THIAGARAJAR COLLEGE (AUTONOMOUS)

(Affiliated to Madurai Kamaraj University)

Re-Accredited with "A++ Grade" by NAAC NIRF 15th Rank in 2024

MADURAI-625009

DEPARTMENT OF COMPUTER SCIENCE

PCS24CL22 – ADVANCED WEB TECHNOLOGY LAB

NAME : _____

REGISTER NUMBER : _____

CLASS : _____

THIAGARAJAR COLLEGE (AUTONOMOUS)

(Affiliated to Madurai Kamaraj University)

Re-Accredited with "A++ Grade" by NAAC NIRF 15th Rank in 2024

Madurai - 625009

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that **PCS24CL22 – ADVANCED WEB TECHNOLOGY LAB** is a Bonafide record of work carried out by _____ Register No._____ is partial fulfilment of II Semester in M.Sc Computer Science during the year 2024 -2025.

Submitted for the University Practical Examination held on _____

Internal Examiner **External Examiner**

TABLE OF CONTENTS

EXERCISE NO : 1	SCIENTIFIC CALCULATOR
DATE :	

AIM :

To implement the Scientific Calculator using visual C# Windows Forms Application.

PROCEDURE :

Step 1:
Click File->New->Project->In visual c#->Click WindowsForm Application->Click OK.
Step 2:
Get a Design View.
Step 3
In Tool box double click on the Groupbox.
Step 4:
In Tool box double click on the Textbox.
Step 5:
Click Textbox Properties Change the Text align left to right.
Step 6:
Add 5*4=20 Buttons.
Step 7:
Rename the buttons name as 9,8,7,6,5,4,3,2,1,0.
Step 8:
Rename the buttons name as +,-,*,/,=.
Step 9:
Rename the buttons name as power,sin,cos,tan.
Step 10:
Click all the buttons and type the code.
Step 11:
Save the Program and Run.

SOURCE 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 Calculator
{
   public partial class Form1 : Form
   {
      double no1, no2, result;
      String op;
      public Form1()
      {
         InitializeComponent();
         var buttons = new[] { button1, button2, button3, button4, button5, button6, button7, button8,
button9, button10 };
         foreach (var button in buttons)
         {
            button.Click += (sender, e) =>
            {
               Button clickedButton = (Button)sender;
               textBox1.Text += clickedButton.Text;
            };
         }}
```

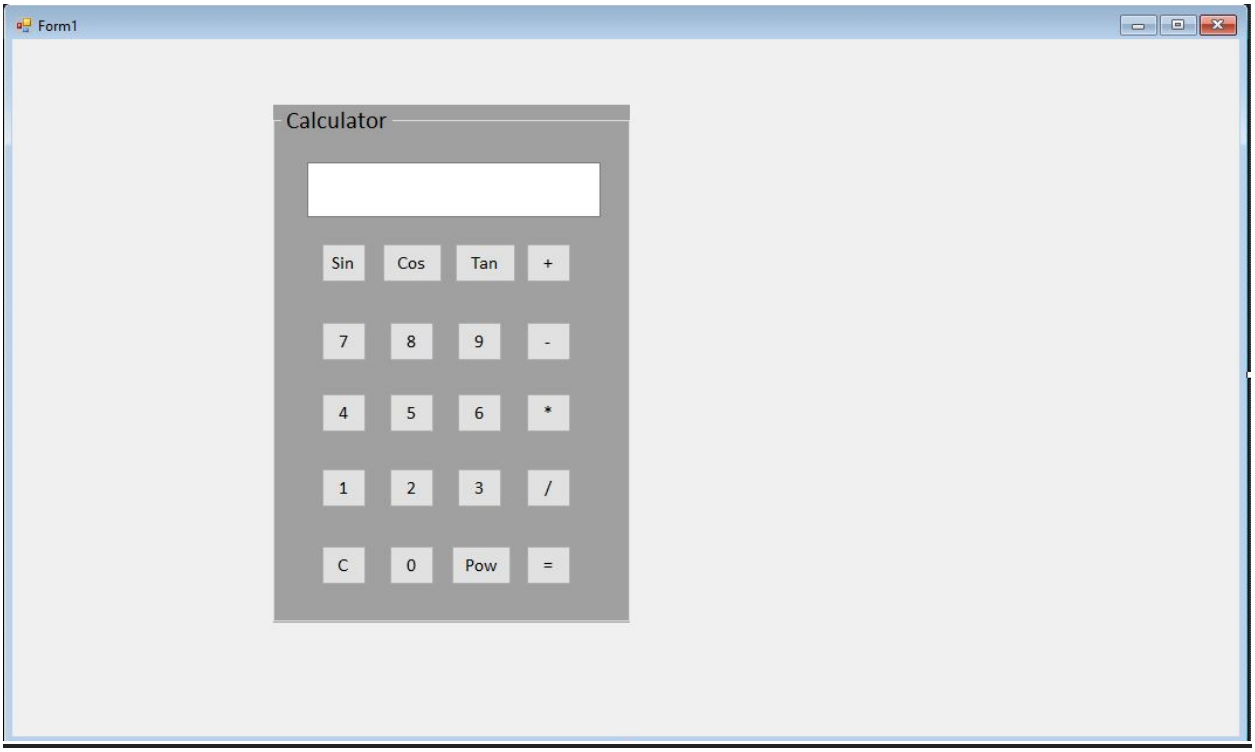
```
private void button1_Click(object sender, EventArgs e)
{

}

private void button11_Click(object sender, EventArgs e)
{
    no2 = double.Parse(textBox1.Text);
    switch(op)
    {
        case "+":
            result = no1 + no2;
            break;
        case "-":
            result = no1 - no2;
            break;
        case "*":
            result = no1 * no2;
            break;
        case "/":
            if (no2 != 0)
                result = no1 / no2;
            else
                MessageBox.Show("Divide by Zero");
            break;
        case "pow":
            result = Math.Pow(no1, no2);
            break;
    }
```

```
    textBox1.Text = result.ToString();
}

private void Form1_Load(object sender, EventArgs e)
{

}

private void button12_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = "";
    op = "+";
}

private void button13_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = "";
    op = "-";
}

private void button14_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = "";
    op = "*";
}

private void button15_Click(object sender, EventArgs e)
{
```

```
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = "";
    op = "/";
}

private void button16_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    no1 = 0;
    no2 = 0;
    op = "";
}

private void button17_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = "";
    op = "pow";
}

private void button18_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = Convert.ToString(Math.Sin(no1));
}
private void button19_Click(object sender, EventArgs e)
{
    no1 = double.Parse(textBox1.Text);
    textBox1.Text = Convert.ToString(Math.Cos(no1));
}
```

```
        private void button20_Click(object sender, EventArgs e)
        {
            no1 = double.Parse(textBox1.Text);
            textBox1.Text = Convert.ToString(Math.Tan(no1));
        }
    }
}
```

OUTPUT :



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 2	QUIZ
DATE :	

AIM :

To implement the Quiz using visual C# Windows Forms Application.

PROCEDURE :

Step 1:

Create a new project using the Windows Forms App (.NET Framework) template.

Step 2:

Form1 :

Add Button and label in the form using the toolbox
Double Click the Form1 Design page and write the form1 code.

Step 3:

Form2 :

Add textbox (change the property of the textbox “multiline=True”)
Add 4 RadioButton for answer choices.
Add label for showing the number of remaining questions.
Add Timer and label for displaying the time remaining.
Add Button to submit the answer.

Double Click the Form2 Design Page and write the form2 code.

Step 4:

Form3 :

Drag a Label from the Toolbox and place it on the form is to provide the score.
Double Click the Form3 Design Page and write the form3 code.

Step 5:

Save and Run the project.

SOURCE CODE :

Form 1:

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

      private void Form1_Load(object sender, EventArgs e)
      {
         this.BackColor = Color.CadetBlue;
         label2.Text = "Total Questions: 5";
      }

      private void button1_Click(object sender, EventArgs e)
      {
         Form2 quizForm = new Form2();
         quizForm.Show();
         this.Hide();
      }
   }
}
```

Form2 :

```
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 WindowsFormsApplication1
{
   public partial class Form2 : Form
   {
      private List<Question> questions = new List<Question>()
      {
         new Question("Which namespace is essential for creating Windows Forms applications?",
"System.Windows.Forms", new string[] { "System.Data", "System.Windows.Forms", "System.IO",
"System.Collections.Generic" }),

         new Question("Which control is used to display static text or links in a
Windows Forms application?", "Label", new string[] { "Button", "Label", "TextBox", "ListView" }),

         new Question("Which event is triggered when a button is clicked in a
Windows Forms application?", "Click", new string[] { "MouseClick", "Click", "OnPress", "KeyDown" }),

         new Question(" How do you stop the execution of a Timer in a Windows Forms application?",
"timer.Stop()", new string[] { "timer.Stop()", "timer.End()", "timer.Disable()", "timer.Pause()" }),

         new Question("What is the purpose of the Dispose() method in a Windows Forms application?",
"To release resources used by the application.", new string[] { "To release resources used by the
application.", "To start a new form.", "To reset a form's controls.", "To end the program execution." })
      };
      private int currentQuestionIndex = 0;
      private int score = 0;
      private int timeLimit = 15; // 30 seconds per question
      private Timer questionTimer;
      public Form2()
      {
         InitializeComponent();
         questionTimer = new Timer();
         questionTimer.Interval = 1000; // 1 second interval
         questionTimer.Tick += timer1_Tick;
         DisplayQuestion();
      }
      private void DisplayQuestion()
      {
         Question currentQuestion = questions[currentQuestionIndex];
         this.textBox1.Text = currentQuestion.Text;
         // Clear previous selections
         radioButton1.Checked = false;
         radioButton2.Checked = false;
         radioButton3.Checked = false;
         radioButton4.Checked = false;
         // Populate radio buttons with options
         radioButton1.Text = currentQuestion.Options[0];
         radioButton2.Text = currentQuestion.Options[1];
         radioButton3.Text = currentQuestion.Options[2];
         radioButton4.Text = currentQuestion.Options[3];
```

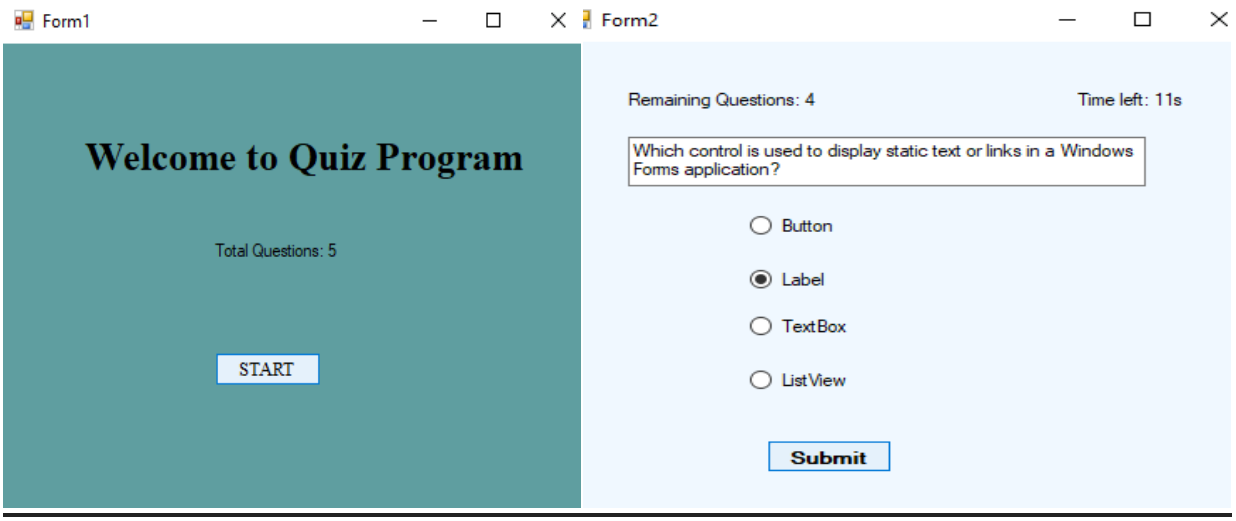
```
    // Start the timer for the current question

    timeLimit = 15; // Reset timer to 15 seconds
    label1.Text = string.Format("Time left: {0}s", timeLimit); // Update label
    questionTimer.Start();
    label2.Text = string.Format("Remaining Questions: {0}", questions.Count - currentQuestionIndex);
}


private void Form2_Load(object sender, EventArgs e)
{
    this.BackColor = Color.AliceBlue;
}

private void button1_Click(object sender, EventArgs e)
{
    string userAnswer = "";
    if (radioButton1.Checked)
    {
        userAnswer = radioButton1.Text;
    }
    else if (radioButton2.Checked)
    {
        userAnswer = radioButton2.Text;
    }
    else if (radioButton3.Checked)
    {
        userAnswer = radioButton3.Text;
    }
    else if (radioButton4.Checked)
    {
        userAnswer = radioButton4.Text;
    }

    if (string.IsNullOrEmpty(userAnswer))
    {
        MessageBox.Show("Please select an answer before submitting.");
        return;
    }
    questionTimer.Stop();
    HandleAnswer(userAnswer);
}
private void HandleAnswer(string userAnswer)
{
    Question currentQuestion = questions[currentQuestionIndex];
    if (userAnswer == currentQuestion.Answer)
    {
        score++;
        MessageBox.Show("Correct!");
```

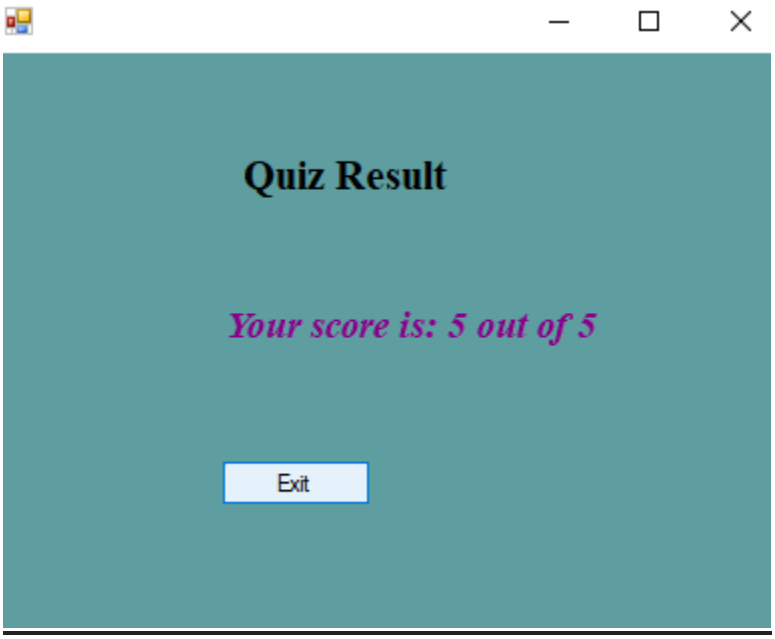
```
        }
        else
        {
            MessageBox.Show("Incorrect");
            MessageBox.Show("THE CORRECT ANSWER IS:  " + currentQuestion.Answer);
        }

        currentQuestionIndex++;
        if (currentQuestionIndex >= questions.Count)
        {
            EndQuiz();
        }
        else
        {
            DisplayQuestion();
        }
    }
    private void EndQuiz()
    {
        this.Hide(); // Hide the current form
        Form3 scoreForm = new Form3(score, questions.Count); // Pass score and total questions
        scoreForm.Show(); // Show the score form
    }

    public class Question
    {
        public string Text { get; set; }
        public string Answer { get; set; }
        public string[] Options { get; set; }
        public Question(string text, string answer, string[] options)
        {
            Text = text;
            Answer = answer;
            Options = options;
        }
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        timeLimit--;
        label1.Text = string.Format("Time left: {0}s", timeLimit);

        if (timeLimit <= 0)
        {
            questionTimer.Stop();
            HandleAnswer(""); // Time's up, proceed with no answer
        }
    }
}}
```

Form 3 :

```
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 WindowsFormsApplication1
{
   public partial class Form3 : Form
   {
      public Form3(int score, int totalQuestions)
      {
         InitializeComponent();
         this.BackColor = Color.CadetBlue;
         label2.Text = " Your score is: " + score + " out of " + totalQuestions;
      }

      private void button2_Click_1(object sender, EventArgs e)
      {
         this.Close();
      }
   }
}
```

OUTPUT :





RESULT :

Thus the program was executed successfully.

EXERCISE NO : 3	TEXT EDITOR
DATE :	

AIM :

To implement the TextEditor using visual C# Windows Forms Application.

PROCEDURE :

Step 1:
Click File->New->Project->In visuaL c#-> Clickk WindowsForm Application->Click OK.

Step 2:
Get a Design View.

Step 3:
In Tool box we Double click on the Rich Text box.

Step 3:
In Tool box we Double click on the Menu Strip.

Step 4:
Rename the menustrips as File,Edit,Format.

Step 5:
Now add the menuitems New,Save,Open,Exit in the File menustrip.

Step 6:
Now add the menuitems Cut,Copy,Paste,Undo,Redo,Find,Replace in the Edit menustrip.

Step 7:
Now add the menuitems Font,Color in the Format menustrip.

Step 8:
Double click all the menuitems to get in the code.

Step 9:
Solution Explorer->References->Click add reference->Click Assemblies->Add Microsoft.Visual.Basic->Click OK.

Step 10:
Save the Program and Run.

SOURCE 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 Microsoft.VisualBasic;

namespace WindowsFormsApplication2
{
   public partial class Form1 : Form
   {
      private ContextMenuStrip contextMenu;
      public Form1()
      {
         InitializeComponent();
         InitializeContextMenu();
      }
      private void InitializeContextMenu()
      {
          // Create the context menu
          contextMenu = new ContextMenuStrip();

          // Add menu items
        ToolStripMenuItem cutMenuItem = new ToolStripMenuItem("Cut", null,
cutToolStripMenuItem_Click);
```

```
        ToolStripMenuItem copyMenuItem = new ToolStripMenuItem("Copy", null,
copyToolStripMenuItem_Click);

        ToolStripMenuItem pasteMenuItem = new ToolStripMenuItem("Paste", null,
pasteToolStripMenuItem_Click);

        ToolStripMenuItem undoMenuItem = new ToolStripMenuItem("Undo", null,
undoToolStripMenuItem_Click);

        ToolStripMenuItem redoMenuItem = new ToolStripMenuItem("Redo", null,
redoToolStripMenuItem_Click);

        ToolStripMenuItem findMenuItem = new ToolStripMenuItem("Find", null,
findToolStripMenuItem_Click);

        ToolStripMenuItem replaceMenuItem = new ToolStripMenuItem("Replace", null,
replaceToolStripMenuItem_Click);

        // Add items to the context menu

        contextMenu.Items.AddRange(new ToolStripItem[] {

            cutMenuItem,

            copyMenuItem,

            pasteMenuItem,

            new ToolStripSeparator(),

            undoMenuItem,

            redoMenuItem,

            new ToolStripSeparator(),

            findMenuItem,

            replaceMenuItem

        });


        // Associate the context menu with the RichTextBox

        richTextBox1.ContextMenuStrip = contextMenu;

    }


    private void richTextBox1_TextChanged(object sender, EventArgs e)

    {


    }
```

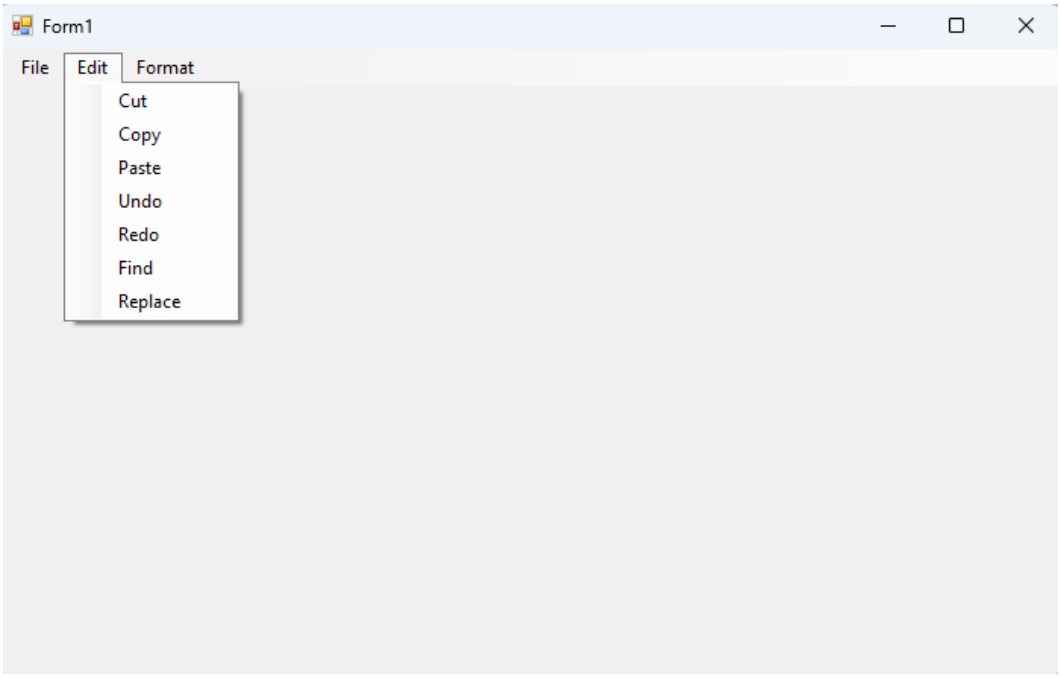
```
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
   richTextBox1.Clear();
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
   OpenFileDialog openFileDialog = new OpenFileDialog();
   openFileDialog.Filter = "Text Files (*.txt)|*.txt";
   if (openFileDialog.ShowDialog() == DialogResult.OK)
   {
      richTextBox1.Text = System.IO.File.ReadAllText(openFileDialog.FileName);
   }

}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
   SaveFileDialog saveFileDialog = new SaveFileDialog();
   saveFileDialog.Filter = "Text Files (*.txt)|*.txt";
   if (saveFileDialog.ShowDialog() == DialogResult.OK)

   {
      System.IO.File.WriteAllText(saveFileDialog.FileName, richTextBox1.Text);
   }
}
```

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

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

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

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

private void redoToolStripMenuItem_Click(object sender, EventArgs e)
{
   richTextBox1.Redo();
}


private void fontToolStripMenuItem_Click(object sender, EventArgs e)
{
   FontDialog fontDialog = new FontDialog();
   if (fontDialog.ShowDialog() == DialogResult.OK)
```

```
    {
        richTextBox1.Font = fontDialog.Font;
    }
}

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

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void Form1_Load(object sender, EventArgs e)
{
    richTextBox1.Font = new Font("Calibri", 18);
    richTextBox1.ForeColor = Color.Black;
}

private void findToolStripMenuItem_Click(object sender, EventArgs e)

{
    string searchText=Interaction.InputBox("Enter the Text");
    int startIndex = 0;
    // Clear previous highlights
```

```
        richTextBox1.SelectAll();
        richTextBox1.SelectionBackColor = Color.White;
        // Highlight search text
        while (startIndex < richTextBox1.TextLength)
        {
            int wordStartIndex = richTextBox1.Find(searchText, startIndex, RichTextBoxFinds.None);
            if (wordStartIndex != -1)
            {
                richTextBox1.SelectionStart = wordStartIndex;
                richTextBox1.SelectionLength = searchText.Length;
                richTextBox1.SelectionBackColor = Color.Yellow;
                startIndex = wordStartIndex + searchText.Length;
            }
            else
            {
                break;
            }
        }
    }
    private void replaceToolStripMenuItem_Click(object sender, EventArgs e)
    {
        string searchText = Interaction.InputBox("Enter the Text");
        string replaceText = Interaction.InputBox("Enter the Text");
        richTextBox1.Text = richTextBox1.Text.Replace(searchText, replaceText);
    }
} }
```

OUTPUT :



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 4	IMAGE MANIPULATIONS
DATE :	

AIM :

To implement the Image Manipulation using visual C# Windows Form Application.

PROCEDURE :

Step 1:
Click File->New->Project->In visuak c#-> Clcik WindowsForm Application->Click OK.

Step 2:
We get a Design View.

Step 3:
In Tool box we Double click on the Picture box.

Step 4:
In Tool box we Double click on the add Three Label.

Step 5:
In Tool box we Double click on the add Four Buttons.

Step 6:
In Tool box we Double click on the add Two Track Bar.

Step 7:
Rename the buttons as Load,Scale,Grayscale,Rotate.

Step 8:
Rename the labels as Image Manipulations,Brightness,Zoom.

Step 9:
Double Click all the buttons to get in the code.

Step 10:
We add the using System.Drawing.Imaging in the header packages.

Step 11:
Save the Program and Run.

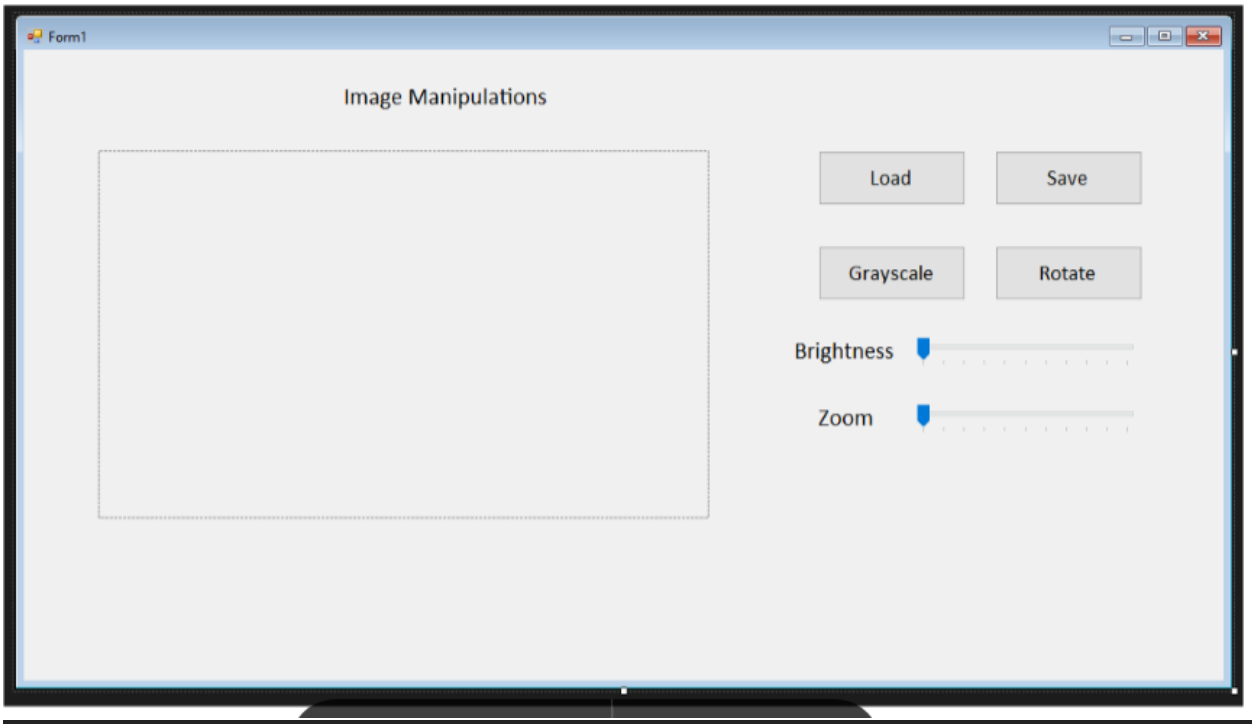
SOURCE CODE :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApplication7
{
public partial class Form1 : Form
{
private Image originalImage;
private Image zoomedImage;
public Form1()
{
InitializeComponent();
trackBar1.Minimum = 0;
trackBar1.Maximum = 200; // 100 as the neutral point
trackBar1.Value = 100;
trackBar1.Scroll += new EventHandler(trackBar1_Scroll);
trackBar2.Minimum = 1;
trackBar2.Maximum = 10; // Adjust the maximum zoom level as needed
trackBar2.Value = 1;
trackBar2.Scroll += new EventHandler(trackBar2_Scroll);
}
private void button1_Click(object sender, EventArgs e)
{
OpenFileDialog openFileDialog = new OpenFileDialog();
openFileDialog.Filter = "Image Files|*.jpg;*.jpeg;*.png;*.bmp;*.gif";
if (openFileDialog.ShowDialog() == DialogResult.OK)
{
originalImage = Image.FromFile(openFileDialog.FileName);

zoomedImage = originalImage;
pictureBox1.Image = originalImage;
}
}
private void button2_Click(object sender, EventArgs e)
{
if (pictureBox1.Image != null)
{
Bitmap bmp = new Bitmap(pictureBox1.Image);
for (int y = 0; y < bmp.Height; y++)
{
for (int x = 0; x < bmp.Width; x++)
```

```
{
Color c = bmp.GetPixel(x, y);
int gray = (c.R + c.G + c.B) / 3;
bmp.SetPixel(x, y, Color.FromArgb(gray, gray, gray));
}
}
pictureBox1.Image = bmp;
zoomedImage = bmp;
}
}
private void button3_Click(object sender, EventArgs e)
{
if (pictureBox1.Image != null)
{
pictureBox1.Image.RotateFlip(RotateFlipType.Rotate90FlipNone);
pictureBox1.Refresh();
zoomedImage = pictureBox1.Image;
}
}
private void button4_Click(object sender, EventArgs e)
{
SaveFileDialog saveFileDialog = new SaveFileDialog();
saveFileDialog.Filter = “PNG Image|*.png|JPEG Image|*.jpg|BMP
Image|*.bmp”;

if (saveFileDialog.ShowDialog() == DialogResult.OK)
{
pictureBox1.Image.Save(saveFileDialog.FileName);
}
}
private void AdjustBrightness(int value)
{
if (originalImage != null)
{
Bitmap adjustedImage = new Bitmap(originalImage.Width, originalImage.Height);
float brightnessFactor = value / 100f; // Convert trackbar value to a scaling factor
// Create a color matrix for brightness adjustment
float bFactor = brightnessFactor - 1.0f; // Photoshop-like adjustment
float[][] matrixItems =
{


new float[] {1, 0, 0, 0, 0},
new float[] {0, 1, 0, 0, 0},
new float[] {0, 0, 1, 0, 0},
new float[] {0, 0, 0, 1, 0},
new float[] {bFactor, bFactor, bFactor, 0, 1}
};
ColorMatrix colorMatrix = new ColorMatrix(matrixItems);
ImageAttributes imgAttributes = new ImageAttributes();
```

```
imgAttributes.SetColorMatrix(colorMatrix);
using (Graphics g = Graphics.FromImage(adjustedImage))
{
g.DrawImage(originalImage, new Rectangle(0, 0,
originalImage.Width, originalImage.Height),
0, 0, originalImage.Width, originalImage.Height,
GraphicsUnit.Pixel, imgAttributes);
}
pictureBox1.Image = adjustedImage;
zoomedImage = adjustedImage;
}

}
private void trackBar1_Scroll(object sender, EventArgs e)
{
AdjustBrightness(trackBar1.Value);
}
private void trackBar2_Scroll(object sender, EventArgs e)
{
AdjustZoom(trackBar2.Value);
}
private void AdjustZoom(int zoomLevel)
{
if (originalImage != null)
{
Bitmap zoomedBmp = new Bitmap(originalImage, new
Size(originalImage.Width * zoomLevel, originalImage.Height * zoomLevel));
zoomedImage = zoomedBmp;
pictureBox1.Image = zoomedImage;
}
}
}
}
```

OUTPUT:



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 5	LOGIN FORM
DATE :	

AIM :

To implement the Login Form using MS ACCESS with visual C# Windows Form Application.

PROCEDURE :

Step 1:
Open Microsoft Access and Create a New Database.
Step 2:
Create a New Table add username, password, other columns and Insert Values into it.
Step 3:
Close Microsoft Access.
Step 4:
Open Visual Studio. Go to File -> New -> Project.
Step 5:
Select Visual C# -> Windows Forms Application. Click Ok.
Step 6:
Design the Form by Adding Labelbox, Textbox, Button into the form.
Step 7:
Click on the Project Name. Add-> Windows Form.
Step 8:
Open Data Source Panel or Shift + Alt + D.
Step 9:
Click Add New Data Source Button.
Step 10:
Select Database and Click Next.
Step 11:
Select Dataset and Click Next.
Step 12:
Click New Connection. In the Add Connection Window, click Browse in the Database file name, Locate the database file that you have created in Step 2.
Step 13:
Click Test Connection. Then, click Advanced and copy the Provider String from the Textbox and paste it in notepad for coding purpose.
Step 14:
Cancel and close all the Data Source Windows.
Step 15:
Double click the button1 on Form1 to implement the code logic and Provider String.
Step 16:
Double click on the Form2 and Implement Code logic in the Form Load, Button.
Step 17: Save and Run the Program.

SOURCE CODE :

LoginForm.cs :

```
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.OleDb;
namespace WindowsFormsApplication6
{
public partial class LoginForm : Form
{
private OleDbConnection conn;
public LoginForm()
{
InitializeComponent();
conn =new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=D:/urmila/vb c#/db/StudentDB.accdb");
}
private void button1_Click(object sender, EventArgs e)
{
string username = textBox1.Text;
string password = textBox2.Text;
string query = "select * from Students where Username=@username and
Password=@password";
```

```
OleDbCommand cmd = new OleDbCommand(query, conn);
cmd.Parameters.AddWithValue("@Username", username);
cmd.Parameters.AddWithValue("@Password";, password);
try
{
conn.Open();
OleDbDataReader reader = cmd.ExecuteReader();
if(reader.Read())
{

StudentDetailsForm detailsForm = new StudentDetailsForm();
detailsForm.ID = reader["id"].ToString();
detailsForm.Sname = reader["Sname"].ToString();
detailsForm.Age = reader["Age"].ToString();
detailsForm.Course = reader["Course"].ToString();
detailsForm.ShowDialog();
}
else
{
MessageBox.Show("Invalid Username or Password", "error");
}
}catch(OleDbException ex)
{
MessageBox.Show(ex.Message,"Error");
}
finally
{
conn.Close();
}
}
```

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

{
MessageBox.Show("Welcome to the Login Form!");
}
}
}
```

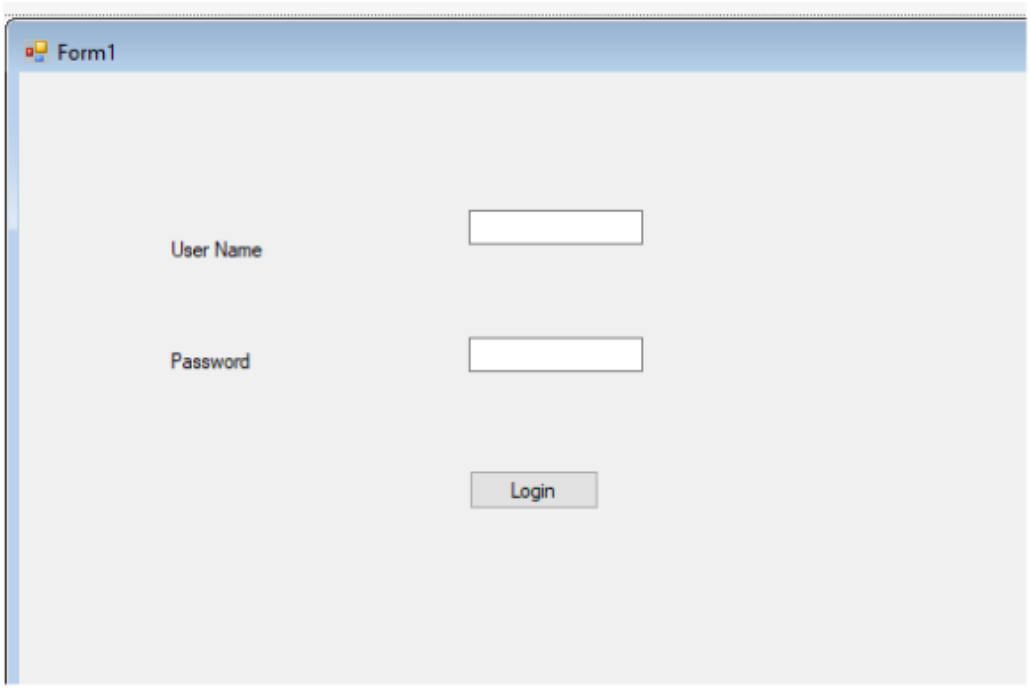
StudentDetailsForm.cs :

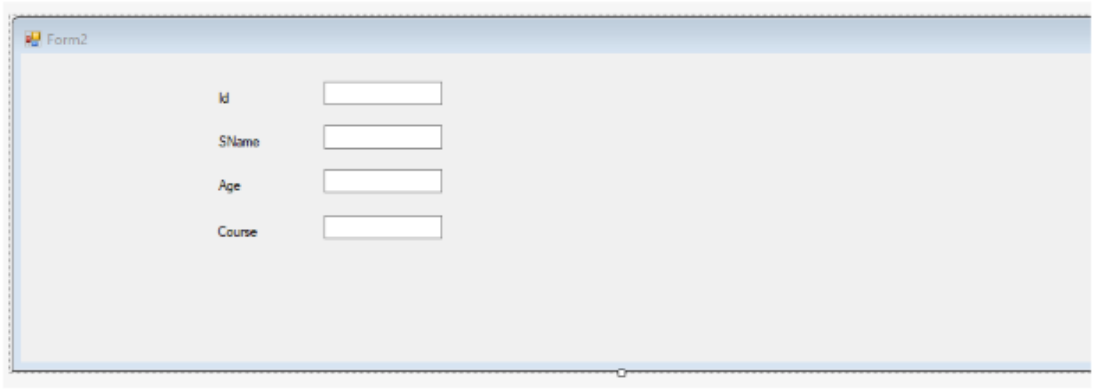
```
using System;
using System.Collections.Generic;

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApplication6
{
public partial class StudentDetailsForm : Form
{
public string ID { get; set; }
public string Sname { get; set; }
public string Age { get; set; }
public string Course { get; set; }
public StudentDetailsForm()
{
InitializeComponent();
}
```

```
private void StudentDetailsForm_Load(object sender, EventArgs e)
{
textBox1.Text = ID;
textBox2.Text = Sname;
textBox3.Text = Age;
textBox4.Text = Course;
}
}
}
```

OUTPUT :

Student Information





RESULT :

Thus the program was executed successfully.

EXERCISE NO : 6	DATA MANIPULATIONS USING MS-ACCESS
DATE :	

AIM :

To implement the Data Manipulations using MS-ACCESS with visual C# Windows Form Application.

PROCEDURE:

Step 1: Open Microsoft Access and Create a New Database.

Step 2: Create a New Table and Insert Values into it.

Step 3: Close Microsoft Access.

Step 4: Open Visual Studio. Go to File -> New -> Project.

Step 5: Select Visual C# -> Windows Forms Application. Click Ok.

Step 6: Design the Form by Adding Labelbox, Textbox, Button, DataGridView into the form.

Step 7: Open Data Source Panel or Shift + Alt + D.

Step 8: Click Add New Data Source Button.

Step 9: Select Database and Click Next.

Step 10: Select Dataset and Click Next.

Step 11: Click New Connection. In the Add Connection Window, click Browse in the Database file name, Locate the database file that you have created in Step 2.

Step 12: Click Test Connection. Then, click Advanced and copy the Provider String from the Textbox and paste it in notepad for coding purpose.

Step 13: Cancel and close all the Data Source Windows.

Step 14: Implement Code logic in the Form Load, Button, DataGridView, TextBox.

Step 15: Save and Run the Program.

SOURCE CODE :

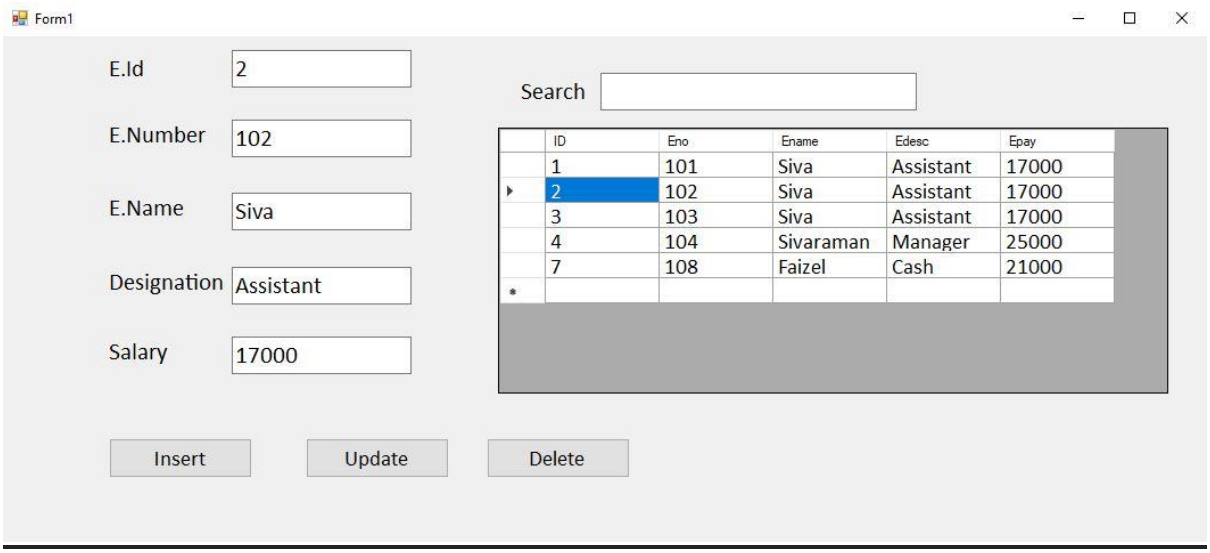
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace WindowsFormsApplication5
{
   public partial class Form1 : Form
   {
      OleDbConnection conn;
      OleDbCommand cmd;
      OleDbDataAdapter adapter;
      DataTable dt;
      public Form1()
      {
         InitializeComponent();


      }



      void getemp()
      {
         conn=new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data
Source=D:/1.MSc/Database2.accdb");
```

```
        dt = new DataTable();
        adapter=new OleDbDataAdapter("SELECT * FROM employee",conn);
        conn.Open();
        adapter.Fill(dt);
        dataGridView1.DataSource=dt;
        conn.Close();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        getemp();

    }

    private void button1_Click(object sender, EventArgs e)
    {
        String query = "INSERT INTO employee(Eno,Ename,Edesc,Epay)
values(@Eno,@Ename,@Edesc,@Epay)";
        cmd = new OleDbCommand(query, conn);
        cmd.Parameters.AddWithValue("@Eno", textBox1.Text);
        cmd.Parameters.AddWithValue("@Ename", textBox2.Text);
        cmd.Parameters.AddWithValue("@Edesc", textBox3.Text);
        cmd.Parameters.AddWithValue("@Epay", textBox4.Text);
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        MessageBox.Show("Inserted Successfully");
        getemp();
    }

    private void button2_Click(object sender, EventArgs e)
```

```
        {
            String query = "UPDATE employee SET Eno=@Eno, Ename=@Ename, Edesc=@Edesc,
Epay=@Epay WHERE ID=@id";
            cmd = new OleDbCommand(query, conn);
            cmd.Parameters.AddWithValue("@Eno", textBox1.Text);
            cmd.Parameters.AddWithValue("@Ename", textBox2.Text);
            cmd.Parameters.AddWithValue("@Edesc", textBox3.Text);
            cmd.Parameters.AddWithValue("@Epay", textBox4.Text);
            cmd.Parameters.AddWithValue("@id", textBox6.Text);
            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();
            MessageBox.Show("Updated Successfully");
            getemp();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            String query = "DELETE FROM employee WHERE ID=@id";
            cmd = new OleDbCommand(query, conn);
            cmd.Parameters.AddWithValue("@id", textBox6.Text);
            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();
            MessageBox.Show("Deleted Successfully");
            getemp();
        }

        private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
```

```
            textBox6.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
            textBox1.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
            textBox3.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
            textBox4.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
        }

        private void textBox5_TextChanged(object sender, EventArgs e)
        {
            DataView dv = dt.DefaultView;
            dv.RowFilter = "Ename LIKE '%" + textBox5.Text + "%'";
            dataGridView1.DataSource = dv;
        }
    }
}
```

OUTPUT :



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 7	REGISTRATION FORM
DATE :	

AIM :

To Demonstrate the Registration Form design using visual C# Web Application.

PROCEDURE :

Step 1:

Open Visual Studio and click on File Menu ->New -> Website->Visual C#->ASP.NET Empty Website->Browser your folder->ok.

Step 2:

Select empty-> click ok In Solution Explorer->Right click the website icon ->Add->Web Form-> Enter web form name->ok

Step 3:

Default.Aspx->design->in div part.

Step 4:

Design the online application form.

Step 5:

Table->Insert Table->column: 2, row: 12->ok

Step 6:

Design the page.

Label	LabelName	Toolbox	Properties	AddItems
Label1	Online Course RegistrationForm	-	-	-
Label2	Nameofthe Applicant	TextBox	TextMode:Single Line ID:TextBox1	-
Label3	DateofBirth	TextBox	TextMode:Date ID:TextBox2	-
Label4	Email Id	TextBox	TextMode:Email ID:TextBox3	-
Label5	MobileNo.	TextBox	TextMode:Phone ID:TextBox3	-
Label6	Gender	ListBox	SelectionMode: Single ID:ListBox1	Male Female

Label7	**Course**	**ListBox**	**SelectionMode: Single** **ID:ListBox1**	**Java** **Python** **C** **C++** **HTM** **L** **CSS**
Label8	**Address**	**TextBox**	**TextMode: Multiline** **ID:TextBox5**	**-**
Label9	**FatherName**	**TextBox**	**TextMode:Single** **ID:TextBox6**	**-**
Label 10	**MotherName**	**TextBox**	**Mode:Single** **ID:TextBox7**	**-**
Label 11	**Education Qualification**	**ButtonRadio List**	**ID:TextBox7**	**10^{th}** **12^{th}** **UG** **PG**

Label 12	**Programming LanguageKnown**	**ListBox**	**SelectionMode: Multiple** **ID:ListBox3**	**Java** **Python** **C** **C++** **HTM** **L CSS** **JavaScript**
Label 13	**PayMode**	**RadioButton**	**Text:Online** **ID:RadioButton1**	**-**
-	**-**	**Button**	**Text:Submit**	**-**
Label 14	**-**	**-**	**ID:lbldisplay**	**-**

Step7:

Double click the submit button and type the below code:

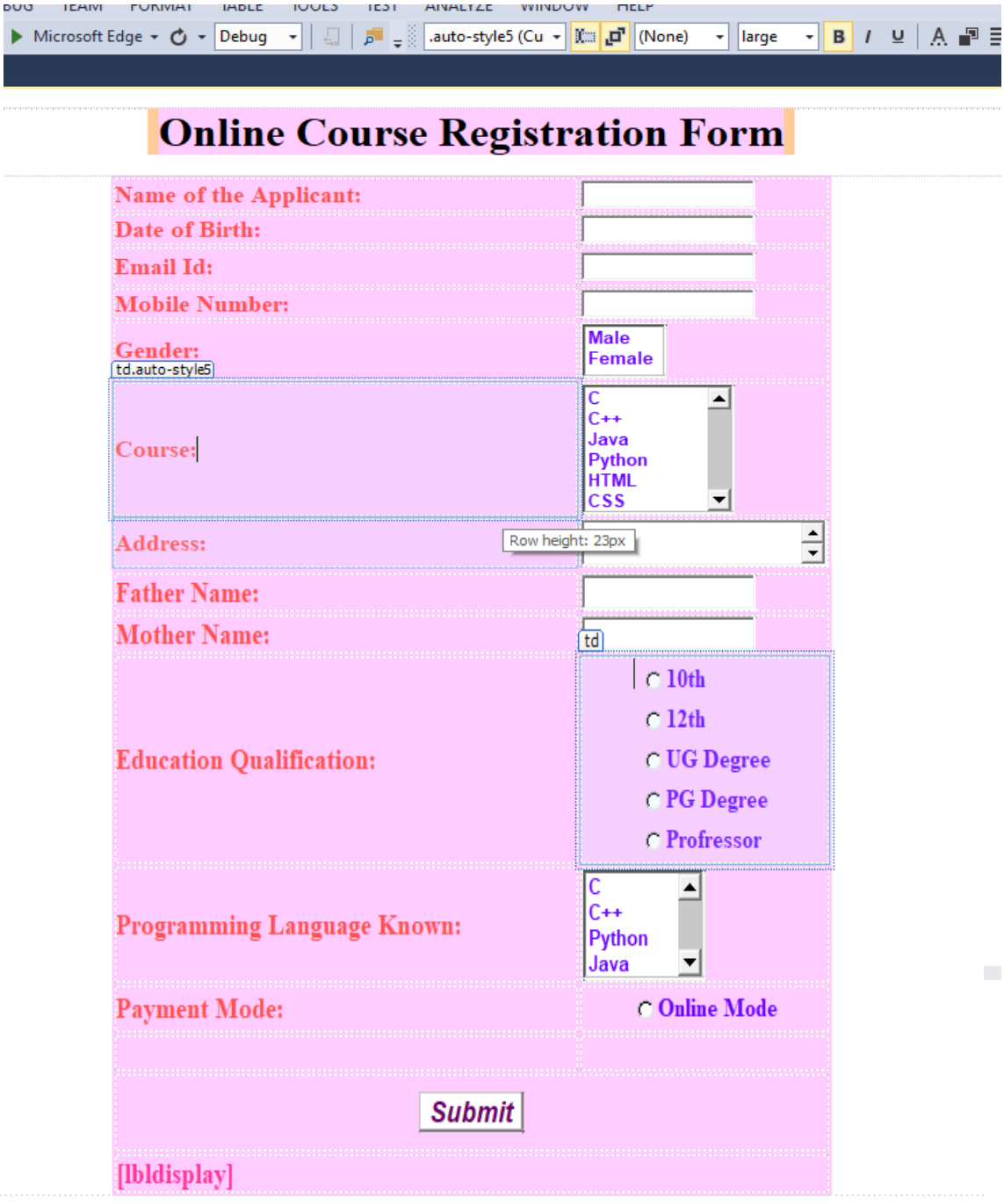
```
protectedvoidButton1_Click(objectsender,EventArgs e)
    {
        lbldisplay.Text = "<center>" + "<br>" + "Online Course Registraion Form" +"</center>"+
"<br>" + "<b>" + "<br>" + "Name of the Applicant:" +
TextBox1.Text+"<br>"+"<br>"+"DateofBirth:"+TextBox2.Text+"<br>"
+ "<br>" + "Email Id:" + TextBox3.Text + "<br>" + "<br>" + "Mobile Number:" +
TextBox4.Text + "<br>" + "<br>" + "Gender:" + ListBox3.Text +
"<br>"+"<br>"+"Course:"+ListBox1.Text+"<br>"+"<br>"+"Address:"
+TextBox6.Text+"<br>"+"<br>"+"FatherName:"+TextBox7.Text+
"<br>"+"<br>"+"MotherName:"+TextBox8.Text+"<br>"+"<br>"+
"EducationQualification:"+RadioButtonList1.Text+"<br>"+"<br>"+ "Payment Mode:" +
RadioButtonList2.Text + "<br>" + "<br>" + "Programming Language Known:" +
```

```
"<br>"+"<center>";

foreach(ListItem slist in ListBox4.Items)
        {
            if(slist.Selected)
            {
                lbldisplay.Text = lbldisplay.Text+slist+"<br>";                    }
        }
```

Step8:

Then save and run the code in the browser**.**

OUTPUT:


Microsoft Edge
Debug
.auto-style5 (Cu
(None)
large
Online Course Registration Form
Name of the Applicant:
Date of Birth:
Email Id:
Mobile Number:
Gender:
Male
Female
td.auto-style5
Course:
C
C++
Java
Python
HTML
CSS
Address:
Row height: 23px
Father Name:
Mother Name:
td
Education Qualification:
10th
12th
UG Degree
PG Degree
Profressor
Programming Language Known:
C
C++
Python
Java
Payment Mode:
Online Mode
Submit
[lbldisplay]

Online Course Registraion Form

Name of the Applicant:R.Rani

Date of Birth:2023-07-25

Email Id:bb@gmail.com

Mobile Number:6754756487

Gender:Female

Course:Java

Address:madurai

Father Name:raja

Mother Name:geetha

Education Qualification:PG Degree

Payment Mode:Online Mode

Programming Language Known:

Python

Java

RESULT :

Thus the program was executed successfully.

EXERCISE NO : 8	DATA LIST
DATE :	

AIM :

To Demonstrate the Data list design using visual C# Web Application.

PROCEDURE:

Step1:

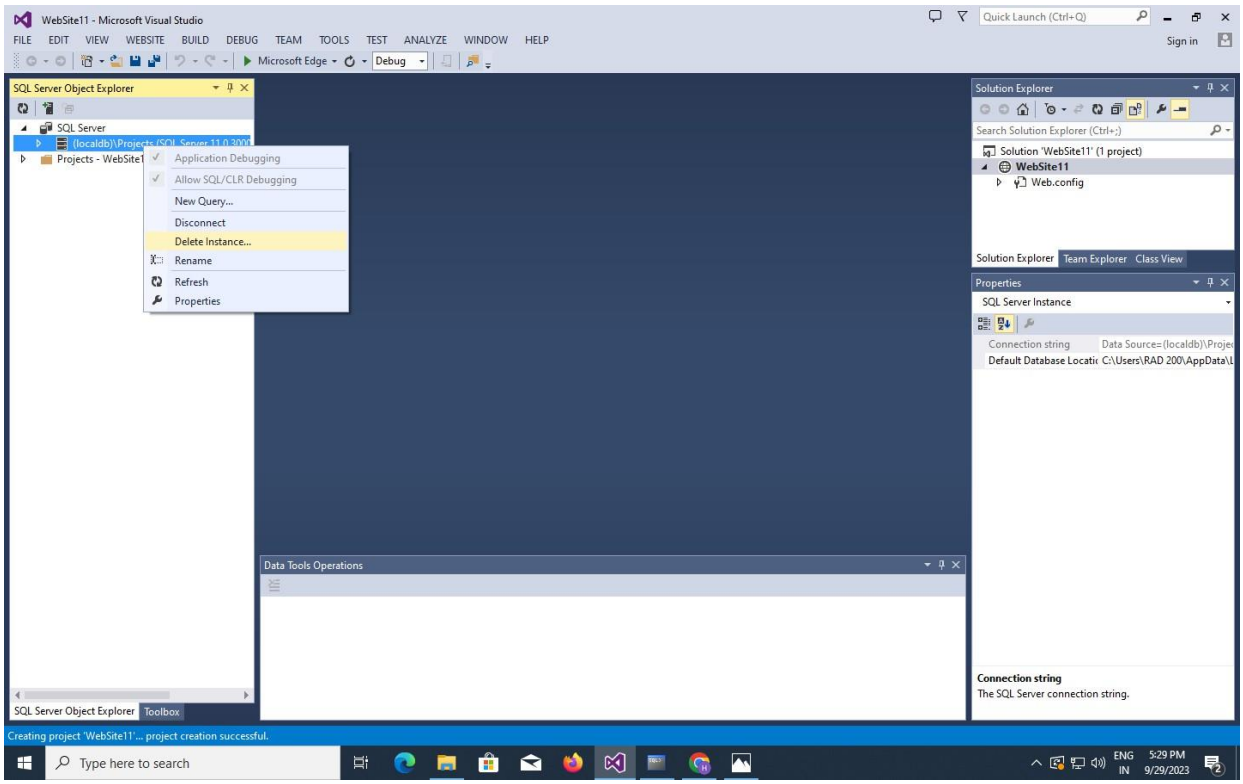
GotoFile→new→website,Invisualc#→choosethe1stoneASP.NET EMPTY web site.

Step2:

ASP.NETEMPTYwebsiteprojectwindowwillbeopen.thenfromtheleft side you can see SQL server Object Explorer.

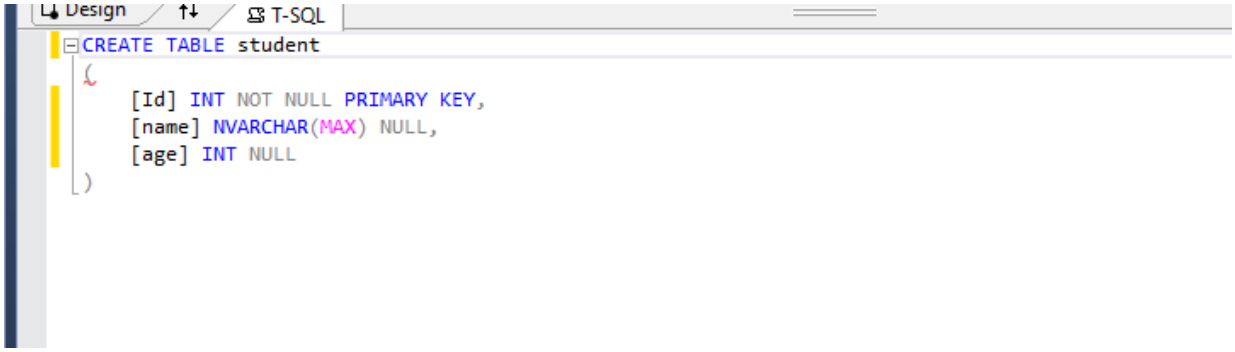
Step3:

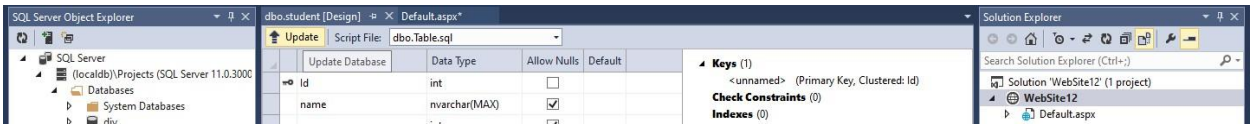
InSQLserverObjectExplorer→selectSQLServerleftarrow→choose (localdb)\Projects then right click and copy the name"(localdb)\Projects".



Step4:

In" (localdb)\Projects"→ Database → right click → add new database as peryourwisheg..(test)→open the left arrow→tables→rightclick→add new table → add Id int ,name as nvarchar(50),Age as int and in the below of your program (you can see the below eg..image) rename the table name as eg..student and select update from the top of the page →update database.



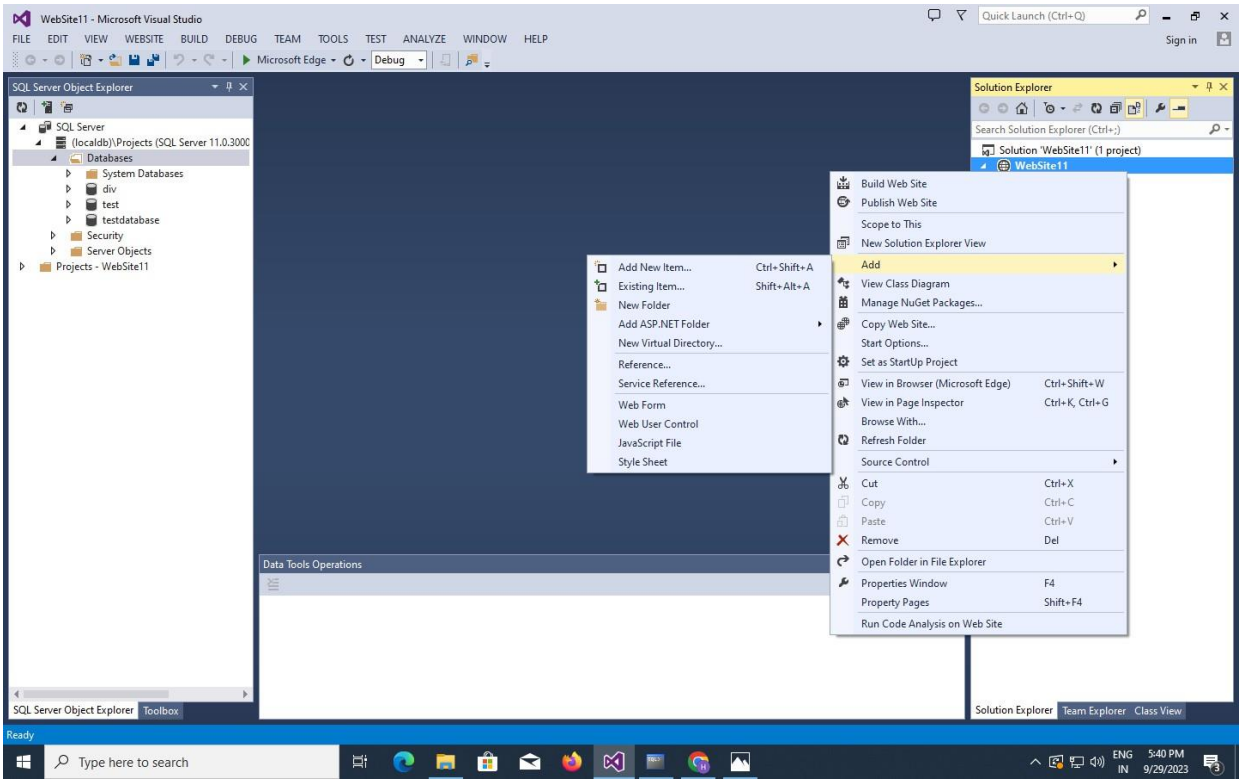


Step5:

Then rightclick on tables→dbo.student→rightclick→viewdata→add values and save.

Step6:

In solution explorer ,select the projectname (eg website11 as shown in belowfigure)→rightclick→add→addnewitem→in[visualc#]→select webfrom(default.aspx will be created).

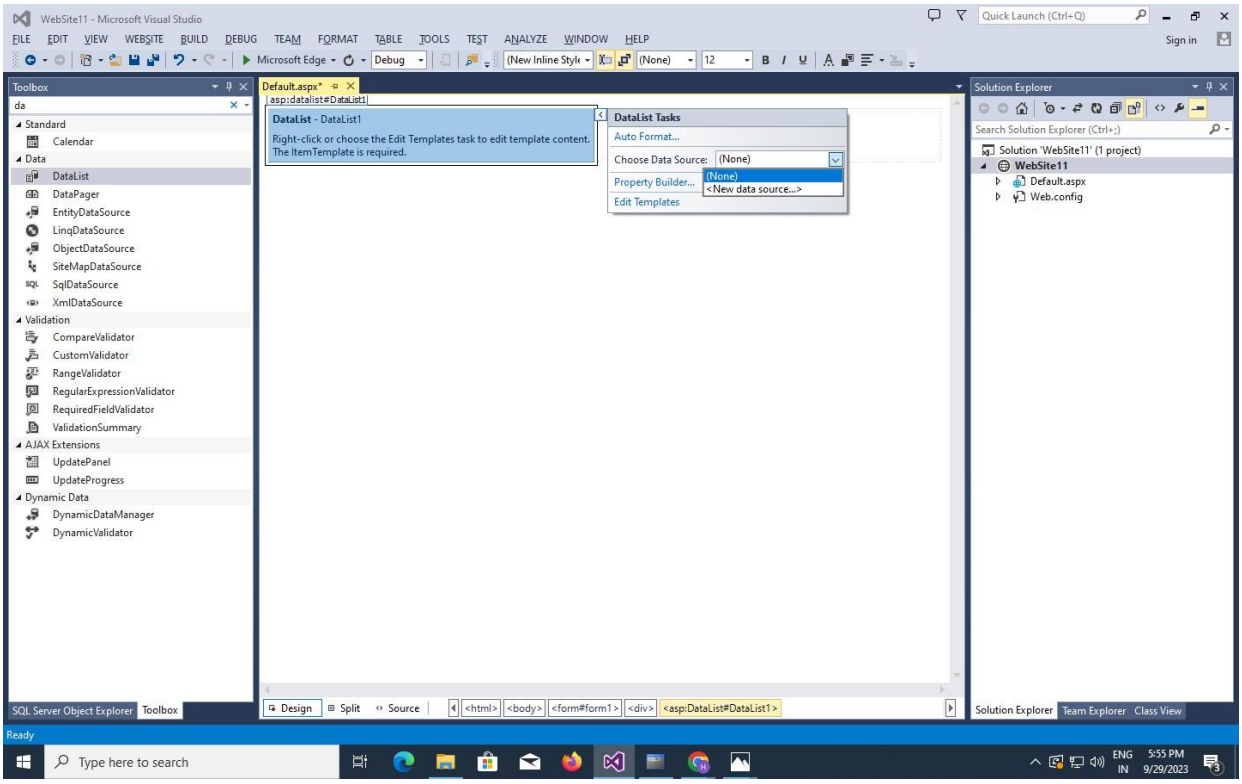


Step7:

In default.aspxchoose “design” from the bottom of the aspx page.then select”datalist”fromthetoolboxdragandplacethedatalistindivtagof aspx page.

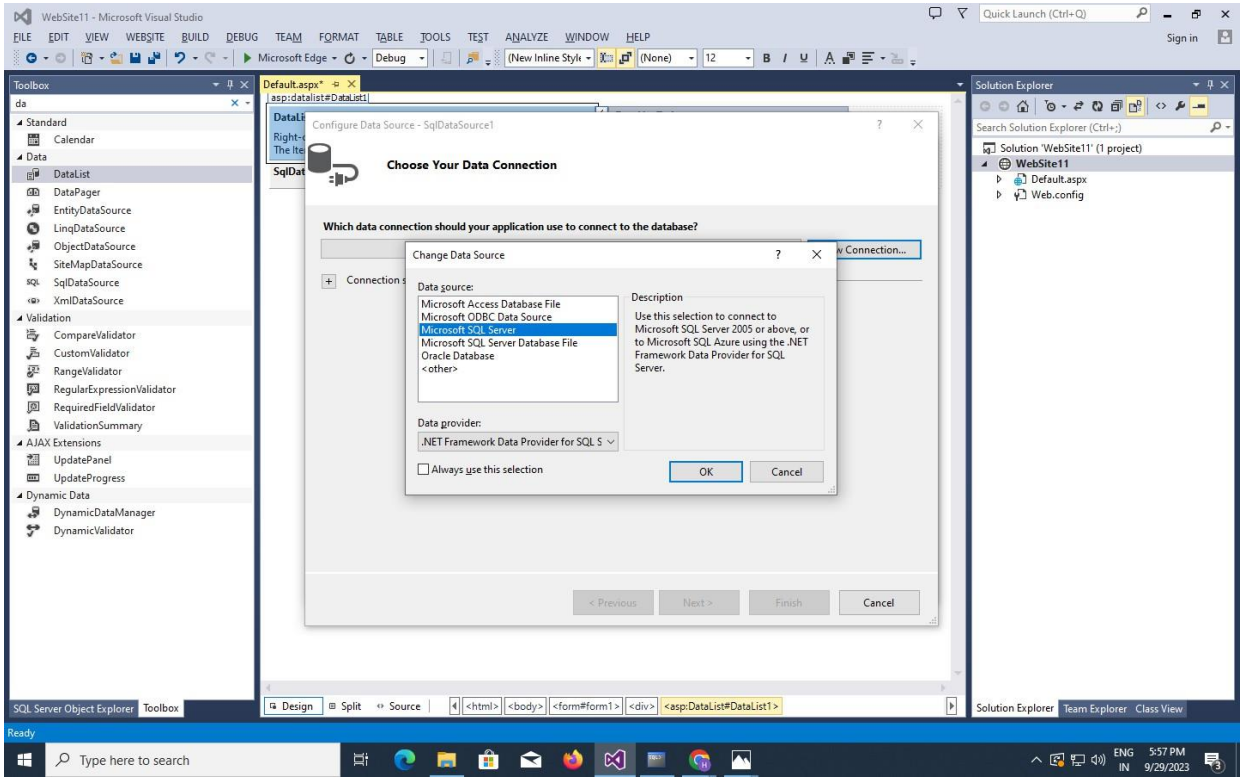
Step8:

Then click the arrow of the datalist as shown in the below image→in choose data source →<new data source>



Step9:

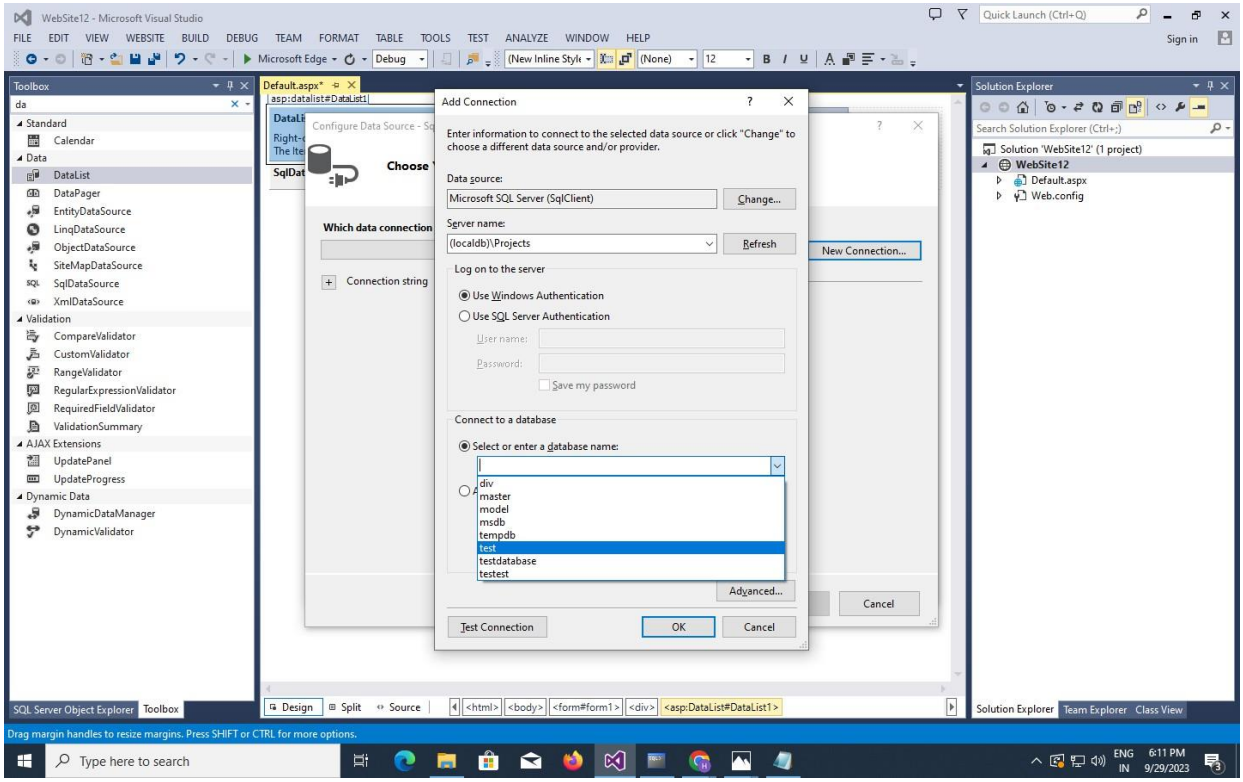
Then select SqlDatabase→ok→newconnection→indatasourceclick change → choose microsoft SQL Server as shown in the below image
→ok.



Step10:

For server name →paste the server name you copied from step 4 as ”(localdb)\Projects”then→choosethedatabasename(youcreatedas eg..test) from the dropdown → ok →next →next →select ID,name,age checkboxes →next →test query →finish.

Step11:

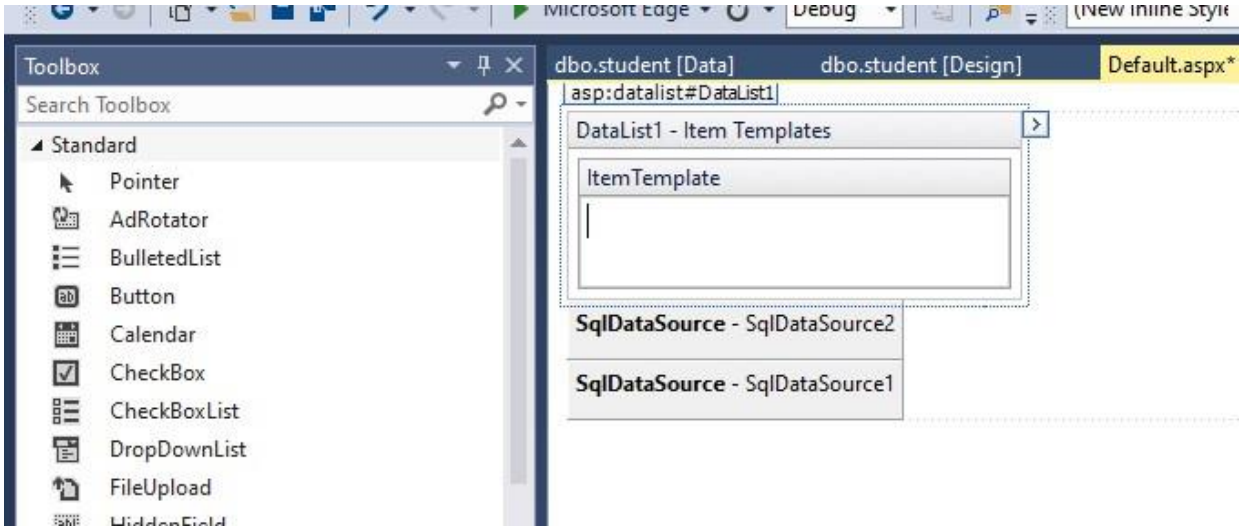


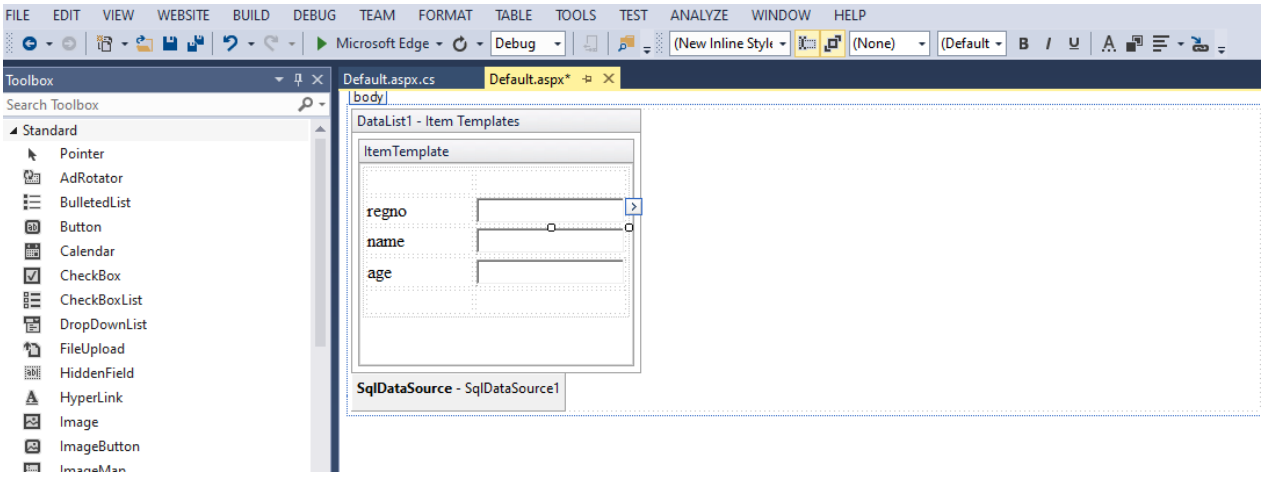
Step12:

Thenagainclickthearrowofthedatalist→edittemplate→thendeletethe items inside the datalist as shown.

Step13:

Selecttablefromthetopofthewindowinserttherowsand2columns.then add label and textbox as shown.


Toolbox
Search Toolbox
Standard
Pointer
AdRotator
BulletedList
Button
Calendar
CheckBox
CheckBoxList
DropDownList
FileUpload
dbo.student [Data]
dbo.student [Design]
Default.aspx*
asp:datalist#DataList1
DataList1 - Item Templates
ItemTemplate
SqlDataSource - SqlDataSource2
SqlDataSource - SqlDataSource1


FILE EDIT VIEW WEBSITE BUILD DEBUG TEAM FORMAT TABLE TOOLS TEST ANALYZE WINDOW HELP
Microsoft Edge
Debug
Toolbox
Search Toolbox
Standard
Pointer
AdRotator
BulletedList
Button
Calendar
CheckBox
CheckBoxList
DropDownList
FileUpload
HiddenField
HyperLink
Image
ImageButton
Default.aspx.cs
Default.aspx*
body
DataList1 - Item Templates
ItemTemplate
regno
name
age
SqlDataSource - SqlDataSource1

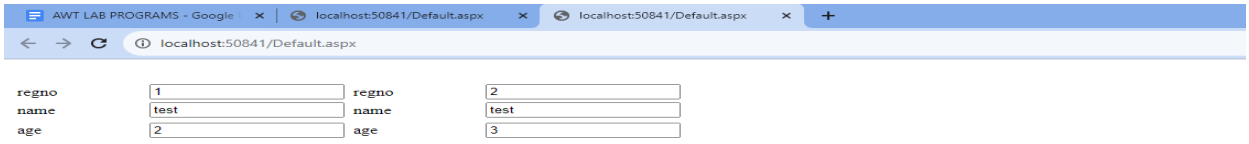
Step14:

Then right click on the arrow of the regno textbox→Editdatabindings→ bound to → select →ID. Then do the same for the rest of the textbox.add colors for background and for the textbox.

Step15:

Then right click the arrow of the datalist→end template editing→goto properties of the datalist → change [repeat direction to horizontal and Repeat column to 3] then run the program.

OUTPUT:



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 9	CRUD OPERATIONS
DATE :	

AIM :

To implement the Login Form using visual C# Windows Form Application.

PROCEDURE:

Step1:

OpenVisualStudio andclick onFileMenu ->New->website

Step2:

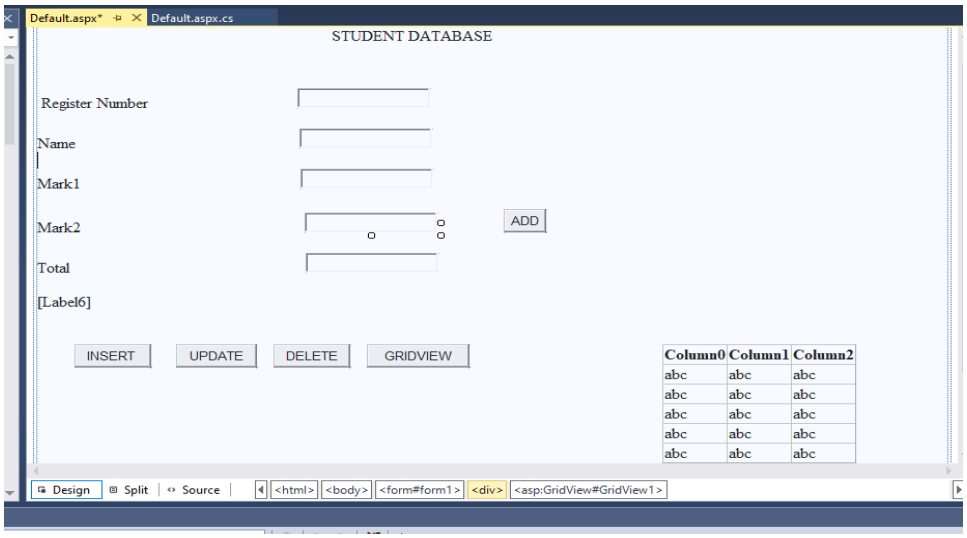
In the left side of the dialoguebox,expand the Templates-> Visual C#
Select theASP.NETWeb formsite Click on OK.

Step3:

In the solutionexplorer->(website11) rightclickit.
Add -> select new item -> select webform -> and then click add.

Step4:

Design the form as per as your wish.Taking the below form as your reference.



Step5:

In view->select SQLSERVEROBJECTEXPLORER

Select SQLSERVER and CLICK(localdbproject)->rightclick-> select rename and copy it(local db\ )project).

Step6:

Rightclick the DATABASE and add NEWDATABASE

Create a DATABASENAME as you like and click ok.

Step7:

Clickthe NEWCREATEDDATABASENAME->select TABLES and rightclick it->select ADDNEWTABLE. To specify the needed attributes in that table..

Step8:

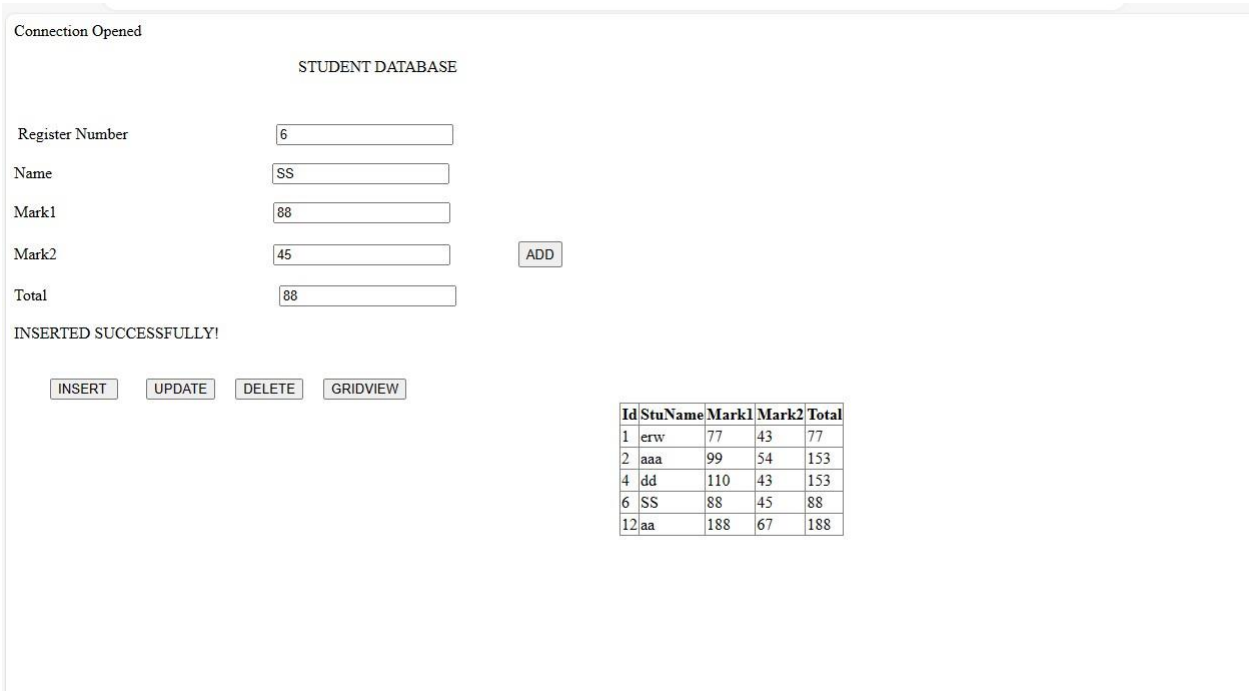
Upon the leftside-click UPDATE to save the database table->click UPDATE DATABASE.

Step9:

Goto DESIGNPAGE->doubleclick the BUTTONS(INSERT/UPDATE/ DELETE/GRIDVIEW) -> To insert the coding.

Step10:

Finally,RUN the program.



Id	StuName	Mark1	Mark2	Total
1	erw	77	43	77
2	aaa	99	54	153
4	dd	110	43	153
6	SS	88	45	88
12	aa	188	67	188

Step7:

Clickthe NEWCREATEDDATABASENAME->select TABLES and rightclick it->select ADDNEWTABLE. To specify the needed attributes in that table..

Step8:

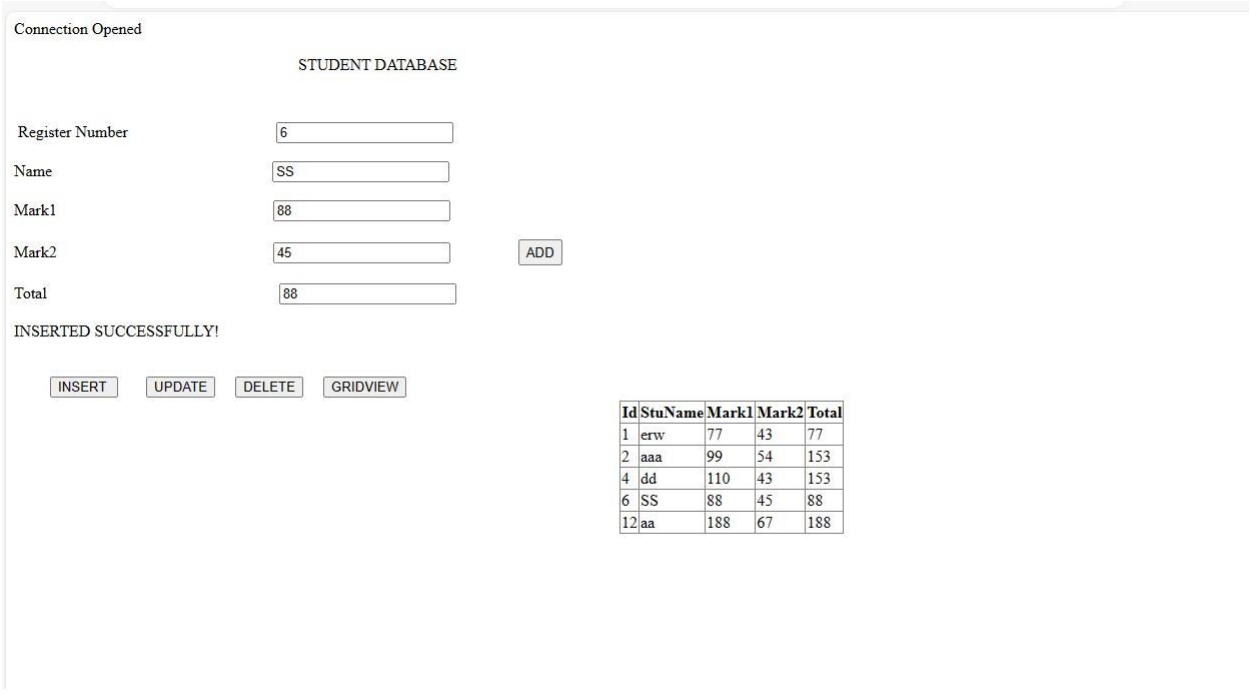
Upon the leftside-click UPDATE to save the database table->click UPDATE DATABASE.

Step9:

Goto DESIGNPAGE->doubleclick the BUTTONS(INSERT/UPDATE/ DELETE/GRIDVIEW) -> To insert the coding.

Step10:

Finally,RUN the program.



Id	StuName	Mark1	Mark2	Total
1	erw	77	43	77
2	aaa	99	54	153
4	dd	110	43	153
6	SS	88	45	88
12	aa	188	67	188

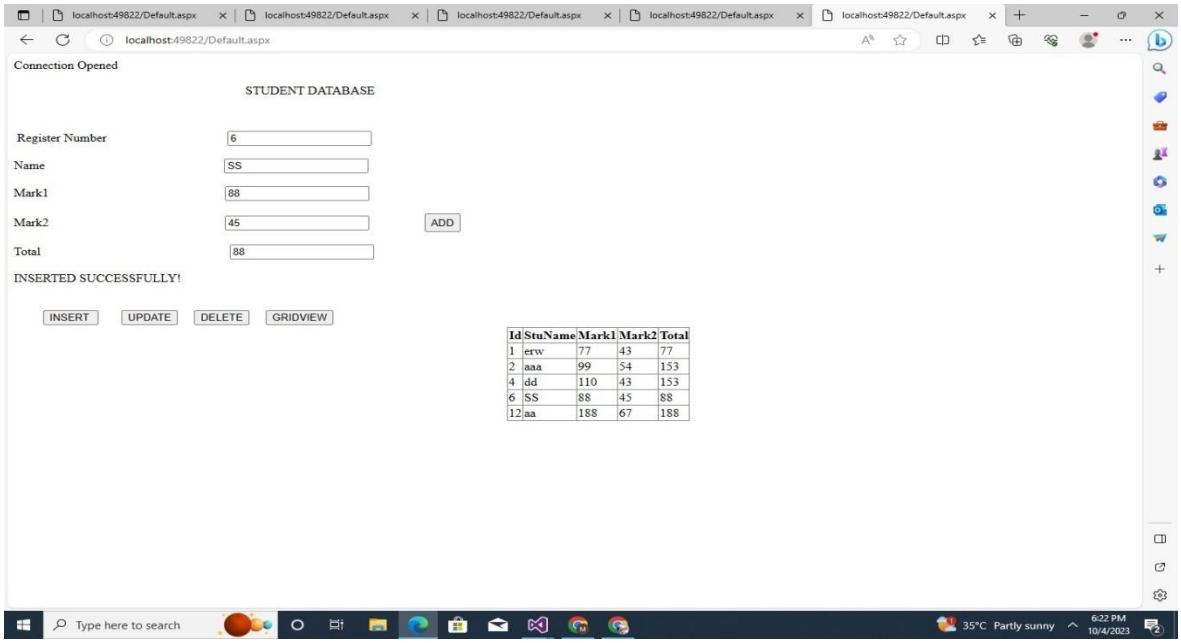
SOURCE CODE:

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

public partial class Default2 : System.Web.UI.Page
{
   SqlConnection con = new SqlConnection("Data Source=(localdb)\\Projects; Initial Catalog=student1;
Integrated Security=True");

   protected void Page_Load(object sender, EventArgs e)
   {

   }
   protected void Button2_Click(object sender, EventArgs e)
   {
      try
      {

         con.Open();
         Response.Write("Connection Opened");

         int a = Int32.Parse(TextBox3.Text);
         int b = Int32.Parse(TextBox4.Text);
         int c = a + b;
         TextBox5.Text = c.ToString();
         string query = "insert into Stud(RegNo,Name,Mark1,Mark2,Total) values(" + TextBox1.Text + ",
'" +TextBox2.Text + "'," + TextBox3.Text + "," + TextBox4.Text + "," +TextBox5.Text + ")";
         SqlCommand sc = new SqlCommand(query, con);

         int status = sc.ExecuteNonQuery();

         Label7.Text = "INSERTED SUCCESSFULLY!";
        }
         catch (Exception ex)
         {
            Response.Write("\nOOPs, something went wrong." + ex);
         }

         finally
         {
            con.Close();
         }
```

```
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
     try
     {
        con.Open();
        Response.Write("Connection Opened");
        int a = Int32.Parse(TextBox3.Text);
        int b = Int32.Parse(TextBox4.Text);
        int c = a + b;
        TextBox5.Text = c.ToString();
        string query = "update Stud set Name='" + TextBox2.Text +"'" + " ,Mark1=" + TextBox3.Text + "
,Mark2=" + TextBox4.Text + " ,Total=" +TextBox5.Text + "where RegNo = " + TextBox1.Text;
        SqlCommand sc = new SqlCommand(query, con);

        int status = sc.ExecuteNonQuery();

        Label7.Text = "UPDATED SUCCESSFULLY!";
        }
        catch (Exception ex)
        {
           Response.Write("\nOOPs, something went wrong." + ex);
        }
        finally
        {
           con.Close();
        }
    }
    protected void Button4_Click(object sender, EventArgs e)
    {
        try
        {
           con.Open();
           Response.Write("Connection Opened");
           string query = "delete Stud where RegNo = " + TextBox1.Text;
           SqlCommand sc = new SqlCommand(query, con);
           int status = sc.ExecuteNonQuery();
           Label7.Text = "DELETED SUCCESSFULLY!";
        }
        catch (Exception ex)
        {
           Response.Write("\nOOPs, something went wrong." + ex);
        }

        finally
        {
           con.Close();
        }
    }
    protected void Button5_Click(object sender, EventArgs e)
```

```
    {
        try
        {
            con.Open();
            Response.Write("Connection Opened");
            SqlCommand cm = new SqlCommand("select * from Stud",con);
            SqlDataAdapter sda = new SqlDataAdapter(cm);
            DataSet ds = new DataSet();
            sda.Fill(ds);
            GridView1.DataSource = ds;
            GridView1.DataBind();
            }
            catch (Exception ex)
            {
                Response.Write("OOPs, something went wrong." + ex);
            }
            finally
                {
                    con.Close();
                }
        }
         protected void Button1_Click(object sender, EventArgs e)
        {
            int a = Int32.Parse(TextBox3.Text);
            int b = Int32.Parse(TextBox4.Text);
            int c = a + b;
            TextBox5.Text = c.ToString();
        }
}
```

OUTPUT:



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 10	MODEL VIEW CONTROLLER
DATE :	

AIM :

To Demonstrate the Model View Controller design using visual C# Web Application.

PROCEDURE:

1. Open webapplication->ChooseMVC

2. View-->SQLServerObjectExplorer->copy the database server name->

Databases->rightclick->addnewdatabase->clickthearrowofyour database->tables->right click add new table.

2. RightclickModels->Add->NewItem->visualC#->data-

>ADO.NETEntitydatamodel.

3. Entity data model wizard->Generate from database->New connection->ServerName-(localdb)\Projects->selectdatabasename-

>clicktestconnection->clickok

4. copy entityname in the web.config->choosetables->clickfinish

5. Rightclick projectnam eform the solutionexplorer->rightclick rebuild

6. Model1.edmxwillbeaddedunderModelsfolder

7. RightclicktheControllers->Add->Controller'

8. MVC5 controllerwithviews,usingentityframework->clickadd

9. Addcontrollername->clickmodelclass->selecttablenamefrom the drop down(MVC.Models)

10. Datacontextclass->chooseentitynames->add

11. Controllers will be added in Controllers folder

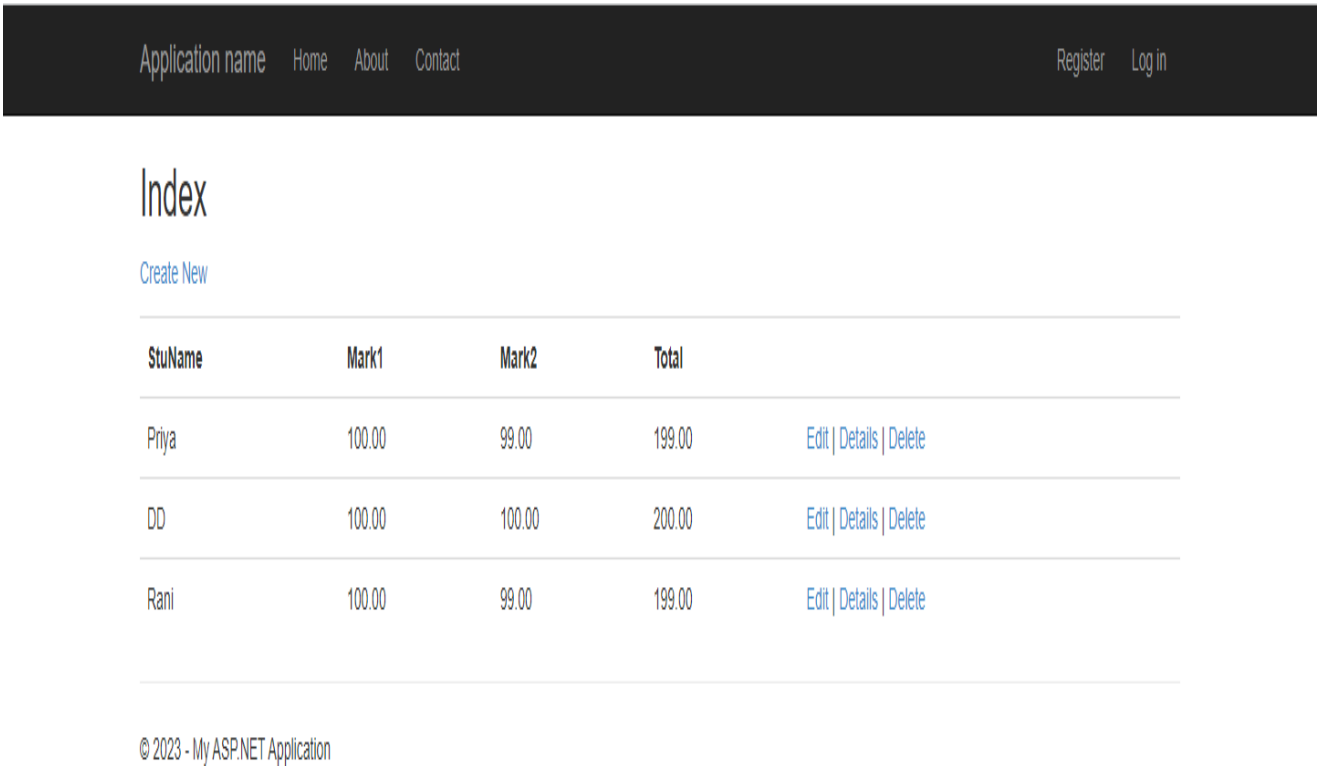
12.Controllernamedfolderwillbeaddedunderviewsfolder

13. clickApp_start->Routeconfig.CS

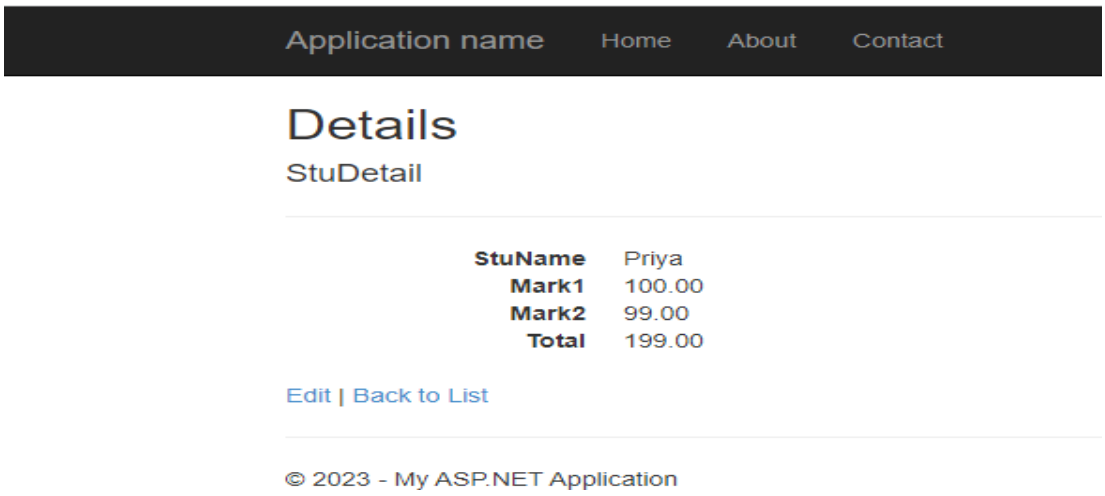
14. Modify the controller name in to the“Yourscontrollername”inthe new controller fields,instead of index add “create”in the action field 15.Rebuild->Run.

OUTPUT:

VIEW



DETAIL



EXERCISE NO : 11	VALIDATION CONTROL
DATE :	

AIM :

To Demonstrate the Validation Control design using visual C# Web Application**.**

PROCEDURE:

STEP 1:

StartVisualStudio.ClickFile->New->WEBSITE.Thiswill bring you a new dialog box.

STEP 2:

In the dialogbox choose VisualC#->ASP.NETEmptyWebsite.

On that choose file location and enter the filename.

STEP 3:

On solutionExplorer,RightclicktheWebSite->Add->NewItem -> Web Form.

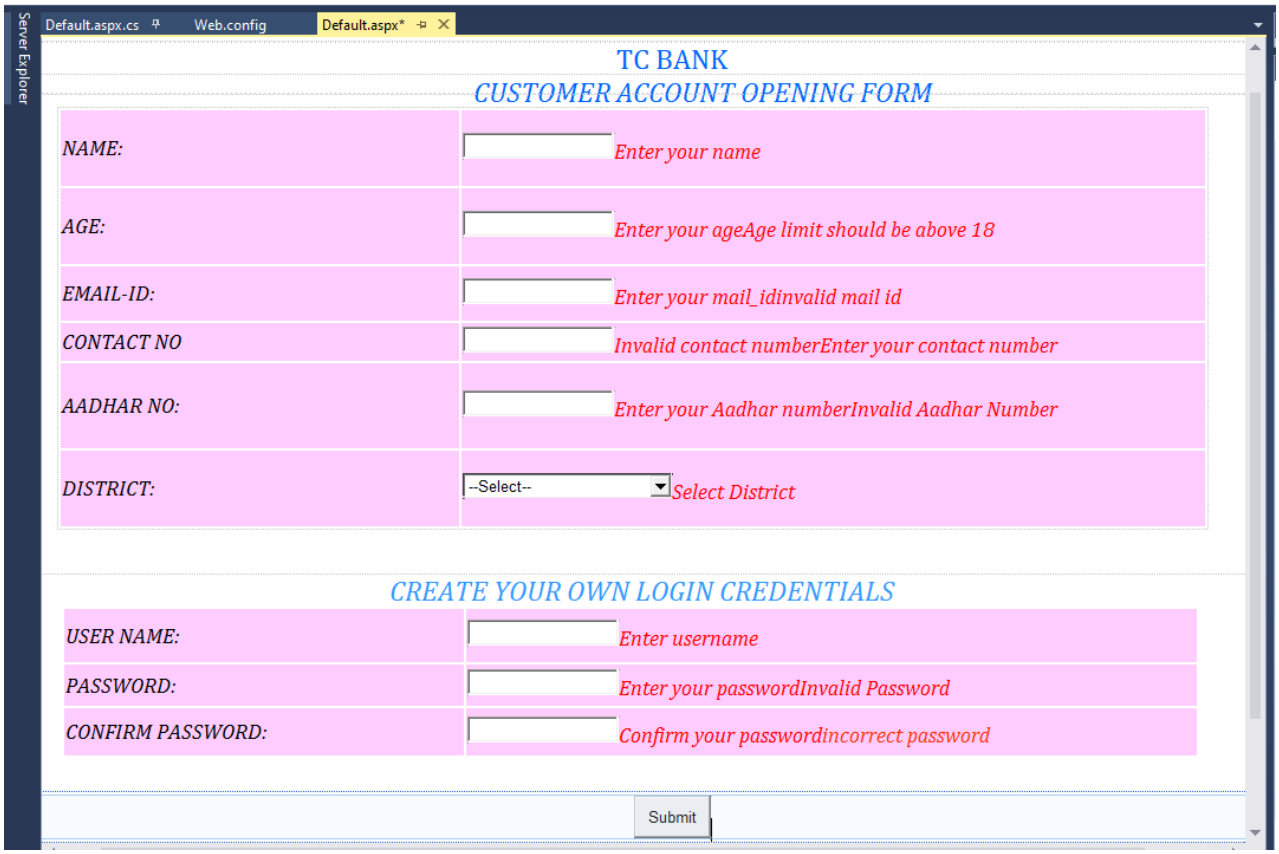
STEP 4:

In solutionExplorer,clickDefault.aspx,thisisaformwe can add design to the form either through coding or pick from the Toolbox.

STEP 5:

For validation control, Add Label boxes, Text Boxes , Dropdownlist,ButtonfromToolbox.Youcanuseatablefor correct alignment.

Designtheformasyourwish.Thebelowgivenformisthe reference.

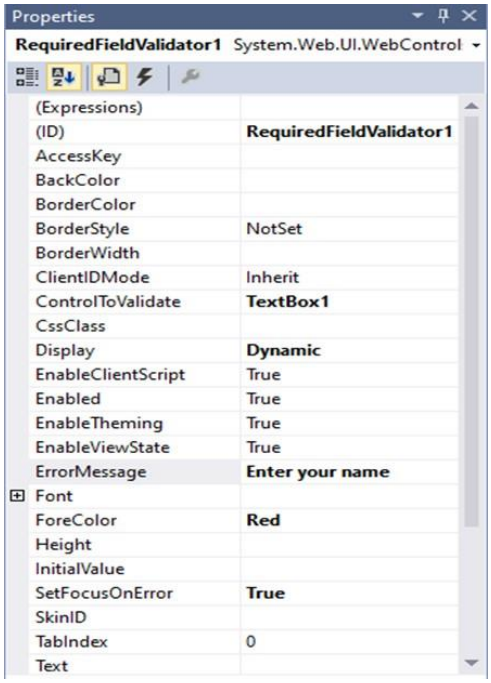


STEP 6:

After Design the Webform add validators from toolbox to the text boxes**.**

STEP 7:

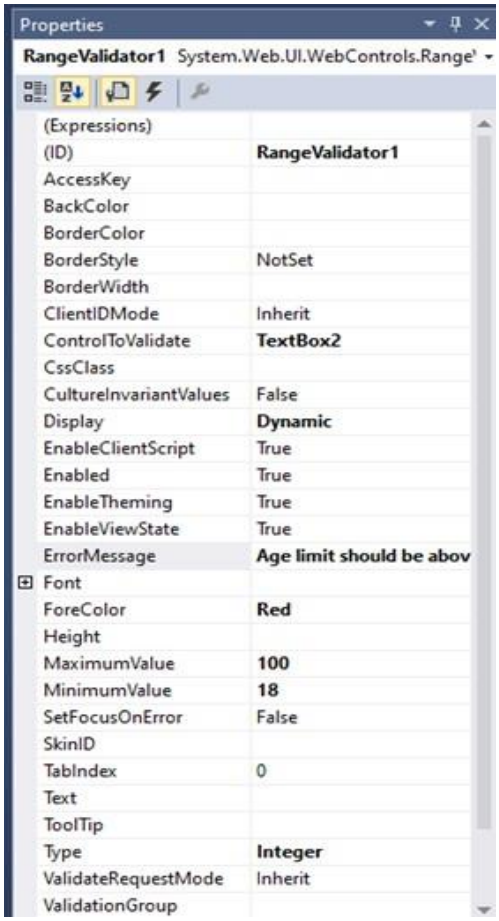
Add Required Field Validator to all textboxes.



Change the following common properties as shown above for all Required field validators according to textboxes such as ControlToValidate, Display, Error message, ForeColor, SetFocusOnError.

STEP 8:

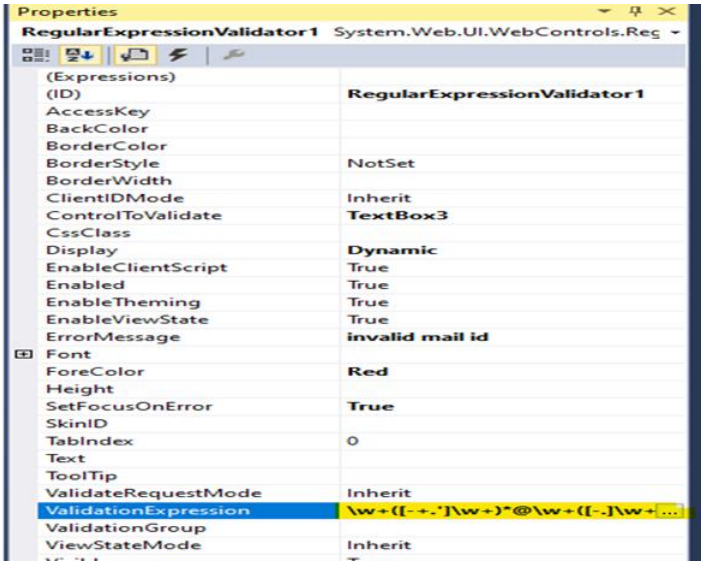
For the Age, textbox add Range Validator.

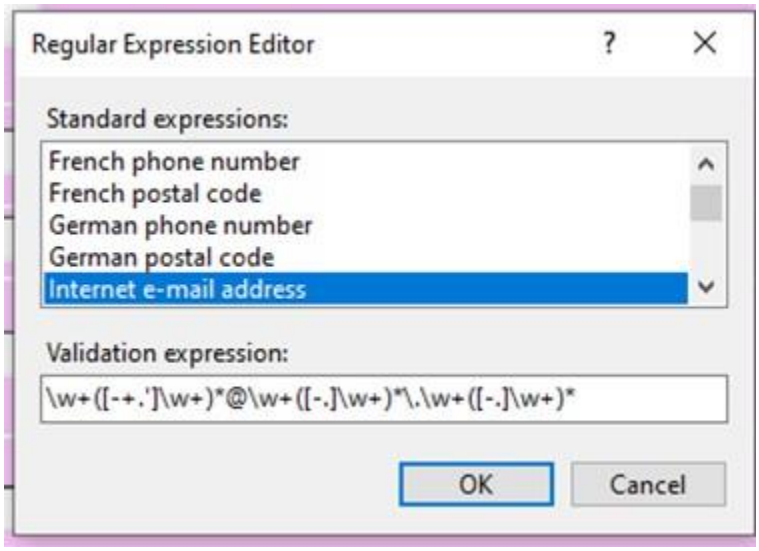


STEP 9:

For EmailID,Phonenumber,Aadharnumber textbox add RegularExpression Validator.

STEP 10:

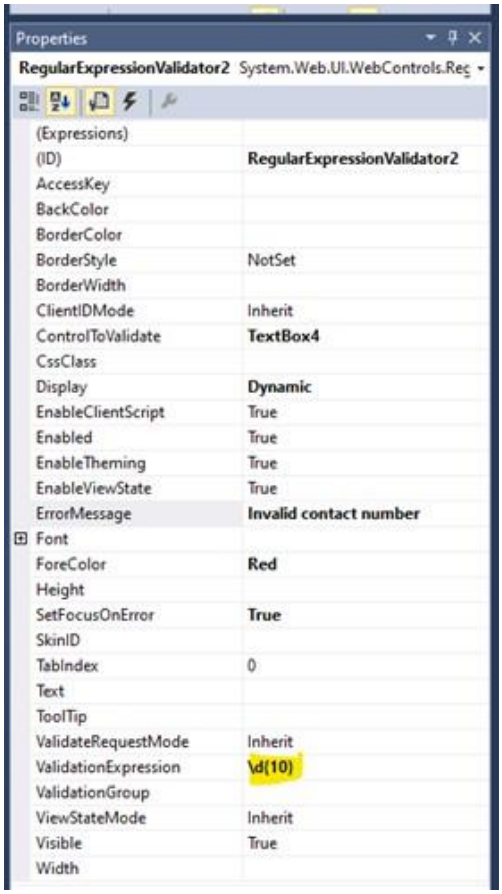




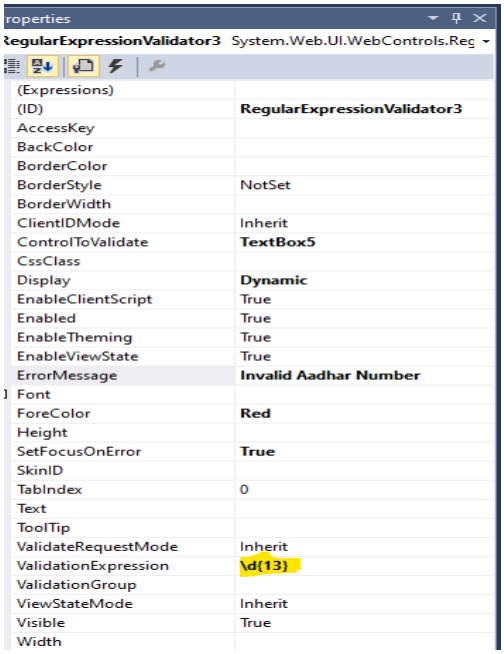
Select the Validation Expression and choose the email address

STEP 11:

For the contact number RegularExpressionValidator,change the value for the validation expression in properties as shown in the image below.


Properties
RegularExpressionValidator2 System.Web.UI.WebControls.Reg
(Expressions)
(ID) RegularExpressionValidator2
AccessKey
BackColor
BorderColor
BorderStyle NotSet
BorderWidth
ClientIDMode Inherit
ControlToValidate TextBox4
CssClass
Display Dynamic
EnableClientScript True
Enabled True
EnableTheming True
EnableViewState True
ErrorMessage Invalid contact number
Font
ForeColor Red
Height
SetFocusOnError True
SkinID
TabIndex 0
Text
ToolTip
ValidateRequestMode Inherit
ValidationExpression \d{10}
ValidationGroup
ViewStateMode Inherit
Visible True
Width

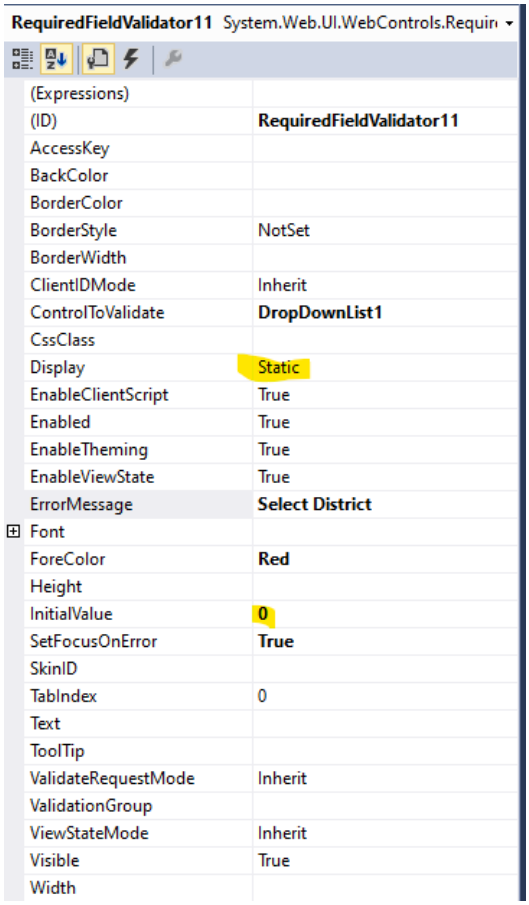
STEP 12:


roperties
RegularExpressionValidator3 System.Web.UI.WebControls.Reg
(Expressions)
(ID) RegularExpressionValidator3
AccessKey
BackColor
BorderColor
BorderStyle NotSet
BorderWidth
ClientIDMode Inherit
ControlToValidate TextBox5
CssClass
Display Dynamic
EnableClientScript True
Enabled True
EnableTheming True
EnableViewState True
ErrorMessage Invalid Aadhar Number
Font
ForeColor Red
Height
SetFocusOnError True
SkinID
TabIndex 0
Text
ToolTip
ValidateRequestMode Inherit
ValidationExpression \d{13}
ValidationGroup
ViewStateMode Inherit
Visible True
Width

For the aadharnumber RegularExpressionValidator,change the value for the validation expression in properties as shown in the image above.

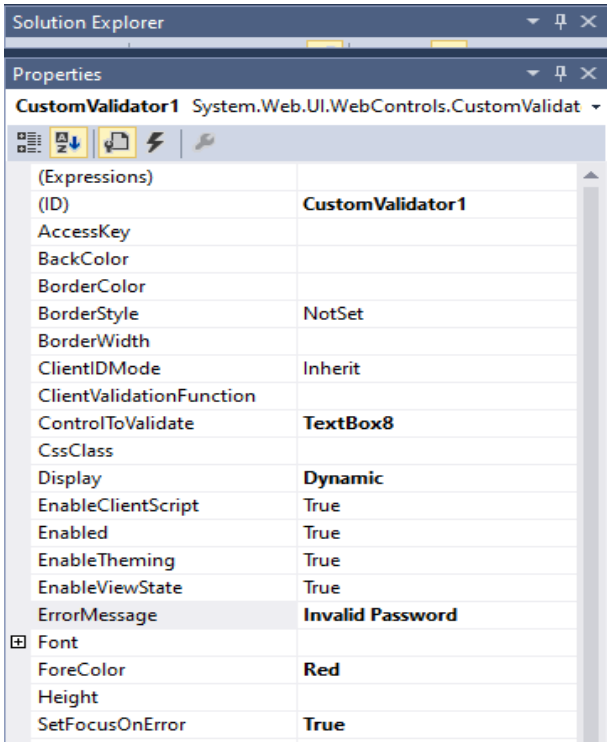
STEP 13:

If you want to add a drop downlist for district,you should change the highlighted values for the properties as shown below.



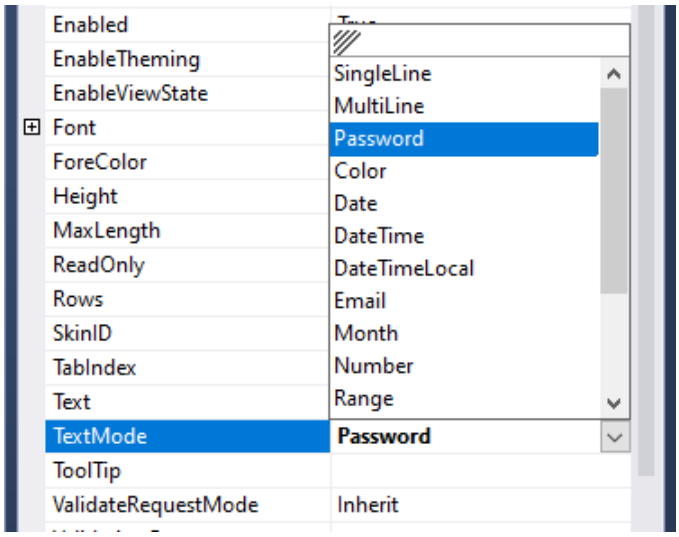
STEP 14:

Set customfieldvalidator for password.Change the following properties as shown in the image below.



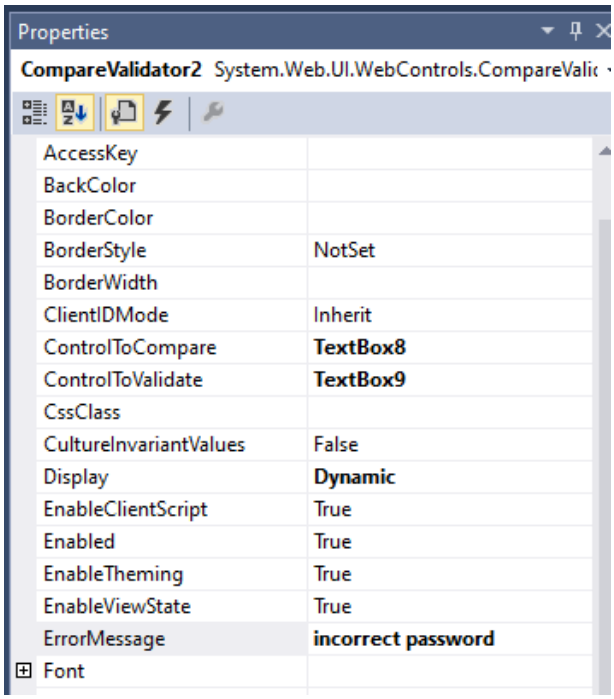
STEP 15:

Select the password textbox and in properties change the text mode property as shownbelow.Sothatthepassword appear as hidden.



STEP 16:

For the confirm password textbox add comparevalidator and change the values in the properties for the compare validatoras shown in the image.



Coding:

Default.aspx

<%@PageLanguage="C#"AutoEventWireup="true"UnobtrusiveValidationMode="None"CodeFile="Default.aspx.cs" Inherits="_Default" %>

- Type the highlighted coding in default.aspx

<asp:DropDownListID="DropDownList1"runat="server" Height="45px" style="margin-left: 0px" Width="177px"TabIndex="5">

<asp:ListItem Value="0"Selected="True">--Select--</asp:ListItem>

<asp:ListItem Value="1">chennai</asp:ListItem>

<asp:ListItem Value="2">coimbatore</asp:ListItem>

<asp:ListItem Value="3">trichy</asp:ListItem>

<asp:ListItem Value="4">kanyakumari</asp:ListItem>

<asp:ListItem Value="5">pondicherry</asp:ListItem>

</asp:DropDownList>

- If you have used drop down list for selecting the district, change the coding as above. REFER IT.

Click the submit button and type the below given coding.

Default.aspx.cs:

Coding:

```
using System;

usingSystem.Collections.Generic; using System.Linq;

using System.Web;

usingSystem.Web.UI;

using System.Web.UI.WebControls;

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

{

    int a=0;

    protectedvoidButton1_Click(objectsender,EventArgse)

    {

Response.Write("<script>alert('SUBMITTED')</script>");
     }
```

```
    protectedvoidCustomValidator1_ServerValidate(object source, ServerValidateEventArgs
args)

    {

    stringstr=args.Value; args.IsValid = false;
        if(str.Length<6||str.Length>12)

        {

        return;

        }

        boolupperCase=false; foreach (char ch in str)

        {

          if (ch >= 'A'&& ch <= 'Z')

                {

                    upperCase=true; break;

                }

         }


if (!upperCase)

        {

             return;

        }

         boollowerCase=false; foreach(charchinstr)

         {

            if (ch >= 'a'&& ch <= 'z')

              {

                  lowerCase=true; break;

                    }
```

```
 }
if (!lowerCase)
{
     return;
}
 bool number = false;

 foreach(charchinstr)
{
  if (ch >= '0'&& ch <= '9')
       {
            number=true; break;
       }
 }
 if (!number)
 {
      return;
 }
```

```
        args.IsValid = true;
        if(args.IsValid==true)
        {
            a = 1;
            return;

            }
       }
   protectedvoidTextBox8_TextChanged(objectsender, EventArgs e)
   {
   }
}
```

RESULT :

Thus the program was executed successfully.

EXERCISE NO : 12	SESSION & COOKIES
DATE :	

AIM :

To Demonstrate the Session and Cookies design using visual C# Web Application**.**

PROCEDURE:

Step 1: Open Visual Studio and click on File Menu ->New -> Project

Step 2:

In the left side template expand Visual C# and select Web -> ASP.NET Web Application. Enter the name of the application. (I named Sessions) and if you want to change project location then click on browse and set location.

Step 3:

Select Web Forms (Note- on Right side Authentication-No

Authentication). Click on OK.

Step 4: Adding a web form

Right click the name of the project "SessionExample" and move the mouse to Add > "Web Form" in submenu.

Name the web form as "SessionExample.aspx" in the name

text box.

Step 5: Add a Web User Control to the project. Right click the

name of the project "SessionExample" and select Visual c# > Web

> Web Forms User Control. Name it as "Session.*ascx"*.

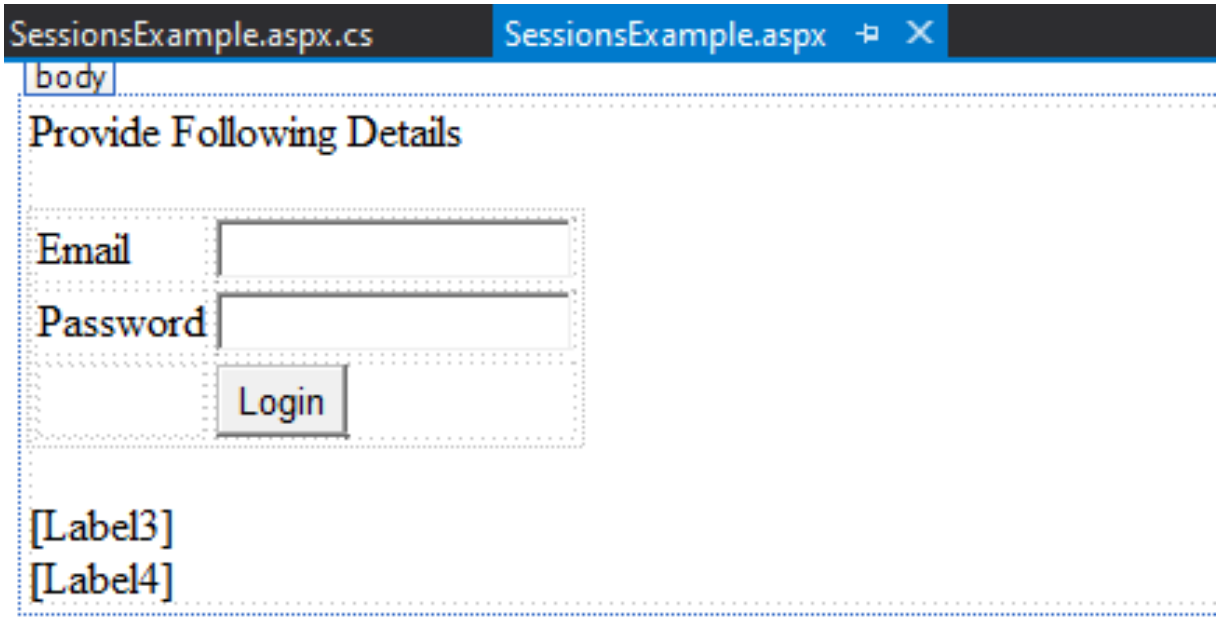
SessionExample.aspx

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Session Example</title>
   <style type="text/css">
      .auto-style2 {
         width: 100%;
      }
      .auto-style2 {
         width: 105px;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
      <p>Provide Following Details</p>
      <table class="auto-style2">
         <tr>
            <td class="auto-style2">Email</td>
            <td>
               <asp:TextBox ID="email" runat="server" TextMode="Email"></asp:TextBox>
            </td>
         </tr>
         <tr>
            <td class="auto-style2">Password</td>
            <td>
               <asp:TextBox ID="password" runat="server"
TextMode="Password"></asp:TextBox>
            </td>
         </tr>
         <tr>
            <td class="auto-style2"></td>
            <td>
               <asp:Button ID="login" runat="server" Text="Login" OnClick="login_Click" />
            </td>
         </tr>
      </table>
      <br />
      <asp:Label ID="Label3" runat="server"></asp:Label>
      <br />
      <asp:Label ID="Label4" runat="server"></asp:Label>
   </form>
</body>
</html>
```

SessionExample.aspx.cs

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Sessions
{
   public partial class SessionsExample : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         // Check if session variable "email" exists
         if (Session["email"] != null)
         {
            Label3.Text = "This email is stored in the session: ";
            Label4.Text = Session["email"].ToString();
         }
      }

      protected void login_Click(object sender, EventArgs e)
      {
         // Check password and store email in session
         if (password.Text == "tcarts")
         {
            Session["email"] = email.Text;
            Label3.Text = "Email is stored in session.";
         }
         else
         {
            Label3.Text = "Invalid password.";
         }
      }
   }
}
```

OUTPUT :


SessionsExample.aspx.cs
SessionsExample.aspx
body
Provide Following Details
Email
Password
Login
[Label3]
[Label4]

Cookies

PROCEDURE:

Step 1:

Open Visual Studio and click on File Menu ->New -> Project

Step 2:

In the left side template expand Visual C# and select Web -> ASP.NET Web Application. Enter the name of the application. (I named cookiesExample) and if you want to change project location then click on browse and set location.

Step 3:

Select Web Forms (Note- on Right side Authentication-No

Authentication). Click on OK.

Step 4:

Adding a web form

Right click the name of the project "cookiesexample" and move the mouse to Add > "Web Form" in submenu.

Name the web form as "cookiesExample.aspx" in the name text box.

Step 5:

Add a Web User Control to the project. Right click the name of the project "cookiesexample" and select Visual c# > Web > Web Forms User Control.

Name it as "cookies.*ascx"*.

CookiesExample

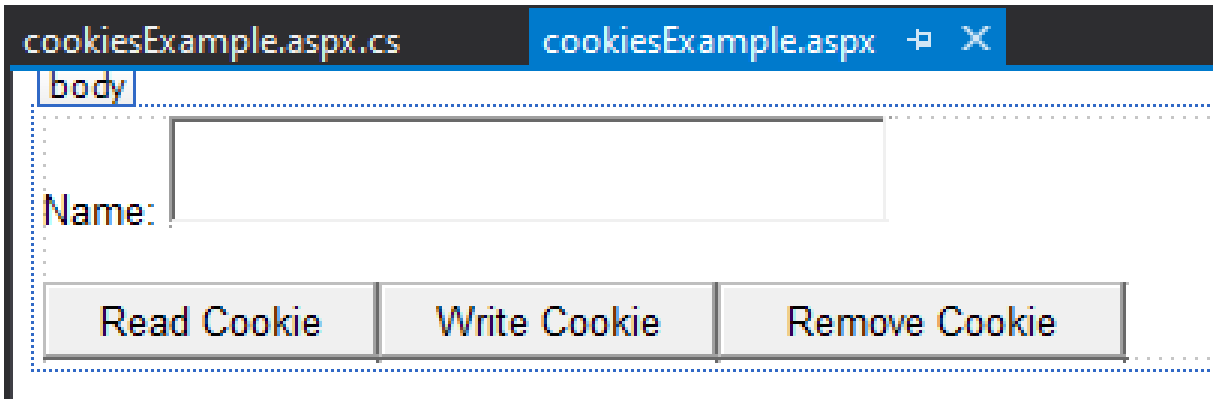
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="cookiesExample.cookiesExample" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
   <title>Cookies Example</title>
   <style type="text/css">
      body {
         font-family: Arial;
         font-size: 10pt;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
      <p>Name:</p>
      <asp:TextBox ID="txtName" runat="server" Height="30px"
Width="232px"></asp:TextBox>
      <br />
      <br />
      <asp:Button ID="btnRead" runat="server" Text="Read Cookie"
OnClick="btnRead_Click"/>
      <asp:Button ID="btnWrite" runat="server" Text="Write Cookie"
OnClick="btnWrite_Click" />
      <asp:Button ID="btnDelete" runat="server" Text="Remove Cookie"
OnClick="btnDelete_Click" />
   </form>
</body>
</html>
```

cookiesExample.aspx.cs

```
using System;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

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

      }
```

```
        // Button to read the cookie value
        protected void btnRead_Click(object sender, EventArgs e)
        {
            // Check if the "Name" cookie exists
            HttpCookie nameCookie = Request.Cookies["Name"];

            // If cookie exists, show its value. If it doesn't, show "undefined"
            string name = nameCookie != null ? nameCookie.Value : "undefined";
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('" + name + "');",
true);
        }

        // Button to write the cookie
        protected void btnWrite_Click(object sender, EventArgs e)
        {
            // Create a new cookie "Name" with the value from the TextBox
            HttpCookie nameCookie = new HttpCookie("Name");
            nameCookie.Values["Name"] = txtName.Text;

            // Set cookie expiration to 10 seconds from now
            nameCookie.Expires = DateTime.Now.AddSeconds(10);
            Response.Cookies.Add(nameCookie);
        }

        // Button to delete the cookie
        protected void btnDelete_Click(object sender, EventArgs e)
        {
            // Check if the "Name" cookie exists
            HttpCookie nameCookie = Request.Cookies["Name"];

            // If the cookie exists, set its expiration to a past date, which will remove it
            if (nameCookie != null)
            {
                nameCookie.Expires = DateTime.Now.AddDays(-1);  // Set the expiration date to 1
day in the past
                Response.Cookies.Add(nameCookie);  // Re-add the cookie with the past expiration
date to delete it
            }

            // Optionally, alert the user that the cookie has been removed
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('Cookie removed.');",
true);
        }
    }
}
```

OUTPUT :



RESULT :

Thus the program was executed successfully.

EXERCISE NO : 13	WEBSITE
DATE :	

AIM :

To Demonstrate the Website using visual C# Web Application**.**

PROCEDURE:

Step1:
Open Visual Studio 2013, click "File" → "New" → "Project".
Step2:
In the New Project window, select Visual C# → "ASP.NET Web Application", name your project (e.g., MyWebApp), set the location, and click OK.
Step3:
Choose "Web Forms" as the project template and click OK to create the project.
Step4:
In Solution Explorer, right-click on the project, select "Add" → "New Item", choose "HTML Page", name it (e.g., index.html), and click Add.
Step5:
Again, right-click on the project in Solution Explorer, select "Add" → "New Item", choose "Style Sheet", name it styles.css, click Add, and then open it to add CSS code.
Step6:
Repeat the same process to add JavaScript: right-click on the project, select "Add" → "New Item", choose "JavaScript File", name it script.js, click Add, and open it to add JavaScript code.
Step7:
Open index.html and ensure the CSS and JavaScript files are linked correctly in the <head> section.
Step8:
Design your webpage in index.html using HTML elements and style them using styles.css.
Step9:
Implement JavaScript functionality in script.js for interactivity, such as handling button clicks.

Step10:
If server-side functionality is needed, modify the Home.aspx.cs file accordingly.
Step11:

Run the project by clicking F5 in Visual Studio; your default web browser will open and display the page.

HTML PAGE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Explore Paris - Travel Guide</title>
    <link rel="stylesheet" href="style.css">
    <script src="JavaScript1.js"></script>
</head>
<body>
    <div class="wrapper">
        <!-- Header Section -->
        <header>
            <h1>Welcome to Paris!</h1>
            <p>Discover the beauty, food, and culture of the City of Lights.</p>
        </header>

        <!-- Navigation Menu -->
        <nav>
            <ul>
                <li><a href="javascript:void(0);"
onclick="showSection('attractions')">Attractions</a></li>
                <li><a href="javascript:void(0);" onclick="showSection('food')">Food</a></li>
                <li><a href="javascript:void(0);" onclick="showSection('best-time')">Best
Time</a></li>
                <li><a href="javascript:void(0);" onclick="showSection('tips')">Travel
Tips</a></li>
            </ul>
        </nav>
```

```
<!-- Sections -->
<section id="home" class="section active">
  <div class="home-content">
    <h2>Welcome to Paris</h2>
    <p>
      Paris, the City of Lights, is known for its breathtaking landmarks, rich history, and romantic atmosphere.
      From the iconic Eiffel Tower to the artistic treasures of the Louvre, every corner of Paris tells a story.
      Stroll through charming streets, indulge in delicious pastries, and experience the vibrant culture that makes
      Paris one of the most beloved destinations in the world.
    </p>
  </div>
</section>

<section id="attractions" class="section">
  <h2>Top Attractions</h2>
  <p></p>
  <p>Paris is home to world-famous landmarks that tell the city's rich history and charm.</p>
  <p>From the iconic Eiffel Tower to the serene Seine River, every corner has a story to tell.</p>
  <p>Explore the artistic treasures of the Louvre, marvel at the grandeur of Notre-Dame,</p>
  <p>and take in breathtaking views from the Arc de Triomphe and Sacré-Cœur.</p>
  <div class="scroll-container">
    <div class="card"><img src="eiffel.JFIF" alt="Eiffel Tower"><p><b>Eiffel Tower</b></p></div>
    <div class="card"><img src="louvre.JFIF" alt="Louvre Museum"><p><b>Louvre Museum</b></p></div>
    <div class="card"><img src="notredame.JFIF" alt="Notre-Dame"><p><b>Notre-Dame</b></p></div>
    <div class="card"><img src="arc.JFIF" alt="Arc de Triomphe"><p><b>Arc de Triomphe</b></p></div>
```

```
            <div class="card"><img src="sacrecoeur.JFIF" alt="Sacré-Cœur"><p><b>Sacré-Cœur</b></p></div>

            <div class="card"><img src="seine.JFIF" alt="Seine River"><p><b>Seine River</b></p></div>

        </div>

    </section>


    <section id="food" class="section">

        <h2>Must-Try Foods</h2>

        <p></p>

        <p>French cuisine is famous worldwide for its exquisite flavors and delicate presentation.</p>

        <p>From buttery croissants to colorful macarons, every bite is a delight.</p>

        <p>Try the classic French baguette, indulge in delicious crêpes,</p>

        <p>and experience the unique taste of escargot and fine French cheese.</p>

        <div class="scroll-container">

            <div class="card"><img src="macaron.JFIF" alt="Macarons"><p><b>Macarons</b></p></div>

            <div class="card"><img src="crepes.JFIF" alt="Crepes"><p><b>Crêpes</b></p></div>

            <div class="card"><img src="baguette.JFIF" alt="Baguette"><p><b>French Baguette</b></p></div>

            <div class="card"><img src="croissant.JFIF" alt="Croissant"><p><b>Croissant</b></p></div>

            <div class="card"><img src="escargot.JFIF" alt="Escargot"><p><b>Escargot</b></p></div>

            <div class="card"><img src="cheese.JFIF" alt="French Cheese"><p><b>French Cheese</b></p></div>

        </div>

    </section>


    <section id="best-time" class="section">

        <h2>Best Time to Visit</h2>

        <p></p>

        <p>Paris offers a magical experience throughout the year, each season bringing its own charm.</p>
```

```
        <p>Spring blooms with cherry blossoms, while summer is perfect for open-air cafés.</p>

        <p>Autumn paints the city in golden hues, creating a cozy and romantic vibe.</p>

        <p>Winter lights up with Christmas markets, making it a festive wonderland.</p>

        <div class="scroll-container">

            <div class="card"><img src="paris-fall.JFIF" alt="Fall in Paris"><p><b>Fall (September-November)</b></p></div>

            <div class="card"><img src="paris-winter.JFIF" alt="Winter in Paris"><p><b>Winter (December-February)</b></p></div>

            <div class="card"><img src="paris-summer.JFIF" alt="Summer in Paris"><p><b>Summer (June-August)</b></p></div>

            <div class="card"><img src="paris-cherryblossom.JFIF" alt="Cherry Blossoms"><p><b>Cherry Blossoms</b></p></div>

            <div class="card"><img src="paris-xmas.JFIF" alt="Christmas in Paris"><p><b>Christmas Lights</b></p></div>

            <div class="card"><img src="paris-rainy.JFIF" alt="Rainy Paris"><p><b>Paris in the Rain</b></p></div>

        </div>

    </section>


    <section id="tips" class="section">

        <h2>Travel Tips</h2>

        <p></p>

        <p>To make the most of your Paris trip, plan ahead and book tickets in advance.</p>

        <p>Use the metro for fast and affordable travel across the city.</p>

        <p>Embrace local culture by enjoying a slow coffee at a Parisian café.</p>

        <p>Stay alert in crowded areas and keep your belongings secure at all times.</p>

        <div class="scroll-container">

            <div class="card"><img src="paris-metro.JFIF" alt="Paris Metro"><p><b>Use the Metro</b></p></div>

            <div class="card"><img src="cafe.JFIF" alt="Paris Café"><p><b>Enjoy Local Cafés</b></p></div>

            <div class="card"><img src="map.JFIF" alt="Paris Map"><p><b>Plan Your Route</b></p></div>

            <div class="card"><img src="tickets.JFIF" alt="Tickets"><p><b>Book Tickets Online</b></p></div>
```

```
            <div class="card"><img src="shopping.JFIF" alt="Shopping in
Paris"><p><b>Shop Smartly</b></p></div>
            <div class="card"><img src="security.JFIF" alt="Safety in Paris"><p><b>Stay
Safe</b></p></div>
        </div>
    </section>

    <!-- Footer -->
    <footer>
        <p>&copy; 2025 Explore Paris | Designed for Travel Lovers</p>
    </footer>

</body>
</html>
```

CSS PAGE:

```
/* Reset Styles */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: Arial, sans-serif;
}

/* Full Width */
body {
  width: 100%;
  overflow-x: hidden;
  height: 100vh;
  display: flex;
  flex-direction: column;
}
```

```
/* Wrapper */
.wrapper {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
}

/* Header */
header {
    background: #333;
    color: white;
    text-align: center;
    padding: 40px 20px;
}

/* Navigation */
nav {
    background: #444;
    padding: 10px 0;
    position: sticky;
    top: 0;
    z-index: 1000;
}

nav ul {
    display: flex;
    justify-content: center;
    list-style: none;
}

nav ul li {
    margin: 0 15px;
}
```

```
nav ul li a {
    color: white;
    text-decoration: none;
    padding: 10px 15px;
    transition: 0.3s;
}

nav ul li a:hover {
    background: #ff7e5f;
    border-radius: 5px;
}

/* Sections */
.section {
    display: none;
    padding: 40px 5% 20px;
    flex: 1;
    align-items: center;
    justify-content: space-between;
    gap: 20px;
    height: calc(100vh - 150px);
    font-size: 1.2rem;
    font-weight: bold;
    text-align: center;
    margin-bottom: 5px;
}

/* Show active section */
.section.active {
    display: flex;
}

/* Scrollable Image Grid */
```

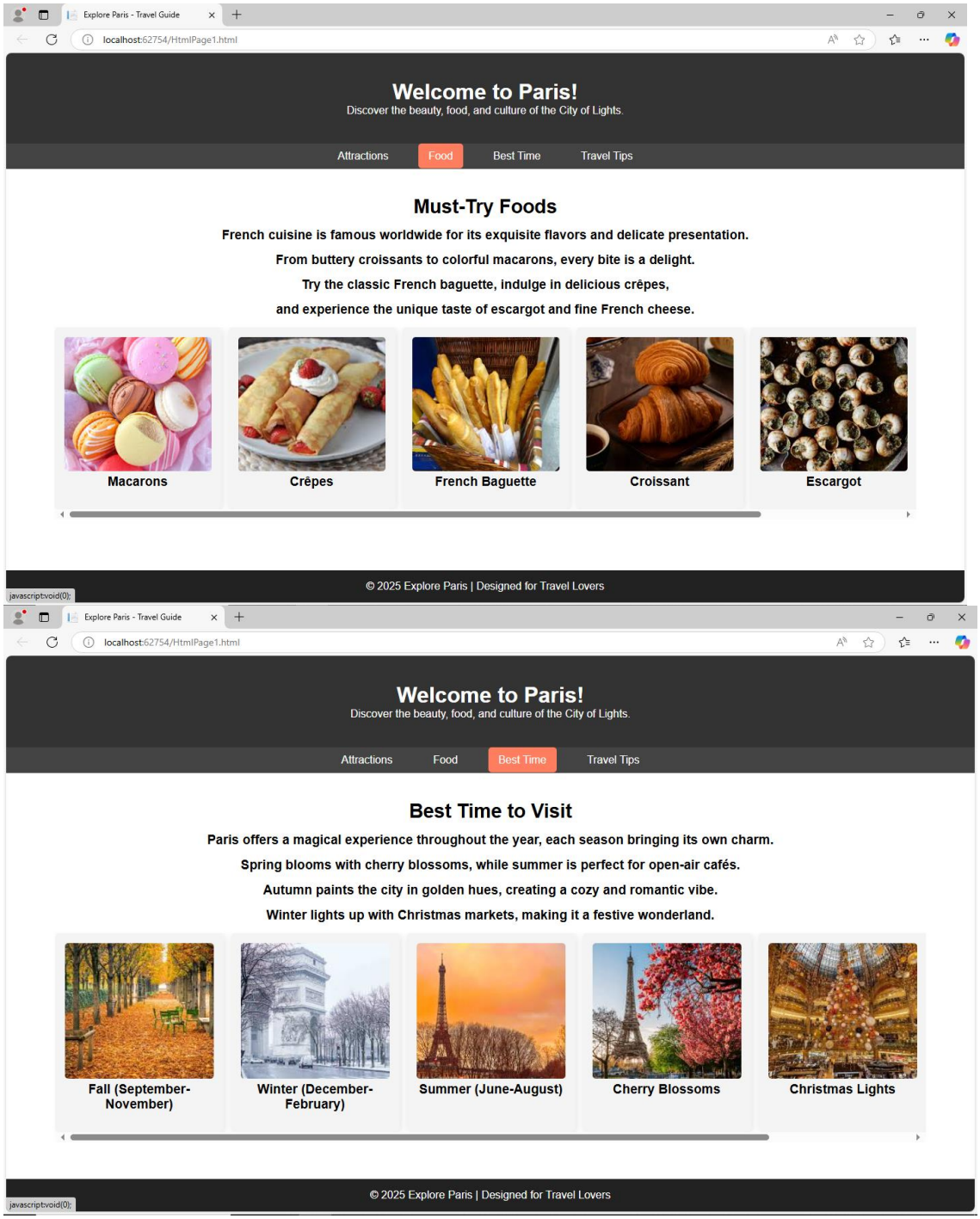
```
.scroll-container {
    display: flex;
    gap: 10px;
    overflow-x: auto;
    width: 100%;
    scroll-behavior: smooth;
    margin-bottom: 10px;
}

/* Card */
.card {
    flex: 0 0 auto;
    width: 250px; /* Increase width of cards */
    background: #f4f4f4;
    padding: 15px;
    border-radius: 8px;
    text-align: center;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s ease-in-out;
}

.card:hover {
    transform: scale(1.05);
}

/* Footer */
footer {
    background: #222;
    color: white;
    text-align: center;
    padding: 15px;
    margin-top: 10px;
}
```

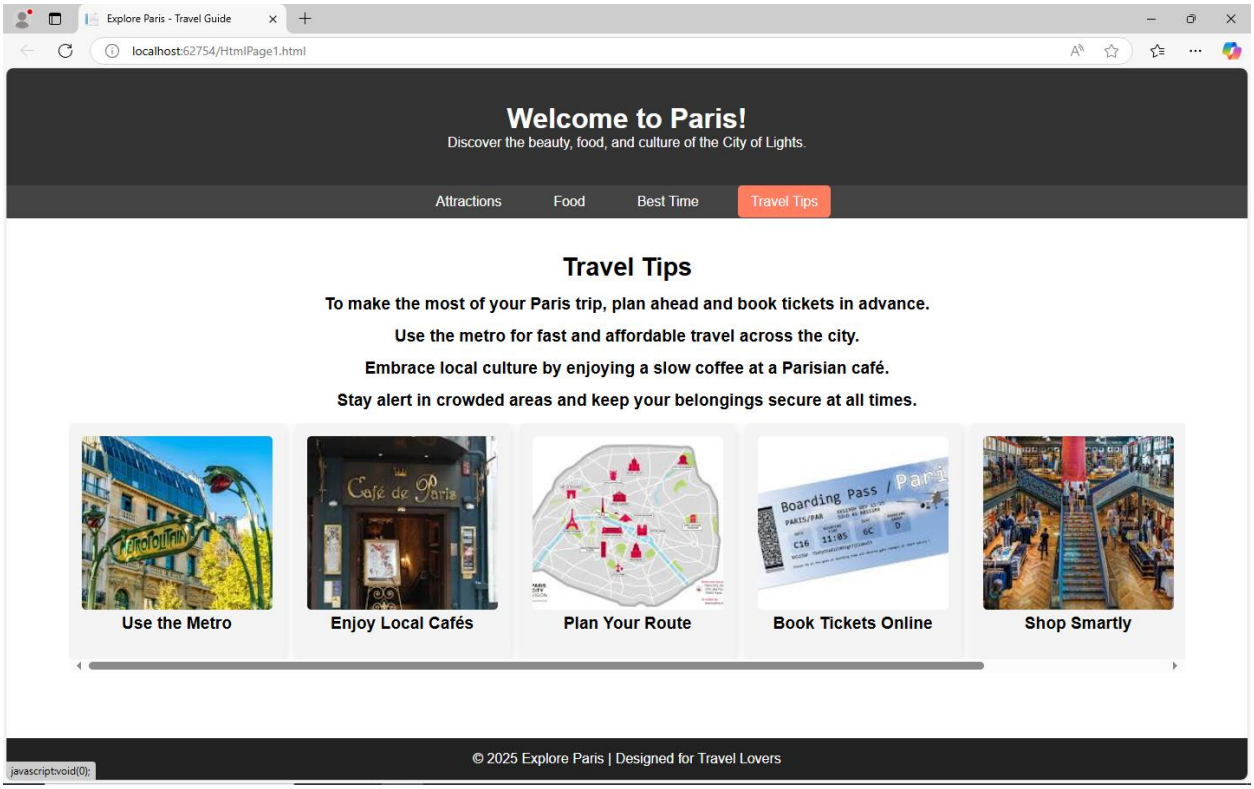
```
.card img {
    width: 100%; /* Ensures images fill the card */
    height: 200px; /* Increase height for larger images */
    object-fit: cover; /* Maintains aspect ratio without stretching */
    border-radius: 5px;
}
#attractions p {
    font-size: 1.2rem; /* Increase the font size */
    font-weight: bold; /* Make it stand out */
    text-align: center; /* Center the text */
    margin-bottom: 15px; /* Reduce white space */
}
#food p {
    font-size: 1.2rem; /* Increase the font size */
    font-weight: bold; /* Make it stand out */
    text-align: center; /* Center the text */
    margin-bottom: 15px; /* Reduce white space */
}
#best-time p {
    font-size: 1.2rem; /* Increase the font size */
    font-weight: bold; /* Make it stand out */
    text-align: center; /* Center the text */
    margin-bottom: 15px; /* Reduce white space */
}
#tips p {
    font-size: 1.2rem; /* Increase the font size */
    font-weight: bold; /* Make it stand out */
    text-align: center; /* Center the text */
    margin-bottom: 15px; /* Reduce white space */
}
```

JS PAGE:

```
function showSection(sectionId) {
    // Hide all sections
    document.querySelectorAll('.section').forEach(section => {
        section.style.display = 'none';
});

// Show the selected section
document.getElementById(sectionId).style.display = 'block';
}

// Show Attractions by default on page load
document.addEventListener("DOMContentLoaded", function () {
    showSection('attractions');
});
```

OUTPUT:


Explore Paris - Travel Guide
localhost:62754/HtmlPage1.html
Welcome to Paris!
Discover the beauty, food, and culture of the City of Lights.
Attractions
Food
Best Time
Travel Tips
Must-Try Foods
French cuisine is famous worldwide for its exquisite flavors and delicate presentation.
From buttery croissants to colorful macarons, every bite is a delight.
Try the classic French baguette, indulge in delicious crêpes,
and experience the unique taste of escargot and fine French cheese.
Macarons
Crêpes
French Baguette
Croissant
Escargot
© 2025 Explore Paris | Designed for Travel Lovers
javascript:void(0);
Explore Paris - Travel Guide
localhost:62754/HtmlPage1.html
Welcome to Paris!
Discover the beauty, food, and culture of the City of Lights.
Attractions
Food
Best Time
Travel Tips
Best Time to Visit
Paris offers a magical experience throughout the year, each season bringing its own charm.
Spring blooms with cherry blossoms, while summer is perfect for open-air cafés.
Autumn paints the city in golden hues, creating a cozy and romantic vibe.
Winter lights up with Christmas markets, making it a festive wonderland.
Fall (September-November)
Winter (December-February)
Summer (June-August)
Cherry Blossoms
Christmas Lights
© 2025 Explore Paris | Designed for Travel Lovers
javascript:void(0);



RESULT :

Thus the program was executed successfully.