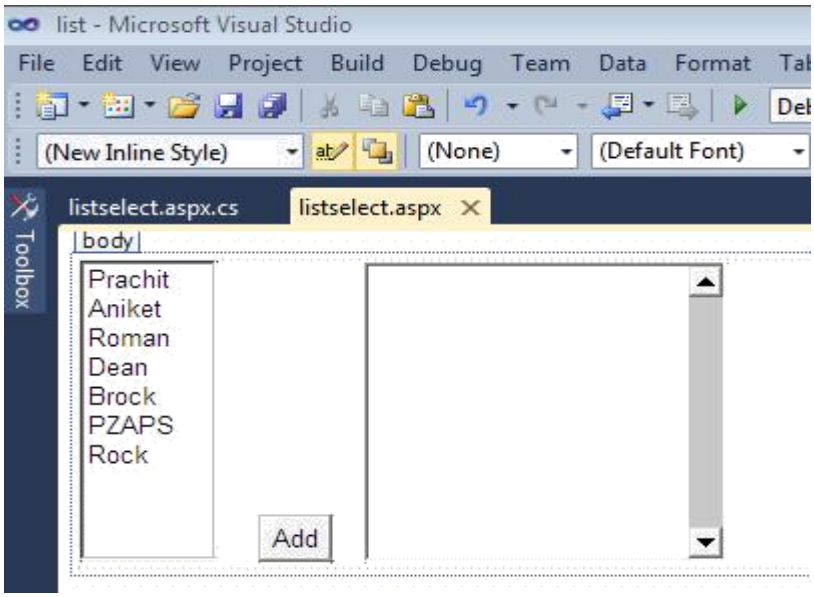
BROWSER OUTPUT:


http://localhost:49383/DisplayTheMessage.aspx - Windows Internet Explore
http://localhost:49383/DisplayTheMessage.aspx
Favorites
Suggested Sites
Web Slice Gallery
http://localhost:49383/DisplayTheMessage.aspx
Enter Name: PRACHIT
BOLD ITALIC UNDERLINE
RED GREEN PINK
We Can WE Will
Enter Message:
Name:PRACHIT
Message:We Can WE Will
Display

PRACTICAL NO. : 03(3)

AIM: List of employees is available in listbox. Write an application to add selected or all records from listbox (assume multi-line property of textbox is true).

DESIGN:

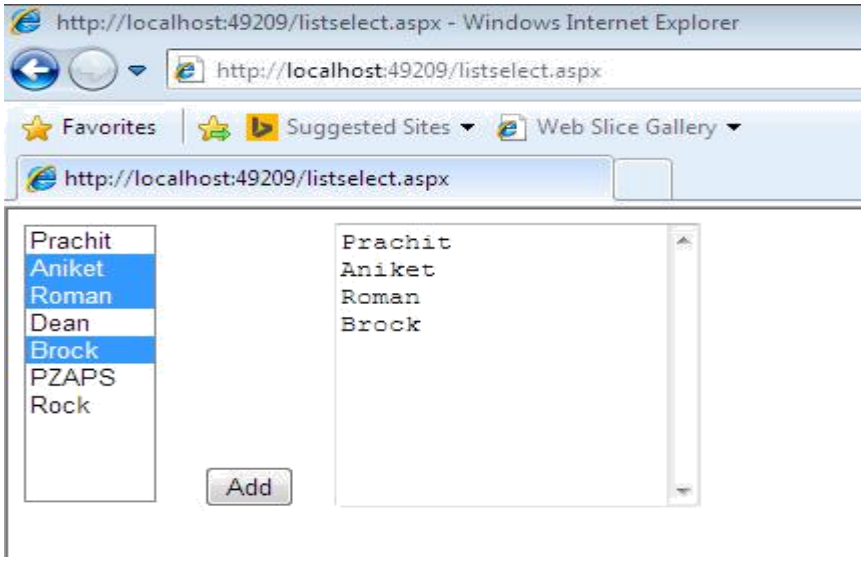


PROPERTIES TABLE:

CODE:

```
using System;
namespace list
{
   public partial class listselect : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void btnAdd_Click(object sender, EventArgs e)
      {
         int i;
         for (i = 0; i < lstEmployee.Items.Count; i++)
         {
            if (lstEmployee.Items[i].Selected == true) txtEmployee.Text
               += lstEmployee.Items[i].Text + "\n";
         }
       } } }
```

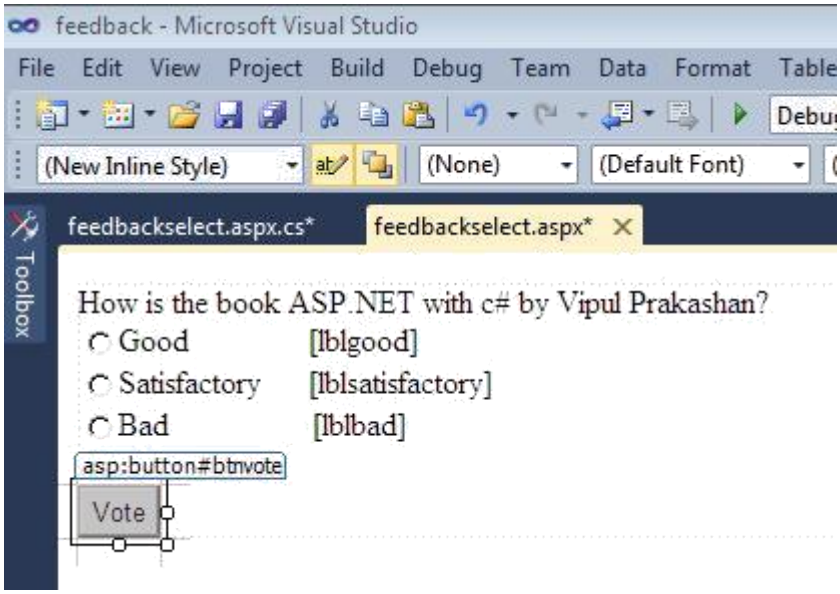
BROWSER OUTPUT:



PRACTICAL NO. : 03(4)

AIM: "How is the book ASP.NET with c# by Vipul Prakashan?" Give the user three choice : i)Good ii)Satisfactory iii)Bad. Provide a VOTE button. After user votes, present the result in percentage using labels next to the choices.

DESIGN:



PROPERTIES TABLE:

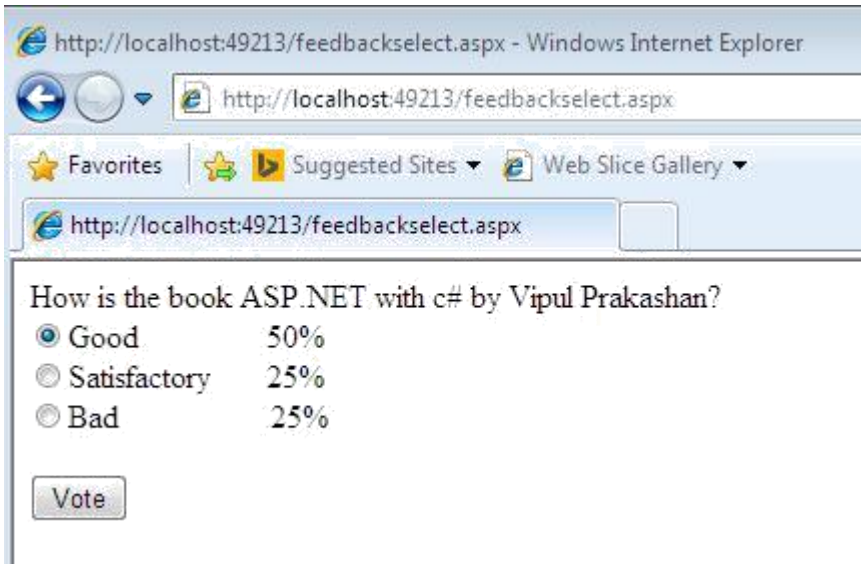
Control	Property	Value
Label1	ID	lbltxt1
	Text	How is the Book ASP.NET with c# Vipul Prakashan
RadioButton1	ID	rdogood
	Text	Good
RadioButton2	ID	rdosatisfactory
	Text	Satisfactory
RadioButton3	ID	rdobad
	Text	Bad
Label2	ID	lblgood
	Text	
Label3	ID	lblsatisfactory
	Text	
Label4	ID	lblbad
	Text	
Button	ID	btnvote
	Text	Vote

CODE:

```
using System;
namespace feedback
{
   public partial class feedbackselect : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
     protected void btnvote_Click(object sender, EventArgs e)
      {
         if (rdogood.Checked == true)
         {
            int goodCount;
            if (ViewState["gcount"] != null)
               goodCount = Convert.ToInt32(ViewState["gcount"]) + 1;
            else
               goodCount = 1;
            ViewState["gcount"] = goodCount;
         }


         if (rdosatisfactory.Checked == true)
         {
            int satisfactoryCount;
            if (ViewState["scount"] != null)
               satisfactoryCount = Convert.ToInt32(ViewState["scount"]) + 1;
            else
               satisfactoryCount = 1;
            ViewState["scount"] = satisfactoryCount;
         }
         if (rdobad.Checked == true)
         {
            int badCount;
            if (ViewState["bcount"] != null)
               badCount = Convert.ToInt32(ViewState["bcount"]) +
            1; else
               badCount = 1;
            ViewState["bcount"] = badCount;
         }
         int totalCount;
         if (ViewState["count"] != null)
            totalCount = Convert.ToInt32(ViewState["count"]) +
         1; else
            totalCount = 1;
         ViewState["count"] = totalCount;
         double gper = (Convert.ToDouble(ViewState["gcount"]) /
Convert.ToDouble(ViewState["count"])) * 100.0f;
```

```
 lblgood.Text = gper.ToString() + "%";
          double sper = (Convert.ToDouble(ViewState["scount"])
/ Convert.ToDouble(ViewState["count"])) * 100.0f;
          lblsatisfactory.Text = sper.ToString() + "%";
          double bper = (Convert.ToDouble(ViewState["bcount"]) /
Convert.ToDouble(ViewState["count"])) * 100.0f;
          lblbad.Text = bper.ToString()+"%";

       } } }
```

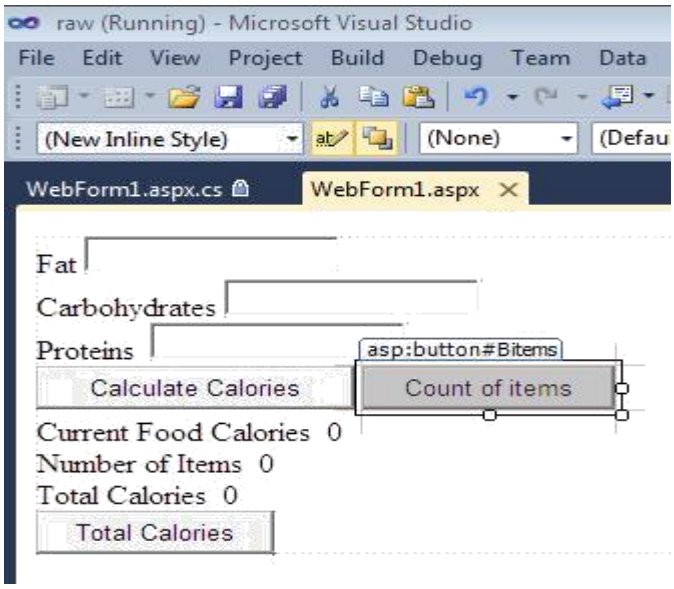
BROWSER OUTPUT:



PRACTICAL NO. : 03(5)

AIM: Create a project that calculates the total of fat, carbohydrate and protein. Allow the user to enter into text boxes. The grams of fat, grams of carbohydrate and grams of protein. Each gram of fat is 9 calories and protein or carbohydrate is 4 calories. Display the total calories of the current food item in a label. Use to other labels to display and accumulated some of calories and the count of items entered. The form food have 3 text boxes for the user to enter the grams for each category include label next to each text box indicating what the user is enter.

DESIGN:



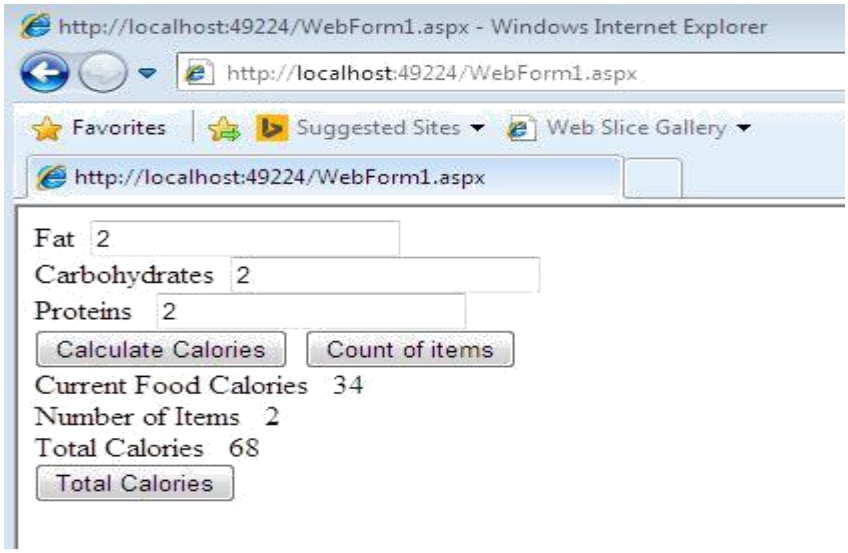
PROPERTIES TABLE:

CODE:

```
using System;
namespace raw
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      int curr_cal, total_cal, total_items;
      protected void Bcalories_Click(object sender, EventArgs e)
      {
         curr_cal = (Convert.ToInt32(txtfat.Text) * 9 + Convert.ToInt32(txtcarbo.Text) * 4 +
Convert.ToInt32(txtpro.Text) * 4);
         lblcfc.Text = Convert.ToString(curr_cal);
        lblnof.Text = Convert.ToString(total_cal);
```

```
        lbltc.Text = Convert.ToString(total_items);
    }
    protected void Bitems_Click(object sender, EventArgs e)
    {
        lblnof.Text = Convert.ToString(Convert.ToInt32(lblnof.Text) + 1);
    }
    protected void Btotalcalo_Click(object sender, EventArgs e)
    {
        lbltc.Text = Convert.ToString(Convert.ToInt32(lbltc.Text) +
Convert.ToInt32(lblcfc.Text));
    }
  } }
```

BROWSER OUTPUT:



PRACTICAL NO. : 04(1)

AIM: Set the label border color of rollno to red using css.

DESIGN:

PROPERTY TABLE :

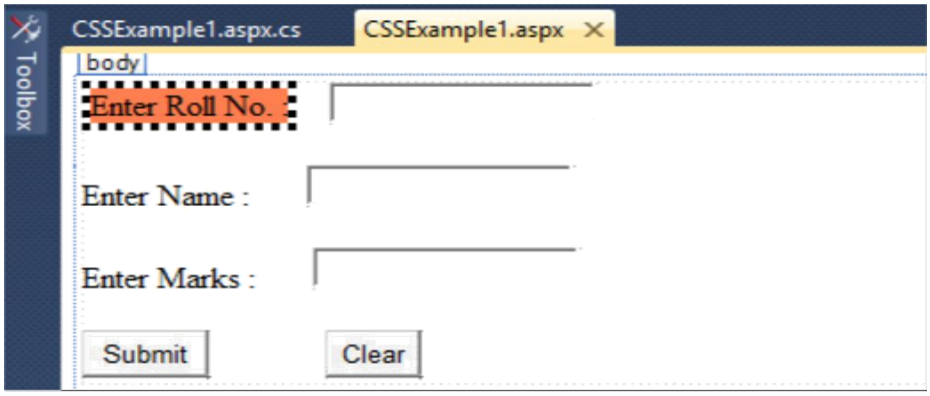
Control	Property	Value
Label1	ID	lblRollNo
Label1	Text	Enter Roll No.
Label1	BorderStyle	Dotted
Label1	BackColor	Coral
Label2	ID	lblName
Label2	Text	Enter Name
Label3	ID	lblMarks
Label3	Text	Enter Marks
TextBox1	ID	txtRollNo
TextBox2	ID	txtName
TextBox3	ID	txtMarks
Button1	ID	btnSubmit
Button1	Text	Submit

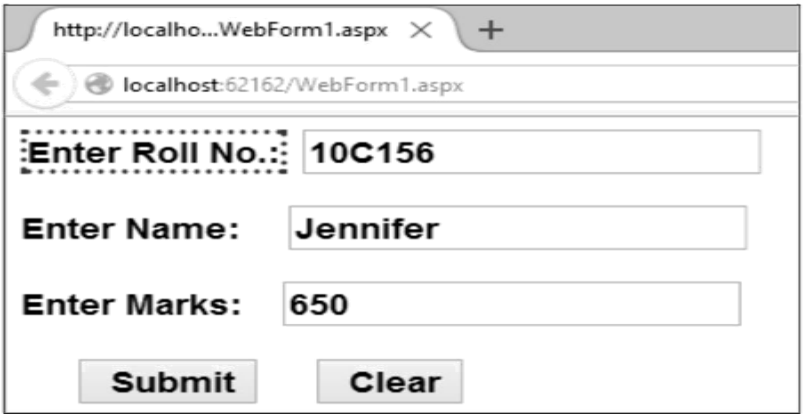
CODE:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="cssexample.aspx.cs" Inherits="practical4css.cssexample" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
```

```
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <asp:Label ID="Label1" runat="server" Text="Enter Roll No.:"
BorderStyle="Dotted" BackColor="Coral"></asp:Label>
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <br />
      <asp:Label ID="Label2" runat="server" Text="Enter
      Name:"></asp:Label> <asp:TextBox ID="TextBox2"
      runat="server"></asp:TextBox> <br />
      <asp:Label ID="Label3" runat="server" Text="Enter
      Marks:"></asp:Label> <asp:TextBox ID="TextBox3"
      runat="server"></asp:TextBox> <br />
      <br />
      <asp:Button ID="Button1" runat="server" Text="Submit" />
           
      <asp:Button ID="Button2" runat="server" Text="Clear" />
   </div>
   </form>
</body>
</html>
```

BROWSER OUTPUT:

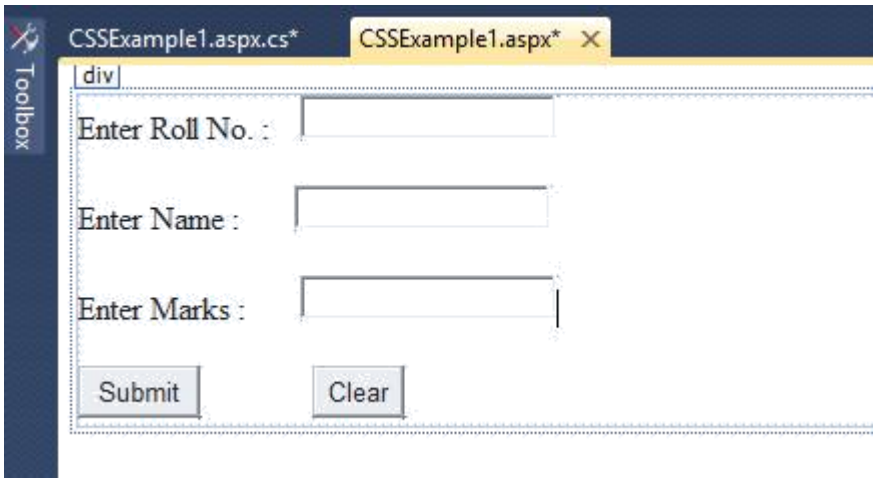




PRACTICAL NO. : 04(2)

AIM: Set the font-Arial , font style-bond , font size-18px of different controls(ie. Label, textbox, button) using css.

DESIGN:



PROPERTY TABLE :

Control	Property	Value
Label1	ID	lblRollNo
Label1	Text	Enter Roll No.
Label1	BorderStyle	Dotted
Label1	BackColor	Coral
Label2	ID	lblName
Label2	Text	Enter Name
Label2	CssClass	Common
Label3	ID	lblMarks
Label3	Text	Enter Marks
Label3	CssClass	Common
TextBox1	ID	txtRollNo
TextBox1	CssClass	Txt Style
TextBox2	ID	txtName
TextBox2	CssClass	Txt Style
TextBox3	ID	txtMarks
TextBox3	CssClass	Txt Style
Button1	ID	btnSubmit
Button1	Text	Submit
Button1	CssClass	btnStyle
Button2	ID	btnClear
Button2	Text	Clear
Button2	CssClass	btnStyle

CODE:

Myformat.css

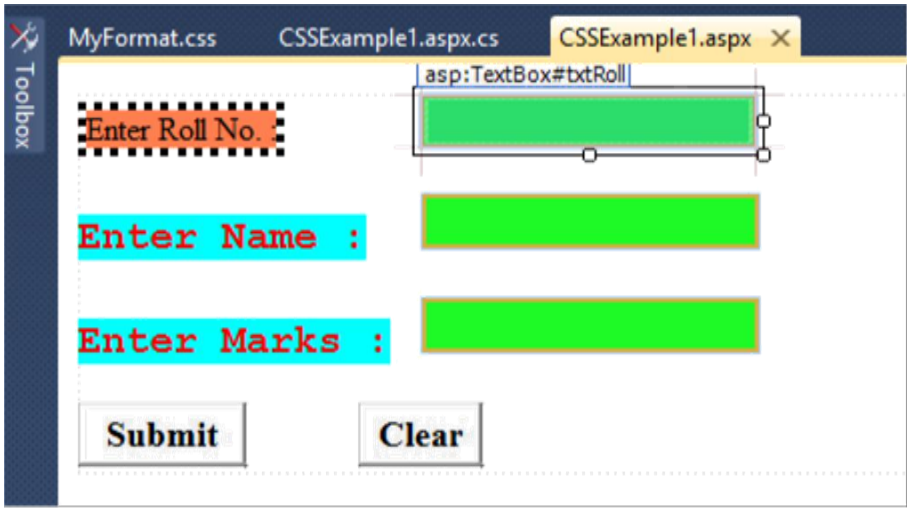
```
.BtnStyle
{
font-family:Times New Roman;
font-size:large;
font-weight:bold;
}
.TxtStyle
{
font-family:Georgia;
font-size:larger;
font-weight:400;
background-color:Maroon;
border:2px solid goldenrod;
}
.Common
{
background-color:Aqua;
color:Red;
font-family:Courier New;
font-size:20px;
font-weight:bolder;
}
```

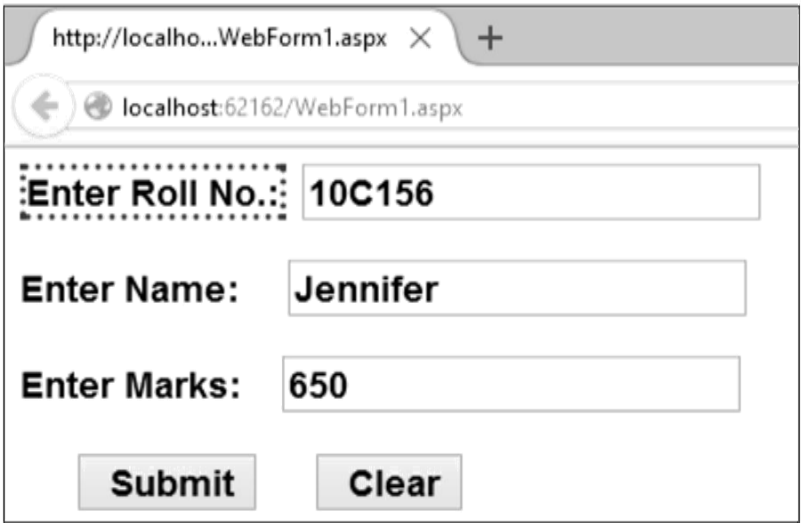
Myformatting.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="cssexample.aspx.cs" Inherits="practical4css.cssexample" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Enter Roll No.:" BorderStyle="Dotted"
BackColor="Coral"></asp:Label>
<asp:TextBox ID="TextBox1" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br />
<asp:Label ID="Label2" runat="server" Text="Enter Name:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox2" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br />
```

```
<asp:Label ID="Label3" runat="server" Text="Enter Marks:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox3" runat="server" CssClass="TxtStyle"></asp:TextBox>
<br />
<br />
<asp:Button ID="Button1" runat="server" Text="Submit" CssClass="BtnStyle" />
<asp:Button ID="Button2" runat="server" Text="Clear" CssClass="BtnStyle" />
</div>
</form>
</body>
</html>
```

BROWSER OUTPUT:





PRACTICAL NO. : 04(3)

AIM: Design the same webpages for BMS, BAF, BscIT students and apply same background color for all the pages using css.

PROPERTY TABLE :

Control	Property	Value
Label1	ID	lblBScIT
Label1	Text	Welcome to BScIT
Label1	CssClass	bk

Control	Property	Value
Label1	ID	lblBAF
Label1	Text	Welcome to BMS
Label1	CssClass	bk

Control	Property	Value
Label1	ID	lblBMS
Label1	Text	Welcome to BAF
Label1	CssClass	bk

CODE:

Myformat.css

```
.BtnStyle
{
font-family:Times New Roman;
font-size:large;
font-weight:bold;
}
.TxtStyle
{
font-family:Georgia;
font-size:larger;
font-weight:400;
background-color:Lime;
border:2px solid goldenrod;
}
.Common
{
```

```
background-color:Aqua;
color:Red;
font-family:Courier New;
font-size:20px;
font-weight:bolder;
}
.bk
{
background-color:Lime;
}
```

BScIT.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="BScIT.aspx.cs" Inherits="cssExample.BScIT" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <link rel="Stylesheet" type="text/css" href="MyFormat.css" />
</head>
<body text="Welcome to BScIT">
   <form id="form1" runat="server">
   <div class="bk">
      <asp:Label ID="lblBScIT" runat="server" Text="Welcome to BscIT"></asp:Label>
   </div>
   </form>
</body>
</html>
```

BAF.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="BAF.aspx.cs" Inherits="cssExample.BAF" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <link rel="Stylesheet" type="text/css" href="MyFormat.css" />
</head>
<body>
   <form id="form1" runat="server">
   <div class="bk">
      <asp:Label ID="lblBAF" runat="server" Text="Welcome to BAF"></asp:Label>
   </div>
   </form>

</body>
```

```
</html>
```

BMS.aspx

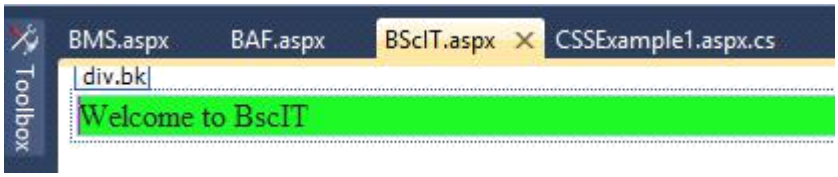
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="BMS.aspx.cs" Inherits="cssExample.BMS" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <link rel="Stylesheet" type="text/css" href="MyFormat.css"
/> </head>
<body>
   <form id="form1" runat="server" class="bk">
   <asp:Label ID="lblBMS" runat="server" Text="Welcome to BMS"></asp:Label>
   </form>
</body>
</html>
```

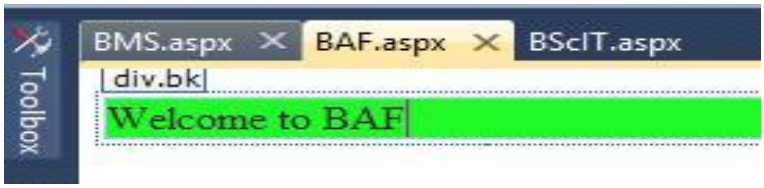
CSSExample1.aspx:

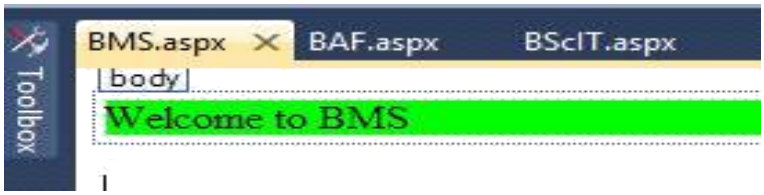
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="CSSExample1.aspx.cs"
Inherits="cssExample.CSSExample1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <link rel="Stylesheet" type="text/css" href="MyFormat.css"
/> </head>
<body>
   <form id="form1" runat="server">
   <div>
      <asp:Label ID="lblRollNo" runat="server" Text="Enter Roll No. :" BorderStyle="Dotted"
BackColor="Coral"></asp:Label>
           &nbs
p;   
      <asp:TextBox ID="txtRoll" runat="server" CssClass="TxtStyle"></asp:TextBox>
      <br />
      <br />
      <asp:Label ID="lblName" runat="server" Text="Enter Name
:" CssClass="Common"></asp:Label>
      
      <asp:TextBox ID="txtName" runat="server"
      CssClass="TxtStyle"></asp:TextBox> <br />

      <br />
```

```
        <asp:Label ID="lblMarks" runat="server" Text="Enter Marks :"
CssClass="Common"></asp:Label>
           
        <asp:TextBox ID="txtMarks" runat="server" CssClass="TxtStyle"></asp:TextBox>
        <br />
        <br />
        <asp:Button ID="btnSubmit" runat="server" onclick="btnSubmit_Click"
           Text="Submit" CssClass="BtnStyle" />
             &nbs
p;
        <asp:Button ID="btnClear" runat="server" Text="Clear" CssClass="BtnStyle"/>
        <br>
        <br>
        <br>
    <h1><a href="BScIT.aspx"</a>Bsc IT</h1>
    <h2><a href ="BAF.aspx"</a>BAF</h2>
    <h3><a href ="BMS.aspx"</a>BMS</h3> <a
    href="http://www.vsit.edu.in/"> Contact
    us</a>
    </div>
    </form>
</body>
</html>
```

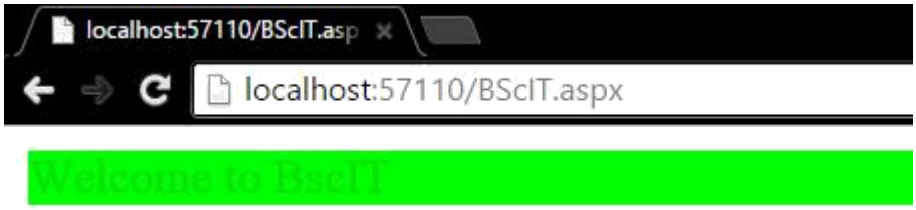
OUTPUT:

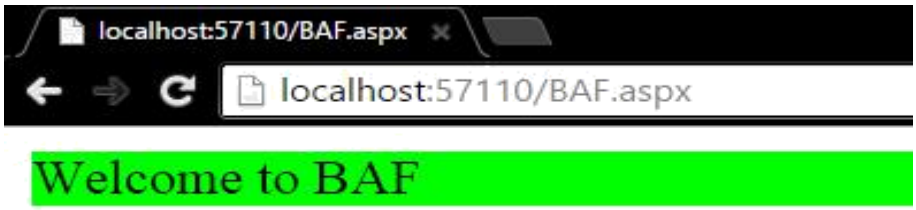

Toolbox
BMS.aspx
BAF.aspx
BScIT.aspx
CSSExample1.aspx.cs
div.bk
Welcome to BscIT

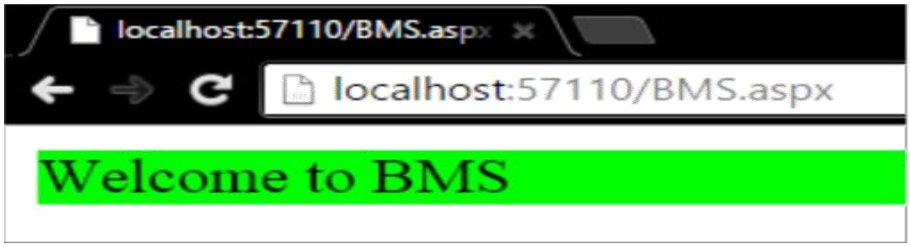

Toolbox
BMS.aspx
BAF.aspx
BScIT.aspx
div.bk
Welcome to BAF


Toolbox
BMS.aspx
BAF.aspx
BScIT.aspx
body
Welcome to BMS


http://localho...WebForm1.aspx
localhost:62162/WebForm1.aspx
Enter Roll No.:
Enter Name:
Enter Marks:
Submit
Clear
BSC IT
BAF
BMS
Contact us


localhost:57110/BScIT.asp
localhost:57110/BScIT.aspx
Welcome to BscIT


localhost:57110/BAF.aspx
localhost:57110/BAF.aspx
Welcome to BAF


localhost:57110/BMS.aspx
localhost:57110/BMS.aspx
Welcome to BMS


Vidyalankar School of Information Technology, ...
Home | Careers @ VSIT | Contact Us | WebMail
Follow us on ..
Login Text Size
VSIT
Vidyalankar School of Information Technology
Affiliated to Mumbai University | Accredited by NAAC
About VSIT
Prospective Students
Academics Courses
Committees
Faculty
Infrastructure

PRACTICAL NO. : 04(4)

AIM: Change the font family and color of all heading of above webpage using css.

DESIGN:

CODE:
myformating.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="myformatting.aspx.cs"
Inherits="WebApplication1.myformatting" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link rel="Stylesheet" type="text/css" href="MyFormat.css"
/> <style type="text/css">
h1,h2,h3{color:Blue; font-family:Agency FB;}
</style>
</head>
```

```
<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Enter Roll No.:" BorderStyle="Dotted"
BackColor="Coral"></asp:Label>
<asp:TextBox ID="TextBox1" runat="server" CssClass="TxtStyle"></asp:TextBox>

<br />
<asp:Label ID="Label2" runat="server" Text="Enter Name:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox2" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br />
<asp:Label ID="Label3" runat="server" Text="Enter Marks:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox3" runat="server" CssClass="TxtStyle"></asp:TextBox>
<br />
<br />

<asp:Button ID="Button1" runat="server" Text="Submit" CssClass="BtnStyle"
/> <asp:Button ID="Button2" runat="server" Text="Clear" CssClass="BtnStyle"
/> <h1><a href="bscit.aspx"</a>Bsc IT</h1> <h2><a href
="baf.aspx"</a>BAF</h2>
<h3><a href ="bms.aspx"</a>BMS</h3>
<a href="http://www.vsit.edu.in/">
Contact us</a>
<br />
<br />
<br />
<br />
</div>
</form>
</body>
</html>
```

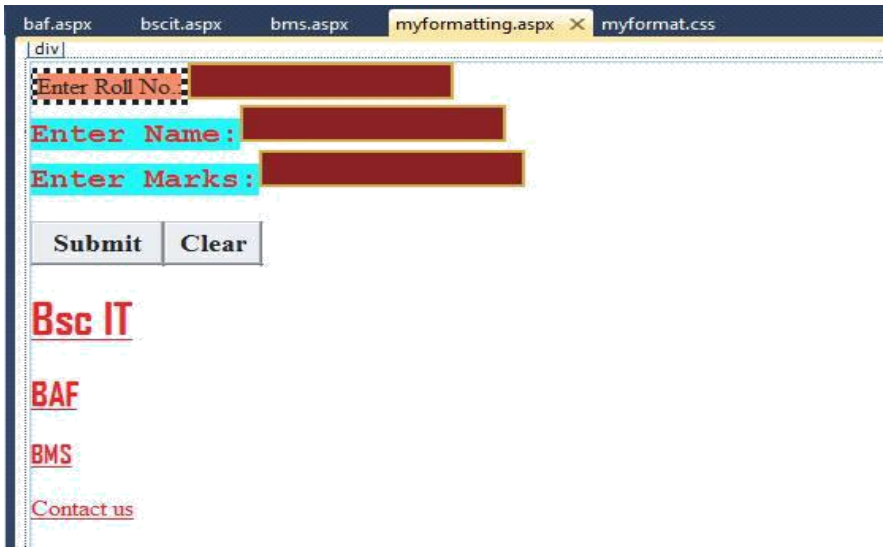
BROWSER OUTPUT:


http://localho...WebForm1.aspx
localhost:62162/WebForm1.aspx
Enter Roll No.:
Enter Name:
Enter Marks:
Submit
Clear
BSC IT
BAF
BMS
Contact us

PRACTICAL NO. : 04(5)

AIM: Use pseudo classes and display link, visited link and active link of contact us differently.

DESIGN:



CODE:

myformatting.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="myformatting.aspx.cs"
Inherits="WebApplication1.myformatting" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<link rel="Stylesheet" type="text/css" href="MyFormat.css"
/> <style type="text/css">
h1,h2,h3{color:Blue; font-family:Agency FB;}
A:link{color:Red;}
A:visited{color:Green;}
A:active{color:Orange;}
</style>
</head>

<body>
<form id="form1" runat="server">
<div>
<asp:Label ID="Label1" runat="server" Text="Enter Roll No.:" BorderStyle="Dotted"
BackColor="Coral"></asp:Label>
```

```
<asp:TextBox ID="TextBox1" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br />
<asp:Label ID="Label2" runat="server" Text="Enter Name:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox2" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br />
<asp:Label ID="Label3" runat="server" Text="Enter Marks:"
CssClass="Common"></asp:Label>
<asp:TextBox ID="TextBox3" runat="server"
CssClass="TxtStyle"></asp:TextBox> <br /><br />
<asp:Button ID="Button1" runat="server" Text="Submit" CssClass="BtnStyle"
/> <asp:Button ID="Button2" runat="server" Text="Clear" CssClass="BtnStyle"
/> <h1><a href="bscit.aspx"</a>Bsc IT</h1> <h2><a href
="baf.aspx"</a>BAF</h2>
<h3><a href ="bms.aspx"</a>BMS</h3>
<a href="http://www.vsit.edu.in/">
Contact us</a>
<br /><br /><br /><br />
</div>
</form>
</body>
</html>
```

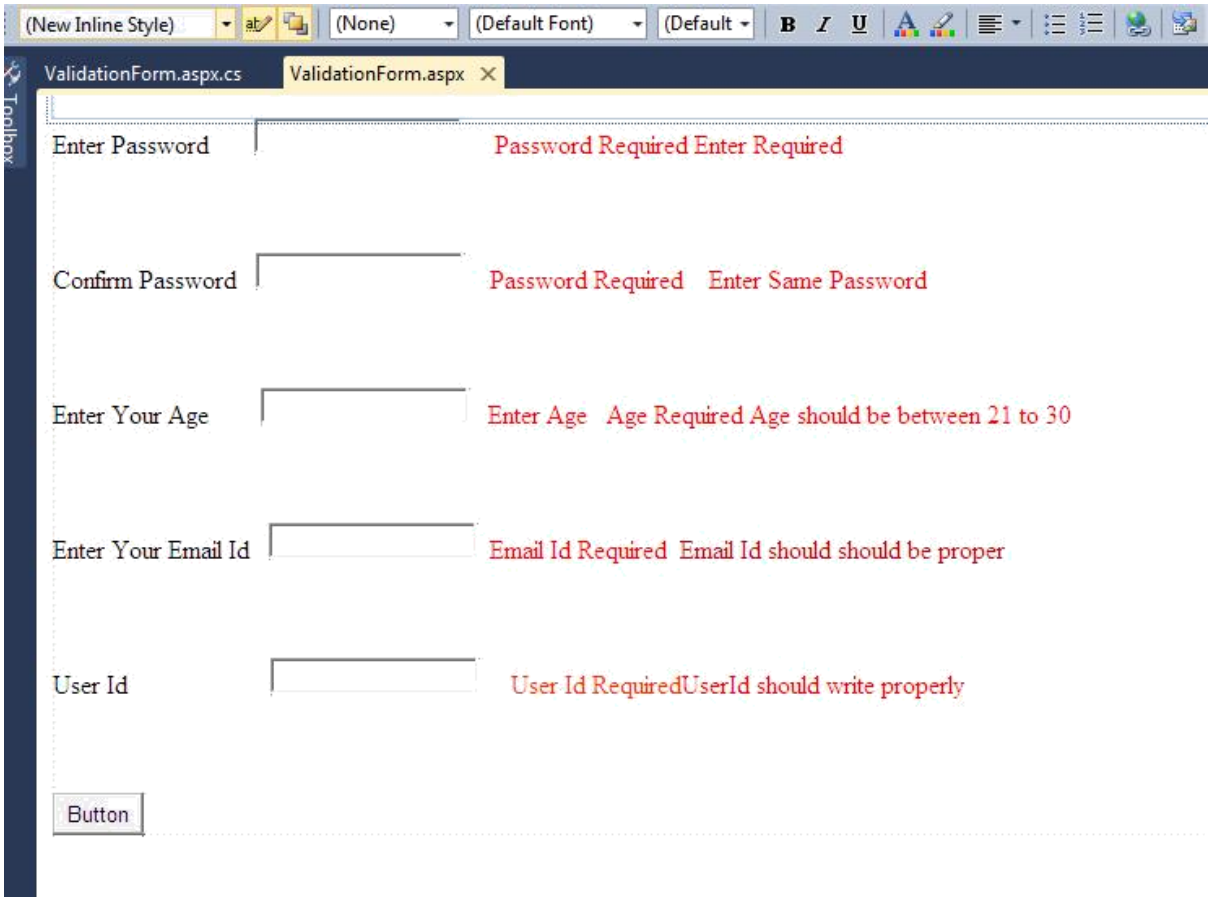
BROWSER OUTPUT:

PRACTICAL NO. : 05(1)

AIM: Programs using ASP.NET Server controls.

Create the application that accepts name, password ,age , email id, and user id. Allthe information entry is compulsory. Password should be reconfirmed. Age should be within 21 to 30. Email id should be valid. User id should have at least a capital letter and digit as well as length should be between 7 and 20 characters.

