1. Write a Program in C# to perform the GCD of given number using Command Line arguments.

```
using System;

public class gcd
{
        static int GCD(int a, int b)
        {
                int Remainder;
                while( b != 0 )
                {
                        Remainder = a % b;
                        a = b;
                        b = Remainder;
                }
                return a;
        }
static int Main(string[] args)
        {
                int x, y;
                Console.WriteLine("This program allows calculating the GCD");
                Console.Write("Value 1: ");
                x = int.Parse(Console.ReadLine());
                Console.Write("Value 2: ");
                y = int.Parse(Console.ReadLine());
                Console.Write("\nThe Greatest Common Divisor of ");
                Console.WriteLine("{0} and {1} is {2}", x, y, GCD(x, y));
                return 0;
        }
}
```

2. Write a program in C# to Design simple calculator using Method overloading.

```
using System;

namespace ConsoleApp1
{
   class Method_overloading
   {
      public int result(int a, int b)
      {
         int x;
         return x = a / b;
      }
      public float result(float a, float b)
      {
         float u;
         return u = a - b;
      }
      public int result(int a, int b, int c)
```

```
    {
        int y;
        return y = a * b * c;
    }

    public float result(float a, float b, float c)
    {
        float v;
        return v = a + b + c;
    }
}
class mtdold
{
    public static void Main(String[] args)
    {
        char c;
        float a, b, d;
        int p, q, r;
        Method_overloading mthover = new Method_overloading();
        menu:
        Console.Clear();
        Console.WriteLine("Menu");
        Console.WriteLine("1.Addition");
        Console.WriteLine("2.Subtraction");
        Console.WriteLine("3.Multiplication");
        Console.WriteLine("4.Division");
        Console.WriteLine("Enter Your Option");
        c = Convert.ToChar(Console.ReadLine());
        switch (c)
        {
            case '1':
                Console.WriteLine("Enter 3 double type values");
                a = float.Parse(Console.ReadLine());
                b = float.Parse(Console.ReadLine());
                d = float.Parse(Console.ReadLine());
                Console.WriteLine("Sum =     " + mthover.result(a, b, d));
                break;
            case '2':
                Console.WriteLine("Enter 2 double type values");
                a = float.Parse(Console.ReadLine());
                b = float.Parse(Console.ReadLine());
                Console.WriteLine("Difference = " + mthover.result(a, b));
                break;
            case '3':
                Console.WriteLine("Enter 3 integer values");
                p = int.Parse(Console.ReadLine());
                q = int.Parse(Console.ReadLine());
                r = int.Parse(Console.ReadLine());
                Console.WriteLine("Product =     " + mthover.result(p, q, r));
                break;
            case '4':
                Console.WriteLine("Enter 2 integer values");
```

```
                p = int.Parse(Console.ReadLine());
                q = int.Parse(Console.ReadLine());
                Console.WriteLine("Quotient   =     " + mthover.result(p, q));
                break;

            }
            Console.ReadLine();
            Console.WriteLine("Enter m for Menu or Any other key to Exit");
            c = Convert.ToChar(Console.ReadLine());
            if (c == 'm')
            {
               goto menu;
            }
        }
    }
}
```

3. Write a Program in C# to show the difference between Constructor overloading and operator overloading

```
using System;
namespace ConsoleApp1
{
    class constructOverload
    {
        string user;
        int age;
        public constructOverload()
        {
            user = "Steve jobs";
            age = 28;
            Console.WriteLine("Previous User {0} and he was {1} year old", user, age);
        }
        public constructOverload(string name, int age1)
        {
            user = name;
            age = age1;
            Console.WriteLine("Current User {0} and he is {1} year old", user, age);
        }
    }
    class Complex
    {
        public int a;
        public int b;
        public Complex(int a, int b)
        {
            this.a = a;
            this.b = b;
        }
        public static Complex operator +(Complex c1, Complex c2)
        {
            return new Complex(c1.a + c2.a, c1.b + c2.b);
        }
        public override string ToString()
```

```
        {
            return (String.Format("a == {0} b === {1}", a, b));
        }

        static void Main(string[] args)
        {
            constructOverload a = new constructOverload();
            constructOverload a2 = new constructOverload("Bill Gates", 41);
            Complex num1 = new Complex(9, 3);
            Complex num2 = new Complex(3, 4);
            Complex sum = num1 + num2;
            Console.WriteLine("result {0}", sum);
            Console.Read();
        }
    }
}
```

5. Using Try, Catch and Finally blocks write a program in C# to demonstrate error handling.

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

namespace error
{
    class err
    {
        public static void Main()
        {
            try
            {
                int zero = 0;
                Console.WriteLine("In try block: attempting division by zero");
                int myInt = 1 / zero;
                Console.WriteLine("You never see this message!");
            }
            catch (ArithmeticException e)
            {
                Console.WriteLine("In catch block:   an exception was thrown");
                      Console.WriteLine("\n" + e);
            }
            try
            {
            string x = “abcde”;
            char a1 = x[1];
            char a2 = x[100];
            Console.WriteLine(a1);
            Console.WriteLine(a2);
            }
            catch (IndexOutOfRangeExpection e)
            {
            Console.WriteLine("\n" + e);
```

```
        }

        finally
        {
            Console.WriteLine("In finally block:   do any cleaning up here");
        }
        Console.ReadLine();
    }
  }
}
```

6.Write a Program to Demonstrate Use of Virtual and override key words in C# with a simple program.

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

namespace prgm6
{

  class super
   {
      protected int x;
      public super(int x)
      {
         this.x = x;
      }
      public virtual void display()
      {
         Console.WriteLine("\n Super x = " + x);
      }
   }
   class sub : super
   {
      int y;
      public sub(int x, int y) : base(x)
      {
         this.y = y;
      }
      public override void display()
      {
         Console.WriteLine("\n Sub x = " + x);     // OR base.display();
         Console.WriteLine("\n Sub y = " + y);
      }
   }
   class overridetest
   {
```

```
        public static void Main()
        {
            sub s = new sub(100, 200);
            s.display();
            Console.ReadLine();
        }
    }
}
```

7. Write a program to demonstrate delegates.

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

namespace delg
{
    public delegate int DelegateSample(int a,int b);
    public class sampleclass
    {
        public int add(int x, int y)
        {
            return x + y;
        }
        public int sub(int x, int y)
        {
            return x - y;
        }
        class program {
            static void Main(String[] args)
            {
                sampleclass s = new sampleclass();
                DelegateSample del = s.add;
                int i = del(10, 20);
                Console.WriteLine("Add result is"+i);
                DelegateSample del1 = s.sub;
                int j = del1(10, 2);
                Console.WriteLine("Sub result is" + j);
                Console.ReadLine();

            }
        }
    }
}
```

8. Write a program to demonstrate abstract class and abstract methods in C#.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace abst
```

```
{
abstract class test1
{
public int add(int i, int j)
{
return i + j;
}
public abstract int mul(int i, int j);
}

class test2: test1
{
public override int mul(int i, int j) {
return i * j;
}
}

class test3: test1
{
public override int mul(int i, int j) {
return i - j;
}
}

class test4: test2
{
public override int mul(int i, int j) {
return i + j;
}
}

class myclass
{
public static void Main(string[] args)
{
test2 ob = new test4();
int a = ob.mul(2, 4);
test1 ob1 = new test2();
int b = ob1.mul(4, 2);
test1 ob2 = new test3();
int c = ob2.mul(4, 2);
Console.Write("{0},{1},{2}", a, b, c);
Console.ReadLine();
}
}
}
```

9. Write a program to illustrate the use of different properties in C#.

```
using System;
class Student {
    private string code = "N.A";
    private string name = "not known";
```

```
   private int age = 0;

   // Declare a Code property of type string:
   public string Code {
      get {
         return code;
      }
      set {
         code = value;
      }
   }

   // Declare a Name property of type string:
   public string Name {
      get {
         return name;
      }
      set {
         name = value;
      }
   }

   // Declare a Age property of type int:
   public int Age {
      get {
         return age;
      }
      set {
         age = value;
      }
   }
   public override string ToString() {
      return "Code = " + Code +", Name = " + Name + ", Age = " + Age;
   }
}

class ExampleDemo {
   public static void Main() {

      // Create a new Student object:
      Student s = new Student();

      // Setting code, name and the age of the student
      s.Code = "001";
      s.Name = "Zara";
      s.Age = 9;
      Console.WriteLine("Student Info: {0}", s);

      //let us increase age
      s.Age += 1;
      Console.WriteLine("Student Info: {0}", s);
      Console.ReadKey();
```

```
    }
  }
```

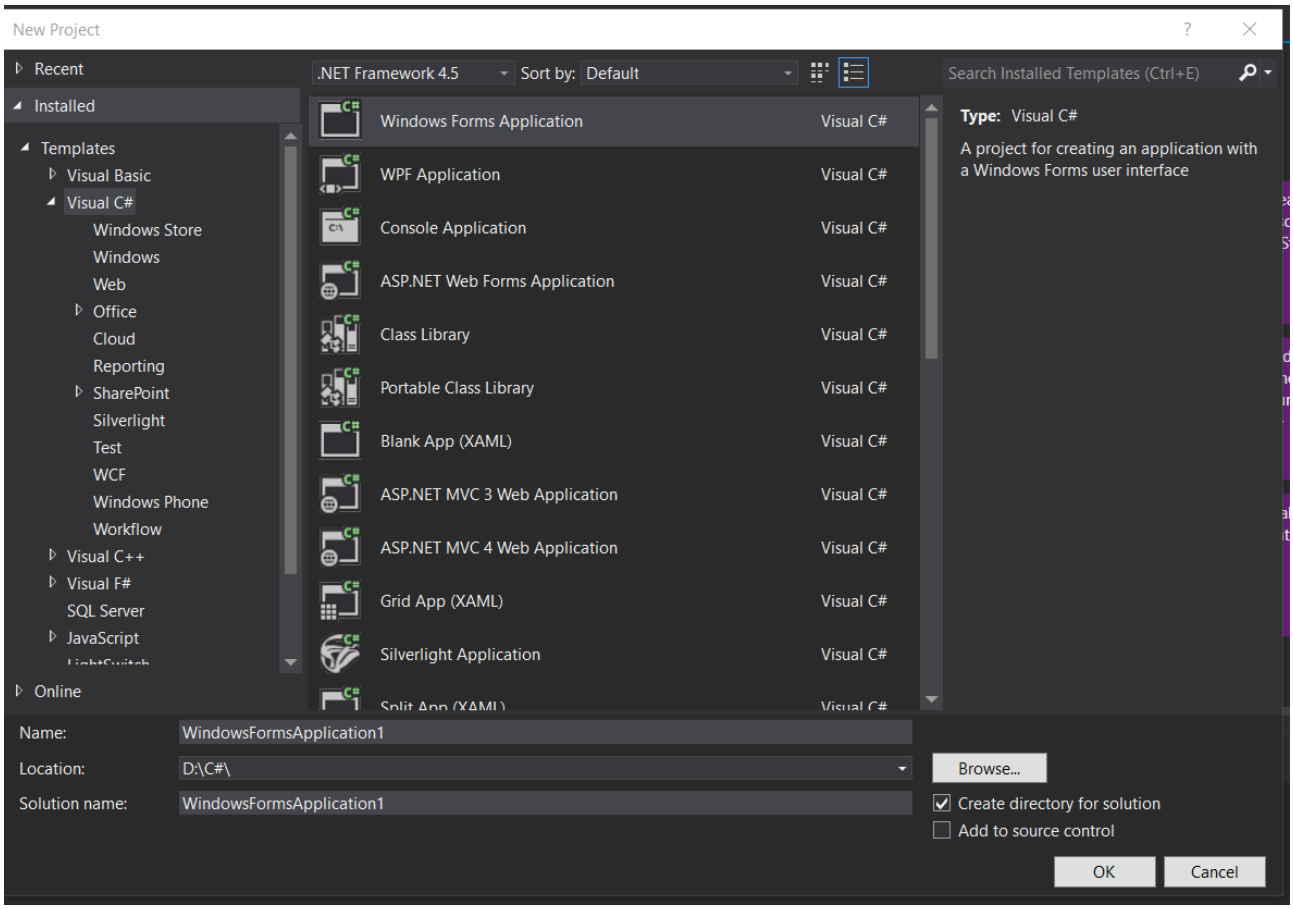
10. Demonstrate arrays of interface types (for runtime polymorphism) with a C# program.

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Aray_Interface
{
class Program
 {
 public static void FindMatches(IList<string>iList,object[] ob)
 {
 Console.WriteLine("Match array is:");
 foreach(object o in ob)
 {
 Console.WriteLine("{0}",o.ToString());
 }
 foreach(object o in ob)
 {
 if(iList.Contains(o.ToString()))
 Console.WriteLine("\niList contains {0} at index{1}", o,iList.IndexOf(o.ToString()));
 }
 }
 static void Main(string[] args)
 {
 string[] strings ={ "one", "two", "four", "eight" };
 Console.WriteLine("Strings array values:\n");
 foreach(String s in strings)
 {
 Console.WriteLine("{0}",s);
 }
 Console.WriteLine("\n");
 FindMatches(strings,new String[]{"zero","one","fiye","eight"});
 Console.ReadKey();
 }
 }}
```

PART B

TO CREATE A WINDOWS FORMS APPLICATION

FILE > NEW > PROJECT



TO CREATE A DATABASE

VIEW > SERVER EXPLORER

CLICK SERVER EXPLORER ON LEFT SIDE

RIGHT CLICK ON DATA CONNECTIONS > ADD CONNECTION > BROWSE > ENTER YOUR DESIRED DATABASE NAME (this creates a new database file if there is no existing file)

EXPAND DATABASE FILE AND RIGHT CLICK ON TABLES > ADD TABLES OR NEW QUERY TO CREATE A TABLE

RIGHT CLICK ON DATABASE FILE AND CLICK PROPERTIES TO COPY THE CONNECTION STRING

C#.Net Part B programs

1.Consider the Database Employee consisting of following tables:
tbl_company (comID: int, ComName: string)
tbl_EMP (empID: string, EMPNAME: string, Address: string, comID: int, YrOfJoining: int)
Develop suitable windows application using C#.NET having following options:

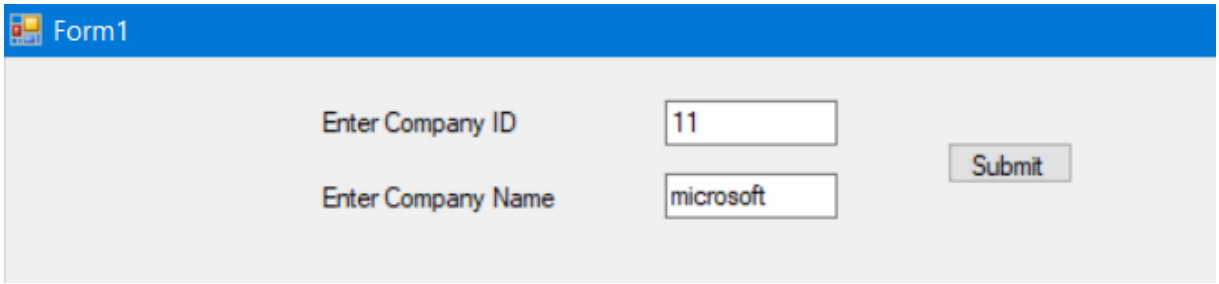
- **Entering new COM details.**
- **Entering new EMPLOYEE details.**
- **Display the details of Employee (in a Grid) that belong to a particular COMPANY.**
- **Display the details the Employee who have taken Joined in a particular year.**

```
namespace prog1
{
   public partial class Form1 : Form
   {
      SqlConnection con = new SqlConnection(@"Data Source =
(LocalDB)\MSSQLLocalDB; AttachDbFilename=C:\Users\Vinod
Kumar\Documents\comp.mdf;Integrated Security = True; Connect Timeout = 30");
      SqlCommand cmd;
      public Form1()
      {
         InitializeComponent();
         string str = "select comid from tblcompany";
         con.Open();
         SqlCommand cmd1 = new SqlCommand(str, con);
         SqlDataReader rs;
         rs = cmd1.ExecuteReader();
         while (rs.Read())
         {
            comboBox1.Items.Add(rs[0]);
            comboBox1.SelectedItem = rs[0];
         }

         con.Close();
      }

      private void button1_Click(object sender, EventArgs e)
      {
         cmd = new SqlCommand("insert into tblcompany(comid,comname) values('"+
textBox1.Text + "','"+ textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Company Inserted Successfully");
```

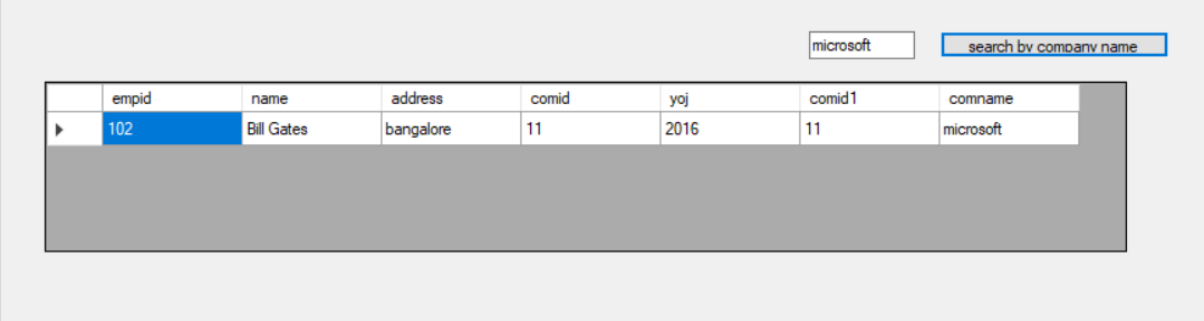
```
            string str = "select comid from tblcompany";
            con.Open();
            SqlCommand cmd1 = new SqlCommand(str, con);
            SqlDataReader rs;
            rs = cmd1.ExecuteReader();
            while (rs.Read())
            {
               comboBox1.Items.Add(rs[0]);
               comboBox1.SelectedItem = rs[0];
            }

            con.Close();

        }

        private void button2_Click(object sender, EventArgs e)
        {
            cmd = new SqlCommand("insert into tblemp(empid,name,address,comid,yoj)
values(@empid,@name,@address,@comid,@yoj)", con);
            con.Open();
            cmd.Parameters.AddWithValue("@empid", textBox3.Text);
            cmd.Parameters.AddWithValue("@name", textBox4.Text);
            cmd.Parameters.AddWithValue("@address", textBox5.Text);
            cmd.Parameters.AddWithValue("@comid", textBox6.Text);
            cmd.Parameters.AddWithValue("@yoj", textBox7.Text);
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Employee details Inserted Successfully");

        }

        private void search_Click(object sender, EventArgs e)
        {

            try
            {
               con.Open();
               SqlCommand cmd = new SqlCommand("Select * from tblemp e,tblcompany c
where e.comid=c.comid and c.comname='" + textBox8.Text + "'", con);
               cmd.ExecuteNonQuery();
               DataTable dt = new DataTable();
               SqlDataAdapter sda = new SqlDataAdapter(cmd);
               sda.Fill(dt);
               dataGridView1.DataSource = dt;
               con.Close();
```

```
            }
            catch (Exception ec)
            {
                MessageBox.Show(ec.Message);
            }
        }


        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from tblemp e,tblcompany c
where e.comid=c.comid and e.yoj='" + textBox9.Text + "'", con);
                cmd.ExecuteNonQuery();
                DataTable dt = new DataTable();
                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                sda.Fill(dt);
                dataGridView1.DataSource = dt;
                con.Close();

            }
            catch (Exception ec)
            {
                MessageBox.Show(ec.Message);
            }
        }
    }
}
```

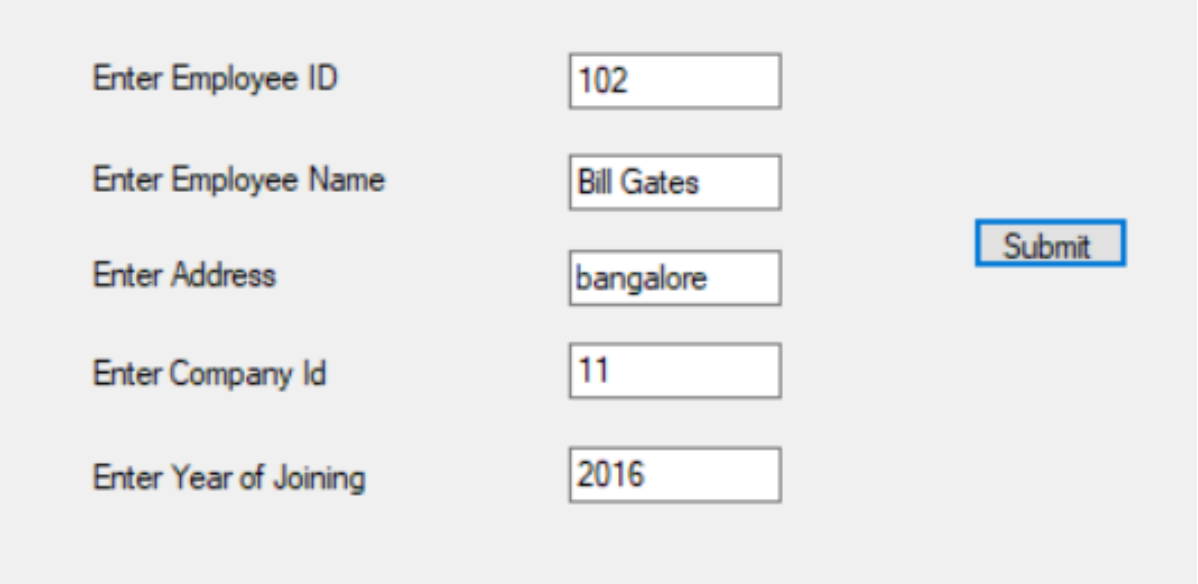
Company Details Entry Form



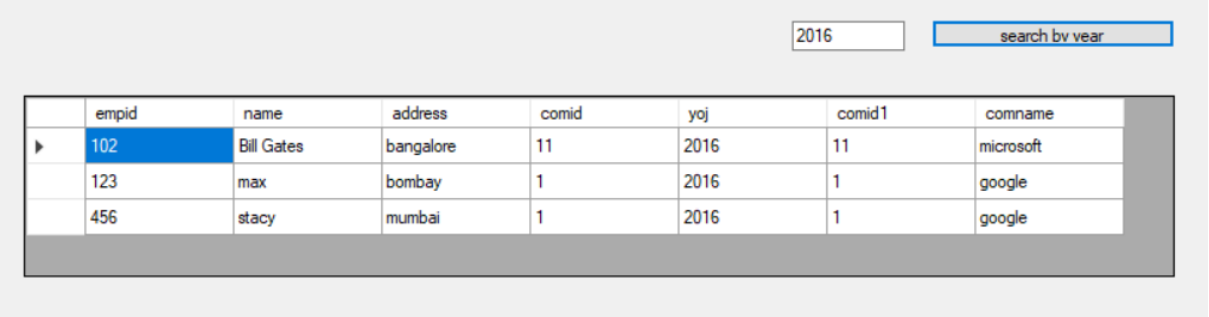
Search by Company



Employee Details Entry Form



Search by Year of Joining



2. Consider the Database BookBANK consisting of following tables:
tbl_BookGroup (BookID: int, BookGroup: string)
tbl_Donor (DonorID: int, DonorName: stirng, Address: string, ContactNo: int, Gender: string, ,BookID: int)
Develop suitable windows application using C#.NET having following options:

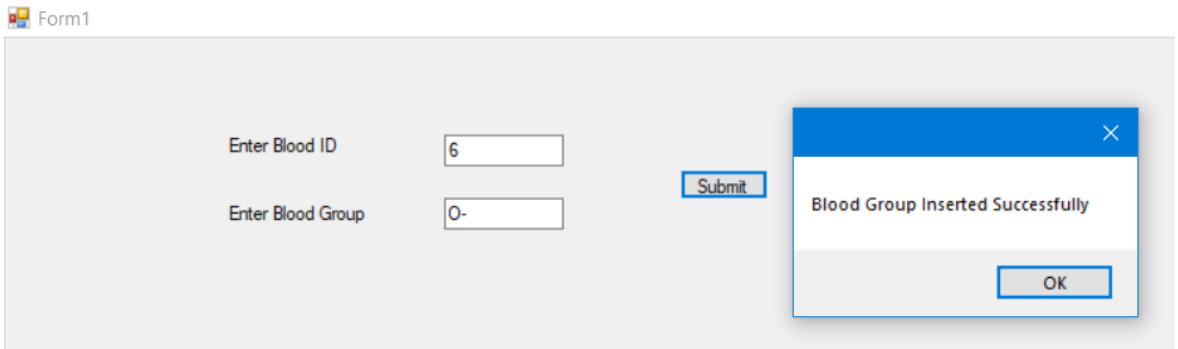
- **Entering Book group details.**
- **Entering new donor details.**
- **Display the details of donors (in a Grid) having particular book group.**
- **Display the details of donors (in a Grid) based on age (above 18).**

```
namespace prog2
{
   public partial class Form1 : Form
   {
      SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Vinod
Kumar\Documents\bloodbank.mdf;Integrated Security=True;Connect Timeout=30");
      SqlCommand cmd;
      public Form1()
      {
         InitializeComponent();
         con.Open();
         string str = "select bloodid from tblblood";
         SqlCommand cmd1 = new SqlCommand(str, con);
         SqlDataReader rs;
         rs = cmd1.ExecuteReader();
         while (rs.Read())
         {
            comboBox2.Items.Add(rs[0]);
            comboBox2.SelectedItem = rs[0];
            comboBox3.Items.Add(rs[0]);
            comboBox3.SelectedItem = rs[0];
         }
         con.Close();
      }

      private void button1_Click(object sender, EventArgs e)
      {
         cmd = new SqlCommand("insert into tblblood(bloodid,bloodname) values('" +
textBox1.Text + "','" + textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Blood Group Inserted Successfully");
         string str = "select bloodid from tblblood";
         con.Open();
```

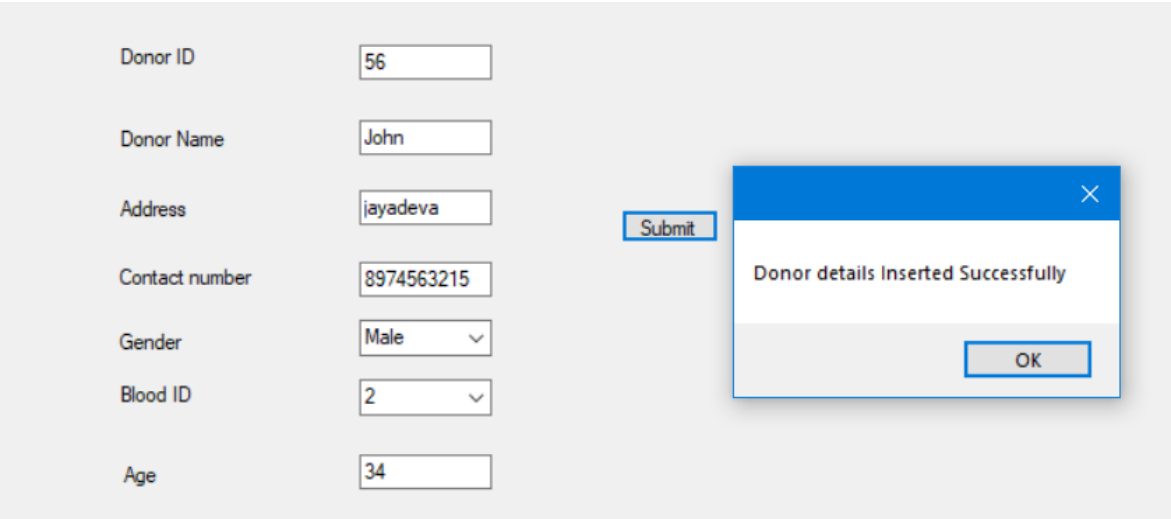
```
            SqlCommand cmd1 = new SqlCommand(str, con);
            SqlDataReader rs;
            rs = cmd1.ExecuteReader();
            while (rs.Read())
            {
                comboBox2.Items.Add(rs[0]);
                comboBox2.SelectedItem = rs[0];
                comboBox3.Items.Add(rs[0]);
                comboBox3.SelectedItem = rs[0];
            }
            con.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            cmd = new SqlCommand("insert into
tbldonor(donorid,donorname,address,contact,gender,bookid,age) values('" + textBox3.Text +
"','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + comboBox1.Text
+ "','" + comboBox2.Text + "','" + textBox7.Text + "')", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Donor details Inserted Successfully");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from tblblood b,tbldonor d where
b.bloodid=d.bloodid and d.age>='" + textBox8.Text + "'", con);
                cmd.ExecuteNonQuery();
                DataTable dt = new DataTable();
                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                sda.Fill(dt);
                dataGridView1.DataSource = dt;
                con.Close();

            }
            catch (Exception ec)
            {
                MessageBox.Show(ec.Message);
            }
        }
```

```
        private void button4_Click(object sender, EventArgs e)
        {
            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("Select * from tblblood b,tbldonor d where
b.bloodid=d.bloodid and b.bloodid='" + comboBox3.Text + "'", con);
                cmd.ExecuteNonQuery();
                DataTable dt = new DataTable();
                SqlDataAdapter sda = new SqlDataAdapter(cmd);
                sda.Fill(dt);
                dataGridView1.DataSource = dt;
                con.Close();
            }
            catch (Exception ec)
            {
                MessageBox.Show(ec.Message);
            }
        }
    }
}
```

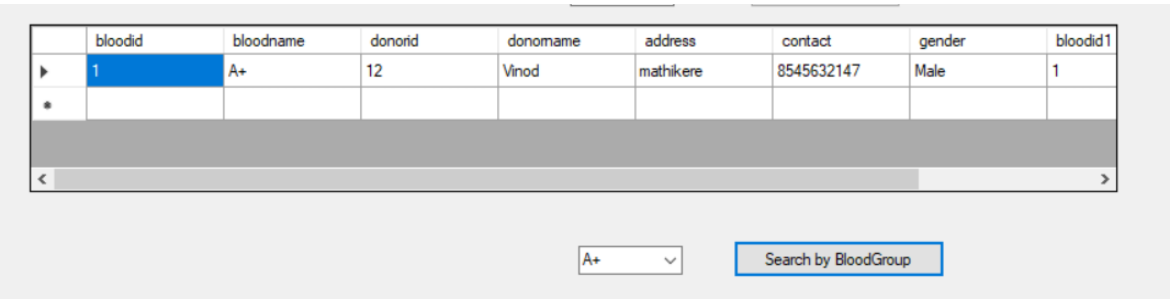
Blood Group Entry form



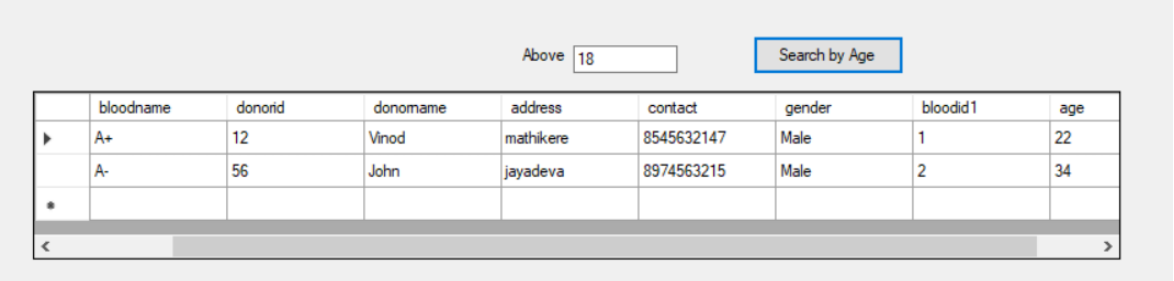
Donor Details Entry Form



Search by Blood Group

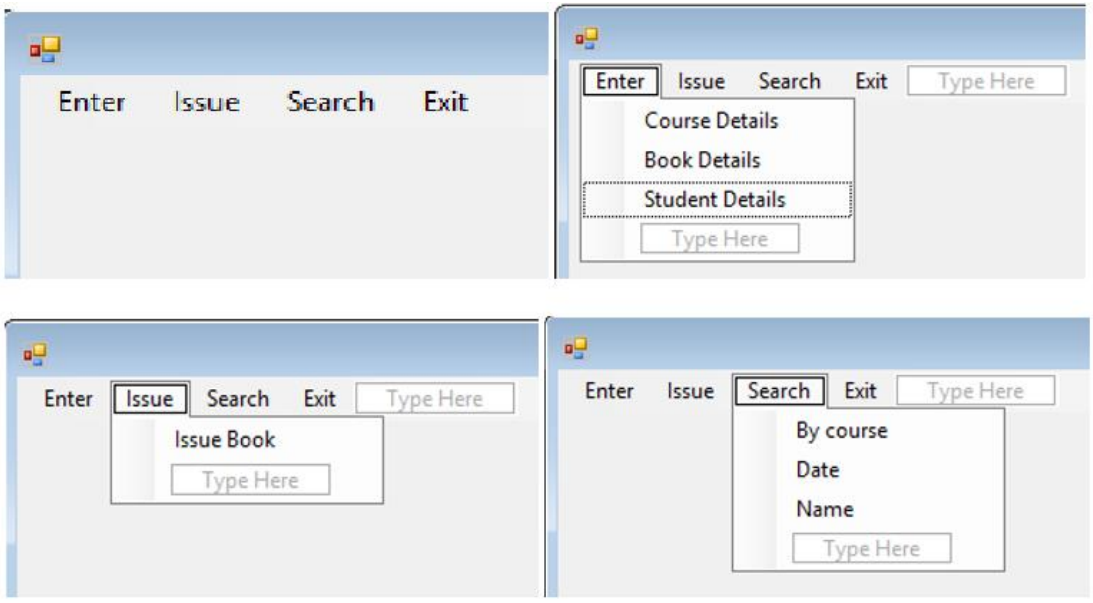


Search by Age



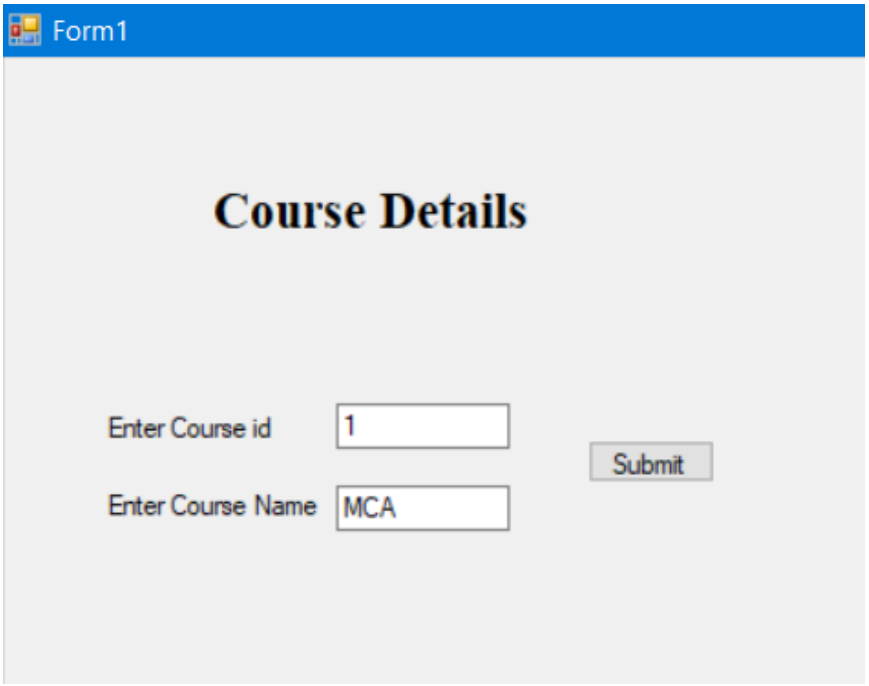
3. Consider the Database STUDENT consisting of following tables:
tbl_Course (CourseID: int, CourseName: string)
tbl_Book (BookID :int, BookTitle: string, Author: string, CourseID: int)
tbl_Student (USN: string, StudName: string, CourseID: int)
tbl_BookIssue(USN: string, BookID: int, IssueDate: Date)
Develop suitable windows application using C#.NET having following options:

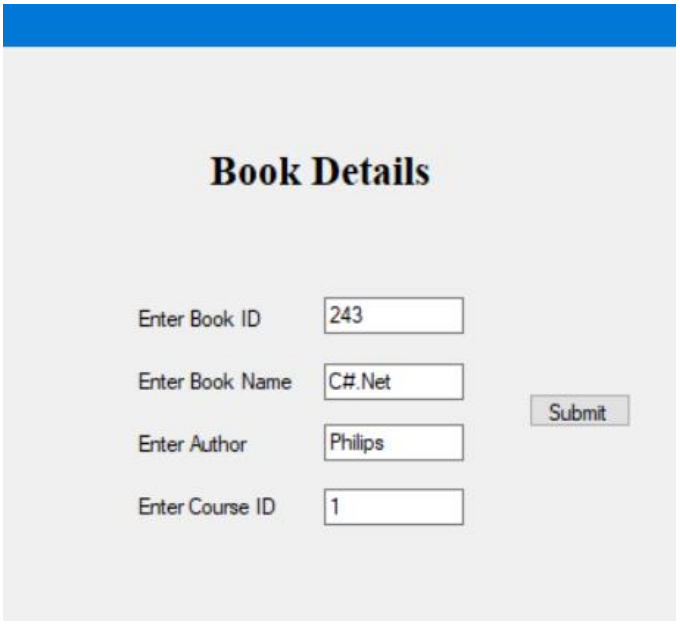
- **New Course Entry.**
- **New Book Entry**
- **New Student Entry**
- **Issue of books to a student.**
- **Generate report (display in a grid) showing all the books belonging to particular course.**
- **Generate report (display in a grid) showing all the books issued on a particular date.**
- **Generate report (display in a grid) showing all the books issued to a particular student.**

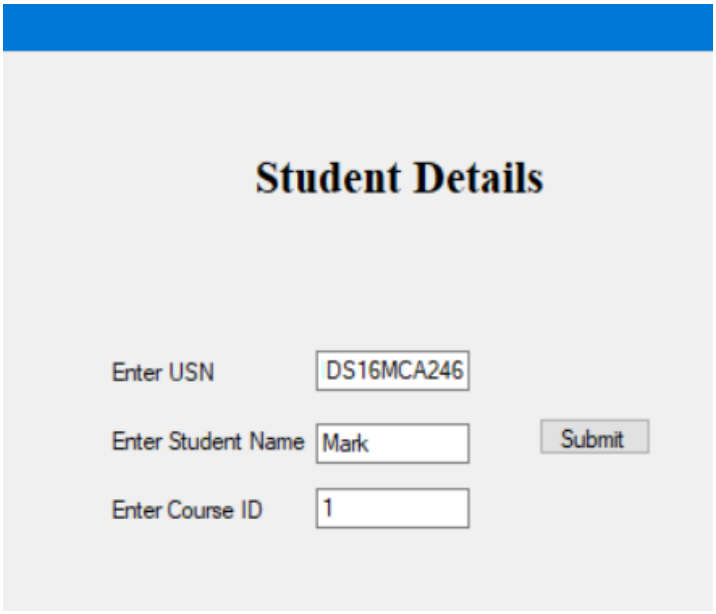


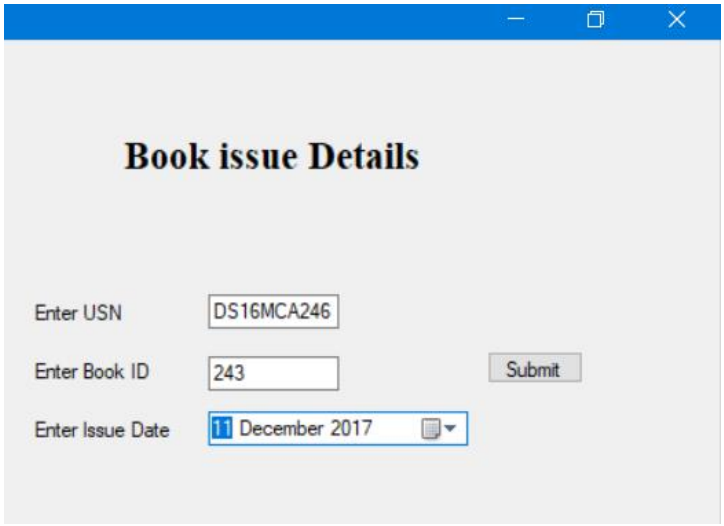

```
private void courseDetailsToolStripMenuItem_Click(object sender, EventArgs e)
    {
       Form2 ob = new Form2();          ob.Show();
    }


    private void bookDetailsToolStripMenuItem_Click(object sender, EventArgs e)      {
       Form3 ob = new Form3();          ob.Show();
    }
```

```
private void studentDetailsToolStripMenuItem_Click(object sender, EventArgs e)
{
   Form4 ob = new Form4();          ob.Show();
}        private void issueBookToolStripMenuItem_Click(object sender, EventArgs e)
{
   Form5 ob = new Form5();          ob.Show();
}

private void byCourseToolStripMenuItem_Click(object sender, EventArgs e)       {
   Form6 ob = new Form6();          ob.Show();
}
private void dateToolStripMenuItem_Click(object sender, EventArgs e)       {
   Form7 ob = new Form7();          ob.Show();
}
private void nameToolStripMenuItem_Click(object sender, EventArgs e)       {
   Form8 ob = new Form8();
   ob.Show();
}
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
   this.Close();

}
```




Book Details
Enter Book ID
243
Enter Book Name
C#.Net
Enter Author
Philips
Enter Course ID
1
Submit

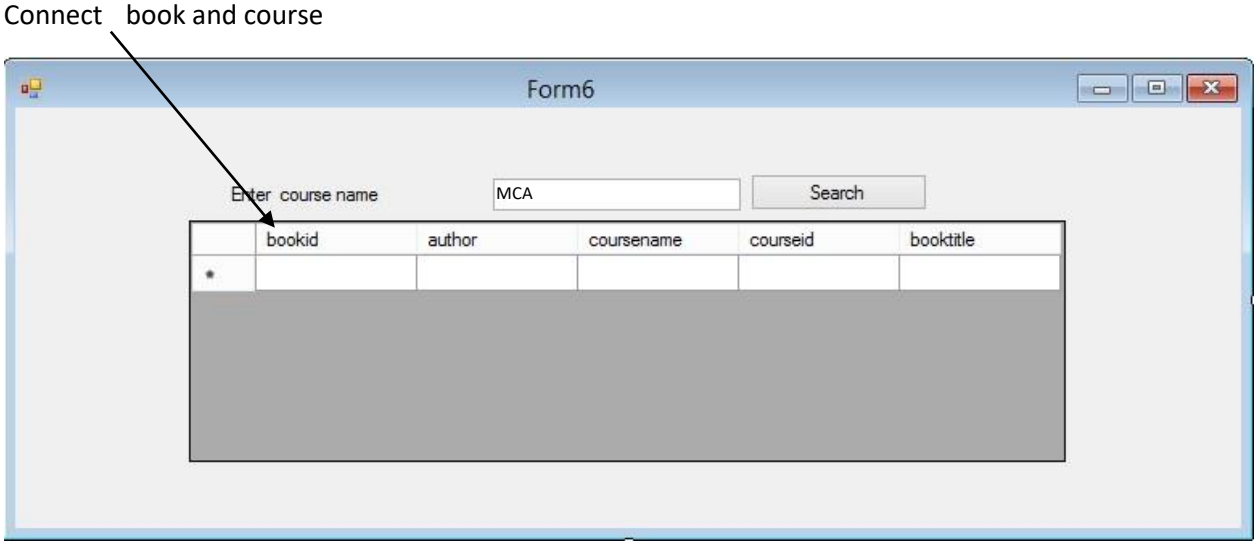

Student Details
Enter USN
DS16MCA246
Enter Student Name
Mark
Enter Course ID
1
Submit


Book issue Details
Enter USN
DS16MCA246
Enter Book ID
243
Enter Issue Date
11 December 2017
Submit

```
namespace prog3
{
   public partial class Form1 : Form
   {
      SqlConnection con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Vinod
Kumar\Documents\student.mdf;Integrated Security=True;Connect Timeout=30");
      SqlCommand cmd;
      public Form1()
      {
         InitializeComponent();
      }



      private void coursesubmit_Click(object sender, EventArgs e)
      {
         cmd = new SqlCommand("insert into tblcourse(courseid,coursename) values('" +
textBox1.Text + "','" + textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Course Inserted Successfully");
      }

      private void booksubmit_Click(object sender, EventArgs e)
      {
         cmd = new SqlCommand("insert into tblbook(bookid,bookname,author,courseid)
values('" + textBox1.Text + "','" + textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Book Inserted Successfully");
      }

      private void studentsubmit_Click(object sender, EventArgs e)
      {
         cmd = new SqlCommand("insert into tblstudent(usn,studname,courseid) values('" +
textBox1.Text + "','" + textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Student Inserted Successfully");
      }

      private void issuesubmit_Click(object sender, EventArgs e)
```

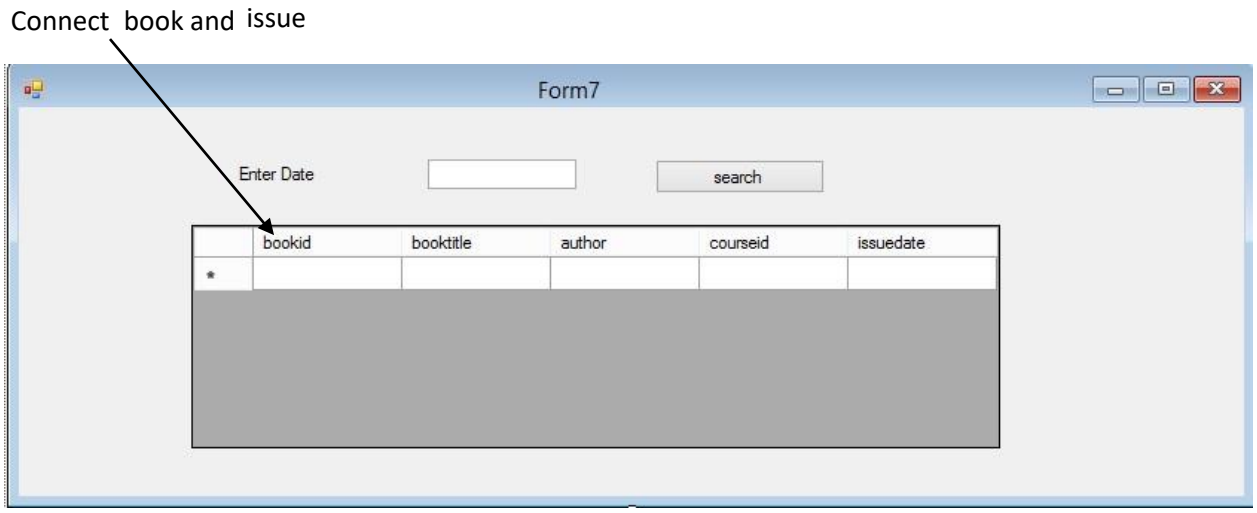
```
      {
         cmd = new SqlCommand("insert into tblbkissue(usn,bookid,issue) values('" +
textBox1.Text + "','" + textBox2.Text + "')", con);
         con.Open();
         cmd.ExecuteNonQuery();
         con.Close();
         MessageBox.Show("Book issued Successfully");
      }


   }
}
```

Search by Course Name



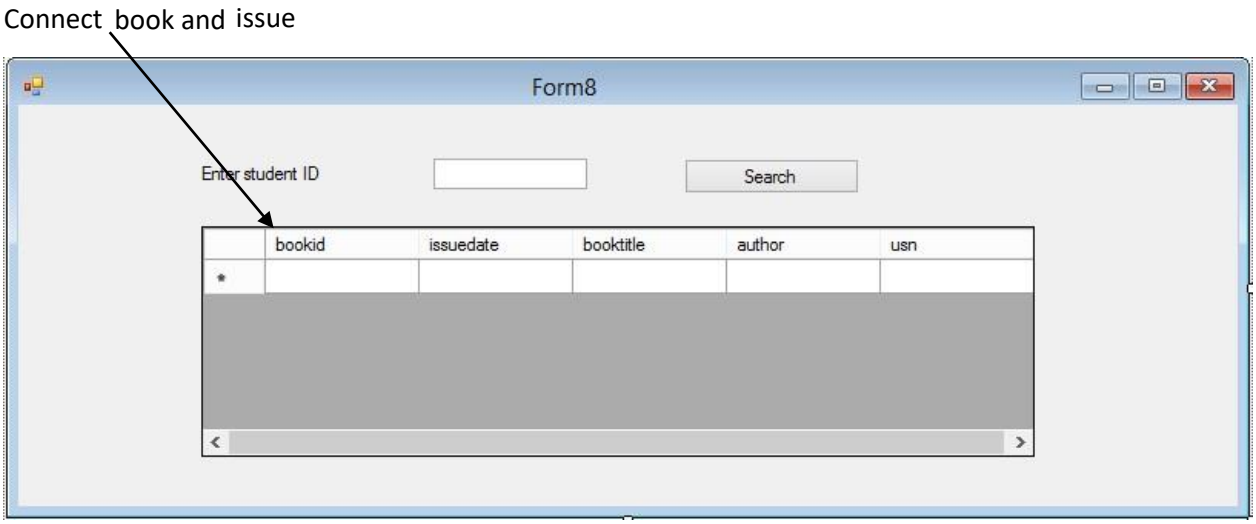

```
      private void SearchCourse_Click(object sender, EventArgs e)
      {
try
         {
            SqlConnection con1 = new SqlConnection(@" ");              con1.Open();
            SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM course c,book b
WHERE c.courseid = b.courseid and c.coursename ='" + textBox1.Text + "'", con1);
DataTable dt = new DataTable();
            sda.Fill(dt);
            dataGridView1.DataSource = dt;
            con1.Close();
         }
         catch (Exception)
         {
            MessageBox.Show("Error ");
}
 }
```

Search by Date




```
    private void SearchDate_Click(object sender, EventArgs e)
    {
 try
        {
            SqlConnection con1 = new SqlConnection(@" ");              con1.Open();
            SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM bissue b,book b1
WHERE b.bookid=b1.bookid and b.issuedate ='" + textBox1.Text + "'", con1);
DataTable dt = new DataTable();
            sda.Fill(dt);
            dataGridView1.DataSource = dt;
            con1.Close();
        }
        catch (Exception)
        {
            MessageBox.Show("Error ");
        }

 }
```

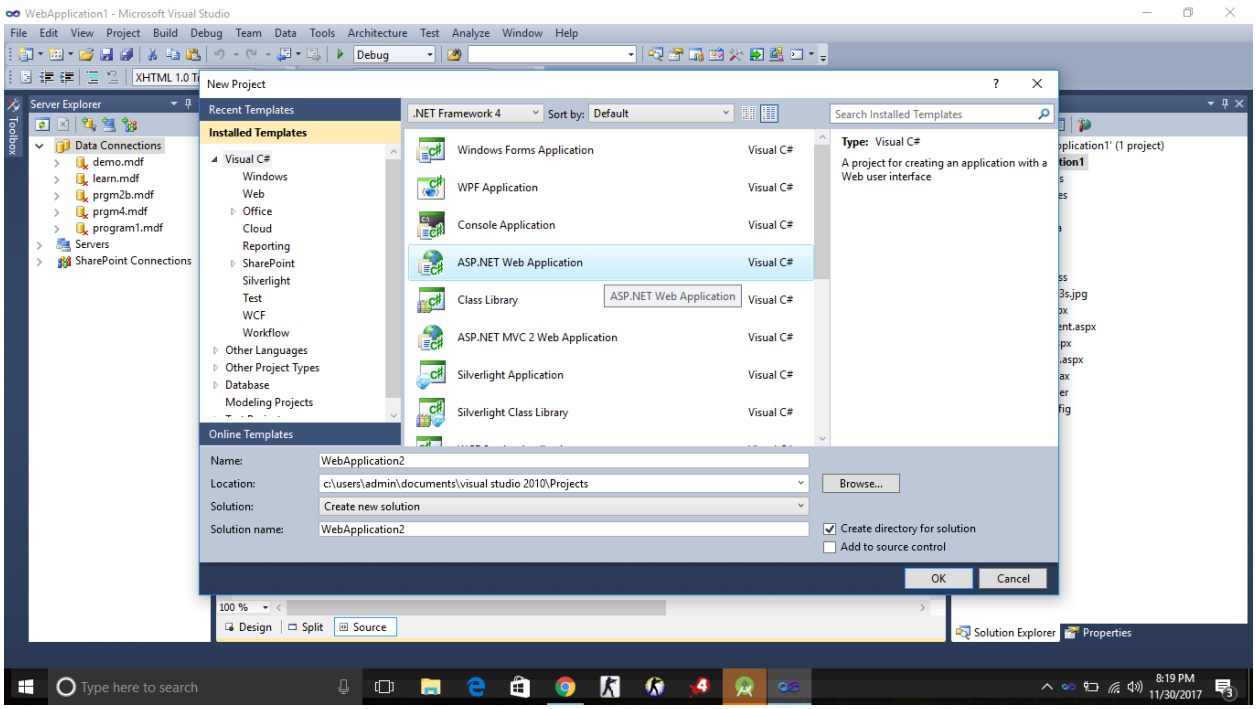
Search by USN




```
    private void SearchUSN_Click(object sender, EventArgs e)
    {
try
        {

            SqlConnection con1 = new SqlConnection(@" ");
con1.Open();
            SqlDataAdapter sda = new SqlDataAdapter("SELECT * FROM bissue b,book
b1,stud s
WHERE b.usn=s.usn and b.bookid=b1.bookid and s.usn ='" + textBox1.Text + "'", con1);
DataTable dt = new DataTable();
            sda.Fill(dt);
            dataGridView1.DataSource = dt;
            con1.Close();
        }
        catch (Exception)
        {
            MessageBox.Show("Error ");
        }

}
```

4. Develop a Web Application using C#.NET and ASP.NET for an educational institution. The master page should consist of Institution Name, Logo and Address. Also, it should provide hyperlinks to Departments, Facilities Available and Feedback. Each department page and facilities page should be designed as static pages. The hyperlinks should navigate to these static pages in the form of Content Pages associated with Master Page designed. The Feedback page should have fields to enter Name, Email and Message with Submit and Cancel Buttons. Database should be created to store these three data.

NEW ->PROJECT->



SITE.MASTER

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="WebApplication1.SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head runat="server">
    <title>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </title>
    <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
    <asp:ContentPlaceHolder ID="HeadContent" runat="server">
    </asp:ContentPlaceHolder>
</head>
```

```
<body>
   <form runat="server">
   <div class="page">
      <div class="header">
      <div class="logo">
         <img src="925717953s.jpg" height="50" width="50"/>
      </div>
         <div class="title">
            <h1>
              Dayananda Sagar College Of Engineering
            </h1>
         </div>
         <div class="address">
        Dayananda Sagar College Of Engineering<br />ks layout bangalore
         </div>
         <div class="loginDisplay">
         </div>
         <div class="clear hideSkiplink">
            <asp:Menu ID="NavigationMenu" runat="server" CssClass="menu"
EnableViewState="false" IncludeStyleBlock="false" Orientation="Horizontal">
               <Items>
                  <asp:MenuItem NavigateUrl="demo.aspx" Text="Home"/>
                  <asp:MenuItem NavigateUrl="department.aspx" Text="Department"/>
                  <asp:MenuItem NavigateUrl="facility.aspx" Text="Facility Available"/>
                  <asp:MenuItem NavigateUrl="~/feedback.aspx" Text="Feedback"/>
               </Items>
            </asp:Menu>
         </div>
      </div>
      <div class="main">
         <asp:ContentPlaceHolder ID="MainContent" runat="server"/>
      </div>
      <div class="clear">
      </div>
   </div>
   <div class="footer">

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

STYLE->SITE.CSS

```
/* DEFAULTS
----------------------------------------------------------*/

/* DEFAULTS
----------------------------------------------------------*/
.logo
{
   height:80px;
   width:80px;
   float:left;

}
.address
{
   float:right;
   padding-top:100px;
   color:White;
}
body
{
   background: #b6b7bc;
   font-size: .80em;
   font-family: "Helvetica Neue", "Lucida Grande", "Segoe UI", Arial, Helvetica, Verdana,
sans-serif;
   margin: 0px;
   padding: 0px;
   color: #696969;
}

a:link, a:visited
{
   color: #034af3;
}

a:hover
{
   color: #1d60ff;
   text-decoration: none;
}

a:active
{
   color: #034af3;
}
```

```
p
{
   margin-bottom: 10px;
   line-height: 1.6em;
}


/* HEADINGS
-----------------------------------------------------------*/

h1, h2, h3, h4, h5, h6
{
   font-size: 1.5em;
   color: #666666;
   font-variant: small-caps;
   text-transform: none;
   font-weight: 200;
   margin-bottom: 0px;
}

h1
{
   font-size: 1.6em;
   padding-bottom: 0px;
   margin-bottom: 0px;
}

h2
{
   font-size: 1.5em;
   font-weight: 600;
}

h3
{
   font-size: 1.2em;
}

h4
{
   font-size: 1.1em;
}

h5, h6
{
```

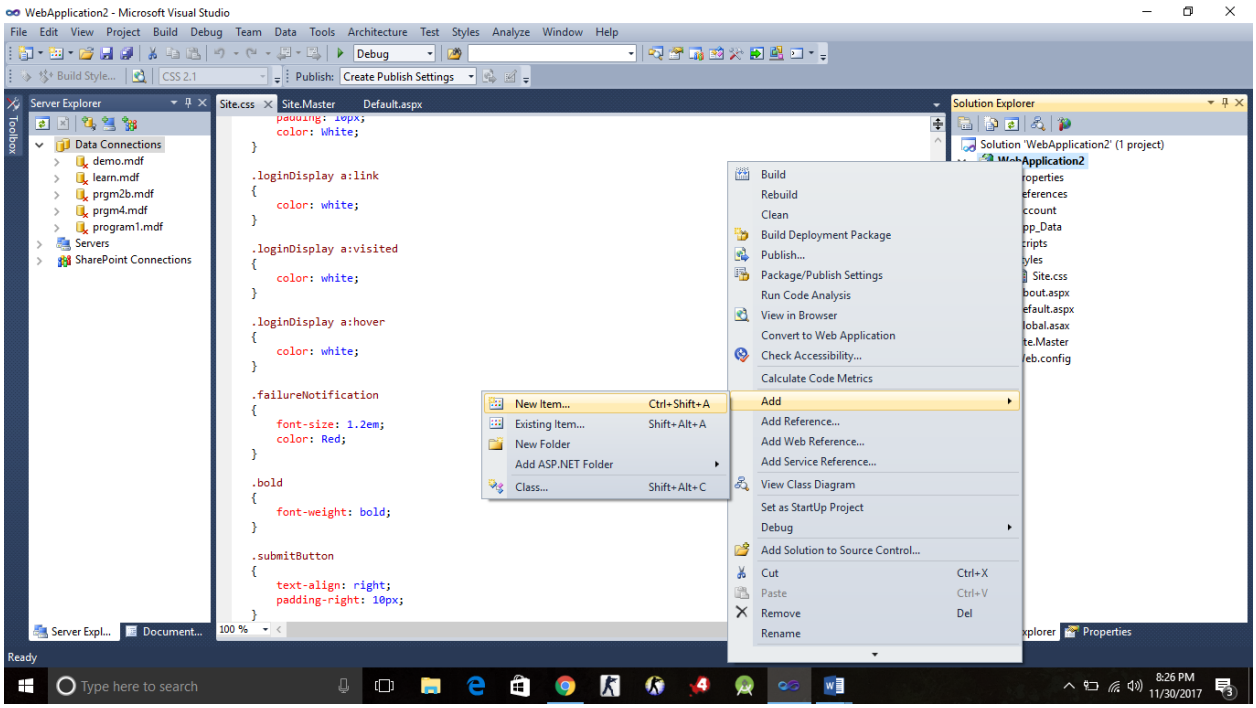
```
    font-size: 1em;
}

/* this rule styles <h1> and <h2> tags that are the
first child of the left and right table columns */
.rightColumn > h1, .rightColumn > h2, .leftColumn > h1, .leftColumn > h2
{
    margin-top: 0px;
}


/* PRIMARY LAYOUT ELEMENTS
-----------------------------------------------------------*/

.page
{
    width: 960px;
    background-color: #fff;
    margin: 20px auto 0px auto;
    border: 1px solid #496077;
}

.header
{
    position: relative;
    margin: 0px;
    padding: 0px;
    background: #4b6c9e;
    width: 100%;
}

.header h1
{
    font-weight: 700;
    margin: 0px;
    padding: 0px 0px 0px 20px;
    color: #f9f9f9;
    border: none;
    line-height: 2em;
    font-size: 2em;
}

.main
{
    padding: 0px 12px;
    margin: 12px 8px 8px 8px;
```

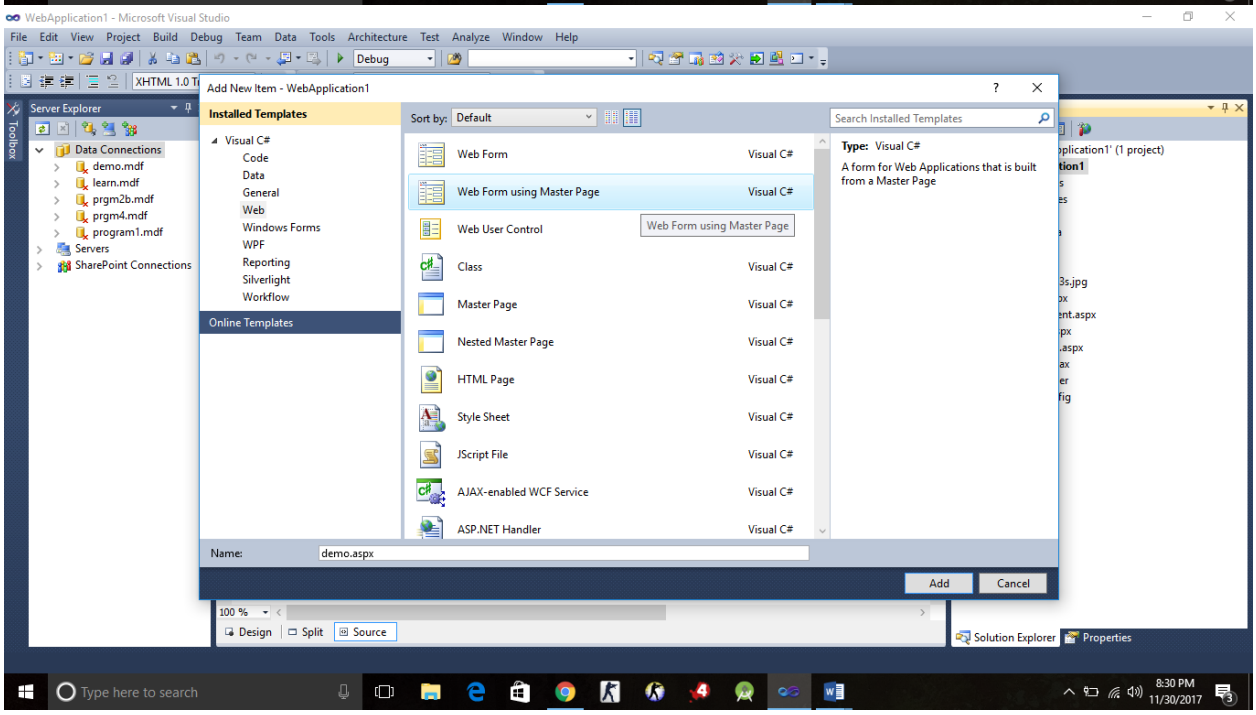
```
   min-height: 420px;
}

.leftCol
{
   padding: 6px 0px;
   margin: 12px 8px 8px 8px;
   width: 200px;
   min-height: 200px;
}

.footer
{
   color: #4e5766;
   padding: 8px 0px 0px 0px;
   margin: 0px auto;
   text-align: center;
   line-height: normal;
}


/* TAB MENU
-----------------------------------------------------------*/

div.hideSkiplink
{
   background-color:#3a4f63;
   width:100%;
}

div.menu
{
   padding: 4px 0px 4px 8px;
}

div.menu ul
{
   list-style: none;
   margin: 0px;
   padding: 0px;
   width: auto;
}

div.menu ul li a, div.menu ul li a:visited
{
   background-color: #465c71;
```

```
    border: 1px #4e667d solid;
    color: #dde4ec;
    display: block;
    line-height: 1.35em;
    padding: 4px 20px;
    text-decoration: none;
    white-space: nowrap;
}

div.menu ul li a:hover
{
    background-color: #bfcbd6;
    color: #465c71;
    text-decoration: none;
}

div.menu ul li a:active
{
    background-color: #465c71;
    color: #cfdbe6;
    text-decoration: none;
}

/* FORM ELEMENTS
----------------------------------------------------------*/

fieldset
{
    margin: 1em 0px;
    padding: 1em;
    border: 1px solid #ccc;
}

fieldset p
{
    margin: 2px 12px 10px 10px;
}

fieldset.login label, fieldset.register label, fieldset.changePassword label
{
    display: block;
}

fieldset label.inline
{
    display: inline;
```

```
}

legend
{
   font-size: 1.1em;
   font-weight: 600;
   padding: 2px 4px 8px 4px;
}

input.textEntry
{
   width: 320px;
   border: 1px solid #ccc;
}

input.passwordEntry
{
   width: 320px;
   border: 1px solid #ccc;
}

div.accountInfo
{
   width: 42%;
}

/* MISC
----------------------------------------------------------*/

.clear
{
   clear: both;
}

.title
{
   display: block;
   float: left;
   text-align: left;
   width: auto;
}

.loginDisplay
{
   font-size: 1.1em;
   display: block;
```

```
    text-align: right;
    padding: 10px;
    color: White;
}

.loginDisplay a:link
{
    color: white;
}

.loginDisplay a:visited
{
    color: white;
}

.loginDisplay a:hover
{
    color: white;
}

.failureNotification
{
    font-size: 1.2em;
    color: Red;
}

.bold
{
    font-weight: bold;
}

.submitButton
{
    text-align: right;
    padding-right: 10px;
}
```

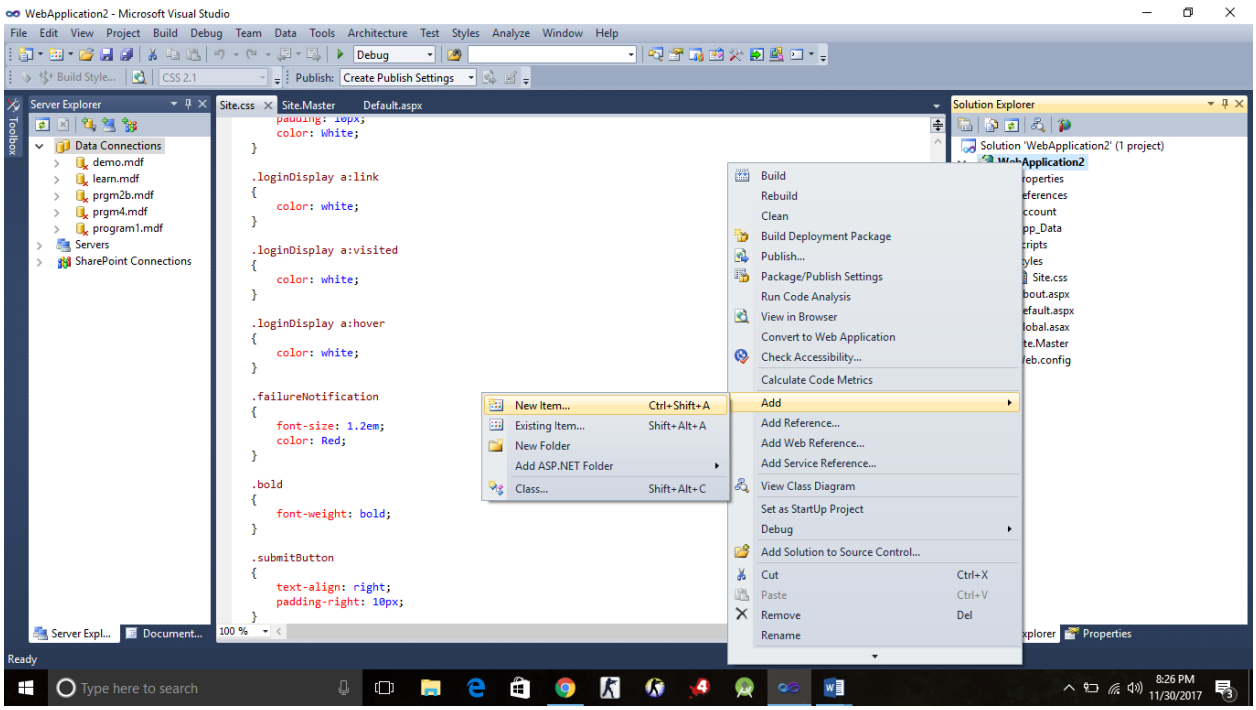


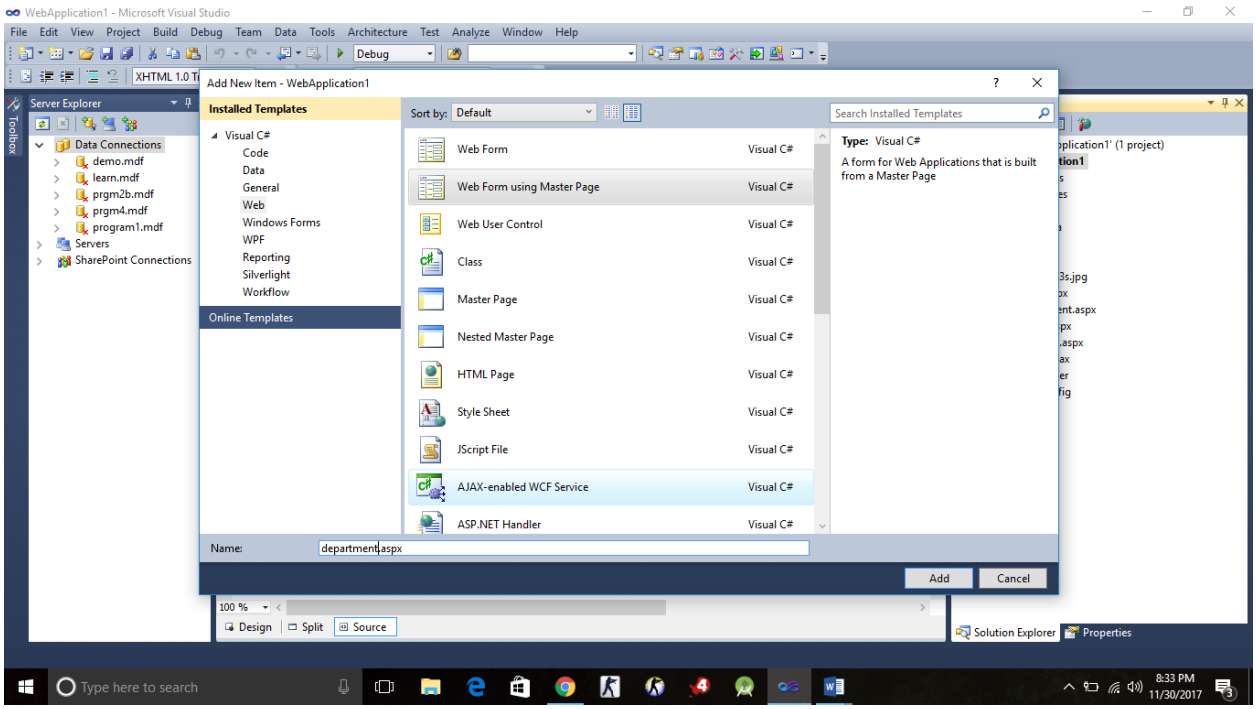


Demo.aspx

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="demo.aspx.cs" Inherits="WebApplication1.demo" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
HOME

```
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<ul>
<li><a href="department.aspx">VISIT TO DEPARTEMNT PAGE</a></li>
<li><a href="facility.aspx">VISIT TO FACILITY PAGE</a></li>
<li><a href="feedback.aspx">VISIT TO FEEDBACK PAGE</a></li>
</ul>
</asp:Content>
```





Department.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="department.aspx.cs"
Inherits="WebApplication1.department" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
Department
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<ul>
<li>MCA</li>
<li>MBA</li>
<li>B.TECH</li>
</ul>
</asp:Content>
```

Same add new item

Facility.aspx

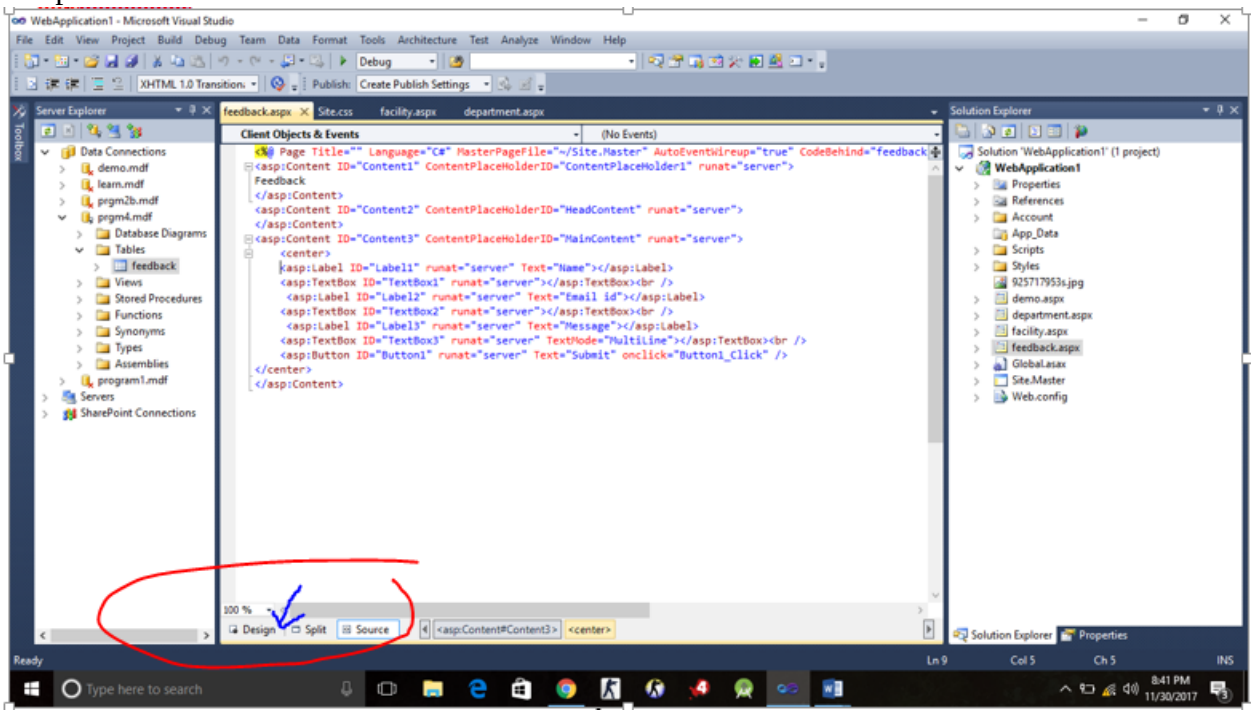
```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="facility.aspx.cs"
Inherits="WebApplication1.facility" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
Facility Available
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<ul>
<li>WIFI CAMPUS</li>
<li>PLAYGROUND</li>
<li>CENTRAL LIBRARY</li></ul>
</asp:Content>
```

Add new item

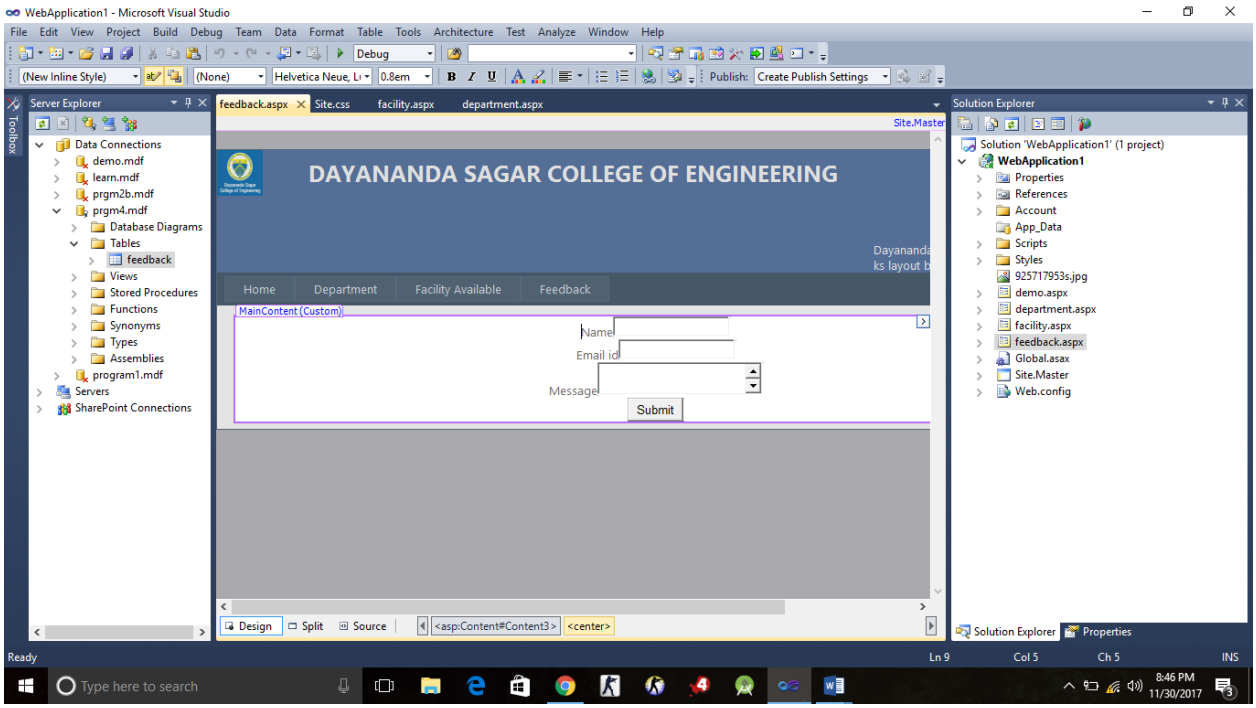
Feedback.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="feedback.aspx.cs"
Inherits="WebApplication1.feedback" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
```

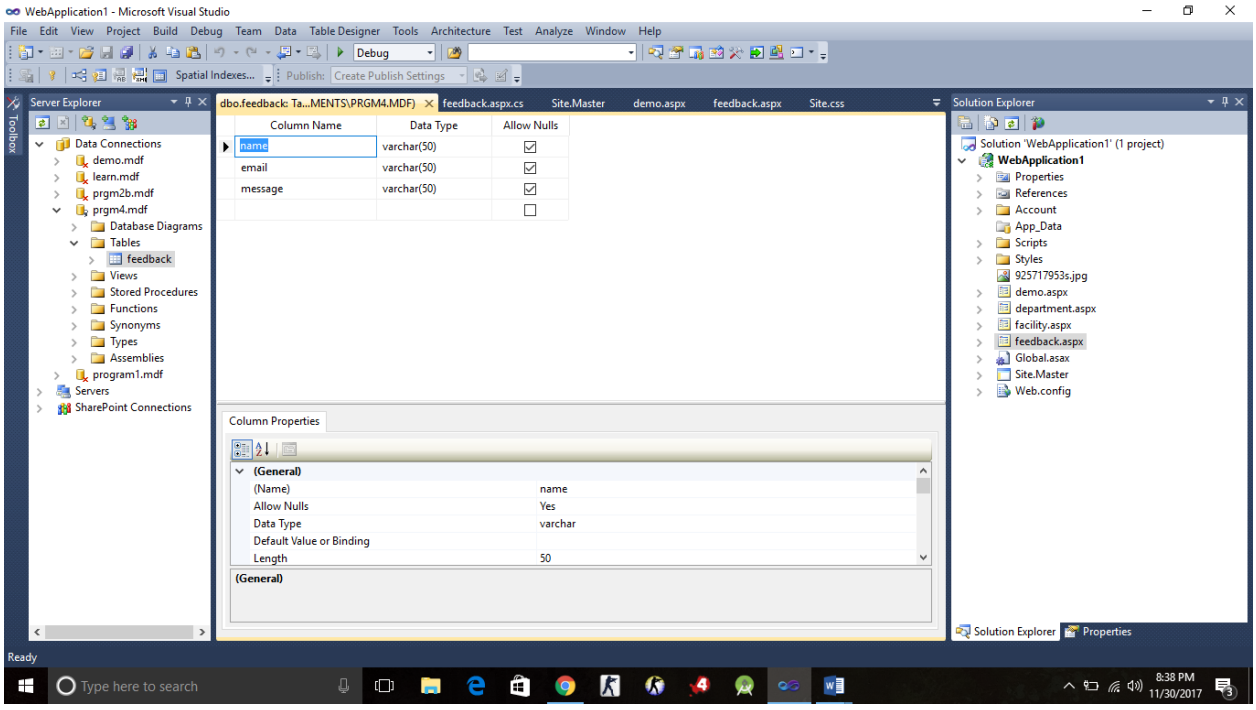
```
Feedback
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
   <center>
   <asp:Label ID="Label1" runat="server" Text="Name"></asp:Label>
   <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br />
    <asp:Label ID="Label2" runat="server" Text="Email id"></asp:Label>
   <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox><br />
    <asp:Label ID="Label3" runat="server" Text="Message"></asp:Label>
   <asp:TextBox ID="TextBox3" runat="server"
TextMode="MultiLine"></asp:TextBox><br />
   <asp:Button ID="Button1" runat="server" Text="Submit" onclick="Button1_Click" />
</center>
</asp:Content>
```

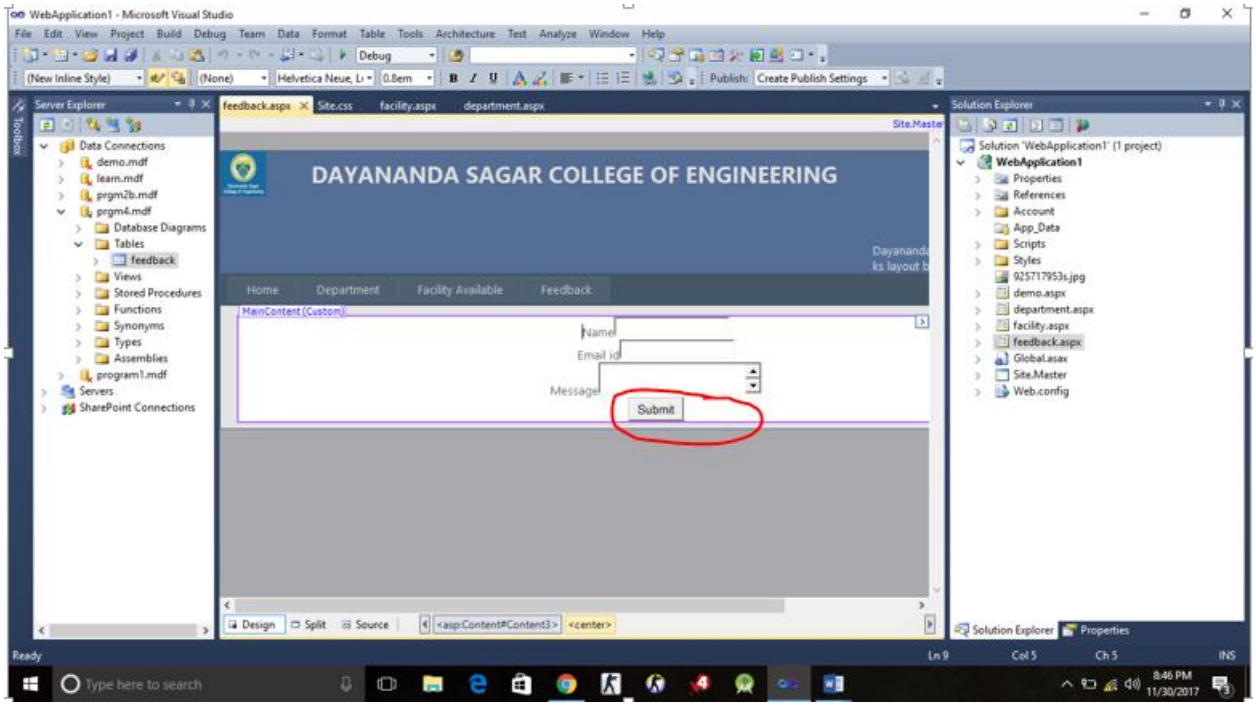


In this page goto design part



Create new database connection in that add new table and name it as feedback





On submit button

```
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;

namespace WebApplication1
{
   public partial class feedback : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Button1_Click(object sender, EventArgs e)
      {
         try
         {

            SqlConnection con = new SqlConnection(@"Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Admin\Documents\prgm4.mdf;Integr
ated Security=True;Connect Timeout=30;User Instance=True");
            con.Open();
```

```
            SqlCommand cmd = new SqlCommand("insert into
feedback(name,email,message)values('" + TextBox1.Text + "','" + TextBox2.Text + "','" +
TextBox3.Text + "')", con);
            cmd.ExecuteNonQuery();

            Response.Write("<script>alert('submited');</script>");
            con.Close();
        }
        catch (Exception ec)
        {
            Response.Write(ec.Message);
        }
    }
  }
}
```