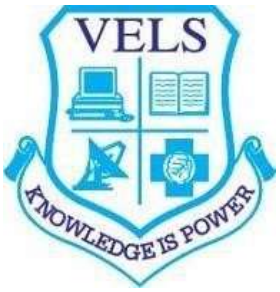
VELS

INSTITUTE OF SCIENCE, TECHNOLOGY & ADVANCED STUDIES (VISTAS)
(Deemed to be University Estd. u/s 3 of the UGC Act, 1956)
PALLAVARAM - CHENNAI
ACCREDITED BY NAAC WITH 'A' GRADE
Marching Beyond 30 Years Successfully



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

21PBAI61–PRACTICAL – .NET PROGRAMMING

LABORATORY

YEAR 2023- 2024

NAME OF THE STUDENT :

REGISTER NUMBER :

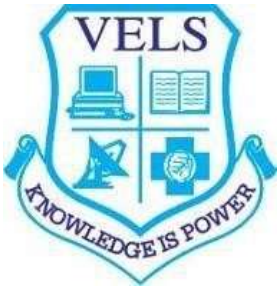
COURSE :

YEAR :

SEMESTER :

VELS

VELS INSTITUTE OF SCIENCE, TECHNOLOGY AND ADVANCED STUDIES
(VISTAS)
Deemed to be University Estd. U/S 3 of the UGC ACT, 1956
NAAC ACCREDITED WITH 'A' GRADE
PALLAVARAM, CHENNAI



BONAFIDE CERTIFICATE

Reg. No. | | | | | | | | |

This is to certify that the Bonafide Record of this Practical Work was completed by Mr./Ms…… of **B.TECH COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** in the .Net Programming Laboratory during the academic year of 2023 -2024 .

HEAD OF THE DEPARTMENT **STAFF-IN-CHARGE**

Submitted for the Practical Examination held on ……..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

S.NO	DATE	EXPERIMENT NAME	PAGE NO	MARKS	SIGNATURE
1A		C# Program to Count total Number of Alphabets, Digits, Special Characters in a Given String			
1B		C# Program to Count Total Number of Duplicates Elements in an array			
1C		C# Program to Implementation of Stack			
2A		C# Program to Calculate Hypotenuse of triangle using dynamic initialization of variables			
2B		C# Program to get input from the user and perform calculations			
2C		C# Program to Calculate the quadrant for the coordinates using if..els..ladder			
2D		C# Program to count total number of words in a string			
2E		C# Program to Check whether the alphabet is a vowel or not using switch..case Statement			
2F		C# Program to understand about for..each loop and strings			
3		Using ADO .Net Create a login page using controls			
4		Using ADO .NET to create a registration page with all controls.			

5		Establish Database Connection Using ADO.NET			
6		Using ASP.NET apply themes and different CSS in a form			
7		**Using ASP.NET** Create a web application with MVC framework			
8		Using ASP.Net Create Master Page			
9		Create a CURD Application using ASP.Net			

Ex.No:1(A)	C# Program to Count total Number of Alphabets, Digits, Special Characters in a Given String

AIM:

ALGORITHM:

PROGRAM:

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

namespace String7
{
   class Program
   {
      static void Main(string[] args)
      {
         string a;
         int alphabets, digits, specialCharacter, i,l;
            alphabets = digits = specialCharacter = i = 0;
            Console.WriteLine("Count the number of digits,alpabets,special characters in a string");
            Console.WriteLine("Enter the input String");
            a = Console.ReadLine();
            l = a.Length;
            while (i < l)
            {
               if ((a[i] >= 'a' && a[i] <= 'z') || (a[i] >= 'A' && a[i] <= 'Z'))
               {
                  alphabets++;
               }
               else if (a[i] >= '0' && a[i] <='9')
               {
                  digits++;
               }
               else
               {
                  specialCharacter++;
               }
               i++;
            }
            Console.WriteLine("The Number of Alphabets in a String is={0}", alphabets);
            Console.WriteLine("The Number of Digits in a String is={0}", digits);
            Console.WriteLine("The number of special Characters in  a string is={0}", specialCharacter);
            Console.ReadLine();


      }
   }
}
```

OUTPUT:

```
Count the number of digits,alpabets,special characters in a string
Enter the input String
Hello123!#
The Number of Alphabets in a String is=5
The Number of Digits in a String is=3
The number of special Characters in  a string is=2
```

RESULT:

Ex. No:1(B)	C# Program to Count Total Number of Duplicates Elements in an array

AIM:

ALGORITHM:

CODING:

```
using System;
public class Exp1
{
  public static void Main()
{
        int[] arr1 = new int[100];
        int[] arr2 = new int[100];
        int[] arr3 = new int[100];
  int s1, s2,mm=1,ctr=0;
  int i, j;
    Console.Write("\n\nCount total number of duplicate elements in an array:\n");
    Console.Write("-------------------------------------------------------\n");

    Console.Write("Input the number of elements to be stored in the array :");
         s1 = Convert.ToInt32(Console.ReadLine());

    Console.Write("Input {0} elements in the array :\n",s1);
    for(i=0;i<s1;i++)
        {
           Console.Write("element - {0} : ",i);
          arr1[i] = Convert.ToInt32(Console.ReadLine());
         }

              for(i=0;i<s1; i++)
    {
              arr2[i]=arr1[i];
              arr3[i]=0;
    }

      for(i=0;i<s1; i++)
    {
              for(j=0;j<s1;j++)
                    {
                          if(arr1[i]==arr2[j])
                          {
                          arr3[j]=mm;
                          mm++;
                          }
                    }
                    mm=1;
    }

 for(i=0; i<s1; i++)
  {
```

```
      if(arr3[i]==2){ctr++;}
    }
    Console.Write("The number of duplicate elements is: {0} \n", ctr);

          Console.Write("\n\n");
  }
}
```

OUTPUT:

```
Count total number of duplicate elements in an array:
------------------------------------------------------------
Input the number of elements to be stored in the array :5
Input 5 elements in the array :
element - 0 : 23
element - 1 : 34
element - 2 : 23
element - 3 : 1
element - 4 : 22
The number of duplicate elements is: 1



=== Code Execution Successful ===
```

RESULT:

Ex. No:1(C)	C# Program to Implementation of Stack

AIM:

PROCEDURE:

PROGRAM:

```
using System;
// Implementation of a Stack data structure
public class Stack
{
   private int[] items; // Array to hold stack elements
   private int top;     // Index representing the top of the stack
   // Constructor to initialize the stack with a specified size
   public Stack(int size)
   {
      items = new int[size]; // Initializing the array with the given size
      top = -1;              // Initializing top to -1, indicating an empty stack
   }
   // Method to check if the stack is empty
   public bool IsEmpty()
   {
      return top == -1; // Returns true if top is -1 (empty stack), otherwise false
   }
   // Method to check if the stack is full
   public bool IsFull()
   {
      return top == items.Length - 1; // Returns true if top is at the last index of the array (full stack)
   }
   // Method to push an element onto the stack
   public void Push(int item)
   {
      if (IsFull())
      {
         Console.WriteLine("Stack Full!"); // Displays a message if the stack is full
         return;
      }
      items[++top] = item; // Inserts the item at the incremented top index
   }
   // Method to pop an element from the stack
   public int Pop()
   {
      if (IsEmpty())
      {
         Console.WriteLine("Stack underflow"); // Displays a message if the stack is empty
         return -1;
      }
      return items[top--]; // Removes and returns the top element by decrementing top
   }
   // Method to peek at the top element of the stack without removing it
```

```
    public int Peek()
    {
        if (IsEmpty())
        {
            Console.WriteLine("Stack is empty"); // Displays a message if the stack is empty
            return -1;
        }
        return items[top]; // Returns the element at the top index without removing it
    }
    // Static method to display the stack elements
    public static void Display(Stack stack)
    {
        if (stack.IsEmpty())
        {
            Console.WriteLine("Stack is empty"); // Displays a message if the provided stack is empty
            return;
        }
        Console.WriteLine("\nStack elements:");
        for (int i = stack.top; i >= 0; i--)
        {
            Console.Write(stack.items[i] + " "); // Displays each element in the stack
        }
    }
}
// Main class to demonstrate the functionality of the Stack class
public class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Initialize a stack:");
        Stack stack = new Stack(5); // Creating a stack with a size of 5
        Console.WriteLine("Checking if stack is empty: " + stack.IsEmpty());
        Console.WriteLine("\nInput some elements onto the stack:");
        stack.Push(10);
        stack.Push(20);
        stack.Push(30);
        stack.Push(40);
        stack.Push(50);
        Stack.Display(stack); // Displaying the elements in the stack
        Console.WriteLine("\nTop element of the stack: " + stack.Peek()); // Displaying the top element
without removing it
        Console.WriteLine("\nChecking if stack is full: " + stack.IsFull());
        Console.WriteLine("\nPopping three elements from the stack:");
        Console.WriteLine(stack.Pop()); // Removing and displaying the popped elements
        Console.WriteLine(stack.Pop());
```

```
   Console.WriteLine(stack.Pop());
        Stack.Display(stack); // Displaying the remaining elements in the stack
        Console.WriteLine("\nTop element of the stack: " + stack.Peek()); // Displaying the top element without removing it
    }
}
```

OUTPUT:

```
Initialize a stack:
Checking if stack is empty: True

Input some elements onto the stack:

Stack elements:
50 40 30 20 10
Top element of the stack: 50

Checking if stack is full: True

Popping three elements from the stack:
50
40
30

Stack elements:
20 10
Top element of the stack: 20

=== Code Execution Successful ===
```

RESULT:

Ex. No:2(A)	C# Program to Calculate Hypotenuse of triangle using dynamic initialization of variables

AIM:

ALGORITHM:

PROGRAM:

```
using System;
class Program
{
   static void Main()
   {
      Console.WriteLine("Enter the length of the first side:");
      double side1 = Convert.ToDouble(Console.ReadLine());

      Console.WriteLine("Enter the length of the second side:");
      double side2 = Convert.ToDouble(Console.ReadLine());

      double hypotenuse = Math.Sqrt(Math.Pow(side1, 2) + Math.Pow(side2, 2));

      Console.WriteLine($"The hypotenuse of the triangle is: {hypotenuse}");
   }
}
```

OUTPUT:

```
Enter the length of the first side:
5
Enter the length of the second side:
7
The hypotenuse of the triangle is: 8.60232526704263

=== Code Execution Successful ===
```

RESULT:

Ex. No:2(B)	C# Program to get input from the user and perform calculations

AIM:

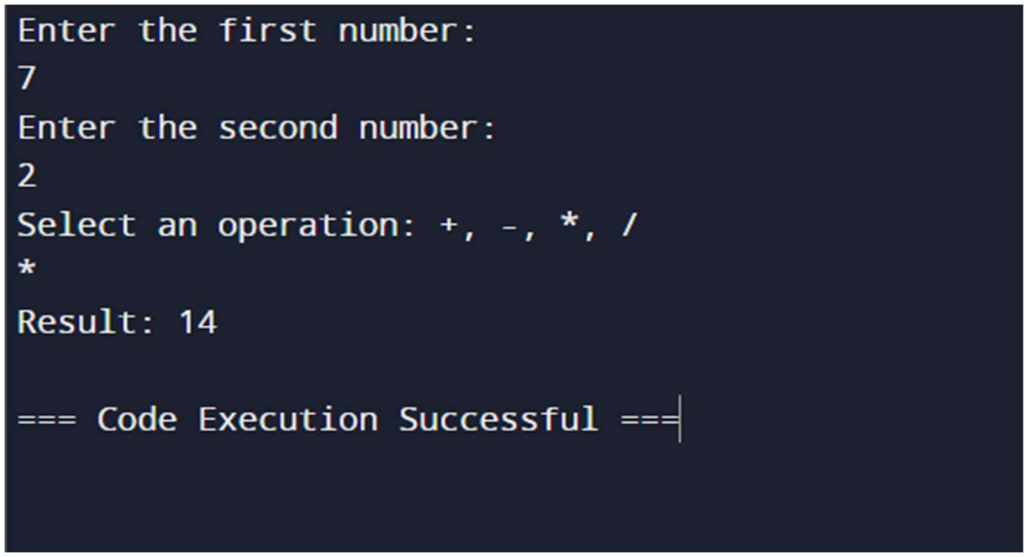
ALGORITHM:

PROGRAM:

```
using System;

class Calculator
{
   static void Main()
   {
      Console.WriteLine("Enter the first number:");
      double num1 = Convert.ToDouble(Console.ReadLine());

      Console.WriteLine("Enter the second number:");
      double num2 = Convert.ToDouble(Console.ReadLine());

      Console.WriteLine("Select an operation: +, -, *, /");
      char operation = Convert.ToChar(Console.ReadLine());

      double result = 0;

      switch (operation)
      {
         case '+':
            result = num1 + num2;
            break;
         case '-':
            result = num1 - num2;
            break;
         case '*':
            result = num1 * num2;
            break;
         case '/':
            if (num2 != 0)
            {
               result = num1 / num2;
            }
            else
            {
               Console.WriteLine("Cannot divide by zero.");
               return;
            }
            break;
         default:
            Console.WriteLine("Invalid operation.");
            return;
      }
```

```
        Console.WriteLine($"Result: {result}");
    }
}
```

OUTPUT:



RESULT:

Ex. No:2(C)	C# Program to Calculate the quadrant for the coordinates using if..els..ladder

AIM:

ALGORITHM:

PROGRAM:

```
using System;  // Importing the System namespace

public class Exercise9  // Declaration of the Exercise9 class
{
   public static void Main()  // Entry point of the program
   {
      int co1, co2;  // Declaration of integer variables co1 and co2 for X and Y coordinates
      Console.Write("\n\n");  // Printing new lines
      Console.Write("Find the quadrant in which the coordinate point lies:\n");  // Displaying the purpose
of the program
      Console.Write("-----------------------------------------------------");  // Displaying a separator
      Console.Write("\n\n");  // Printing new lines

      Console.Write("Input the value for X coordinate :");  // Prompting user to input the X coordinate
      co1 = Convert.ToInt32(Console.ReadLine());  // Reading the input X coordinate from the user

      Console.Write("Input the value for Y coordinate :");  // Prompting user to input the Y coordinate
      co2 = Convert.ToInt32(Console.ReadLine());  // Reading the input Y coordinate from the user

      if (co1 > 0 && co2 > 0)  // Checking if X and Y coordinates are both positive
         Console.Write("The coordinate point ({0} {1}) lies in the First quadrant.\n\n", co1, co2);  //
Printing a message for the first quadrant
      else if (co1 < 0 && co2 > 0)  // Checking if X coordinate is negative and Y coordinate is positive
         Console.Write("The coordinate point ({0} {1}) lies in the Second quadrant.\n\n", co1, co2);  //
Printing a message for the second quadrant
      else if (co1 < 0 && co2 < 0)  // Checking if both X and Y coordinates are negative
         Console.Write("The coordinate point ({0} {1}) lies in the Third quadrant.\n\n", co1, co2);  //
Printing a message for the third quadrant
      else if (co1 > 0 && co2 < 0)  // Checking if X coordinate is positive and Y coordinate is negative
         Console.Write("The coordinate point ({0} {1}) lies in the Fourth quadrant.\n\n", co1, co2);  //
Printing a message for the fourth quadrant
      else if (co1 == 0 && co2 == 0)  // Checking if both X and Y coordinates are zero
         Console.Write("The coordinate point ({0} {1}) lies at the origin.\n\n", co1, co2);  // Printing a
message if the coordinates are at the origin
   }
}
```

OUTPUT:

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

Input the value for X coordinate :4
Input the value for Y coordinate :7
The coordinate point (4 7) lies in the First quadrant.


=== Code Execution Successful ===
```

RESULT:

Ex. No:2(D)	C# Program to count total number of words in a string

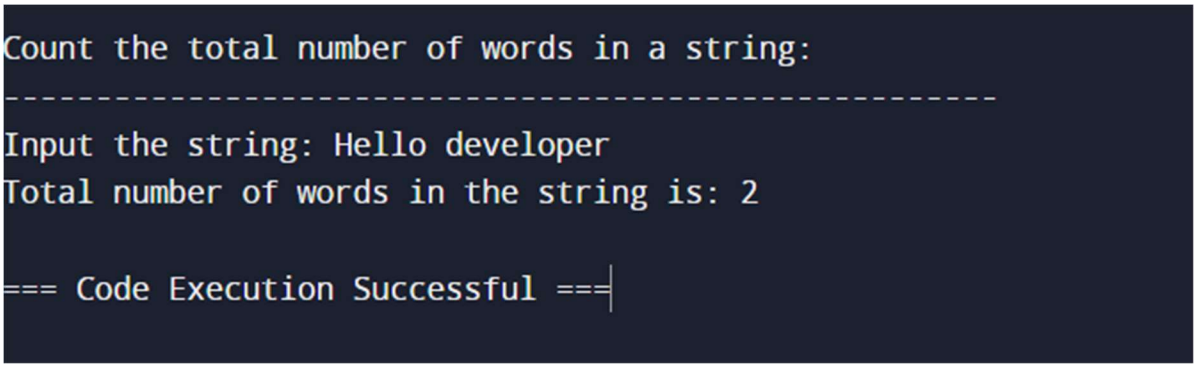
AIM:

ALGORITHM:

PROGRAM:

```
using System;
// Define the Exercise5 class
public class Exercise5
{
   // Main method - entry point of the program
   public static void Main()
   {
      string str; // Declare a string variable
      int l, wrd; // Declare variables for string traversal and word count
      // Prompt the user to count the total number of words in a string
      Console.Write("\n\nCount the total number of words in a string:\n");
      Console.Write("-------------------------------------------------\n");
      Console.Write("Input the string: ");
      str = Console.ReadLine(); // Read the user input string
      l = 0; // Initialize a variable for string traversal
      wrd = 1; // Initialize word count assuming at least one word exists
      /* Loop till the end of the string */
      while (l <= str.Length - 1)
      {
         /* Check whether the current character is whitespace, newline, or tab character */
         if (str[l] == ' ' || str[l] == '\n' || str[l] == '\t')
         {
            wrd++; // Increment word count if whitespace, newline, or tab character is found
         }
         l++; // Move to the next character in the string
      }
      // Display the total number of words in the string
      Console.Write("Total number of words in the string is: {0}\n", wrd);
   }}
```

OUTPUT:



RESULT:

Ex. No:2(E)	C# Program to Check whether the alphabet is a vowel or not using switch..case Statement

AIM:

ALGORITHM:

PROGRAM:

```
using System;  // Importing necessary namespaces
public class exercise16  // Declaration of the exercise16 class
{
   static void Main(string[] args)  // Entry point of the program
   {
      char ch;  // Declaration of a character variable to store the input
      Console.Write("\n\n");  // Printing new lines
      Console.Write("check whether the input alphabet is a vowel or not:\n");  // Displaying the purpose of the
program
      Console.Write("-----------------------------------------------------");  // Displaying a separator
      Console.Write("\n\n");
      Console.Write("Input an Alphabet (A-Z or a-z) : ");  // Prompting user to input an alphabet
      ch = Convert.ToChar(Console.ReadLine().ToLower());  // Reading the input and converting it to lowercase
      int i = ch;  // Converting the character to its corresponding ASCII value
      if (i >= 48 && i <= 57)  // Checking if the input is a number
      {
         Console.Write("You entered a number, Please enter an alphabet.");  // Prompting user to enter an alphabet
      }
      else
      {
         switch (ch)  // Switch statement to check for vowels
         {
            case 'a':
               Console.WriteLine("The Alphabet is vowel");  // Printing a message if 'a' is entered
               break;
            case 'i':
               Console.WriteLine("The Alphabet is vowel");  // Printing a message if 'i' is entered
               break;
            case 'o':
               Console.WriteLine("The Alphabet is vowel");  // Printing a message if 'o' is entered
               break;
            case 'u':
               Console.WriteLine("The Alphabet is vowel");  // Printing a message if 'u' is entered
               break;
            case 'e':
               Console.WriteLine("The Alphabet is vowel");  // Printing a message if 'e' is entered
               break;
            default:
               Console.WriteLine("The Alphabet is not a vowel");  // Printing a message for non-vowel characters
               break;
         }



  }
      Console.ReadKey();  // Waiting for a key press before closing the console window
   }
}
```

OUTPUT:

```
check whether the input alphabet is a vowel or not:
-----------------------------------------------------

Input an Alphabet (A-Z or a-z) : I
The Alphabet is vowel
```

RESULT:

Ex. No:2(F)	C# Program to understand about for..each loop and strings

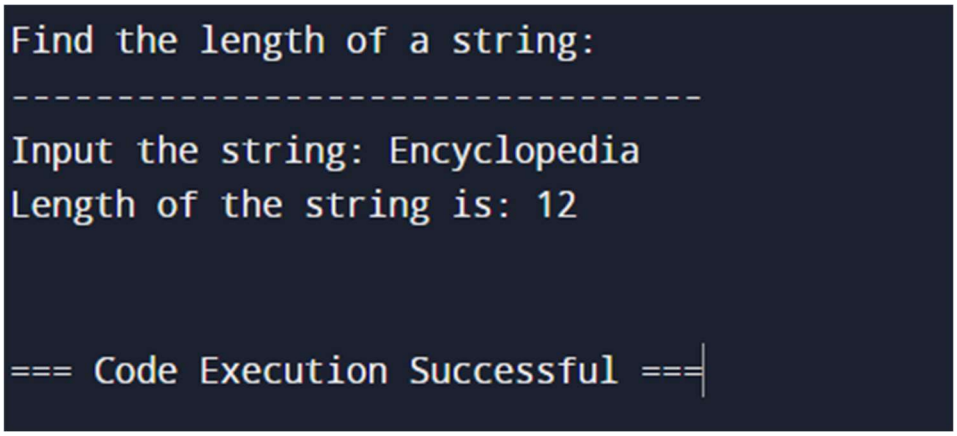
AIM:

PROCEDURE:

PROGRAM:

```
using System;
// Define the Exercise2 class
public class Exercise2
{
   // Main method - entry point of the program
   public static void Main()
   {
      string str; // Declare a string variable
      int l = 0; // Initialize a variable to store the length of the string

      // Prompt the user to find the length of a string
      Console.Write("\n\nFind the length of a string:\n");
      Console.Write("---------------------------------\n");

      // Request user input for a string
      Console.Write("Input the string: ");
      str = Console.ReadLine(); // Read the user input

      // Loop through each character in the string to calculate its length
      foreach (char chr in str)
      {
         l += 1; // Increment the length counter for each character encountered
      }

      // Display the length of the entered string
      Console.Write("Length of the string is: {0}\n\n", l);
   }
}
```

OUTPUT:



RESULT:

Ex. No: 3	Using ADO .Net Create a login page using controls

AIM:

PROCEDURE:

PROGRAM:

LoginForm.cs

```
using System;
using System.Windows.Forms;

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

      private void btnLogin_Click(object sender, EventArgs e)
      {
         string username = "admin";
         string password = "password";

         if (txtUsername.Text == username && txtPassword.Text == password)
         {
            MessageBox.Show("Login successful!");
         }
         else
         {
            MessageBox.Show("Login failed. Invalid username or password.");
         }
      }
   }
}
```

LoginForm.Designer.cs

```
namespace LoginForm
{
   partial class LoginForm
   {
      private System.ComponentModel.IContainer components = null;

      protected override void Dispose(bool disposing)
      {
```

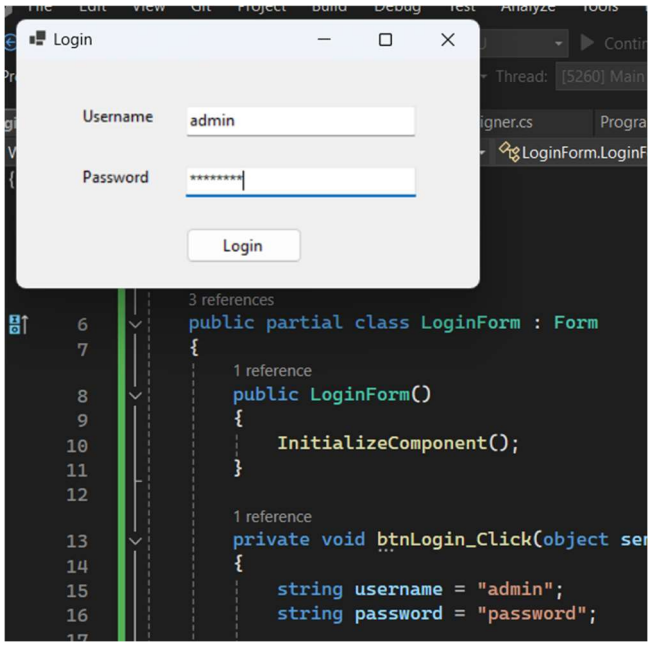
```
if (disposing && (components != null))
        {
           components.Dispose();
        }
        base.Dispose(disposing);
    }

    private void InitializeComponent()
    {
        this.lblUsername = new System.Windows.Forms.Label();
        this.lblPassword = new System.Windows.Forms.Label();
        this.txtUsername = new System.Windows.Forms.TextBox();
        this.txtPassword = new System.Windows.Forms.TextBox();
        this.btnLogin = new System.Windows.Forms.Button();
        this.SuspendLayout();
        //
        // lblUsername
        //
        this.lblUsername.AutoSize = true;
        this.lblUsername.Location = new System.Drawing.Point(40, 30);
        this.lblUsername.Name = "lblUsername";
        this.lblUsername.Size = new System.Drawing.Size(55, 13);
        this.lblUsername.TabIndex = 0;
        this.lblUsername.Text = "Username";
        //
        // lblPassword
        //
        this.lblPassword.AutoSize = true;
        this.lblPassword.Location = new System.Drawing.Point(40, 70);
        this.lblPassword.Name = "lblPassword";
        this.lblPassword.Size = new System.Drawing.Size(53, 13);
        this.lblPassword.TabIndex = 1;
        this.lblPassword.Text = "Password";
        //
        // txtUsername
        //
        this.txtUsername.Location = new System.Drawing.Point(110, 30);
        this.txtUsername.Name = "txtUsername";
        this.txtUsername.Size = new System.Drawing.Size(150, 20);
        this.txtUsername.TabIndex = 2;
        //
        // txtPassword
        //
        this.txtPassword.Location = new System.Drawing.Point(110, 70);
```

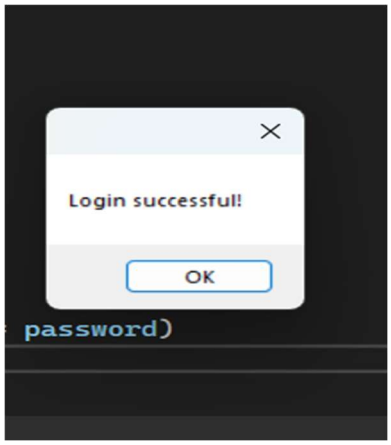
```
        this.txtPassword.Name = "txtPassword";
        this.txtPassword.PasswordChar = '*';
        this.txtPassword.Size = new System.Drawing.Size(150, 20);
        this.txtPassword.TabIndex = 3;
        //
        // btnLogin
        //
        this.btnLogin.Location = new System.Drawing.Point(110, 110);
        this.btnLogin.Name = "btnLogin";
        this.btnLogin.Size = new System.Drawing.Size(75, 23);
        this.btnLogin.TabIndex = 4;
        this.btnLogin.Text = "Login";
        this.btnLogin.UseVisualStyleBackColor = true;
        this.btnLogin.Click += new System.EventHandler(this.btnLogin_Click);
        //
        // LoginForm
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(300, 150);
        this.Controls.Add(this.btnLogin);
        this.Controls.Add(this.txtPassword);
        this.Controls.Add(this.txtUsername);
        this.Controls.Add(this.lblPassword);
        this.Controls.Add(this.lblUsername);
        this.Name = "LoginForm";
        this.Text = "Login";
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    private System.Windows.Forms.Label lblUsername;
    private System.Windows.Forms.Label lblPassword;
    private System.Windows.Forms.TextBox txtUsername;
    private System.Windows.Forms.TextBox txtPassword;
    private System.Windows.Forms.Button btnLogin;
  }
}
```

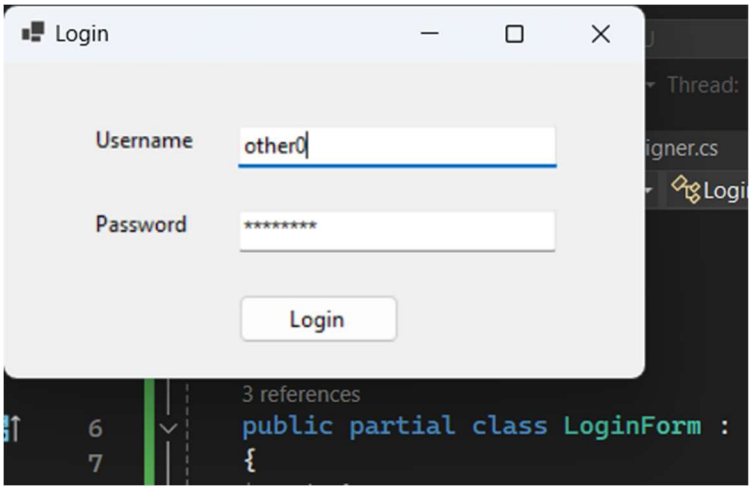
Program.cs

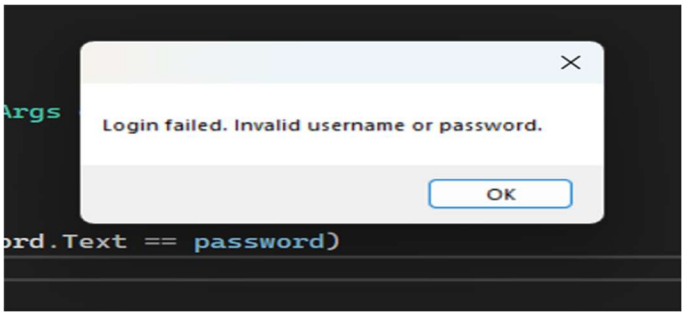
```
using System;
using System.Windows.Forms;

namespace LoginForm
{
   static class Program
   {
      [STAThread]
      static void Main()
      {
         Application.EnableVisualStyles();
         Application.SetCompatibleTextRenderingDefault(false);
         try
         {
            Application.Run(new LoginForm());
         }
         catch (Exception ex)
         {
            MessageBox.Show($"An error occurred: {ex.Message}", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
         }
      }
   }
}
```

OUTPUT:




Login successful!
OK
password)


Login
Username
other0
Password
****
Login
3 references
public partial class LoginForm :
{


Login failed. Invalid username or password.
OK
password)

RESULT:

Ex. No: 4	Using ADO .NET to create a registration page with all controls.

AIM:

PROCEDURE:

PROGRAM:
In RegistrationForm.cs

```
using System;
using System.Windows.Forms;
namespace RegistrationForm
{
   public partial class RegistrationForm : Form
   {
      public RegistrationForm()
      {
         InitializeComponent();
      }

      private void btnRegister_Click(object sender, EventArgs e)
      {
         string username = txtUsername.Text;
         string password = txtPassword.Text;
         string confirmPassword = txtConfirmPassword.Text;

         if (password != confirmPassword)
         {
            MessageBox.Show("Passwords do not match. Please try again.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
         }

         MessageBox.Show($"Registration successful!\nUsername: {username}\nPassword: {password}",
"Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
      }
   }
}
```

In DESIGNER.cs

```
namespace RegistrationForm
{
   partial class RegistrationForm
   {
      private System.ComponentModel.IContainer components = null;

      protected override void Dispose(bool disposing)
      {
         if (disposing && (components != null))
```

```
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

private void InitializeComponent()
{
    this.lblUsername = new System.Windows.Forms.Label();
    this.lblPassword = new System.Windows.Forms.Label();
    this.lblConfirmPassword = new System.Windows.Forms.Label();
    this.txtUsername = new System.Windows.Forms.TextBox();
    this.txtPassword = new System.Windows.Forms.TextBox();
    this.txtConfirmPassword = new System.Windows.Forms.TextBox();
    this.btnRegister = new System.Windows.Forms.Button();
    this.SuspendLayout();
    this.lblUsername.AutoSize = true;
    this.lblUsername.Location = new System.Drawing.Point(40, 30);
    this.lblUsername.Name = "lblUsername";
    this.lblUsername.Size = new System.Drawing.Size(55, 13);
    this.lblUsername.TabIndex = 0;
    this.lblUsername.Text = "Username";
    //
    // lblPassword
    //
    this.lblPassword.AutoSize = true;
    this.lblPassword.Location = new System.Drawing.Point(40, 70);
    this.lblPassword.Name = "lblPassword";
    this.lblPassword.Size = new System.Drawing.Size(53, 13);
    this.lblPassword.TabIndex = 1;
    this.lblPassword.Text = "Password";
    //
    // lblConfirmPassword
    //
    this.lblConfirmPassword.AutoSize = true;
    this.lblConfirmPassword.Location = new System.Drawing.Point(40, 110);
    this.lblConfirmPassword.Name = "lblConfirmPassword";
    this.lblConfirmPassword.Size = new System.Drawing.Size(91, 13);
    this.lblConfirmPassword.TabIndex = 2;
    this.lblConfirmPassword.Text = "Confirm Password";
    //
    // txtUsername
    //
    this.txtUsername.Location = new System.Drawing.Point(150, 30);
    this.txtUsername.Name = "txtUsername";
    this.txtUsername.Size = new System.Drawing.Size(150, 20);
```

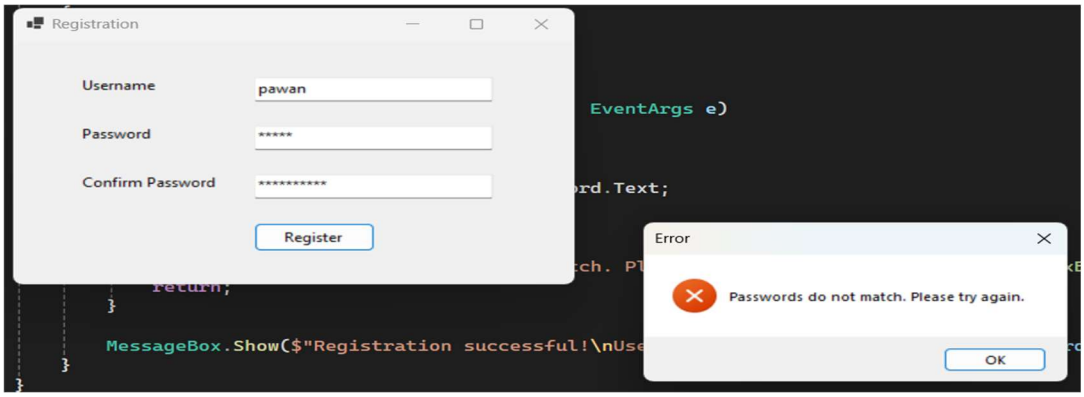
```
    this.txtUsername.TabIndex = 3;
    //
    // txtPassword
    //
    this.txtPassword.Location = new System.Drawing.Point(150, 70);
    this.txtPassword.Name = "txtPassword";
    this.txtPassword.PasswordChar = '*';
    this.txtPassword.Size = new System.Drawing.Size(150, 20);
    this.txtPassword.TabIndex = 4;
    //
    // txtConfirmPassword
    //
    this.txtConfirmPassword.Location = new System.Drawing.Point(150, 110);
    this.txtConfirmPassword.Name = "txtConfirmPassword";
    this.txtConfirmPassword.PasswordChar = '*';
    this.txtConfirmPassword.Size = new System.Drawing.Size(150, 20);
    this.txtConfirmPassword.TabIndex = 5;
    //
    // btnRegister
    //
    this.btnRegister.Location = new System.Drawing.Point(150, 150);
    this.btnRegister.Name = "btnRegister";
    this.btnRegister.Size = new System.Drawing.Size(75, 23);
    this.btnRegister.TabIndex = 6;
    this.btnRegister.Text = "Register";
    this.btnRegister.UseVisualStyleBackColor = true;
    this.btnRegister.Click += new System.EventHandler(this.btnRegister_Click);
    //
    // RegistrationForm
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(350, 200);
    this.Controls.Add(this.btnRegister);
    this.Controls.Add(this.txtConfirmPassword);
    this.Controls.Add(this.txtPassword);
    this.Controls.Add(this.txtUsername);
    this.Controls.Add(this.lblConfirmPassword);
    this.Controls.Add(this.lblPassword);
    this.Controls.Add(this.lblUsername);
    this.Name = "RegistrationForm";
    this.Text = "Registration";
    this.ResumeLayout(false);
    this.PerformLayout();

}
```

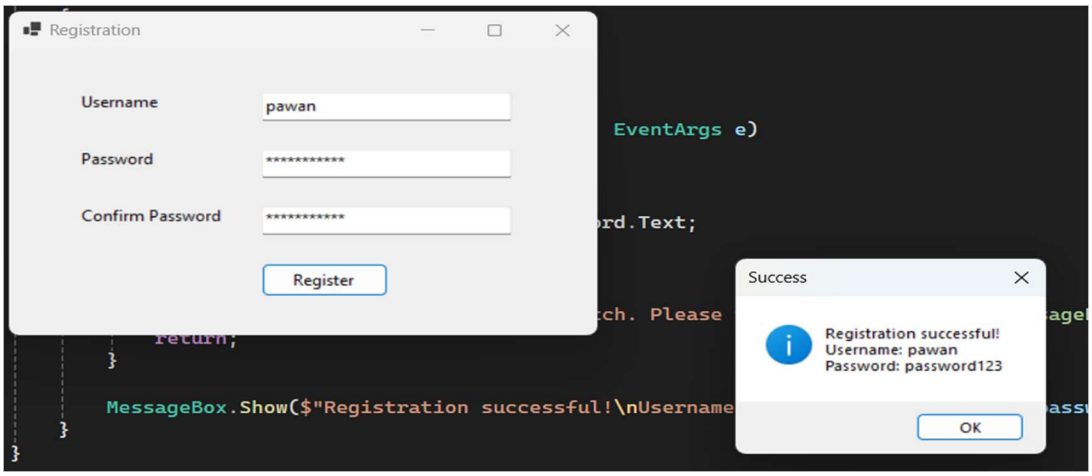
```
#endregion

        private System.Windows.Forms.Label lblUsername;
        private System.Windows.Forms.Label lblPassword;
        private System.Windows.Forms.Label lblConfirmPassword;
        private System.Windows.Forms.TextBox txtUsername;
        private System.Windows.Forms.TextBox txtPassword;
        private System.Windows.Forms.TextBox txtConfirmPassword;
        private System.Windows.Forms.Button btnRegister;
    }
}
```

In Program.cs

```
using System;
using System.Windows.Forms;

namespace RegistrationForm
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new RegistrationForm());
        }
    }
}
```

OUTPUT:





RESULT:

Ex. No: 5	Establish Database Connection Using ADO.NET

AIM:

ALGORITHM:

PROGRAM:

Configure Data Source:

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

namespace DatabaseConnectivity
{
   public partial class loginpage  System.Web.UI.Page
   {
protected void Page_Load(object sender, EventArgs e)
 {
  if(IsPostBack)
   {
            SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RegiConnectionString"].ConnectionString);
            conn.Open();
            string checkuser = "select count(*) from RegisterDataBase where
StudentName='"+TextBox1.Text+"'";
            SqlCommand cmd = new SqlCommand(checkuser, conn);
            int temp = Convert.ToInt32(cmd.ExecuteScalar().ToString());

            if (temp == 1)
            {
               Response.Write("Student Already Exis");
            }

            conn.Close();
        }

        }

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

```
SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RegiConnectionString"].ConnectionString);
                    conn.Open();
                    string insertQuery = "insert into
RegisterDataBase(StudentName,Passwords,EmailId,Department,College)values
(@studentname,@passwords,@emailid,@department,@college)";
                    SqlCommand cmd = new SqlCommand(insertQuery, conn);
                    cmd.Parameters.AddWithValue("@studentname", TextBox1.Text);
                    cmd.Parameters.AddWithValue("@passwords", TextBox2.Text);
                    cmd.Parameters.AddWithValue("@emailid", TextBox3.Text);
                    cmd.Parameters.AddWithValue("@department", TextBox4.Text);
                    cmd.Parameters.AddWithValue("@college", TextBox5.Text);
                    cmd.ExecuteNonQuery();

                    Response.Write("Student registeration Successfully!!!thank you");

                    conn.Close();

                }
                catch (Exception ex)
                {
                    Response.Write("error" + ex.ToString());
                }
            }
        }
```

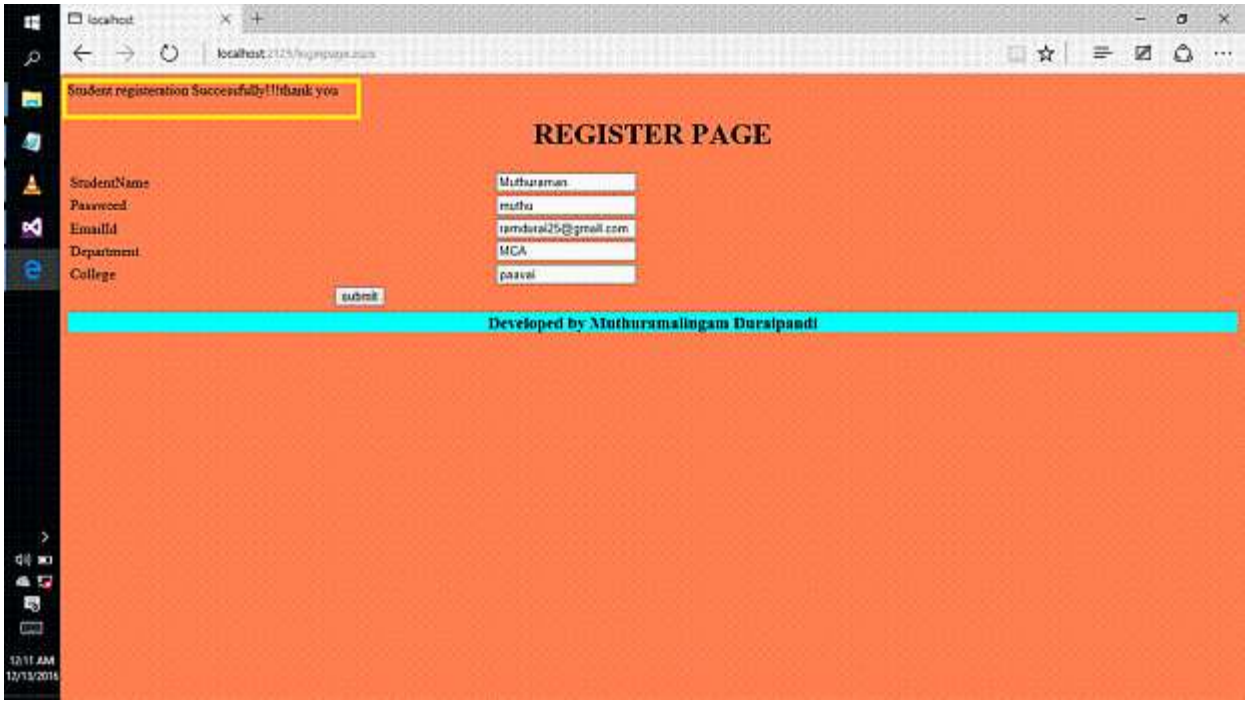
Loginpage.aspx code:

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

    <!DOCTYPE html>

    <html xmlns="http//www.w3.org/1999/xhtml">
    <head runat="server">
        <title></title>
        <link href="stylepage.css" type="text/css" rel="stylesheet" />
        <style type="text/css">
            .auto-style1 {
                width 100%;
    }
        </style>
    </head>
    <body>
        <form id="form1" runat="server">
        <div id="title">
```

```
        <h1>REGISTER PAGE</h1>
        </div>
           <div id ="teble"></div>
        <table class="auto-style1">
           <tr>
              <td>
                 <aspLabel ID="Label1" runat="server" Text="StudentName"></aspLabel></td>
              <td>
                 <aspTextBox ID="TextBox1" runat="server"></aspTextBox></td>
           </tr>
           <tr>
              <td>
                 <aspLabel ID="Label2" runat="server" Text="Password"></aspLabel></td>
              <td>
                 <aspTextBox ID="TextBox2" runat="server"></aspTextBox></td>
           </tr>
           <tr>
              <td>
                 <aspLabel ID="Label3" runat="server" Text="EmailId"></aspLabel></td>
              <td>
                 <aspTextBox ID="TextBox3" runat="server"></aspTextBox></td>
           </tr>
           <tr>
              <td>
                 <aspLabel ID="Label4" runat="server" Text="Department"></aspLabel></td>
              <td>
                 <aspTextBox ID="TextBox4" runat="server"></aspTextBox></td>
           </tr>
           <tr>
              <td>
                 <aspLabel ID="Label5" runat="server" Text="College"></aspLabel></td>
              <td>
                 <aspTextBox ID="TextBox5" runat="server"></aspTextBox></td>
           </tr>
        </table>
        <div id="button">
           <aspButton ID="Button1" runat="server" Text="submit" OnClick="Button1_Click"
BackColor="Yellow" />
        </div>
           <div id="sim"></div>
           <aspSqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$
ConnectionStringsRegiConnectionString %>" SelectCommand="SELECT * FROM
[RegisterDataBase]"></aspSqlDataSource>
```

```
            <div id="grid">
            <aspGridView ID="GridView1" runat="server" AllowPaging="True"
AllowSorting="True" AutoGenerateColumns="False" CellPadding="4" DataSourceID="SqlDataSource1"
ForeColor="#333333" GridLines="None">
                <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
                <Columns>
                   <aspBoundField DataField="Id" HeaderText="Id" SortExpression="Id" />
                   <aspBoundField DataField="StudentName" HeaderText="StudentName"
SortExpression="StudentName" />
                   <aspBoundField DataField="Passwords" HeaderText="Passwords"
SortExpression="Passwords" />
                   <aspBoundField DataField="EmailId" HeaderText="EmailId"
SortExpression="EmailId" />
                   <aspBoundField DataField="Department" HeaderText="Department"
SortExpression="Department" />
                   <aspBoundField DataField="College" HeaderText="College"
SortExpression="College" />
                </Columns>
                <EditRowStyle BackColor="#999999" />
                <FooterStyle BackColor="#5D7B9D" -Bold="True" ForeColor="White" />
                <HeaderStyle BackColor="#5D7B9D" -Bold="True" ForeColor="White" />
                <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
                <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
                <SelectedRowStyle BackColor="#E2DED6" -Bold="True" ForeColor="#333333" />
                <SortedAscendingCellStyle BackColor="#E9E7E2" />
                <SortedAscendingHeaderStyle BackColor="#506C8C" />
                <SortedDescendingCellStyle BackColor="#FFFDF8" />
                <SortedDescendingHeaderStyle BackColor="#6F8DAE" />
            </aspGridView>
         </div>

            <div id="last">
            <h3>Developed by
                  Muthuramalingam Duraipandi</h3>
           </div>
        </form>
        </body>
      </html>
```

OUTPUT:



RESULT:

Ex. No: 6	Using ASP.NET apply themes and different CSS in a form

AIM:

ALGORITHM:

PROGRAM:

Create a New ASP.NET Web Application:
Open Visual Studio and create a new ASP.NET Web Application project.

Create an ASP.NET Form:
Add a new ASP.NET Web Form (e.g., "Default.aspx") to your project.

Add Form Elements with CSS Classes:
In your ASP.NET form (Default.aspx), add form elements (e.g., textboxes, buttons) and apply CSS classes from your theme's CSS file using the CssClass attribute.

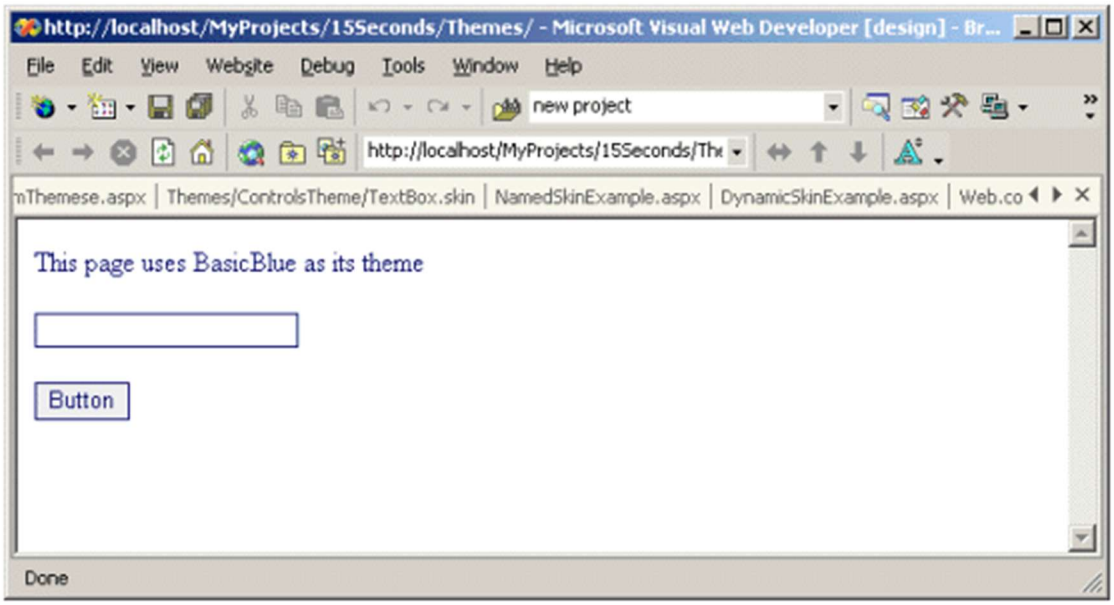
Handle Button Click Event:
In your code-behind file (Default.aspx.cs),

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="css_themes.Default" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Form</title>
    <link rel="stylesheet" href="App_Themes/MyTheme/styles.css" />
</head>
    <style>
        /* styles.css */

/* Blue background color */
.bg-blue {
    background-color: #3498db; /* Use hex color code or color name */
    color: #fff; /* Text color */
}

/* Green background color */
.bg-green {
    background-color: #2ecc71;
    color: #fff;
}

/* Yellow background color */
.bg-yellow {
    background-color: #f1c40f;
    color: #000;
```

```
}

/* Red background color */
.bg-red {
   background-color: #e74c3c;
   color: #fff;
}

   </style>
<body>
   <form id="myForm" runat="server">
      <div class="bg-blue">
         <label for="txtName">Name:</label>
         <asp:TextBox ID="txtName" runat="server" CssClass="input-text" />
      </div>
      <div class="bg-green">
         <label for="txtEmail">Email:</label>
         <asp:TextBox ID="txtEmail" runat="server" CssClass="input-text" />
      </div>
      <div class="bg-yellow">
         <asp:Button ID="btn" runat="server" Text="Submit" CssClass="btn-primary"
OnClick="btnSubmit_Click" />
      </div>
   </form>
</body>
</html>

==default.cs==
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace css_themes
{

   public partial class Default : System.Web.UI.Page
   {
      protected void btnSubmit_Click(object sender, EventArgs e)
      {
         string name = txtName.Text;
         string email = txtEmail.Text;

         // Perform validation
```

```
 if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(email))
        {
           // Save data to the database
           // Redirect to another page
           Response.Redirect("ThankYou.aspx");
        }
        else
        {
           // Display error message
           lblError.Text = "Please fill in all fields.";
        }
     }

     private class lblError
     {
        internal static string Text;
     }
  }
}
```

OUTPUT:



RESULT:

Ex. No: 7	Using ASP.NET Create a web application with MVC framework

AIM:

ALGORITHM:

PROGRAM:

HomeController.cs:

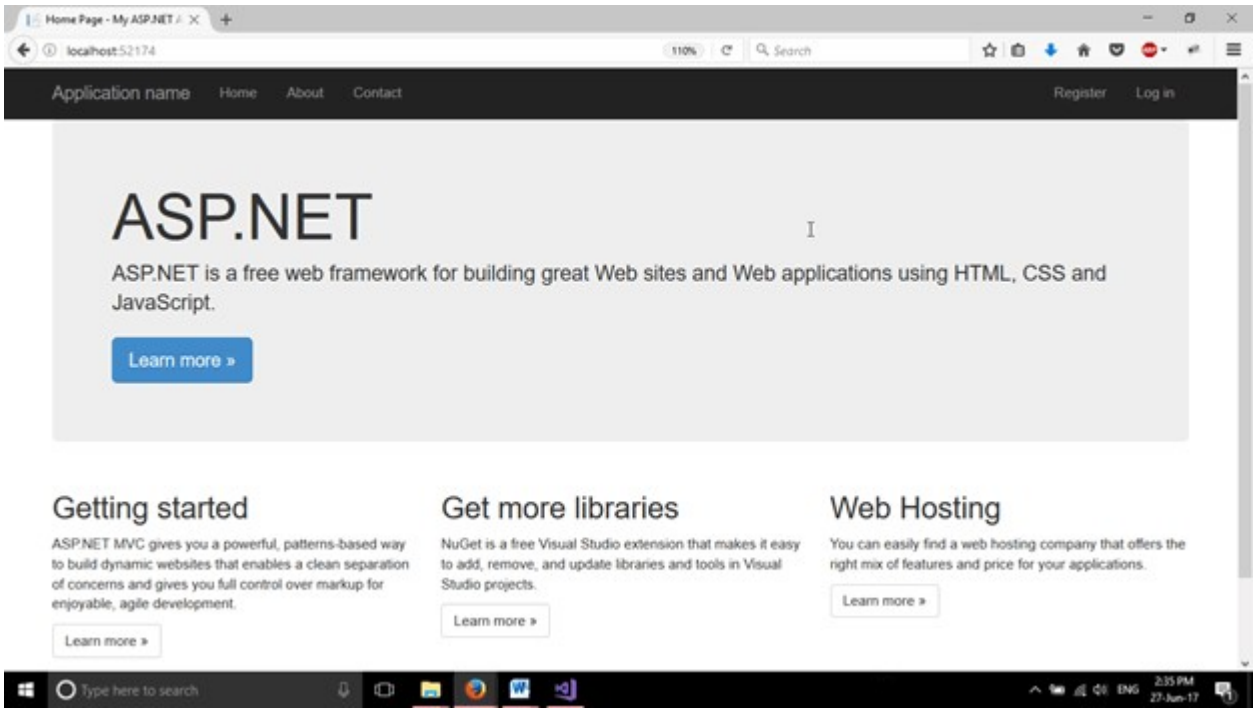
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace MvcApplicationDemo.Controllers
{
   public class HomeController : Controller
   {
      public ActionResult Index()
      {
         return View();
      }
      public ActionResult About()
      {
        ViewBag.Message = "Your application description page.";
         return View();
      }
      public ActionResult Contact()
      {
         ViewBag.Message = "Your contact page.";
         return View();
  }
   }
}
```

index.cshtml:

```
@{
    ViewBag.Title = "Home Page";
}
<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications
    using HTML, CSS and JavaScript.</p>
    <p><a href="https://asp.net" class="btn btn-primary btn-lg">Learn more</a></p>
</div>
<div class="row">
    <div class="col-md-4">
        <h2>Getting started</h2>
        <p>
            ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that
            enables a clean separation of concerns and gives you full control over markup
            for enjoyable, agile development.
        </p>
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301865">
        Learn more</a></p>
    </div>
    <div class="col-md-4">
        <h2>Get more libraries</h2>
        <p>NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries
        and tools in Visual Studio projects.</p>
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301866">
        Learn more</a></p>
    </div>
    <div class="col-md-4">
        <h2>Web Hosting</h2>
        <p>You can easily find a web hosting company that offers the right mix of features and price
        for your applications.</p>
```

```
        <p><a class="btn btn-default" href="https://go.microsoft.com/fwlink/?LinkId=301867">
            Learn more</a></p>
        </div>
    </div>
```

OUTPUT:



RESULT:

Ex. No: 8	Using ASP.Net Create Master Page

AIM:

ALGORITHM:

PROGRAM:

Coding behind Default.master page-

```
<%@ Master Language="VB" CodeFile="MasterPage.master.vb" Inherits="MasterPage" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <asp:ContentPlaceHolder id="head" runat="server">
   </asp:ContentPlaceHolder>
   <style type="text/css">
      .style1
      {
         color: #003399;
      }
      .style2
      {
         color: #000066;
      }
      .style3
      {
         color: #3333FF;
         font-weight: bold;
      }
   </style>
</head>
<body style="Verdana">
   <form id="form1" runat="server">
   <div style="font-family: Verdana; font-size: small; background-color: #D7EBFF; height: 235px;">
      <b style="border-style: solid; border-color: #999999; background-color: #CCCCCC;">
      <span class="style1">
      <br />  
      </span><span class="style2">A Simple Master Page    </span>
      <span class="style1">  
      </span></b><br />
      <br />
      This is the header section.<br /><br />
      <asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
          <p class="style3" style="border-style: ridge">
           Here goes the ContentPlaceHolder...</p>
      </asp:contentplaceholder>
      <br /><br />
      This is the footer section.</div>
   </form>
</body>
</html>
```

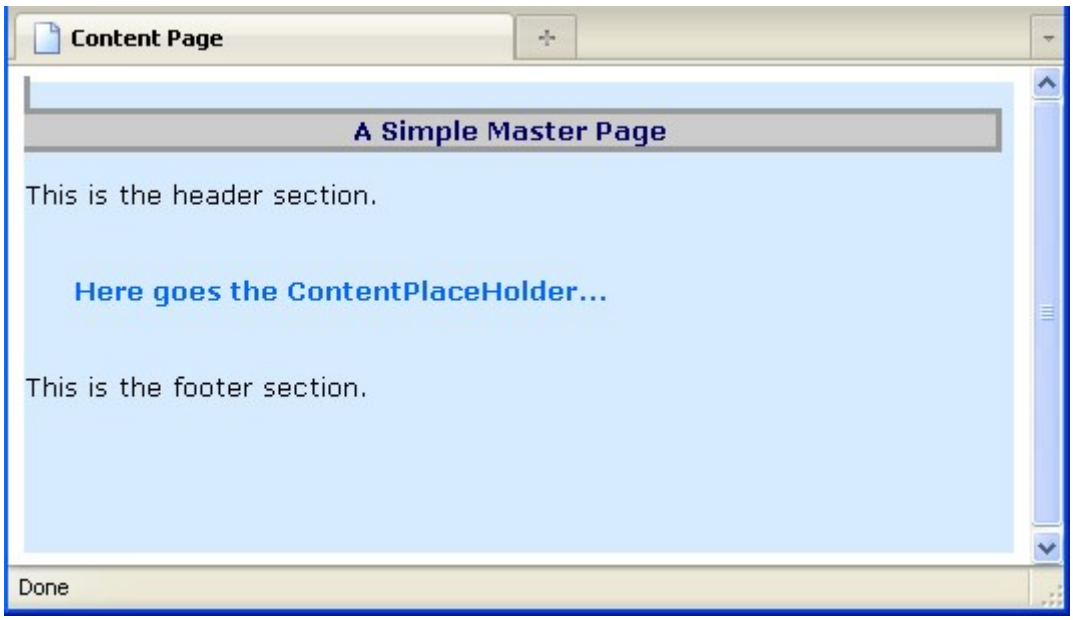
Coding behind Default.master.vb page

```
Imports System
Imports System.Data
Imports System.Configuration
Imports System.Collections
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Partial Public Class SiteTemplate
   Inherits System.Web.UI.MasterPage
   Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)
   End Sub
End Class
```

Coding behind Default.aspx page-

```
<%@ Page Language="vb" MasterPageFile="MasterPage.master" Title="Content Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
     <br />
          <span class="style4">Here goes the ContentPlaceHolder...</span><br />
</asp:Content>
<asp:Content ID="Content2" runat="server" contentplaceholderid="head">
   <style type="text/css">
   .style4
   {
      color: #0066FF;
      font-weight: bold;
   }
</style>
</asp:Content>
```

OUTPUT:



RESULT:

Ex. No: 9	Create a CURD Application using ASP.Net

AIM:

ALGORITHM:

PROGRAM:

Select EF Designer From database and click on next button.

```
public partial class Employee
{
      public int EmployeeId { get; set; }
       public string EmployeeName { get; set; }
       public Nullable<decimal> EmployeeSalary { get; set; }
       public string EmployeeCity { get; set; }
}
```

Design file code

```
@model IEnumerable<CrudOperationInMVC.Controllers.Employee>

@{
   ViewBag.Title = "Index";
}

<p>Index</p>

<p class="text-right">
   @Html.ActionLink("Create New", "Create")
</p>

@if (ViewBag.Message != null)
{

<p class="alert alert-success" role="alert">
   @ViewBag.Message
</p>
```

```
}


<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.EmployeeName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.EmployeeSalary)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.EmployeeCity)
        </th>
        <th></th>
    </tr>


    @if (Model != null && Model.Count() > 0)
    {
        foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.EmployeeName)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.EmployeeSalary)
                </td>
                <td>
```

```
                @Html.DisplayFor(modelItem => item.EmployeeCity)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id = item.EmployeeId }) |
                @Html.ActionLink("Details", "Detail", new { id = item.EmployeeId }) |
                @Html.ActionLink("Delete", "Delete", new { id = item.EmployeeId })
            </td>
        </tr>
    }
}
else
{
    <td colspan="4" class="text-center"><b>No Data Available . Please Add Data By CLick On Create Button</b></td>
}
</table>
```

Design for create view is below

```
@model CrudOperationInMVC.Controllers.Employee

@{
    ViewBag.Title = "Create";
}

<p>Create</p>

@if (ViewBag.Message != null)
{
<p class="alert alert-success" role="alert">
 @ViewBag.Message
</p>
}


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <p class="form-horizontal">
        <p>Employee</p>
        <hr />
```

```
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <p class="form-group">
            @Html.LabelFor(model => model.EmployeeName, htmlAttributes: new { @class = "control-label col-md-
2" })
            <p class="col-md-10">
                @Html.EditorFor(model => model.EmployeeName, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.EmployeeName, "", new { @class = "text-danger" })
            </p>
        </p>

        <p class="form-group">
            @Html.LabelFor(model => model.EmployeeSalary, htmlAttributes: new { @class = "control-label col-md-
2" })
            <p class="col-md-10">
                @Html.EditorFor(model => model.EmployeeSalary, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.EmployeeSalary, "", new { @class = "text-danger" })
            </p>
        </p>

        <p class="form-group">
            @Html.LabelFor(model => model.EmployeeCity, htmlAttributes: new { @class = "control-label col-md-2"
})
            <p class="col-md-10">
                @Html.EditorFor(model => model.EmployeeCity, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.EmployeeCity, "", new { @class = "text-danger" })
            </p>
        </p>

        <p class="form-group">
            <p class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </p>
        </p>
    </p>
}

<p>
    @Html.ActionLink("Back to List", "Index")
</p>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

OUTPUT:

RESULT: