Assignment – 1

1 WCP to display a message "HELLO INDIA" on console.

```
using System;

namespace HelloIndiaApp
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.WriteLine("HELLO INDIA");
      }
   }
}
```

2 WCP to perform arithmetic operation.

```
using System;

namespace ArithmeticOperationsApp
{
   class Program
   {
      static void Main(string[] args)
      {
         // Input two numbers
         Console.Write("Enter first number: ");
         double num1 = Convert.ToDouble(Console.ReadLine());

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

         // Perform operations
         double sum = num1 + num2;
         double difference = num1 - num2;
         double product = num1 * num2;
         double quotient = num1 / num2;

         // Output results
         Console.WriteLine("\n--- Results ---");
         Console.WriteLine("Sum: " + sum);
         Console.WriteLine("Difference: " + difference);
         Console.WriteLine("Product: " + product);
         Console.WriteLine("Quotient: " + quotient);
```

```
    }
  }
}
```

Output –

Enter first number: 12

Enter second number: 4

--- Results ---

Sum: 16

Difference: 8

Product: 48

Quotient: 3

3 WCP to find weather a year is leap or not.

```
using System;

class Program
{
   static void Main()
   {
      Console.Write("Enter a year: ");
      int year = int.Parse(Console.ReadLine());

      if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
         Console.WriteLine("Leap Year");
      else
         Console.WriteLine("Not a Leap Year");
   }
}
```

4 WCP to find the grades of a student according to their scores.

```
using System;

class Program
{
   static void Main()
   {
      Console.Write("Enter student's score (0-100): ");
      int score = int.Parse(Console.ReadLine());
```

```
        if (score >= 90)
            Console.WriteLine("Grade: A");
        else if (score >= 80)
            Console.WriteLine("Grade: B");
        else if (score >= 70)
            Console.WriteLine("Grade: C");
        else if (score >= 60)
            Console.WriteLine("Grade: D");
        else
            Console.WriteLine("Grade: F");
    }
}
```

5 WCP to find factorial using for, while, do…While loops.

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a number: ");
        int num = int.Parse(Console.ReadLine());

        // Factorial using for loop
        int factFor = 1;
        for (int i = 1; i <= num; i++)
            factFor *= i;
        Console.WriteLine("Factorial using for loop: " + factFor);

        // Factorial using while loop
        int factWhile = 1, j = 1;
        while (j <= num)
        {
            factWhile *= j;
            j++;
        }
        Console.WriteLine("Factorial using while loop: " + factWhile);

        // Factorial using do...while loop
        int factDoWhile = 1, k = 1;
        do
        {
            factDoWhile *= k;
            k++;
        } while (k <= num);
```

```
        Console.WriteLine("Factorial using do...while loop: " + factDoWhile);
    }
}
```

Output-

```
Enter a number: 5
Factorial using for loop: 120
Factorial using while loop: 120
Factorial using do...while loop: 120
```

6 WCP to print table from 1-10.

```
using System;

class Program
{
    static void Main()
    {
        for (int row = 1; row <= 10; row++)
        {
            for (int col = 1; col <= 10; col++)
            {
                Console.Write($"{row * col,4}");
            }
            Console.WriteLine();
        }

        Console.ReadKey();
    }
}
```

Output –

```
   1   2   3   4   5   6   7   8   9  10
   2   4   6   8  10  12  14  16  18  20
   3   6   9  12  15  18  21  24  27  30
   ...
  10  20  30  40  50  60  70  80  90 100
```

7 WCP to perform various string operations.

```
using System;

class Program
{
    static void Main()
    {
```

```
        Console.WriteLine("String Operations");
        Console.WriteLine("------------------");

        // 1. Concatenation
        string str1 = "Hello";
        string str2 = "World";
        string result = str1 + " " + str2;
        Console.WriteLine("Concatenation: " + result);

        // 2. Substring
        string str = "Hello World";
        string substr = str.Substring(6);
        Console.WriteLine("Substring: " + substr);

        // 3. IndexOf
        string str3 = "Hello World";
        int index = str3.IndexOf("World");
        Console.WriteLine("IndexOf: " + index);

        // 4. LastIndexOf
        string str4 = "Hello World Hello";
        int lastIndex = str4.LastIndexOf("Hello");
        Console.WriteLine("LastIndexOf: " + lastIndex);

        // 5. Replace
        string str5 = "Hello World";
        string replaced = str5.Replace("World", "Universe");
        Console.WriteLine("Replace: " + replaced);

        // 8. ToUpper
        string str8 = "hello world";
        string upper = str8.ToUpper();
        Console.WriteLine("ToUpper: " + upper);

        // 9. ToLower
        string str9 = "HELLO WORLD";
        string lower = str9.ToLower();
        Console.WriteLine("ToLower: " + lower);

        // 10. Contains
        string str10 = "Hello World";
        bool contains = str10.Contains("World");
        Console.WriteLine("Contains: " + contains);
    }
}
```

Output –

```
String Operations
-----------------
Concatenation: Hello World
Substring: World
IndexOf: 6
LastIndexOf: 13
Replace: Hello Universe
ToUpper: HELLO WORLD
ToLower: hello world
Contains: True
```

8 WCP to identify choice of user using select…Case.

```
using System;

class Program
{
   static void Main()
   {
      Console.WriteLine("Menu:");
      Console.WriteLine("1. Say Hello");
      Console.WriteLine("2. Show Date");
      Console.WriteLine("3. Exit");
      Console.Write("Enter your choice (1-3): ");

      int choice = int.Parse(Console.ReadLine());

      switch (choice)
      {
         case 1:
            Console.WriteLine("Hello, User!");
            break;
         case 2:
            Console.WriteLine("Today's Date: " + DateTime.Now.ToShortDateString());
            break;
         case 3:
            Console.WriteLine("Exiting... Goodbye!");
            break;
         default:
            Console.WriteLine("Invalid choice.");
            break;
      }
   }
}
```

Ouput –

```
Menu:
```

```
1. Say Hello
2. Show Date
3. Exit
Enter your choice (1-3): 2
Today's Date: 4/25/2025
```

9 WCP to declare & initialize various types of variables.

```
using System;

class Program
{
   static void Main()
   {
      // Integer types
      int age = 25;
      long population = 7800000000;
      short temp = -10;
      byte level = 255;

      // Floating point types
      float height = 5.9f;
      double weight = 65.5;
      decimal price = 199.99m;

      // Character and string
      char grade = 'A';
      string name = "Amit";

      // Boolean
      bool isActive = true;

      // Display all variables
      Console.WriteLine("Integer (int): " + age);
      Console.WriteLine("Long: " + population);
      Console.WriteLine("Short: " + temp);
      Console.WriteLine("Byte: " + level);

      Console.WriteLine("Float: " + height);
      Console.WriteLine("Double: " + weight);
      Console.WriteLine("Decimal: " + price);

      Console.WriteLine("Char: " + grade);
      Console.WriteLine("String: " + name);

      Console.WriteLine("Boolean: " + isActive);
   }
```

```
}
```

Output –

```
Integer (int): 25
Long: 7800000000
Short: -10
Byte: 255
Float: 5.9
Double: 65.5
Decimal: 199.99
Char: A
String: Amit
Boolean: True
```

10 WCP to find the largest number among three numbers.

```
using System;

class Program
{
    static void Main()
    {
        int a = 10, b = 25, c = 15;
        int largest;

        if (a > b && a > c)
            largest = a;
        else if (b > c)
            largest = b;
        else
            largest = c;

        Console.WriteLine($"The largest number among {a}, {b}, and {c} is:
{largest}");
    }
}
```

11 WCP to find maximum element in array a of size n.

```
using System;

class Program
{
    static void Main()
```

```
    {
        int[] a = { 15, 42, 3, 99, 23, 56 }; // Array of size n
        int max = a[0];

        for (int i = 1; i < a.Length; i++)
        {
            if (a[i] > max)
                max = a[i];
        }

        Console.WriteLine("Maximum element in array: " + max);
    }
}
```

Output – Maximum element in array: 99

12 WCP to find minimum element in array a of size n.

```
using System;

class Program
{
    static void Main()
    {
        int[] a = { 15, 42, 3, 99, 23, 56 }; // Array of size n
        int min = a[0];

        for (int i = 1; i < a.Length; i++)
        {
            if (a[i] < min)
                min = a[i];
        }

        Console.WriteLine("Minimum element in array: " + min);
    }
}
```

Output – Minimum element in array: 3

13 WCP to find LCM of two numbers.

```
using System;

class Program
{
    static void Main()
```

```
    {
        int a = 12, b = 18;
        int max = (a > b) ? a : b;

        while (true)
        {
            if (max % a == 0 && max % b == 0)
            {
                Console.WriteLine($"LCM of {a} and {b} is: {max}");
                break;
            }
            max++;
        }
    }
}
```

Output - LCM of 12 and 18 is: 36

14 WCP to find the area of a circle.

```
using System;

class Program
{
    static void Main()
    {
        double radius = 7;
        double area = Math.PI * radius * radius;

        Console.WriteLine($"Radius: {radius}");
        Console.WriteLine($"Area of Circle: {area}");
    }
}
```

Output –

Radius: 7

Area of Circle: 153.93804002589985

15 WCP to convert decimal to binary number.

```
using System;

class Program
{
```

```
    static void Main()
    {
        int decimalNumber = 25;
        string binary = Convert.ToString(decimalNumber, 2);

        Console.WriteLine($"Decimal: {decimalNumber}");
        Console.WriteLine($"Binary: {binary}");
    }
}
```

Output –

Decimal: 25

Binary: 11001

16 WCP to swap four numbers without using fifth variable.

```
using System;

class Program
{
    static void Main()
    {
        int a = 10, b = 20, c = 30, d = 40;

        Console.WriteLine($"Before Swap:\na = {a}, b = {b}, c = {c}, d = {d}");

        // Swapping in a circular fashion: a → b, b → c, c → d, d → a
        a = a + b + c + d;
        d = a - (b + c + d);
        c = a - (b + c + d);
        b = a - (b + c + d);
        a = a - (b + c + d);

        Console.WriteLine($"\nAfter Swap:\na = {a}, b = {b}, c = {c}, d = {d}");
    }
}
```

Output –

Before Swap:

a = 10, b = 20, c = 30, d = 40

After Swap:

a = 20, b = 30, c = 40, d = 10

17 WCP to sort an array.

```
using System;

class Program
{
   static void Main()
   {
      int[] arr = { 42, 15, 8, 23, 4, 16 };

      Console.WriteLine("Original Array:");
      foreach (int num in arr)
         Console.Write(num + " ");

      Array.Sort(arr); // Built-in sort

      Console.WriteLine("\n\nSorted Array (Ascending):");
      foreach (int num in arr)
         Console.Write(num + " ");
   }
}
```

Output –

Original Array:

42 15 8 23 4 16

Sorted Array (Ascending):

4 8 15 16 23 42

18 WCP to implement function.

```
using System;

class Program
{
   // Function to add two numbers
   static int Add(int a, int b)
   {
      return a + b;
   }
```

```
    static void Main()
    {
        int x = 10, y = 20;
        int sum = Add(x, y); // Calling the function

        Console.WriteLine($"Sum of {x} and {y} is: {sum}");
    }
}
```

Output –

Sum of 10 and 20 is: 30

19 WCP to print

```
*

* *

* * *

* * * *
```

```
using System;

class Program
{
    static void Main()
    {
        for (int i = 1; i <= 4; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                Console.Write("* ");
            }
            Console.WriteLine();
        }
    }
}
```

20 WCP to print

```
1

12

123

1234
```

```
using System;

class Program
{
   static void Main()
   {
      for (int i = 1; i <= 4; i++)
      {
         for (int j = 1; j <= i; j++)
         {
            Console.Write(j);
         }
         Console.WriteLine();
      }
   }
}
```

21 WCP to print

```
   1
  21
 321
4321
```

```
using System;

class Program
{
   static void Main()
   {
      int rows = 4;

      for (int i = 1; i <= rows; i++)
      {
         // Print leading spaces
         for (int space = 1; space <= rows - i; space++)
         {
            Console.Write(" ");
         }

         // Print decreasing numbers
         for (int num = i; num >= 1; num--)
         {
            Console.Write(num);
```

```
        }

        Console.WriteLine();
    }

    Console.ReadKey(); // Wait for key press
  }
}
```

22 WCP to print

```
   1
  22
 333
4444
```

```
using System;

class Program
{
  static void Main()
  {
    int rows = 4;

    for (int i = 1; i <= rows; i++)
    {
      // Print leading spaces
      for (int space = 1; space <= rows - i; space++)
      {
        Console.Write(" ");
      }

      // Print repeating numbers
      for (int num = 1; num <= i; num++)
      {
        Console.Write(i);
      }

      Console.WriteLine();
    }

    Console.ReadKey(); // Wait for key press
  }
}
```

23 WCP to print

```
using System;

class Program
{
   static void Main()
   {
      for (int i = 1; i <= 4; i++)
      {
         for (int j = 1; j <= i; j++)
         {
            Console.Write(Math.Pow(j, 3) + " ");
         }
         Console.WriteLine();
      }
   }
}
```

##

Assignment – 2

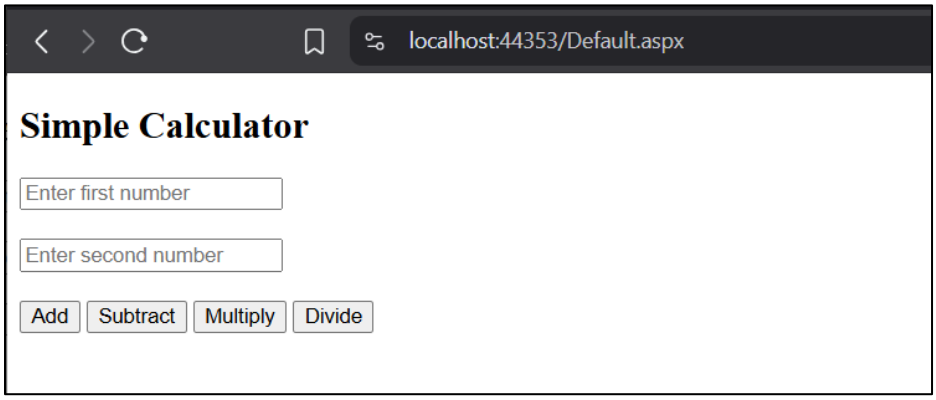
1 WWP to implement calculator in ASP.net with C#.

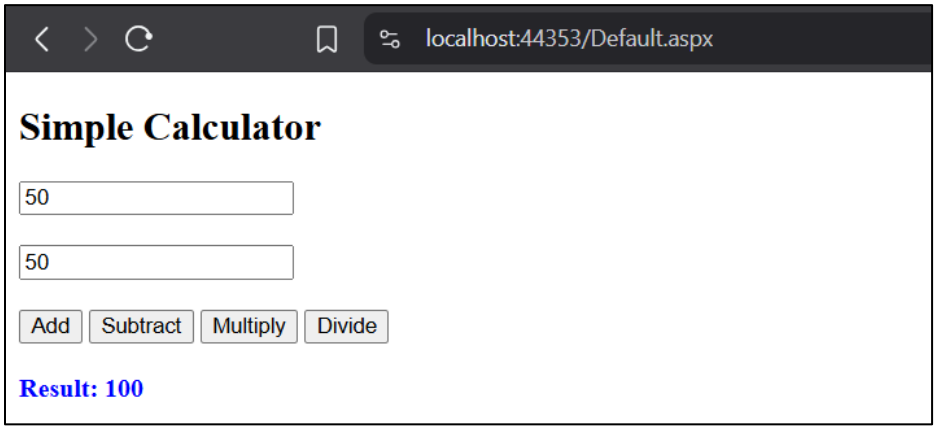
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Calculator.aspx.cs"
Inherits="Calculator" %>

<!DOCTYPE html>
<html>
<head>
   <title>Simple Calculator</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Simple Calculator</h2>

      <asp:TextBox ID="txtNum1" runat="server" Placeholder="Enter first number"
/><br /><br />
```

```
        <asp:TextBox ID="txtNum2" runat="server" Placeholder="Enter second
number" /><br /><br />

        <asp:Button ID="btnAdd" runat="server" Text="Add" OnClick="btnAdd_Click"
/>
        <asp:Button ID="btnSubtract" runat="server" Text="Subtract"
OnClick="btnSubtract_Click" />
        <asp:Button ID="btnMultiply" runat="server" Text="Multiply"
OnClick="btnMultiply_Click" />
        <asp:Button ID="btnDivide" runat="server" Text="Divide"
OnClick="btnDivide_Click" /><br /><br />

        <asp:Label ID="lblResult" runat="server" Font-Bold="true" ForeColor="Blue" />
    </form>
</body>
</html>
```

```
using System;

public partial class Calculator : System.Web.UI.Page
{
    protected void btnAdd_Click(object sender, EventArgs e)
    {
        int a = int.Parse(txtNum1.Text);
        int b = int.Parse(txtNum2.Text);
        lblResult.Text = "Result: " + (a + b);
    }

    protected void btnSubtract_Click(object sender, EventArgs e)
    {
        int a = int.Parse(txtNum1.Text);
        int b = int.Parse(txtNum2.Text);
        lblResult.Text = "Result: " + (a - b);
    }

    protected void btnMultiply_Click(object sender, EventArgs e)
    {
        int a = int.Parse(txtNum1.Text);
        int b = int.Parse(txtNum2.Text);
        lblResult.Text = "Result: " + (a * b);
    }

    protected void btnDivide_Click(object sender, EventArgs e)
    {
        double a = double.Parse(txtNum1.Text);
        double b = double.Parse(txtNum2.Text);

        if (b != 0)
```

```
            lblResult.Text = "Result: " + (a / b);
        else
            lblResult.Text = "Cannot divide by zero!";
    }
}
```



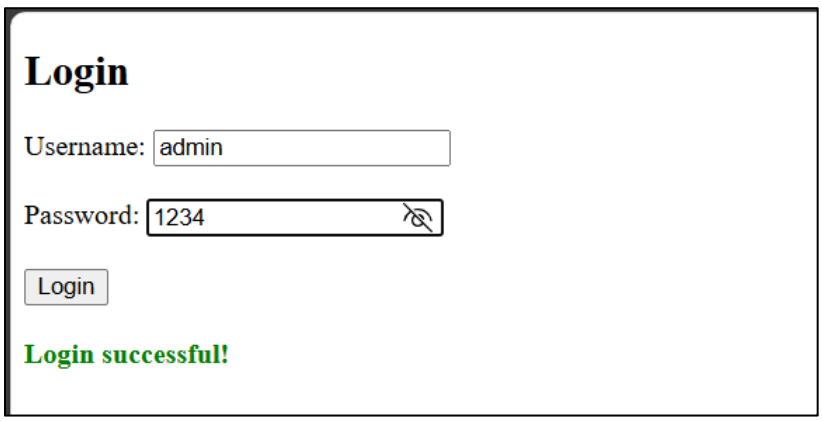


2 WWP to implement login form in ASP.net which field name are:-Username, Password.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="Login" %>

<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Login</h2>

        <asp:Label ID="lblUsername" runat="server" Text="Username: " />
        <asp:TextBox ID="txtUsername" runat="server" /><br /><br />
```

```
        <asp:Label ID="lblPassword" runat="server" Text="Password: " />
        <asp:TextBox ID="txtPassword" runat="server" TextMode="Password" /><br
/><br />

        <asp:Button ID="btnLogin" runat="server" Text="Login"
OnClick="btnLogin_Click" /><br /><br />

        <asp:Label ID="lblMessage" runat="server" Font-Bold="true" ForeColor="Red"
/>
    </form>
</body>
</html>
```

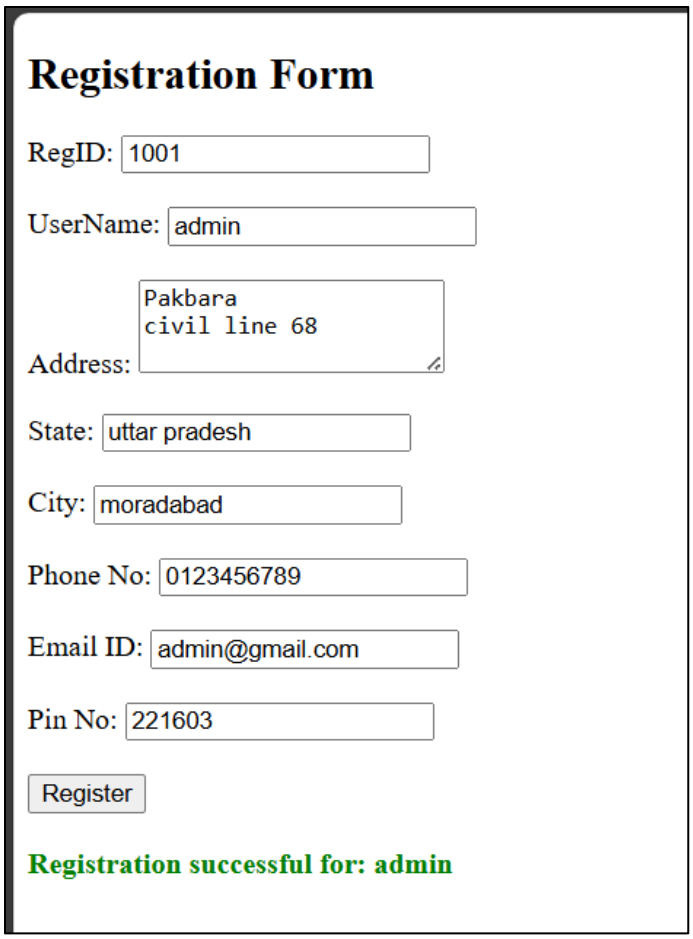
```
using System;

public partial class Login : System.Web.UI.Page
{
    protected void btnLogin_Click(object sender, EventArgs e)
    {
        string username = txtUsername.Text;
        string password = txtPassword.Text;

        // Hardcoded credentials for demo
        if (username == "admin" && password == "1234")
        {
            lblMessage.ForeColor = System.Drawing.Color.Green;
            lblMessage.Text = "Login successful!";
        }
        else
        {
            lblMessage.ForeColor = System.Drawing.Color.Red;
            lblMessage.Text = "Invalid username or password.";
        }
    }
}
```



3 WWP to implement Registration form in ASP.net which field name are:-RegID, UserName, Address, State, City, PhoneNo, EmailId, PinNo.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Registration.aspx.cs" Inherits="Registration" %>

<!DOCTYPE html>
<html>
<head>
   <title>Registration Form</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Registration Form</h2>

      RegID: <asp:TextBox ID="txtRegID" runat="server" /><br /><br />
      UserName: <asp:TextBox ID="txtUserName" runat="server" /><br /><br />
      Address: <asp:TextBox ID="txtAddress" runat="server" TextMode="MultiLine" Rows="3" /><br /><br />
      State: <asp:TextBox ID="txtState" runat="server" /><br /><br />
      City: <asp:TextBox ID="txtCity" runat="server" /><br /><br />
      Phone No: <asp:TextBox ID="txtPhone" runat="server" /><br /><br />
      Email ID: <asp:TextBox ID="txtEmail" runat="server" /><br /><br />
      Pin No: <asp:TextBox ID="txtPin" runat="server" /><br /><br />

      <asp:Button ID="btnRegister" runat="server" Text="Register" OnClick="btnRegister_Click" /><br /><br />

      <asp:Label ID="lblResult" runat="server" Font-Bold="true" ForeColor="Green" />
   </form>
</body>
</html>
```

```
using System;

public partial class Registration : System.Web.UI.Page
{
   protected void btnRegister_Click(object sender, EventArgs e)
   {
      // Just displaying confirmation message
      lblResult.Text = "Registration successful for: " + txtUserName.Text;
   }
}
```



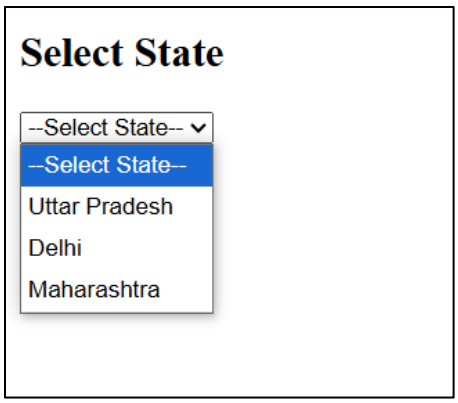
4 WWP to implement combobox in ASP.net and fetch the State or City data from the database.

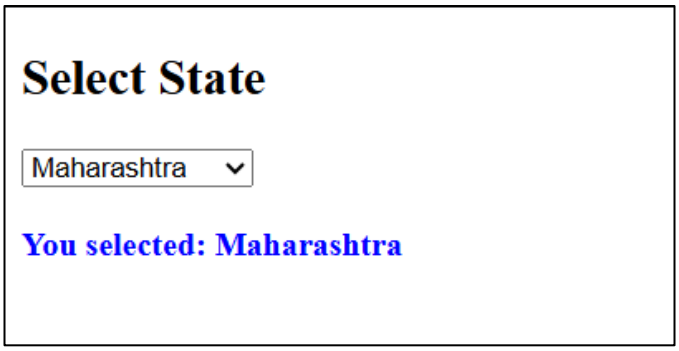
```
CREATE TABLE StateTable (
    StateID INT PRIMARY KEY,
    StateName VARCHAR(100)
);
INSERT INTO StateTable VALUES (1, 'Uttar Pradesh'), (2, 'Delhi'), (3, 'Maharashtra');
```

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="StateDropdown.aspx.cs" Inherits="StateDropdown" %>

<!DOCTYPE html>
<html>
<head>
    <title>Dropdown Fetch from DB</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Select State</h2>

        <asp:DropDownList ID="ddlStates" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="ddlStates_SelectedIndexChanged" />
        <br /><br />
```

```
        <asp:Label ID="lblSelectedState" runat="server" Font-Bold="true"
ForeColor="Blue" />
    </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StateDropdown : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            LoadStates();
        }
    }

    private void LoadStates()
    {
        string connStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlCommand cmd = new SqlCommand("SELECT StateID, StateName FROM
StateTable", con);
            con.Open();
            ddlStates.DataSource = cmd.ExecuteReader();
            ddlStates.DataTextField = "StateName";
            ddlStates.DataValueField = "StateID";
            ddlStates.DataBind();
            ddlStates.Items.Insert(0, new System.Web.UI.WebControls.ListItem("--Select
State--", "0"));
        }
    }

    protected void ddlStates_SelectedIndexChanged(object sender, EventArgs e)
    {
        lblSelectedState.Text = "You selected: " + ddlStates.SelectedItem.Text;
    }
}
```





5 Create a ASP.Net web applications application in which contain three field:

- Student Id (Perform Auto Increment and initial start with 10001)
- Student Name
- Student Course

Use New Button for Auto increment in Id and save button for save the record with the disconnected architecture.

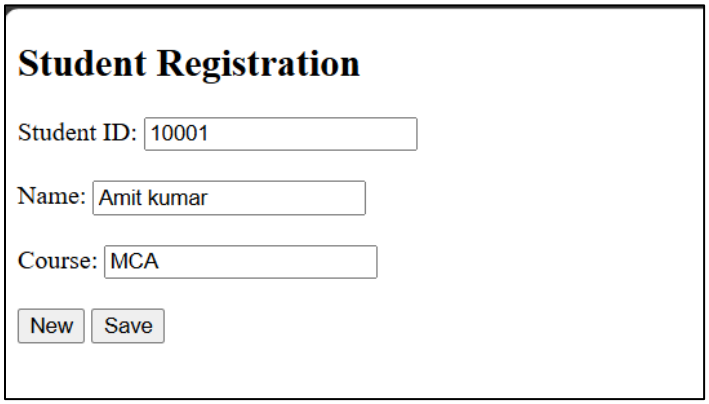
```
CREATE TABLE Students (
    StudentId INT PRIMARY KEY,
    StudentName VARCHAR(100),
    StudentCourse VARCHAR(100)
);
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="StudentForm.aspx.cs"
Inherits="StudentForm" %>

<!DOCTYPE html>
<html>
<head>
    <title>Student Registration</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Student Registration</h2>

        Student ID: <asp:TextBox ID="txtStudentId" runat="server" ReadOnly="true"
/><br /><br />
        Name: <asp:TextBox ID="txtName" runat="server" /><br /><br />
```

```
        Course: <asp:TextBox ID="txtCourse" runat="server" /><br /><br />

        <asp:Button ID="btnNew" runat="server" Text="New" OnClick="btnNew_Click"
/>
        <asp:Button ID="btnSave" runat="server" Text="Save"
OnClick="btnSave_Click" /><br /><br />

        <asp:Label ID="lblMessage" runat="server" Font-Bold="true"
ForeColor="Green" />
    </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StudentForm : System.Web.UI.Page
{
    string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

    protected void btnNew_Click(object sender, EventArgs e)
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlCommand cmd = new SqlCommand("SELECT ISNULL(MAX(StudentId),
10000) + 1 FROM Students", con);
            con.Open();
            int newId = (int)cmd.ExecuteScalar();
            txtStudentId.Text = newId.ToString();
        }
    }

    protected void btnSave_Click(object sender, EventArgs e)
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Students", con);
            DataSet ds = new DataSet();
            da.Fill(ds, "Students");

            DataRow newRow = ds.Tables["Students"].NewRow();
            newRow["StudentId"] = int.Parse(txtStudentId.Text);
            newRow["StudentName"] = txtName.Text;
            newRow["StudentCourse"] = txtCourse.Text;
            ds.Tables["Students"].Rows.Add(newRow);
```

```
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            da.Update(ds, "Students");

            lblMessage.Text = "Record saved successfully!";
        }
    }
}
```



6. Create ASP.Net web applications which contain a data grid, when form will be load then data grid fill with predefine

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="PredefinedGrid.aspx.cs" Inherits="PredefinedGrid" %>

<!DOCTYPE html>
<html>
<head>
    <title>Predefined Data Grid</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Student Details (Predefined Data)</h2>
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="true"
BorderColor="Black" BorderWidth="1px" />
    </form>
</body>
</html>
```

```
using System;
using System.Data;

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

```
    {
        if (!IsPostBack)
        {
            FillGrid();
        }
    }

    private void FillGrid()
    {
        DataTable dt = new DataTable();
        dt.Columns.Add("StudentId", typeof(int));
        dt.Columns.Add("StudentName", typeof(string));
        dt.Columns.Add("Course", typeof(string));

        dt.Rows.Add(10001, "Amit", "BCA");
        dt.Rows.Add(10002, "Priya", "B.Tech");
        dt.Rows.Add(10003, "Rahul", "MCA");

        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
}
```

Student Details (Predefined Data)

StudentId	StudentName	Course
10001	Amit	BCA
10002	Priya	B.Tech
10003	Rahul	MCA

7. Create ASP.Net web application which contains a data grid, when form load this data grid fill with predefined database in sorted manner and contain two textbox one for Student Name and another for Student id to perform pattern search operation as per textbox value in the Data Grid with disconnected architecture.

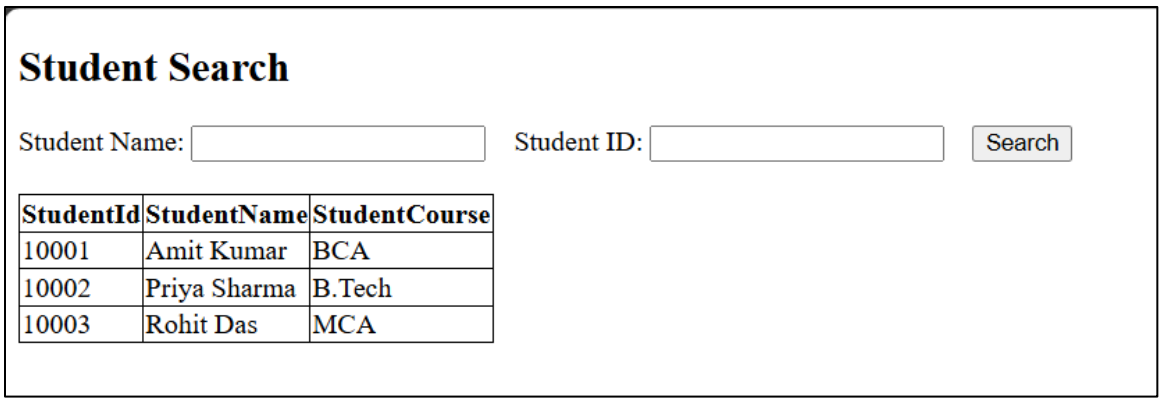
```
CREATE TABLE Students (
    StudentId INT PRIMARY KEY,
    StudentName VARCHAR(100),
    Course VARCHAR(100)
);

-- Sample Data
INSERT INTO Students VALUES
(10001, 'Amit Kumar', 'BCA'),
(10002, 'Priya Sharma', 'B.Tech'),
```

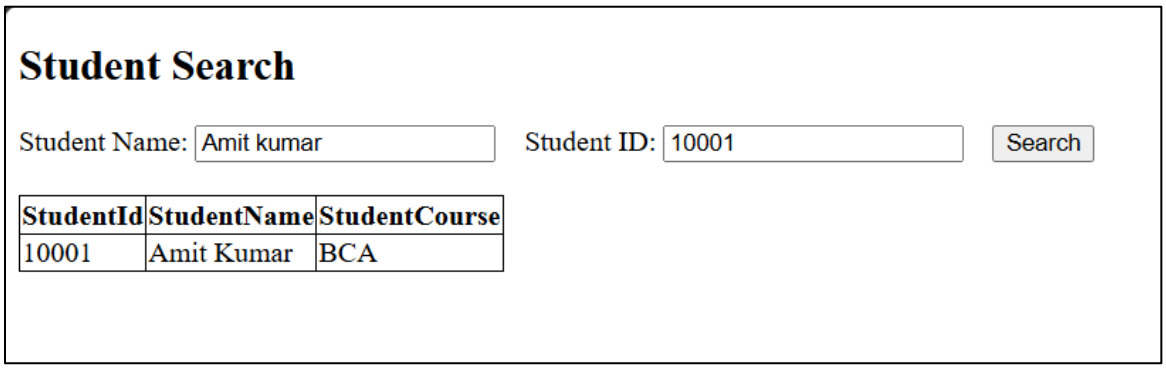
```
(10003, 'Rohit Das', 'MCA');
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="StudentGrid.aspx.cs"
Inherits="StudentGrid" %>

<!DOCTYPE html>
<html>
<head>
   <title>Sorted Student Grid with Search</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Student Search</h2>

      Student Name:
      <asp:TextBox ID="txtName" runat="server" />
        
      Student ID:
      <asp:TextBox ID="txtId" runat="server" />
        
      <asp:Button ID="btnSearch" runat="server" Text="Search"
OnClick="btnSearch_Click" /><br /><br />

      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="true"
BorderWidth="1px" BorderColor="Black" />
   </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StudentGrid : System.Web.UI.Page
{
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

   DataSet ds = new DataSet();

   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
         LoadSortedData();
      }
```

```
    }

    private void LoadSortedData()
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Students ORDER
BY StudentName", con);
            ds.Clear();
            da.Fill(ds, "Students");
            GridView1.DataSource = ds.Tables["Students"];
            GridView1.DataBind();
        }
    }

    protected void btnSearch_Click(object sender, EventArgs e)
    {
        string name = txtName.Text.Trim();
        string id = txtId.Text.Trim();

        using (SqlConnection con = new SqlConnection(connStr))
        {
            string query = "SELECT * FROM Students WHERE 1=1";

            if (!string.IsNullOrEmpty(name))
                query += " AND StudentName LIKE @name";

            if (!string.IsNullOrEmpty(id))
                query += " AND CAST(StudentId AS VARCHAR) LIKE @id";

            SqlDataAdapter da = new SqlDataAdapter(query, con);

            if (!string.IsNullOrEmpty(name))
                da.SelectCommand.Parameters.AddWithValue("@name", "%" + name +
"%");

            if (!string.IsNullOrEmpty(id))
                da.SelectCommand.Parameters.AddWithValue("@id", "%" + id + "%");

            ds.Clear();
            da.Fill(ds, "Students");
            GridView1.DataSource = ds.Tables["Students"];
            GridView1.DataBind();
        }
    }
}
```





8 Create an application which contains fallowing fields:

- Employee ID (Created By User)
- Employee Name
- Employee Date Of Birth
- Employee City
- Employee Mobile No

To perform the following operations

- Add Button for New Record
- Save Button for Save new Record
- Delete Button for Delete a record as per click on the Grid and fill all the Text boxes
- Update Button for Update a Record as per click on the Grid and fill all the Text Boxes

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    EmployeeName VARCHAR(100),
    DOB DATE,
    City VARCHAR(100),
    MobileNo VARCHAR(15)
);
```

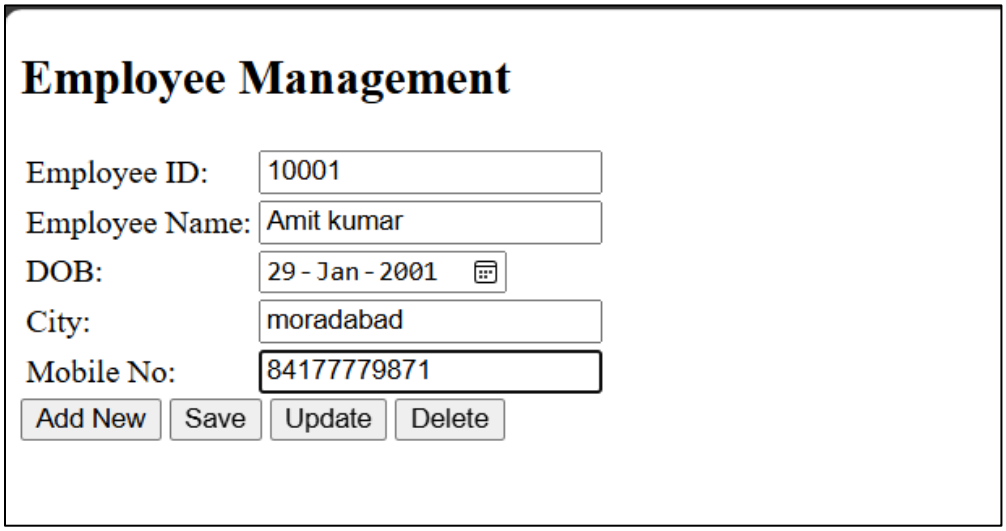
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="EmployeeForm.aspx.cs" Inherits="EmployeeForm" %>

<!DOCTYPE html>
<html>
<head>
    <title>Employee CRUD</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Employee Management</h2>

        <table>
            <tr><td>Employee ID:</td><td><asp:TextBox ID="txtEmpID"
runat="server" /></td></tr>
            <tr><td>Employee Name:</td><td><asp:TextBox ID="txtEmpName"
runat="server" /></td></tr>
            <tr><td>DOB:</td><td><asp:TextBox ID="txtDOB" runat="server"
TextMode="Date" /></td></tr>
            <tr><td>City:</td><td><asp:TextBox ID="txtCity" runat="server"
/></td></tr>
            <tr><td>Mobile No:</td><td><asp:TextBox ID="txtMobile" runat="server"
/></td></tr>
        </table>

        <asp:Button ID="btnAdd" runat="server" Text="Add New"
OnClick="btnAdd_Click" />
        <asp:Button ID="btnSave" runat="server" Text="Save"
OnClick="btnSave_Click" />
        <asp:Button ID="btnUpdate" runat="server" Text="Update"
OnClick="btnUpdate_Click" />
        <asp:Button ID="btnDelete" runat="server" Text="Delete"
OnClick="btnDelete_Click" /><br /><br />

        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
            OnSelectedIndexChanged="GridView1_SelectedIndexChanged"
AutoGenerateSelectButton="True">
            <Columns>
                <asp:BoundField DataField="EmployeeID" HeaderText="Employee ID" />
                <asp:BoundField DataField="EmployeeName" HeaderText="Name" />
```

```
            <asp:BoundField DataField="DOB" HeaderText="DOB"
DataFormatString="{0:yyyy-MM-dd}" />
            <asp:BoundField DataField="City" HeaderText="City" />
            <asp:BoundField DataField="MobileNo" HeaderText="Mobile No" />
         </Columns>
      </asp:GridView>
   </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class EmployeeForm : System.Web.UI.Page
{
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";
   DataSet ds = new DataSet();

   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
      {
         LoadGrid();
      }
   }

   private void LoadGrid()
   {
      using (SqlConnection con = new SqlConnection(connStr))
      {
         SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Employees",
con);
         ds.Clear();
         da.Fill(ds);
         GridView1.DataSource = ds;
         GridView1.DataBind();
      }
   }

   protected void btnAdd_Click(object sender, EventArgs e)
   {
      ClearFields();
   }

   protected void btnSave_Click(object sender, EventArgs e)
```

```
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlCommand cmd = new SqlCommand("INSERT INTO Employees VALUES
(@ID, @Name, @DOB, @City, @Mobile)", con);
            cmd.Parameters.AddWithValue("@ID", Convert.ToInt32(txtEmpID.Text));
            cmd.Parameters.AddWithValue("@Name", txtEmpName.Text);
            cmd.Parameters.AddWithValue("@DOB", Convert.ToDateTime(txtDOB.Text));
            cmd.Parameters.AddWithValue("@City", txtCity.Text);
            cmd.Parameters.AddWithValue("@Mobile", txtMobile.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }

        LoadGrid();
        ClearFields();
    }

    protected void btnUpdate_Click(object sender, EventArgs e)
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlCommand cmd = new SqlCommand("UPDATE Employees SET
EmployeeName=@Name, DOB=@DOB, City=@City, MobileNo=@Mobile WHERE
EmployeeID=@ID", con);
            cmd.Parameters.AddWithValue("@ID", Convert.ToInt32(txtEmpID.Text));
            cmd.Parameters.AddWithValue("@Name", txtEmpName.Text);
            cmd.Parameters.AddWithValue("@DOB", Convert.ToDateTime(txtDOB.Text));
            cmd.Parameters.AddWithValue("@City", txtCity.Text);
            cmd.Parameters.AddWithValue("@Mobile", txtMobile.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
        }

        LoadGrid();
        ClearFields();
    }

    protected void btnDelete_Click(object sender, EventArgs e)
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            SqlCommand cmd = new SqlCommand("DELETE FROM Employees WHERE
EmployeeID=@ID", con);
            cmd.Parameters.AddWithValue("@ID", Convert.ToInt32(txtEmpID.Text));
            con.Open();
            cmd.ExecuteNonQuery();
```

```
            con.Close();
        }

        LoadGrid();
        ClearFields();
    }

    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {
        txtEmpID.Text = GridView1.SelectedRow.Cells[1].Text;
        txtEmpName.Text = GridView1.SelectedRow.Cells[2].Text;
        txtDOB.Text =
Convert.ToDateTime(GridView1.SelectedRow.Cells[3].Text).ToString("yyyy-MM-dd");
        txtCity.Text = GridView1.SelectedRow.Cells[4].Text;
        txtMobile.Text = GridView1.SelectedRow.Cells[5].Text;
    }

    private void ClearFields()
    {
        txtEmpID.Text = "";
        txtEmpName.Text = "";
        txtDOB.Text = "";
        txtCity.Text = "";
        txtMobile.Text = "";
    }
}
```



9 WWP to fetch student Name in a combobox from a Student table in connected mode. (Take Table Fields as Rollno, StudentName, Course, Contactno, Address) in ASP.Net.

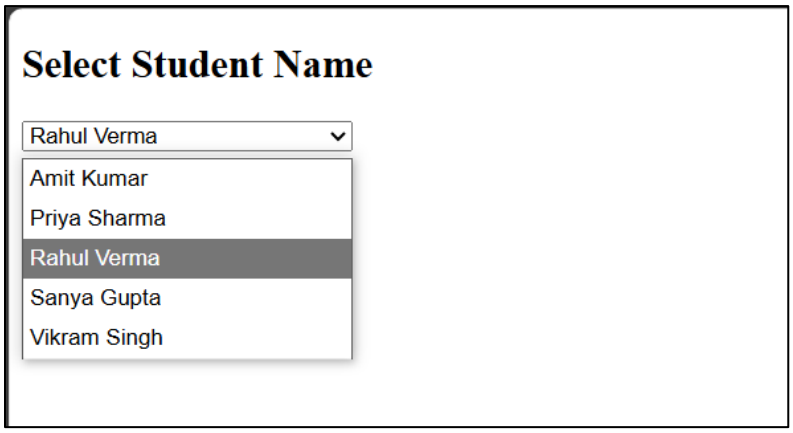
```
CREATE TABLE Student (
    RollNo INT PRIMARY KEY,
    StudentName VARCHAR(100),
```

```
    Course VARCHAR(100),
    ContactNo VARCHAR(15),
    Address VARCHAR(200)
);
```

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="StudentDropdown.aspx.cs" Inherits="StudentDropdown" %>

<!DOCTYPE html>
<html>
<head>
    <title>Student Name ComboBox</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Select Student Name</h2>
        <asp:DropDownList ID="ddlStudentName" runat="server" Width="200px" />
    </form>
</body>
</html>
```

```
using System;
using System.Data.SqlClient;
using System.Configuration;

public partial class StudentDropdown : System.Web.UI.Page
{
    string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            LoadStudentNames();
        }
    }

    private void LoadStudentNames()
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            string query = "SELECT StudentName FROM Student";
            SqlCommand cmd = new SqlCommand(query, con);

            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
```

```
            ddlStudentName.Items.Clear();
            while (reader.Read())
            {
                ddlStudentName.Items.Add(reader["StudentName"].ToString());
            }

            reader.Close();
        }
    }
}
```



10 WWP to fetch student Record in a DataGridview from a Student table in connected mode in ASP.Net.

```
CREATE TABLE Student (
    RollNo INT PRIMARY KEY,
    StudentName VARCHAR(100),
    Course VARCHAR(100),
    ContactNo VARCHAR(15),
    Address VARCHAR(200)
);
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="StudentGrid.aspx.cs"
Inherits="StudentGrid" %>

<!DOCTYPE html>
<html>
<head>
    <title>Student Records in GridView</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Student Record</h2>
```

```
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="True" />
    </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class StudentGrid : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["DBConn"].ConnectionString;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            LoadStudentData();
        }
    }

    private void LoadStudentData()
    {
        using (SqlConnection con = new SqlConnection(connStr))
        {
            string query = "SELECT * FROM Student";
            SqlCommand cmd = new SqlCommand(query, con);
            con.Open();

            SqlDataReader reader = cmd.ExecuteReader();
            GridView1.DataSource = reader;
            GridView1.DataBind();
            reader.Close();
        }
    }
}
```

Student Record

RollNo	StudentName	Course	ContactNo	Address
1	Amit Kumar	Computer Science	9876543210	New Delhi, India
2	Priya Sharma	Mechanical Engineering	9123456780	Pune, Maharashtra
3	Rahul Verma	Electrical Engineering	9988776655	Lucknow, Uttar Pradesh
4	Sanya Gupta	Business Administration	9871234567	Jaipur, Rajasthan
5	Vikram Singh	Information Technology	9654321870	Chandigarh, India

##

Assignment – 3

1 WWP to fetch, insert, delete and update student Record from Student table in connected mode and also use try…catch… statement in ASP.Net.

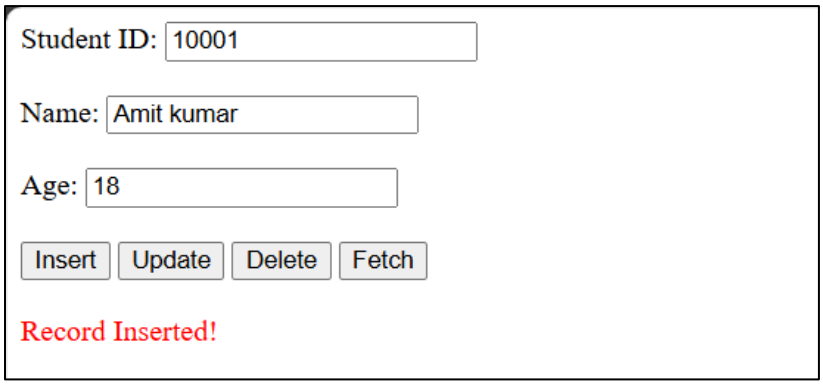
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Student.aspx.cs"
Inherits="Student" %>

<!DOCTYPE html>
<html>
<head>
   <title>Student CRUD</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Student ID: <asp:TextBox ID="txtId" runat="server" /><br />
         Name: <asp:TextBox ID="txtName" runat="server" /><br />
         Age: <asp:TextBox ID="txtAge" runat="server" /><br />
         <asp:Button ID="btnInsert" Text="Insert" runat="server"
OnClick="btnInsert_Click" />
         <asp:Button ID="btnUpdate" Text="Update" runat="server"
OnClick="btnUpdate_Click" />
         <asp:Button ID="btnDelete" Text="Delete" runat="server"
OnClick="btnDelete_Click" />
         <asp:Button ID="btnFetch" Text="Fetch" runat="server"
OnClick="btnFetch_Click" />
         <br /><br />
         <asp:Label ID="lblMsg" runat="server" ForeColor="Red" />
      </div>
```

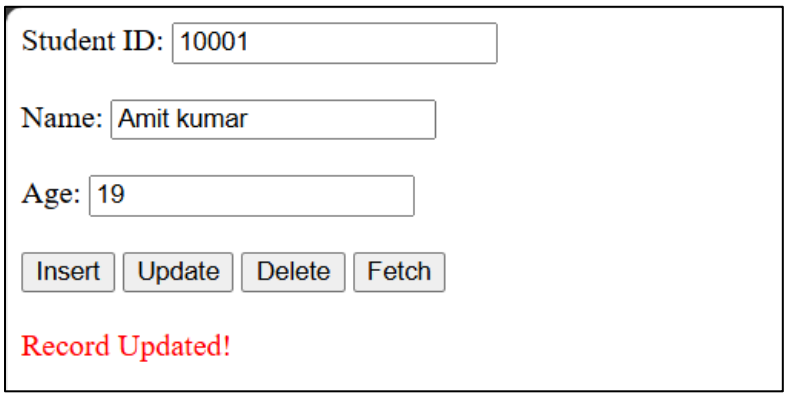
```
    </form>
</body>
</html>
```

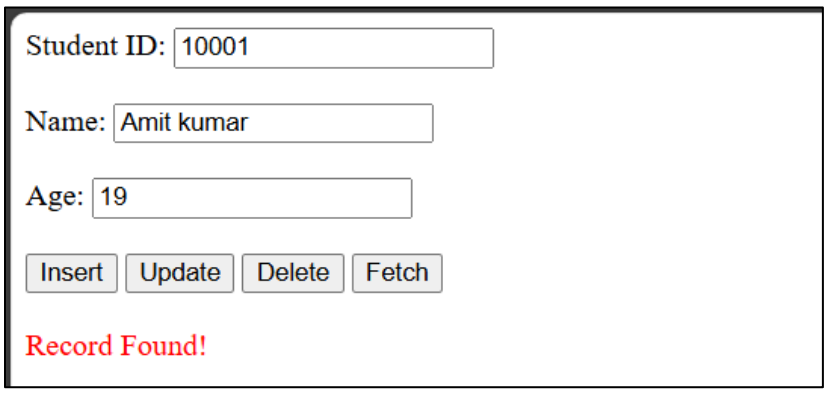
```
using System;
using System.Data.SqlClient;

public partial class Student : System.Web.UI.Page
{
    string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

    protected void btnInsert_Click(object sender, EventArgs e)
    {
        try
        {
            using (SqlConnection con = new SqlConnection(conStr))
            {
                string query = "INSERT INTO Student (StudentId, Name, Age) VALUES
(@Id, @Name, @Age)";
                SqlCommand cmd = new SqlCommand(query, con);
                cmd.Parameters.AddWithValue("@Id", txtId.Text);
                cmd.Parameters.AddWithValue("@Name", txtName.Text);
                cmd.Parameters.AddWithValue("@Age", txtAge.Text);
                con.Open();
                cmd.ExecuteNonQuery();
                lblMsg.Text = "Record Inserted!";
            }
        }
        catch (Exception ex)
        {
            lblMsg.Text = ex.Message;
        }
    }

    protected void btnUpdate_Click(object sender, EventArgs e)
    {
        try
        {
            using (SqlConnection con = new SqlConnection(conStr))
            {
                string query = "UPDATE Student SET Name=@Name, Age=@Age WHERE
StudentId=@Id";
                SqlCommand cmd = new SqlCommand(query, con);
                cmd.Parameters.AddWithValue("@Id", txtId.Text);
                cmd.Parameters.AddWithValue("@Name", txtName.Text);
                cmd.Parameters.AddWithValue("@Age", txtAge.Text);
                con.Open();
```

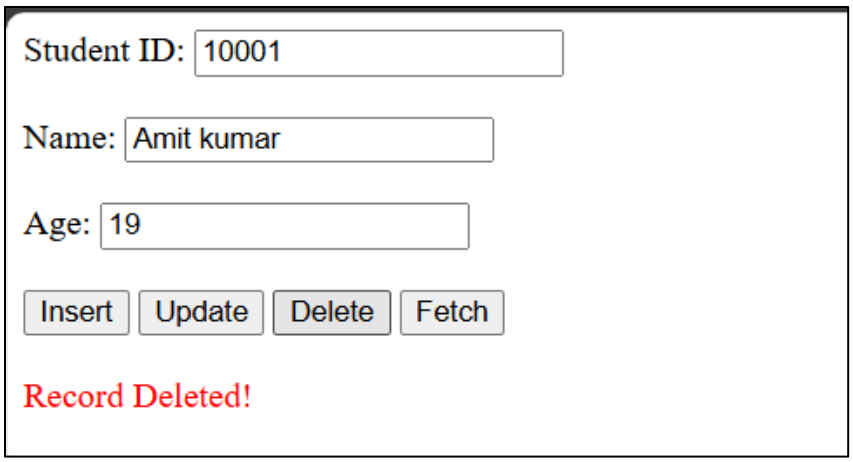
```
            cmd.ExecuteNonQuery();
            lblMsg.Text = "Record Updated!";
        }
    }
    catch (Exception ex)
    {
        lblMsg.Text = ex.Message;
    }
}

protected void btnDelete_Click(object sender, EventArgs e)
{
    try
    {
        using (SqlConnection con = new SqlConnection(conStr))
        {
            string query = "DELETE FROM Student WHERE StudentId=@Id";
            SqlCommand cmd = new SqlCommand(query, con);
            cmd.Parameters.AddWithValue("@Id", txtId.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            lblMsg.Text = "Record Deleted!";
        }
    }
    catch (Exception ex)
    {
        lblMsg.Text = ex.Message;
    }
}

protected void btnFetch_Click(object sender, EventArgs e)
{
    try
    {
        using (SqlConnection con = new SqlConnection(conStr))
        {
            string query = "SELECT Name, Age FROM Student WHERE
StudentId=@Id";
            SqlCommand cmd = new SqlCommand(query, con);
            cmd.Parameters.AddWithValue("@Id", txtId.Text);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            if (reader.Read())
            {
                txtName.Text = reader["Name"].ToString();
                txtAge.Text = reader["Age"].ToString();
                lblMsg.Text = "Record Found!";
            }
            else
```

```
            {
                lblMsg.Text = "No Record Found.";
            }
        }
    }
    catch (Exception ex)
    {
        lblMsg.Text = ex.Message;
    }
  }
}
```









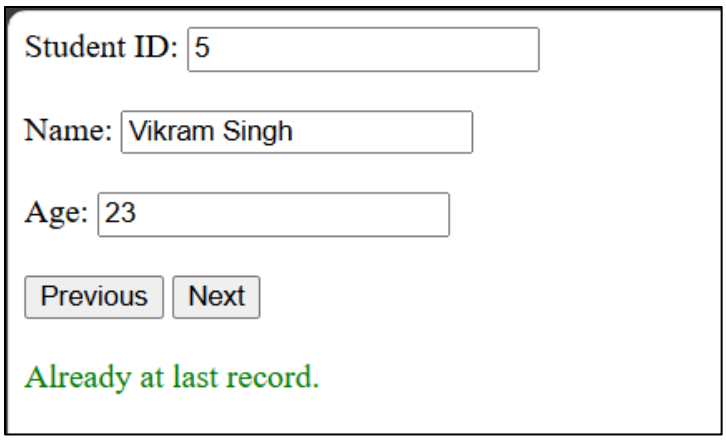
2 WWP to fetch student Record in a textboxes from a Student table and use previous and next button to navigate records in ASP.Net.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="StudentNavigation.aspx.cs" Inherits="StudentNavigation" %>

<!DOCTYPE html>
<html>
<head>
   <title>Student Record Navigation</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Student ID: <asp:TextBox ID="txtId" runat="server" ReadOnly="true" /><br
/>
         Name: <asp:TextBox ID="txtName" runat="server" ReadOnly="true" /><br
/>
         Age: <asp:TextBox ID="txtAge" runat="server" ReadOnly="true" /><br
/><br />

         <asp:Button ID="btnPrevious" Text="Previous" runat="server"
OnClick="btnPrevious_Click" />
         <asp:Button ID="btnNext" Text="Next" runat="server"
OnClick="btnNext_Click" /><br /><br />

         <asp:Label ID="lblMsg" runat="server" ForeColor="Green" />
      </div>
   </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;
```

```
public partial class StudentNavigation : System.Web.UI.Page
{
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";
   static DataTable dt;
   static int currentIndex = 0;

   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
         LoadStudents();
   }

   void LoadStudents()
   {
      try
      {
         using (SqlConnection con = new SqlConnection(conStr))
         {
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Student
ORDER BY StudentId", con);
            dt = new DataTable();
            da.Fill(dt);

            if (dt.Rows.Count > 0)
            {
               currentIndex = 0;
               DisplayRecord();
            }
            else
            {
               lblMsg.Text = "No Records Found!";
            }
         }
      }
      catch (Exception ex)
      {
         lblMsg.Text = ex.Message;
      }
   }

   void DisplayRecord()
   {
      txtId.Text = dt.Rows[currentIndex]["StudentId"].ToString();
      txtName.Text = dt.Rows[currentIndex]["Name"].ToString();
      txtAge.Text = dt.Rows[currentIndex]["Age"].ToString();
   }

   protected void btnPrevious_Click(object sender, EventArgs e)
```

```
    {
        if (dt != null && dt.Rows.Count > 0)
        {
            if (currentIndex > 0)
            {
                currentIndex--;
                DisplayRecord();
            }
            else
            {
                lblMsg.Text = "Already at first record.";
            }
        }
    }

    protected void btnNext_Click(object sender, EventArgs e)
    {
        if (dt != null && dt.Rows.Count > 0)
        {
            if (currentIndex < dt.Rows.Count - 1)
            {
                currentIndex++;
                DisplayRecord();
            }
            else
            {
                lblMsg.Text = "Already at last record.";
            }
        }
    }
}
```



3 Write Web based Program to fetch student Record in a Gridview from a Student table by using the DataSet Class in disconnected mode in ASP.Net.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="StudentGridView.aspx.cs" Inherits="StudentGridView" %>

<!DOCTYPE html>
<html>
<head>
   <title>Student Records in GridView</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="true"></asp:GridView>
      </div>
   </form>
</body>
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;

public partial class StudentGridView : System.Web.UI.Page
{
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
         LoadStudents();
   }

   void LoadStudents()
```

```
    {
        try
        {
            using (SqlConnection con = new SqlConnection(conStr))
            {
                SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Student", con);
                DataSet ds = new DataSet();
                da.Fill(ds);
                GridView1.DataSource = ds.Tables[0];
                GridView1.DataBind();
            }
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
}
```

StudentId	Name	Age
1	Amit Kumar	20
2	Priya Sharma	22
3	Rahul Verma	21
4	Sanya Gupta	19
5	Vikram Singh	23

4 Write Web based Program to fetch student Record in a Gridview from a Student table by using the DataTable Class in disconnected mode in ASP.Net.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="StudentGridView_DT.aspx.cs" Inherits="StudentGridView_DT" %>

<!DOCTYPE html>
<html>
<head>
    <title>Student Records - DataTable</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="true"></asp:GridView>
        </div>
    </form>
</body>
```

```
</html>
```

```
using System;
using System.Data;
using System.Data.SqlClient;

public partial class StudentGridView_DT : System.Web.UI.Page
{
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

   protected void Page_Load(object sender, EventArgs e)
   {
      if (!IsPostBack)
         LoadStudents();
   }

   void LoadStudents()
   {
      try
      {
         using (SqlConnection con = new SqlConnection(conStr))
         {
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Student", con);
            DataTable dt = new DataTable();
            da.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
         }
      }
      catch (Exception ex)
      {
         Response.Write(ex.Message);
      }
   }
}
```

StudentId	Name	Age
1	Amit Kumar	20
2	Priya Sharma	22
3	Rahul Verma	21
4	Sanya Gupta	19
5	Vikram Singh	23

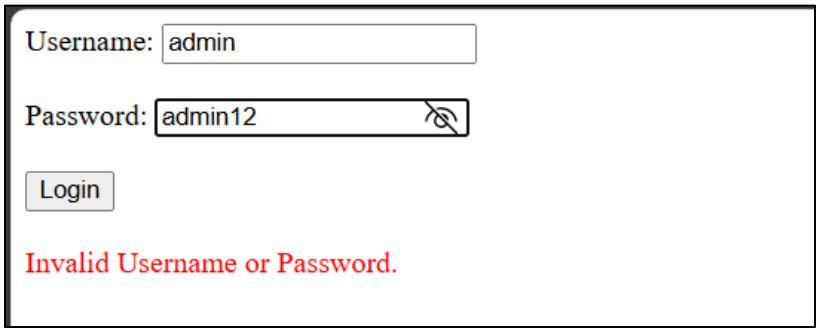
5 Write Web based Program to create a Login Page and take Login table having fields as Username, Password in ASP.Net.

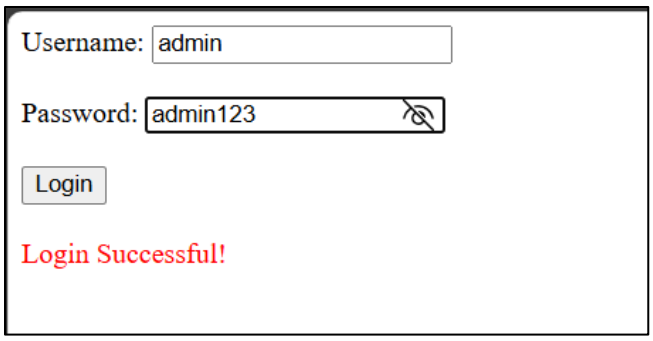
```
CREATE TABLE Login (
    Username VARCHAR(50) PRIMARY KEY,
    Password VARCHAR(50)
);

INSERT INTO Login (Username, Password) VALUES ('admin', 'admin123');
```

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="Login" %>

<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Username: <asp:TextBox ID="txtUsername" runat="server" /><br />
            Password: <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password" /><br /><br />
            <asp:Button ID="btnLogin" Text="Login" runat="server"
OnClick="btnLogin_Click" /><br /><br />
            <asp:Label ID="lblMsg" runat="server" ForeColor="Red" />
        </div>
    </form>
</body>
</html>
```

```
using System;
using System.Data.SqlClient;

public partial class Login : System.Web.UI.Page
{
```

```
   string conStr = "Data Source=YOUR_SERVER;Initial
Catalog=YOUR_DATABASE;Integrated Security=True";

   protected void btnLogin_Click(object sender, EventArgs e)
   {
      try
      {
         using (SqlConnection con = new SqlConnection(conStr))
         {
            string query = "SELECT COUNT(*) FROM Login WHERE
Username=@Username AND Password=@Password";
            SqlCommand cmd = new SqlCommand(query, con);
            cmd.Parameters.AddWithValue("@Username", txtUsername.Text);
            cmd.Parameters.AddWithValue("@Password", txtPassword.Text);

            con.Open();
            int count = (int)cmd.ExecuteScalar();

            if (count == 1)
               lblMsg.Text = "Login Successful!";
            else
               lblMsg.Text = "Invalid Username or Password.";
         }
      }
      catch (Exception ex)
      {
         lblMsg.Text = ex.Message;
      }
   }
}
```





6 Write Web based Program to create master page having menus as Home, About Us, and Contact Us. Products in ASP.Net.

SiteMaster.master (the Master Page)

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="SiteMaster.master.cs" Inherits="SiteMaster" %>

<!DOCTYPE html>
<html>
<head>
   <title>My Website</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <!-- Navigation Menu -->
         <asp:Menu ID="Menu1" runat="server" Orientation="Horizontal">
            <Items>
               <asp:MenuItem Text="Home" NavigateUrl="~/Home.aspx" />
               <asp:MenuItem Text="About Us" NavigateUrl="~/AboutUs.aspx" />
               <asp:MenuItem Text="Contact Us" NavigateUrl="~/ContactUs.aspx" />
               <asp:MenuItem Text="Products" NavigateUrl="~/Products.aspx" />
            </Items>
         </asp:Menu>
         <br /><br />

         <!-- Content Placeholder -->
         <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server" />
      </div>
   </form>
</body>
</html>
```

SiteMaster.master.cs

```
using System;

public partial class SiteMaster : System.Web.UI.MasterPage
{
```

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

Home.aspx

```
<%@ Page Title="Home"  Language="C#" MasterPageFile="~/SiteMaster.master"
AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
    <h2>Welcome to Home Page</h2>
</asp:Content>
```

Similarly create:

- **AboutUs.aspx**
- **ContactUs.aspx**
- **Products.aspx**

7 Create two web pages and display data of one web page on another web page using statement management technique namely Application in ASP.Net.

Page1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Page1.aspx.cs"
Inherits="Page1" %>

<!DOCTYPE html>
<html>
<head>
    <title>Page 1 - Enter Data</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Enter Your Name:
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox><br /><br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit and Go to Page2"
OnClick="btnSubmit_Click" />
        </div>
    </form>
</body>
</html>
```

Page1.aspx.cs

```
using System;

public partial class Page1 : System.Web.UI.Page
{
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        Application["UserName"] = txtName.Text; // Store data in Application state
        Response.Redirect("Page2.aspx"); // Redirect to second page
    }
}
```

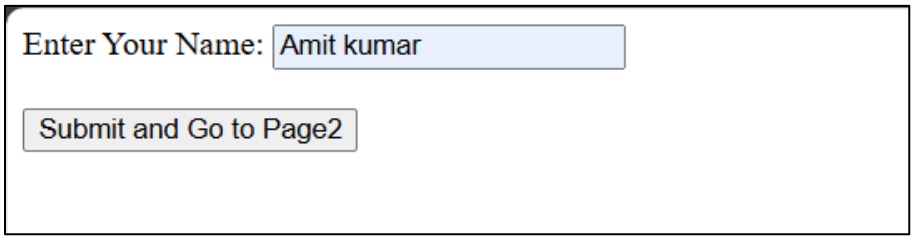
Page2.aspx

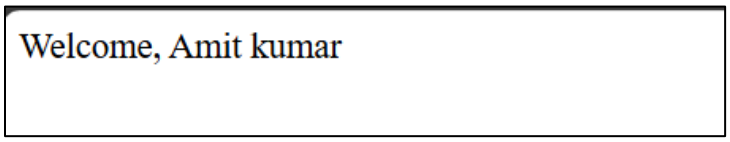
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Page2.aspx.cs"
Inherits="Page2" %>

<!DOCTYPE html>
<html>
<head>
    <title>Page 2 - Display Data</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="lblName" runat="server" Font-Size="Large" />
        </div>
    </form>
</body>
</html>
```

Page2.aspx.cs

```
using System;

public partial class Page2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Application["UserName"] != null)
        {
            lblName.Text = "Welcome, " + Application["UserName"].ToString();
        }
        else
        {
            lblName.Text = "No data found!";
        }
    }
```

```
}
```





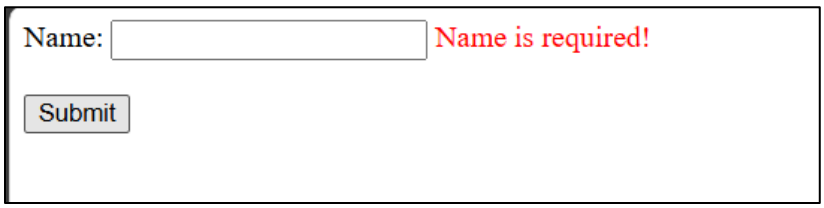
8 Write Web based Program to use the RequiredFieldValidator in ASP.Net.

RequiredFieldValidatorExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RequiredFieldValidatorExample.aspx.cs"
Inherits="RequiredFieldValidatorExample" %>

<!DOCTYPE html>
<html>
<head>
   <title>RequiredFieldValidator Example</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Name:
         <asp:TextBox ID="txtName" runat="server" />
         <asp:RequiredFieldValidator
            ID="rfvName"
            runat="server"
            ControlToValidate="txtName"
            ErrorMessage="Name is required!"
            ForeColor="Red"
            Display="Dynamic" />
         <br /><br />

         <asp:Button ID="btnSubmit" Text="Submit" runat="server"
OnClick="btnSubmit_Click" /><br /><br />
         <asp:Label ID="lblMsg" runat="server" ForeColor="Green" />
      </div>
   </form>
</body>
</html>
```

RequiredFieldValidatorExample.aspx.cs

```
using System;

public partial class RequiredFieldValidatorExample : System.Web.UI.Page
{
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        if (Page.IsValid)
        {
            lblMsg.Text = "Form submitted successfully!";
        }
    }
}
```

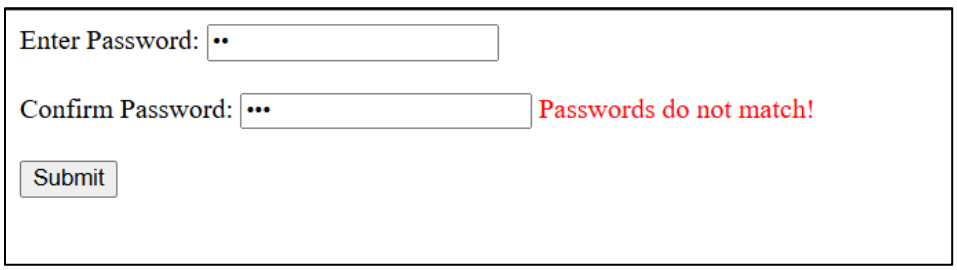


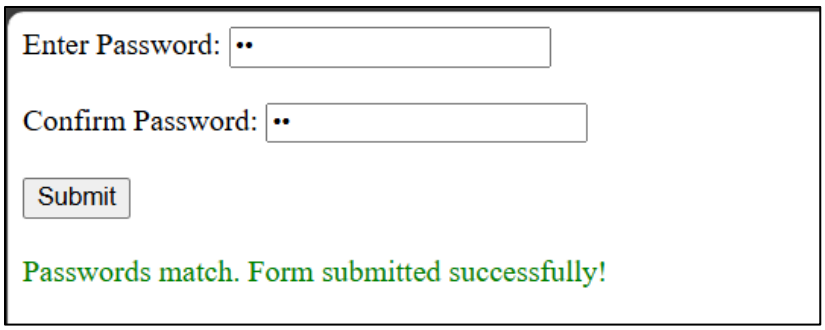
9 Write Web based Program to use the CompareValidator in ASP.Net

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CompareValidatorExample.aspx.cs" Inherits="CompareValidatorExample"
%>

<!DOCTYPE html>
<html>
<head>
    <title>CompareValidator Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Enter Password:
            <asp:TextBox ID="txtPassword" runat="server" TextMode="Password" />
```

```
        <br /><br />

        Confirm Password:
        <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password" />
        <asp:CompareValidator
           ID="cvPassword"
           runat="server"
           ControlToValidate="txtConfirmPassword"
           ControlToCompare="txtPassword"
           ErrorMessage="Passwords do not match!"
           ForeColor="Red"
           Display="Dynamic" />
        <br /><br />

        <asp:Button ID="btnSubmit" Text="Submit" runat="server"
OnClick="btnSubmit_Click" /><br /><br />
        <asp:Label ID="lblMsg" runat="server" ForeColor="Green" />
      </div>
   </form>
</body>
</html>
```

```
using System;

public partial class CompareValidatorExample : System.Web.UI.Page
{
   protected void btnSubmit_Click(object sender, EventArgs e)
   {
      if (Page.IsValid)
      {
         lblMsg.Text = "Passwords match. Form submitted successfully!";
      }
   }
}
```





10 Write Web based Program to use the RegularExpression Validator in ASP.Net.

RegularExpressionValidatorExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RegularExpressionValidatorExample.aspx.cs"
Inherits="RegularExpressionValidatorExample" %>

<!DOCTYPE html>
<html>
<head>
   <title>RegularExpressionValidator Example</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         Enter Your Email:
         <asp:TextBox ID="txtEmail" runat="server" />
         <asp:RegularExpressionValidator
            ID="revEmail"
            runat="server"
            ControlToValidate="txtEmail"
            ValidationExpression="^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,}$"
            ErrorMessage="Please enter a valid email address."
            ForeColor="Red"
            Display="Dynamic" />
         <br /><br />

         <asp:Button ID="btnSubmit" Text="Submit" runat="server"
OnClick="btnSubmit_Click" /><br /><br />
         <asp:Label ID="lblMsg" runat="server" ForeColor="Green" />
      </div>
   </form>
</body>
</html>
```

RegularExpressionValidatorExample.aspx.cs

```
using System;
```

```
public partial class RegularExpressionValidatorExample : System.Web.UI.Page
{
   protected void btnSubmit_Click(object sender, EventArgs e)
   {
      if (Page.IsValid)
      {
         lblMsg.Text = "Email is valid. Form submitted successfully!";
      }
   }
}
```