OBJECTIVES:

1. Understanding and working with regular expressions.
2. Understanding and working with File system.
3. Understanding and working with menus.
4. Understanding and working with RichTextBox.
5. Understanding and working with databases.
6. Practice activities.

OBJECTIVE 1: Understanding and working with regular expressions

Regular Expressions

1. A regular expression is a pattern or set of rules that regular expression engine attempts to match in input text.

2. A pattern consists of one or more character literals, operators, or construct.

3. The Regex class is used for representing a regular expression.

4. Regular expressions provide a powerful, flexible, and efficient method for processing text.

5. System.Text.RegularExpressions.Regex is path for Regex existence.

6. At a minimum, processing text using regular expressions requires that the regular expression engine be provided with the following two items of information:

    1. The regular expression pattern to identify in the text.

    2. In the .NET Framework, regular expression patterns are defined by a special syntax or language, which is compatible with Perl 5 regular expressions and adds some additional features such as right-to-left matching

    3. The text to parse for the regular expression pattern.

```
using System;
using System.Text.RegularExpressions;

public class Example
{
   public static void Main()
   {
      string[] partNumbers= { "1298-673-4192", "A08Z-931-468A",
                              "_A90-123-129X", "12345-KKA-1230",
                              "0919-2893-1256" };
      Regex rgx = new Regex(@"^[a-zA-Z0-9]\d{2}[a-zA-Z0-9](-\d{3}){2}[A-Za-z0-9]$");
      foreach (string partNumber in partNumbers)
         Console.WriteLine("{0} {1} a valid part number.",
                           partNumber,
                           rgx.IsMatch(partNumber) ? "is" : "is not");
   }
}
```

<u>OBJECTIVE 2:</u> Understanding and working with File system.

File System

- ⇨ Accessing drives, directories and files on system.
- ⇨ *System.IO.DriveInfo class* ○ This class models a drive and provides methods and properties to query for drive information.
  - ○ Use DriveInfo to determine available drives
  - ○ You can also query to determine the capacity and available free space on the drive.
  - ○ The drive name must be either an uppercase or lowercase letter from 'a' to 'z'

```
DriveInfo[] allDrives = DriveInfo.GetDrives();
string str ="";
foreach (DriveInfo d in allDrives)
{
    str += d.Name + "\n";
}
this.lblInfo.Text = str;
```

- ⇨ ***System.IO.DirectoryInfo class*** ○ Exposes instance methods for creating, moving, and enumerating through directories and subdirectories. ○ This class cannot be inherited.

- ⇨ ***System.IO.FileInfo class*** ○ Provides properties and instance methods for the creation, copying, deletion, moving, and opening of files.
  - ○ This class cannot be inherited.

OBJECTIVE 3: Understanding and working with Menus

Menus

⇨ Menus provides a way to place multiple commands in less space in an application windows.

⇨ Menus can be fixed but can also be popped up at area where user right clicks the mouse.

⇨ Fixed menus are created by using MenuStrip class in windows forms applications.

⇨ MenuStrip is the top-level container that supersedes MainMenu.

⇨ ToolStripDropDownItem and ToolStripMenuItem work along with MenuStrip for adding menu items in menu strip or bar.

⇨ Following classes can also be used with MenuStrip.

⇨ ToolStripMenuItem
⇨ ToolStripTextBox
⇨ ToolStripComboBox
⇨ ToolStripSeparator

⇨ Popup menus are created by using ContextMenuStrip class.

⇨ It represents a short cut menu.

⇨ One can associate a ContextMenuStrip with any control, and a right mouse click automatically displays the shortcut menu.

⇨ ContextMenuStrip supports images, menu-item check state, text, access keys, shortcuts, and cascading menus.

⇨ The following items are specifically designed to work seamlessly with ContextMenuStrip.

⇨ ToolStripMenuItem
⇨ ToolStripTextBox
⇨ ToolStripComboBox
⇨ ToolStripSeparator

⇨ Shortcut menus are typically used to combine different menu items from a MenuStrip of a form that are useful for the user given the context of the application.

OBJECTIVE 4: Understanding and working with Rich Text Box.

Rich Text Box

⇨ Represents a Windows rich text box control.
⇨ The control also provides more advanced formatting features than the standard TextBox control.

⇨ Text can be assigned directly to the control, or can be loaded from a rich text format (RTF) or plain text file.
⇨ The text within the control can be assigned character and paragraph formatting.

⇨ To change the formatting of text, it must first be selected.

⇨ Only selected text can be assigned character and paragraph formatting.

⇨ Once a setting has been made to a selected section of text, all text entered after the selection is also formatted with the same settings until a setting change is made or a different section of the control's document is selected.

⇨ The SelectionFont property enables you to make text bold or italic or to change typeface & size.

⇨ The SelectionColor property enables you to change the color of the text.

⇨ To create bulleted lists you can use the SelectionBullet property.

⇨ The LoadFile method enables you to load an existing RTF or ASCII text file into the control.

⇨ The SaveFile enables you to save a file to RTF or ASCII text.

OBJECTIVE 5: Understanding and working with databases.
⇨ Database is collection of tables, that are collection of records and each record is collection of attributes of that record.

⇨ Databases are used as persistent storage mechanisms.

⇨ **Database Access**

⇨ ADO.NET supports two types of data access.
- Connected data access
- Disconnected data access

⇨ **SQLClient** namespace is imported for stated purpose.

- Classes used to work with SQL Server.
  - SqlConnection: for creating connection
    - Open() is used for opening of connection and Close() for closing the connection
  - SqlCommand: for wrapping SQL query in object
    - CommandText = “query”
    - ExecuteReader() // for query operation

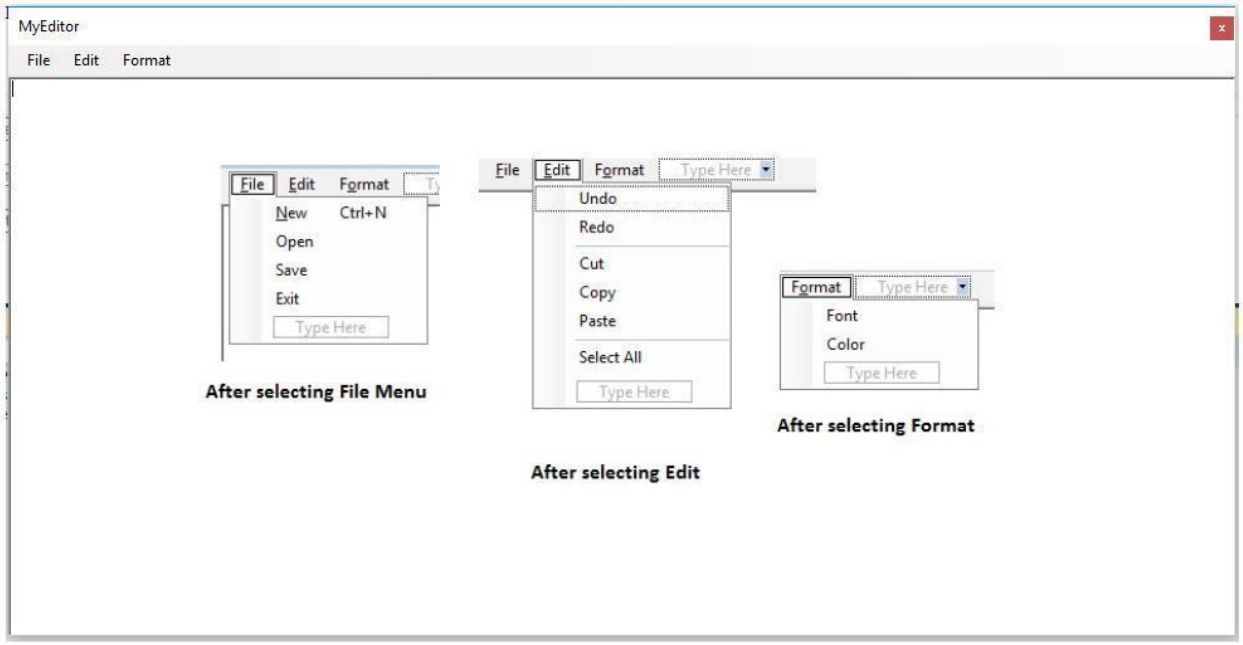
VISUAL PROGRAMMING: LAB 7 5

- ExecuteNonQuery() // for DML operation ○ SqlDataReader: provides forward access only on tables.
- Read() // read a record from recordset

ACTIVITIES SECTION

ACITVITY 1: STEPS

⇨ Create a windows forms application named EditorApp.
⇨ Create windows form in it named frmEditor.
⇨ Create Graphical interface of frmEditor as given in picture below.



⇨ Use following instance methods of rich text box for relevant functionality.
⇨ Undo(); // for undo the last action
⇨ Redo(); // for again doing the last action
⇨ Cut(); // for copying data by removing selected data from source
⇨ Copy(); // for copying data by leaving source data as it is
⇨ Paste(); // for pasting copied or cut data
⇨ SelectAll(); // for selecting complete text of rich text box.
⇨ For Font use **FontDialog** and set **SelectionFont** property for selected text of rich text box.
⇨ For Color use ColorDialog and set SelectionColor property for selected text of rich text box.
⇨ For opening file use OpenDialog to get file name and after it use LoadFile(name_of_file) method of rich text box to load a file in it.
⇨ For saving file use same path of current opened file and use SaveFile(name_of_file) method of rich text box.
⇨ After performing required operations assign access keys and shortcut keys for every menu item in menu.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

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

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

        private void newToolStripMenuItem1_Click(object sender, EventArgs e)
        {

            richTextBox1.Clear();
        }

        private void openToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            OpenFileDialog dialogbox = new OpenFileDialog();
            dialogbox.Title = "Open";
            dialogbox.Filter = "Text Document(*.txt)|*.txt| All Files (*.*)|*.* ";
            if(dialogbox.ShowDialog()== DialogResult.OK)
            {
                richTextBox1.LoadFile(dialogbox.FileName,
RichTextBoxStreamType.PlainText);
                this.Text = dialogbox.FileName;
            }
        }

        private void saveToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            SaveFileDialog dialogbox = new SaveFileDialog();
            dialogbox.Title = "Save";
            dialogbox.Filter = "Text Document(*.txt)|*.txt| All Files (*.*)|*.* ";
            if (dialogbox.ShowDialog() == DialogResult.OK)
            {
                richTextBox1.SaveFile(dialogbox.FileName,
RichTextBoxStreamType.PlainText);
                this.Text = dialogbox.FileName;
            }
        }

        private void exitToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void undoToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.Undo();
        }

        private void redoToolStripMenuItem1_Click(object sender, EventArgs e)
        {
```

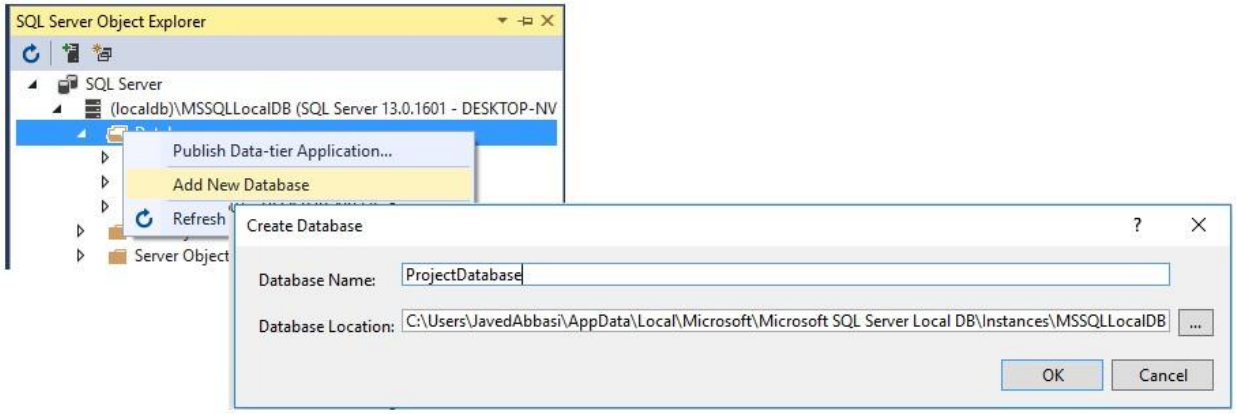
```
            richTextBox1.Redo();
        }

        private void copyToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.Copy();
        }

        private void cutToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.Cut();
        }

        private void pasteToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.Paste();
        }

        private void selectAllToolStripMenuItem1_Click(object sender, EventArgs e)
        {
            richTextBox1.SelectAll();
        }

        private void fontsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FontDialog test = new FontDialog();
            if (test.ShowDialog() == DialogResult.OK)
            {
                richTextBox1.Font = test.Font;
            }
        }

        private void colorToolStripMenuItem_Click(object sender, EventArgs e)
        {
            ColorDialog test = new ColorDialog();
            if (test.ShowDialog() == DialogResult.OK)
            {
                richTextBox1.ForeColor = test.Color;
            }
        }
    }
}
```

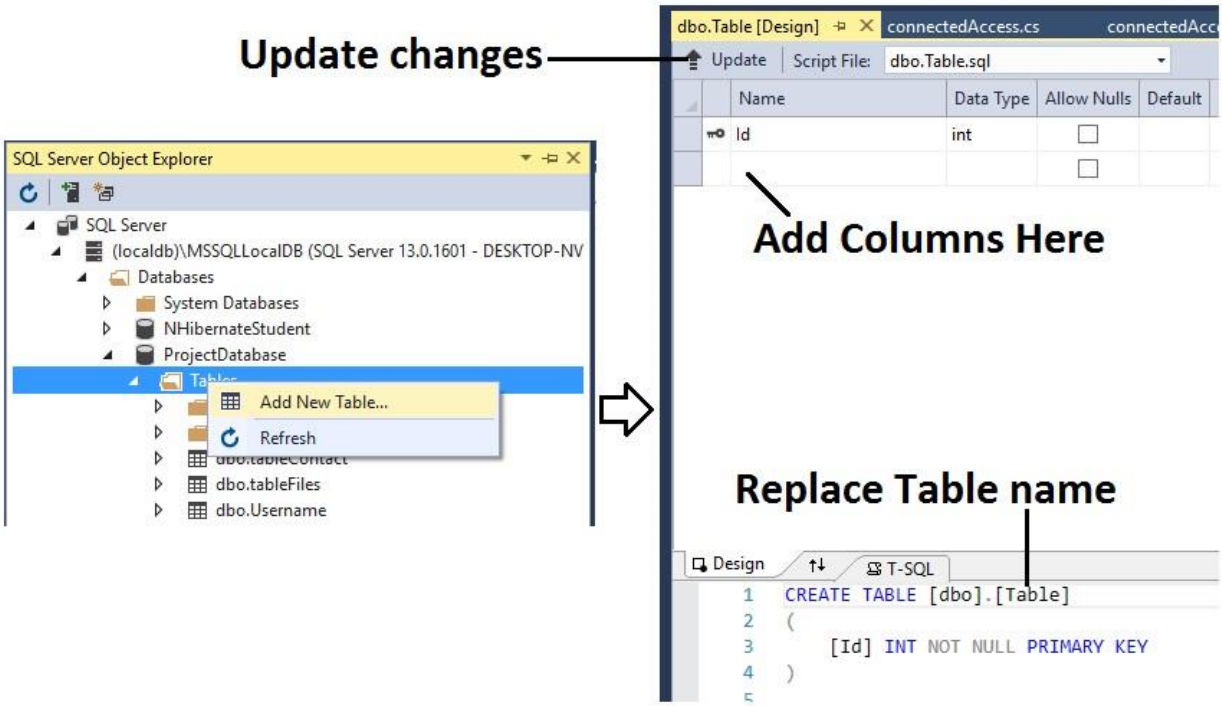
OUPUT:

ACITVITY 2: STEPS

⇨ Create a windows forms application named **DatabaseApplication.**
⇨ Create a form named **frmConnectedAccess.**
⇨ Place label on this form named **lblInfo.**

⇨ **Setup resources for this example first.**
⇨ Open Sql Server Object Explorer from View Menu or by short cut CTRL+\ and CTRL + S


SQL Server Object Explorer
SQL Server
(localdb)\MSSQLLocalDB (SQL Server 13.0.1601 - DESKTOP-NV
Publish Data-tier Application...
Add New Database
Refresh
Server Object
Create Database
Database Name:
ProjectDatabase
Database Location:
C:\Users\JavedAbbasi\AppData\Local\Microsoft\Microsoft SQL Server Local DB\Instances\MSSQLLocalDB
OK
Cancel


Update changes
dbo.Table [Design]
connectedAccess.cs
Update
Script File:
dbo.Table.sql
Name
Data Type
Allow Nulls
Default
Id
int
Add Columns Here
SQL Server Object Explorer
SQL Server
(localdb)\MSSQLLocalDB (SQL Server 13.0.1601 - DESKTOP-NV
Databases
System Databases
NHibernateStudent
ProjectDatabase
Add New Table...
Refresh
dbo.tableFiles
dbo.Username
Replace Table name
Design
T-SQL
CREATE TABLE [dbo].[Table]
(
[Id] INT NOT NULL PRIMARY KEY
)

⇨ **Managing connection string now.**
⇨ Open app.config file.
⇨ Add following lines of code as child element of configuration element in xml file.

```
<connectionStrings>
  <add name="cAString"
       connectionString="Data Source=(localDB)\MSSQLLocalDB;
       initial catalog=ProjectDatabase;
       integrated security=SSPI;"
   />
</connectionStrings>
```

⇨ **Setup for classes used in this program.**
⇨ Import System.Data.SqlClient namespace in program by use of **using** keyword.
⇨ Import System.Configuration namespace by use of **using** keyword.
⇨ Now add a reference of System.Configuration namespace by right clicking on references option of project in solution explorer.
⇨ Create method named getUserData() and in its implementation place following code.

```
string connectionString = ConfigurationManager.ConnectionStrings["cAString"].ConnectionString;

// Creating connection object
SqlConnection connection = new SqlConnection(connectionString);

// Creating command object
SqlCommand command = new SqlCommand();

// Assining connection object to connection property of command object
command.Connection = connection;

// Assining query to commantText property of command object
command.CommandText = "Select * from username";

// opening connection
connection.Open();
```

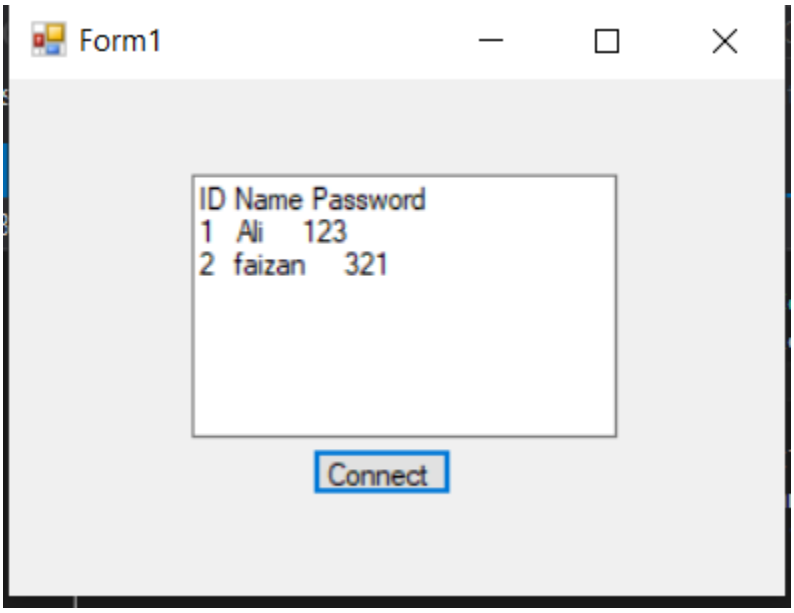
```
// executing query using command object and assining it to datareader object
SqlDataReader datareader = command.ExecuteReader();

// reading records one by one by using Read method of datareader object
while (datareader.Read())
{
    // accessing id column by using datareader
    id = (int)datareader[0];    // or datareader.GetInt32(0);

    // accessing name column by using datareader
    username = (string)datareader["username"]; // datareader.GetString(1);

    // password from password column
    password = (string)datareader["password"]; //datareader.GetString(2);

    this.lblInfo.Text += id + "\n";
    this.lblInfo.Text += username + "\n";
    this.lblInfo.Text += password + "\n\n";
}
// closing datareader object
datareader.Close();
// closing connection close
connection.Close();
```

⇨ Execute the program and observe the results.

```
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.Data.SqlClient;

namespace DataBase
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        string connectionstring = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=master;Integrated Security=True;";
        private void button1_Click(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection
            SqlConnection connection = new SqlConnection();
```

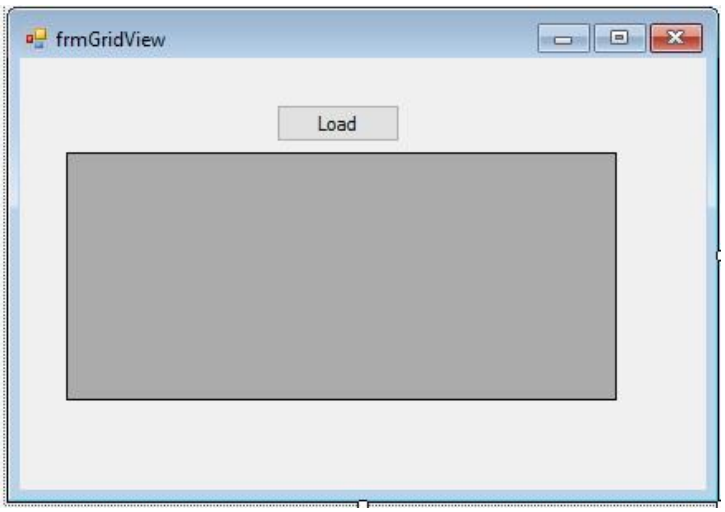
```
            connection.ConnectionString = connectionstring;
            connection.Open();

            //step 2 Create sql command

            SqlCommand commands = new SqlCommand();
            commands.Connection = connection;

            // step 3 run quesries
            commands.CommandText = @"SELECT *FROM malik";
            SqlDataReader datareader = commands.ExecuteReader();
            textBox1.Text += "ID" + " " + "Name" + " " + "Password" + "\r\n";
            while (datareader.Read()) {

                String data = datareader.GetValue(0) + "   " + datareader.GetValue(1) + "
" + datareader.GetValue(2);
                textBox1.Text += data + "\r\n";
            }


        }
    }
}
```

OUTPUT:



ACITVITY 3: STEPS

- ⇨ Create a form in application named frmGridView.
- ⇨ Place button on this form.
- ⇨ Place GridView on this form.



- Create a class named Person with following auto implemented properties.
  - FirstName
  - LastName
  - City

- In Load event of form create an array of Person class with length 5.
- Populate array with objects of Person class and objects of Person class should also contain data of every property.
- Now in Click event of button assign array to DataSource property of GridView control.

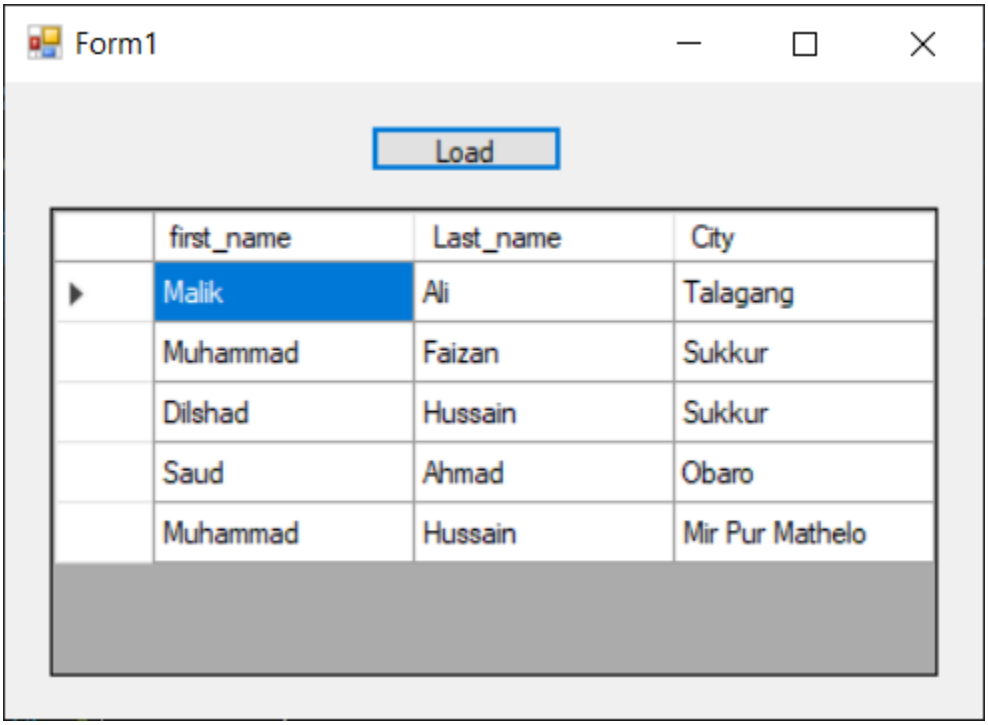
Code:

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

namespace Lab_07
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
```

```
        private person[] arr = new person[5];

        private void Form1_Load(object sender, EventArgs e)
        {


            dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;




        }

        private void button1_Click(object sender, EventArgs e)
        {

            arr[0] = new person("Malik", "Ali", "Talagang");
            arr[1] = new person("Muhammad", "Faizan", "Sukkur");
            arr[2] = new person("Dilshad", "Hussain", "Sukkur");
            arr[3] = new person("Saud", "Ahmad", "Obaro");
            arr[4] = new person("Muhammad", "Hussain", "Mir Pur Mathelo");
            dataGridView1.DataSource = arr;
        }
    }
    public class person
    {
        String First_name;
        String last_name;
        String city;
        public person() { }
       public  person(String First_name,
        String last_name,
        String city)
        {
            this.first_name = First_name;
            this.Last_name = last_name;
            this.City = city;
        }
        public string first_name
        {
            get
            {
                return First_name;
            }
            set
            {
                First_name = value;
            }
        }
        public string Last_name
        {
            get { return last_name; }
            set { last_name = value; }
        }
        public string City
        {
            get { return city; }
            set { city = value; }
        }
```

```
    }
}
```

OUTPUT:



ACITVITY 4: STEPS

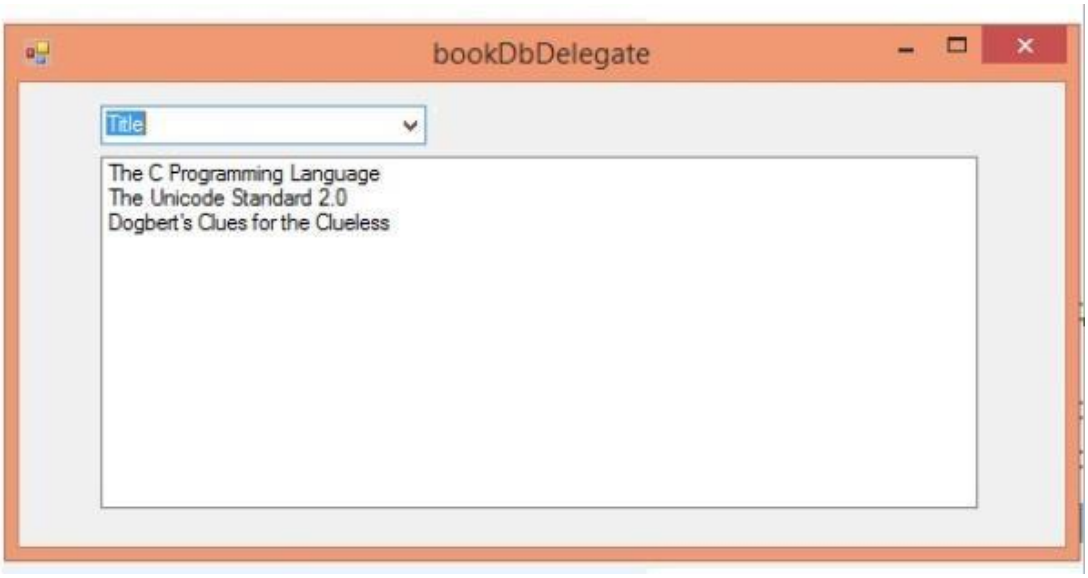
- Create form named frmBookStore.
- Place controls on form as displayed in exhibit.

- combo box has values like Title and Author to display list of books accordingly in ListBox.

-There is a class named BookDB having list of books or maintaining book database.
-This class exposes a method to other classes, which passes instance of Book, that's structure for further processing.
-In other words method requires a delegate to which it passes a Book's instance.

Requirements of Task

- use the exposed method by passing it a delegate instance and display title of every book maintained by BookDB class in ListBox.
-use the exposed method to calculate the average cost of books also.



```
// Describes a book in the book list:
public struct Book
{
    public string Title;        // Title of the book.
    public string Author;       // Author of the book.
    public decimal Price;       // Price of the book.
    public bool Paperback;      // Is it paperback?

    public Book(string title, string author, decimal price, bool paperBack)
    {
        Title = title;
        Author = author;
        Price = price;
        Paperback = paperBack;
    }
}

public class BookDB
{// List of all books in the database:
    ArrayList list = new ArrayList();
    // Add a book to the database:
    public void AddBook(string title, string author, decimal price, bool paperBack)
    {
        list.Add(new Book(title, author, price, paperBack));
    }
    // Call a passed-in delegate on each paperback book to process it:
    public void ProcessPaperbackBooks(ProcessBookDelegate processBook)
    {
        foreach (Book b in list)
        {
            if (b.Paperback)
                // Calling the delegate:
                processBook(b);
        }
    }
}
```

```
// Initialize the book database with some test books:
public void AddBooks(BookDB bookDB)
{
    bookDB.AddBook("The C Programming Language",
       "Brian W. Kernighan and Dennis M. Ritchie", 19.95m, true);
    bookDB.AddBook("The Unicode Standard 2.0",
       "The Unicode Consortium", 39.95m, true);
    bookDB.AddBook("The MS-DOS Encyclopedia",
       "Ray Duncan", 129.95m, false);
    bookDB.AddBook("Dogbert's Clues for the Clueless",
       "Scott Adams", 12.00m, true);
}
```

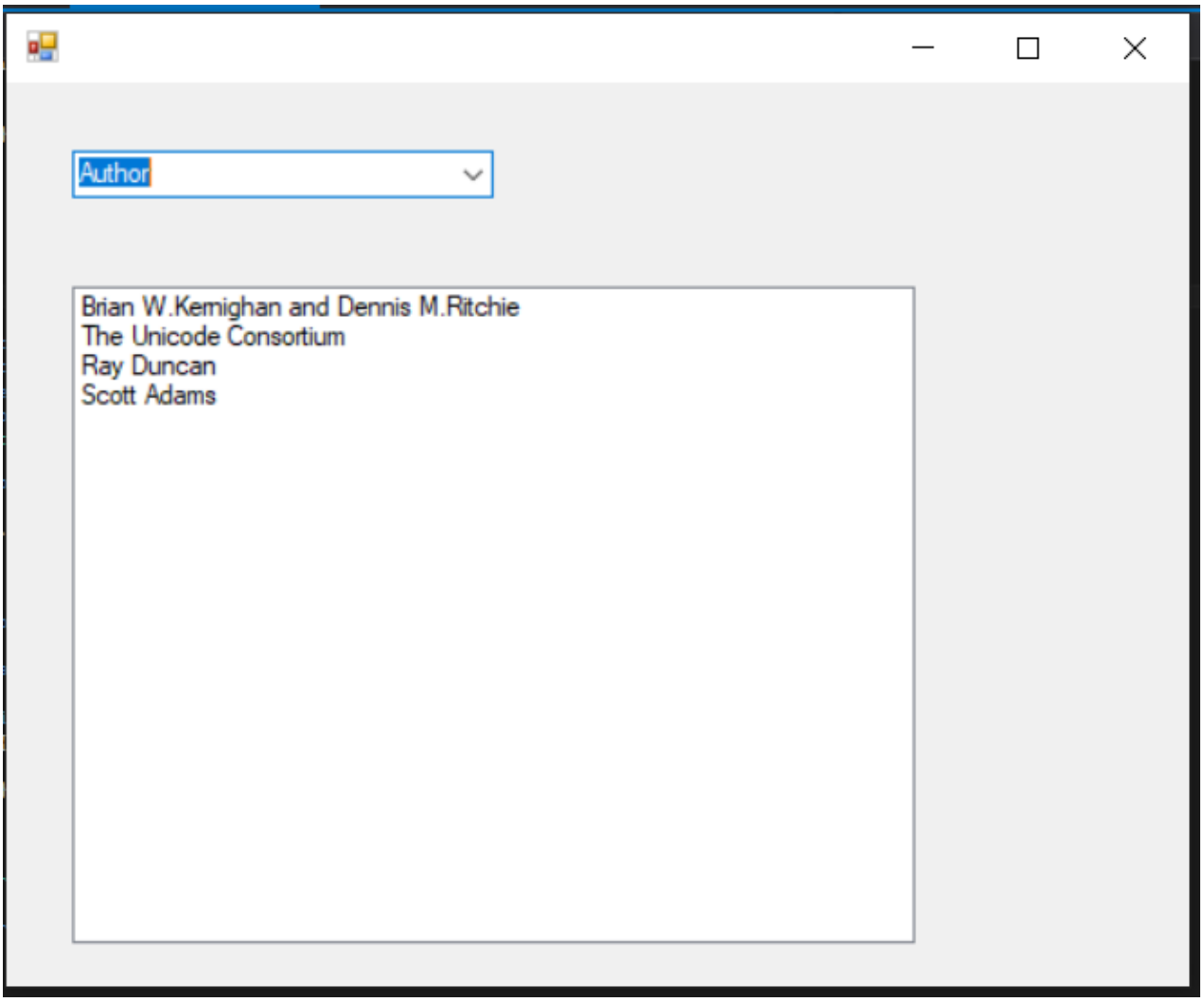
Code :

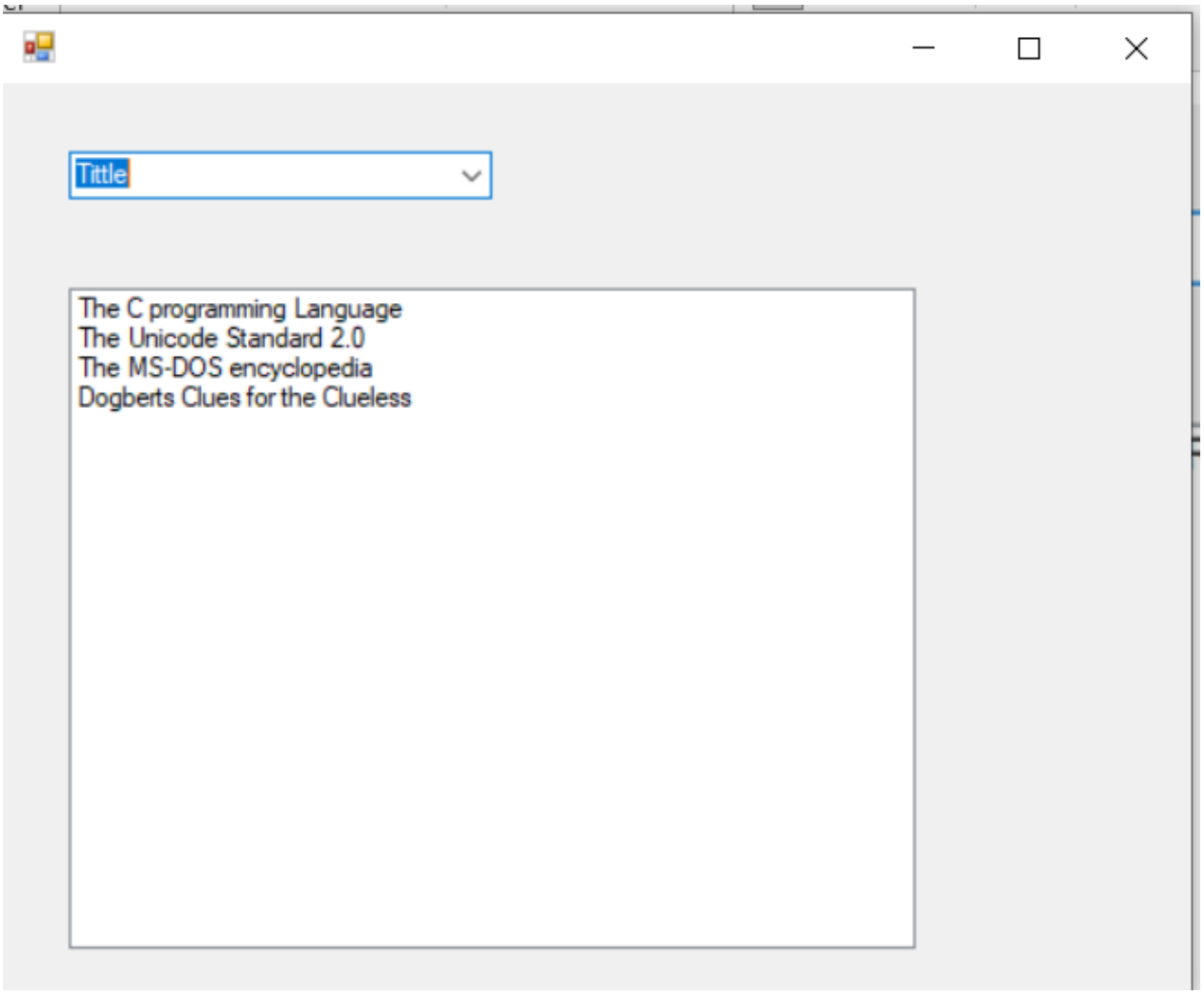
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Collections;
namespace DB
{
    public partial class Form1 : Form
    {
        public delegate void ProcessBookDelegate(book b);
        ArrayList list;
        public struct book
        {
            public string tittle;
            public string author;
            public decimal price;
            public bool paperbook;

            public book(string Tittle, string Author, decimal Price, bool Paperbook)
            {
                tittle = Tittle;
                author = Author;
                price = Price;
                paperbook = Paperbook;

            }
        }
        public Form1()
        {
            InitializeComponent();
            BookDB book = new BookDB();
            add(book);
            list = book.Getter();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
```

```
        }
        public void add(BookDB book)
        {
            book.addBook("The C programming Language", "Brian W.Kernighan and Dennis
M.Ritchie", 19.95m, true);
            book.addBook("The Unicode Standard 2.0","The Unicode Consortium",39.95m,true);
            book.addBook("The MS-DOS encyclopedia","Ray Duncan",129.95m,true);
            book.addBook("Dogberts Clues for the Clueless","Scott Adams",12.00m,true);

        }


        private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
        {
            string str = comboBox2.SelectedItem.ToString();
            listBox2.Items.Clear();
            if (str == "Tittle")
                for (int i = 0; i < list.Count; i++)
                {
                    book book = (book)list[i];
                    listBox2.Items.Add(book.tittle);
                }
            else if (str == "Author")
                for (int i = 0; i < list.Count; i++)
                {
                    book book = (book)list[i];
                    listBox2.Items.Add(book.author);
                }
        }

    }

    public class BookDB
    {

        public string Title;
        public string Author;
        public decimal Price;
        public bool Paperbook;
        ArrayList list = new ArrayList();
        public void addBook(string Tittle, string Author, decimal Price, bool Paperbook)
        {
            list.Add(new Form1.book(Tittle, Author, Price, Paperbook));
        }

        public void processpaperbook(Form1.ProcessBookDelegate processBook)
        {
            foreach(Form1.book b in list)
            {
                if (b.paperbook)
                {
                    processBook(b);
                }
            }
        }
        public ArrayList Getter()
        {
            return list;
        }
    }
```

```
}
```

OUTPUT:

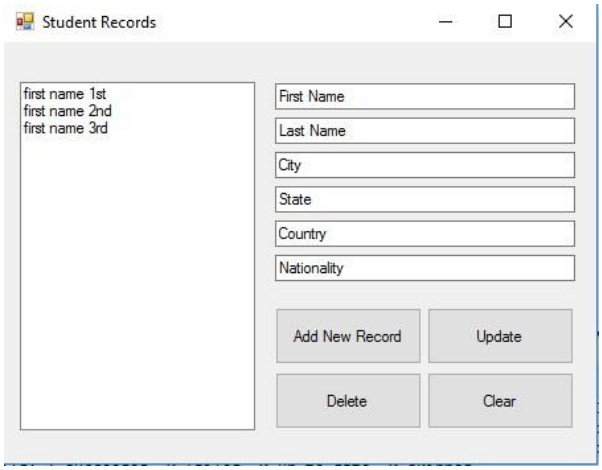

Author
Brian W.Kernighan and Dennis M.Ritchie
The Unicode Consortium
Ray Duncan
Scott Adams

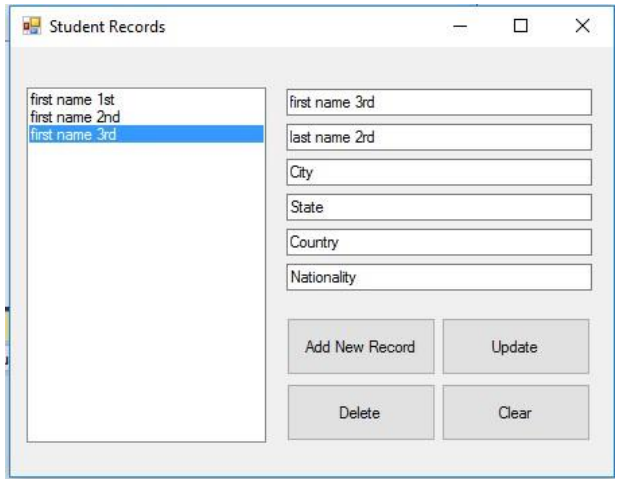


ACITVITY 5: STEPS

⇨ Create a windows forms application named StudentManagement.
⇨ Create interface of form according to given image. ⇨ Before execution.

After Execution





[NOTE: Use **ExecuteNonQuery()** method of Command object for dml operations]

CODE:

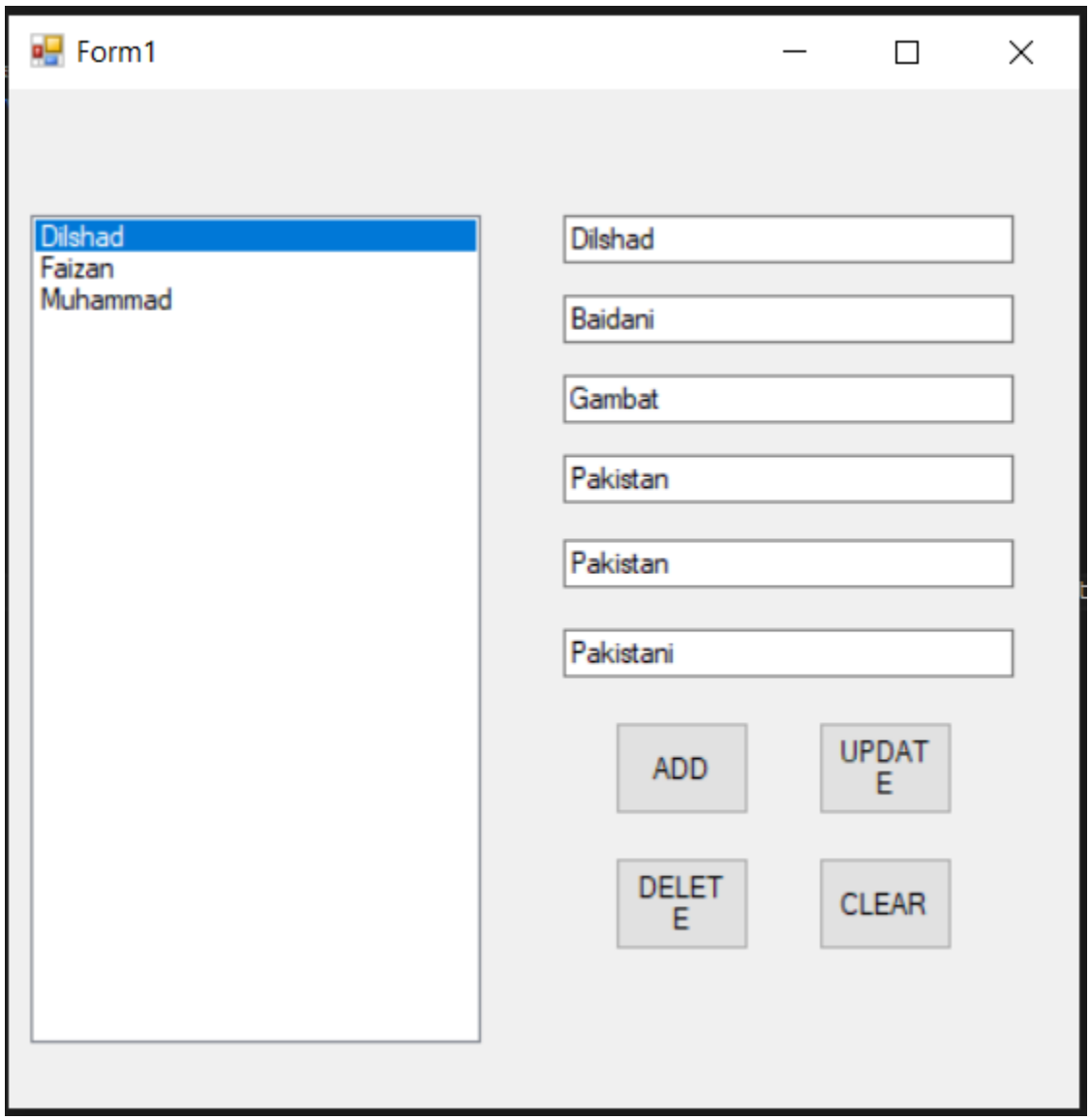
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace STDBMS
{
```

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

        string connectionstring = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=STDDB;Integrated Security=True;";


        private void Form1_Load(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = connectionstring;
            connection.Open();

            //step 2 Create sql command

            SqlCommand commands = new SqlCommand();
            commands.Connection = connection;

            // step 3 run quesries
            commands.CommandText = @"SELECT *FROM stdDB";
            SqlDataReader datareader = commands.ExecuteReader();

            while (datareader.Read())
            {

                listBox1.Items.Add(datareader.GetValue(0));
            }


        }
        public void reset()
        {
            textBox1.Text = "First Name";
            textBox2.Text = "Last Name";
            textBox3.Text = "City";
            textBox4.Text = "State";
            textBox5.Text = "Country";
            textBox6.Text = "Nationality";
        }
        private void button1_Click(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection

            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = connectionstring;
            connection.Open();

            //step 2 Create sql command

            SqlCommand commands = new SqlCommand();
            commands.Connection = connection;
            int i = 2;


            // step 3 run quesries
```

```
            commands.CommandType = CommandType.Text;
            commands.CommandText = "INSERT into stdDB
(First_Name,Last_Name,City,State,Country,Nationality) VALUES
('"+textBox1.Text+"','"+textBox2.Text+"','"+textBox3.Text+"','"+textBox4.Text+"','"+textBo
x5.Text+"','"+textBox6.Text+"')";



            commands.ExecuteNonQuery();
            MessageBox.Show("SuccessFully Updated");
            listBox1.Items.Add(textBox1.Text);
            reset();
            i++;
            connection.Close();
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {
            reset();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = connectionstring;
            connection.Open();

            //step 2 Create sql command

            SqlCommand commands = new SqlCommand("delete from stdDb where
First_Name=@First_Name");
            commands.Connection = connection;
            commands.Parameters.AddWithValue("@First_Name",textBox1.Text);
            SqlDataAdapter da = new SqlDataAdapter(commands);
            DataSet ds = new DataSet();
            da.Fill(ds);
            MessageBox.Show("Deleted Successfully");
            connection.Close();

            listBox1.Items.Remove(textBox1.Text);


        }

        private void button2_Click(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = connectionstring;
            connection.Open();
```

```
              //step 2 Create sql command

            SqlCommand commands = new SqlCommand();
            commands.Connection = connection;

            // step 3 run quesries
                    SqlCommand command = new SqlCommand("Update stdDb set
First_Name=@First_Name , Last_Name=@Last_Name , City=@City , State=@State ,
Country=@Country , Nationality=@Nationality where First_Name=@First_Name");
                    command.Connection = connection;
                    command.Parameters.AddWithValue("@First_Name", textBox1.Text);
                    command.Parameters.AddWithValue("@Last_Name", textBox2.Text);
                    command.Parameters.AddWithValue("@City", textBox3.Text);
                    command.Parameters.AddWithValue("@State", textBox4.Text);
                    command.Parameters.AddWithValue("@Country", textBox5.Text);
                    command.Parameters.AddWithValue("@Nationality", textBox6.Text);
                    command.ExecuteNonQuery();
                    MessageBox.Show("Updated Successfully");

                    connection.Close();



        }

        private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            //step 1 Create SQL Connection
            SqlConnection connection = new SqlConnection();
            connection.ConnectionString = connectionstring;
            connection.Open();

            //step 2 Create sql command

            SqlCommand commands = new SqlCommand();
            commands.Connection = connection;

            // step 3 run quesries
            commands.CommandText = @"SELECT *FROM stdDB where
First_Name='"+listBox1.Text+"'";
            SqlDataReader datareader = commands.ExecuteReader();

            while (datareader.Read())
            {

                textBox1.Text =datareader.GetValue(0).ToString();
                textBox2.Text = datareader.GetValue(1).ToString();
                textBox3.Text = datareader.GetValue(2).ToString();
                textBox4.Text = datareader.GetValue(3).ToString();
                textBox5.Text = datareader.GetValue(4).ToString();
                textBox6.Text = datareader.GetValue(5).ToString();
            }
        }
    }
    }
```

OUTPUT:


Form1
Dilshad
Faizan
Muhammad
Dilshad
Baidani
Gambat
Pakistan
Pakistan
Pakistani
ADD
UPDAT
E
DELET
E
CLEAR