DESIGN:



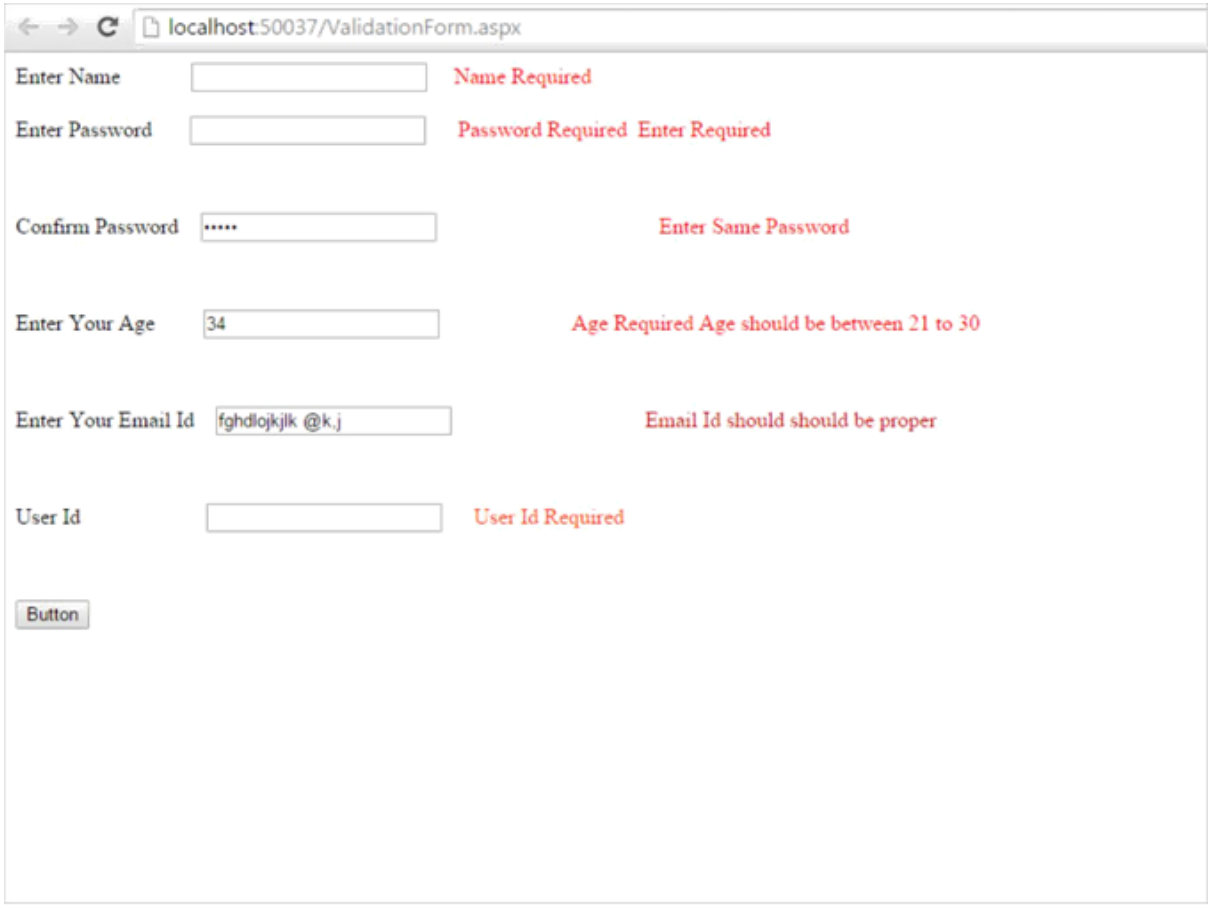
CODE:

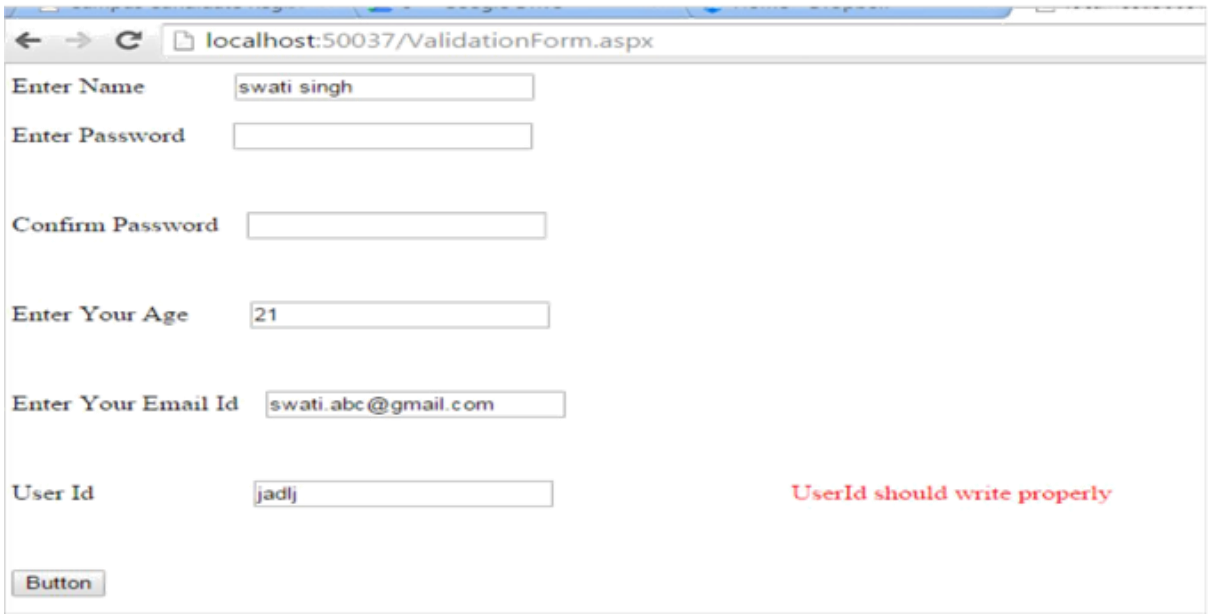
ValidateControlForm.aspx

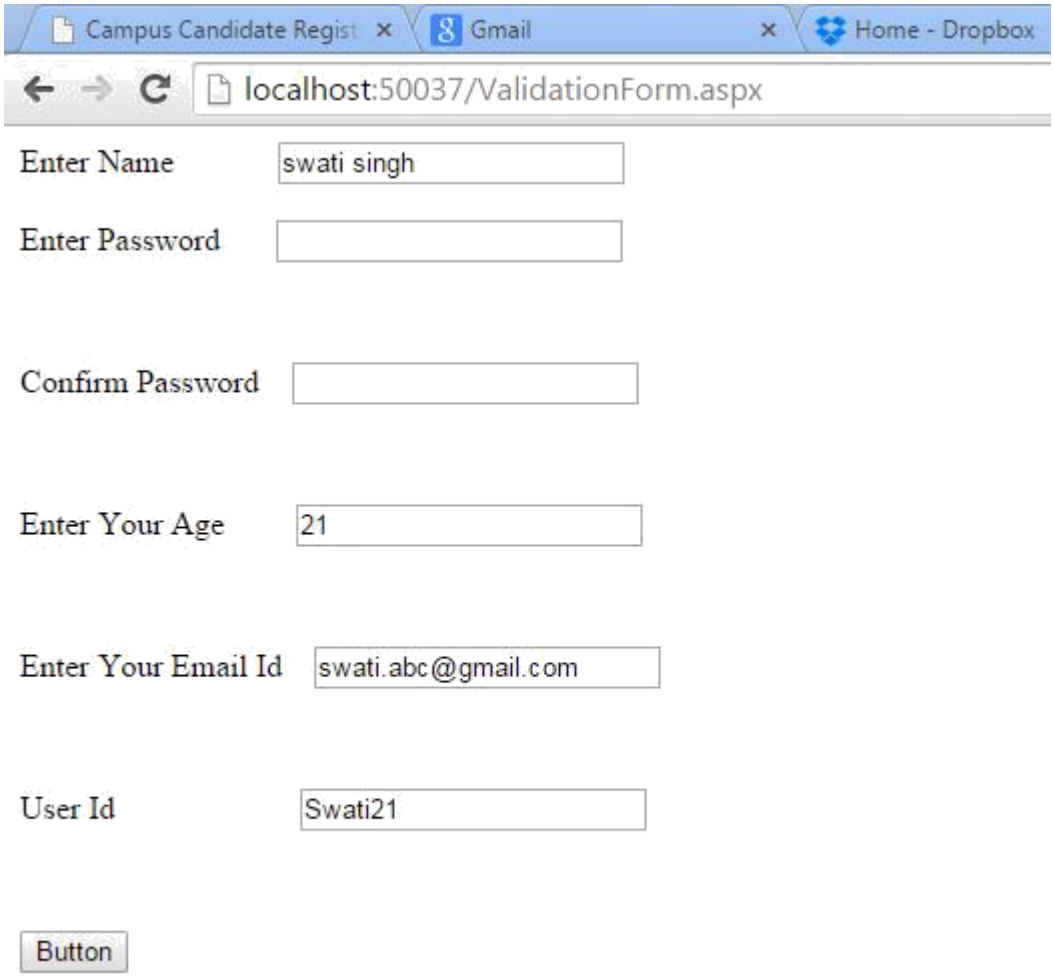
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace ValidationControl
{
   public partial class ValidationControlForm : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs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

```
    }
    protected void CustomValidator1_ServerValidate(object
source, ServerValidateEventArgs args)
    {
        string str = args.Value;
        args.IsValid = false;
        if (str.Length < 7 || str.Length > 20)
        {

            return;
        }
        bool capital = false;
        foreach (char ch in str)
        {
            if (ch >= 'A' && ch <= 'Z')
            {
                capital = true;
                break;
            }
        }
        if (!capital)
            return;
        bool digit = false;
        foreach (char ch in str)
        {
            if (ch >= '0' && ch <= '9')
            {
                digit = true;
                break;
            }
        }
        if (!digit)
            return;
        args.IsValid = true;
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
    }
    } }
```

OUTPUT:


localhost:50037/ValidationForm.aspx
Enter Name
Name Required
Enter Password
Password Required Enter Required
Confirm Password
Enter Same Password
Enter Your Age
34
Age Required Age should be between 21 to 30
Enter Your Email Id
fghdlojkjlk @k.j
Email Id should should be proper
User Id
User Id Required
Button

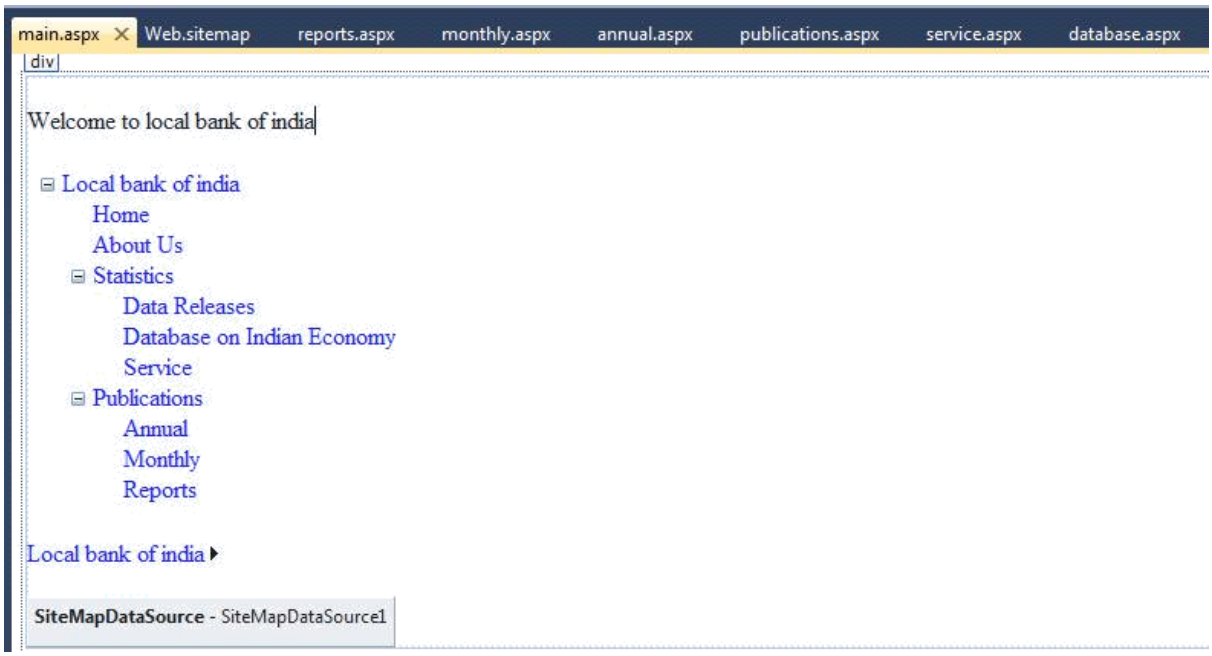

localhost:50037/ValidationForm.aspx
Enter Name
swati singh
Enter Password
Confirm Password
Enter Your Age
21
Enter Your Email Id
swati.abc@gmail.com
User Id
jadlj
UserId should write properly
Button


Campus Candidate Regist
Gmail
Home - Dropbox
localhost:50037/ValidationForm.aspx
Enter Name
swati singh
Enter Password
Confirm Password
Enter Your Age
21
Enter Your Email Id
swati.abc@gmail.com
User Id
Swati21
Button

PRACTICAL NO. : 05(2)

AIM: Programs using ASP.NET Server controls.
Create a website for a bank and include types of navigation.

DESIGN:

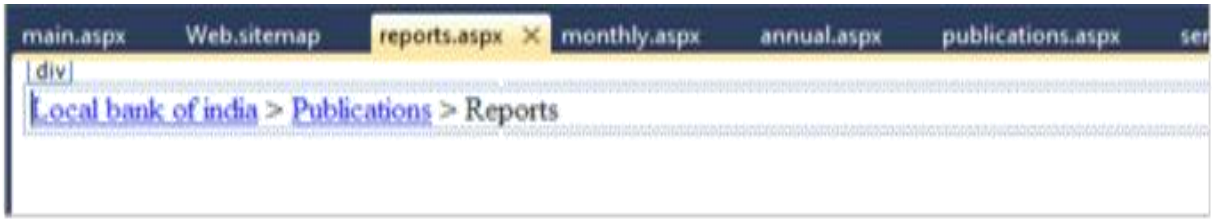


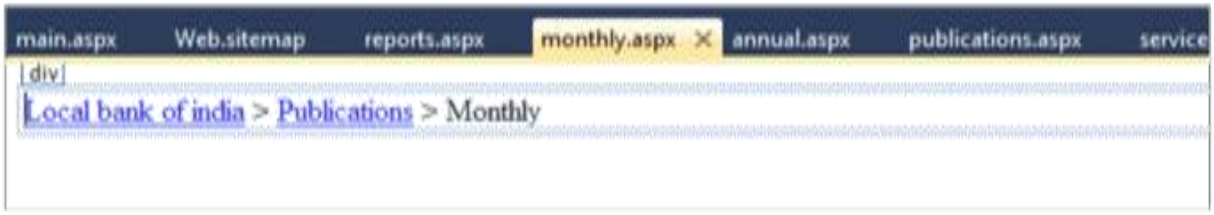
CODE:

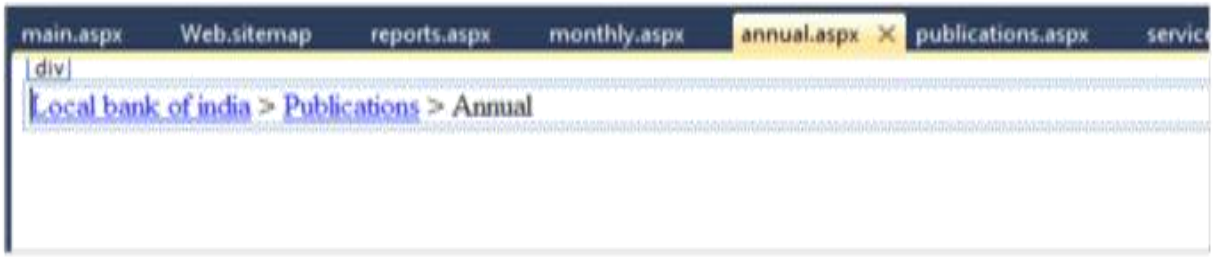
Web.sitemap

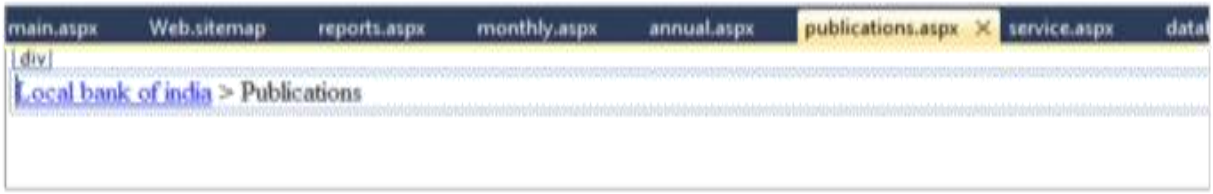
```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
   <siteMapNode url="~\" title="Local bank of india" description="Online Banking">
     <siteMapNode url="default.aspx" title="Home" description="Go to the homepage"
     /> <siteMapNode url="about.aspx" title="About Us" description="About us"/>
     <siteMapNode url="statistics.aspx" title="Statistics" description="Statistics">
       <siteMapNode url="data.aspx" title="Data Releases" description="Data Releases"/>
       <siteMapNode url="database.aspx" title="Database on Indian Economy"
description="Economy of India"/>
       <siteMapNode url="service.aspx" title="Service" description="Service Information"/>
       </siteMapNode>
       <siteMapNode url="publications.aspx" title="Publications"
         description="Publications"> <siteMapNode url="annual.aspx" title="Annual"
         description="Annual"/> <siteMapNode url="monthly.aspx" title="Monthly"
         description="Monthly"/> <siteMapNode url="reports.aspx" title="Reports"
         description="Reports"/> </siteMapNode>
         </siteMapNode>
</siteMap>
```

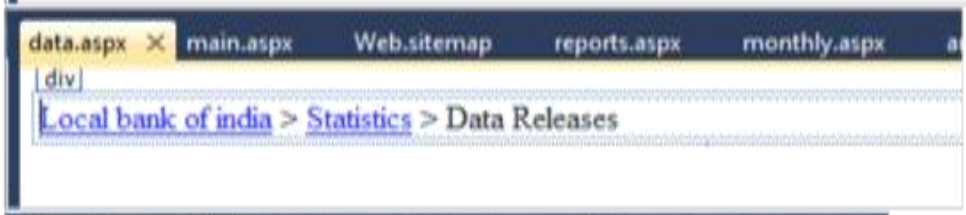
OUTPUT: (sitemap)

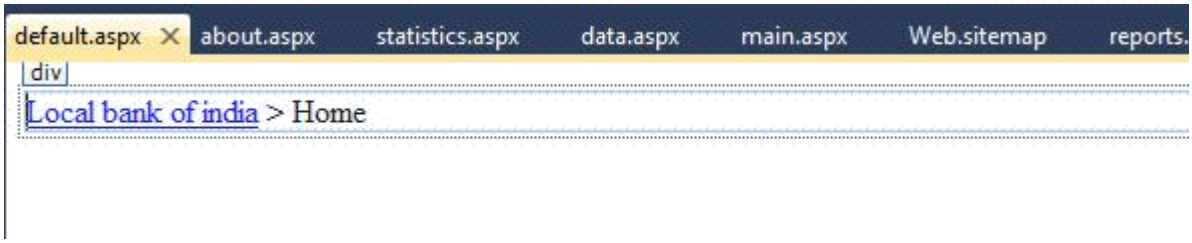












<u>OUTPUT:</u> (Website form Tree view Controls)

PRACTICAL NO. : 06(1)

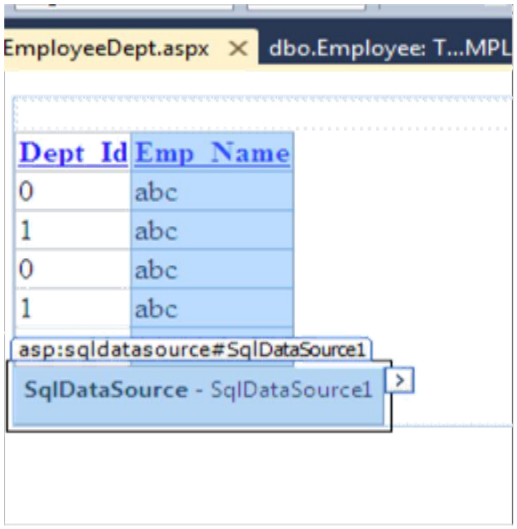
AIM: Database programs with ASP.NET and ADO.NET.

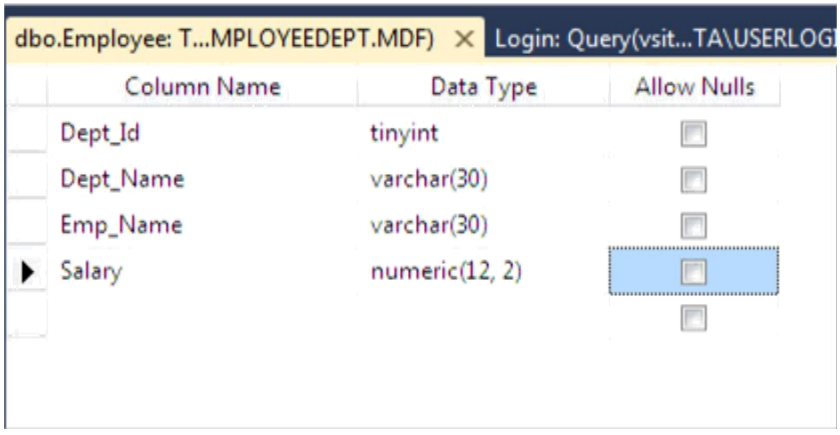
Create a Web App to display all the Empname and Deptid of the employee from the database using SQL source control and bind it to GridView . Database fields are(DeptId, DeptName, EmpName, Salary).

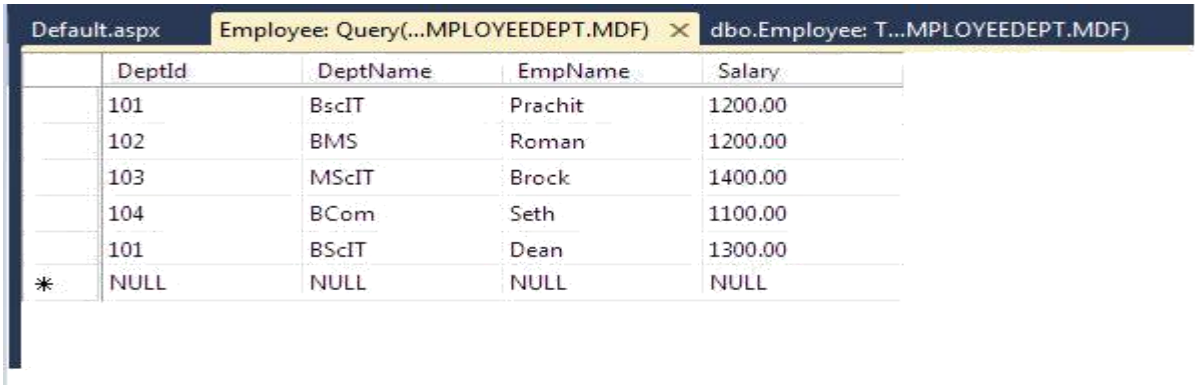
Steps:

1. File → new → website → empty website → name it → ok
2. Right click on website made → add new item → sql server database → name it → add → yes
3. Right click on table In server explorer → add new table → add columns → save the table
4. Right click on table made → show table data → add values
5. Right click on website → add new item → webform → name it
6. Go to design view
7. Add a gridview → below that add sqldatasource
8. Configure sqldatasource → then add it to the gridview
9. Go to gridview menu enable sorting

DESIGN:







Default.aspx | Employee: Query(...MPLOYEEDEPT.MDF) | dbo.Employee: T...MPLOYEEDEPT.MDF)

	DeptId	DeptName	EmpName	Salary
	101	BscIT	Prachit	1200.00
	102	BMS	Roman	1200.00
	103	MScIT	Brock	1400.00
	104	BCom	Seth	1100.00
	101	BScIT	Dean	1300.00
*	NULL	NULL	NULL	NULL

OUTPUT:

Deptid	Empname
1	swati
2	natasha
3	thor
4	max
5	mahi

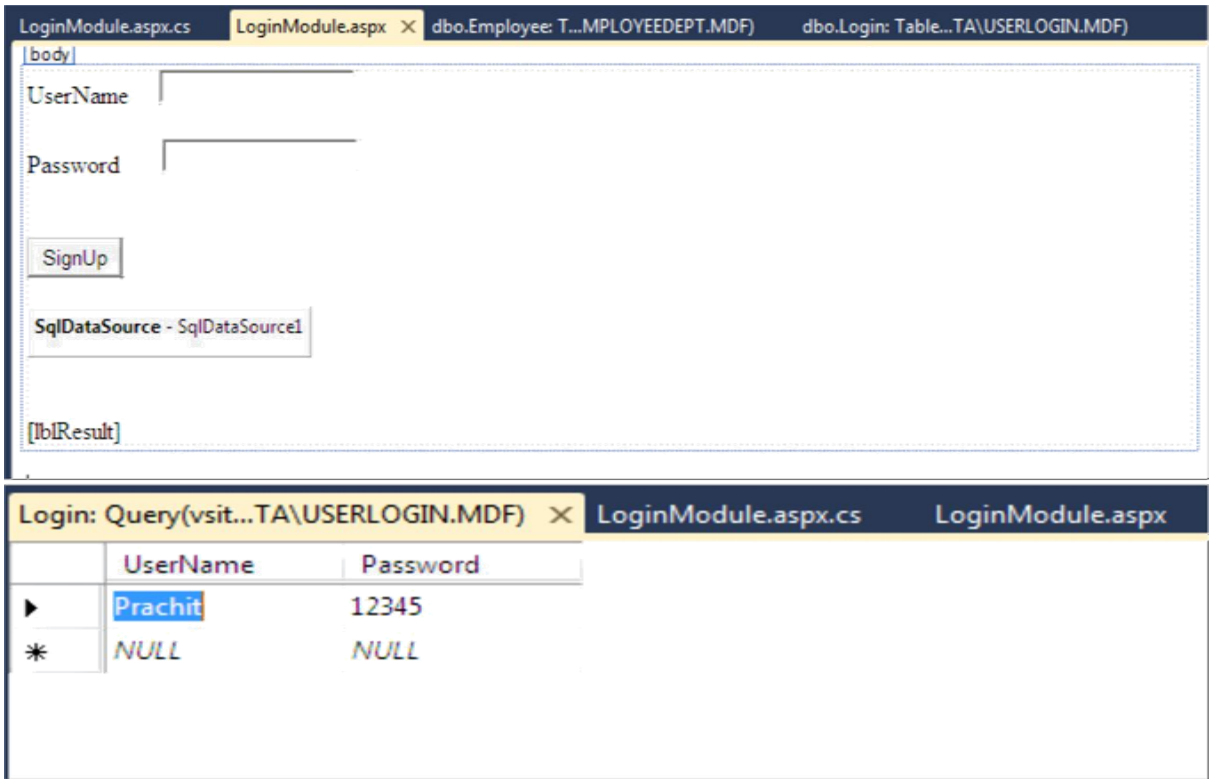
PRACTICAL NO. : 06(2)

AIM: Database programs with ASP.NET and ADO.NET
Create a Login Module which adds Username and Password in the database. Username in the database should be a primary key.

Steps2:

1. File → new → website → empty website → name it → ok
2. Right click on website made → add new item → sql server database → name it → add → yes
3. Right click on table In server explorer → add new table → add columns → save the table
4. Right click on table made → show table data → add values
5. Right click on website → add new item → webform → name it
6. Go to design view → add form for login
7. Add sqldatasource → configure it
8. Write code

DESIGN:

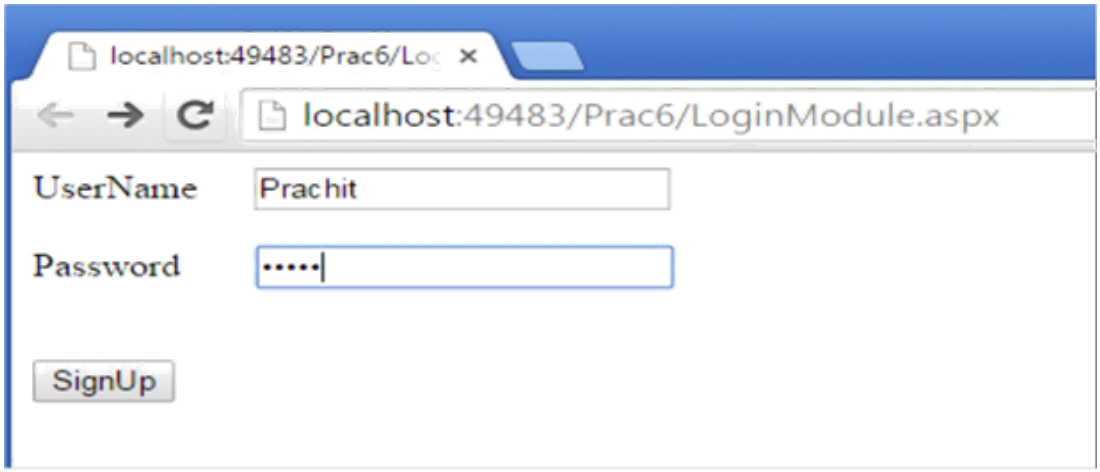


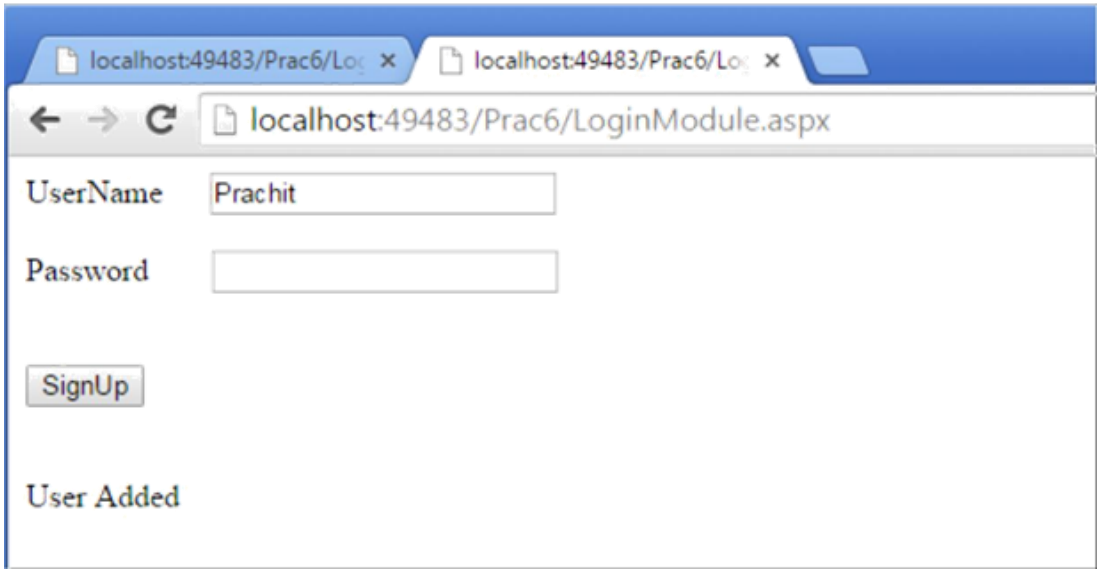
CODE:
LoginModule.aspx

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

using System.Web;
```

```
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class LoginModule : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
   }
   protected void btnSignUp_Click(object sender, EventArgs e)
   {
      SqlDataSource1.InsertParameters["Username"].DefaultValue = txtUserName.Text;
      SqlDataSource1.InsertParameters["Password"].DefaultValue = txtPassword.Text;
         SqlDataSource1.Insert();
      lblResult.Text = "User Added";
   }
}
```

OUTPUT:





PRACTICAL NO. : 06(3)

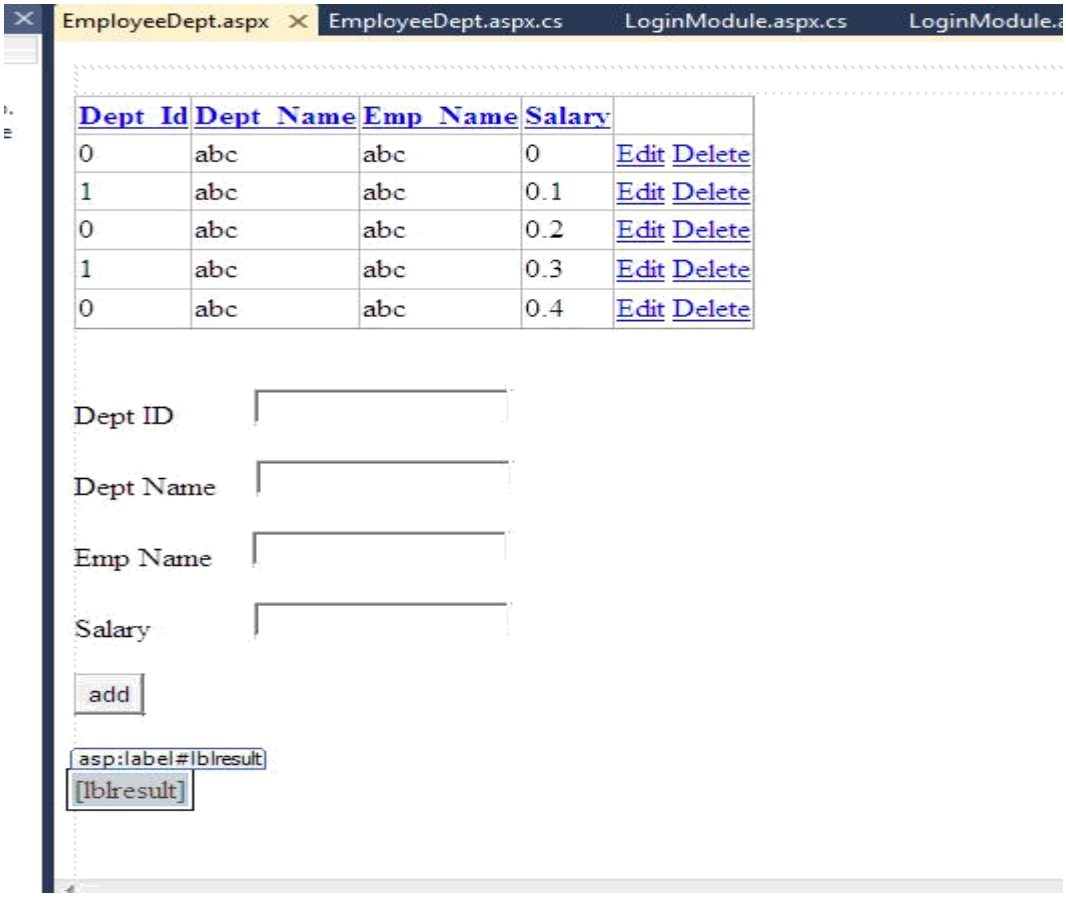
AIM: Database programs with ASP.NET and ADO.NET

Create a web application to insert 3 records inside the SQL database table having following fields(DeptId, DeptName, EmpName, Salary). Update the salary for any one employee and increment it to 15% of the present salary. Perform delete operation on 1 row of the database table.

Steps:

9. File → new → website → empty website → name it → ok
10. Right click on website made → add new item → sql server database → name it → add → yes
11. Right click on table In server explorer → add new table → add columns → save the table
12. Right click on table made → show table data → add values
13. Right click on website → add new item → webform → name it
14. Go to design view → add necessary form
15. Add a grid view below the form → below that add sqldatasource
16. Configure sqldatasource → then add it to the gridview
17. Go to grid view menu → add columns → select command field → check on delete and edit → ok
10. Double click on button → write code.

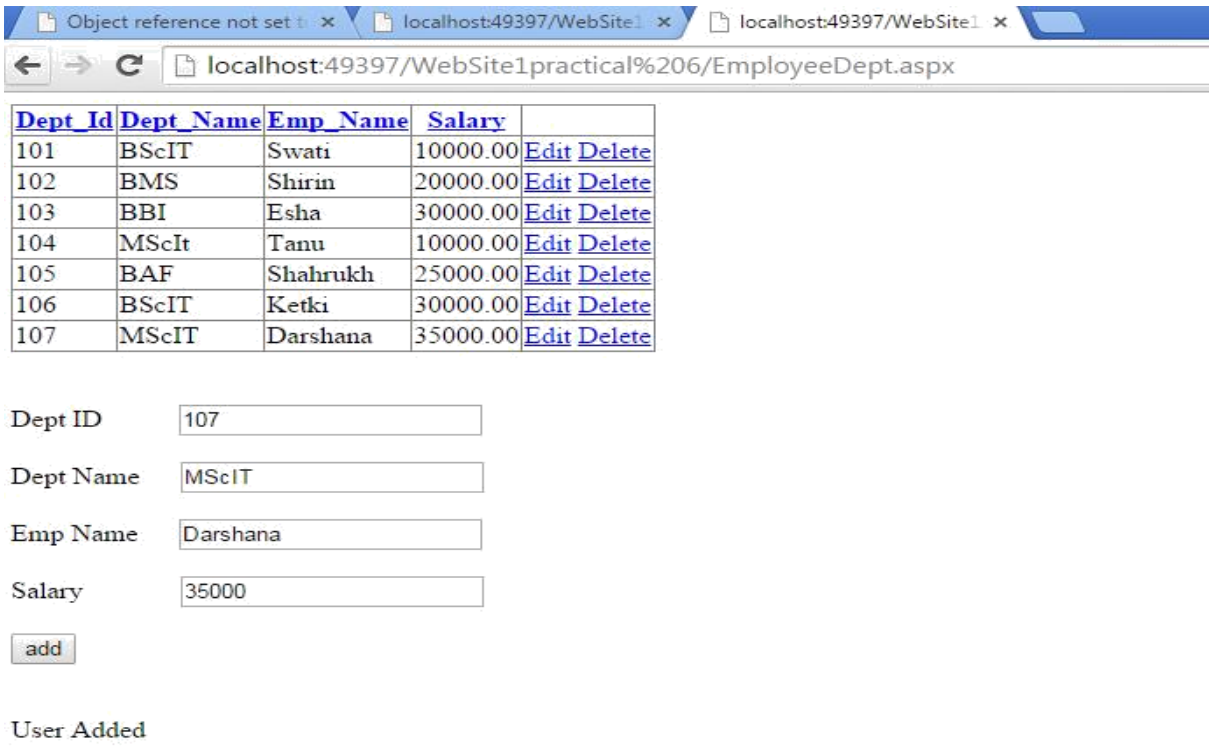
DESIGN:

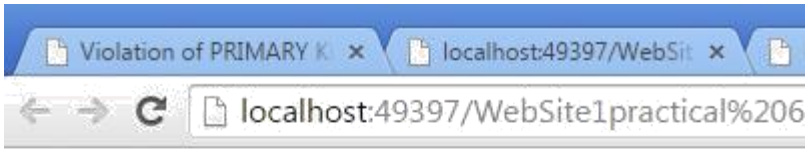


CODE:

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

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

      SqlDataSource1.InsertParameters["Username"].DefaultValue = txtUserName.Text;
      SqlDataSource1.InsertParameters["Password"].DefaultValue = txtPassword.Text;
         SqlDataSource1.Insert();
         Textbox1.Text="";
           Textbox2.Text="";
   }
}
```

OUTPUT:





Dept_Id	Emp_Name	command
101	Swati	Edit Delete
102	Shirin	Edit Delete
103	Esha	Edit Delete
104	Shiwani	Edit Delete
105	Shahrukh	Edit Delete
106	Ketki	Edit Delete

PRACTICAL NO. : 07(1)

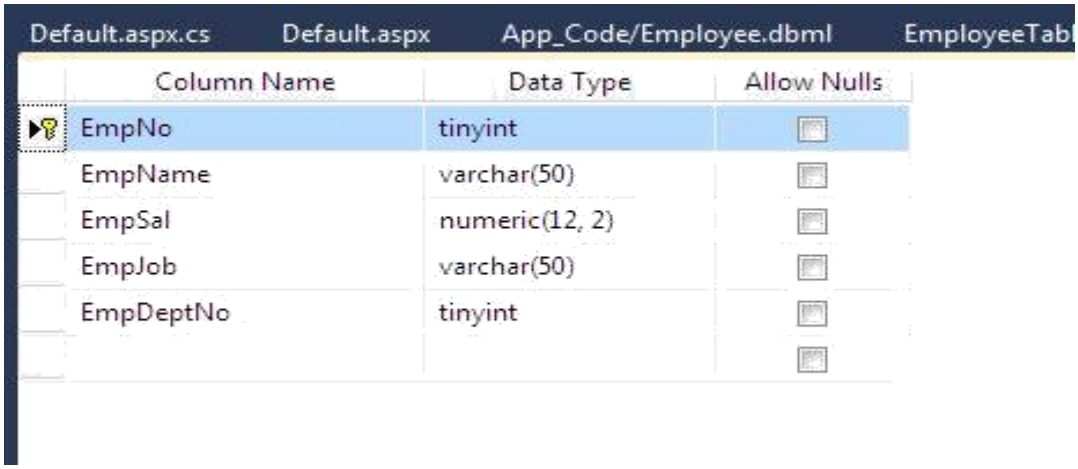
AIM: Programs using Language Integrated query.
Create the table with the given fields.
FIELD NAME DATA TYPE EmpNo number
EmpName varchar EmpSal number EmpJob
varchar EmpDeptNo number

For the given table design a web page to display the employee information from table to grid control. Use LINQ TO ADO.NET.

STEPS:

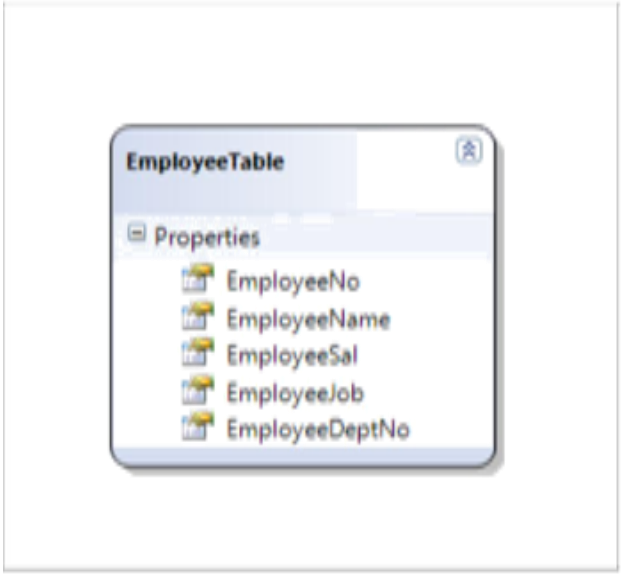
1. File → new → Website → Empty Website → name it → Add
2. Right click on website on solution explorer → Add new item → Sql server database → name it → add → yes
3. Server Explorer → table → right click → add new table → enter the columns → save the table
4. Server explorer → right click on table which is made → show table data → add values
5. Server explorer → right click on website created → add new item → web form → name it
6. Go to design view of aspx page → add grid view from toolbox.
   Double click on aspx page.

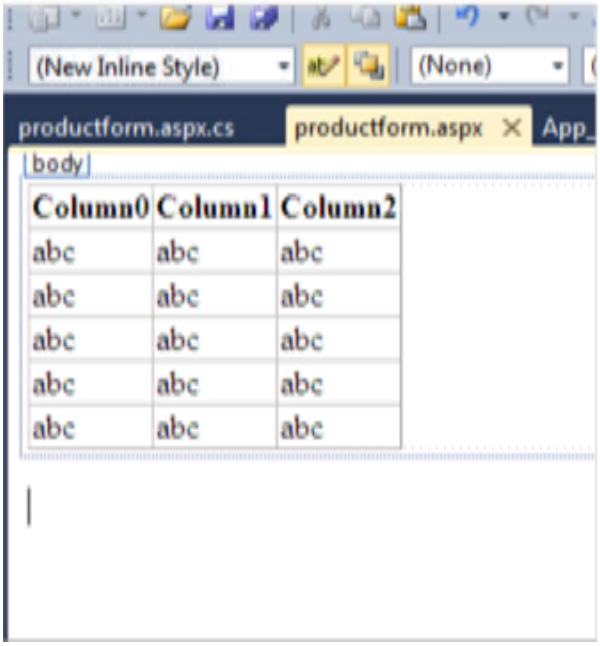
DESIGN:



EmployeeTable: Q...\EMPLOYEEDB.MDF) | dbo.EmployeeTab...\EMPLOYEEDB.MDF)

EmployeeNo	EmployeeName	EmployeeSal	EmployeeJob	EmployeeDept...
1	Swati	10000.00	HR	10
2	Shirin	25000.00	Manager	11
3	Shiwani	15000.00	MD	12
4	Esha	50000.00	CEO	13
5	Prince	5000.00	programmer	14
15	Ankita	1000.00	Clerk	17
NULL	NULL	NULL	NULL	NULL





CODE:

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Data.Linq;
using System.Data.SqlClient;
using System.Web.UI.WebControls;
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      EmployeeDataContext dc = new EmployeeDataContext();
      var query = from m in dc.EmployeeTables select m;

      GridView1.DataSource = query;
      GridView1.DataBind();
   }
}
```

OUTPUT:

EmployeeNo	EmployeeName	EmployeeSal	EmployeeJob	EmployeeDeptNo
1	Swati	10000.00	HR	10
2	Shirin	25000.00	Manager	11
3	Shiwani	15000.00	MD	12
4	Esha	50000.00	CEO	13
5	Prince	5000.00	programmer	14
15	Ankita	1000.00	Clerk	17

PRACTICAL NO. : 07(2)

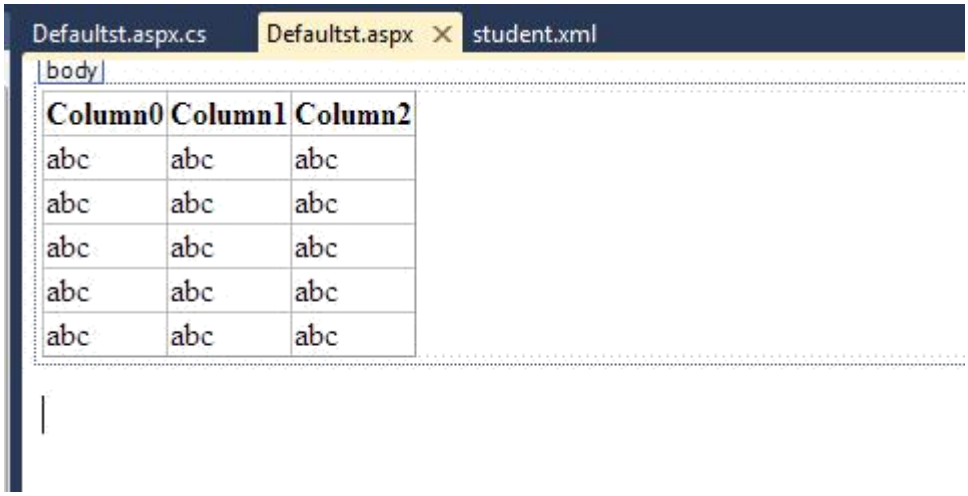
AIM: Programs using Language Integrated query.
Create the table with the given fields.
FIELD NAME DATA TYPE SRollno int SName
string SAddress string SFees int

For the given table design a web page to display the employee information from table to grid control. Use LINQ TO XML.

STEPS:

1. File → New → website → Empty Website → name it
2. Solution Explorer → right click on website made → add new item → XML file → name it → add → write code
3. Solution explorer → right click on website → add new item → webform → name it → add
4. Go to design view → double click page → write code.

DESIGN:



CODE:

student.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<TYStudents>
  <student>
    <srollno>1</srollno>
    <sname>swati</sname>
    <saddress>Wadala</saddress>
    <sfees>1000</sfees>
  </student>
  <student>
    <srollno>2</srollno>
    <sname>natasha</sname>
```

```
    <saddress>Dadar</saddress>
    <sfees>3000</sfees>
  </student>
</TYStudents>
```

Defaultst.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Xml.Linq;
using System.Web.UI.WebControls;
public partial class Defaultst : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        XDocument xmlDoc =
XDocument.Load(HttpContext.Current.Server.MapPath("student.xml"));
        var studs = from s in xmlDoc.Descendants("student")
                    select s;
        GridView1.DataSource = studs;
        GridView1.DataBind();
    }
}
```

OUTPUT:

PRACTICAL NO. : 07(3)

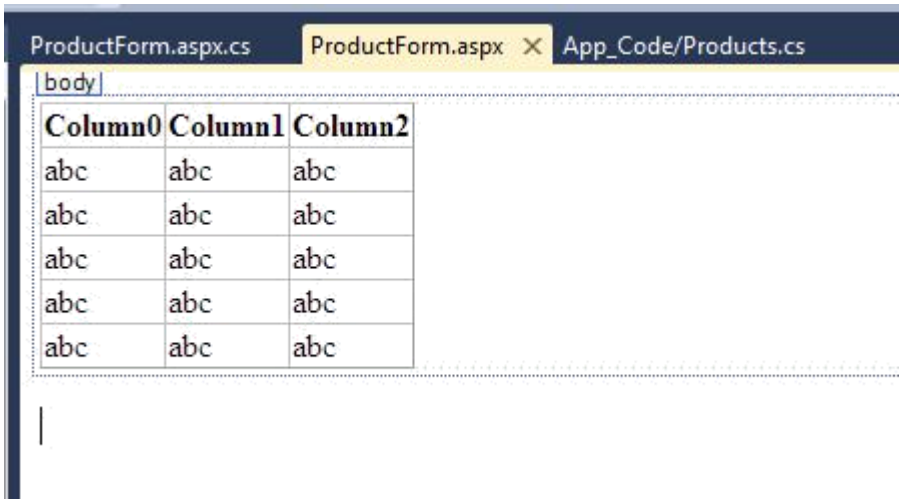
AIM: Programs using Language Integrated query.
Create the table with the given fields .
FIELD NAME DATA TYPE PID string PName
string PPrice int PWeight int

For the given table design a web page to display the employee information from table to grid control. Use LINQ TO Objects.

STEPS:

1. File → new → website → name it
2. Solution explorer → right click on website made → class → name it → yes → write code
3. Solution explorer → right click on website → add new item → webform → name it → add
4. Go to design view → add GridView → Double click on page → write code.

DESIGN:



CODE:
App_Code/Products.cs

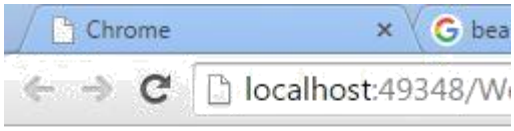
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
public class Products
{
public string PID { get; set; }
   public string PName { get; set; }
   public int PPrice { get; set; }
   public int PWeight { get; set; }
```

```
    public Products()
    {
    } }
```

ProductForm.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class ProductForm : System.Web.UI.Page
{
    public List<Products> GetProdData()
    {
        return new List<Products> {
                new Products { PID="P101", PName="Laptop", PPrice=25000 , PWeight=1500},
                 new Products { PID="P102", PName="Desktop", PPrice=22000 , PWeight=8000},
                 new Products { PID="P103", PName="Mouse", PPrice=500 , PWeight=250}
            };
    }
    protected void Page_Load(object sender, EventArgs e)
    {
        var prod = GetProdData();
        var query = from f in prod
                    orderby f.PName
                    select f;
        this.GridView1.DataSource = query;
        this.GridView1.DataBind();
    }
}
```

OUTPUT:



PID	PName	PPrice	PWeight
P102	Desktop	15000	1000
P101	Laptop	25000	1500
P103	Mouse	25000	1500

PRACTICAL NO. : 08

AIM: (A) For the web page created for the display OF Employee data change the authentication mode to Windows

CODE:

```
<system.web>
<authentication mode=”Windows”>
<forms loginUrl=”~/”Prac8/EmployeeForm.aspx”>
</authentication>
</system.web
```

Steps for changing the authentication mode

1. Open the website created for displaying the Employee data
2.From the solution Explorer window open the web.config file
3 .In the web.config file search the <system.web> xml tag and in <system.web> xml tag go to authentication tag
4. Change the authentication mode to windows as given above.

AIM: (B) For the webpage created for the display of Student data change the authorization mode so that only users who have logged in as VSIT will have the authority to aces the page

CODE:

```
<system.web>
<authentication>
<allow users=”VSIT”/>
<deny users =” *”/>
</authentication>
</system.web>
```

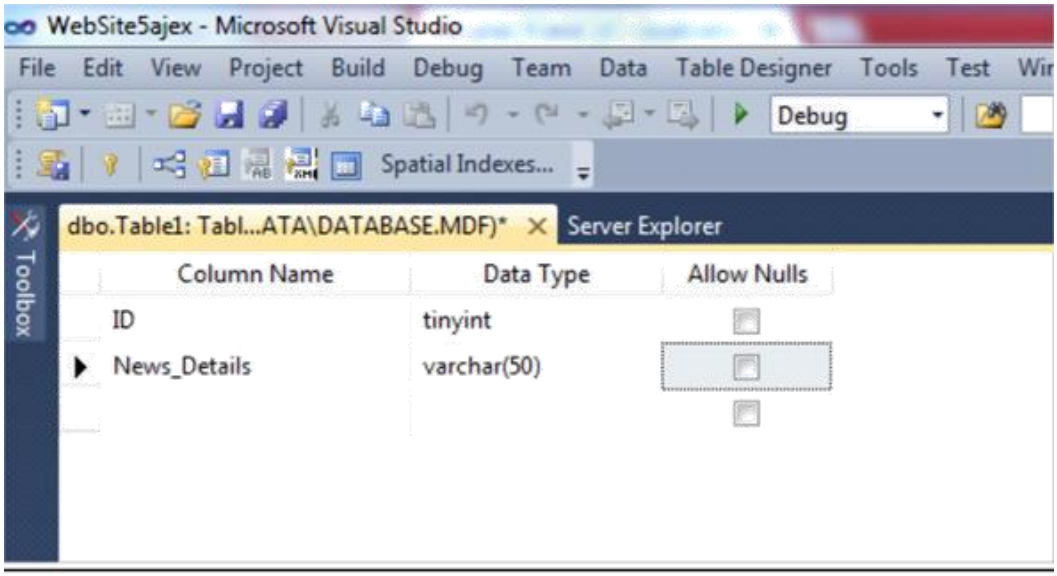
Steps for changing the authorization

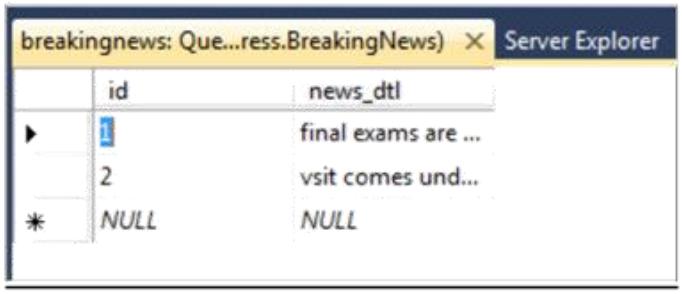
1. Open the website created for displaying the Student data
2. From the solution Explorer window open the web.config file
3. In the Web.config file search the <system.web> xml tag and in <system.web> xml tag go to authentication tag
4. Change the coding in the tag as given above

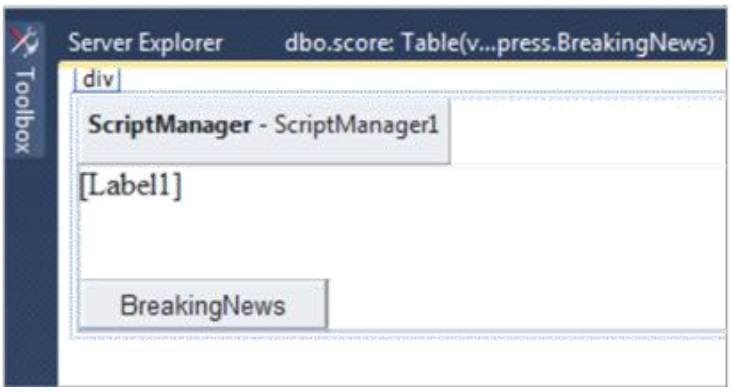
PRACTICAL NO: 9(A)

AIM: Create a web page to display the news from the news table(id, news_dtl). Use AJAX.

DESIGN :



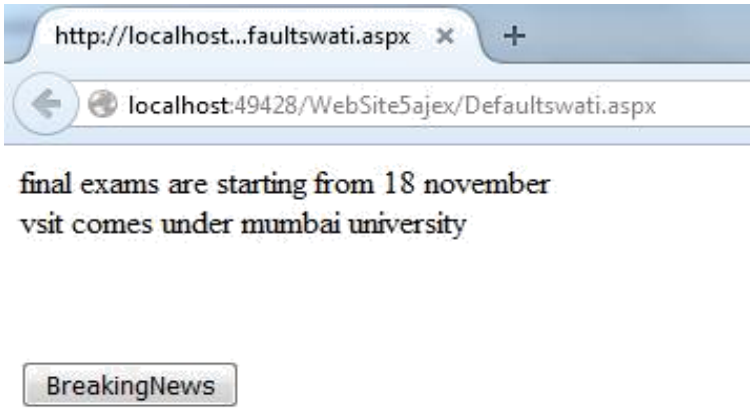




CODE:

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

publicpartialclassajaxform : System.Web.UI.Page
{
protectedvoidPage_Load(object sender, EventArgs e)
   {

   }
protectedvoid Button1_Click(object sender, EventArgs e)
   {
SqlConnection con = newSqlConnection(@"Data Source=.\sqlexpress;Initial
Catalog=BreakingNews;Integrated Security=True"); con.Open();

SqlCommand com = newSqlCommand("select * from news",
con); SqlDataReaderdr = com.ExecuteReader();
while (dr.Read())
      {
         Label1.Text +=dr[1].ToString()+"<br>";
   }
con.Close();
   }
}
```

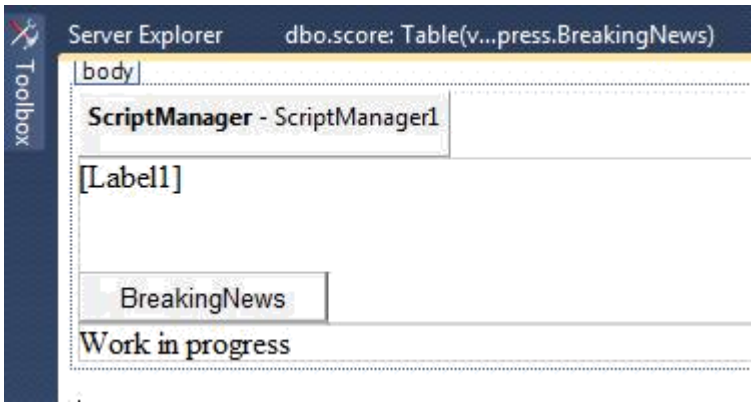
OUTPUT:



PRACTICAL NO: 9(B)

AIM: In the above website also display the feedback on the browser as "work is in progress".

DESIGN:



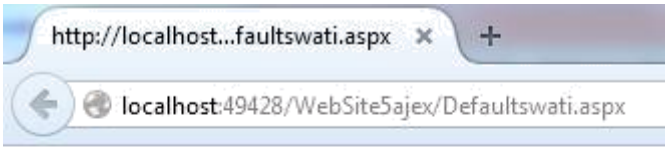
CODE:

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

publicpartialclassajaxform : System.Web.UI.Page
{
protectedvoidPage_Load(object sender, EventArgs e)
   {
System.Threading.Thread.Sleep(5000);
   }
protectedvoid Button1_Click(object sender, EventArgs e)
   {
SqlConnection con = newSqlConnection(@"Data Source=.\sqlexpress;Initial
Catalog=BreakingNews;Integrated Security=True"); con.Open();

SqlCommand com = newSqlCommand("select * from news",
con); SqlDataReaderdr = com.ExecuteReader();
while (dr.Read())
      {
         Label1.Text +=dr[1].ToString()+"<br>";
   }
con.Close();
   }
}
```

Source Code:

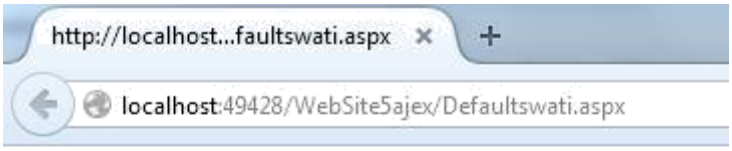
```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile="ajaxform.aspx.cs"Inherits="aj
axform"%>

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

<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title></title>
</head>
<body>
<formid="form1"runat="server">
<div>

<asp:ScriptManagerID="ScriptManager1"runat="server">
</asp:ScriptManager>
<br/>
<asp:UpdatePanelID="UpdatePanel1"runat="server">
<ContentTemplate>
<asp:LabelID="Label1"runat="server"></asp:Label>
<br/>
<br/>
<asp:ButtonID="Button1"runat="server"Text="Breaking news"/>
<br/>
</ContentTemplate>
</asp:UpdatePanel>
<br/>
<br/>
<br/>


<asp:UpdateProgressID="UpdateProgress1"runat="server">
<ProgressTemplate>Work in progress</ProgressTemplate>
</asp:UpdateProgress>
<br/>
<br/>
</div>
</form>
</body>
</html>
```

Output:



BreakingNews

Work in progress



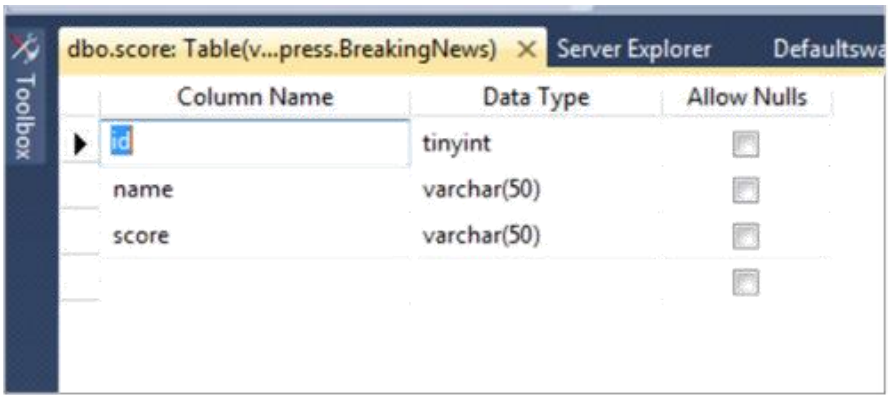
final exams are starting from 18 november
vsit comes under mumbai university

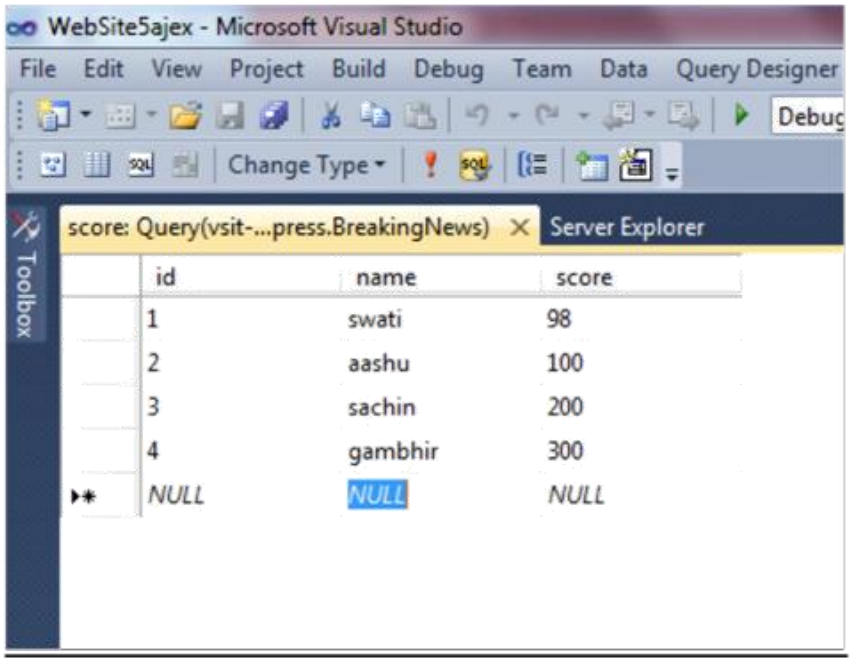
BreakingNews

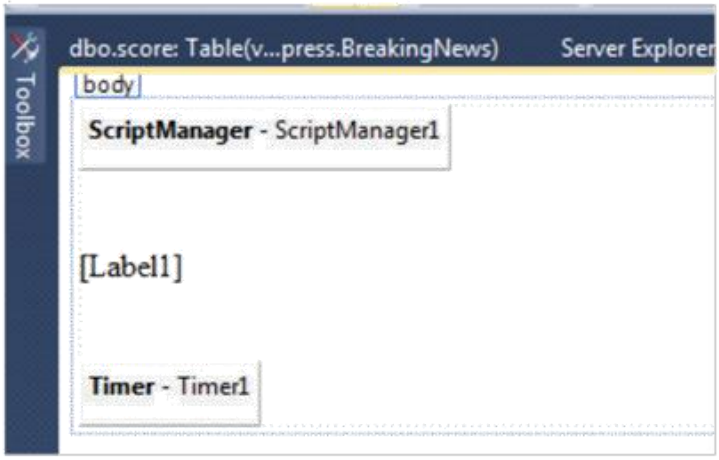
PRACTICAL NO: 9(C)

AIM: Create a web page to display the cricket score from the table event(id, name, score). Refresh the website automatically after every 30 seconds.

DESIGN:





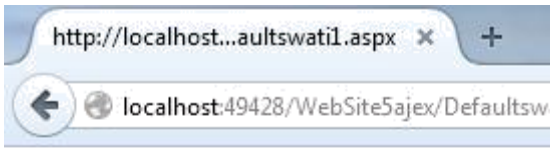


CODE:

Default.aspx

```
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;
public partial class Defaultswati1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Timer1_Tick(object sender, EventArgs e)
    {
        SqlConnection conn = new SqlConnection(@"Data Source=.\sqlexpress;Initial
Catalog=BreakingNews;Integrated Security=True");
        SqlDataReader dr = null;

        conn.Open();
        SqlCommand cmd = new SqlCommand("Select * from score", conn);
        dr = cmd.ExecuteReader();

        while (dr.Read())
        {
            Label1.Text += dr[0].ToString() + " " + dr[1].ToString() + " " + dr[2].ToString()
+ "<br>";
        }
        conn.Close();

    }
}
```

OUTPUT:

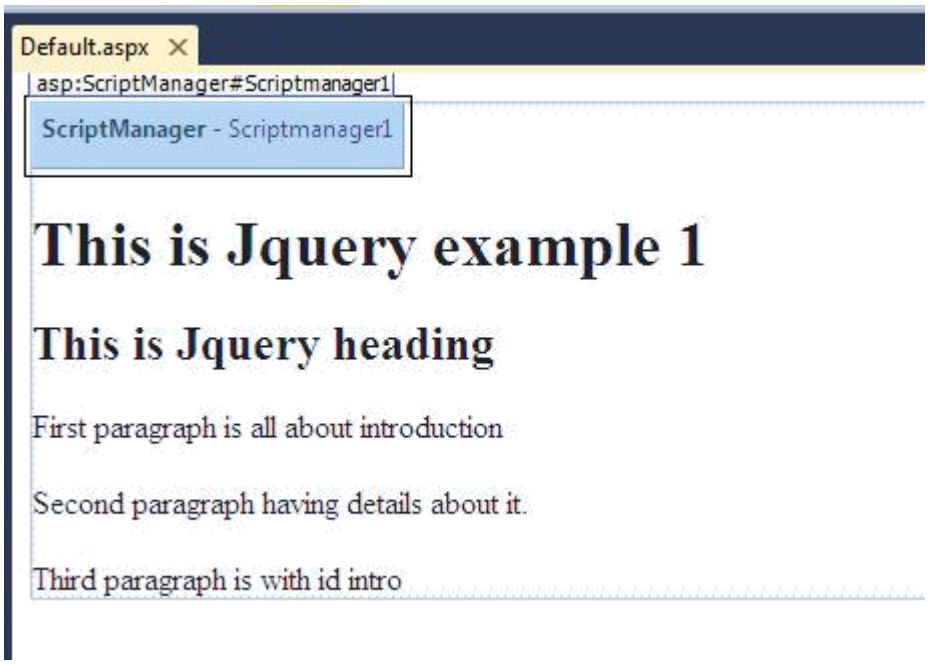


1 swati 98
2 aashu 100
3 sachin 200
4 gambhir 300
1 swati 98
2 aashu 100
3 sachin 200
4 gambhir 300
1 swati 98
2 aashu 100
3 sachin 200
4 gambhir 300

PRACTICAL NO: 10(A)

AIM: Create a web page to give different color effects for paragraph tags, heading tags and complete web page using JQuery.

DESIGN:

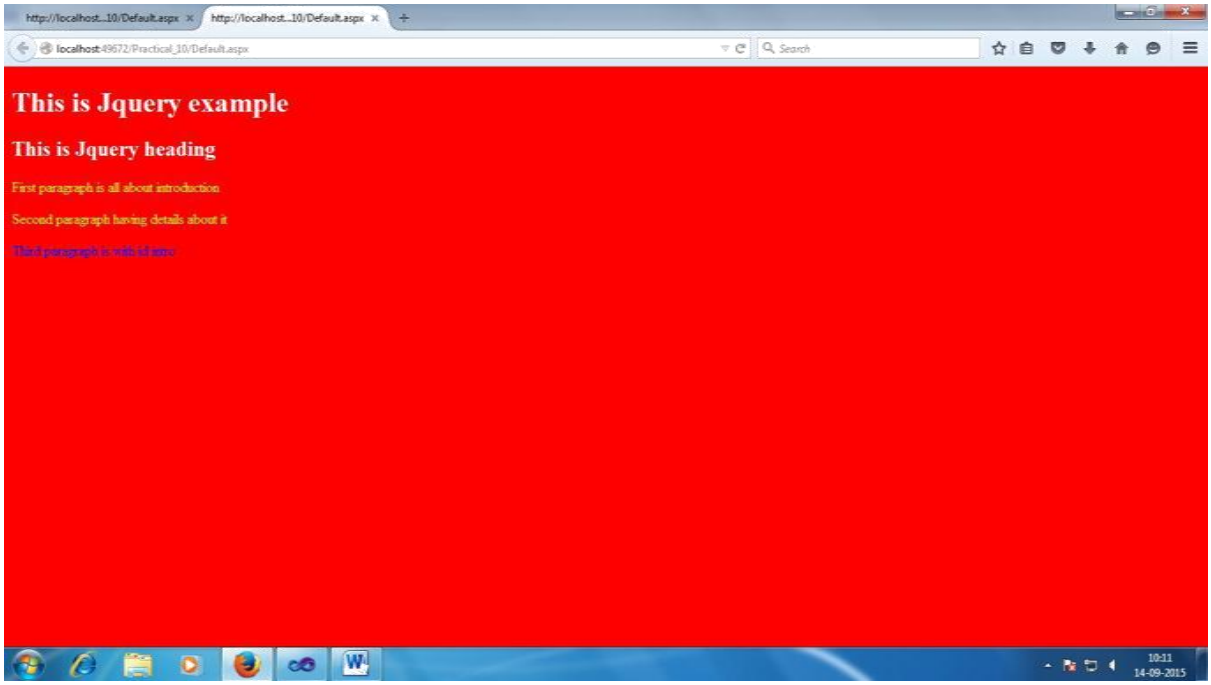


Source Code:

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

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

<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title></title>
</head>
<body>
<formid="form1"runat="server">
<div>
<scripttype="text/javascript">
        $(document).ready(function () {
            $("p").css("color", "Yellow");
            $("h1,h2").css("color", "White");
            $("p#intro").css("color", "Blue");
            $("*").css("background-color", "Red");
        });
```

```
</script>
<asp:ScriptManagerID="Scrpitmanager1"runat="server">
<Scripts>
<asp:ScriptReferencePath="~/scrpits/jquery-1.11.3.js"/>

</Scripts>
</asp:ScriptManager>
<h1>This is Jquery example</h1>
<h2>This is Jquery heading</h2>
<p>First paragraph is all about introduction</p>
<p>Second paragraph having details about it</p>
<pid="intro">Third paragraph is with id intro</p>
</div>
</form>
</body>
</html>
```

OUTPUT:



PRACTICAL NO: 10(B)

AIM: Create a web page to display animation using JQuery.

DESIGN:

Source Code:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <script type="text/javascript"">
      $(document).ready(function () {
         $('p').hide(1000);
         $('p').show(2000);
         $('p').toggle(3000);
         $('p').slideDown(4000);
         $('p').slideUp(5000);

         $('h1').animate({
            opacity: 0.4, marginLeft: '50px', fontSize: '100px'
         }, 8000);
      });
```

```
    </script>
    <asp:ScriptManager ID="Scriptmanager1" runat="server">
<Scripts>
<asp:ScriptReference Path="~/Scripts/jquery-1.11.3.js" />
</Scripts>
</asp:ScriptManager>
<p>First Paragraph</p>
<h1>First Heading</h1>

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

OUTPUT:

PRACTICAL NO: 10(C)

AIM: Create a web page to display hide, show, slidedown, slideup and Toggle effects for paragraph tags, using JQuery.

DESIGN:

Source Code:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <script type="text/javascript">
   $(document).ready(function(){
   $('h1').animate({
   opacity:
   0.4,marginLeft:'50px',fontSize:'100px'},8000); });
   </script>
   <asp:ScriptManager ID="ScriptManager1"
   runat="server"> <Scripts>
   <asp:ScriptReference Path="~/script/jquery-1.11.3.js" /></Scripts></asp:ScriptManager>
   <p>First paragraph</p>
   <h1>First heading heading</h1>
```

```
    </div>

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

OUTPUT: