MUTHAYAMMAL
POLYTECHNIC
INSTITUTION
A UNIT OF VANETRA GROUP | Learn. Lead

(Approved by AICTE,New Delhi & Affiliated to Directorate of Technical Education, Chennai)

1052-Diploma in Computer Engineering

III Year, V-Semester

SCHEME - M

35257- COMPONENT BASED TECHNOLOGY LAB MANUAL

MUTHAYAMMAL
POLYTECHNIC INSTITUTION
Rasipuram, Namakkal Dt, Tamil Nadu, India-637 408

Phone : 04287-225137, 221837, Fax : 04287-221937
E-mail : info@muthayammal.org

www.muthayammal.org

35257- Component Based Technology Practical

LAB EXERCISE

PART – A

VB.NET PROGRAMMING

1. Accept a character from console and check the case of the character.
2. Write a program to accept any character from keyboard and display whether it is vowelor not.
3. Write a VB.Net program to accept a string and convert the case of the characters.
4. Develop a menu based VB.Net application to implement a text editor with cut, copy, paste, save and close operations.
5. Write a program to implement a calculator with memory and recall operations.
6. Develop a form in VB.NET to pick a date from Calendar control and display the day, month, and year details in separate text boxes.
7. Develop a VB.Net application to perform timer based quiz of 10 questions.
8. Develop a VB.Net application using the File and Directory controls to implement a common dialog box.
9. Develop a database application to store the details of students using ADO.NET
10. Develop a database application using ADO.NET to insert, modify, update and deleteoperations.
11. Develop a VB.Net application using Datagrid to display records.
12. Develop a VB.Net application using Datagrid to add, edit and modify records.

PART – B

ASP.NET and XML PROGRAMMING

1. Create a simple ASP.NET page to Output Text with a form, two HTML text boxes, an HTML button, and an HTML <span> element. Create an event procedure for thebutton.
2. Create a web application in ASP.NET using three different controls to the ASP.NETpage for reserving rooms in hotel. The three controls are a button control, a label control, and a drop down list control.
3. Create a application for Accessing a SQL Database by Using ADO.NET by connecting to the SQL Server database and call a stored procedure. You then displaythe data in a Repeater control.
4. Develop a web application to read the details of the selected country stored in XML database and display back to the user using Webcontrol
5. Develop a web application to read an XML document containing subject, mark scored, year of passing into a Dataset

EX.NO:1 CHECKING THE CASE OF THE GIVEN CHARACTER

AIM

To write a C#.NET program to accept a character from console and check the case of the character.

PROCEDURE

1. Open Microsoft Visual studio and create a new console application from C# language.

2.Give application name and select the location to save the application in the Solution explorer.

3. Edit the code in the application window.

4. Run the application by pressing F5 key or clicking Debug button.

PROGRAM

```
using System;
public class Exno1
{
   static void Main()
   {
      Console.Write("Input a character: ");
      char ch = (char)Console.Read();
           if (Char.IsUpper(ch))
         {
            Console.WriteLine("\nThe character is uppercase.");
         }
         else
         {
            Console.WriteLine("\nThe character is lowercase.");
         }
         Console.ReadKey();
      }
  }
```

OUTPUT

```
Input a character: a
The character is lowercase.
Input a character: A
The character is uppercase.
```

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO:2 CHECK THE GIVEN CHARACTER IS VOWEL OR NOT

AIM

To write a C#.NET program to accept any character from keyboard and display whether it is vowel or not.

PROCEDURE

1.Open Microsoft Visual studio and create a new console application from C# language.

2.Give application name and select the location to save the application in the Solution explorer.

3. Edit the code in the application window.

4. Run the application by pressing F5 key or clicking Debug button.

PROGRAM

```
using System;
static class Exno2
{
   static string c;
   public static void Main()
   {
      Console.WriteLine("Enter a character : ");
      c = Console.ReadLine();
      if (c == "a" | c == "A" | c == "e" | c == "E" | c == "i" | c == "I" | c == "o" | c == "O" | c ==
         "u" | c == "U")
      {
         Console.WriteLine("Given character is vowel");
      }
      else
      {
         Console.WriteLine("Given character is not vowel");
      }
      Console.ReadLine();
   }
}
```

OUTPUT

```
Enter a character :
a
Given character is vowel
Enter a character :
b
Given character is not vowel
```

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO:3 CHANGING THE CASE OF THE GIVEN STRING

AIM

To write a C#.NET program to accept a string and convert the case of the character.

PROCEDURE

1.Open Microsoft Visual studio and create a new console application from C# language.

2.Give application name and select the location to save the application in the Solution explorer.

3. Edit the code in the application window.

4. Run the application by pressing F5 key or clicking Debug button.

PROGRAM

```
using System;
public class Exno3
{
   public static void Main()
   {
      string  str1;
      char[] arr1;
      int l, i;
      l = 0;
      char ch;
      Console.Write("Input the string : ");
      str1 = Console.ReadLine();
      l = str1.Length;
      arr1 = str1.ToCharArray(0, l); // Converts string into char array.
      Console.Write("\nAfter case conversion, the string is : ");
      for (i = 0; i < l; i++)
      {
         ch = arr1[i];
         if (ch >= 65 && ch <= 90)
            Console.Write(Char.ToLower(ch));
         else if (ch >= 97 && ch <= 122)
            Console.Write(Char.ToUpper(ch));
      }
      Console.Write("\n\n");
      Console.ReadKey();
   }
}
```

OUTPUT

```
Input the string : RaJa
After case conversion, the string is : rAjA
```

RESULT:

Thus the above program has been executed and verified successfully.

EX.NO:4 ADDING MENU TO THE FORM

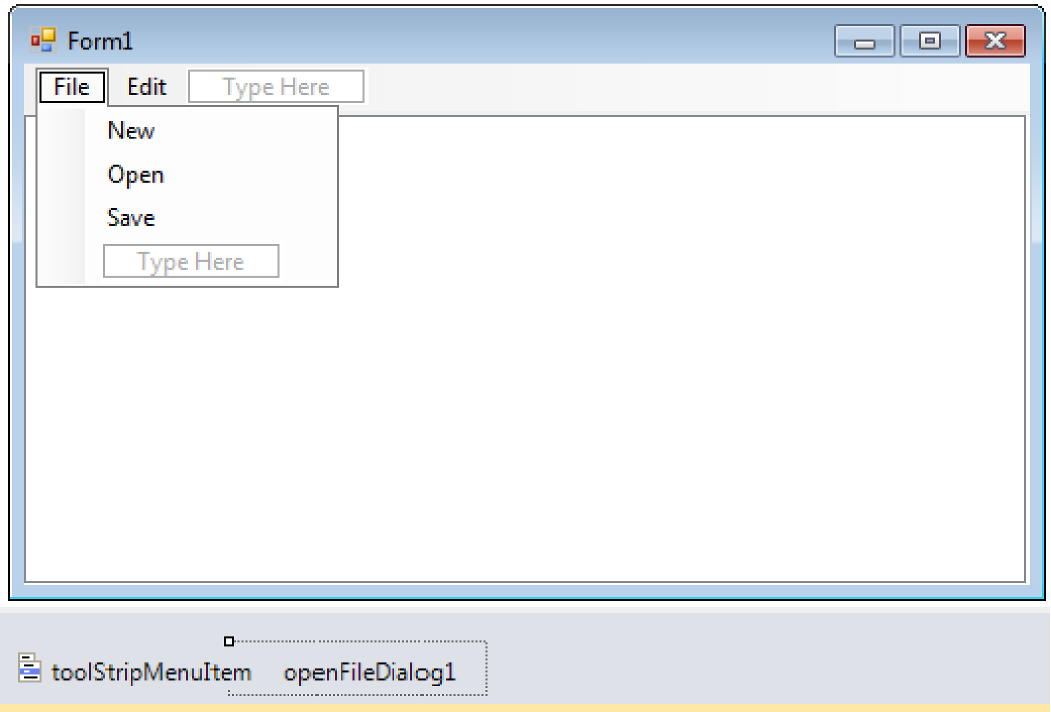
AIM

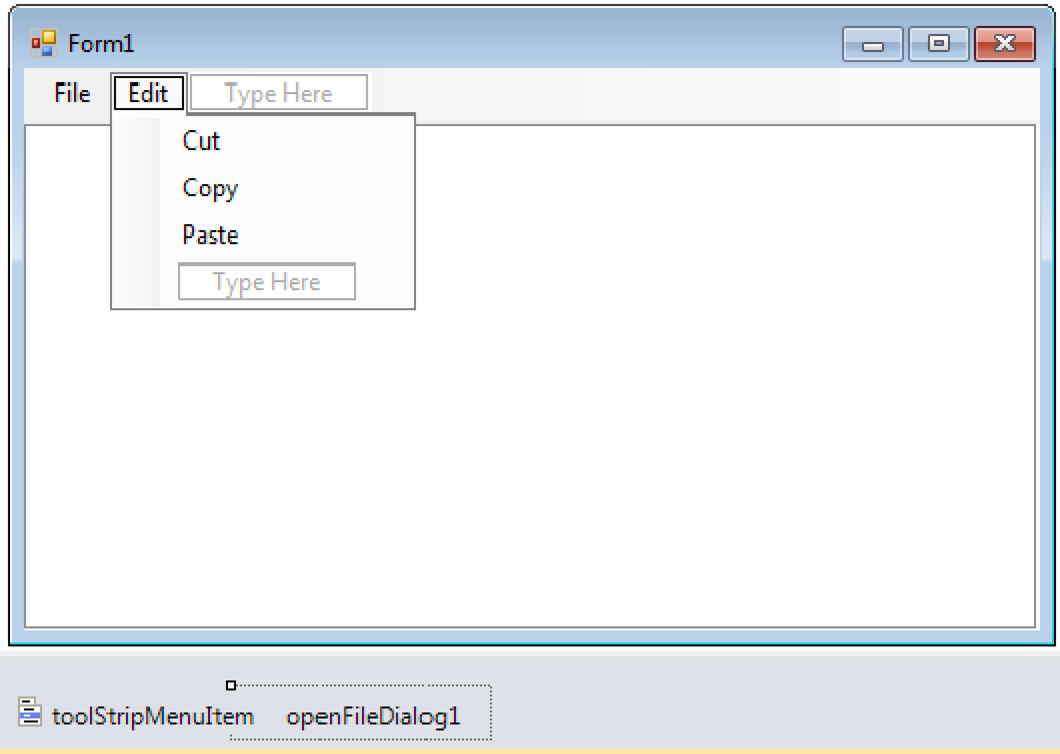
To develop a menu based C#.NET application to implement a text editor with cut, copy, paste, saved and close operations.

PROCEDURE

1.Open Microsoft Visual studio and create a new Window Application
2.Design a form
3.Edit the code in the event of all controls
4.Run the application

FORM DESIGN



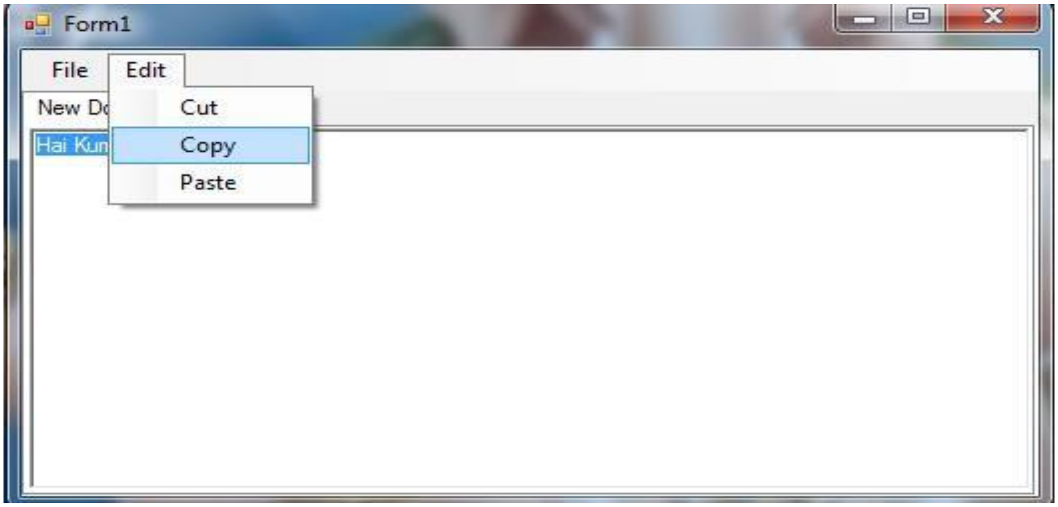


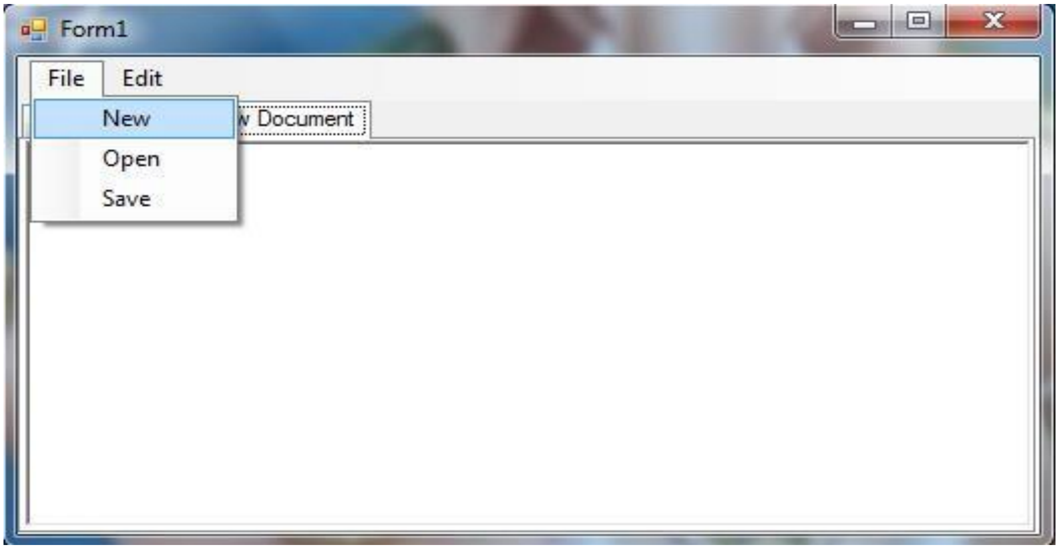
Control	Properties	
	Name	Text
toolStrip	toolStripMenuItem	menuStrip1
	fileToolStripMenuItem	File
	newToolStripMenuItem	New
	openToolStripMenuItem	Open
	saveToolStripMenuItem	Save
	editToolStripMenuItem	Edit
	cutToolStripMenuItem	Cut
	copyToolStripMenuItem	Copy
	pasteToolStripMenuItem	Paste
TabControl	tabControl1	Delete tabs and change Dock to Fill
openFileDialog1	openFileDialog1	

CODING:

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

namespace Notepad
{
   public partial class Form1 : Form
   {
      RichTextBox rtb;
      public Form1()
      {
         InitializeComponent();
      }

      private RichTextBox GetRichTextBox()
      {
         rtb = null; //making it initially null
         TabPage tp = tabControl1.SelectedTab;
         if (tp != null)
         {
            rtb = tp.Controls[0] as RichTextBox;
         }
         return rtb;
      }
      private void toolStripMenuItem1_Click(object sender, EventArgs e)
      {

      }
      private void newToolStripMenuItem_Click_1(object sender, EventArgs e)
      {
         TabPage tp = new TabPage("New Document");
         RichTextBox rtb = new RichTextBox();
         rtb.Dock = DockStyle.Fill;
         tp.Controls.Add(rtb);
         tabControl1.TabPages.Add(tp);
      }
      private void cutToolStripMenuItem_Click_1(object sender, EventArgs e)
      {
         GetRichTextBox().Cut();
      }

        private void copyToolStripMenuItem_Click_1(object sender, EventArgs e)
      {
           GetRichTextBox().Copy();
      }

      private void pasteToolStripMenuItem_Click_1(object sender, EventArgs e)
      {
            GetRichTextBox().Paste();
        }

        private void openToolStripMenuItem_Click_1(object sender, EventArgs e)
        {
            Stream myStream;
            if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                if ((myStream = openFileDialog1.OpenFile()) != null)
                {
                    string filename = openFileDialog1.FileName
                    string readfiletext = File.ReadAllText(filename);
                    TabPage tp = new TabPage("New Document");
                    RichTextBox rtb = new RichTextBox();
                    rtb.Dock = DockStyle.Fill;
                    tp.Controls.Add(rtb);
                    tabControl1.TabPages.Add(tp);
                    rtb.Text = readfiletext;
                }

            }
        }
        private void saveToolStripMenuItem_Click_1(object sender, EventArgs e)
        {
            SaveFileDialog savefile = new SaveFileDialog();
            savefile.Filter = "*.txt(textfile)|*.txt";
            if (savefile.ShowDialog() == DialogResult.OK)
            {
                rtb.SaveFile(savefile.FileName, RichTextBoxStreamType.PlainText);
            }
        }
      }
}
```

OUTPUT:





RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:5 SIMPLE CALCULATOR

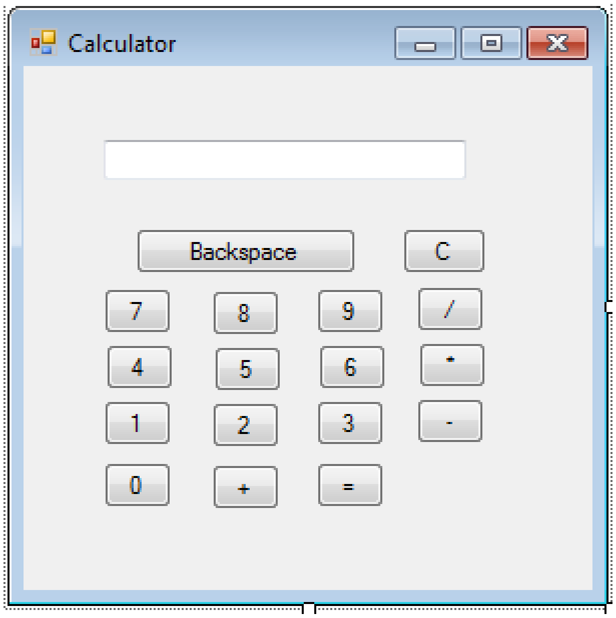
AIM

To develop a C#.NET application to implement a simple calculator.

PROCEDURE

1.Open Microsoft Visual studio and create a new Window Application
2.Design a form
3.Edit the code in the event of all controls
4.Run the application

FORM DESIGN



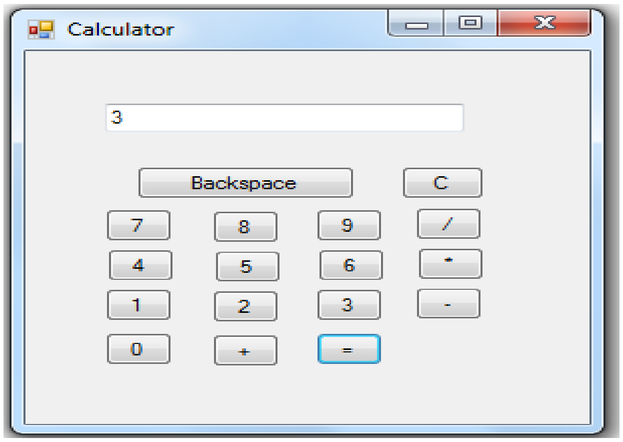
Control	Properties	
	Name	Text
TextBox	txtResult	
Button	btn0	0
Button	btn1	1
Button	btn2	2
Button	btn3	3
Button	btn4	4
Button	btn5	5
Button	btn6	6
Button	btn7	7
Button	btn8	8
Button	btn9	9

Button	btnPlus	+
Button	btnMinus	-
Button	btnMult	*
Button	btnDiv	/
Button	btnEqual	=
Button	btnClear	C
Button	btnBackspace	Backspace

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

namespace Calculator
{
    public partial class Calculator : Form
    {
        string Result;
        double EndResult;
        string CheckForManipulation;

        public Calculator()
        {
            InitializeComponent();
        }
        private void Calculator_Load(object sender, EventArgs e)
        {
            btn0.Click += new EventHandler(digitButton_Click);
            btn1.Click += new EventHandler(digitButton_Click);
            btn2.Click += new EventHandler(digitButton_Click);
            btn3.Click += new EventHandler(digitButton_Click);
            btn4.Click += new EventHandler(digitButton_Click);
            btn5.Click += new EventHandler(digitButton_Click);
            btn6.Click += new EventHandler(digitButton_Click);
            btn7.Click += new EventHandler(digitButton_Click);
            btn8.Click += new EventHandler(digitButton_Click);
            btn9.Click += new EventHandler(digitButton_Click);
            btnPlus.Click += new EventHandler(digitCalculate_Click);
            btnMinus.Click += new EventHandler(digitCalculate_Click);
            btnMult.Click += new EventHandler(digitCalculate_Click);
            btnDiv.Click += new EventHandler(digitCalculate_Click);
            btnBackspace.Click += new EventHandler(btnBackspace1_Click);
            btnClear.Click += new EventHandler(btnClear1_Click);
            btnEqual.Click += new EventHandler(btnEqual1_Click);

        }

        void digitButton_Click(object sender, EventArgs e)
        {
            Button ButtonThatWasPushed = (Button)sender;
            txtResult.Text += ButtonThatWasPushed.Text;

        }


        void digitCalculate_Click(object sender, EventArgs e)
        {
            Button ButtonThatWasPushed = (Button)sender;
            string ButtonText = ButtonThatWasPushed.Text;


            if (txtResult.Text != string.Empty)
            {
                double valueHolder1 = Convert.ToDouble(txtResult.Text);
            }

            if (ButtonText == "+")
            {
                Result = txtResult.Text;
                CheckForManipulation = "Add";
                txtResult.Clear();
                txtResult.Focus();
            }
            else if (ButtonText == "-")
            {

                Result = txtResult.Text;
                CheckForManipulation = "Substract";
                txtResult.Clear();
                txtResult.Focus();

            }
            else if (ButtonText == "*")
            {

                Result = txtResult.Text;
                CheckForManipulation = "Multiply";
                txtResult.Clear();
                txtResult.Focus();
            }
            else if (ButtonText == "/")
            {

                Result = txtResult.Text;
                CheckForManipulation = "Division";
                txtResult.Clear();
                txtResult.Focus();
            }

         }

        void btnEqual1_Click(object sender, EventArgs e)
         {

            if (txtResult.Text != string.Empty && Result != string.Empty)
            {
                double valueHolder2 = Convert.ToDouble(txtResult.Text);
                double chk = Convert.ToDouble(Result);
                if (CheckForManipulation == "Add")
                {

                    EndResult = chk + valueHolder2;
                    txtResult.Text = EndResult.ToString();

                }
                else if (CheckForManipulation == "Substract")
                {

                    EndResult = chk - valueHolder2;
                    txtResult.Text = EndResult.ToString();
                }
                else if (CheckForManipulation == "Multiply")
                {

                    EndResult = chk * valueHolder2;
                    txtResult.Text = EndResult.ToString();
                }
                else if (CheckForManipulation == "Division")
                {

                    if (valueHolder2 == 0)
                    {
                        txtResult.Text = "Cannot divide by Zero";
                        return;
                    }
                    EndResult = chk / valueHolder2;
                    txtResult.Text = EndResult.ToString();
```

```
                    }
                }
            }

      void btnClear1_Click(object sender, EventArgs e)
        {
            txtResult.Clear();
            txtResult.Focus();
        }
        void btnBackspace1_Click(object sender, EventArgs e)
        {
            if (txtResult.Text != string.Empty)
            {
                int txtlength = txtResult.Text.Length;
                if (txtlength != 1)
                {
                    txtResult.Text = txtResult.Text.Remove(txtlength - 1);
                }
                else
                {
                    txtResult.Text = 0.ToString();
                }

            }

        }
    }
}
```

OUTPUT:



3

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:6 DISPLAY DAY, MONTH, YEAR FROM CALENDAR CONTROL

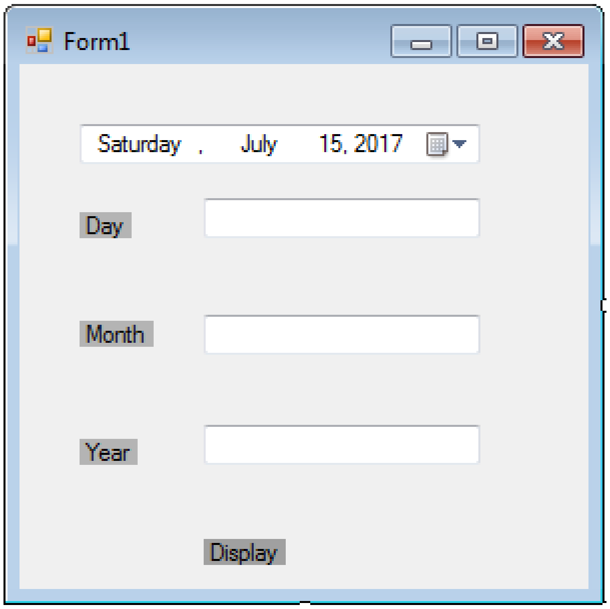
AIM

To develop a form in C#.NET to pick a date form calendar control and display the Day,month,year in separate text boxes.

PROCEDURE

1.Open Microsoft Visual studio and create a new Window Application
2.Design a form
3.Edit the code in the event of all controls
4.Run the application

FORM DESIGN

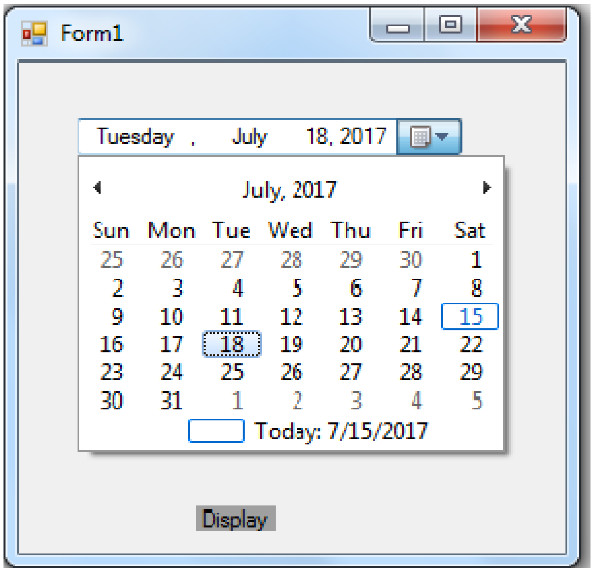


Control	Properties	
	Name	Text
DateTimePicker		
Label	label1	Day
Label	label2	Month
Label	label3	Year
Label	label4	Display
TextBox	textBox1	
TextBox	textBox2	
TextBox	textBox3	

CODING:

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

namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
        {

        }
        private void label4_Click(object sender, EventArgs e)
        {
            textBox1.Text = dateTimePicker1.Value.Day.ToString();
            textBox2.Text = dateTimePicker1.Value.Month.ToString();
            textBox3.Text = dateTimePicker1.Value.Year.ToString();

       }
    }
}
```

OUTPUT:

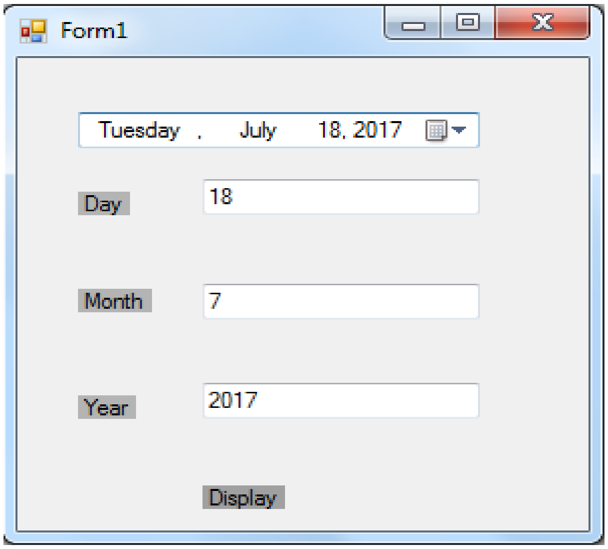


Tuesday , July 18, 2017

July, 2017

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

Today: 7/15/2017



Tuesday , July 18, 2017

Day 18

Month 7

Year 2017

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:7

QUIZ PROGRAM

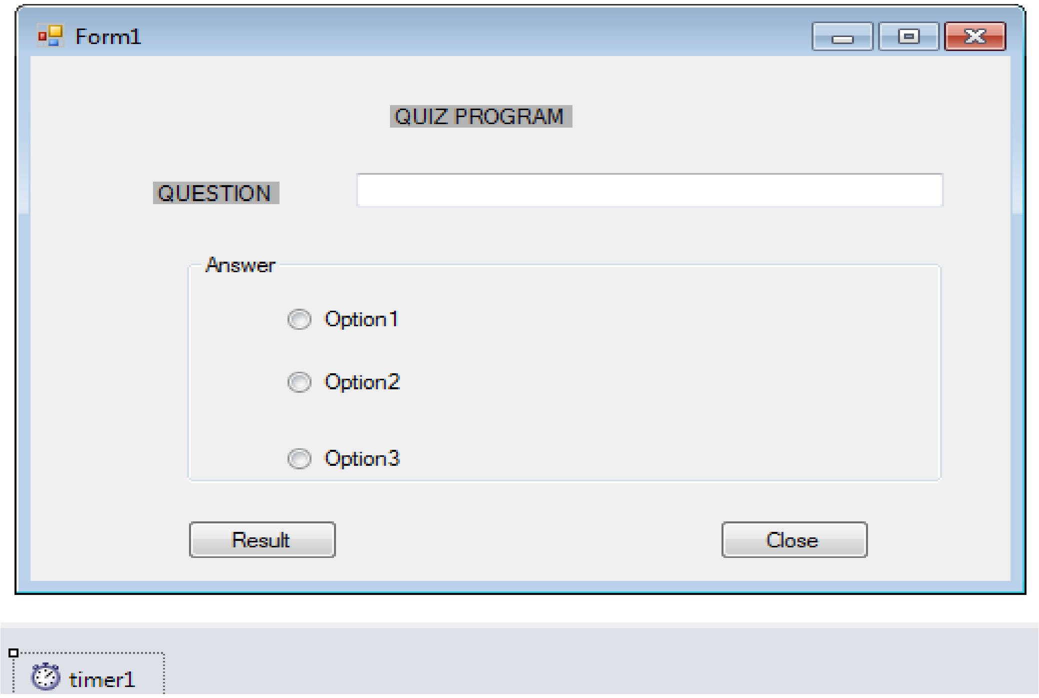
AIM

To develop a C#.NET application to perform timer based quiz of 5 questions.

PROCEDURE

1. Open Microsoft Visual studio and create a new Window Application
2. Design a form
3. Edit the code in the event of all controls
4. Set the interval property of the timer by 5000.
5. Run the application

FORM DESIGN



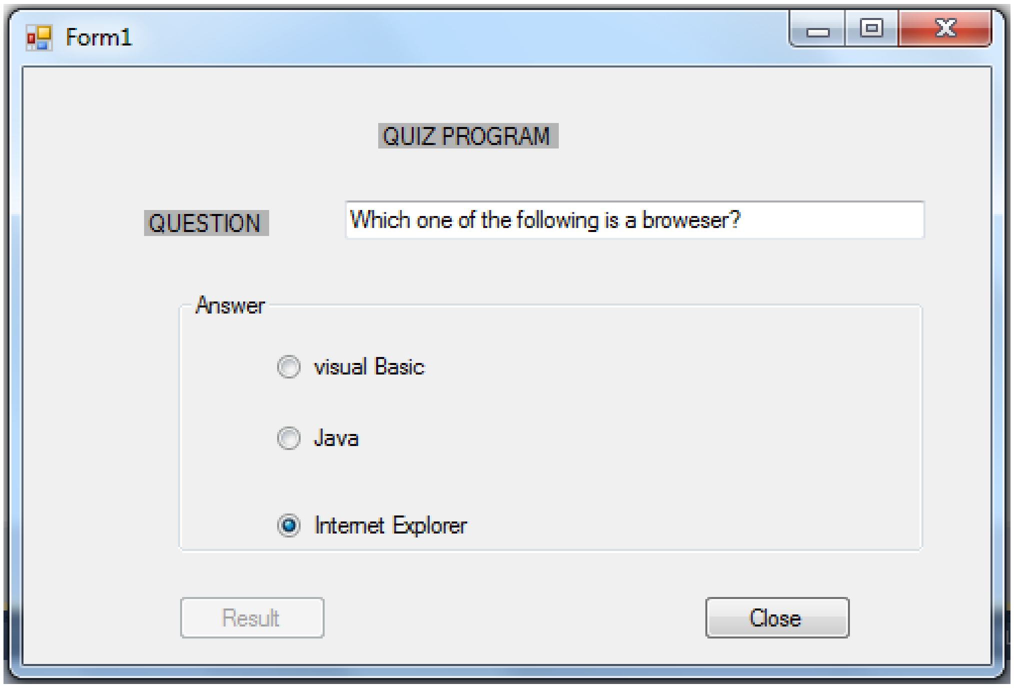
Control	Properties	
	Name	**Text**
DateTimePicker		
Label	label1	QUIZ PROGRAM
Label	label2	Question
TextBox	textBox1	
Group Box	groupBox1	Answer
Radio Button	radioButton1	Option1

Radio Button	radioButton2	Option2
Radio Button	radioButton3	Option3
Button	button1	Result
Button	button2	Close
Timer	timer1	interval set as 5000

CODING:

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

namespace WindowsFormsApplication6
{
    public partial class Form1 : Form
    {
        string[,] ans = new string[5,2];
        string[,] quest = new string[5, 4];
        int i = -1;
        public Form1()
        {
          InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            quest[0, 0] = "In which year india won the cricket world cup?";
            quest[0, 1] = "1973";
            quest[0, 2] = "1983";
            quest[0, 3] = "1986";
            quest[1, 0] = "Scanner is a________________device";
            quest[1, 1] = "Input";
            quest[1, 2] = "Output";
            quest[1, 3] = "i/o";
            quest[2, 0] = "Which one of the following is a broweser?";
            quest[2, 1] = "visual Basic";
            quest[2, 2] = "Java";
            quest[2, 3] = "Internet Explorer";
            quest[3, 0] = "___________is the desciples of Jesus came to india?";
            quest[3, 1] = "Thomas";
            quest[3, 2] = "Andrew";
            quest[3, 3] = "Philip";
            quest[4, 0] = "Sunami hit india on_________________________";
            quest[4, 1] = "24-12-2006";
            quest[4, 2] = "26-12-2004";
            quest[4, 3] = "26-12-2006";
            ans[0, 0] = "1983";
            ans[1, 0] = "Input";
            ans[2, 0] = "Internet Explorer";
            ans[3, 0] = "Thomas";
            ans[4, 0] = "26-12-2004";
            timer1.Enabled = true;
            button1.Enabled = false;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            i = i + 1;
            if ((i == 5))
            {
                timer1.Enabled = false;
                button1.Enabled = true;
            }
            else
            {
                radioButton1.Checked = false;
                radioButton2.Checked = false;
                radioButton3.Checked = false;
                textBox1.Text = quest[i, 0];
                radioButton1.Text = quest[i, 1];
                radioButton2.Text = quest[i, 2];
                radioButton3.Text = quest[i, 3];
            }

        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
            ans[i, 1] = radioButton1.Text;
        }

        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            ans[i, 1] = radioButton2.Text;
        }

        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {
            ans[i, 1] = radioButton3.Text;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int j = 0;
            int count = 0;
            for (j = 0; j <= 4; j++)
            {
                if ((ans[j, 0] == ans[j, 1]))
                {
                  count += 1;
                }
              }
              MessageBox.Show("You have Scored " + count + " marks");
            }

            private void button2_Click(object sender, EventArgs e)
            {
              System.Environment.Exit(0);
            }
          }
        }
```

OUTPUT:

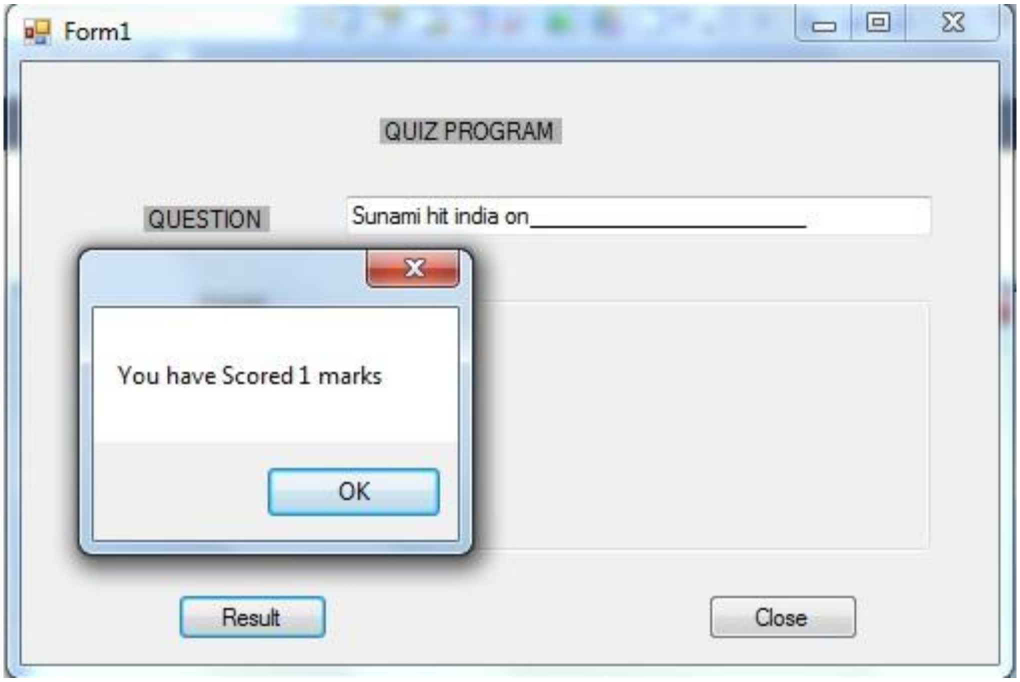


Which one of the following is a broweser?

○ visual Basic

○ Java

● Internet Explorer



Sunami hit india on______________________

You have Scored 1 marks

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:8 OPERATIONS ON FILE and DIRECTORY LIST BOX

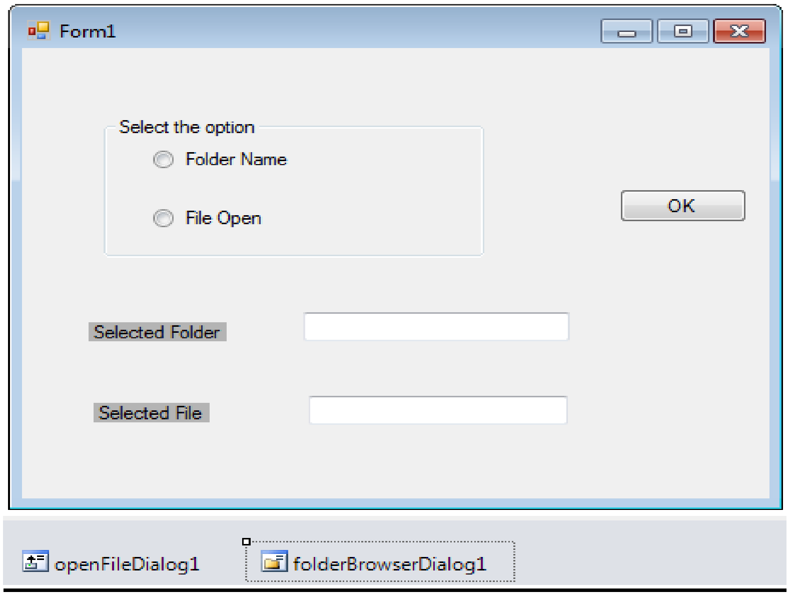
AIM

To develop a C#.NET application using the File and directory controls to implement a common dialog box.

PROCEDURE

1. Open Microsoft Visual studio create a new windows application.
2. Design a Form.
3. Edit the code in the Click event of the Button1.
4. Run the application.

FORM DESIGN

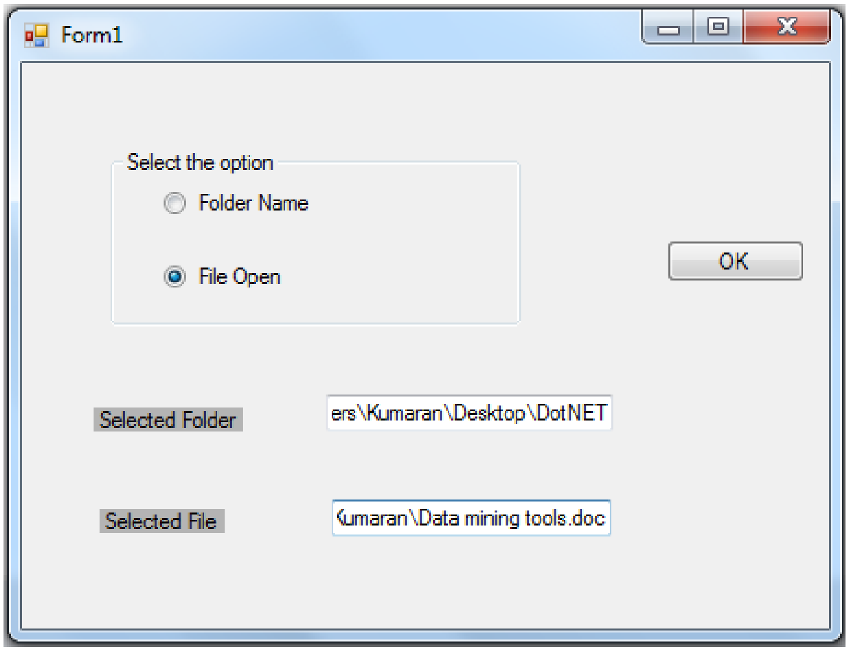


Control	Properties	
	Name	**Text**
Label	label1	Selected Folder
Label	label2	Selected File
TextBox	textBox1	
TextBox	textBox2	
Group Box	groupBox1	Select the option
Radio Button	radioButton1	Folder Name
Radio Button	radioButton2	Open File
Button	button1	OK
OpenFileDialog	openFileDialog1	
FolderBrowserDialog	folderBrowserDialog1	

CODING:

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

namespace WindowsFormsApplication7
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void button1_Click(object sender, EventArgs e)
      {
         if (radioButton1.Checked)
         {
            folderBrowserDialog1.ShowDialog();
            textBox1.Text = folderBrowserDialog1.SelectedPath;
         }
         else
         {
            openFileDialog1.ShowDialog();
            textBox2.Text = openFileDialog1.FileName;
         }
      }
   }
}
```

OUTPUT:



Select the option

○ Folder Name

● File Open

Selected Folder: ers\Kumaran\Desktop\DotNET

Selected File: Kumaran\Data mining tools.doc

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:9 STORE THE DETAILS OF STUDENTS USING ADO.NET

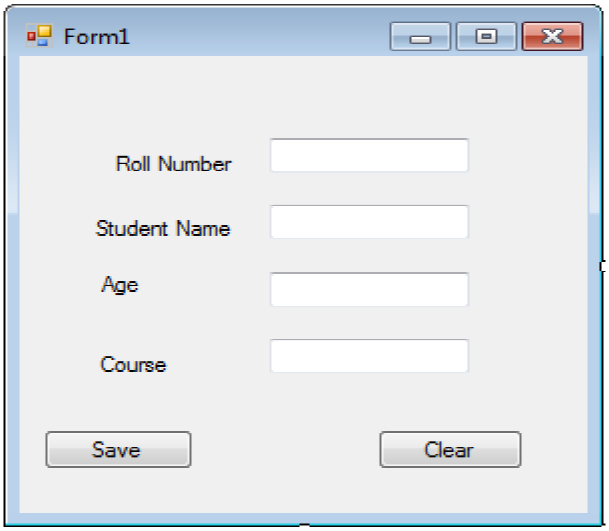
AIM

To develop a C#.NET application for storing the details of students using ADO.NET.

PROCEDURE

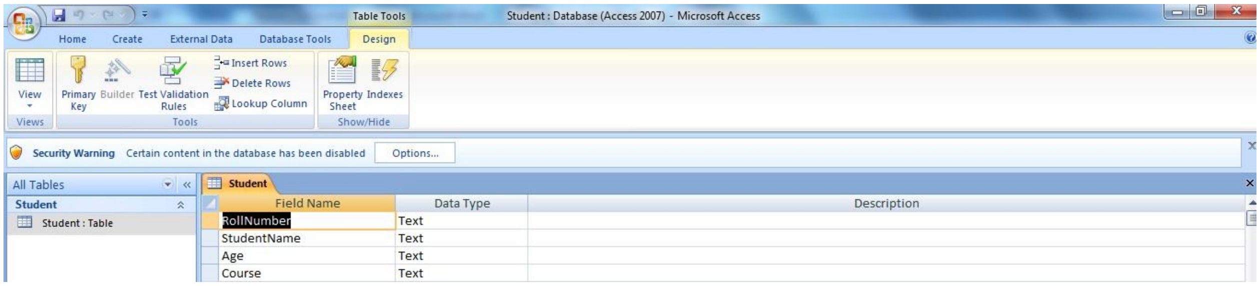
1. Open Microsoft Visual studio create a new windows application.
2. Create MS-Access database and design a Form.
3. Connect the database into the project.
4. Edit the code in the Click event of the Button1 and Button2.
5. Run the application.

FORM DESIGN



Control	Properties	
	Name	**Text**
Label	label1	Roll Number
Label	label2	Student Name
Label	label3	Age
Label	label4	Course
TextBox	textBox1	
TextBox	textBox2	
TextBox	textBox3	
TextBox	textBox4	
Button	button1	Save
Button	button2	Clear

DATABASE CREATION & CONNECTION

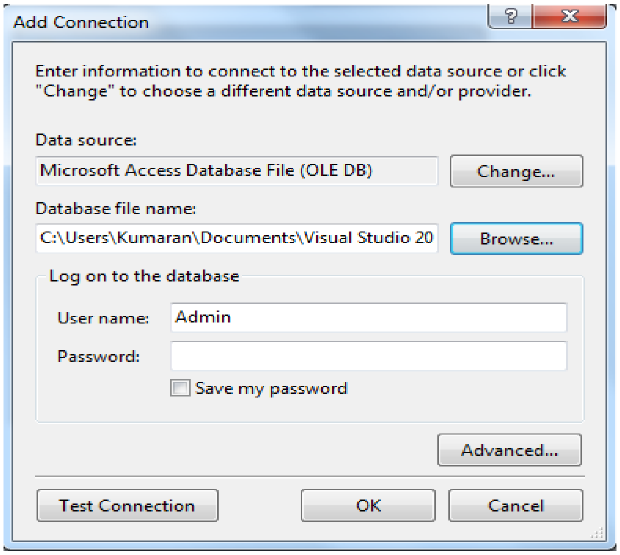


Security Warning Certain content in the database has been disabled

Student

Field Name	Data Type	Description
RollNumber	Text	
StudentName	Text	
Age	Text	
Course	Text	

Now get a connection string, Go to Tools menu and select connect to the database and browse database from the Project Directory.



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

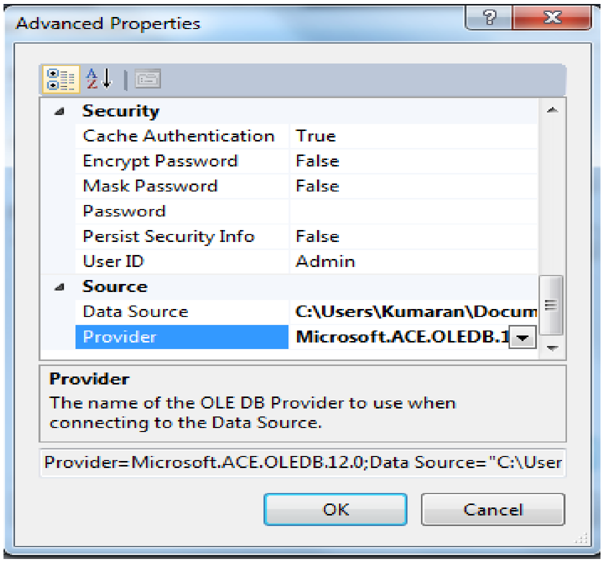
Data source:
Microsoft Access Database File (OLE DB)

Database file name:
C:\Users\Kumaran\Documents\Visual Studio 20

User name: Admin

Password:

☐ Save my password



Security	
Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin
Source	
Data Source	**C:\Users\Kumaran\Docum**
Provider	**Microsoft.ACE.OLEDB.1**

Provider
The name of the OLE DB Provider to use when connecting to the Data Source.

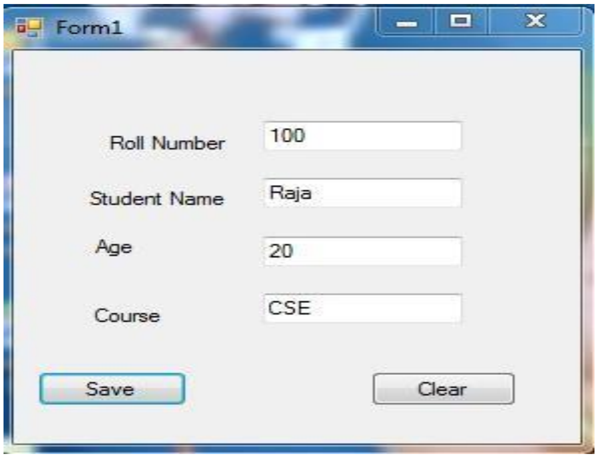
Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

CODING:

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

namespace WindowsFormsApplication8
{
   public partial class Form1 : Form
   {
      public Form1()
```

```
        {
          InitializeComponent();
        }
         OleDbConnection con = new
         OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
         Source=Student.accdb");
        private void button1_Click(object sender, EventArgs e)
        {
          OleDbCommand cmd = con.CreateCommand();
          con.Open();
          cmd.CommandText = "insert into Student values('" + textBox1.Text + "','" +
          textBox2.Text +"','" + textBox3.Text + "','" + textBox4.Text + "')";
          cmd.Connection = con;
          cmd.ExecuteNonQuery();
          MessageBox.Show("Record Saved", "Congrats");
          con.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
          textBox1.Clear();
          textBox2.Clear();
          textBox3.Clear();
          textBox4.Clear();
          textBox1.Focus();
        }
    }
}
```

OUTPUT

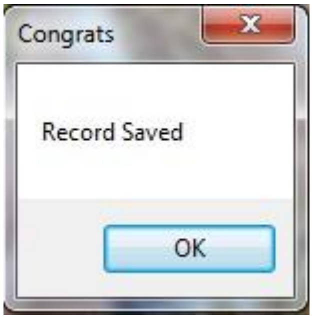


Roll Number 100

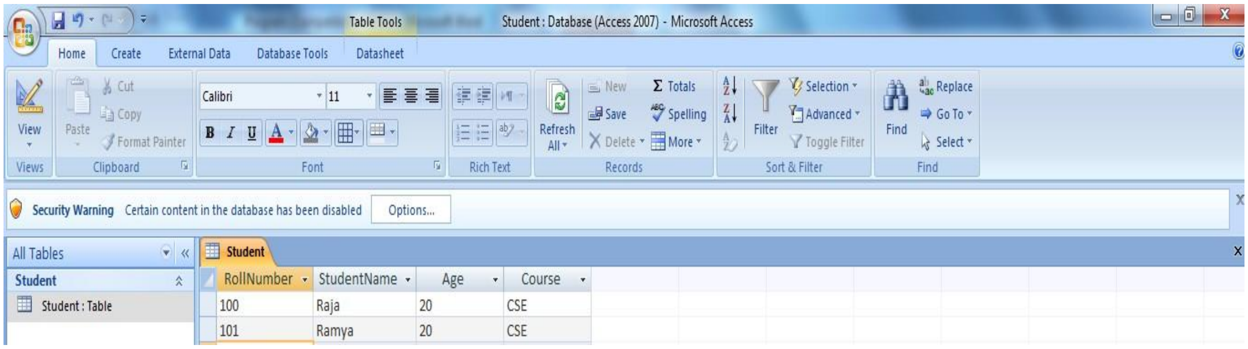
Student Name Raja

Age 20

Course CSE



Record Saved



Student : Database (Access 2007) - Microsoft Access

Security Warning Certain content in the database has been disabled

Student

RollNumber	StudentName	Age	Course
100	Raja	20	CSE
101	Ramya	20	CSE

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:10 INSERT, UPDATE & DELETE THE DETAILS OF STUDENTS USING ADO.NET

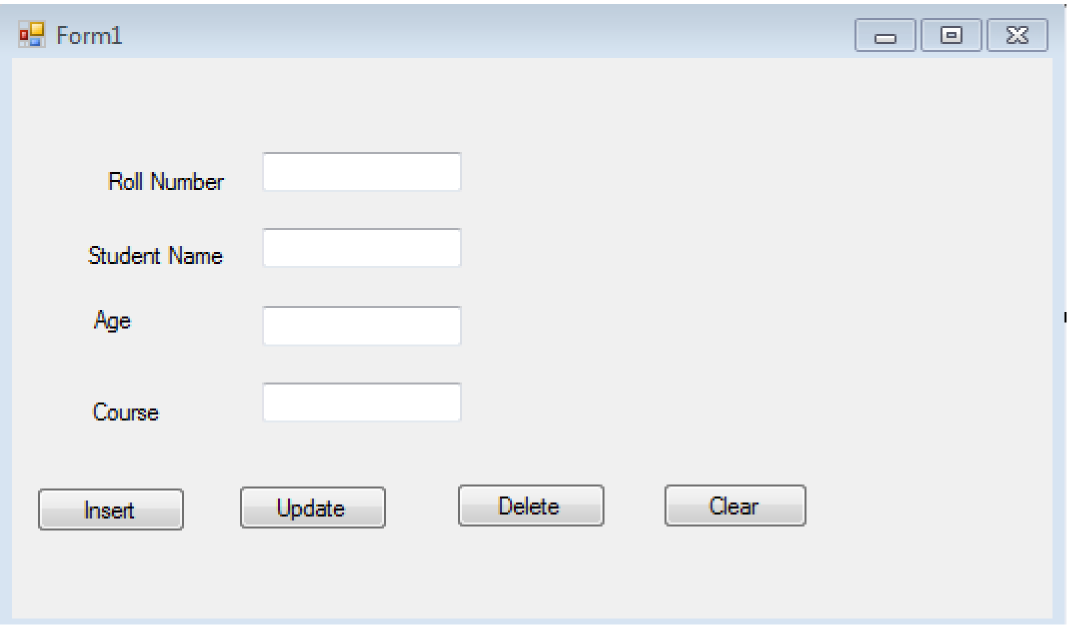
AIM

To develop a C#.NET application for insert, update & delete the details of students using ADO.NET.

PROCEDURE

1. Open Microsoft Visual studio create a new windows application.
2. Create MS-Access database and design a Form.
3. Connect the database into the project.
4. Edit the code in the Click event of the Button1, Button2, Button3 and Button4.
5. Run the application.

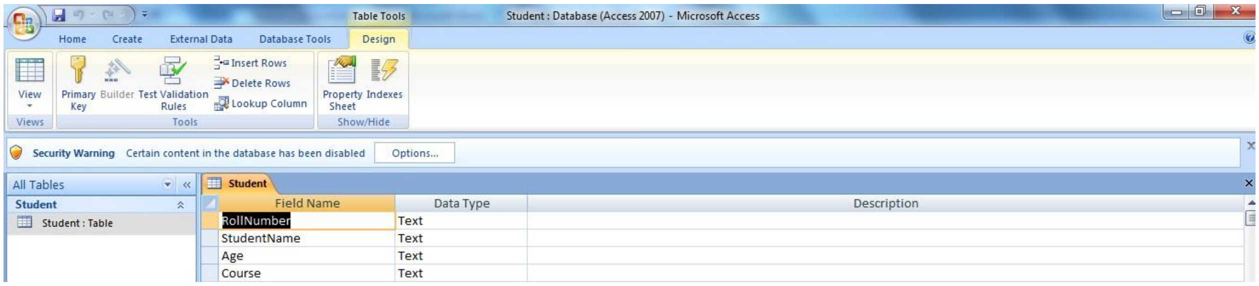
FORM DESIGN



Control	Properties	
	Name	Text
Label	label1	Roll Number
Label	label2	Student Name
Label	label3	Age
Label	label4	Course
TextBox	textBox1	
TextBox	textBox2	
TextBox	textBox3	
TextBox	textBox4	

Button	button1	Insert
Button	button2	Clear
Button	button3	Update
Button	button4	Delete

DATABASE CREATION & CONNECTION

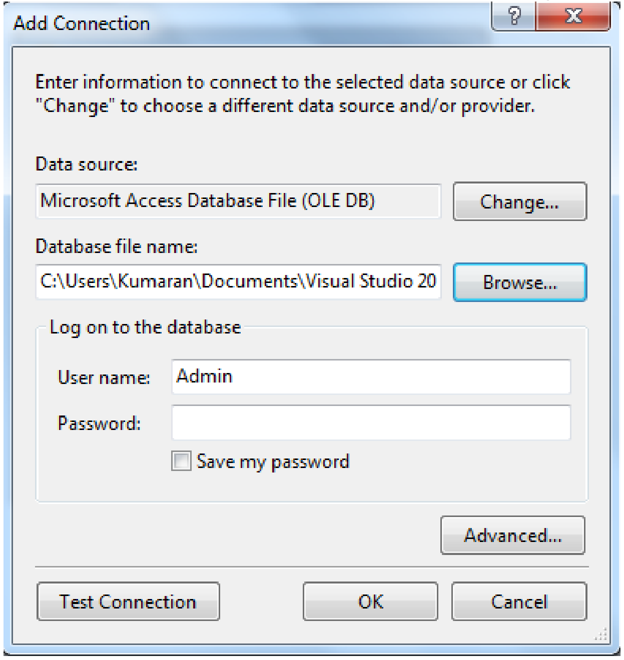


Security Warning Certain content in the database has been disabled

Student

Field Name	Data Type	Description
RollNumber	Text	
StudentName	Text	
Age	Text	
Course	Text	

Now get a connection string, Go to Tools menu and select connect to the database and browse database from the Project Directory.



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft Access Database File (OLE DB)

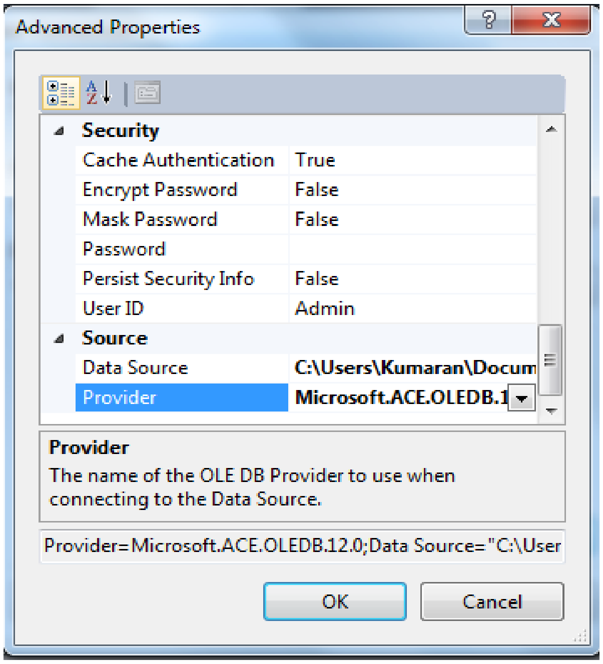
Database file name:
C:\Users\Kumaran\Documents\Visual Studio 20

Log on to the database

User name: Admin

Password:

□ Save my password



Security	
Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin
Source	
Data Source	**C:\Users\Kumaran\Docum**
Provider	**Microsoft.ACE.OLEDB.1**

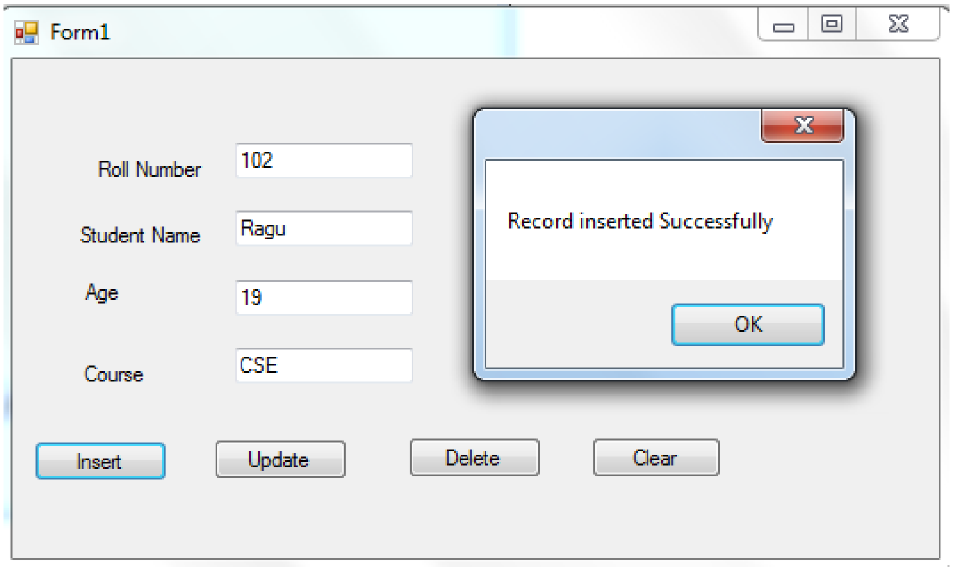
Provider
The name of the OLE DB Provider to use when connecting to the Data Source.

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

CODING:

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

namespace WindowsFormsApplication8
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
            OleDbConnection con = new
            OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=Student.accdb");
            private void button1_Click(object sender, EventArgs e)
        {
            con.Open();
            OleDbCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "insert into Student values('" + textBox1.Text + "','" +textBox2.Text
          + "','" + textBox3.Text + "','" + textBox4.Text + "')";
            cmd.Connection = con;
            cmd.ExecuteNonQuery();
            MessageBox.Show("Record inserted Successfully");
            con.Close();
        }

            private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Clear();
            textBox2.Clear();
            textBox3.Clear();
            textBox4.Clear();
         }

            private void button4_Click(object sender, EventArgs e)
        {
            con.Open();
            OleDbCommand cmd = con.CreateCommand();cmd.CommandType =
            CommandType.Text;cmd.CommandText = "delete from Student where StudentName='" +
            textBox2.Text +"'";
            cmd.ExecuteNonQuery();con.Close();
            MessageBox.Show("Record deleted Successfully");
        }
            private void button3_Click(object sender, EventArgs e)
        {
            con.Open();
            OleDbCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "update Student set StudentName='" + textBox1.Text + "' where
            StudentName='" + textBox2.Text + "'";
            cmd.ExecuteNonQuery();
            con.Close();
            MessageBox.Show("Record updated Successfully");
        }
    }
}
```

OUTPUT



Form1

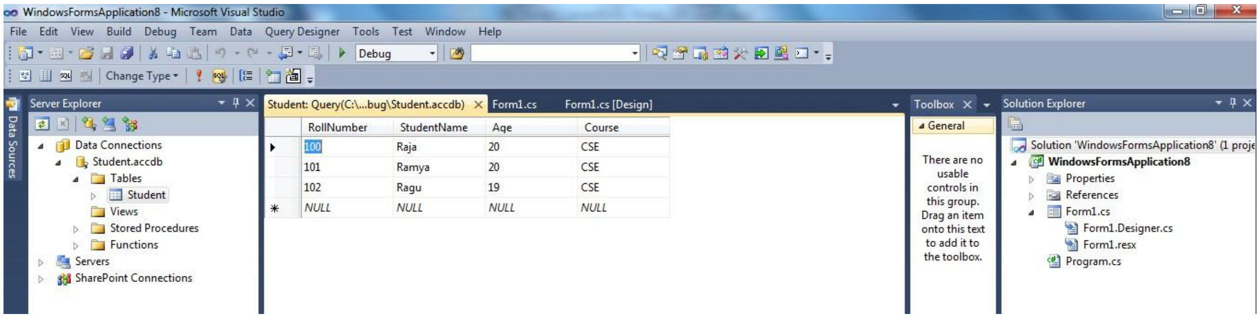
Roll Number: 102

Student Name: Ragu

Age: 19

Course: CSE

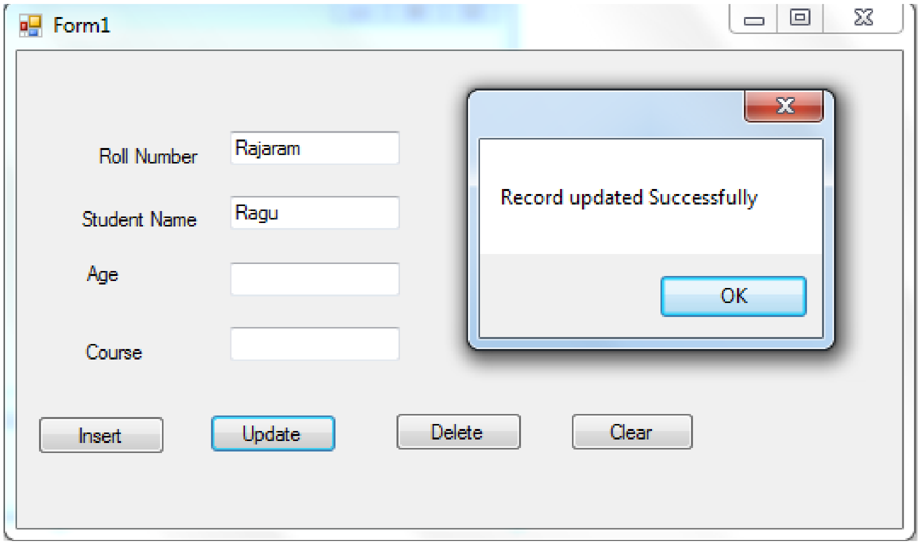
Record inserted Successfully



WindowsFormsApplication8 - Microsoft Visual Studio

RollNumber	StudentName	Age	Course
100	Raja	20	CSE
101	Ramya	20	CSE
102	Ragu	19	CSE
NULL	NULL	NULL	NULL

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.



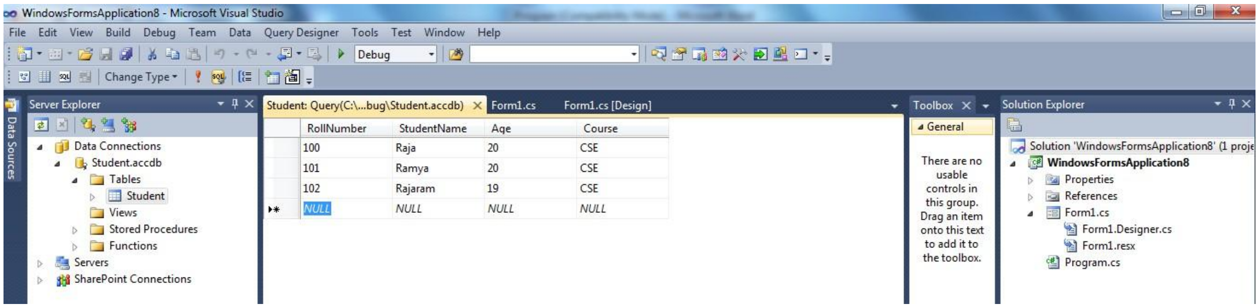
Roll Number: Rajaram

Student Name: Ragu

Age:

Course:

Record updated Successfully



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

RollNumber	StudentName	Age	Course
100	Raja	20	CSE
101	Ramya	20	CSE
102	Rajaram	19	CSE
NULL	NULL	NULL	NULL

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:11 **USING DATAGRID TO DISPLAY RECORDS**

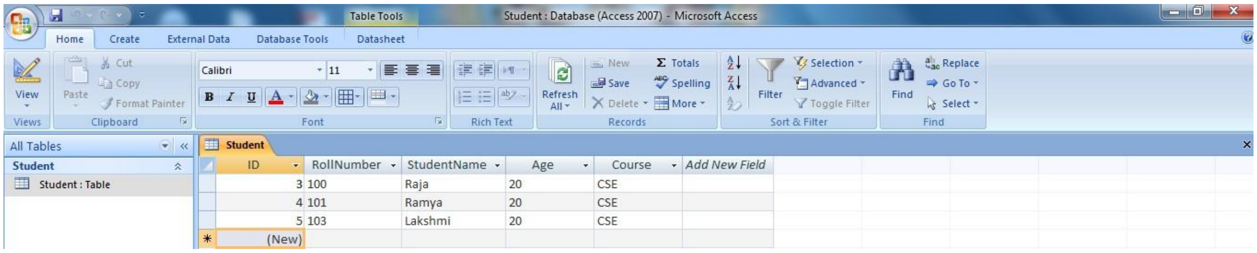
AIM

To develop a C#.NET application using datagride to display records.

PROCEDURE:

To Create MS Access file

- Go to Start → All program → Microsoft office → Microsoft Access.
- Click the office button → New file
- After creating New file, click on the Browse icon in the screen.
- Create a new table and choose Microsoft Access Database (2000 format).
- Next, click OK button for the given tab.
- Click the Create option.
- Now, the new table will be created in Database. Fill data in the new table.
- After filling the Data in Database, just Save that.
- One dialog box will be open. Enter the table name for the given Data and press OK.



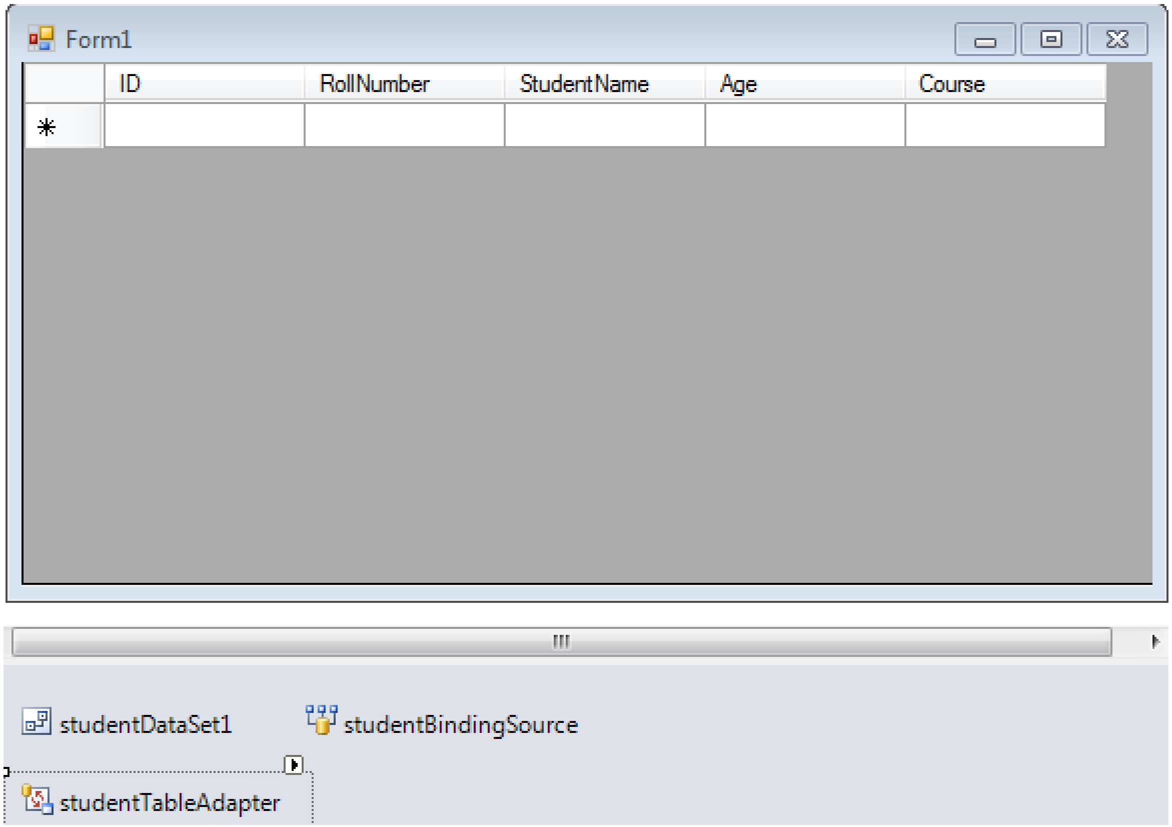
ID	RollNumber	StudentName	Age	Course	Add New Field
3	100	Raja	20	CSE	
4	101	Ramya	20	CSE	
5	103	Lakshmi	20	CSE	
(New)					

To create the Windows form Application to Display the content

- Start → visual studio 2010 (any version) → New project → Windows form Application.
- Give the file name in your particular database name.
- Press ok to create a new form.
- The new form will open.
- To draw the data grid view control on the form.
- In that control will be toolbox.
- To click on the top of data grid view icon.
- And click choose data source
- To click Add project data source option.
- The data source configuration wizard dialog box will be open to click the next button.
- The database icon will be selected and click the next button.
- Select a new connection button to select a database.
- Click the change option
- Choose Microsoft Access Database File (OLE DB).
- Click ok button.

- After selecting the browse option select the saved table (Access file).
- To select the already saved file.
- Click open button.
- The data source configuration wizard dialog box will open and click the next button.
- One dialog box will open and click the yes button.
- The data source configuration wizard dialog box will open and click the next button.
- To select the tables option in this given window.
- To open the form design page.

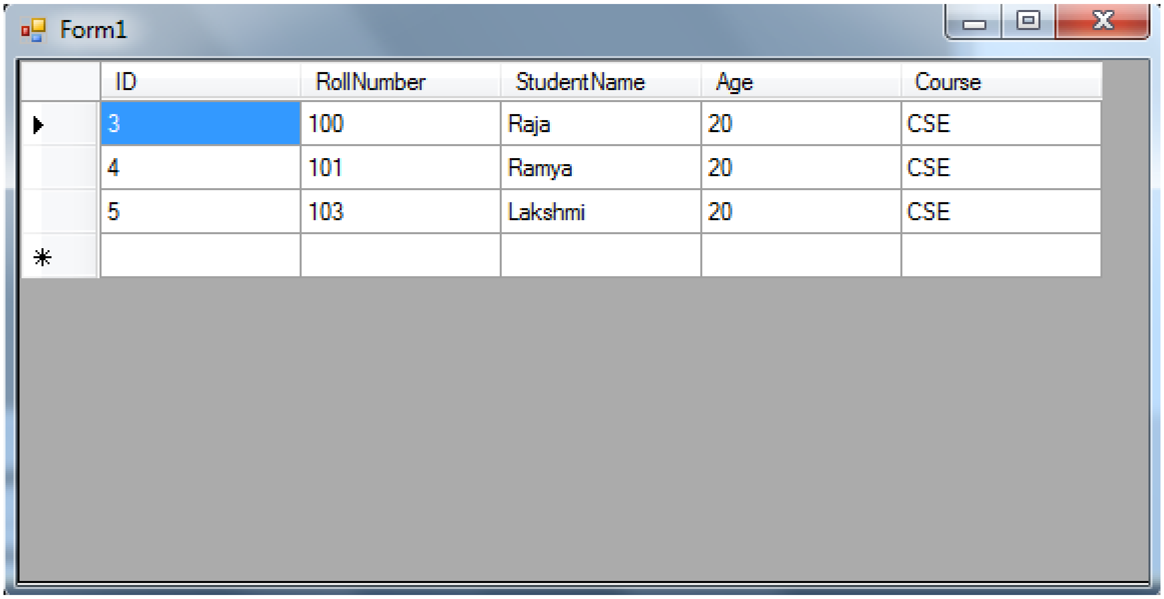
FORM DESIGN



	ID	RollNumber	StudentName	Age	Course
*					

- Run the given program to press F5 key (or) click run option.
- The output will be displayed in the Data Grid view box.

OUTPUT



	ID	RollNumber	StudentName	Age	Course
▶	3	100	Raja	20	CSE
	4	101	Ramya	20	CSE
	5	103	Lakshmi	20	CSE
*					

RESULT:

Thus the above application has been executed and verified successfully.

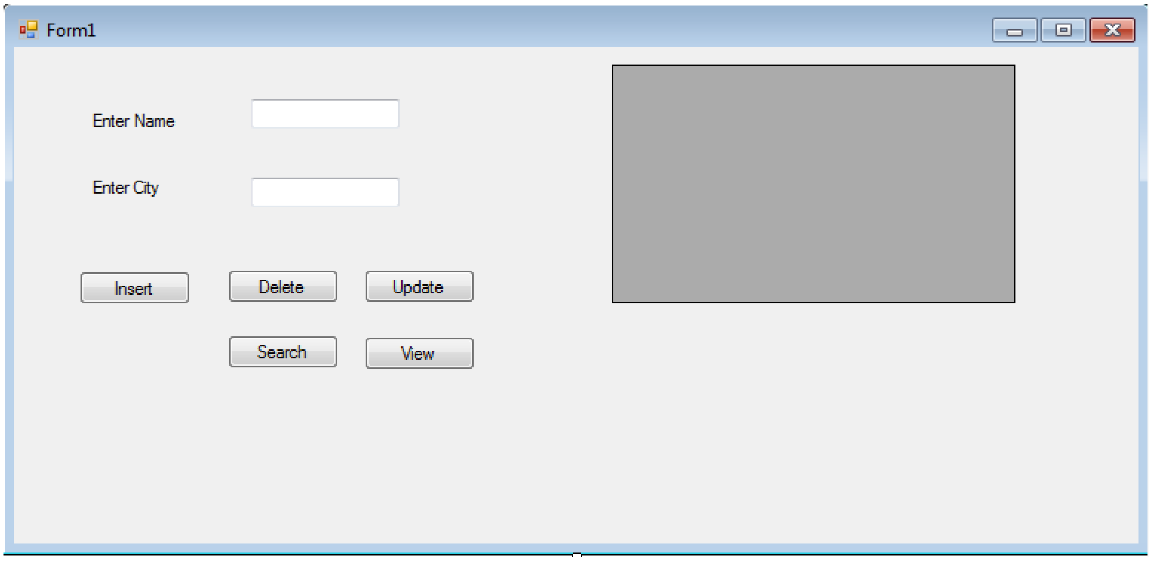
EX.NO:12 **USING DATAGRID TO INSERT, UPDATE & DELETE RECORDS**

DATE:

AIM

To develop a C#.NET application using datagride to insert, update & delete records.

FORM DESIGN



Control	Properties	
	Name	**Text**
Label	label1	Enter Name
Label	label2	Enter City
TextBox	textBox1	
TextBox	textBox2	
Button	button1	Insert
Button	button2	Delete
Button	button3	Update
Button	button4	View
Button	button5	Search
DataGridView	dataGridView1	

DATABASE CREATION & CONNECTION

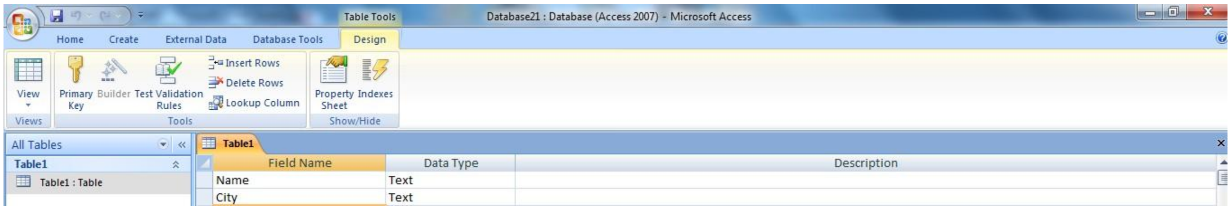
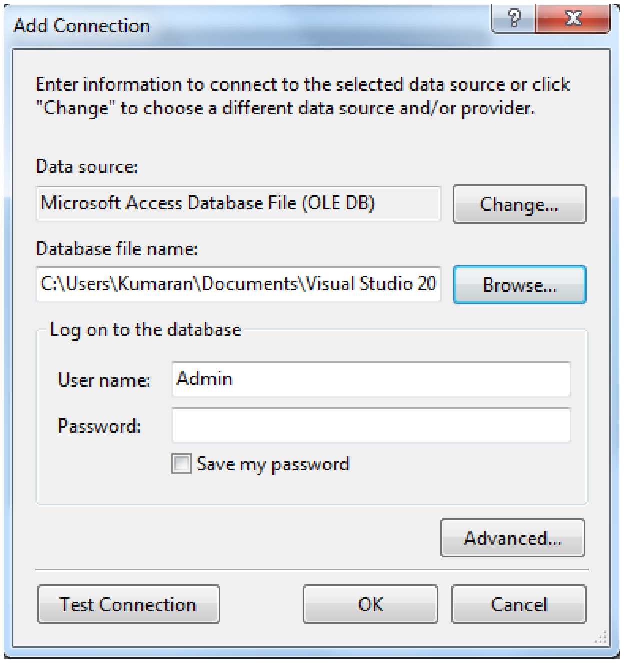


Table1

Field Name	Data Type	Description
Name	Text	
City	Text	

Now get a connection string, Go to Tools menu and select connect to the database and browse database from the Project Directory.



Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

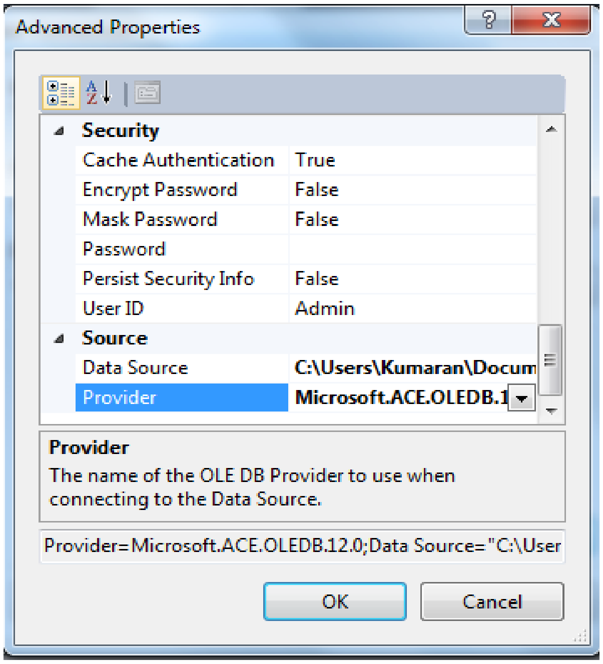
Database file name: C:\Users\Kumaran\Documents\Visual Studio 20

Log on to the database

User name: Admin

Password:

□ Save my password



Advanced Properties

Security

Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin

Source

Data Source	C:\Users\Kumaran\Docum
Provider	Microsoft.ACE.OLEDB.1

Provider
The name of the OLE DB Provider to use when connecting to the Data Source.

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

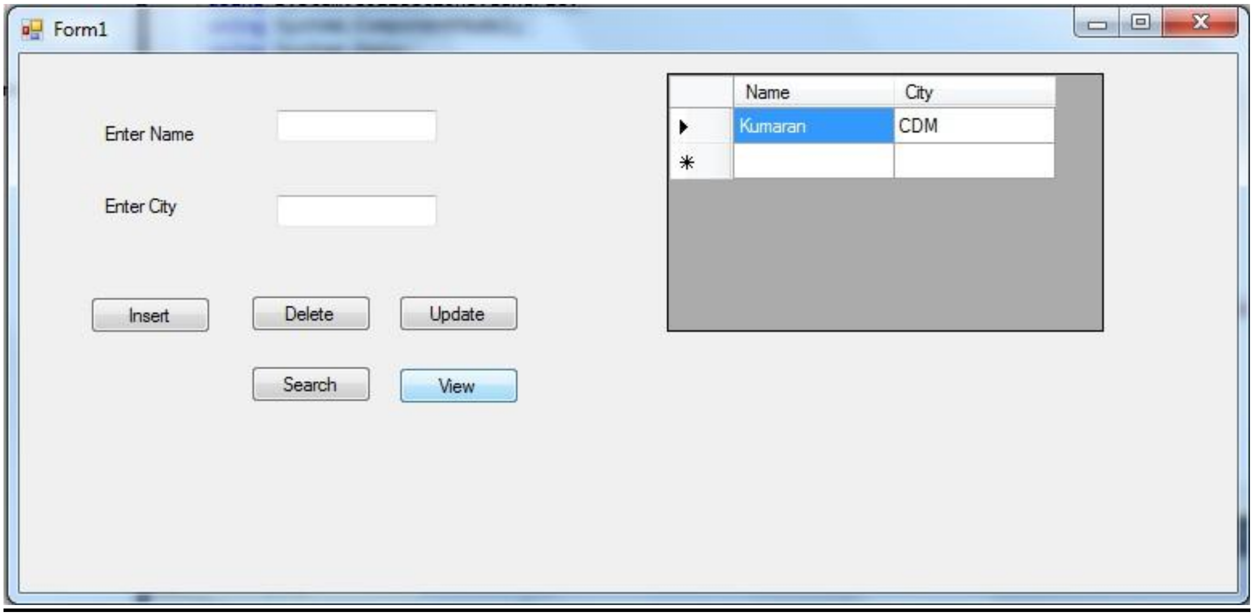
CODING:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
namespace WindowsFormsApplication10
{
    public partial class Form1 : Form
    {
        OleDbConnection con = new OleDbConnection ("Provider=Microsoft.ACE.OLEDB.12.0;
Data Source=C:\\Users\\Kumaran\\Desktop\\Database21.accdb");
        int count = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
```

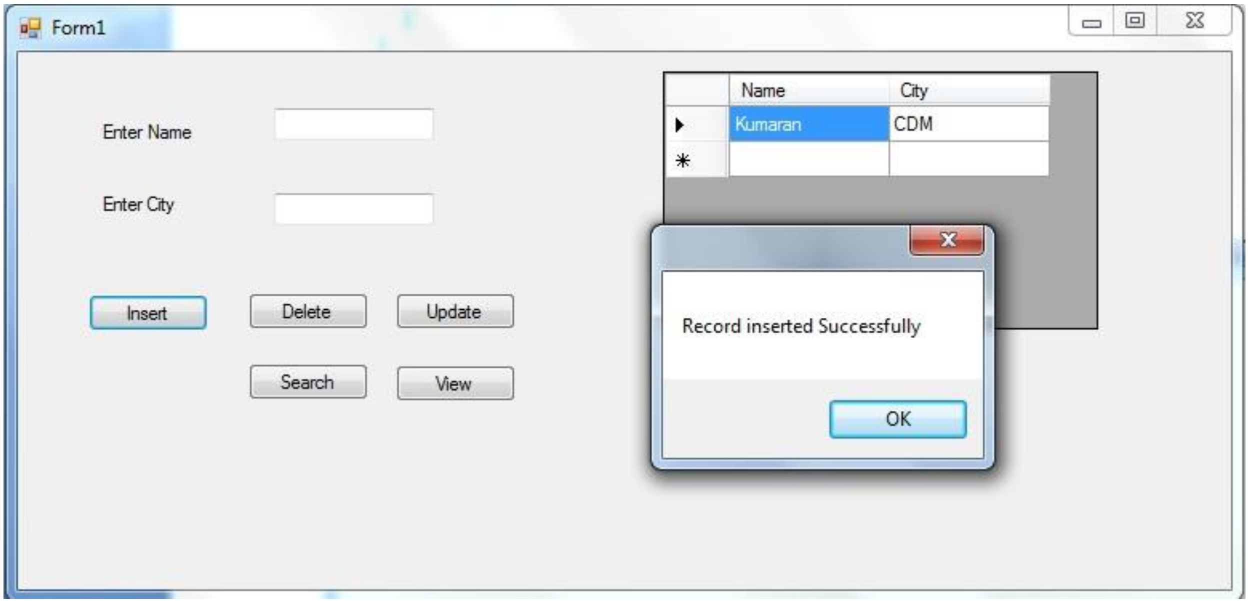
```
  con.Open();
  OleDbCommand cmd = con.CreateCommand();
  cmd.CommandType = CommandType.Text;
  cmd.CommandText = "insert into table1 values ('" + textBox1.Text + "','" +
  textBox2.Text + "')";
  cmd.ExecuteNonQuery();
  con.Close();
  textBox1.Text = "";
  textBox2.Text = "";
  MessageBox.Show("Record inserted Successfully");
}

  private void button4_Click(object sender, EventArgs e)
{
  con.Open();
  OleDbCommand cmd = con.CreateCommand();
  cmd.CommandType = CommandType.Text;
  cmd.CommandText = "select * from table1";
  cmd.ExecuteNonQuery();
  DataTable dt = new DataTable();
  OleDbDataAdapter da = new OleDbDataAdapter(cmd);
  da.Fill(dt);
  dataGridView1.DataSource = dt;
  con.Close();
}

private void button2_Click(object sender, EventArgs e)
{
  con.Open();
  OleDbCommand cmd = con.CreateCommand();
  cmd.CommandType = CommandType.Text;
  cmd.CommandText = "delete from table1 where name='" + textBox1.Text + "'";
  cmd.ExecuteNonQuery();
  con.Close();
  MessageBox.Show("Record deleted Successfully");
}

private void button3_Click(object sender, EventArgs e)
{
  con.Open();
  OleDbCommand cmd = con.CreateCommand();
  cmd.CommandType = CommandType.Text;
  cmd.CommandText = "update table1 set name='"+textBox2.Text+"' where
  name='"+textBox1.Text+"'";
  cmd.ExecuteNonQuery();
  con.Close();
  MessageBox.Show("Record updated Successfully");
```

```
        }

        private void button5_Click(object sender, EventArgs e)
        {
            count = 0;
            con.Open();
            OleDbCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "select * from table1 where name='"+textBox1.Text+"'";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            da.Fill(dt);
            count = Convert.ToInt32(dt.Rows.Count.ToString());
            dataGridView1.DataSource = dt;
            con.Close();
            if (count == 0)
            {
                MessageBox.Show("Record not Found");
            }
        }

    }
}
```

OUTPUT



	Name	City
▶	Kumaran	CDM
*		

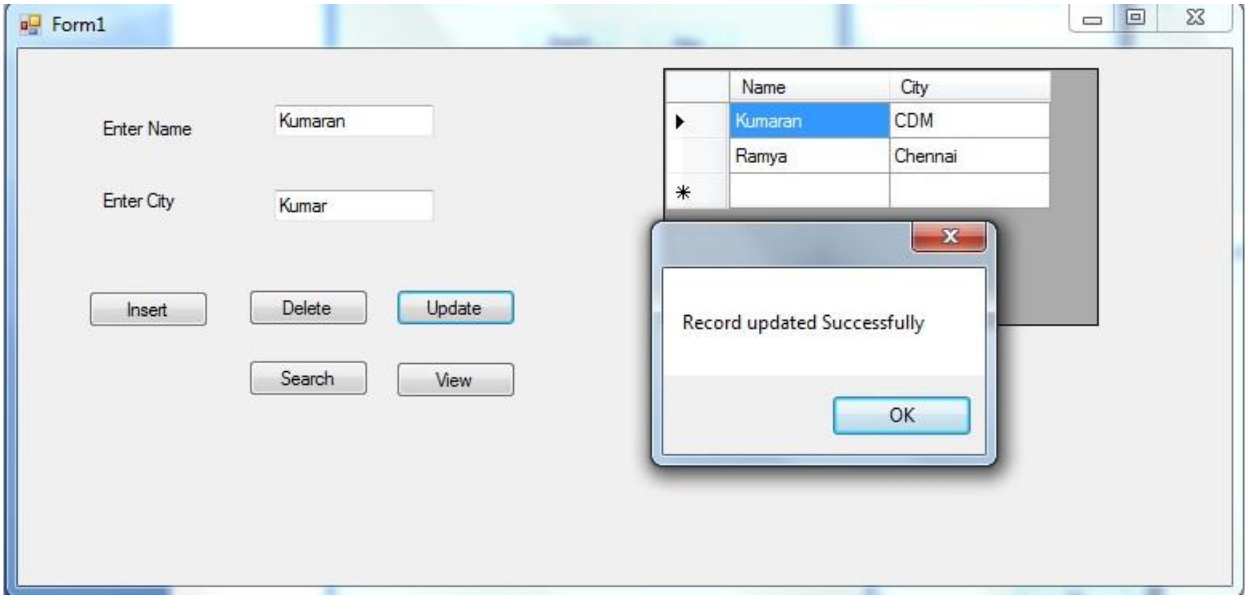


Enter Name

Enter City

Name	City
Kumaran	CDM

Record inserted Successfully



Enter Name Kumaran

Enter City Kumar

Name	City
Kumaran	CDM
Ramya	Chennai

Record updated Successfully

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:13 XML DATABASE CREATION

AIM

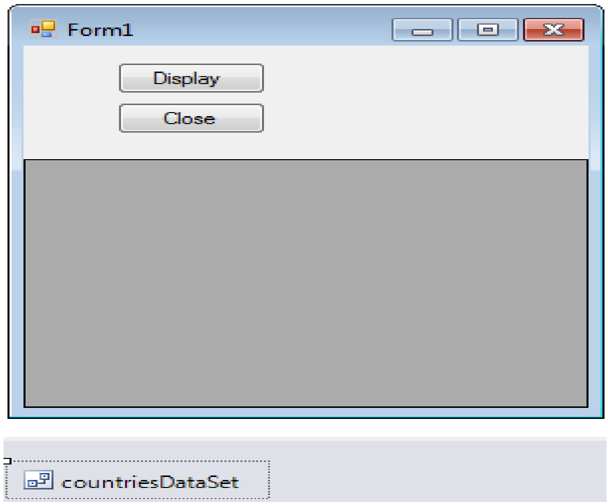
To develop a application to read the details of the selecter country stored in a Xml database and display back to the user.

PROCEDURE

1. Open Microsoft Visual Studio2010 and create a new windows application with the name ReadingXml1.
2. On the project menu click Add New item and then click Xml File option.
3. Enter the file name as countries.xml and click Add button. Now edit the following Xml code

```
    <countries>
        <country>
                <name>India</name>
                <year>1988</year>
                <sports>Hockey</sports>
        </country>
        <country>
                <name>Srilanka</name>
                <year>1996</year>
                <sports>Cricket</sports>
        </country>
        <country>
                <name>England</name>
                <year>1979</year>
                <sports>Football</sports>
        </country>
   </countries>
```

4. On File menu cilck Save countries.xml option.



5. On form1 add two Button controls and a DataGridView control from the tool box.
6. Change the Text property of Butten1 as Display County and of Butten2 as Back.
7. On Form1 add a DataSet Control from the tool box. In the AddDataSet dialogue box, select Untyped dataset option and then click Ok.Now DataSet1 is added to the component tray.
8. Set the Name and DataSetName properties for countriesDataSet.
9. Edit the following c# code in the click event of Button1

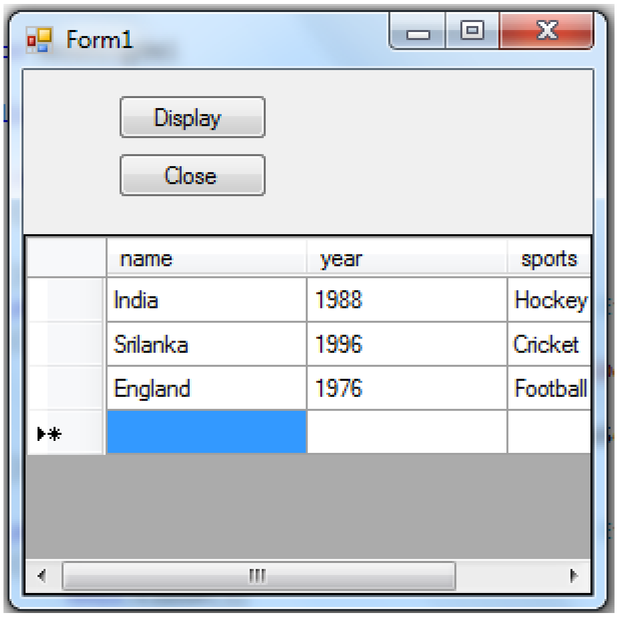
```
{
 String filePath=@"C:\Users\Documents\Projects\ReadingXml\ReadingXml\countries.xml";
 countriesDataSet.ReadXml(filePath);
 dataGridView1.DataSource=countriesDataSet.Tables[0];
 }
```

10. Edit the following code in the click event of Button2

```
        this.close( );
```

11. Press F5 to run the application.

OUTPUT



name	year	sports
India	1988	Hockey
Srilanka	1996	Cricket
England	1976	Football

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:14 READING XML DATABASE

AIM

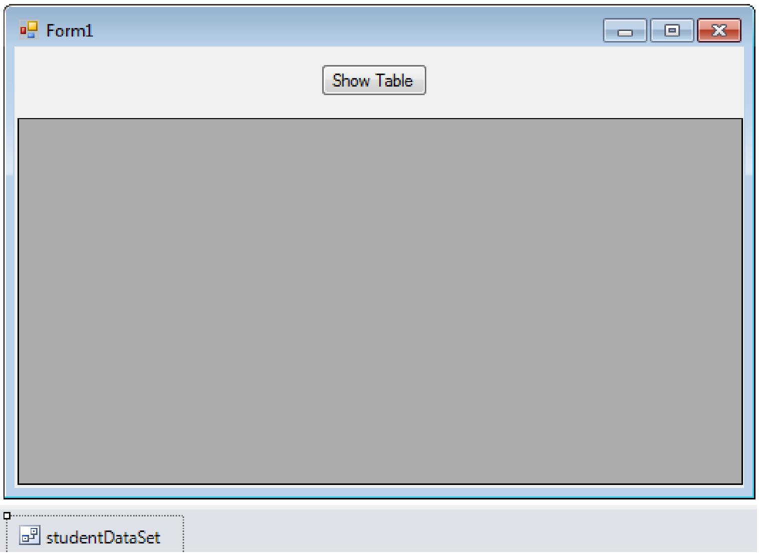
To develop a window application to read an Xml document containing subject, marks scored, year of passing into a dataset.

PROCEDURE

1. Open Microsoft Visual Studio2010 and create a new windows application with the name ReadingXml.
2. On the project menu click Add New item and then click Xml File option.
3. Enter the file name as student.xml and click Add button. Now edit the following Xml code

```
<student>
        <stud>
            <name>th</name>
            <subject>Mech</subject>
            <marks>567</marks>
            <year>2012</year>
        </stud>
        <stud>
            <name>Jeba</name>
            <subject>ECE</subject>
            <marks>480</marks>
            <year>2017</year>
        </stud>
        <stud>
            <name>Banu</name>
            <subject>CSE</subject>
            <marks>520</marks>
            <year>2014</year>
         </stud>
</student>
```

4. On File menu cilck Save student.xml option.
5. On form1 add a Button control and a DataGridView control from the tool box. Change the Text property of Button1 as Show Table.
7. On Form1 add a DataSet Control from the tool box. In the AddDataSet dialogue box,select Untyped dataset option and then click Ok. Now DataSet1 is added to the component tray.
8. Set the Name and DataSetName properties for studentDataSet.

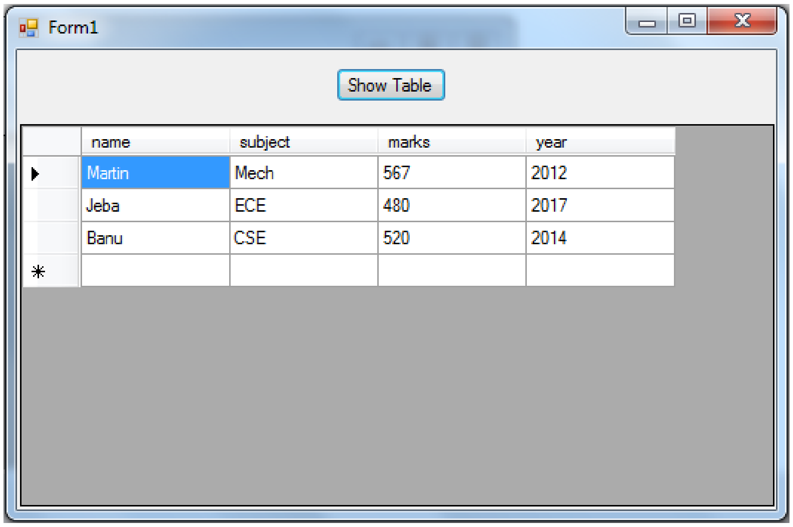


9. Edit the following C# code in the click event of Button1

```
{
 String filePath=@"C:\Users\Documents\Projects\ReadingXml1\student.xml";
 studentDataSet.ReadXml(filePath);
 dataGridView1.DataSource=studentDataSet.Tables[0];
 }
```

10. Press F5 to run application.

OUTPUT



name	subject	marks	year
Martin	Mech	567	2012
Jeba	ECE	480	2017
Banu	CSE	520	2014

RESULT:

Thus the above application has been executed and verified successfully.

EX.NO:15 ACCESSING XML DATABASE USING DATASET

AIM

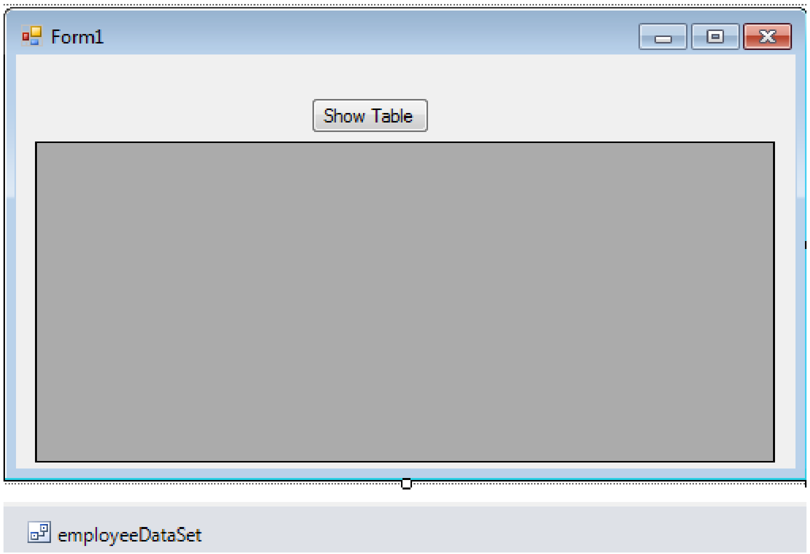
To develop a Windows application to read an Xml document containing employee name, code, Basic pay, HRA, DA into a dataSet.

PROCEDURE

1. Open Microsoft Visual Studio 2010 and create a new windows application with the name ReadingXml2.
2. On the project menu click Add New item and then click Xml File option.
3. Enter the file name as employee.xml and click Add button. Now edit the following Xml code.

```
<employee>
        <emp>
              <name>Jose</name>
              <code>MGIT0002</code>
              <Bpay>20000</Bpay>
              <HRA>30</HRA>
               <DA>28</DA>
        </emp>
        <emp>
              <name>Lin</name>
              <code>MGCSE0001</code>
              <Bpay>24000</Bpay>
              <HRA>26</HRA>
               <DA>30</DA>
        </emp>
        <emp>
              <name>Edwin</name>
              <code>MGP0001</code>
              <Bpay>35000</Bpay>
              <HRA>40</HRA>
               <DA>30</DA>
        </emp>
  </employee>
```

4. On File menu cilck Save employee.xml option.
5. On form1 add a Button control and a DataGridView control from the tool box. Change the Text property of Button1 as Show Table.
7. On Form1 add a DataSet Control from the tool box. In the AddDataSet dialogue box, select Untyped dataset option and then click OK. Now DataSet1 is added to the component tray.
8. Set the Name and DataSetName properties for employeeDataSet.

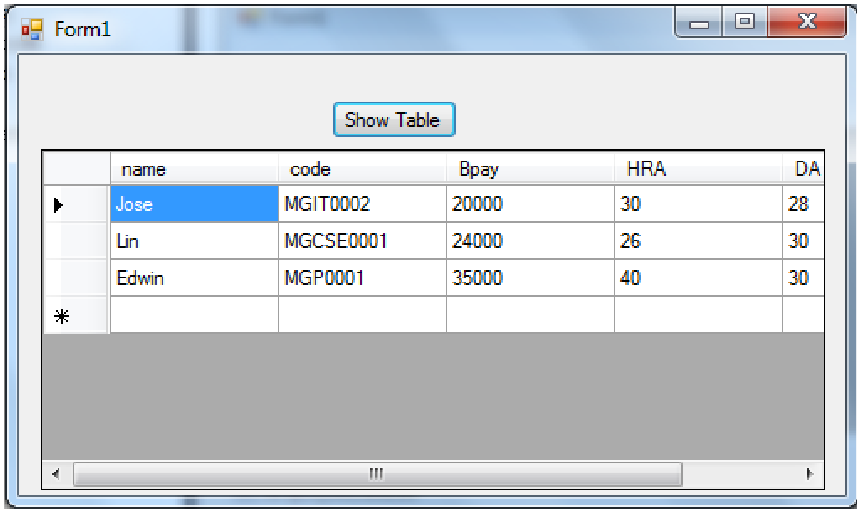


9. Edit the following C# code in the click event of Button1.

```
{
        String filePath=@"C:\Users\Documents\Projects\ReadingXml2\employee.xml";
        employeeDataSet.ReadXml(filePath);
        dataGridView1.DataSource=employeeDataSet.Tables[0];
 }
```

10. Press F5 to run the application.

OUTPUT



	name	code	Bpay	HRA	DA
▶	Jose	MGIT0002	20000	30	28
	Lin	MGCSE0001	24000	26	30
	Edwin	MGP0001	35000	40	30
*					

RESULT:

Thus the above application has been executed and verified successfully.

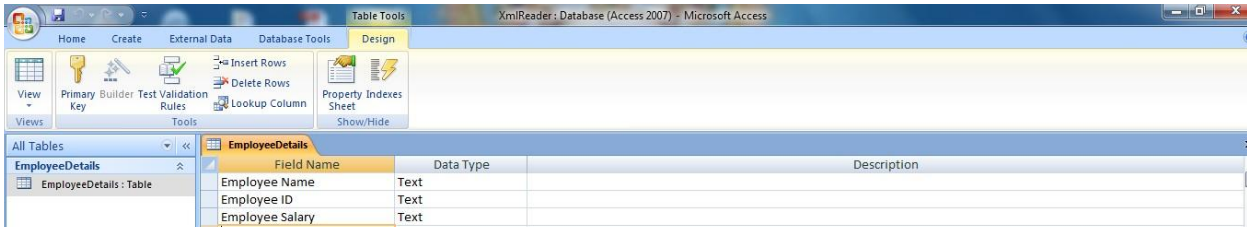
EX.NO:16 GENERATING XML DOCUMENT FROM DATABASE

AIM

To develop a windows application to read employee records from Database and generate Xml document containing employee records.

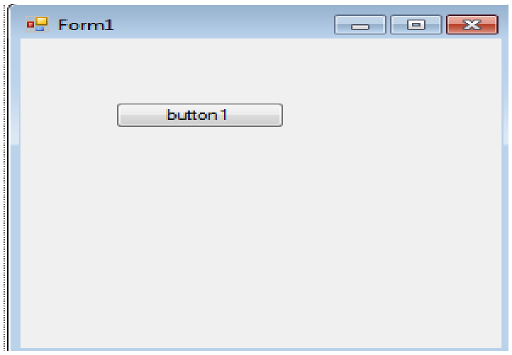
PROCEDURE

1. Create a EmployeeDetails table in Ms-Access named XmlReader.

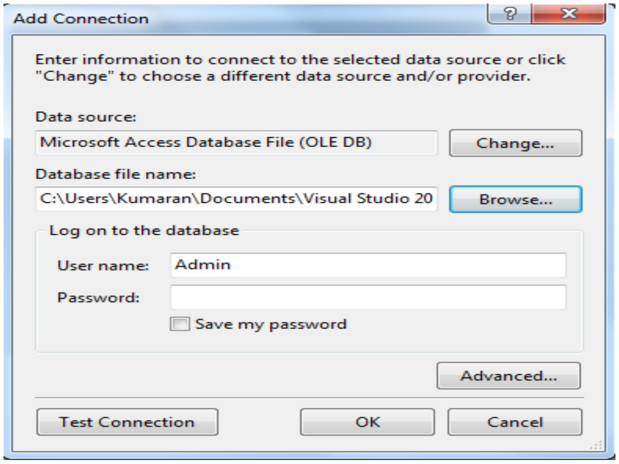


Field Name	Data Type	Description
Employee Name	Text	
Employee ID	Text	
Employee Salary	Text	

2. Open Visual studio 2010 and create a new Windows application. Design Form1 with a button control.



3. Now get a connection string, Go to Tools menu and select connect to the database and browse database from the Project Directory.



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft Access Database File (OLE DB)

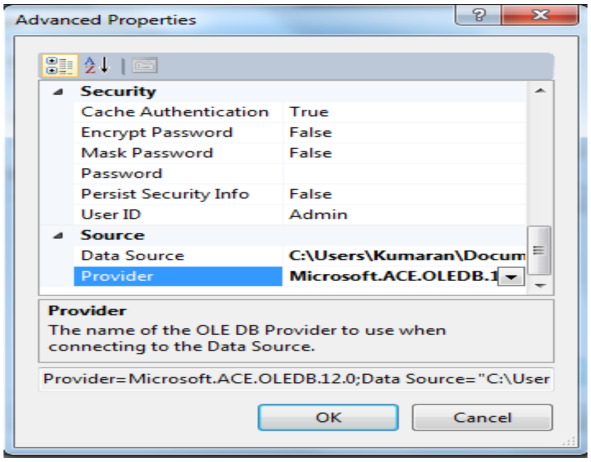
Database file name:
C:\Users\Kumaran\Documents\Visual Studio 20

Log on to the database

User name: Admin

Password:

□ Save my password



Security	
Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin
Source	
Data Source	**C:\Users\Kumaran\Docum**
Provider	**Microsoft.ACE.OLEDB.1**

Provider
The name of the OLE DB Provider to use when connecting to the Data Source.

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

3. Edit the following code at the top before Public class from1

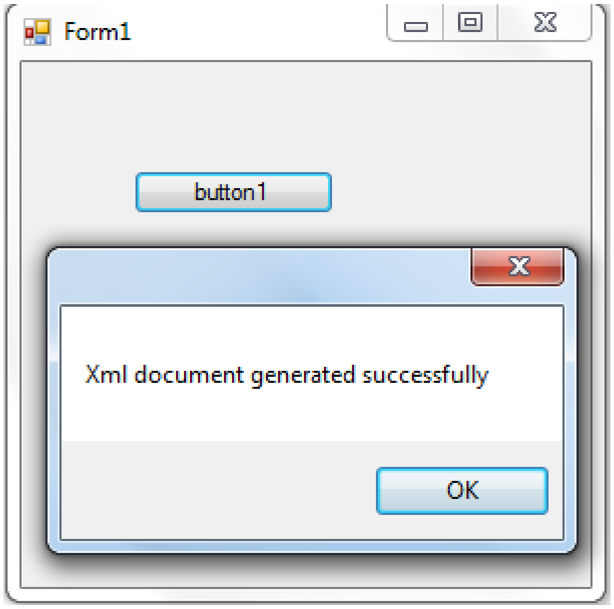
```
using System.Data.OleDb;
```

4. Edit the following code in the click event of Button1

```
 {
OleDbDataAdapter oledbAdapter ;
DataSet ds = new DataSet();
OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=XmlReader.accdb");
try
{
   con.Open();
   oledbAdapter = new OleDbDataAdapter("select * from EmployeeDetails", con);
   oledbAdapter.Fill(ds);
   ds.WriteXml("empolyee.xml");
   MessageBox.Show("Xml document generated successfully");
   con.Close();
}
catch (Exception ex)
{
   MessageBox.Show(ex.ToString());
}
}
```

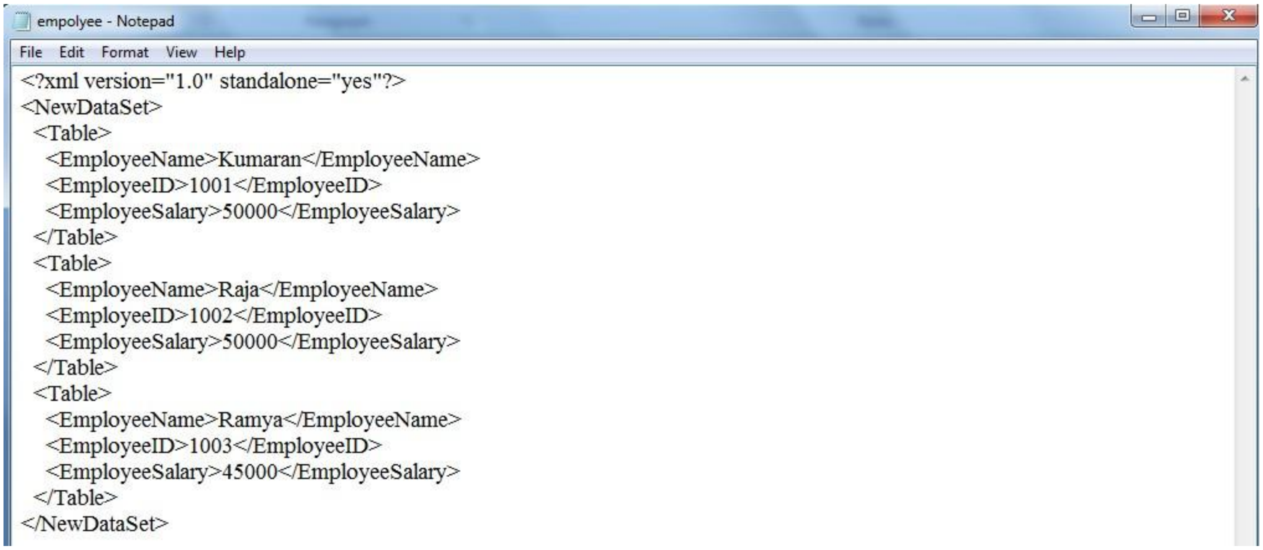
5. Run the application and find the employee.xml file in the project directory.

OUTPUT



Xml document generated successfully

employee.xml



```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Table>
    <EmployeeName>Kumaran</EmployeeName>
    <EmployeeID>1001</EmployeeID>
    <EmployeeSalary>50000</EmployeeSalary>
  </Table>
  <Table>
    <EmployeeName>Raja</EmployeeName>
    <EmployeeID>1002</EmployeeID>
    <EmployeeSalary>50000</EmployeeSalary>
  </Table>
  <Table>
    <EmployeeName>Ramya</EmployeeName>
    <EmployeeID>1003</EmployeeID>
    <EmployeeSalary>45000</EmployeeSalary>
  </Table>
</NewDataSet>
```

RESULT:

Thus the above application has been executed and verified successfully.

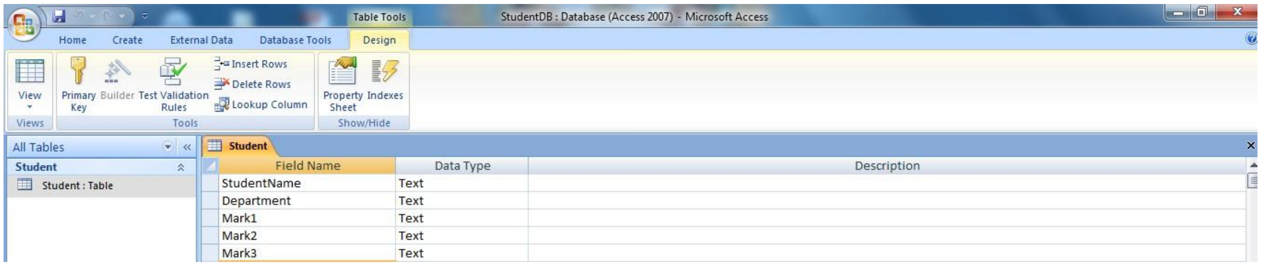
EX.NO:17 GENERATING XML DOCUMENT USING ADO.NET

AIM

To develop a windows application to read student records from Database using ADO.NET and generate Xml document containing student records.

PROCEDURE

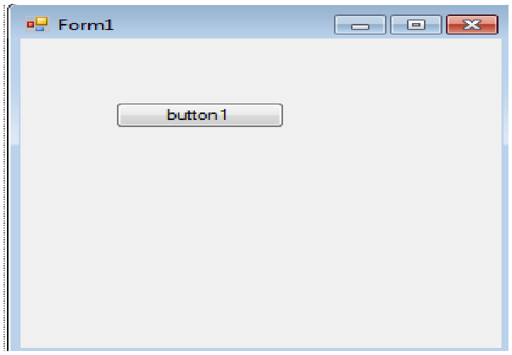
1. Create a Student table in Ms-Access named StudentDB.



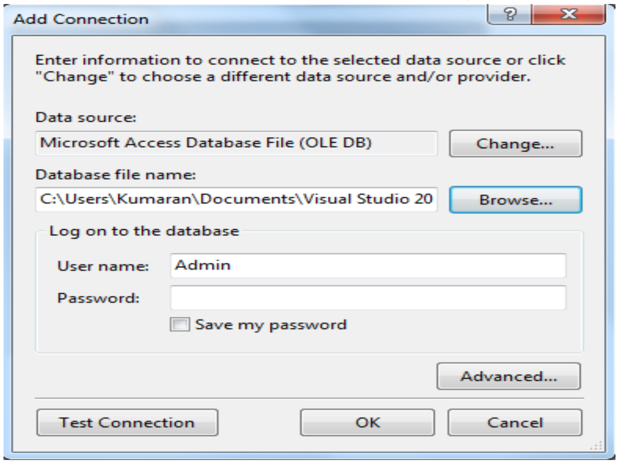
Student

Field Name	Data Type	Description
StudentName	Text	
Department	Text	
Mark1	Text	
Mark2	Text	
Mark3	Text	

2. Open Visual studio 2010 and create a new Windows application. Design Form1 with a button control.



3. Now get a connection string, Go to Tools menu and select connect to the database and browse database from the Project Directory.



Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft Access Database File (OLE DB)

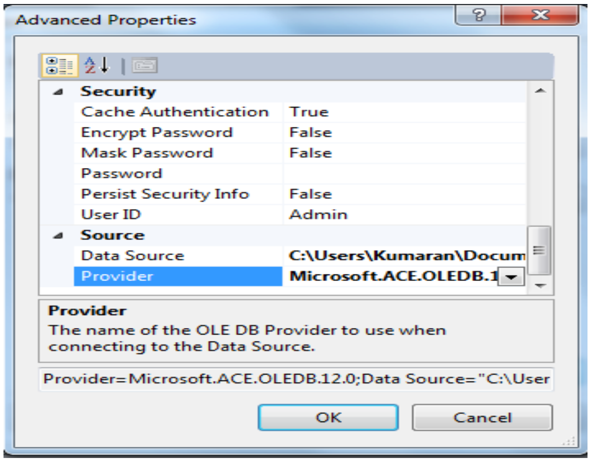
Database file name: C:\Users\Kumaran\Documents\Visual Studio 20

Log on to the database

User name: Admin

Password:

□ Save my password



Security	
Cache Authentication	True
Encrypt Password	False
Mask Password	False
Password	
Persist Security Info	False
User ID	Admin
Source	
Data Source	C:\Users\Kumaran\Docum
Provider	Microsoft.ACE.OLEDB.1

Provider
The name of the OLE DB Provider to use when connecting to the Data Source.

Provider=Microsoft.ACE.OLEDB.12.0;Data Source="C:\User

3. Edit the following code at the top before Public class from1

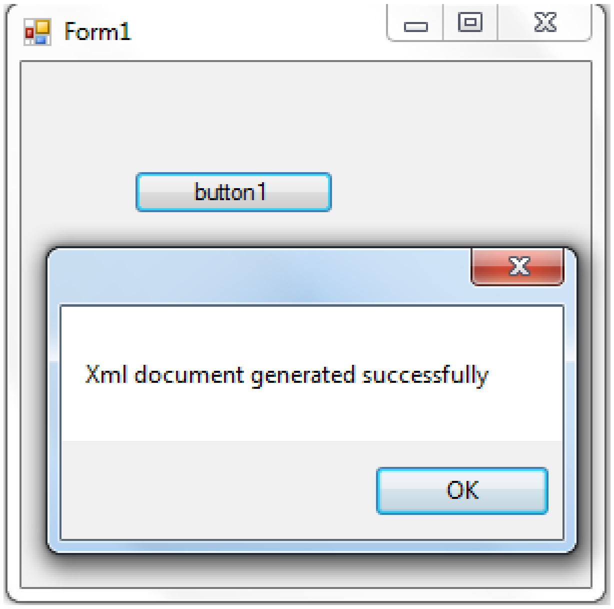
```
using System.Data.OleDb;
```

4. Edit the following code in the click event of Button1

```
{
OleDbDataAdapter oledbAdapter ;
DataSet ds = new DataSet();
OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;DataSource=StudentDB.accdb");
try
  {
    con.Open();
    oledbAdapter = new OleDbDataAdapter("select * from Student", con);
    oledbAdapter.Fill(ds);
    ds.WriteXml("Student.xml");
    MessageBox.Show("Xml document generated successfully");
    con.Close();
  }
  catch (Exception ex)
  {
    MessageBox.Show(ex.ToString());
  }
}
```

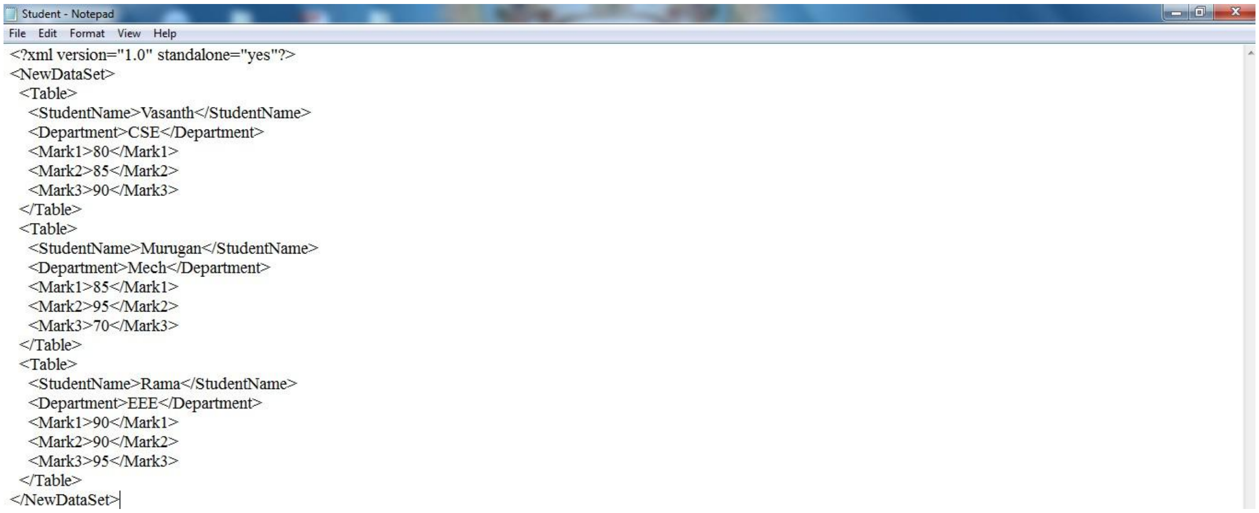
5. Run the application and find the Student.xml file in the project directory.

<u>OUTPUT</u>



Xml document generated successfully

Student.xml



```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <Table>
    <StudentName>Vasanth</StudentName>
    <Department>CSE</Department>
    <Mark1>80</Mark1>
    <Mark2>85</Mark2>
    <Mark3>90</Mark3>
  </Table>
  <Table>
    <StudentName>Murugan</StudentName>
    <Department>Mech</Department>
    <Mark1>85</Mark1>
    <Mark2>95</Mark2>
    <Mark3>70</Mark3>
  </Table>
  <Table>
    <StudentName>Rama</StudentName>
    <Department>EEE</Department>
    <Mark1>90</Mark1>
    <Mark2>90</Mark2>
    <Mark3>95</Mark3>
  </Table>
</NewDataSet>
```

RESULT:

Thus the above application has been executed and verified successfully.