

DEPARTMENT OF COMPUTER SCIENCE

Periyar Nagar, Vallam Thanjavur - 613403, Tamil Nadu, India

Phone: +91 - 4362 - 264600 Fax: +91 - 4362 - 264660

Email: headmcs@pmu.edu Web: www.pmu.edu



**PERIYAR
MANIAMMAI**
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University)
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited
think • innovate • transform

XAI506A-.NET TECHNOLOGIES LABORATORY

B.SC AI (III YEAR-V SEM)

(2022-2024)

Class Incharge Signature :

HOD Signature :

Dean Signature :

Dean Academic Signature :

DEPARTMENT OF COMPUTER SCIENCE

DEPARTMENT VISION	
To be a leading department in the field of software development and digital design that offers the software education with the State-of-the-art skills. The Graduates will be recognized as globally competent by their dynamic work and produce valuable digital solutions for the society.	
DEPARTMENT MISSION	
DM1	To construct the software related technical skills among the students.
DM2	To practice the cutting-edge technologies in the various areas of digital design and software development.
DM3	To contribute towards the betterment of the society by producing enhanced software solutions through research.
DM4	To generate the spirit of inquiry, teamwork, novelty and professionalism among the students.

Mapping of University Mission (UM) and Department Mission (DM)

	DM1	DM2	DM3	DM4	Total
UM1	2	3	1	0	6
UM2	1	2	0	2	5
UM3	1	1	3	0	5
UM4	3	1	1	1	6
UM5	0	0	2	3	5

1-Low

2-Medium

3-High

PROGRAMME EDUCATIONAL OBJECTIVES (PEO)

Based on the mission of the department, the programme educational objective is formulated as The B.Sc. Computer Science dedicated to produce graduates who have ability to

PEO1	evolve as globally proficient computer professionals by giving enriched performance in problem solving, analysis and synthesis for computer science related issues..
PEO2	exercise with contemporary tools and technologies to provide an effective user friendly interface for the real time social concerns.
PEO3	communicate effectively in a multidisciplinary team and manage the team members through the acquired leadership skills to achieve the target in time.
PEO4	handle the customers and stakeholder effectively with the awareness of human values and ethical concerns.
PEO5	pursue lifelong learning through the cutting-edge Learning Management Systems and thus satisfy the up-to-date industry expectations.

Mapping of Program Educational Objectives (PEOs) with Department Mission (DM)

B.Sc.(CS)	PEO1	PEO2	PEO3	PEO4	Total
DM1	3	2	0	0	5
DM2	2	2	1	1	6
DM3	1	2	1	1	5
DM4	0	1	3	1	5

1- Low

2-Medium

3-High

PROGRAMME OUTCOME (PO)

At the time of graduation, competency of the student is measured through the attainment of programme outcomes. The quantification of programme outcomes attainment is measured through the assessment of established course outcomes for each course.

Graduates of the B.Sc. Computer Science programme will have attained the ability to

PROGRAMME OUTCOMES	
PO1	identify and analyze the acquainted or unacquainted real time issues and afford solution using the necessary computing, mathematical and basic science skill set.
PO2	design and develop algorithms for providing an appropriate solution to gratify the industrial and social needs.
PO3	express ideas and thoughts effectively to the team members and customers

	through written and oral communication.
PO4	work jointly with different team members in order to complete the agreed work in time.
PO5	inspire and guide the team members using management skills to achieve the target in an efficient and smooth way.
PO6	provide a remarkable impact on the society by contributing resolutions to social issues with the awareness of ethical responsibility by discriminating ethical & unethical behaviors and understanding human, professional values & responsibilities.
PO7	utilize computer literacy in the learning and working places and self-adapt with the changing environment by participating in learning activities throughout the life.
PROGRAMME SPECIFIC OUTCOME	
PSO1	provide the professional user friendly interface with the help of state-of-the-art tools and technologies.
PSO2	design the interactive & responsive web-based and mobile applications.

Learning Objectives and Course Outcomes

Learning Objectives:

This course aims at

- facilitating the student to understand the various concepts and functionalities of Database Management Systems, the method and model to store data.
- How to manipulate through query languages, the effective designing of relational database.
- How the system manages the concurrent usage of data in multi-user environment.

LIST OF EXPERIMENTS

S.No	Experiment	PageNo
1.	WorkingWithConsoleApplications	6
2.	ConsoleApplicationusingconditionalandLoopingstatements	9
3.	SimplecalculatorprogramusingC#.NetwindowsApplication	13
4.	WorkingwithvariousControlssuchastimer,calendar,etc.,	16
5.	AccessingDatawithADO.NET	18
6.	InsertUpdateDeleteSelectSearchOperationUsingOledbConnection	20
7.	WorkingwithvariousControlsinASP.NET	25
8.	UsingValidationControls	27
9.	Usingstored Procedures	30
10.	UsingRequiredFieldValidatorControl	34

AIM

To understand about basics of C# and execute simple C# programs to perform the following actions: (a). Create simple Console Application Program to display a text message.
(b). Taking nonnumerical data from keyboard into Console Application. (c). Taking numerical data in Console Application

ALGORITHM

Step 1: Open Visual Studio Express edition 2010

Step 2: Click File New project Select C# under installed tab and select console application Step 3:

Give name for your application and click OK

Step 4: Give any class name and declare variables and write methods Step 5:

Create objects for classes to execute methods

Step 6: Click save and click run button for execution

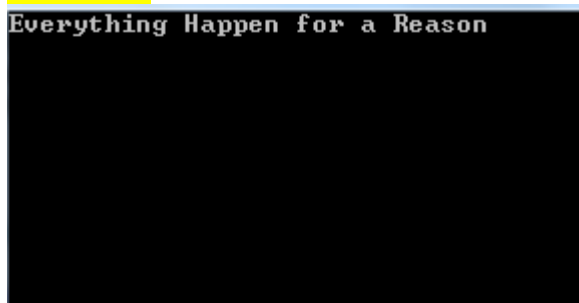
PROGRAM:

(a). Create simple Console Application Program to display a text message. `using`
`System;`
`using System.Collections.Generic;`
`using System.Linq;`
`using System.Text;`

```
namespace lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Everything Happen for a Reason");

            Console.ReadKey();
        }
    }
}
```

OUTPUT:



```
Everything Happen for a Reason
```

(b). Taking nonnumerical data from keyboard into Console Application.

PROGRAM:

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

```
namespace ConsoleApplication2
{
    class Program
```



```

{
    static void Main(string[] args)
    {
        string name = "";
        Console.WriteLine("Please enter your Name");
        name = Console.ReadLine();
        Console.WriteLine("name:" + name);
        Console.ReadKey();
    }
}

```

OUTPUT:



```

Please enter your Name
Atchatha
name:Atchatha

```

(c). Taking numerical data in Console Application

PROGRAM:

```

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

namespace ConsoleApplication3
{
    class Program
    {
        static void Main(string[] args)
        {
            int age = 0;
            Console.WriteLine("Please enter your Age:");
            age = Convert.ToInt16(Console.ReadLine());
            Console.WriteLine("Age:" + age);
            Console.ReadKey();
        }
    }
}

```

OUTPUT:


```
Please Enter your Age :  
21  
Age:21
```

RESULT:

ThustounderstandaboutbasicsofC#andexecutesimple c#programshasbeen verified

AIM:

To understand about basics of C# and execute simple C# programs to perform the following actions:

- (a) Calculate the quadrant for the coordinates using if..else...ladder
- (b) Check whether the alphabet is a vowel or not using switch..case...
- (c) To understand about for..each loop and strings

ALGORITHM:

Step 1: Open Visual Studio Express edition 2010

Step 2: Click File New project Select C# under installed tab and select console application Step 3:

Give name for your application and click OK

Step 4: Give any class name and declare variables and write methods Step 5:

Create objects for classes to execute methods

Step 6: Click save and click run button for execution

PROGRAM CODING:

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

namespace ConsoleApplication4
{
    class Program
    {
        static void Main(string[] args)
        {
            int co1, co2;
            Console.WriteLine("\n\n");
            Console.WriteLine("Find the quadrant in which the coordinate point lies:\n"); Console.WriteLine("_");
            Console.WriteLine("\n\n");
            Console.WriteLine("Input the value for X coordinate:"); co1
            = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Input the value for Y coordinate:"); co2
            = Convert.ToInt32(Console.ReadLine());
            if (co1 > 0 && co2 > 0)
            {
                Console.WriteLine("The coordinate point ({0} {1}) lies in the First quadrant.\n\n", co1, co2); else
            if (co1 < 0 && co2 > 0)
            {
                Console.WriteLine("The coordinate point ({0} {1}) lies in the second quadrant.\n\n", co1, co2); else
            if (co1 < 0 && co2 < 0)
            {
                Console.WriteLine("The coordinate point ({0} {1}) lies in the Third quadrant.\n\n", co1, co2); else
            if (co1 > 0 && co2 < 0)
            {
                Console.WriteLine("The coordinate point ({0} {1}) lies in the Fourth quadrant.\n\n", co1, co2); else
            if (co1 == 0 && co2 == 0)
            {
                Console.WriteLine("the coordinate point ({0} {1}) lies at the origin.\n\n", co1, co2);
            Console.ReadKey();
            }
        }
    }
}
```

OUTPUT:

```
Find the quadrant in which the coordinate point lies:

Input the value for X coordinate:2
Input the value for Y coordinate:6
The coordinate point (2,6) lies in the First quadrant.
```

B) Program:

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

namespace ConsoleApplication5
{
    class Program
    {
        static void Main(string[] args)
        {
            char ch;
            Console.WriteLine("\n\n");
            Console.WriteLine("check whether the input alphabet is vowel or not:\n");
            Console.WriteLine("_____");
            Console.WriteLine("\n\n");
            Console.WriteLine("input an alphabet (A-Z or a-z):");
            ch = Convert.ToChar(Console.ReadLine().ToLower());
            int i = ch;
            if (i >= 48 && i <= 57)
            {
                Console.WriteLine("You entered a number, please enter an alphabet.");
            }
            else
            {
                switch (ch)
                {
                    case 'a':
                        Console.WriteLine("the alphabet is vowel");
                        break;
                    case 'i':
                        Console.WriteLine("the alphabet is vowel");
                        break;
                    case 'o':
                        Console.WriteLine("the alphabet is vowel");
                        break;
                    case 'u':
                        Console.WriteLine("the alphabet is vowel");
                        break;
                    case 'e':
                        Console.WriteLine("the alphabet is vowel");
                        break;
                    default:
                        Console.WriteLine("The alphabet is consonant");
                        break;
                }
            }
        }
    }
}
```

```

    }
    Console.ReadKey();
}
}
}
}

```

Output:

```

check whether the input alphabet is vowel or not:
-----
input an alphabet <A-Z or a-z>:i
the Alphabet is vowel

```

2.c.StringlengthProgram:

```

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

namespace ConsoleApplication6
{
    class Program
    {
        static void Main(string[] args)
        {
            string str;
            int length=0;
            Console.WriteLine("\n\n Find the length of a string:");
            Console.WriteLine("_____ \n");
            Console.WriteLine("input the string:");
            str=Console.ReadLine();
            foreach (char chr in str)
            {
                length+= 1;
            }
            Console.WriteLine("Length of the String is: {0} \n\n", length);
            Console.ReadKey();

        }
    }
}

```

Output:

```
Find the length of a string:_____
input the string:Nothing is permanent
Length of the String is:20
```

Result:

ThustounderstandaboutbasicsofC#andexecutesimplec#programshasbeen verified

Ex.No:3

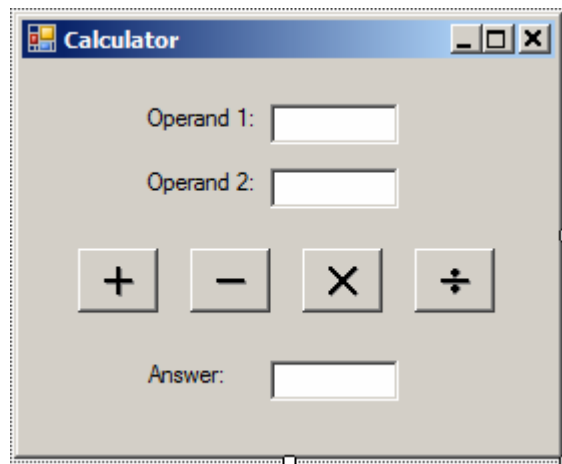
Simplecalculatorprogramusing C#.Netwindows Application

AIM

To build simple calculator that performs addition, subtraction, multiplication and division using C#.Net windows application

ALGORITHM:

1. Create a new C# Windows Forms Application named **MyCalculator**. Name the form class and the associated file **Calculator**. Save the solution.
2. Design the form window controls (from toolbox) for the four arithmetic operations
3. Set the properties of each control
4. Trap the **Click** event for each of the four buttons that specify math operations.
5. In each handler, write code to convert the string data in each text box to a floating point value. Perform the appropriate math operation for the button.
6. Finally, place the result back in the text box that holds the answer. Compile and run the program.



FORM DESIGN:

PROGRAM CODING:

a) Simple calculator program using C#.net window form application.

```
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 WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

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

        private void button1_Click(object sender, EventArgs e)
        {
            var a = Convert.ToInt32(textBox1.Text);
            var b = Convert.ToInt32(textBox2.Text);
            var c = a + b;
            textBox3.Text = c.ToString();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            var a = Convert.ToInt32(textBox1.Text);
            var b = Convert.ToInt32(textBox2.Text);
            var c = a - b;
            textBox3.Text = c.ToString();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            var a = Convert.ToInt32(textBox1.Text);
            var b = Convert.ToInt32(textBox2.Text);
            var c = a * b;
            textBox3.Text = c.ToString();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            var a = Convert.ToInt32(textBox1.Text);
            var b = Convert.ToInt32(textBox2.Text);
            var c = a % b;
            textBox3.Text = c.ToString();
        }
    }
}

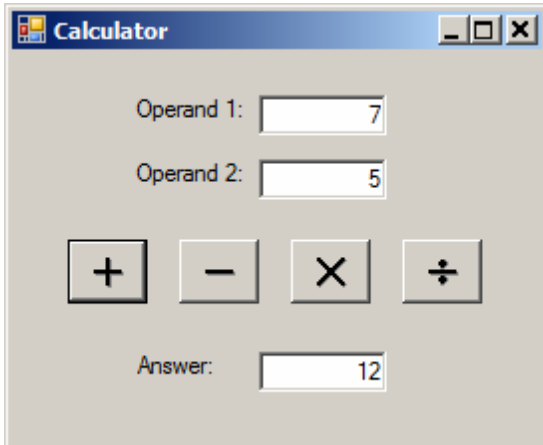
```

```

    }
}

```

OUTPUT:



B)USINGCHECKBOX

```

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 WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void checkBox1_CheckedChanged(object sender, EventArgs e)
            {
                label1.Font = new Font(label1.Font, FontStyle.Bold);
            }

            private void checkBox2_CheckedChanged(object sender, EventArgs e)
            {
                label1.Font = new Font(label1.Font, FontStyle.Italic);
            }
        }
    }
}

```

RESULT:

Thus to build c#. Net windows application and access various controls has been verified.

AIM:

To create DateTimePicker control to display current date and time using c#.net window form application.

ALGORITHM:

1. Create a new project -> Windows Application -> Name -> ok
2. Design the form window controls (from toolbox) Drag and drop DateTimePicker control.
3. Set the properties of the control
4. Write the coding to display system date and time in form load event.
5. Finally Compile and run the program.

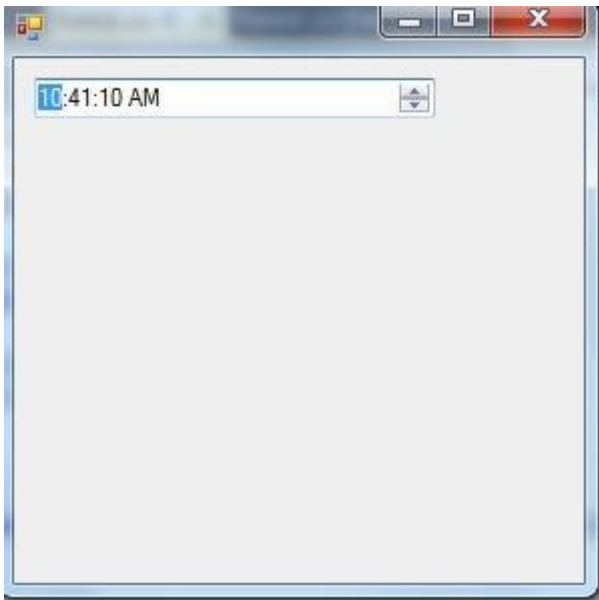
PROGRAM CODING:

```
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 WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private DateTimePicker timePicker;
            private void form1_load(object sender, EventArgs e)
            {
                timePicker = new DateTimePicker();
                timePicker.Format = DateTimePickerFormat.Time;
                timePicker.ShowUpDown = true; timePicker.Width
                = 100; Controls.Add(timePicker);
            }
            [STAThread]
            static void main()
            {
                Application.EnableVisualStyles();
                Application.Run(new Form1());
            }
        }
    }
}
```

Output



Result:

Thustobuildc#. Net windowsapplicationandusingDateTimePickercontrolshasbeenverified.

AIM:

To create C#.net console application to connect Ms Access database to display the table values using OleDbConnection object.

ALGORITHM:

1. Create new project -> Console Application -> Name -> ok
1. To select tools menu -> connect to database
2. Select database and select dataset, click next, click new connection and click change button and select Microsoft Ms Access data source -> ok button
3. Click browse button and select Northwind and select open button
4. Click Test connection button and click ok then Select Next -> Yes button
5. Double-click Tables folder to view the list of tables available for the Northwind database
6. To display employee table in the window form

PROGRAM CODING:

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

namespace ConsoleApplication19
{
    class Program
    {
        static void Main(string[] args)
        {
            string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\S342\\Documents\\theciyasiva.accdb";
            OleDbConnection conn = new OleDbConnection(connectionString);
            string sql = "select name,address,salary from employee";
            OleDbCommand cmd = new OleDbCommand(sql, conn);

            Console.WriteLine("person name\taddress\t\tsalary");
            Console.WriteLine("=====");

            try
            {
                conn.Open();
                using (OleDbDataReader reader = cmd.ExecuteReader())
                {
                    Console.WriteLine(" ");
                    while (reader.Read())
                    {
                        Console.WriteLine("{0} {1} {2}", reader["name"].ToString() + "\t\t",
reader["address"].ToString() + "\t\t", reader["salary"].ToString());
                    }
                }
            }
        }
    }
}
```

```

    }
    catch(Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    Console.ReadKey();
    conn.Close();
}
}
}

```

Output:

```

file:///C:/Users/S342/documents/visual studio 2010/Projects/theciya/theciya/bin/Debug/theciya.EXE
=====
person name      address          salary
=====
theciya          papanasan        30000
bramma           thanjore         50000
tanil            krishanakiri     60000
nisha            padukottai       30000
snekk            kumbakonam       40000
gayu             thanjore         50000
atchu            kadampatti       20000
=====

```

RESULT:

Thustobuildc#. Net windowsapplicationandMsaccessdatabaseconnectionhasbeen verified.

AIM:

To create C#.net window form application to insert, update, delete and select operation in OleDbConnection object.

ALGORITHM:

2. Create new project -> Windows Application -> Name -> ok
3. Design your form with necessary labels and pictures
4. From toolbox, select "DataGridView" control and place it in form
5. Select database and select dataset, click next, click new connection and click change button and select Microsoft Ms Access data source -> ok button
6. Click Test connection button and click ok
7. Run the application
8. result will be displayed on the form.

PROGRAM 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 thecu
{
    public partial class Form1 : Form
    {
        int count = 0;
        OleDbConnection conn = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Users\\S342\\Desktop\\tamil.accdb");

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            conn.Open();
            OleDbCommand cmd = conn.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "insert into student values('"+textBox1.Text+"','"+textBox2.Text+"')";
            cmd.ExecuteNonQuery();
        }
    }
}
```

```

        conn.Close();
        MessageBox.Show("recordinsertedsuccessfully");
    }

privatevoidlabel1_Click(object sender,EventArgs)
{

}

privatevoidbutton4_Click(objectsender,EventArgs)
{
    conn.Open();
    OleDbCommandCmd=conn.CreateCommand();
    Cmd.CommandType = CommandType.Text;
    Cmd.CommandText = "select * from student";
    Cmd.ExecuteNonQuery();
    conn.Close();
    DataTabledt=newDataTable();
    OleDbDataAdapterda=newOleDbDataAdapter(Cmd);
    da.Fill(dt);
    dataGridView1.DataSource=dt;
    MessageBox.Show("recordviewedsuccessfully");
}

privatevoidbutton2_Click(objectsender,EventArgs)
{
    conn.Open();
    OleDbCommandCmd=conn.CreateCommand();
    Cmd.CommandType = CommandType.Text;
    Cmd.CommandText="deletefromstudentwherename='"+textBox1.Text+"'"; Cmd.ExecuteNonQuery();
    conn.Close();
    MessageBox.Show("recorddeletedsuccessfully");
}

privatevoidbutton3_Click(objectsender,EventArgs)
{
    conn.Open();
    OleDbCommandCmd=conn.CreateCommand();
    Cmd.CommandType = CommandType.Text;
    Cmd.CommandText ="updatestudent set name='"+textBox2.Text +"'wherename='"+
textBox1.Text + "'";
    Cmd.ExecuteNonQuery();
    conn.Close();
    MessageBox.Show("recordupdated successfully");
}

privatevoidbutton5_Click(objectsender,EventArgs)
{
    count = 0;
    conn.Open();
    OleDbCommandCmd=conn.CreateCommand();
    Cmd.CommandType = CommandType.Text;
    Cmd.CommandText="select*fromstudentwherename='"+textBox1.Text+"'";

```

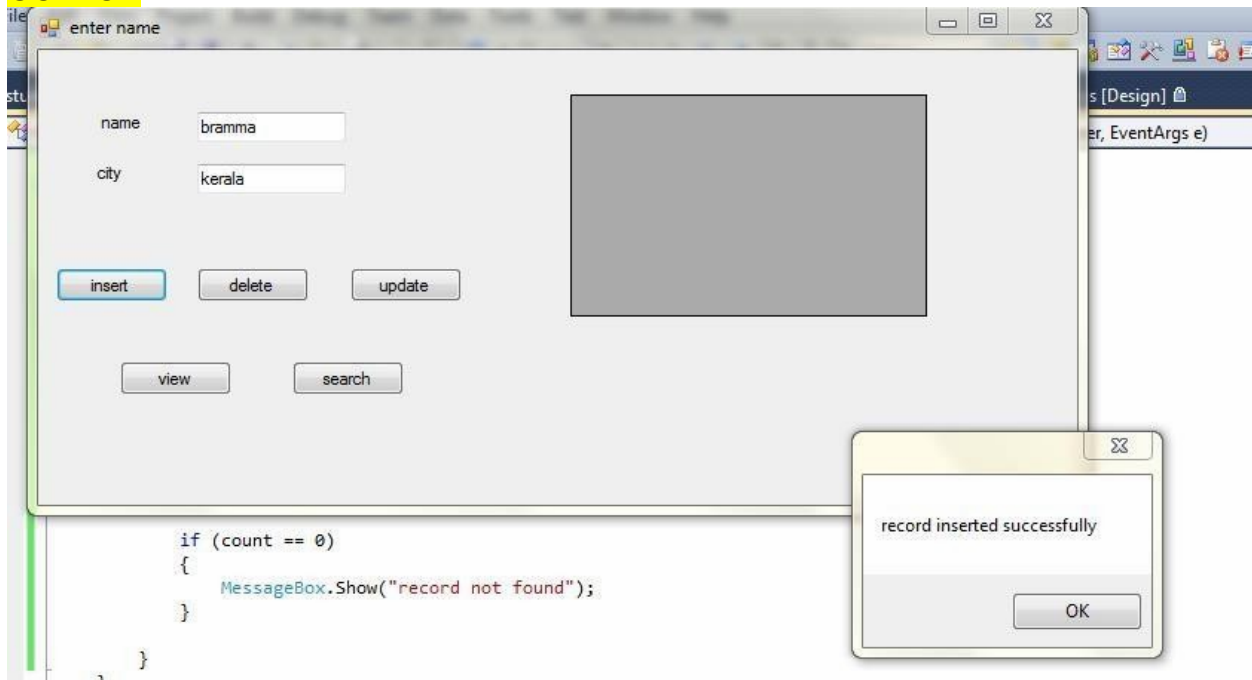
```

Cmd.ExecuteNonQuery();
DataTable dt=new DataTable();
OleDbDataAdapter da=new OleDbDataAdapter(Cmd);
da.Fill(dt);
count=Convert.ToInt32(dt.Rows.Count.ToString());
dataGridView1.DataSource = dt;
conn.Close();

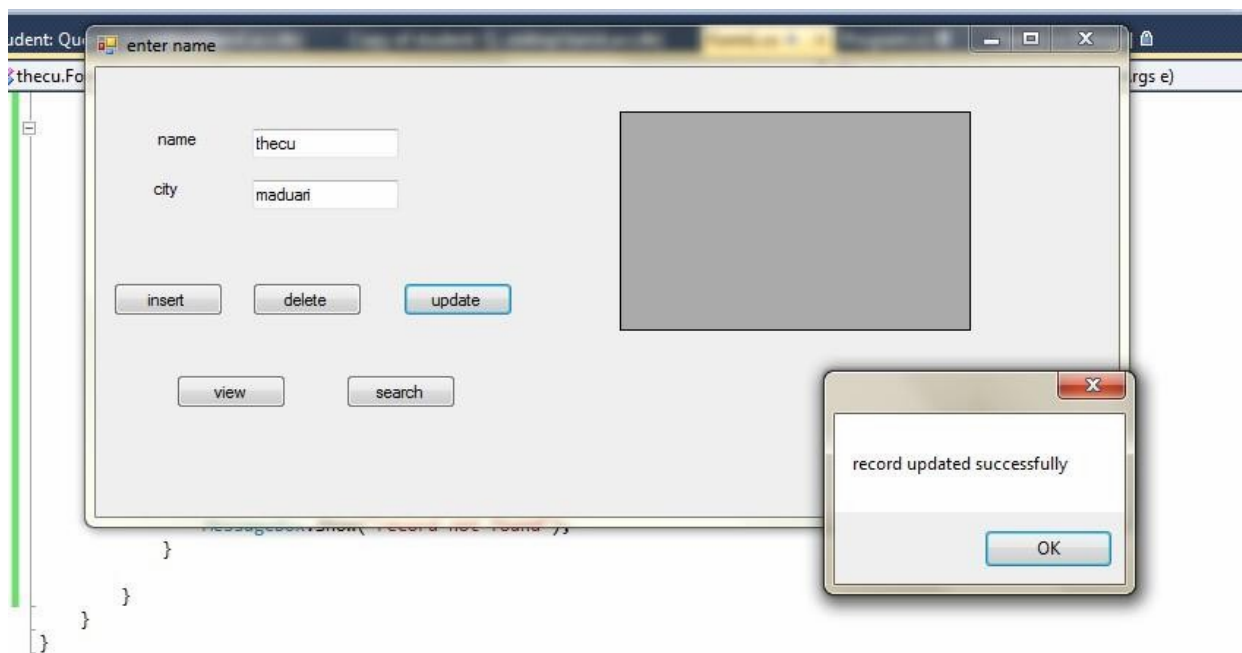
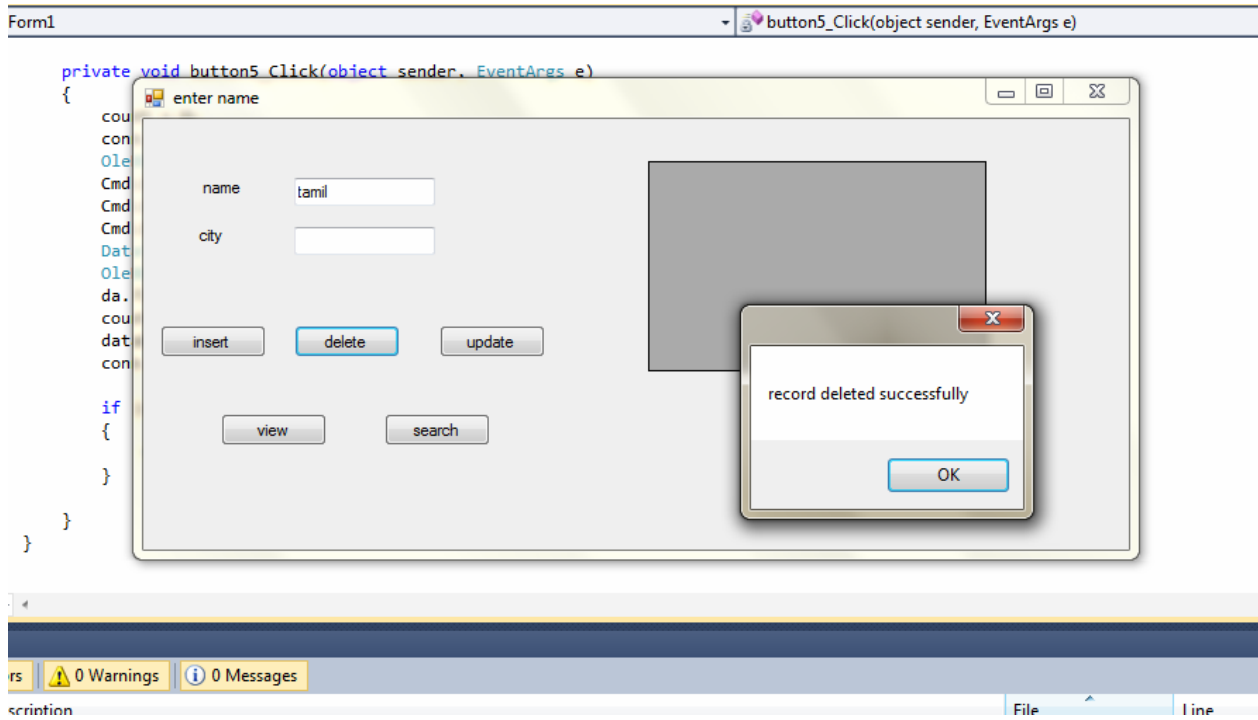
if(count==0)
{
    MessageBox.Show("recordnot found");
}
}
}
}

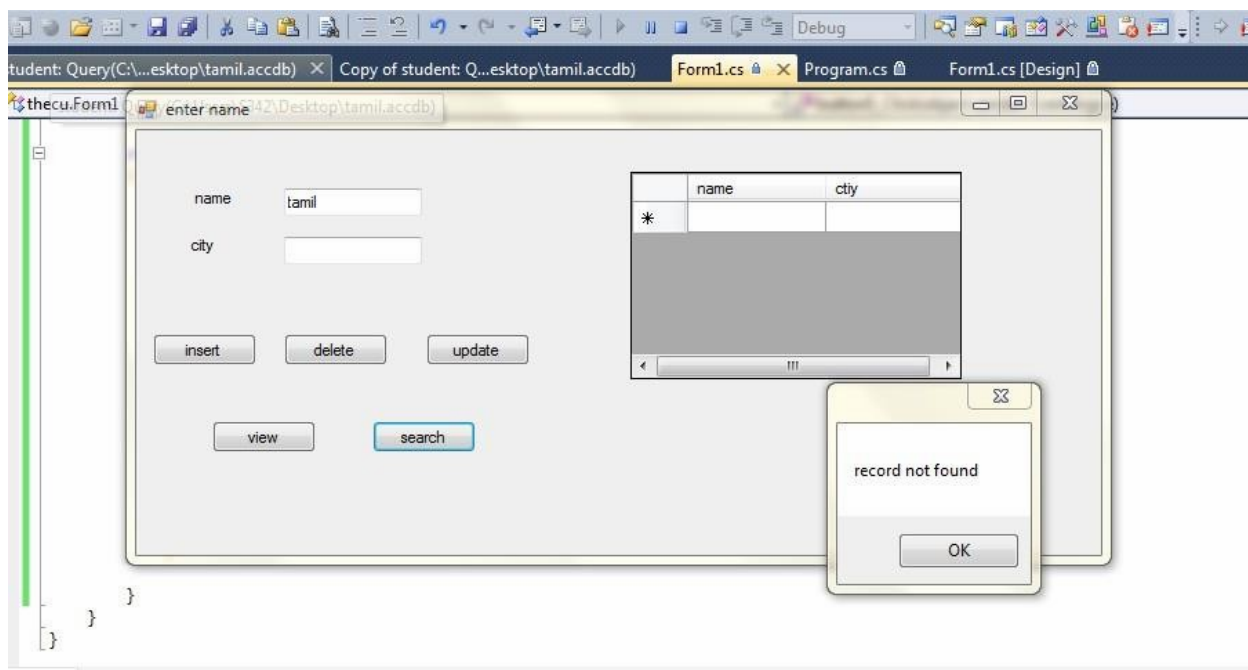
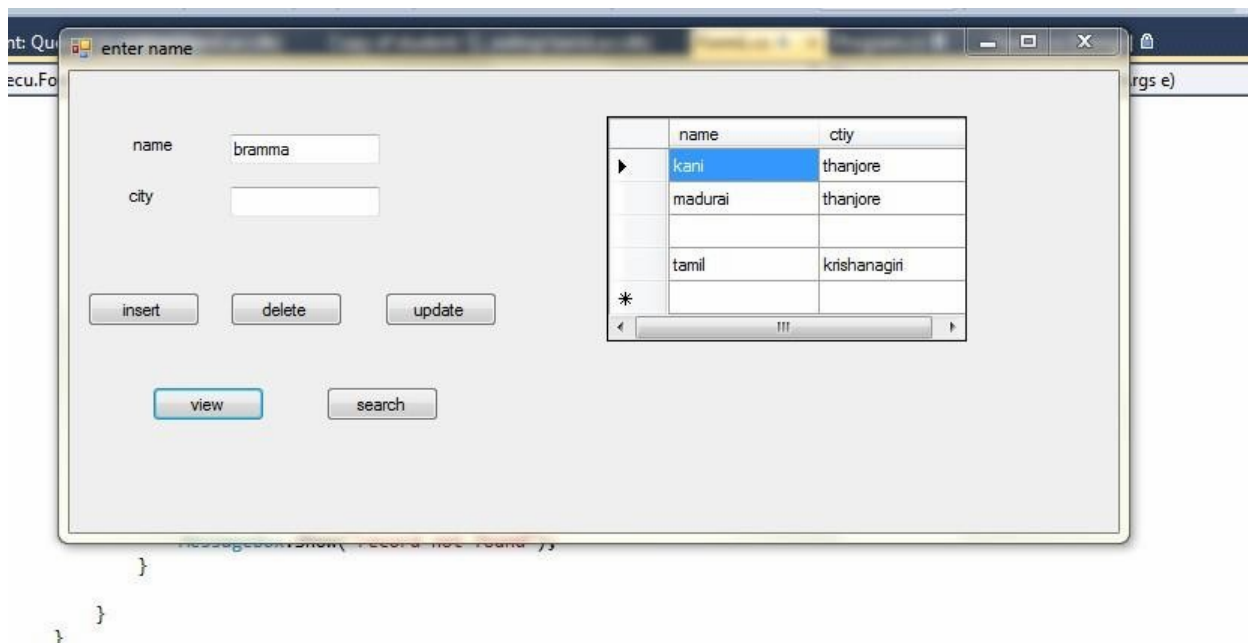
```

OUTPUT



student: Query(C:\...esktop\tamil.accdb) X Copy of stude		
	name	ctiy
▶	kani	thanjore
	thecu	than
*	NULL	NULL





RESULT:

Thusto createC#.net windowformapplication-toinsert,update,deleteandselectoperationin OleDbConnection object has been verified.

To create ASP.NET web application using server controls.

PROGRAMCODING:

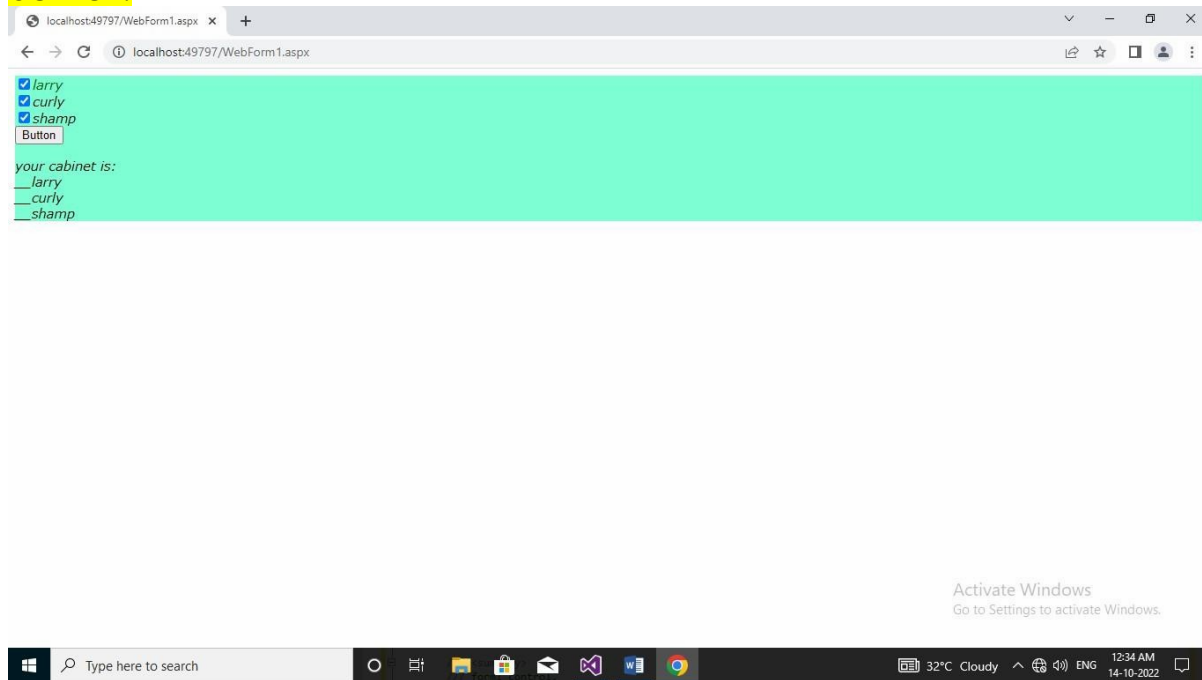
```
<formid="form1"runat="server">
```

</html>

C# CBox.aspx.cs

```
using System;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
  
public partial class Cbox : System.Web.UI.Page  
{  
    private string cabinet;  
    protected void Button1_Click(object sender, EventArgs e)  
    {  
        cabinet = "Your cabinet is:";  
        cabinet += CheckBox1.Checked == true ? " " + CheckBox1.Text + "<br/>": null;  
        cabinet += CheckBox2.Checked == true ? " " + CheckBox2.Text + "<br/>": null;  
        cabinet += CheckBox3.Checked == true ? " " + CheckBox3.Text + "<br/>": null;  
        cabinet += CheckBox4.Checked == true ? " " + CheckBox4.Text + "<br/>": null;  
        Label1.Text = cabinet;  
    }  
}
```

OUTPUT:



RESULT:

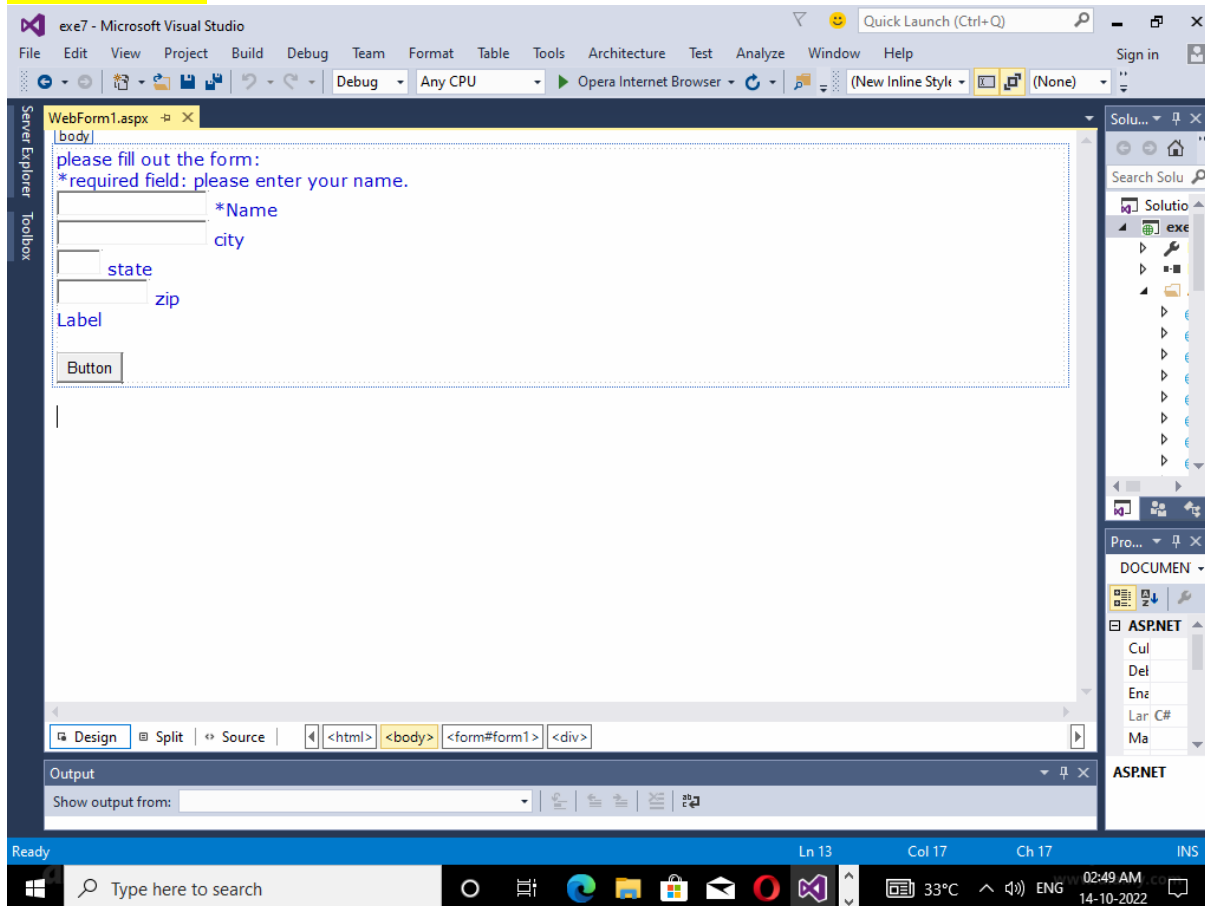
Thus to create an ASP.NET web application using web server control has been developed successfully.

Ex.No:8 Using Validation Controls

Aim:

To create ASP.NET web application using validation controls.

FORM DESIGN:



PROGRAM CODING:

WEBPAGE FORM DESIGN CODE:

```
<%@Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="exe7.WebForm1"%>
```

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<style type="text/css">div{
font-family: verdana;
font-size: 11pt;
color: #0000cc;
}
```

```
</style>
<title>requiredfieldvalidation</title>
</head>
```

```
<body>
<form id="form1" runat="server">
<div>
```

```
    please fill out the form:<br/>
```

```
    *required field:
```

```
<asp:requiredFieldValidator ID="requiredFieldValidator2" runat="server" ControlToValidate="Textbox1" Er
```

```

rorMessage="pleaseenteryourname."/><br/>
<asp:TextBoxID="textbox1"runat="server"/>&nbsp;*Name<br/>
<asp:textboxID="textbox2"runat="server"/>&nbsp;city<br/>
<asp:textboxID="textbox3"runat="server"Width="38px"/>&nbsp;state<br/>
<asp:textboxID="textbox4"runat="server"width="78px"/>&nbsp;zip<br/>

<asp:LabelID="Label1"runat="server"Text="Label"></asp:Label>

</div>
<p>
<asp:ButtonID="Button1"runat="server"OnClick="Button1_Click"Text="Button"/>
</p>
</form>
</body>
</html>

```

C#Code:

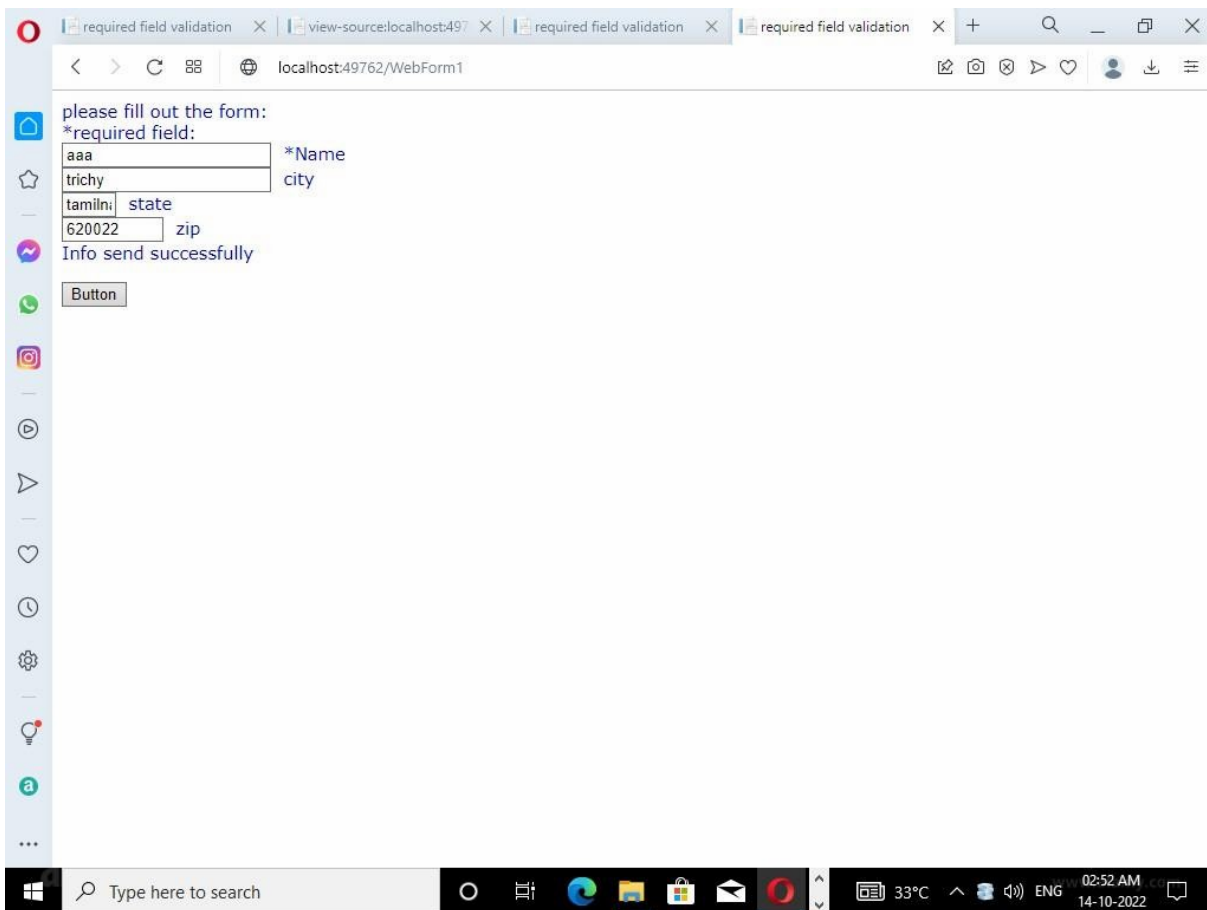
```

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

namespaceexe7
{
    publicpartialclassWebForm1:System.Web.UI.Page
    {
        protectedvoidButton1_Click(objectsender,EventArgs)
        {
            Label1.Text="Infosendsuccessfully";
        }
    }
}

```

OUTPUT:



RESULT:

ThustocreateASP.NETwebapplicationusingvalidationcontrolshasbeenverified successfully.

Aim:

To create a SQL Server Stored Procedure declaring parameters in ASP.NET Web application.

ALGORITHM:

Step 1: First open Microsoft SQL Server -> Enterprise Manager,

Step 2: then navigate to the database in which you want to create the stored procedure

Step 3: select New Stored Procedure. then select Stored Procedure Properties for what to enter, and then click OK.

Step: Now create an application named Store Procedure in .net to use the above procs. Step 5:

Display the output. Stop the Execution.

Declaring Parameters in SQL Server Stored Procedures:

1. The name
2. The datatype
3. The default value
4. The direction

The syntax

@parameter_name[AS]datatype[=default][NULL][VARYING][OUTPUT|OUT]

Let's now create a stored procedure named "Submitrecord".

PROGRAM CODING:**StoredProcedure.aspx page code**

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

```
<!DOCTYPEhtmlPUBLIC"-
//W3C//DTD XHTML 1.0 Transitional//EN"http://www.w3.org/TR/xhtml1/DTD/xhtml1transitional
.dtd">
```

```
<htmlxmlns="http://www.w3.org/1999/xhtml">
```

```
<headrunat="server">
```

```
<title>StoreProcedure</title>
```

```
</head>
```

```
<body>
```

```
<formid="form1"runat="server">
```

```
<div>
```

```

        <asp:LabelID="Label1"runat="server"Text="ID"></asp:Label>

        <asp:TextBoxID="TextBox1"runat="server"></asp:TextBox><br/><br/>

        <asp:LabelID="Label2"runat="server"Text="Password"></asp:Label>

        <asp:TextBoxID="TextBox2"runat="server"></asp:TextBox><br/>

<br/>

        <asp:LabelID="Label3"runat="server"Text="ConfirmPassword">

</asp:Label>

        <asp:TextBoxID="TextBox3"runat="server"></asp:TextBox><br/>

<br/>

        <asp:LabelID="Label4"runat="server"Text="EmailID">

</asp:Label>

        <asp:TextBoxID="TextBox4"runat="server"></asp:TextBox>

<br/><br/><br />

        <asp:ButtonID="Button1"runat="server"Text="SubmitRecord"OnClick="Button1_Click"/></div>

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

```

StoredProcedure.aspx.cs

```

using System;

using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

```



```

public partial class _Default : System.Web.UI.Page {

    DataSet ds = new DataSet();

    SqlConnection con;

    // Here we declare the parameter which we have to use in our application

    SqlCommand cmd = new SqlCommand();

    SqlParameter sp1 = new SqlParameter();

    SqlParameter sp2 = new SqlParameter();

    SqlParameter sp3 = new SqlParameter();

    SqlParameter sp4 = new SqlParameter();

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

    protected void Button1_Click(object sender, EventArgs e) {

        con = new SqlConnection("server=(local);database=gaaurav;uid=sa;pwd=");

        cmd.Parameters.Add("@ID", SqlDbType.VarChar).Value = TextBox1.Text;

        cmd.Parameters.Add("@Password", SqlDbType.VarChar).Value = TextBox2.Text; cmd.Parameters.Add("@ConfirmPassword", SqlDbType.VarChar).Value = TextBox3.Text;
        cmd.ExecuteNonQuery();

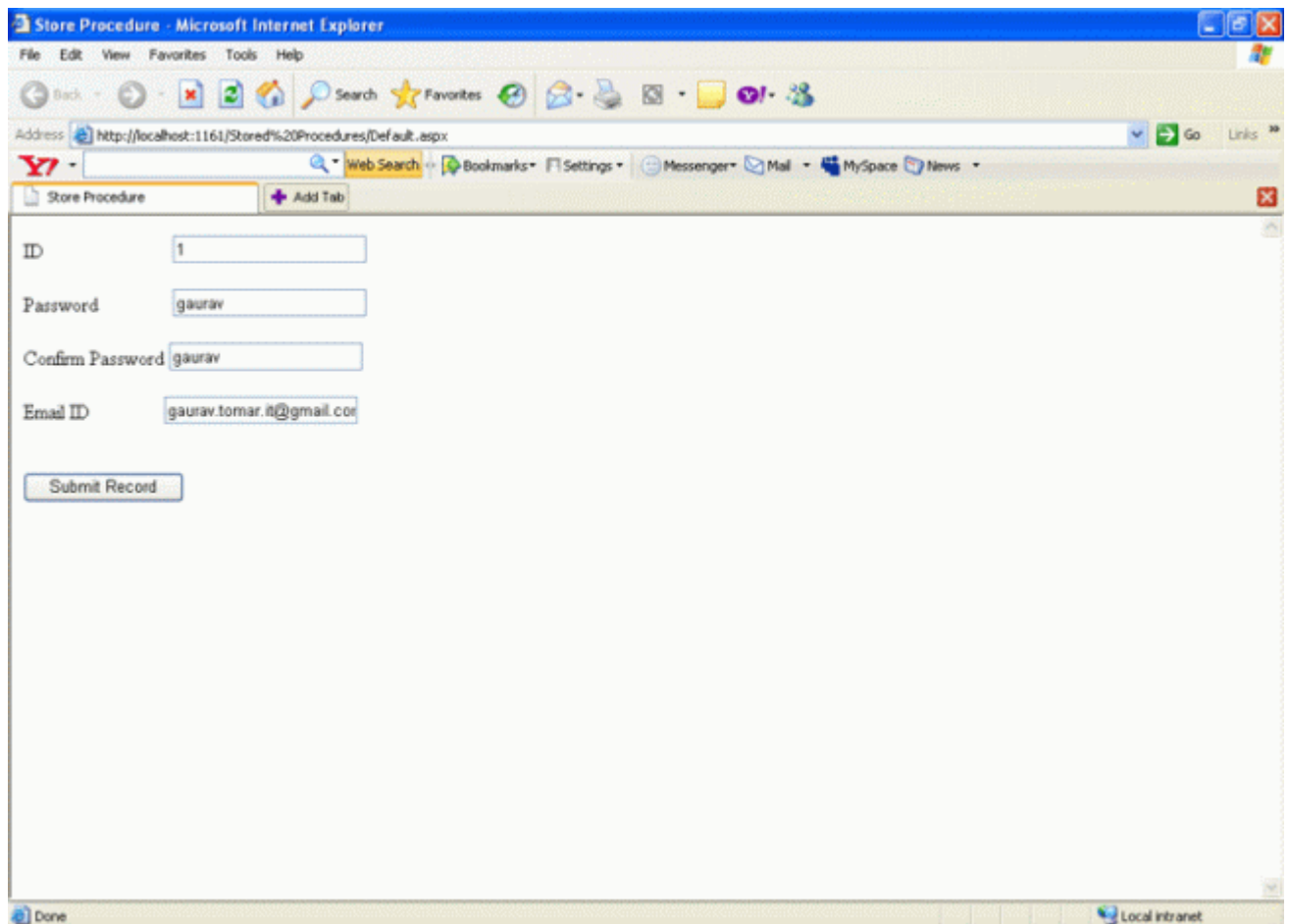
        // Here we declare the parameter which we have to use in our application

        cmd = new SqlCommand("Submitrecord", con);

    }
}

```

OUTPUT:



After clicking the submit button, the data is appended to the database as seen below in the SQL Server table record:

Result:

To create a SQL Server Stored Procedure declaring parameters in an ASP.NET Web application has been verified.

Aim:

To create program using RequiredFieldValidation control in ASP.NET Web application.

PROGRAMCODING:

```
<html>
<body>
<form runat="server">
<asp:label id="label" text="Enter the Boiling point of water:"
runat="server"
/> &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<asp:textbox id="text1" text="" runat="server" />
<asp:comparevalidator id="compBoilpt" controltovalidate="text1" Type="Integer"
ValueToCompare=100 Operator="Equal" display="static" errorMessage="Please
enter correct value" runat="server">
</asp:comparevalidator>
<asp:ValidationSummary id="sumErrors" runat="server"
showSummary = true
displayMode="BulletList" />
<br>
<asp:button id="bt1" runat="server" text="click" />
</form>
</body>
</html> Exercise
```

: 1

LabSolutions

Centre for Information Technology and Engineering, Manonmaniam Sundaranar University 21

<html>

<body>

RequiredFieldValidation

<formrunat=server>

Name:<asp:Textboxid="txtName" runat="server"></asp:Textbox>

```
<asp:buttonid="Button1"runat="server"text="Validate"/>
```

<p>

```
<asp:RequiredFieldValidatorid="RequiredFieldValidator1"runat="server">
```

ControlToValidate="txtName"

ErrorMessage="Nameisarequiredfield"

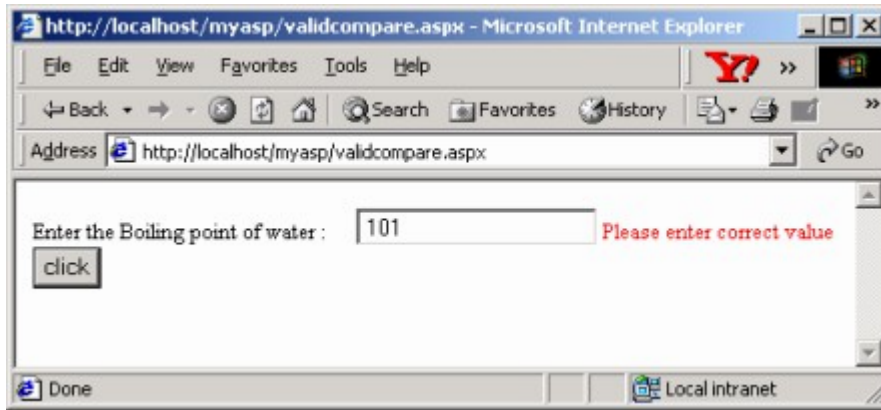
```
ForeColor="Red">
```

</asp:RequiredFieldValidator>

</body>

</html>

OUTPUT:



Result:

ThustocreateprogramusingReuiredFieldValidation controlinASP.NETWebapplicationhasbeen verified successfully.