

# Bharati Vidyapeeth’s

Institute of Management & Information Technology

# C.B.D. Belapur, Navi Mumbai 400614

**Vision:**

Providing high-quality, innovative, and value-based education in information technology to build competent professionals.

**Mission**

M1. Technical Skills:-To provide a solid technical foundation theoretically as well as practically capable of providing quality services to the industry.

M2. Development: -Department caters to the needs of students through comprehensive educational programs and promotes lifelong learning in the field of computer Applications.

M3. The ethical leadership:-Department develops ethical leadership insight in the students to succeed in industry, government, and academia.

**CERTIFICATE**

# This is to certify that the journal is the work of Ms.

**Vaishnavi Vidyadhar Kothawade** Roll No**. 32** of MCA (Sem- 2 Div. B) For the academic year 2023 - 2025

Subject Code: MCALE232

Subject Name: Advanced Web Technologies

Subject-in-charge Principal

Date:

External Examiner Date:

**Bharati Vidyapeeth's Institute of Management & Information Technology**

**MCA Semester II AY 2021-22**

**Subject : MCA L24 Advanced Web Technologies INDEX**

|  |  |  |
| --- | --- | --- |
| I | **Module: Basics of C#** |  |
| 1 | Create a Functional calculator using C#. |  |
| 2 | Write a program in C# to list the squares of given range of numbers in a list box. |  |
| 3 | Create a Windows based application to design a screensaver. |  |
| 4 | Create a class to check whether user input year is leap year on not. |  |
| 5 | Write a Program to create a class to calculate simple interest and get and set method. |  |
| 6 | WAP in C# to implement Single level inheritance |  |
| 7 | WAP in C# to implement hierarchical level inheritance |  |
| 8 | WAP in C# to implement Abstract class. |  |
| 9 | WAP in C# to implement Interface class. |  |
| II | **Module: Introduction to ASP.NET** |  |
| 1 | Design ASP.NET web page to demonstrate page life cycle. |  |
| 2 | Design ASP.NET web page to demonstrate page life cycle. |  |
| 3 | Design an ASP.NET Application to Display Random Advertisements using ADRotator Control |  |
| 4 | Design an ASP.NET application to Display Current Month’s Calender. Render the calendar.Show Some image in the cell along message. |  |
| 5 | Design Registration form with Validation Controls. |  |
| 6 | Design ASP.NET web page to demonstrate File Upload control |  |
| 7 | Create website using master page concept. |  |
| III | **Module: Database Programming in ASP.NET** |  |
| 1 | Design a webpage For database connectivity (connected and disconnected connectivity) |  |
| 2 | Design a webpage to insert update, delete, search details using connect Architecture. |  |
| 3 | Design a webpage to insert update, delete, search details using Disconnect Architecture. |  |
| 4 | Design a webpage to demonstrate the working of a simple stored procedure. |  |
| 5 | Design a webpage to demonstrate the working of parameterized stored Procedure |  |
| 6 | Web Application to demonstrate LINQ with object data source. |  |

|  |  |  |
| --- | --- | --- |
| 7 | Web Application to demonstrate LINQ with data set. |  |
| 8 | Create a webpage that demonstrates the use of data bound controls |  |
| 9 | Build websites to demonstrate the working of entity framework |  |
| 10 | Design a web page to perform CRUD operation using SqlDataSource |  |
| IV | **Module: Session Management and AJAX** |  |
| 1 | Display the number of times current page is visited using View State, Session, Cookies and Application State. |  |
| 2 | Design a webpage to read, write and remove cookies. |  |
| 3 | Design Web Application using Server Side Session Management Techniques(login) |  |
| 4 | Design Web Application to show a list of books.on click of submit  button selected book should be added in session and book details should be displayed on another page. |  |
| 5 | Design Web Application to read 10 students info from database table into a gridview.cache this output for 10 sec.use cache object. |  |
| 6 | Design Web Application using ASP.NET Ajax Controls |  |
| V | **Module: Web Services and WCF** |  |
| 1 | Create xml based webservice to create calculator and consume it in website. |  |
| 2 | Design a Web Service to Fetch & insert Details of Employee Table using ADO.NET. Design a Web Client to show contents of table in a Grid View. |  |
| 3 | Create and consume the WCF service to calculate simple interest. |  |
| 4 | Create a WCF Web Service Using Database |  |
| VI | **Module: ASP.NET MVC** |  |
| 1 | MVC Application With Model view controller display information(students info) |  |
| 2 | MVC Application for customer data entry using HTML helper and Validation |  |
| 3 | Build MVC Application to from CURD operations using EF. |  |
| 4 | Build an application Using JQuery.(Table even odd row format,Filter()) |  |
| 5 | Build an angular web application.(basic calculator) |  |

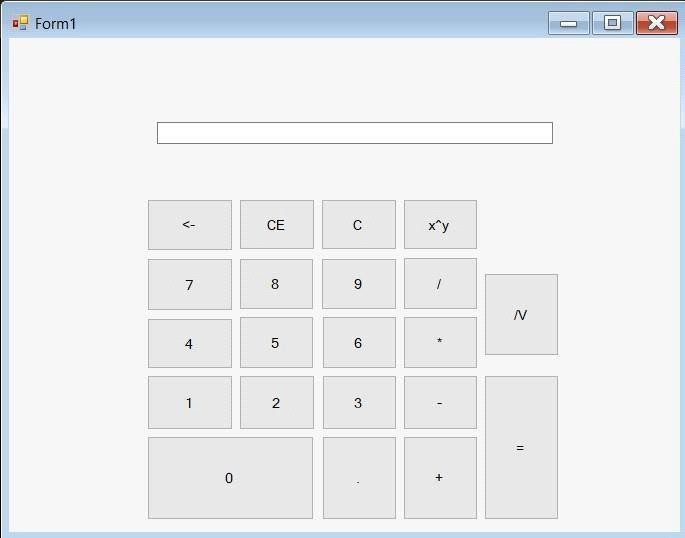
# Module : Basics of C#

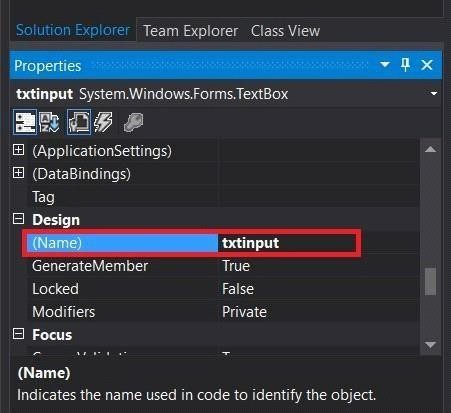
**Q1. Create a Functional calculator using C#.**

**Steps-**

Open visual studio -> file-> new project -> select visual c# -> window form application -> save (calculator.cs)

Form-1





**INPUT-**

**Filename-cal.cs** using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;

namespace Calculator

{

public partial class cal : Form

{

// variable to hold operands string calfun; double v1, v2; // hold operand

public cal()

{

InitializeComponent();

}

private void btn0\_Click(object sender, EventArgs e)

{

txtinput.Text += btn0.Text;

}

private void btn1\_Click(object sender, EventArgs e)

{

txtinput.Text += btn1.Text;

} private void btn2\_Click(object sender, EventArgs e)

{

txtinput.Text += btn2.Text;

}

private void btn3\_Click(object sender, EventArgs e)

{

txtinput.Text += btn3.Text;

}

private void btn4\_Click(object sender, EventArgs e)

{

txtinput.Text += btn4.Text;

}

private void btn5\_Click(object sender, EventArgs e)

{

txtinput.Text += btn5.Text;

}

private void btn6\_Click(object sender, EventArgs e) {

txtinput.Text += btn6.Text;

}

private void btn7\_Click(object sender, EventArgs e)

{

txtinput.Text += btn7.Text;

} private void btn8\_Click(object sender, EventArgs e)

{

txtinput.Text += btn8.Text;

}

private void btn9\_Click(object sender, EventArgs e)

{

txtinput.Text += btn9.Text;

}

private void btnaddition\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text); calfun = "add" ; txtinput.Text = "

";

}

private void btnsub\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text); calfun = "mins"; txtinput.Text = " ";

}

private void btnmulti\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text); calfun = "multi"; txtinput.Text =

"";

}

private void btndivi\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text); calfun = "div"; txtinput.Text =

"";

}

private void btnsquareroot\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text);

txtinput.Text = Math.Sqrt(v1).ToString();// converted into string

} private void btnequal\_Click(object sender, EventArgs e)

{

v2 = Convert.ToDouble(txtinput.Text); switch (calfun)

{

case "add": v1 = v1 + v2; break;

case "minus": v1 = v1 - v2; break;

case "multi": v1 = v1 \* v2; break; case "div": v1 = v1 / v2; break;

case "PowerOf": v1 = System.Math.Pow(v1, v2); break; }//switch closed

txtinput.Text = v1.ToString();

}

private void btnbackspace\_Click(object sender, EventArgs e)

{

if (txtinput.Text != "")

{

int l = txtinput.Text.Length;

txtinput.Text = txtinput.Text.Remove(l - 1);//

}

}

private void btndecimal\_Click(object sender, EventArgs e)

{

txtinput.Text += btndecimal.Text;

} private void btnclearentry\_Click(object sender, EventArgs e)

{

txtinput.Text = string.Empty;

}

private void btnclearall\_Click(object sender, EventArgs e)

{ v1 = 0; v2 = 0;

txtinput.Text = "";

}

private void btnpowerof\_Click(object sender, EventArgs e)

{

v1 = Convert.ToDouble(txtinput.Text); calfun = "PowerOf"; txtinput.Text = "";

}

}}

**Filename-Program.cs** using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace Calculator

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary> [STAThread] static void Main()

{

Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new cal());

}

}

}

OUTPUT-

**Q2. Write a program in C# to list the squares of given range of numbers in a list box. INPUT-**

**Filename- form1.cs** using System; using System.Collections.Generic; using System.ComponentModel;

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

namespace SquaresofrangNumber

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{ int n1, n2; n1 = int.Parse(textBox1.Text); n2 =

int.Parse(textBox2.Text); while (n1 <= n2)

{ int s = n1 \* n1;

string b = String.Format("{0,5}{1,30}", n1, s); // n1 first argu andu s is next argu listBox1.Items.Add(b); n1++;

}

}

private void button2\_Click(object sender, EventArgs e)

{ textBox1.Clear(); textBox2.Clear(); listBox1.Items.Clear();

}

private void button3\_Click(object sender, EventArgs e)

{

Close();

}

}

}

**Filename-Program.cs** using System;

using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace SquaresofrangNumber

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary> [STAThread] static void Main()

{

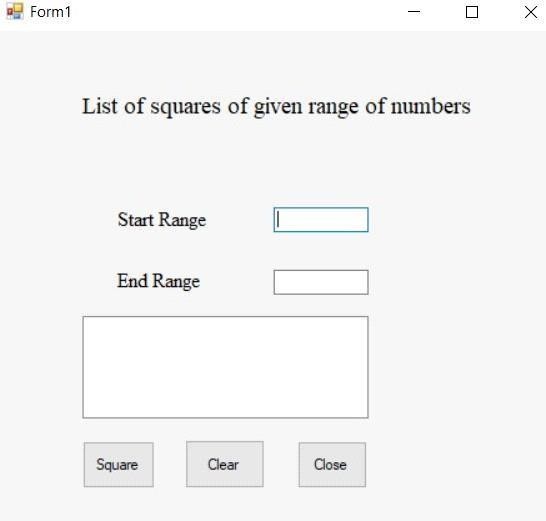
Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new Form1());

}

}

}

**OUTPUT-**



**Q3. Create a Windows based application to design a screensaver. Steps-**

Open visual studio -> file-> new project -> select visual c# -> window form application -> save (Screensaver.cs)

**INPUT-**

**Filename- form1.cs** using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;

namespace Screensaver

{

public partial class Form1 : Form

{

string[] s = new string[5]; int i = 1; public

Form1()

{

InitializeComponent();

}

private void btn1\_Click(object sender, EventArgs e)

{

timer1.Enabled = true;

s[0] = @"D:\A\_39\_Visual Studio\_Practicals\Screensaver\Screensaver\imges\h1.jpg"; s[1] = @"D:\A\_39\_Visual Studio\_Practicals\Screensaver\Screensaver\imges\h2.jpg"; s[2] = @"D:\A\_39\_Visual Studio\_Practicals\Screensaver\Screensaver\imges\h3.jpg"; s[3] = @"D:\A\_39\_Visual Studio\_Practicals\Screensaver\Screensaver\imges\h4.jpg";

}

private void btn2\_Click(object sender, EventArgs e)

{

timer1.Enabled = false;

}

private void timer1\_Tick(object sender, EventArgs e)

{

pictureBox1.Image = System.Drawing.Image.FromFile(s[i]); i++; if (i == 4) { i = 0; }

}

}

} **Filename-Program.cs**

using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace Screensaver

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary> [STAThread] static void Main()

{

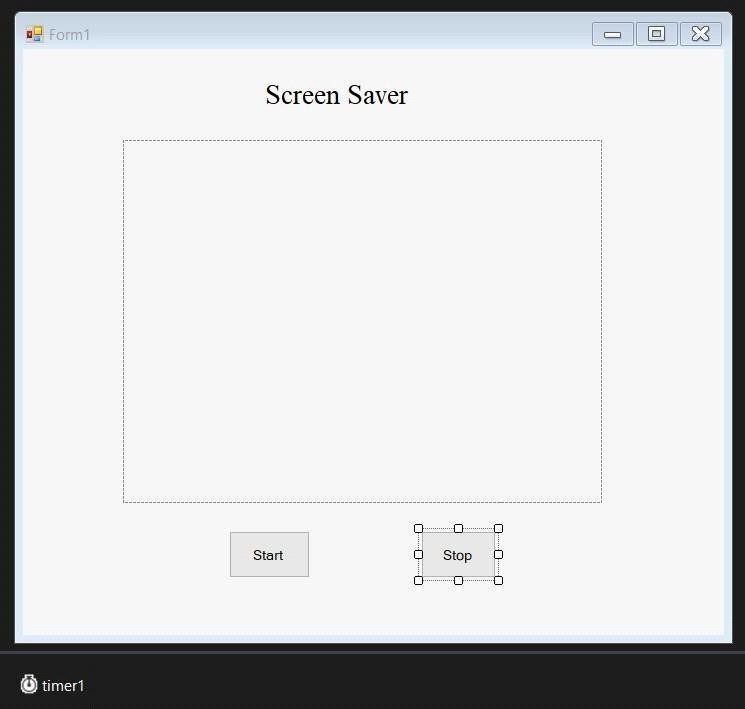
Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new Form1());

}

}

}

**Output:**



**Run file- > After click on Start button and stop button**

**Q4. Create a class to check whether user input year is leap year on not.**

**Steps-**

Open visual studio -> file-> new -> project -> select visual c# -> window form application -> save (Cshapprograms.cs)

Right click on project -> add -> class-> give name (leapyear.cs)

**INPUT-**

**Filename- Class file(leapyear.cs)** using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;

namespace Csharpprograms

{ class leapyear

{

public int getLeapYear(int year)

{

if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0) return 1; // It is a leap year

else return 0;

}

}

}

**Filename- leapyear1.cs** using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Linq; using

System.Text; using System.Threading.Tasks; using System.Windows.Forms;

namespace Csharpprograms

{ public partial class leapyear1 : Form

{

public leapyear1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{ bool

flag; int

year; int leap;

leapyear myDate = new leapyear();

flag = int.TryParse(textBox1.Text, out year); if (flag == false)

{

MessageBox.Show("Digit characters only in YYYY format.", "Input Error"); textBox1.Focus();

return;

}

leap = myDate.getLeapYear(year);

lblLeapYearResult.Text = year.ToString() + " is " + ((leap == 1) ? "" : "not ") + "a leap year";

}

}

}

**Filename- Program.cs** using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace Csharpprograms

{ static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary> [STAThread] static void Main()

{

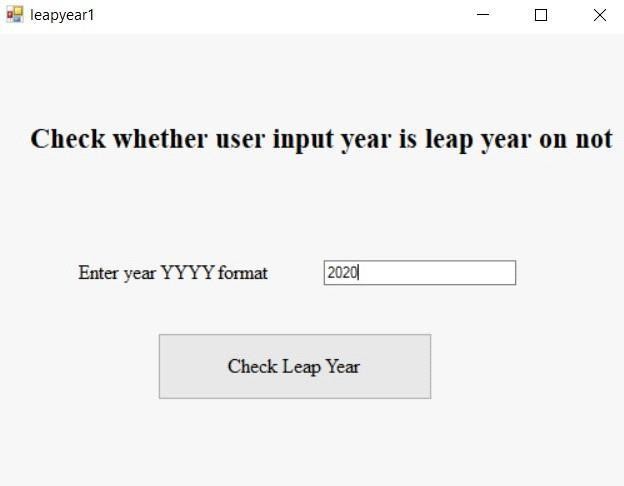
Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new leapyear1());

}

}

}

**OUTPUT-**

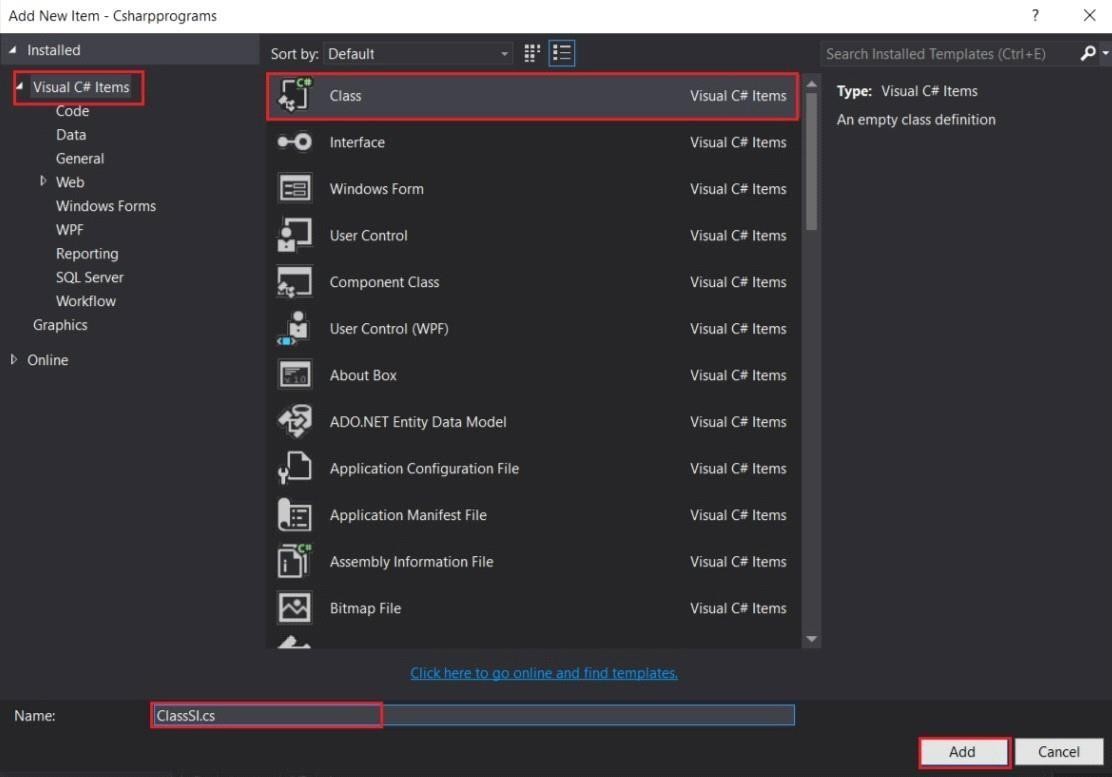
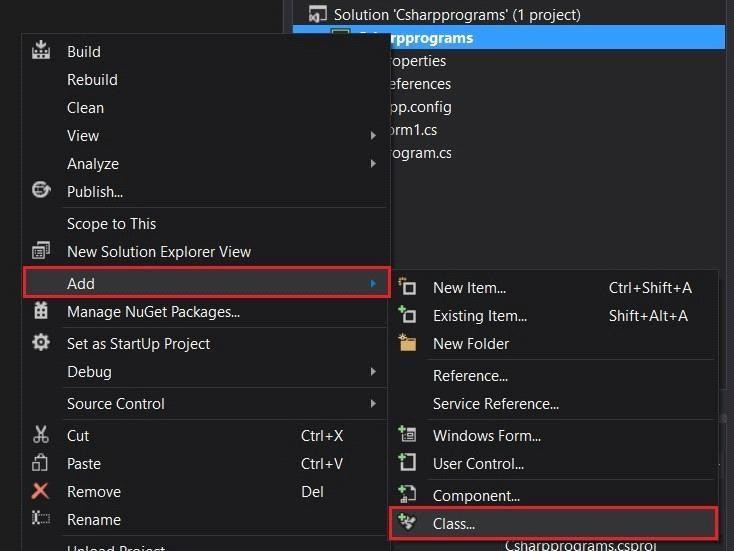


**After click on button**

**Q5. Write a Program to create a class to calculate simple interest and get and set method Steps-**

Open visual studio -> file-> new -> project -> select visual c# -> window form application -> save (Cshapprograms.cs)

Right click on project -> add -> class



**INPUT-**

**Filename- ClassSI.cs** using System; using System.Collections.Generic;

using System.Linq; using System.Text; using System.Threading.Tasks; namespace Csharpprograms

{ class ClassSI

{ int p,

y; double

r; public int year

{ get

{ return y;

} set { y = value;

}

}

public int Princ

{ get

{ return p;

} set { p = value;

}

}

public double Interrate

{

get {

return r;

} set

{ r

= value;

}

}

public double simpleInterest()

{

double sI = (p \* r \* y) / 100; return sI;

}

}

}

**Filename- simpleInt.cs** using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms; namespace Csharpprograms

{ public partial class simpleInt : Form

{

public simpleInt()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

ClassSI s = new ClassSI();

s.year = Convert.ToInt32(textBox3.Text); s.Princ = Convert.ToInt32(textBox1.Text); s.Interrate = Convert.ToDouble(textBox2.Text);

MessageBox.Show(s.year.ToString());

MessageBox.Show(s.Princ.ToString()); MessageBox.Show(s.Interrate.ToString()); double interest; interest

= s.simpleInterest();

MessageBox.Show(interest.ToString()); label5.Text = "SIMPLE INTEREST IS : " + interest.ToString();

}

private void simpleInt\_Load(object sender, EventArgs e)

{

}

}

}

**Filename- Program.cs** using System; using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace Csharpprograms

{ static class Program

{

/// <summary>

/// The main entry point for the application. /// </summary> [STAThread] static void Main()

{

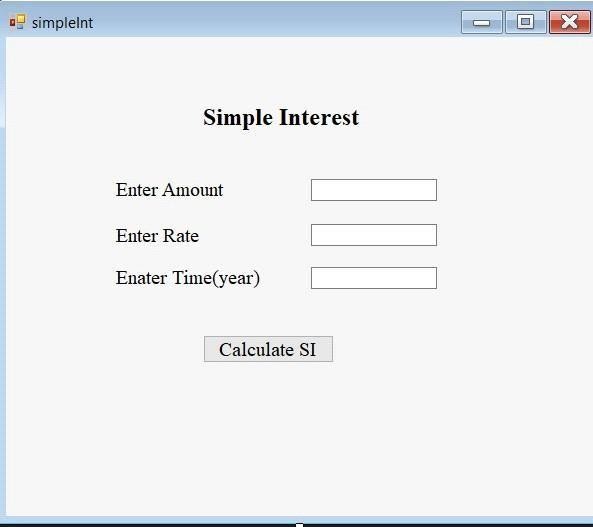
Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new simpleInt());

}

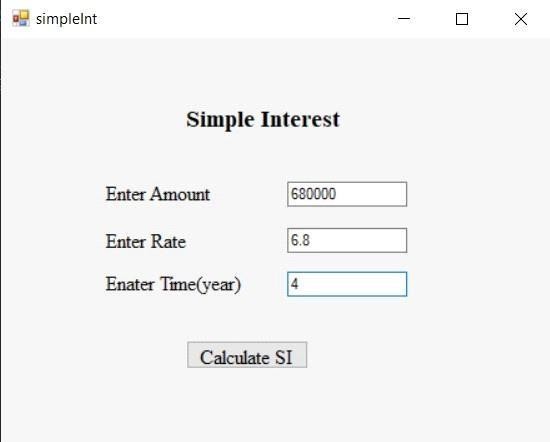
}

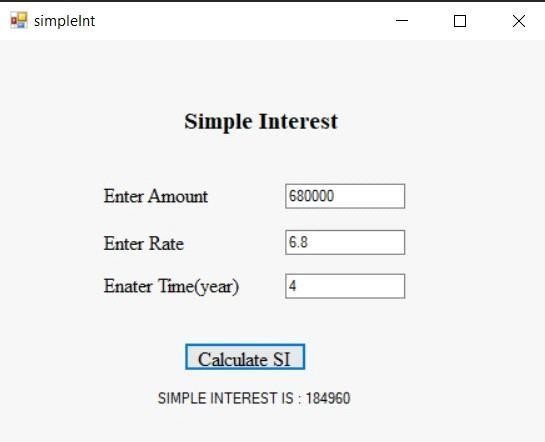
}

OUTPUT-



**After run file**





**Q6. Write a Program using C# to create a class to calculate to implement Single level inheritance Steps-**

Open visual studio -> file-> new -> project -> select visual c# -> window form application -> save (Cshapprograms.cs)

Right click on project -> add -> class -> file name (ClsEMP.cs)

**INPUT- Filename- ClsEMP.cs**

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

namespace Csharpprograms

{ class ClsEMP

{

public string name; public ClsEMP(string name)

{

this.name = name;

}

}

class salary : ClsEMP

{

public double salaryvar;

// string n;

public salary(string name, double sal) : base(name)

{

salaryvar = sal;

}

public double getsal()

{

double cal\_sal = salaryvar + 0.4\*salaryvar + 0.3\*salaryvar; return cal\_sal;

}

public string getname() // writing extra fo accessing name variable in form

{

return name;

}

}

}

**Filename- simpleInt1.cs** using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;

namespace Csharpprograms

{ public partial class simpleInt1 : Form

{

public simpleInt1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

salary s1 = new salary(textBox1.Text, Convert.ToInt32(textBox2.Text));// two parameter passed double sal = s1.getsal(); string name = s1.getname(); //txtname.Text =

name; label4.Text = "Name: "+ name + "Your Simple Interst: " + Convert.ToString(sal);

}

}

}

**Filename- Program.cs** using System;

using System.Collections.Generic; using System.Linq; using System.Threading.Tasks; using System.Windows.Forms;

namespace Csharpprograms

{ static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary> [STAThread] static void Main()

{

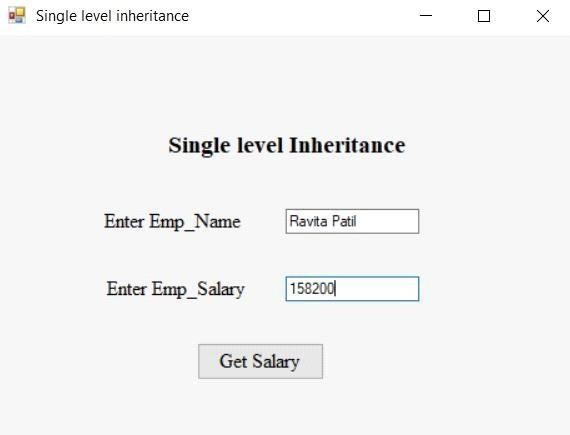
Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new simpleInt1());

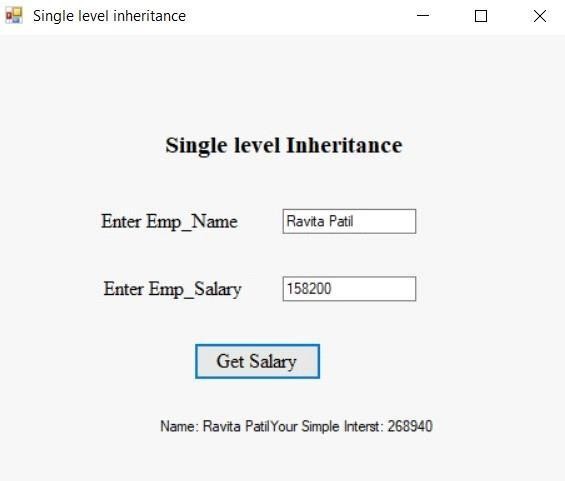
}

}

}

**OUTPUT-**



**After click on button**

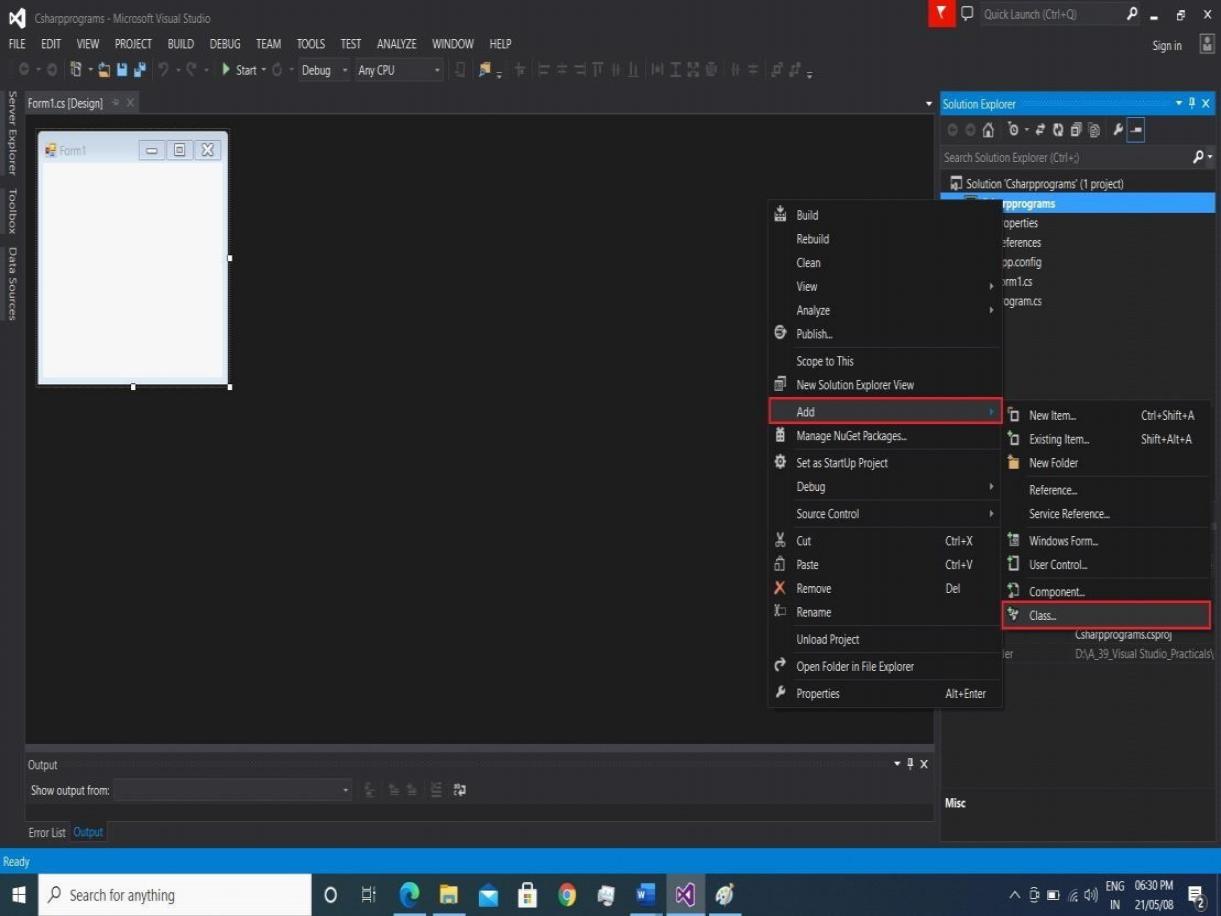
* 1. **WAP in C# to implement hierarchical level inheritance**

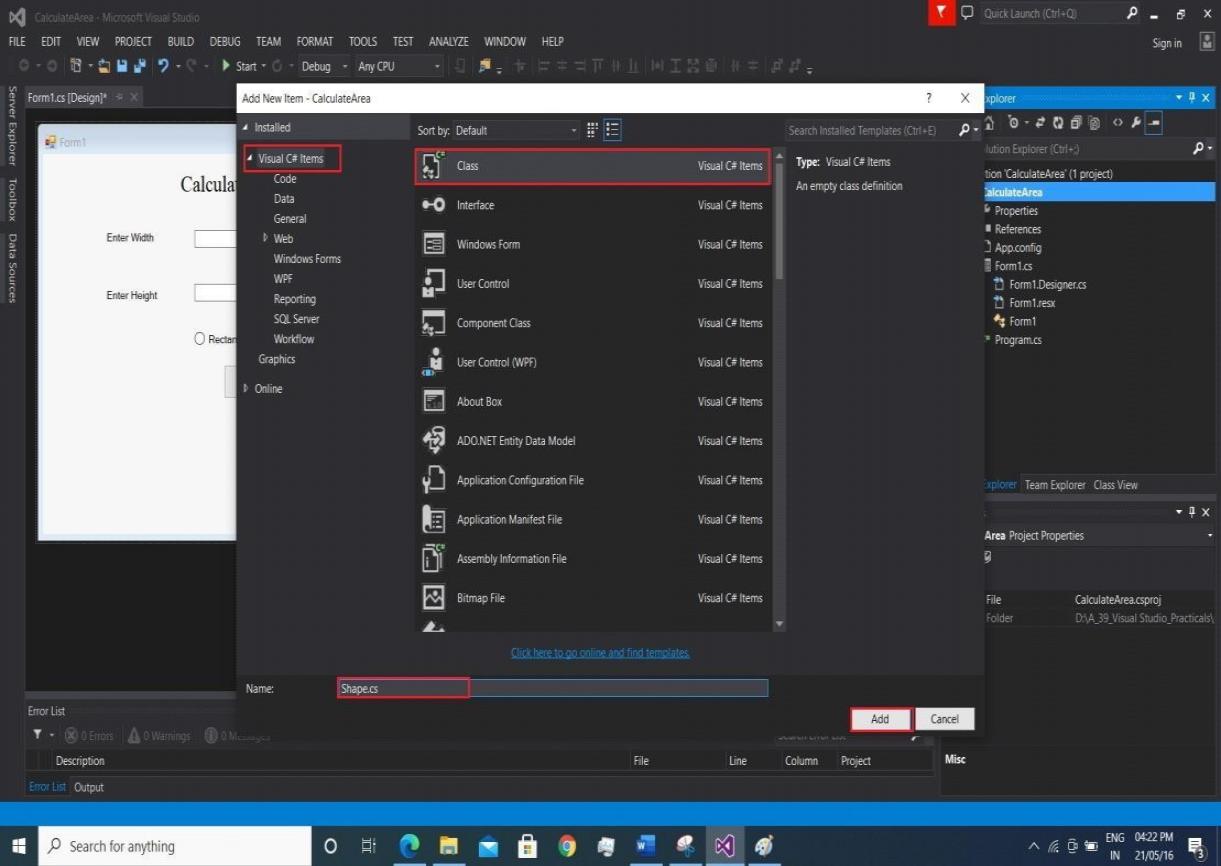
Open visual studio -> file-> new project -> select visual c# -> window form application -> save (calculator.cs)





Right click on project -> add -> class -> file name (ClsEMP.cs)





**Filename- class file(Shape.cs)**

using System;

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

using System.Text;

using System.Threading.Tasks;

namespace CalculateArea

{

class Shape

{

protected int width, height;

public void setwidth(int w)

{

width = w;

}

public void setheight(int h)

{

height = h;

}

}

class Rectangle : Shape

{

public int getarea()

{

return (width \* height);

}

}

class Square : Shape

{

public int getarea()

{

return (width \* width);

}

}

}

**Filename-form1.cs**

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

namespace CalculateArea

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void calculate\_Click(object sender, EventArgs e)

{

if (radioRectangle.Checked)

{

Rectangle r = new Rectangle(); r.setheight(Convert.ToInt32(textBox1.Text)); r.setwidth(Convert.ToInt32(textBox2.Text)); label4.Text = " Area of Rectangle is: " + r.getarea();

}

if (radioSquare.Checked)

{

Square s = new Square(); s.setwidth(Convert.ToInt32(textBox1.Text)); label4.Text = "Area of Square is: " + s.getarea();

}

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void radioRectangle\_CheckedChanged(object sender, EventArgs e)

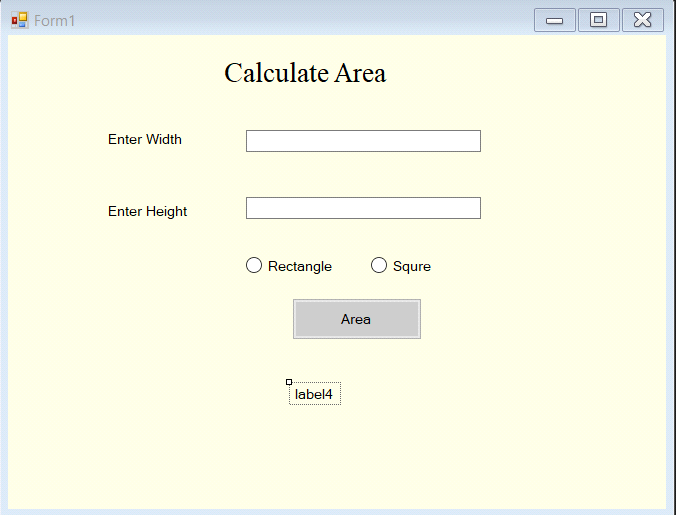
{

}

}

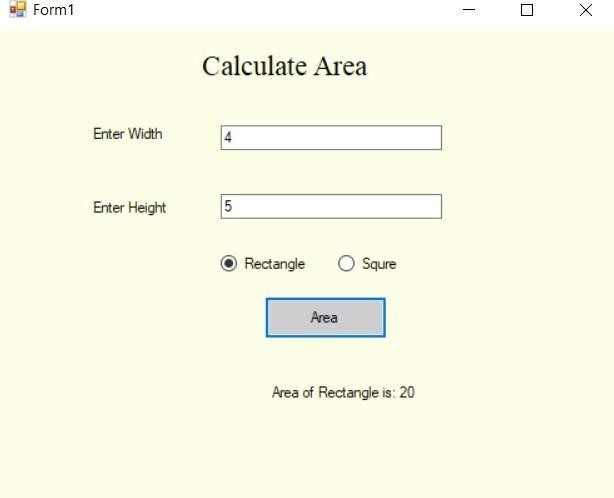
}

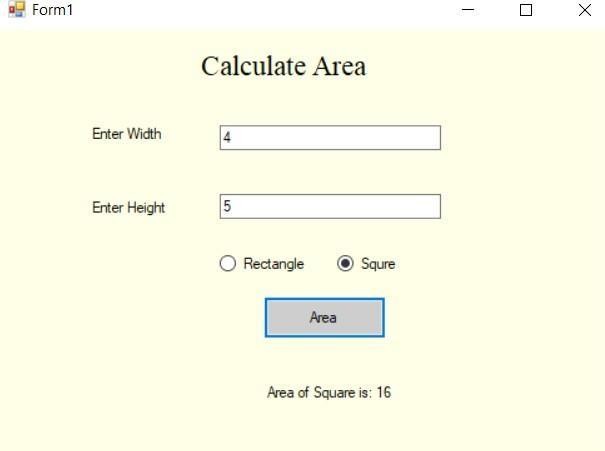
**Design-**



## OUTPUT-

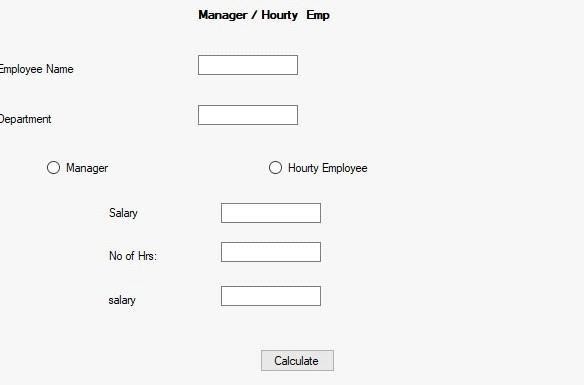
**After click on button**





* 1. **WAP in C# to implement Abstract class.**

**WAP in C# to design Salary class with appropriate properties and inherit the class into Manager and Hourly Employee to calculate their salary(consider da = 0.5, pf = 0.3, ta = 0.2)**



**Class Salary:**

using System;

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

using System.Threading.Tasks;

namespace salary

{

abstract class Salary

{

string name, dname;

public string Name

{ get {

return name;

} set { name = value;

} }

public string Dname

{ get { return dname;

} set { dname = value;

}

}

public abstract double salary(string type);

}

class Manager : Salary

{

double basic;

double da = 0.5, pf = 0.3, ta = 0.2; public double Basic

{ get { return basic;

} set { basic = value;

} }

public override double salary(string type)

{ string t = type; if (t == "Manager")

{

return basic + (basic \* da) + (basic \* ta) - (basic

\* pf);

}

return 0;

}

}

class HEmp : Salary

{ int hrs; public int Hrs

{ get { return hrs;

} set { hrs = value;

} }

public override double salary(string type)

{ string t = type; if (t == "HEmp")

{

return hrs \* 500;

} return 0;

}

}

}

**.cs File:**

using System;

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

namespace salary

{

public partial class Form1 : Form

{

RadioButton r = new RadioButton();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

label4.Enabled = false; label5.Enabled = false; txtbasic.Enabled = false; txthr.Enabled = false;

}

private void click(object sender, EventArgs e)

{

r = (RadioButton)sender; if (r.Name == "rdhremp")

{

label5.Enabled = true; txthr.Enabled = true;

label4.Enabled = false; txtbasic.Enabled = false;

}

if (r.Name == "rdmanager")

{

label4.Enabled = true; txtbasic.Enabled = true; label5.Enabled = false; txthr.Enabled = false;

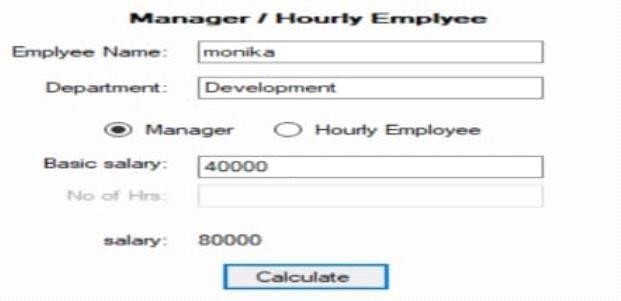
}

}

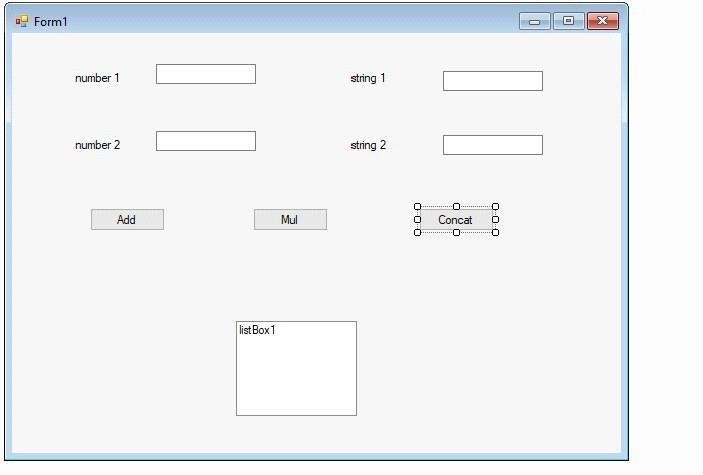
}

}

**Output:**



* 1. **WAP in C# to implement Interface class.**



**Add Interface:**

using System;

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

using System.Threading.Tasks;

namespace multiinheritance

{

interface Interface1

{

int ADD(int num1, int num2); int Mul(int num1, int num2);

}

interface IStringOperation

{

string concat(string fname, string lname);

}

}

**Add Class:**

using System;

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

namespace multiinheritance

{

class multiinher

{

public int ADD(int num1, int num2)

{

return num1 + num2;

}

public int Mul(int num1, int num2)

{

return num1 \* num2;

}

public string concat(string f1, string f2)

{

return (f1 + f2);

}

}

}

**.cs File:**

using System;

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

namespace multiinheritance

{

public partial class Form1 : Form

{

multiinher obj = new multiinher();

public Form1()

{

InitializeComponent();

}

private void label3\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e) {

}

private void button1\_Click(object sender, EventArgs e)

{ listBox1.Items.Add("Addition :" + obj.ADD(Convert.ToInt16(textBox1.Text), Convert.ToInt16(textBox2.Text)).ToString());

}

private void button2\_Click(object sender, EventArgs e)

{

listBox1.Items.Add("Multiplication :" + obj.Mul(Convert.ToInt16(textBox1.Text), Convert.ToInt16(textBox2.Text)).ToString());

}

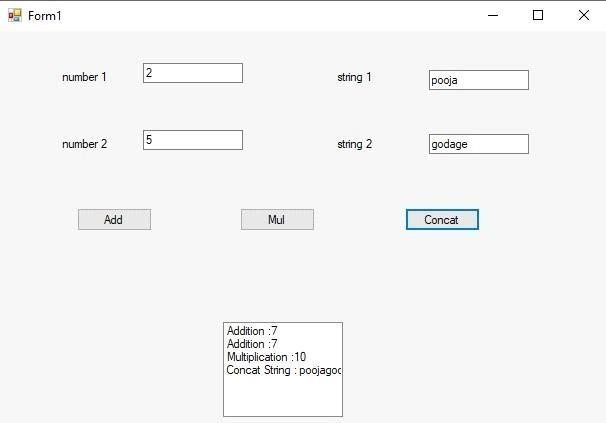
private void button3\_Click(object sender, EventArgs e) { listBox1.Items.Add("Concat String : " + obj.concat(textBox3.Text, textBox4.Text));

}

}

}

**Output:**



In this type of inheritance, the multiple classes derives from one base class.

# Module: Introduction to ASP.NET

**1)Design ASP.NET web page to demonstrate page life cycle.**

Input:



Code:

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace lifecycle

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_PriInit(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am PreInit Event ";

}

protected void Page\_Init(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am Init Event ";

}

protected void Page\_InitComplete(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am InitComplete Event ";

}

protected override void OnPreLoad(EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am preload Event ";

}

protected void Page\_Load(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am Load Event ";

}

protected void Page\_LoadComplete(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am LoadComplete Event ";

}

protected override void OnPreRender(EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am prerend er Event ";

}

protected void Page\_Unload(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am UnLoad Event ";

}

protected void Button1\_Click(object sender, EventArgs e)

{

Label1.Text = Label1.Text + "<br/>" + "I am button\_click event ";

}

protected void Button1\_Click1(object sender, EventArgs e)

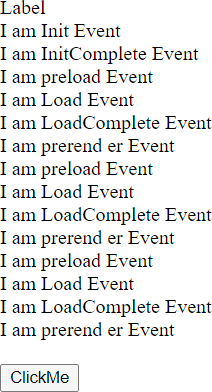
{

}

}

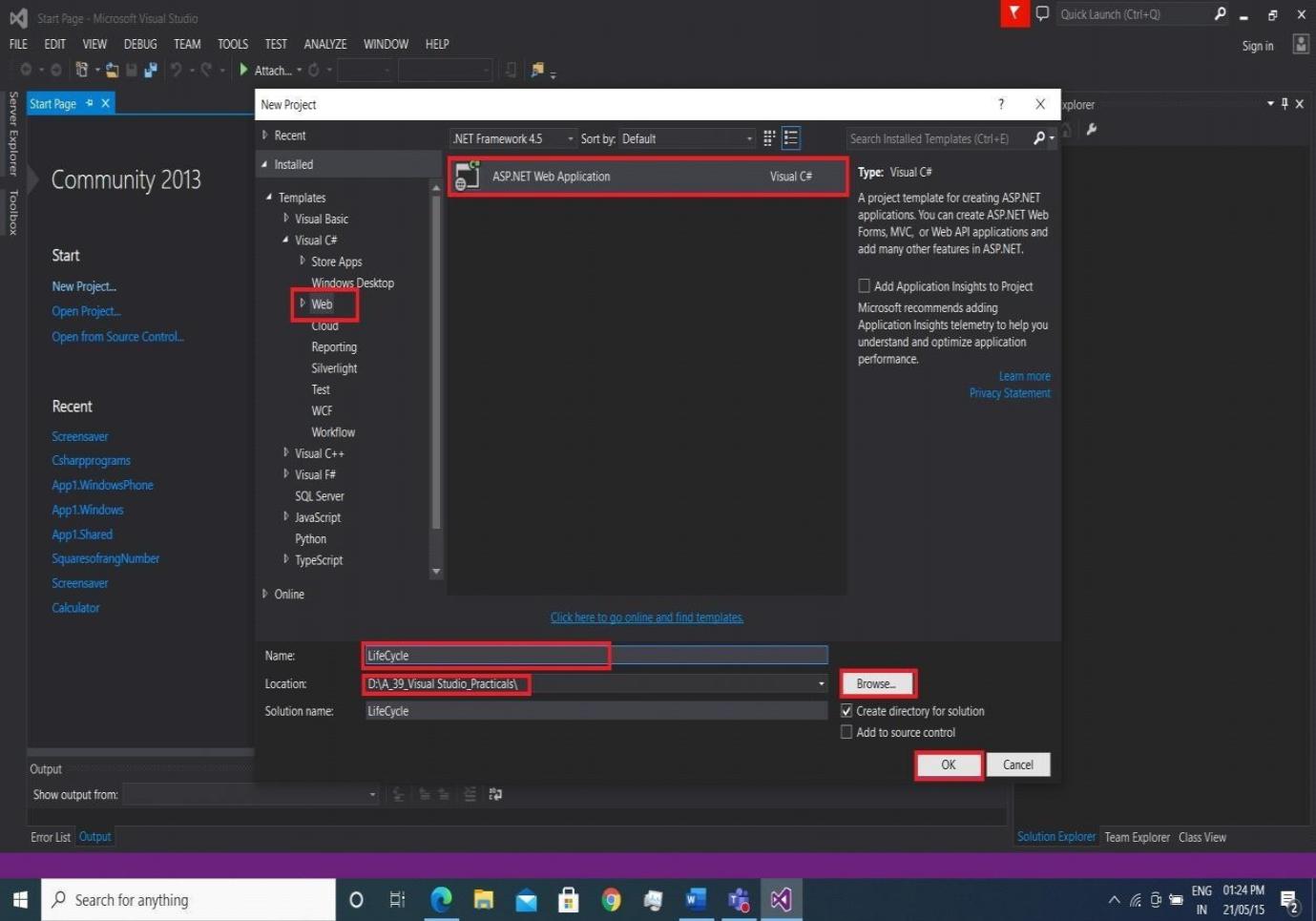
}

Output:

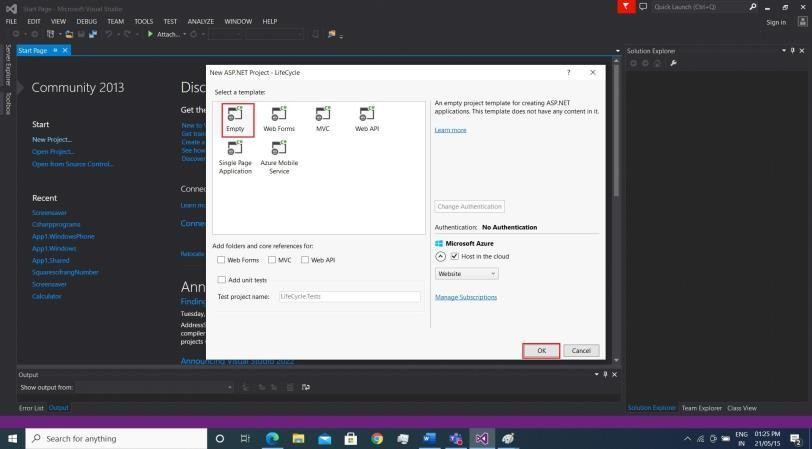


1. **Design registration from using Web server controls Steps-**

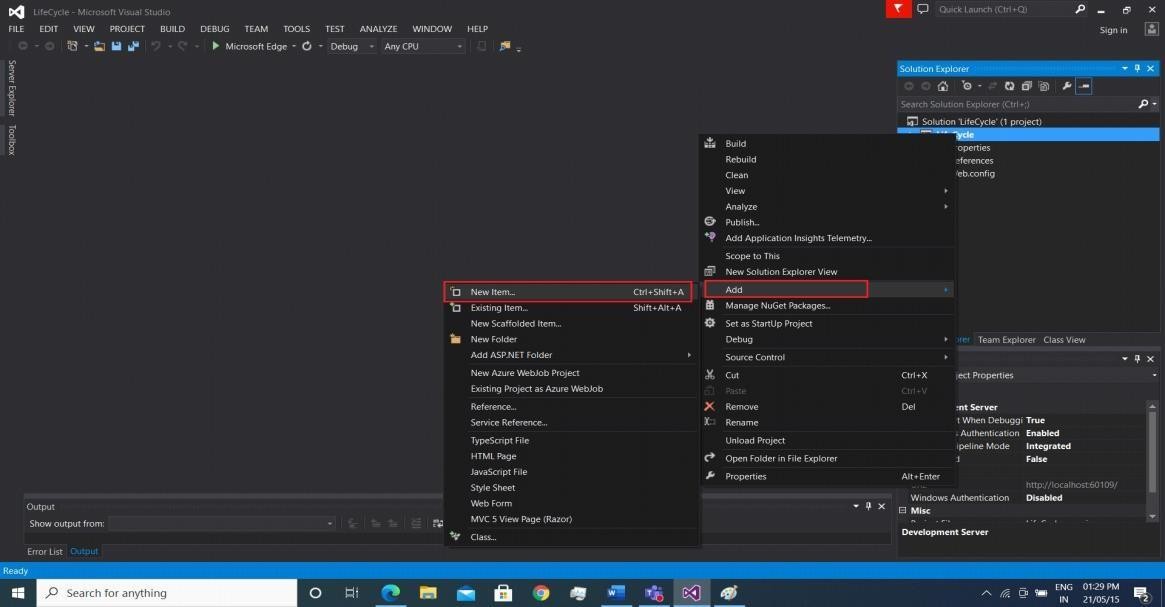
Open visual studio -> file-> new -> project -> select web -> ASP.NET Web application -> save (LifeCycle)

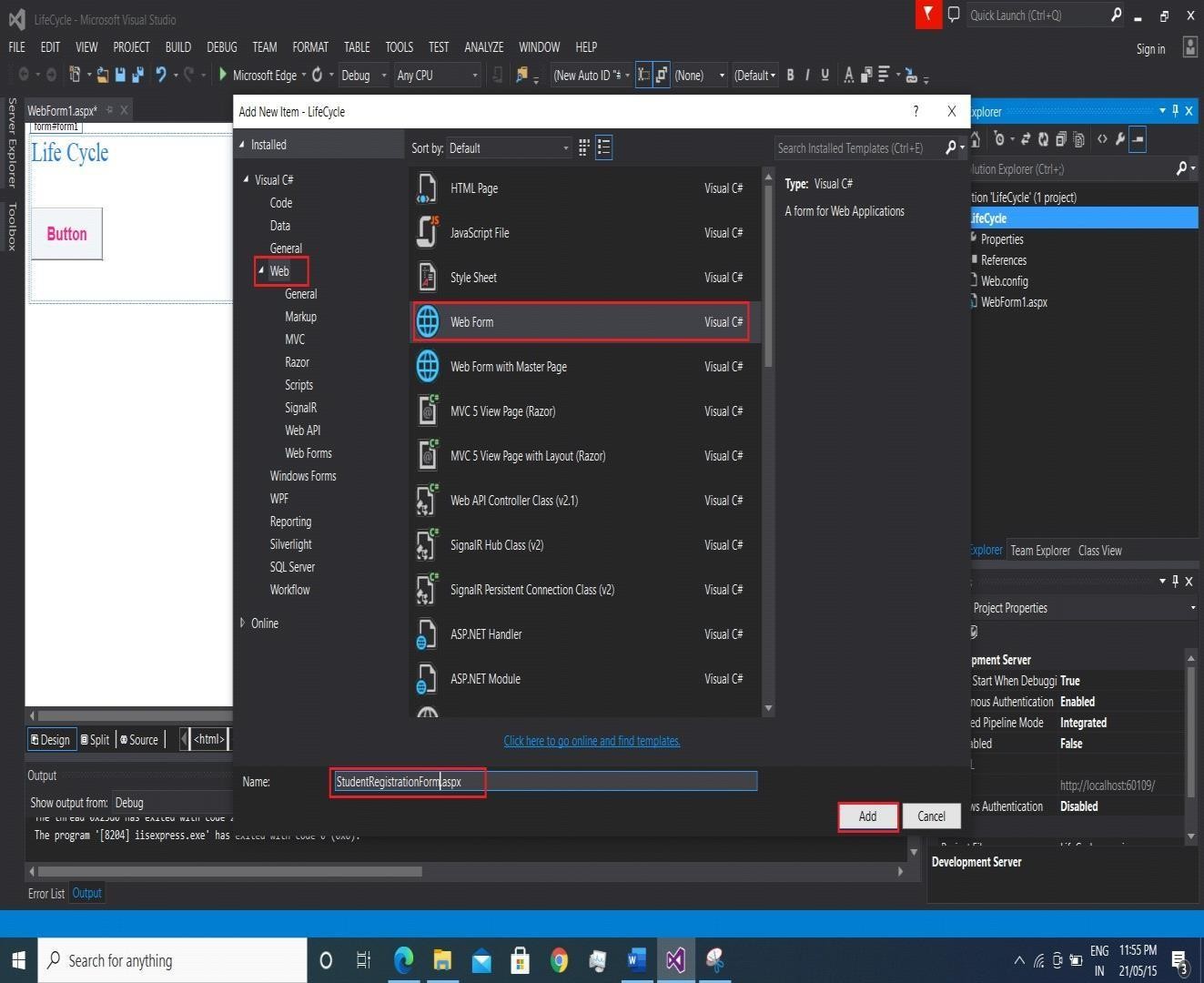


Select Empty -> ok



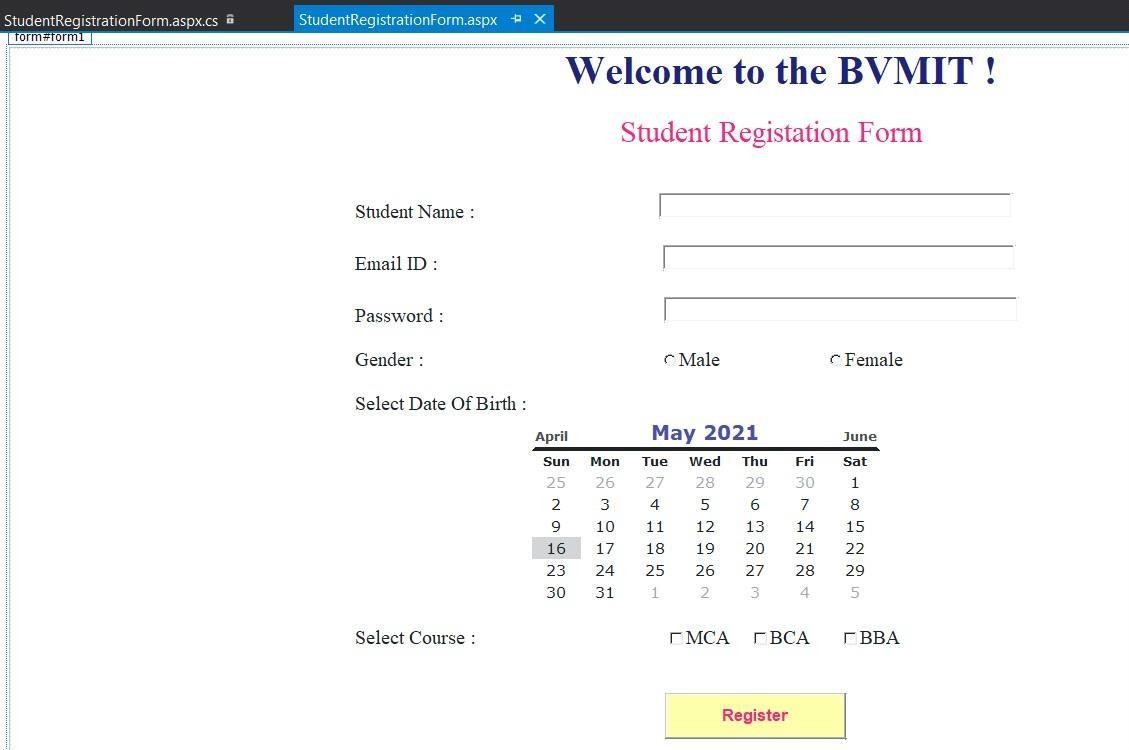
Right click on project -> add -> new item -> web -> web form -> ok





## INPUT-

**Design-**





**Filename-StudentRegistrationForm.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace LifeCycle

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

msg.Text = "Hello " + TxtName.Text + " ! ";

msg.Text = msg.Text + " You have successfuly Registered with the following details."; username1.Text = "Name is:" + TxtName.Text;

emailid1.Text = "Email id :" + TxtEmail.Text; gender1.Text = "Gender :" + RadioButton1.Text;

dob1.Text = "dob :" + Calendar1.SelectedDate.ToString("D"); course1.Text = "Course :" + CheckBox1.Text;

if (RadioButton1.Checked)

{

gender.Text = RadioButton1.Text;

}

else gender.Text = RadioButton2.Text; var courses = "";

if (CheckBox1.Checked)

{

courses = CheckBox1.Text + " ";

}

if (CheckBox2.Checked)

{

courses += CheckBox2.Text + " ";

}

if (CheckBox3.Checked)

{

courses += CheckBox3.Text;

}

course.Text = "selected course:" + courses;

dob.Text = "Your DOB is: " + Calendar1.SelectedDate.ToString("D"); TxtName.Text = "";

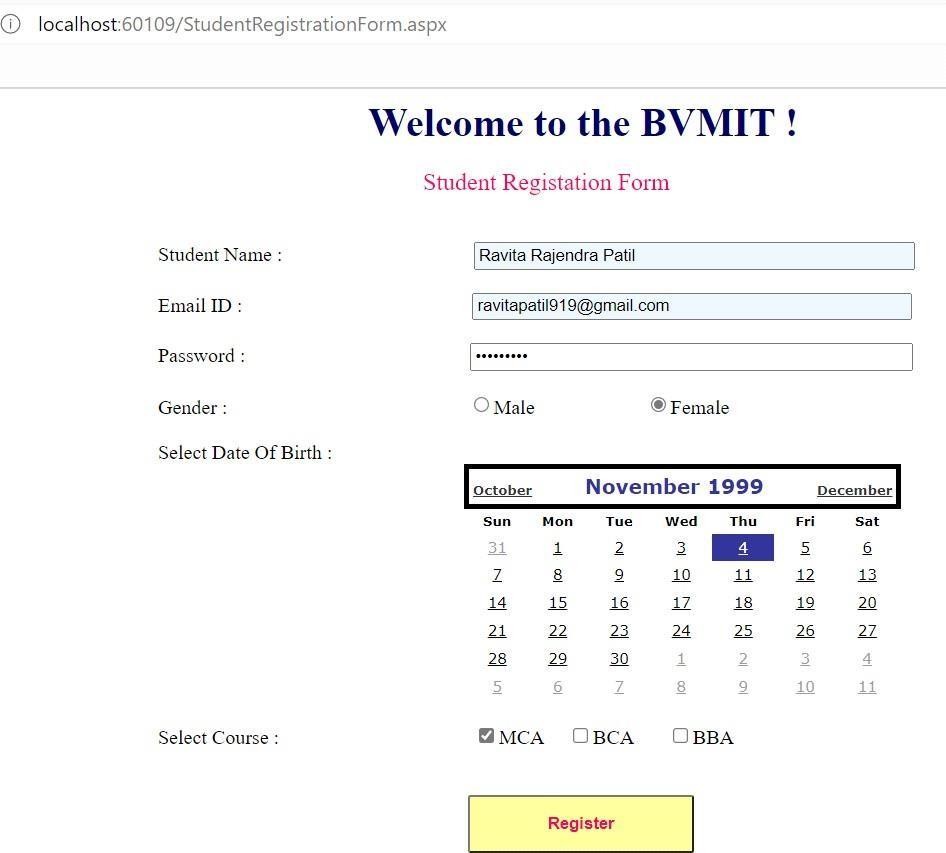
TxtEmail.Text = ""; TextBox3.Text = ""; RadioButton1.Checked = false; RadioButton2.Checked = false; CheckBox1.Checked = false; CheckBox2.Checked = false; CheckBox3.Checked = false;

}

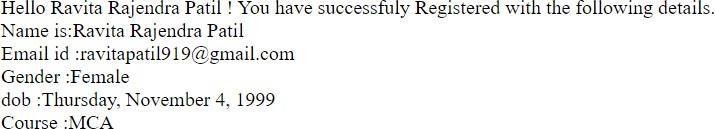
}

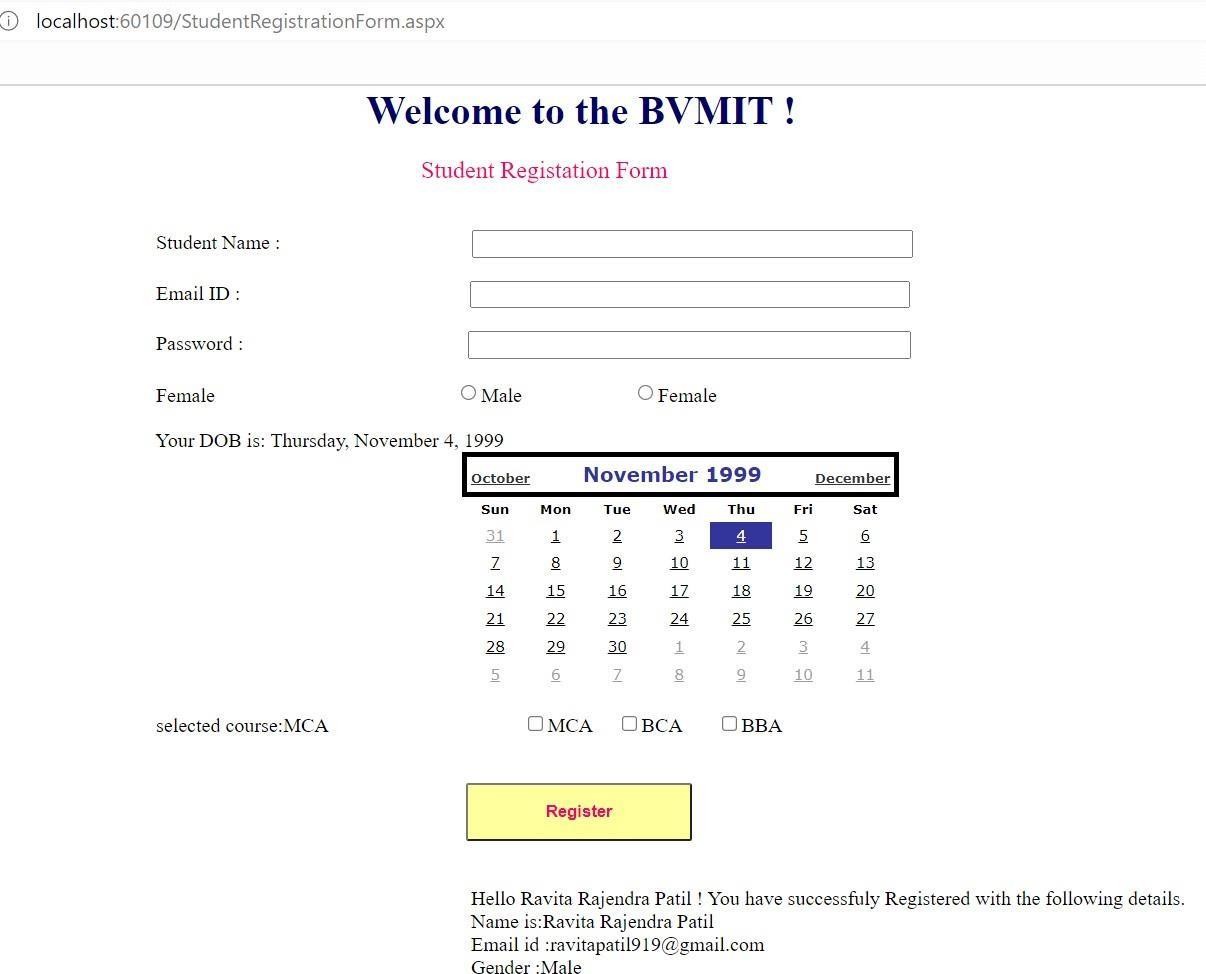
}

## OUTPUT-



**After click on Register button**



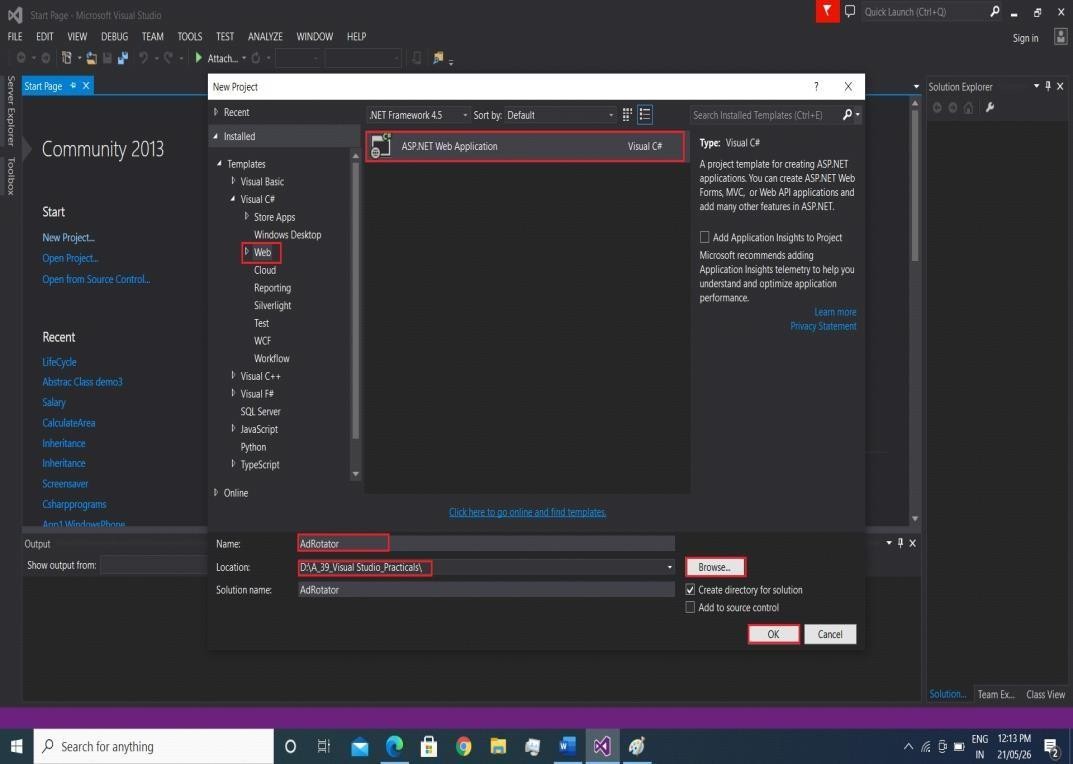


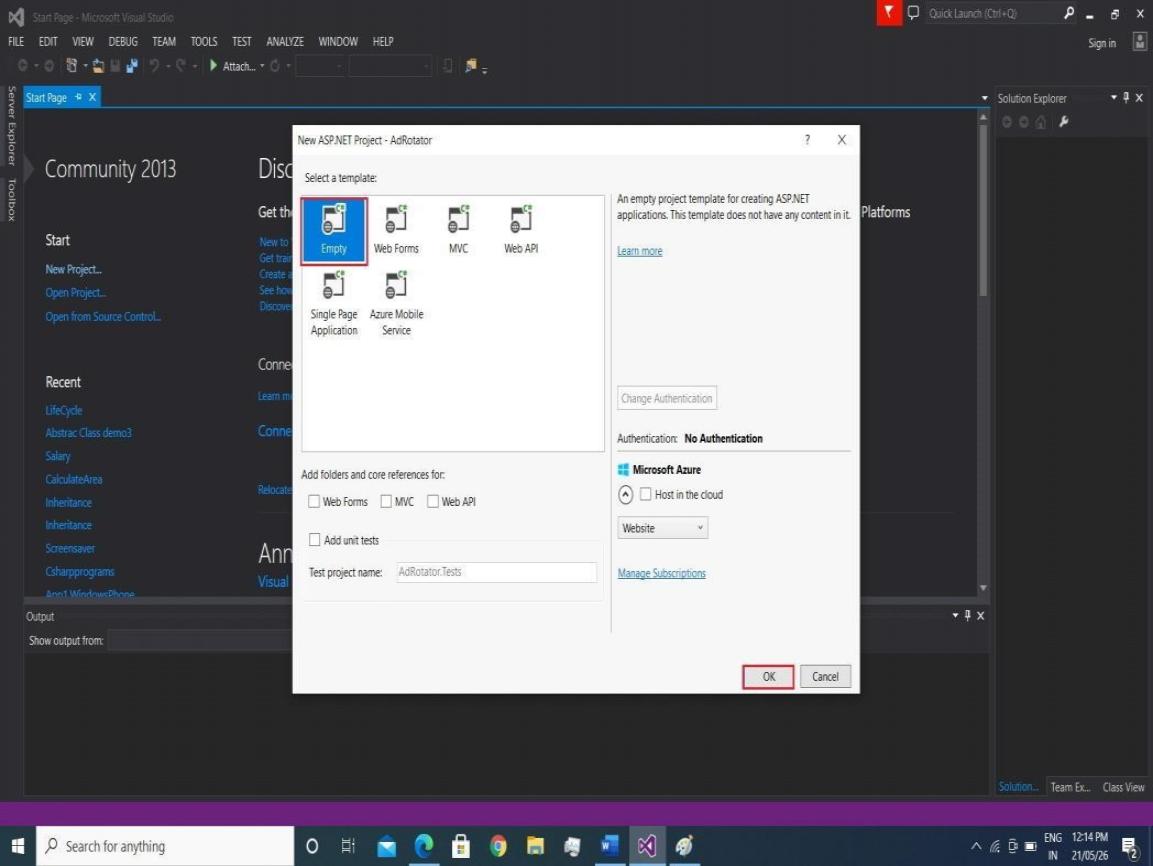
1. **Design an ASP.NET Application to Display Random Advertisements using ADRotator Control**

**Steps-**

Open visual studio -> file-> new project -> select web -> web form application -> save(AdRotator) -> select empty -> ok

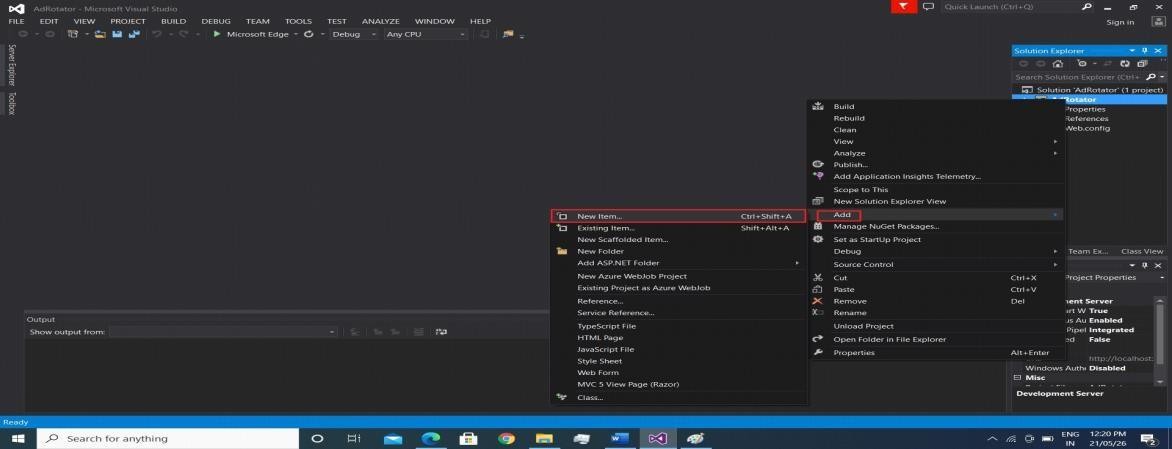


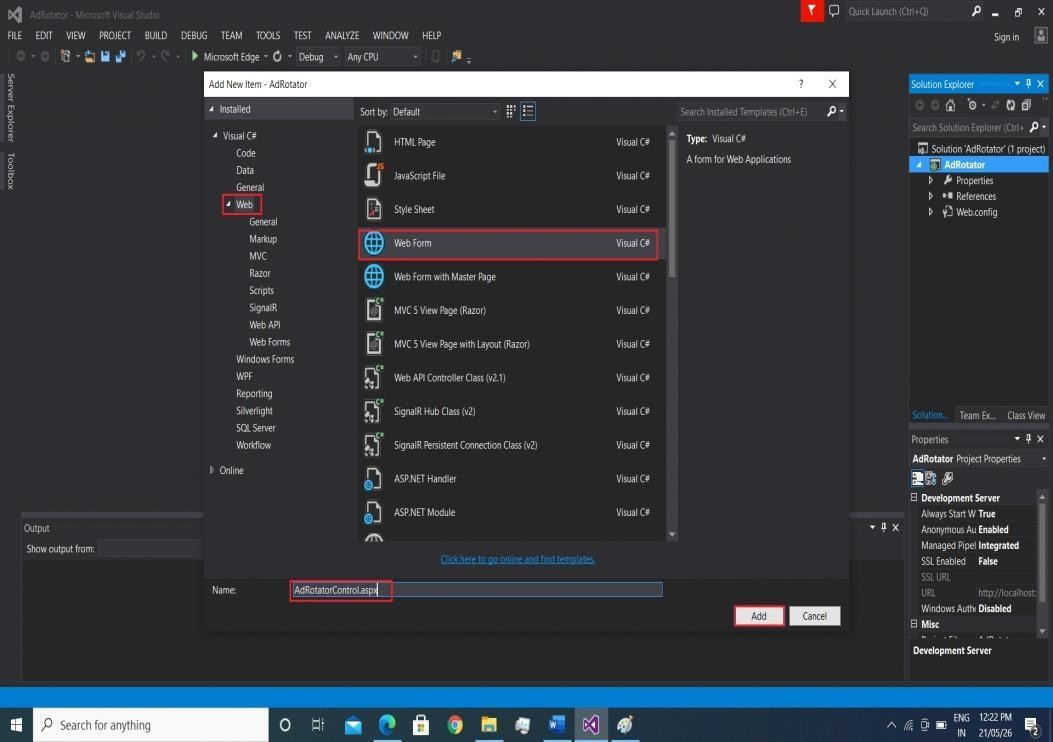




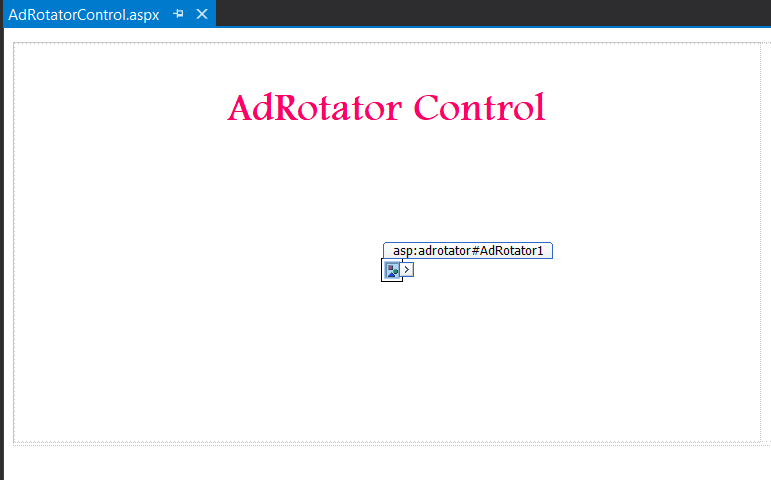
**Create New Web page for display AdRotator control.**

**Right Click on Project-> add -> New Iteam -> Web Form(AdRotatorControl.aspx)**

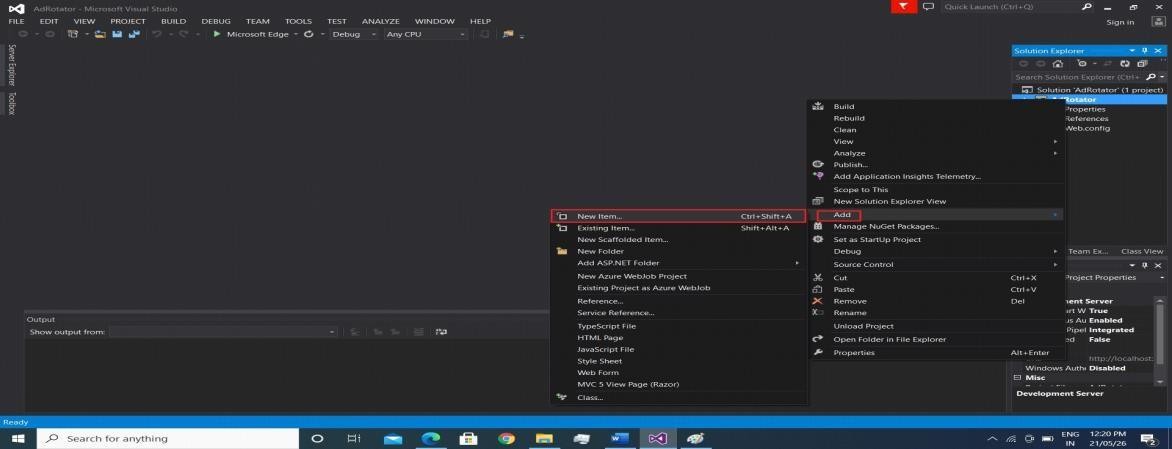


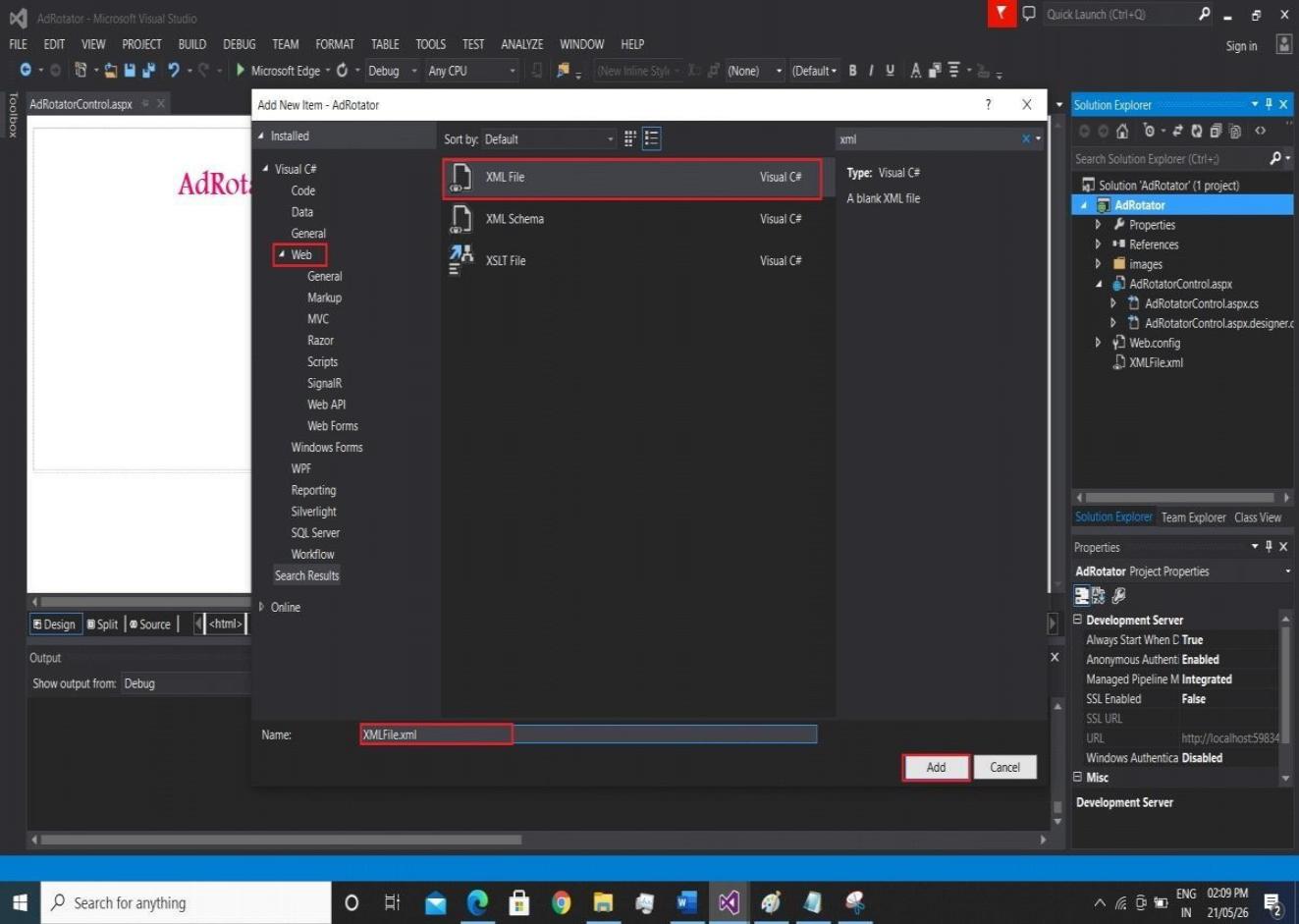


**Drag and Drop AdRotator Control on web page from toolbox.**



**Right click on project-> Add -> new item -> Add new XML file in project for write advertisement details.**





**Write code in xml file for advertisement. INPUT-**

**Filename- XMLFile.xml**

<?xml version="1.0" encoding="utf-8"?>

<Advertisements>

<Ad>

<ImageUrl>~\images\1.jpg</ImageUrl>

[<NavigateUrl>http://bvimit.co.in/bvimit</NavigateUrl>](http://bvimit.co.in/bvimit)

<AlternateText> BVIMIT </AlternateText>

<Impressions>30</Impressions>

<Keyword>BVIMIT</Keyword>

</Ad>

<Ad>

<ImageUrl>~\images\flipkart.png</ImageUrl>

[<NavigateUrl>https://www.flipkart.com</NavigateUrl>](http://www.flipkart.com/)

<AlternateText>Flipkart</AlternateText>

<Impressions>20</Impressions>

<Keyword>FLIPKART</Keyword>

</Ad>

<Ad>

<ImageUrl>~\images\google.png</ImageUrl>

[<NavigateUrl>https://www.google.com</NavigateUrl>](http://www.google.com/)

<AlternateText>Google</AlternateText>

<Impressions>30</Impressions>

<Keyword>google</Keyword>

</Ad>

<Ad>

<ImageUrl>~\images\amazon.jpg</ImageUrl>

[<NavigateUrl>https://www.amazon.in</NavigateUrl>](http://www.amazon.in/)

<AlternateText>Amazon</AlternateText>

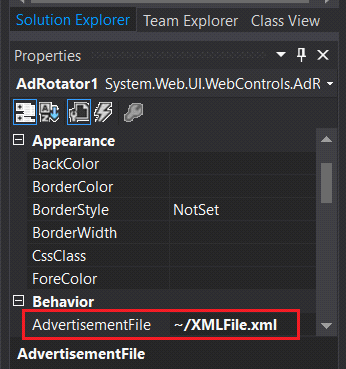
<Impressions>30</Impressions>

<Keyword>shopping site</Keyword>

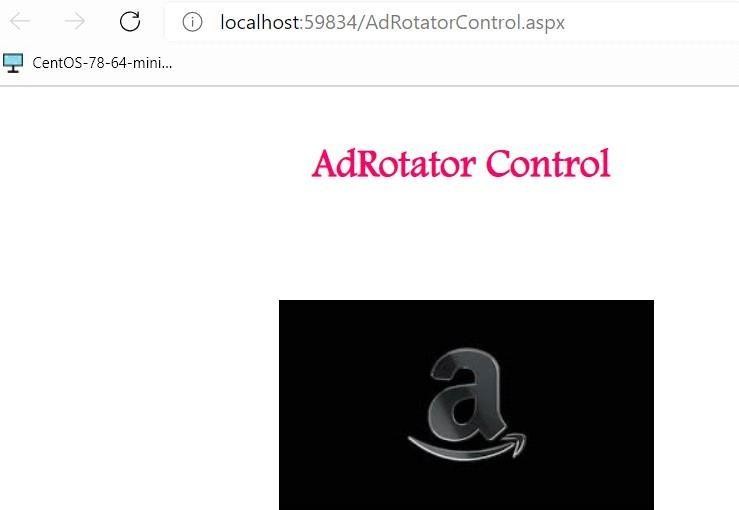
</Ad>

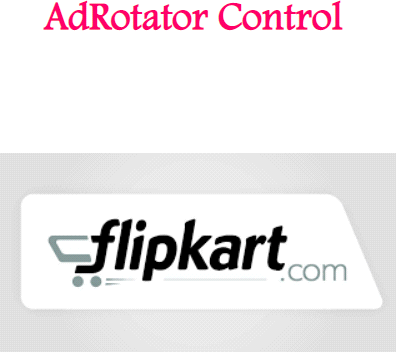
</Advertisements>

**Assign XML file to Advertisement File Properties of AdRotator control.**



## OUTPUT-



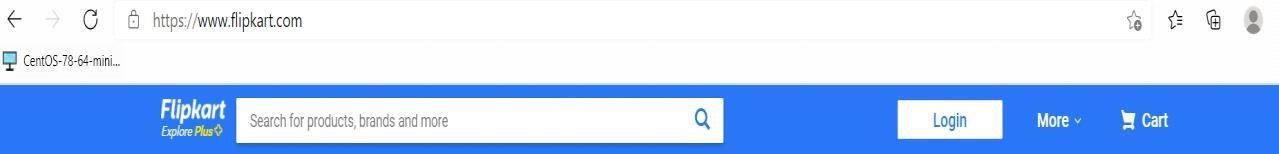
**After Click on imag - Amazon image**



**Click on -> google image**



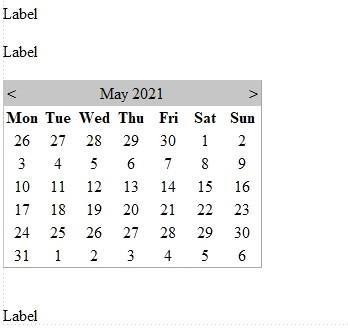
**Click on -> Flipkart image**



**After click on last image**



**4)Design an ASP.NET application to Display Current Month’s Calender. Render the calendar.Show Some image in the cell along message.**



**Program:**

using System;

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

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

Label2.Text = "Todays Date is: " + Calendar1.TodaysDate.ToShortDateString();

}

protected void Calendar1\_SelectionChanged(object sender, EventArgs e)

{

Label1.Text = Calendar1.SelectedDate.ToLongDateString();

}

protected void Calendar1\_DayRender(object sender, DayRenderEventArgs e)

{

if (e.Day.Date == new DateTime(2021, 2, 14))

{

Literal l1 = new Literal(); Image img1 = new Image();

e.Cell.Width = 70;

e.Cell.Height = 70; e.Cell.Font.Italic = true; e.Cell.Font.Size = FontUnit.XLarge;

e.Cell.BackColor = System.Drawing.Color.LightPink;

img1.ImageUrl = "~\\1.jpg"; img1.Width = 30; img1.Height = 30;

l1.Text = "<br /><font size=2 color=Red>Holiday</font>"; e.Cell.Controls.AddAt(1, l1);

e.Cell.Controls.AddAt(0, img1);

}

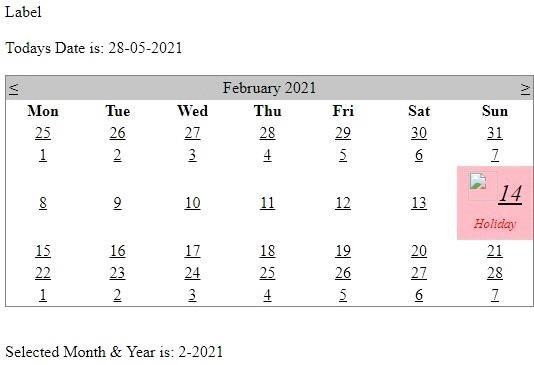
}

protected void Calendar1\_VisibleMonthChanged(object sender, MonthChangedEventArgs e) { Label3.Text="Selected Month & Year is: " + e.NewDate.Month + "-" + e.NewDate.Year;

}

}

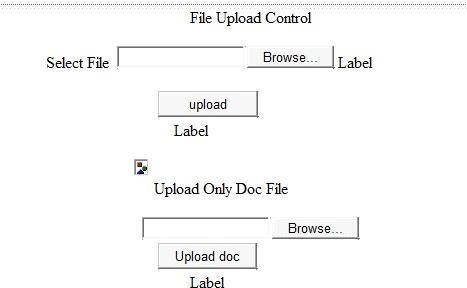
**Output:**



5)

1. **Design ASP.NET web page to demonstrate File Upload control**

**File Upload Control:**



**Program:**

using System;

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

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (FileUpload1.HasFile)

{

FileUpload1.SaveAs(Server.MapPath("~/img/") + FileUpload1.FileName); path.ImageUrl = "~/img/" + FileUpload1.FileName;

Label2.Text = "Image Uploaded Successfully !!";

} else

{

Label3.Text = "Select image first !!";

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

string fn = Path.GetFileName(FileUpload2.PostedFile.FileName);

string fextn = Path.GetExtension(fn); int size = FileUpload2.PostedFile.ContentLength; if (FileUpload2.HasFile)

{

if (fextn.ToLower() != ".doc" && fextn.ToLower() != ".docx")

{

Labelmsg.Text = "only doc and docx file allowed";

}

else if (size < 10024)

{

Labelmsg.Text = "file should be less than 1mb";

} else

{

FileUpload2.SaveAs(Server.MapPath("~/Upload/" + fn)); Labelmsg.Text = "file uploaded";

} }

else

{

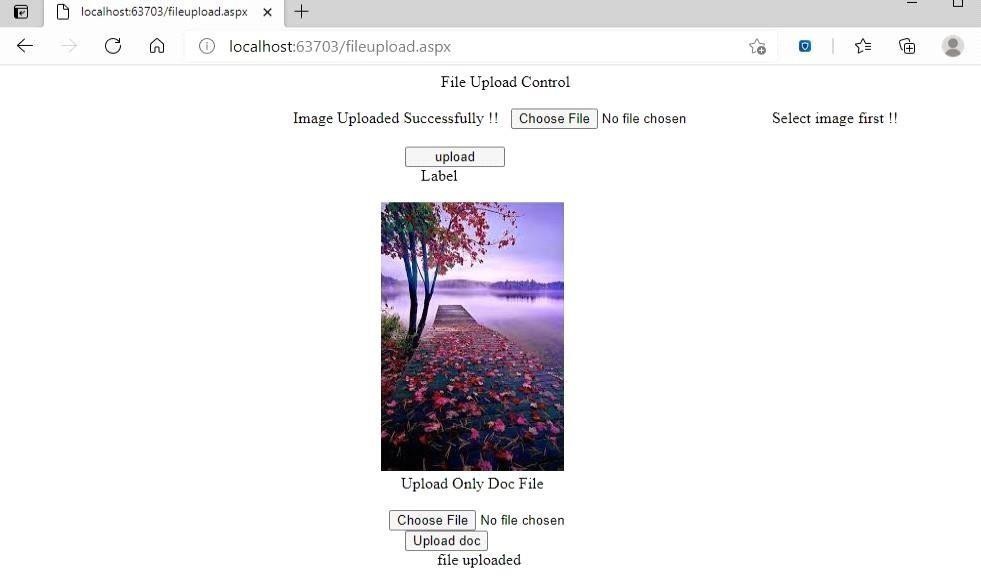
Labelmsg.Text = "please select file before uploading";

}

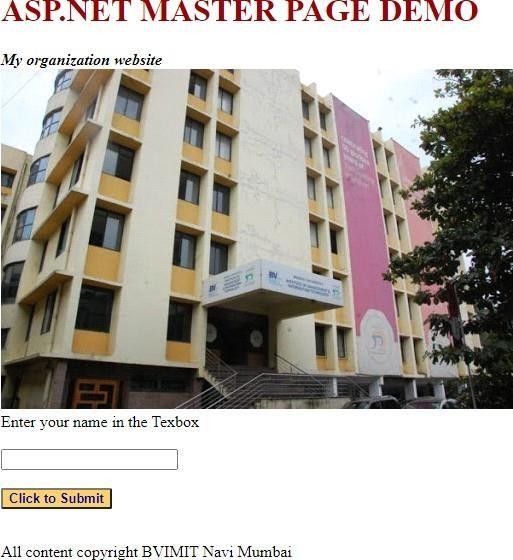
}

}

**Output:**



1. **Create website using master page concept.**



**master page:**

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

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml)

<head runat="server">

<title>Master Page</title>

<asp:ContentPlaceHolder id="head" runat="server">

</asp:ContentPlaceHolder>

<style type="text/css">

.style1 {

color: #990000;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<div id="header">

<h1 class="style1"> ASP.NET MASTER PAGE DEMO</h1>

</div>

<div id="slidebar">

<div id ="nav">

</div>

</div>

<div id="content">

<div class="itemcontent">

<strong> <em>My organization website <br />

<asp:Image ID="Image2" runat="server" ImageUrl="bv.JPG" />

<br />

</em></strong>

</div>

</div>

<asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

<div id="footer">

<p class="left">

All content copyright BVIMIT Navi Mumbai</p>

</div>

</form>

</body>

</html>

**Bvimit.aspx page:**

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

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

Enter your name in the Texbox<br />

<br />

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

<br />

<asp:Button ID="Button1" runat="server" BackColor="#FFCC66" Font- Bold="True"

Font-Size="Small" ForeColor="#000099" Text="Click to Submit" onclick="Button1\_Click" />

<br />

<br />

<asp:Label ID="Label1" runat="server" Text="Label" CssClass="title" Visible="False"></asp:Label>

</asp:Content>

**Bvimit.aspx.cs page:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

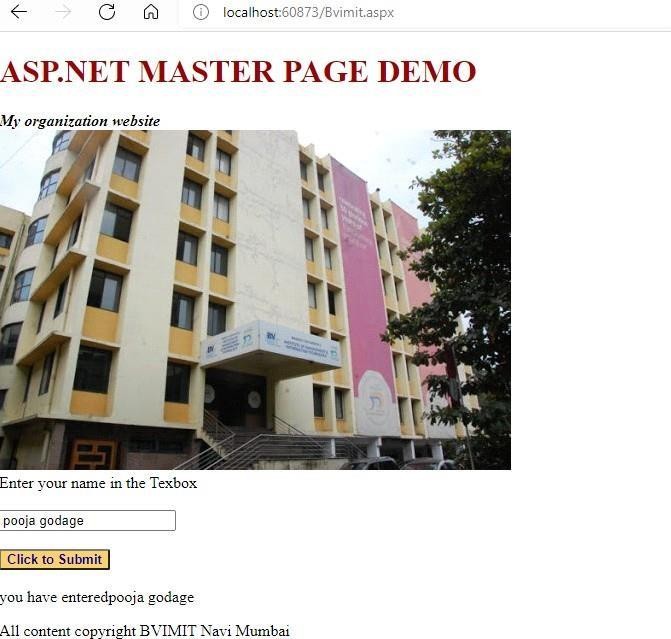
{

Label1.Text = "you have entered" + TextBox1.Text; Label1.Visible = true;

}

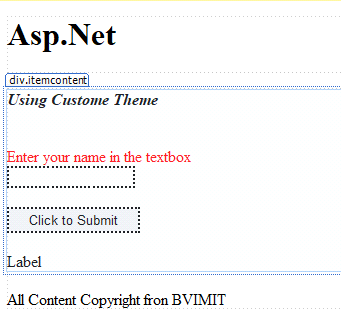
}

**Output:**



**Input:**

**Working with theme**



**skin.Skin Page:**

<%--

Default skin template. The following skins are provided as examples only.

1. Named control skin. The SkinId should be uniquely defined because duplicate SkinId's per control type are not allowed in the same

theme.

<asp:GridView runat="server" SkinId="gridviewSkin" BackColor="White" >

<AlternatingRowStyle BackColor="Blue" />

</asp:GridView>

1. Default skin. The SkinId is not defined. Only one default control skin per control type is allowed in the same theme.

<asp:Image runat="server" ImageUrl="~/images/image1.jpg" />

--%>

<asp:Label runat="server" Width="350" Height="50" Font-Bold="True" Font-Size="Larger" ForeColor="#CC0000" BackColor="#6699FF"/>

<asp:Button runat="server" ForeColor="#6666FF" BackColor="#CCCCCC"

/>

<asp:TextBox runat="server" Font-Bold="True" Font-Size="Medium" ForeColor="#0066FF" BackColor="#CCCCCC" Font-Italic="True"/>

**Skinthems.aspx Page:**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="skinthems.aspx.cs" Inherits="skinthems" Theme="Theme1" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml)

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<div id="header">

<h1>Asp.Net</h1>

</div>

<div id="slideshare">

<div id="nav">

</div>

</div>

<div id="content">

<div class="itemcontent">

<strong> <em> Using Custome Theme</em></strong><br

/><br />

<br />

<asp:Label ID="Label1" runat="server" Text="Enter your name in the textbox" ForeColor="Red"></asp:Label><br />

<asp:TextBox ID="TextBox1" runat="server" BorderStyle="Dotted"></asp:TextBox><br />

<br />

<asp:Button ID="Button1" runat="server" Text="Click to Submit" BorderStyle="Dotted" OnClick="Button1\_Click" />

<br />

<br />

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

</div></div>

<div id="footer">

<p class="left">

All Content Copyright fron BVIMIT

</p>

</div>

</div>

</form>

</body>

</html>

**Stylesheet.aspx:**

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="stylesheet.aspx.cs" Inherits="\_Default" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml)

<head runat="server">

<title></title>

<link href="StyleSheet.css" rel="Stylesheet" type="text/css" />

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label ID="Label2" runat="server" Text="using the CSS file in this Web Page."></asp:Label>

<br />

<br />

<asp:TextBox ID="TextBox2" runat="server" Width="269">This is a text Box</asp:TextBox>

</div>

</form>

</body>

</html>

**Stylesheet.css:**

\*{margin:0;}body {

color:green; background-color:aqua; font-family:sans-serif;

}

#Label1

{

font-size:18pt

}

p{

font-size:small

}

#form1{

background-color:lime; border-style:solid; border-width:2px; border-color:red; height:200px; width:600px;

}

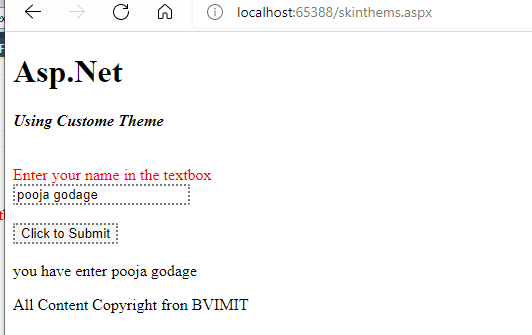
#TextBox1

{

background-color:orange; font-weight:bold; padding:3px;

}

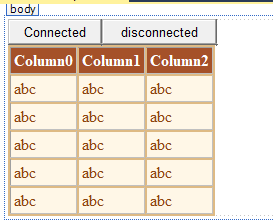
**Output:**



**Module:III): Database Programming in ASP.NET**

1. **Design a webpage For database connectivity (connected and disconnected connectivity)**

**Input:**



**.CS File:**

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.Data;

namespace WebApplication1

{

public partial class WebForm1 : System.Web.UI.Page

{

string StrSQL = "",StrConnection ="";

protected void Page\_Load(object sender, EventArgs e)

{

StrSQL = "SELECT \* FROM student"; StrConnection = @"Data

Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\stude nt.mdf;Integrated Security=True;Connect Timeout=30";

}

protected void Button1\_Click(object sender, EventArgs e)

{

using (SqlConnection objConn = new SqlConnection(StrConnection))

{

SqlCommand objCmd = new SqlCommand(StrSQL, objConn); objCmd.CommandType=CommandType.Text;

objConn.Open();

SqlDataReader objDr=objCmd.ExecuteReader(); GridView1.DataSource = objDr; GridView1.DataBind();

objConn.Close();

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

SqlDataAdapter objDa = new SqlDataAdapter(); DataSet objDs = new DataSet(); using(SqlConnection objConn = new

SqlConnection(StrConnection))

{

SqlCommand objCmd = new SqlCommand(StrSQL, objConn); objCmd.CommandType = CommandType.Text; objDa.SelectCommand = objCmd;

objDa.Fill(objDs, "stud"); GridView1.DataSource = objDs.Tables[0]; GridView1.DataBind();

}

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

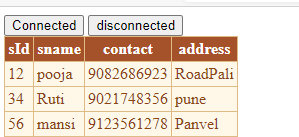
{

}

}

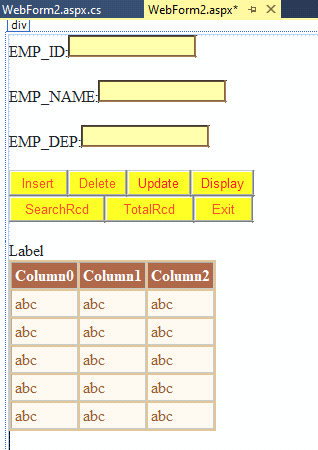
}

**Output:**



1. **Design a webpage to insert update, delete, search details using connect Architecture.**
   * **Insert record.**
   * **Search record**
   * **Update record.**
   * **Delete record.**

**Input:**



**.CS File:**

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.Data;

namespace WebApplication1

{

public partial class WebForm2 : System.Web.UI.Page

{

SqlConnection con; SqlCommand cmd; SqlDataReader dr;

protected void Page\_Load(object sender, EventArgs e)

{

con = new SqlConnection(@"Data

Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30");

}

private void Display()

{

con.Open();

cmd = new SqlCommand("select \* from Emptb", con); dr = cmd.ExecuteReader();

DataTable dt = new DataTable(); dt.Load(dr); GridView1.DataSource = dt; GridView1.DataBind(); con.Close();

}

protected void Button1\_Click(object sender, EventArgs e)

{

con.Open();

int aa = Convert.ToInt32(TextBox1.Text); string bb = TextBox2.Text;

string cc = TextBox3.Text;

cmd = new SqlCommand("INSERT INTO Emptb(Eid, Ename,Edept)VALUES ('" + aa + "','" + bb + "','" + cc + "')", con);

cmd.ExecuteNonQuery(); Response.Write("one record inserted:"); con.Close();

Display();

}

protected void Button2\_Click(object sender, EventArgs e)

{

+ "'", con);

con.Open();

int aa = Convert.ToInt32(TextBox1.Text);

cmd = new SqlCommand("DELETE FROM Emptb where Eid='" + aa

cmd.ExecuteNonQuery(); Response.Write("one record Delete:"); con.Close();

Display();

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

protected void Button3\_Click(object sender, EventArgs e)

{

con.Open();

int aa = Convert.ToInt32(TextBox1.Text); string bb = TextBox2.Text;

string cc = TextBox3.Text;

string abc = "UPDATE Emptb SET Eid ='" + aa + "', Ename

='" + bb + "',Edept ='" + cc + "' WHERE Eid = '" + aa + "'";

SqlCommand cmd = new SqlCommand(abc, con); cmd.ExecuteNonQuery();

Response.Write("one record updated:"); con.Close();

Display();

}

protected void Button4\_Click(object sender, EventArgs e)

{

Display();

}

protected void Button5\_Click(object sender, EventArgs e)

{

con.Open();

int aa = Convert.ToInt32(TextBox1.Text);

string abc = "SELECT Eid,Ename,Edept FROM Emptb where Eid='" + aa + "'";

cmd = new SqlCommand(abc, con); Response.Write("one record search:"); dr = cmd.ExecuteReader();

DataTable dt = new DataTable(); dt.Load(dr); GridView1.DataSource = dt; GridView1.DataBind(); con.Close();

}

protected void Button6\_Click(object sender, EventArgs e)

{

con.Open();

cmd = new SqlCommand("select Count(\*) from Emptb", con); int a = (int)cmd.ExecuteScalar();

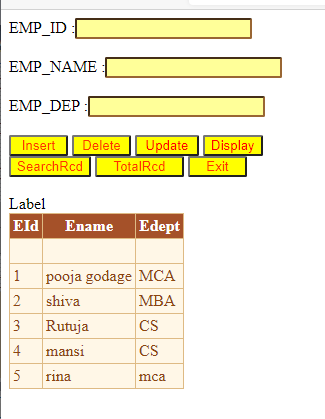
Label1.Text = "Total Record:--> " + a.ToString(); con.Close();

}

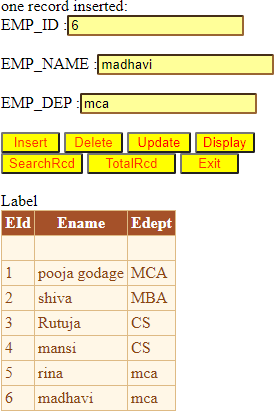
}

}

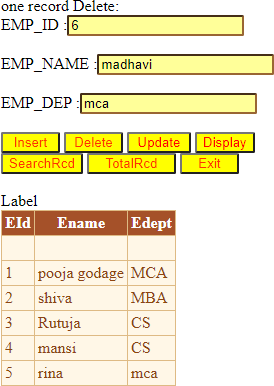
**Output: Display Record:**



**Insert Record:**



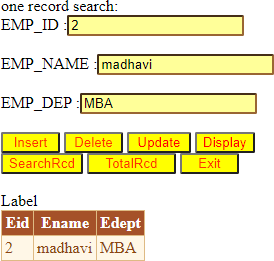
**Delete Record:**



**Update Record:**



**Search Record:**

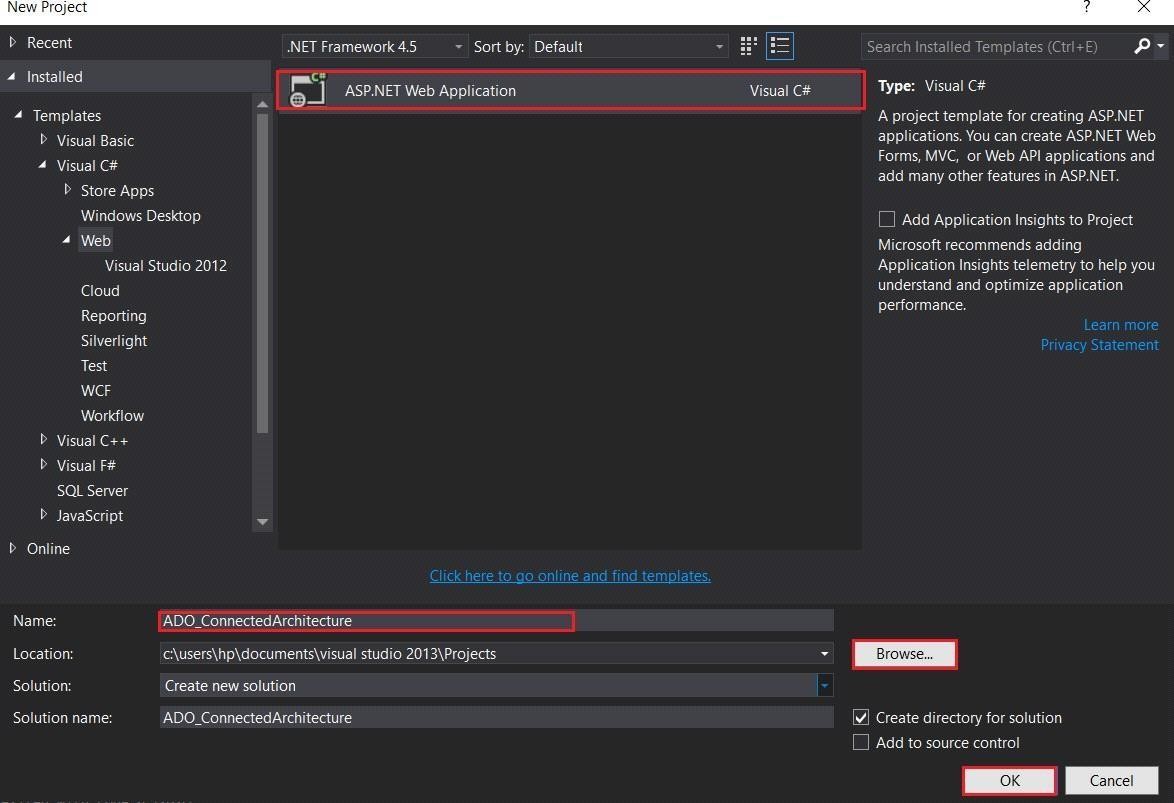


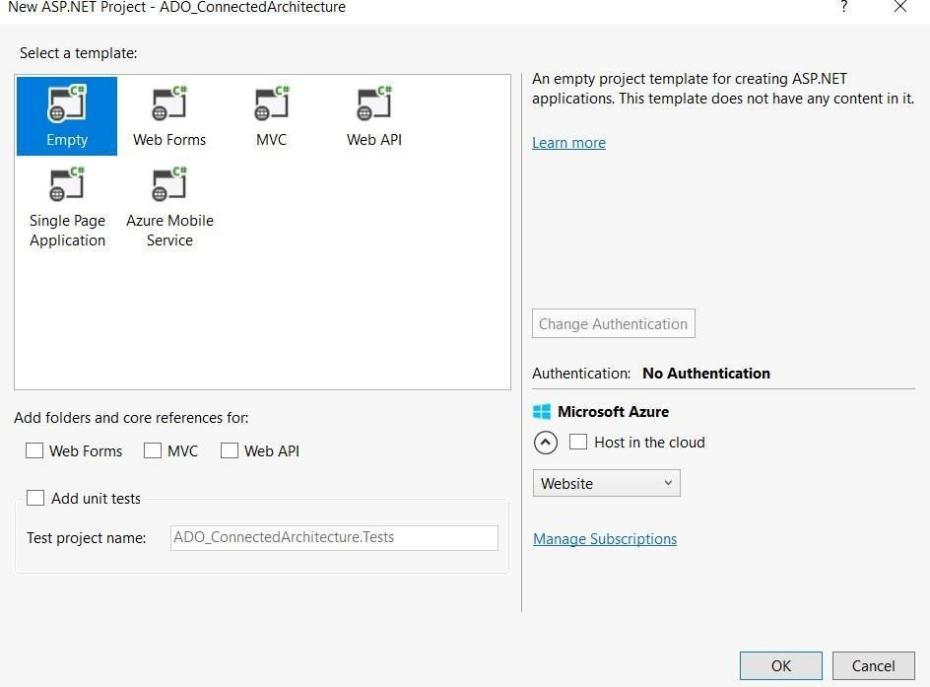
1. **Design a webpage to insert update, delete, search details using Disconnect Architecture.**

**Steps-**

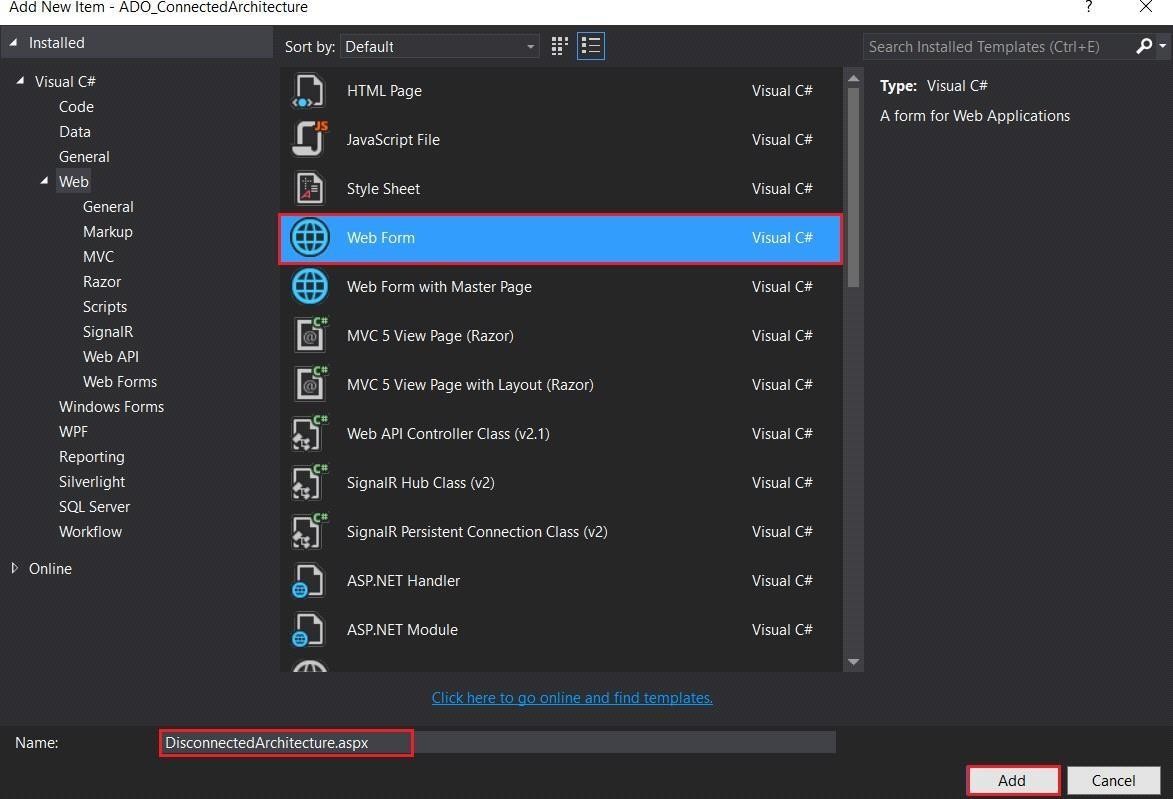
File new projectweb Application ok give project name (ADO\_ConnectedArchitecture.aspx)







Right click on projectAdd new item web form give name then save (DisconnectedArchitecture.aspx)



## INPUT-

**Filename**- **DisconnectedArchitecture.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

using System.Data.SqlClient;

namespace ADO\_ConnectedArchitecture

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\hp\Documents\Student.mdf;Int egrated Security=True;Connect Timeout=30;User Instance=False");

SqlCommand sc = new SqlCommand("insert into EMP (ID,Name,Salary) values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "')", con);

SqlDataAdapter sda = new SqlDataAdapter(sc); DataSet ds = new DataSet();

sda.Fill(ds, "emp");

sda.Update(ds, "emp"); GridView1.DataSource = ds;

Label5.Text = " Data Inserted Sucessfully";

}

catch (Exception ex)

{

Label5.Text = "ERROR :: " + ex.Message;

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\hp\Documents\Student.mdf;Int egrated Security=True;Connect Timeout=30;User Instance=False");

SqlCommand sc = new SqlCommand("update EMP set Name='" + TextBox2.Text + "', Salary='" + TextBox3.Text + "' where ID='" + TextBox1.Text + "'", con);

SqlDataAdapter sda = new SqlDataAdapter(sc); DataSet ds = new DataSet();

sda.Fill(ds, "emp");

sda.Update(ds, "emp"); GridView1.DataSource = ds;

Label5.Text = " Data Updated Sucessfully";

}

catch (Exception ex)

{

Label5.Text = "ERROR :: " + ex.Message;

}

}

protected void Button3\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\hp\Documents\Student.mdf;Int egrated Security=True;Connect Timeout=30;User Instance=False");

SqlCommand sc = new SqlCommand("delete EMP where ID='" + TextBox1.Text + "'", con);

SqlDataAdapter sda = new SqlDataAdapter(sc);

DataSet ds = new DataSet(); sda.Fill(ds, "emp"); GridView1.DataSource = ds; sda.Update(ds, "emp");

Label5.Text = " Data Deleted Sucessfully";

}

catch (Exception ex)

{

Label5.Text = "ERROR :: " + ex.Message;

}

}

protected void Button4\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\hp\Documents\Student.mdf;Int egrated Security=True;Connect Timeout=30;User Instance=False");

SqlCommand sc = new SqlCommand("select \*from EMP ", con); SqlDataAdapter sda = new SqlDataAdapter(sc);

DataSet ds = new DataSet(); sda.Fill(ds, "emp"); GridView1.DataSource = ds; GridView1.DataBind();

}

catch (Exception ex)

{

Label5.Text = "ERROR :: " + ex.Message;

}

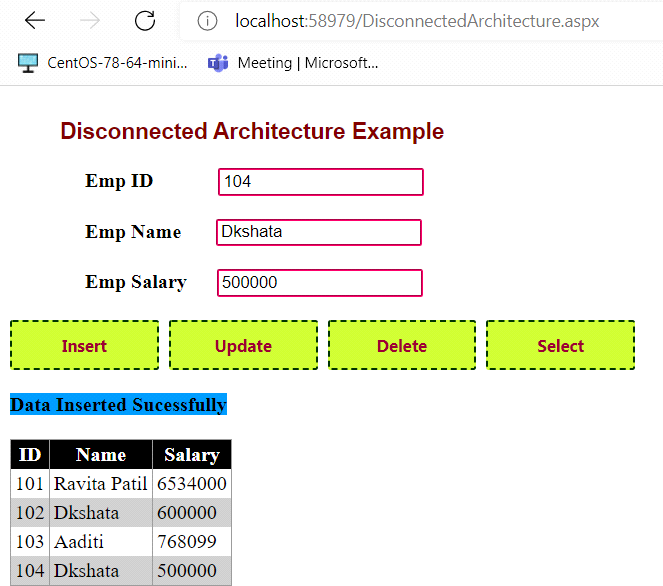
}

}

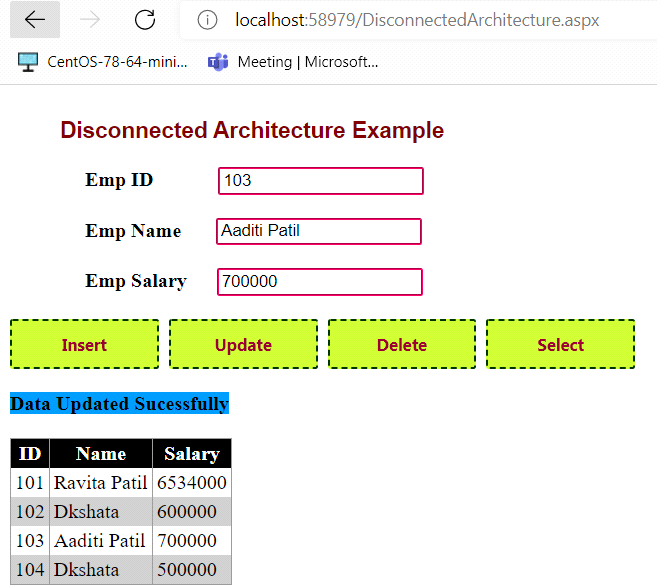
}

## OUTPUT-

**After Insert data then click on select button.**



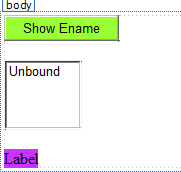
**After Click on update button.**



**Click on Delete Button data is deleted.**



1. **Design a webpage to demonstrate the working of a simple stored procedure. Input:**



Step 1 create stored procedure

CREATE PROCEDURE [dbo].[SPEname]

AS

SELECT Ename from Emptb

**.CS File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

using System.Data.SqlClient;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click1(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30");

SqlCommand command = new SqlCommand(); SqlDataAdapter adapter = new SqlDataAdapter(); DataSet ds = new DataSet();

try

{

con.Open(); command.Connection = con;

command.CommandType = CommandType.StoredProcedure; command.CommandText = "SPEname";

adapter = new SqlDataAdapter(command); adapter.Fill(ds);

con.Close();

ListBox1.DataSource = ds.Tables[0]; ListBox1.DataTextField = "Ename"; ListBox1.DataBind();

}

catch (Exception ex)

{

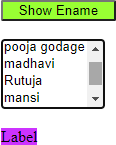
Label1.Text = "Error in execution " + ex.ToString();

}

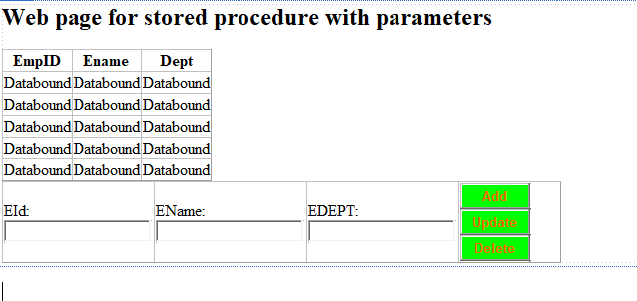
}

}

**Output:**



1. **Design a webpage to demonstrate the working of parameterized stored Procedure**



Create SP

CREATE PROCEDURE [dbo].[MyProcedure] @Action VARCHAR(10),

@Eid VARCHAR(20) = NULL, @Ename VARCHAR(50) = NULL, @Edept VARCHAR(50) = NULL

AS

BEGIN

SET NOCOUNT ON;

--SELECT

IF @Action = 'SELECT' BEGIN

SELECT Eid, Ename, Edept FROM Emptb

END

--INSERT

IF @Action = 'INSERT' BEGIN

INSERT INTO Emptb(Eid,Ename, Edept) VALUES (@Eid,@Ename, @Edept)

END

--UPDATE

IF @Action = 'UPDATE' BEGIN

UPDATE Emptb

SET Ename = @Ename, Edept = @Edept WHERE Eid = @Eid

END

--DELETE

IF @Action = 'DELETE' BEGIN

DELETE FROM Emptb

WHERE Eid = @Eid END

END

**.CS File:**

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.Data;

public partial class para\_sp : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

this.BindGrid();

}

private void BindGrid()

{

using (SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30"))

{

using (SqlCommand cmd = new SqlCommand("MyProcedure"))

{

cmd.Parameters.AddWithValue("@Action", "SELECT"); using (SqlDataAdapter sda = new SqlDataAdapter())

{

cmd.CommandType = CommandType.StoredProcedure; cmd.Connection = con;

sda.SelectCommand = cmd;

using (DataTable dt = new DataTable())

{

sda.Fill(dt); GridView1.DataSource = dt; GridView1.DataBind();

}

}

}

}

}

protected void btnAdd\_Click(object sender, EventArgs e)

{

string name = txtName.Text; string Dept = txtdept.Text;

using (SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30"))

{

using (SqlCommand cmd = new SqlCommand("MyProcedure"))

{

cmd.CommandType = CommandType.StoredProcedure; cmd.Parameters.AddWithValue("@Action", "INSERT"); cmd.Parameters.AddWithValue("@Eld", txtid.Text); cmd.Parameters.AddWithValue("@Ename", name); cmd.Parameters.AddWithValue("@Edept", Dept); cmd.Connection = con;

con.Open(); cmd.ExecuteNonQuery(); con.Close();

}

}

this.BindGrid();

}

protected void Button2\_Click(object sender, EventArgs e)

{

string str = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30";

SqlConnection cn = new SqlConnection(str); SqlCommand cmd = new SqlCommand("MyProcedure", cn); cmd.CommandType = CommandType.StoredProcedure; cmd.Parameters.AddWithValue("@Action", "DELETE");

cmd.Parameters.AddWithValue("@Eld", txtid.Text); cn.Open();

cmd.ExecuteNonQuery(); cn.Close(); this.BindGrid();

}

protected void Button1\_Click(object sender, EventArgs e)

{

string str = @"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\Emp.m df;Integrated Security=True;Connect Timeout=30";

SqlConnection cn = new SqlConnection(str); SqlCommand cmd = new SqlCommand("MyProcedure", cn); cmd.CommandType = CommandType.StoredProcedure; cmd.Parameters.AddWithValue("@Action", "UPDATE"); cmd.Parameters.AddWithValue("@Eld", txtid.Text); cmd.Parameters.AddWithValue("@Ename", txtName.Text); cmd.Parameters.AddWithValue("@Edept", txtdept.Text);

cn.Open(); cmd.ExecuteNonQuery(); cn.Close(); this.BindGrid();

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

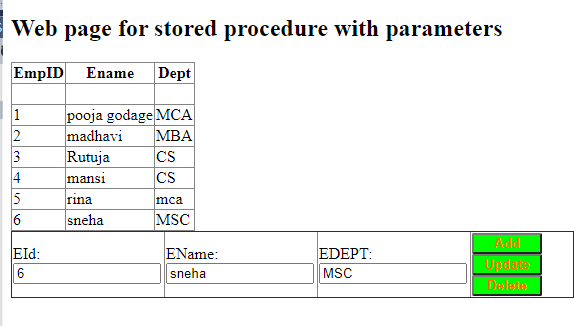
{

}

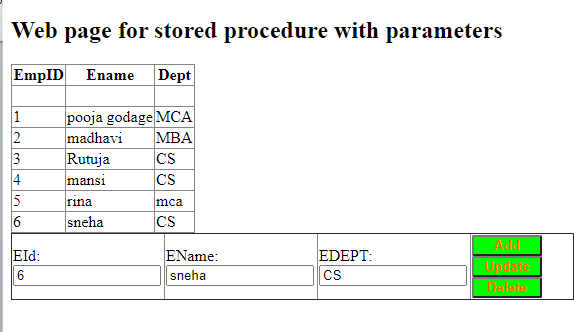
}

**Output:**

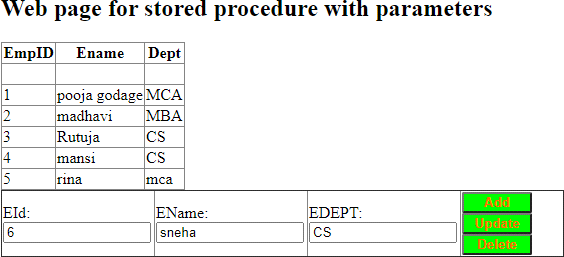
**Add Record**



**Update Record:**



**Delete Record:**



1. **Web Application to demonstrate LINQ with object data source.**

**Input:**



**Class File:**

using System;

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

using System.Web;

public class Dept

{

public int DepartmentId { get; set; }

public string Name { get; set; }

}

**.Cs File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

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

{

public List<Dept> Getdata()

{

return new List<Dept>

{

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| new | Dept | { | DepartmentId | = | 1, | Name | = | "Account" }, |
| new | Dept | { | DepartmentId | = | 2, | Name | = | "Sales" }, |
| new | Dept | { | DepartmentId | = | 3, | Name | = | "Marketing" }, |
| new | Dept | { | DepartmentId | = | 4, | Name | = | "Account" }, |
| new | Dept | { | DepartmentId | = | 5, | Name | = | "Testing" } |

};

}

protected void Page\_Load(object sender, EventArgs e)

{

var departmentList = Getdata();

var q = from d in departmentList // replaced Dept with DepartmentLIst

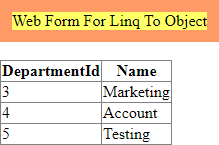
where d.DepartmentId >= 3 select d;

this.GridView1.DataSource = q; this.GridView1.DataBind();

}

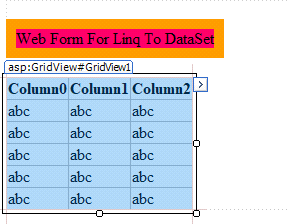
}

**Output:**



1. **Web Application to demonstrate LINQ with data set.**

**Input:**



**.CS File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

using System.Data.SqlClient;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

BindGridview();

}

}

protected void BindGridview()

{

DataSet ds = new DataSet();

using (SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\inser t diss.mdf;Integrated Security=True;Connect Timeout=30"))

{

con.Open();

SqlCommand cmd = new SqlCommand("select \* from EMP", con); cmd.CommandType = CommandType.Text;

SqlDataAdapter da = new SqlDataAdapter(cmd); da.Fill(ds);

con.Close();

if (ds.Tables[0].Rows.Count > 0)

{

var result = from dt in ds.Tables[0].AsEnumerable() where

(dt.Field<string>("Name").EndsWith("i"))

select new

{

Name = dt.Field<string>("Name"), Salary = dt.Field<int>("Salary")

};

GridView1.DataSource = result; GridView1.DataBind();

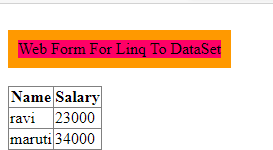
}

}

}

}

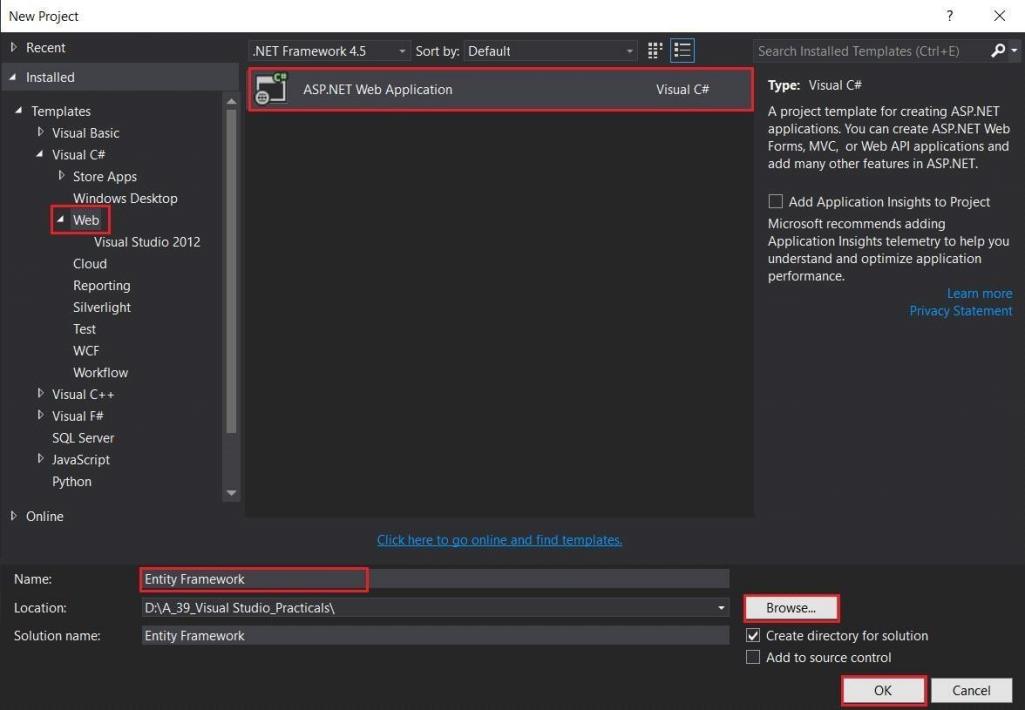
**Output:**



1. **Create a webpage that demonstrates the use of data bound controls**
2. **Build websites to demonstrate the working of entity framework**

**Steps-**

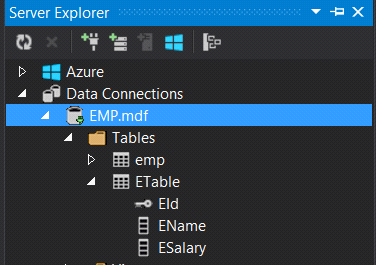
File new project web Application(Entity Framework)



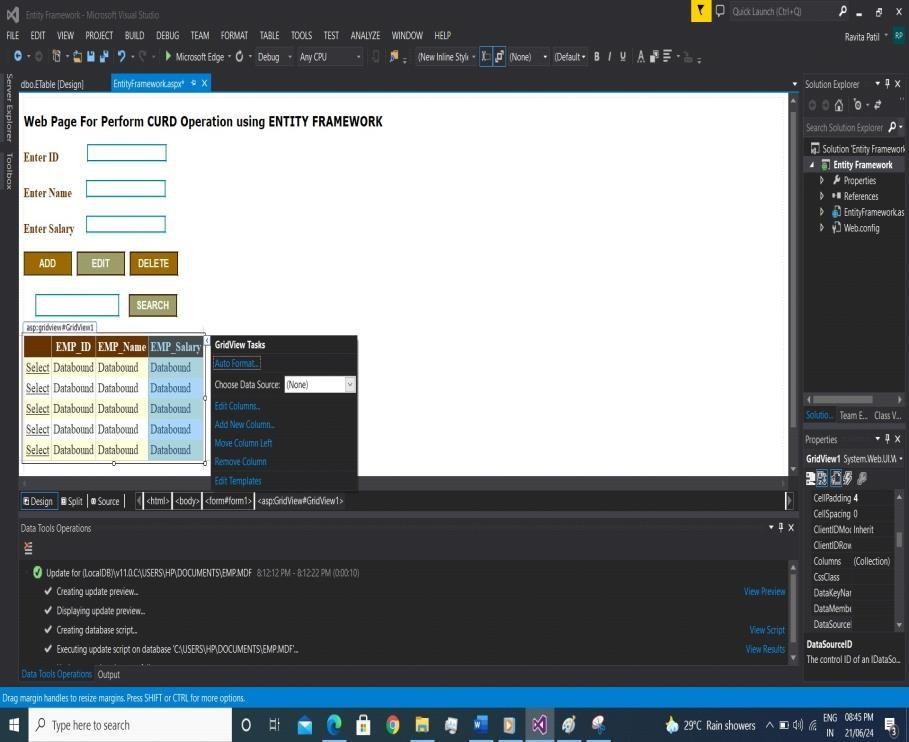
Right click on project addnew itemweb form(EntityFramework.aspx)

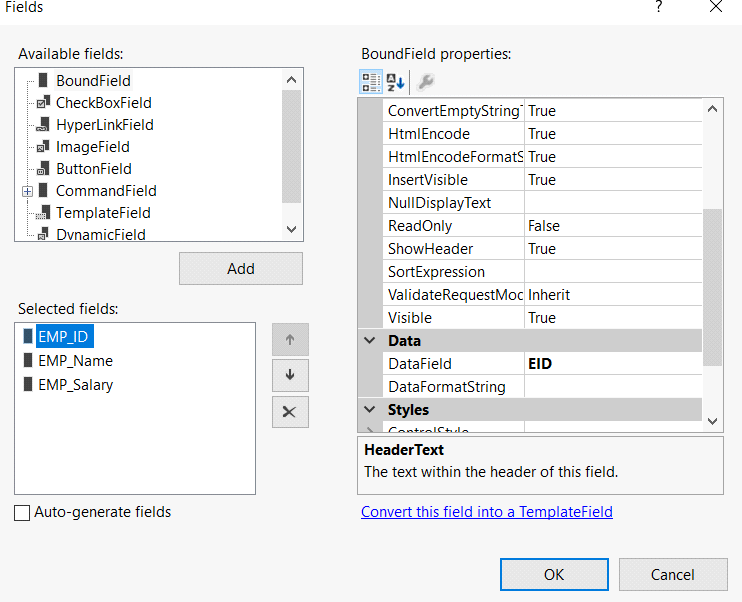


**Create connection EMP.mdf Add table(ETable)**



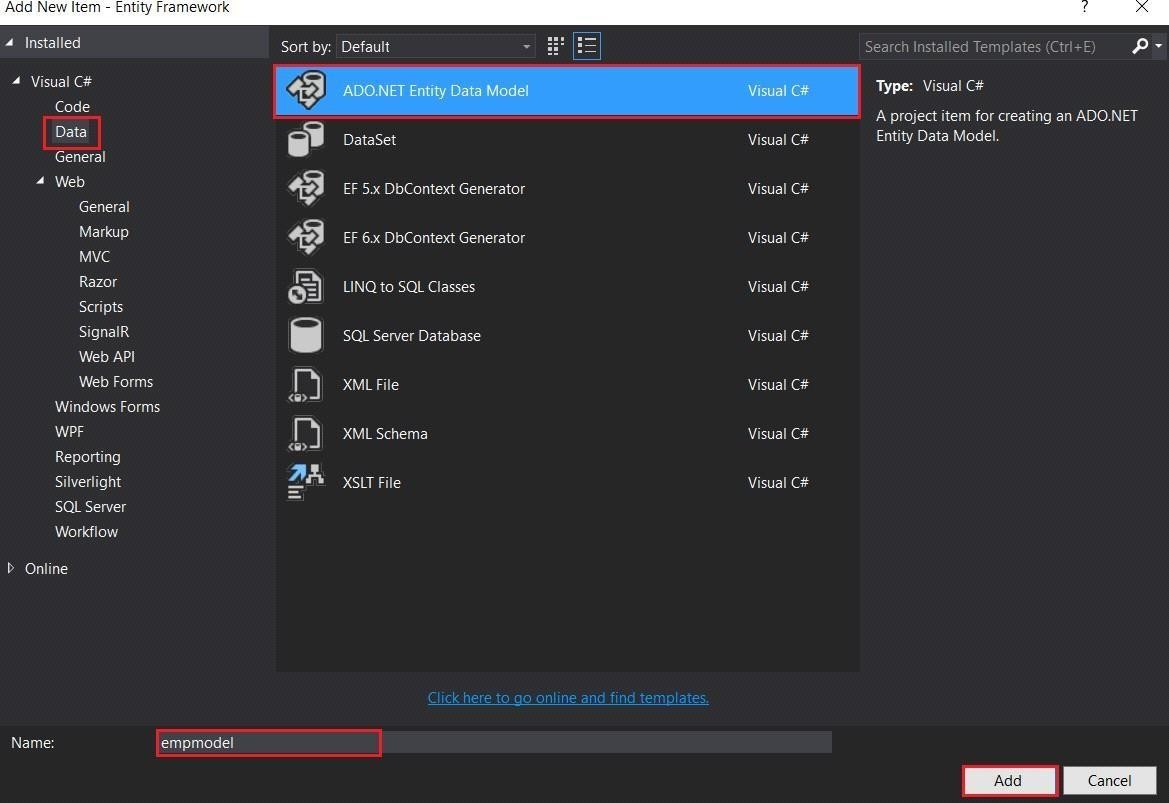
**Gridviewedit column BoundField add**

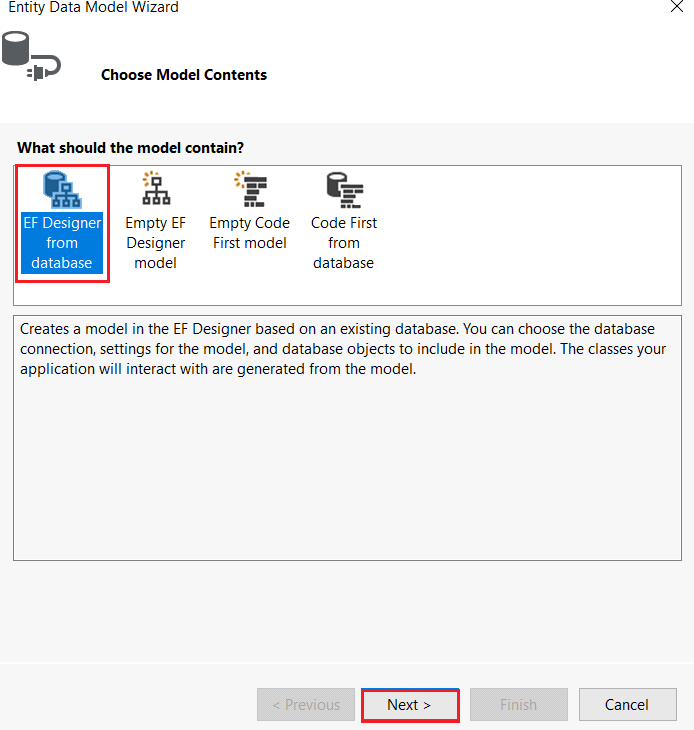


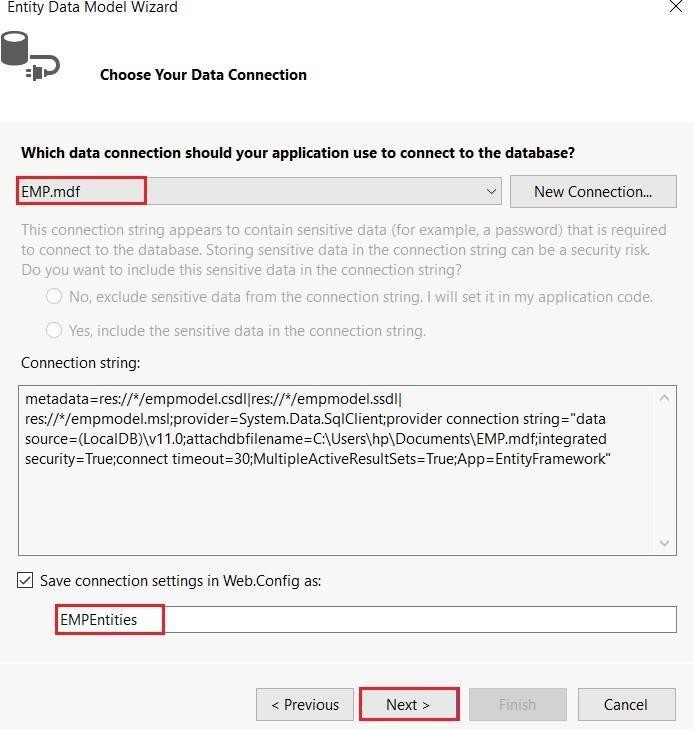


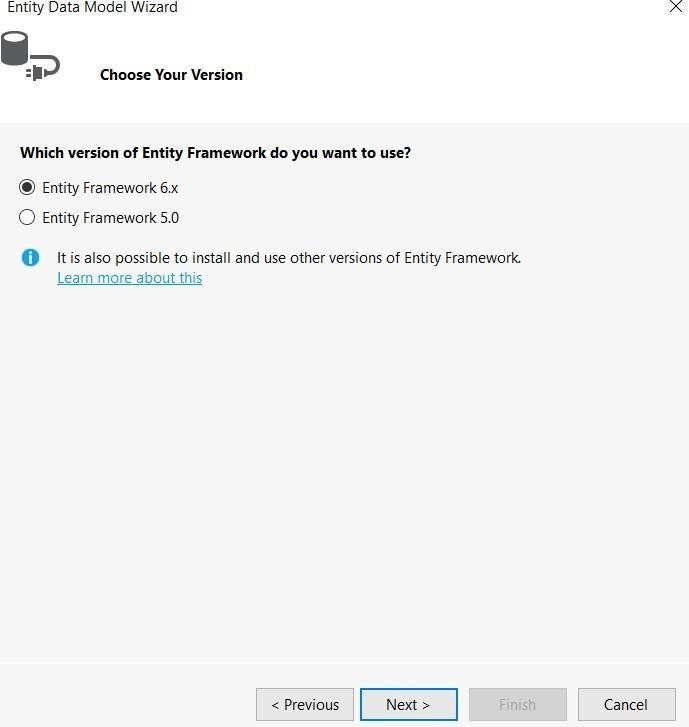
In design part select GridView control property set Auto generate select Button (True) and Auto generate Column (False).

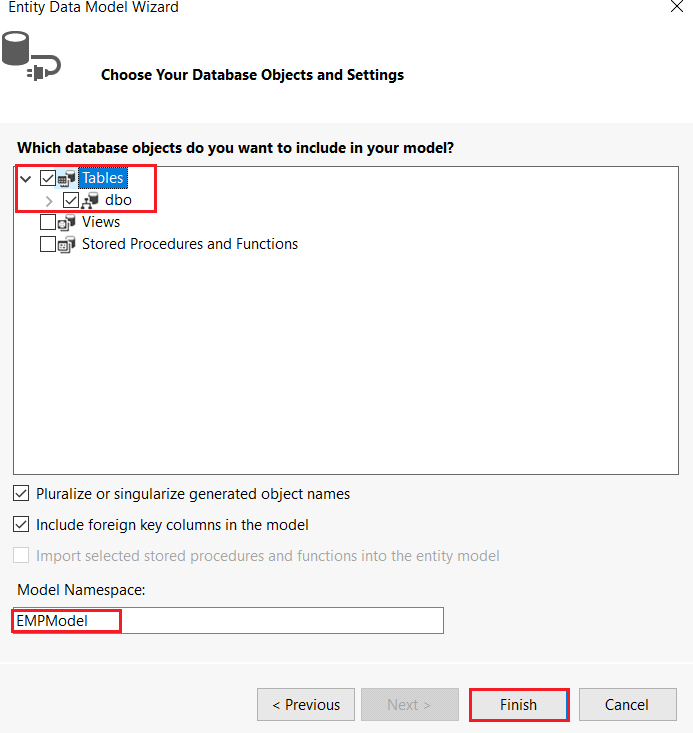
Right click on project add new item data select ADO.NET Entity data model (empmodel)

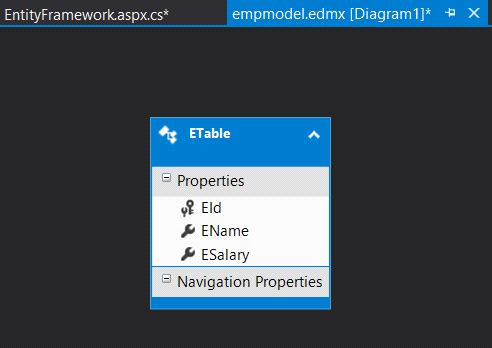












## INPUT-

**Filename-EntityFramework.aspx.cs**

using System;

using System.Collections.Generic;

using System.Linq; using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

namespace Entity\_Framework

{

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

{

////Create an object of Data Model and emp name of table class EMPEntities db = new EMPEntities();

ETable e1 = new ETable();

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

BindGridView();

}

}

public void BindGridView()

{

var result = (from E in db.ETables

select new {E.EId , E.EName , E.ESalary}).ToList(); GridView1.DataSource = result;

GridView1.DataBind();

}

protected void Button1\_Click(object sender, EventArgs e)

{

//insert

e1.EId = int.Parse(TextBox1.Text); e1.EName = TextBox2.Text; e1.ESalary = int.Parse(TextBox3.Text); db.ETables.Add(e1); db.SaveChanges();

BindGridView();

}

protected void Button2\_Click(object sender, EventArgs e)

{

//Edit

e1.EId = int.Parse(TextBox1.Text);

var result = (from E in db.ETables where E.EId == e1.EId select E).Single(); result.EName = TextBox2.Text;

result.ESalary = int.Parse(TextBox3.Text); db.SaveChanges();

BindGridView();

}

protected void Button3\_Click(object sender, EventArgs e)

{

e1.EId = int.Parse(TextBox1.Text);

var result = (from E in db.ETables where E.EId == e1.EId select E).Single(); db.ETables.Remove(result);

db.SaveChanges(); BindGridView();

}

protected void Button4\_Click(object sender, EventArgs e)

{

e1.EId = int.Parse(TextBox4.Text);

var result = (from E in db.ETables where E.EId == e1.EId select new { E.EId,

E.EName, E.ESalary }).ToList(); GridView1.DataSource = result; GridView1.DataBind();

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

TextBox1.Text = GridView1.SelectedRow.Cells[1].Text; TextBox2.Text = GridView1.SelectedRow.Cells[2].Text; TextBox3.Text = GridView1.SelectedRow.Cells[3].Text;

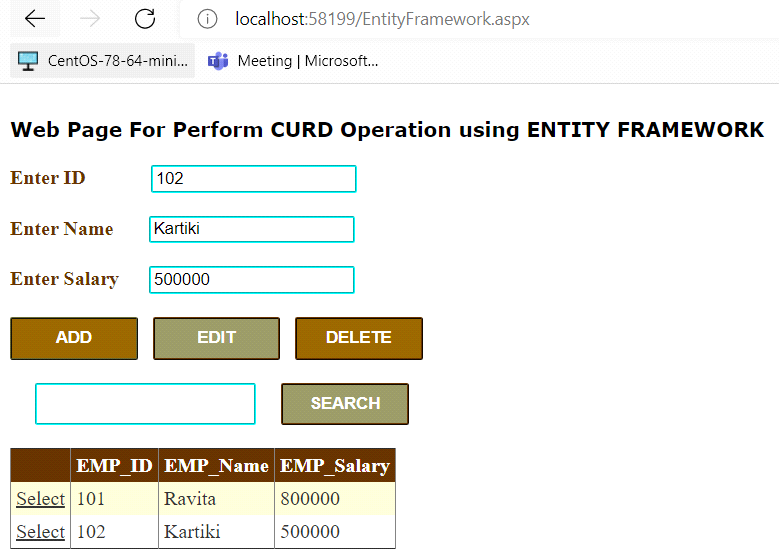
}

}

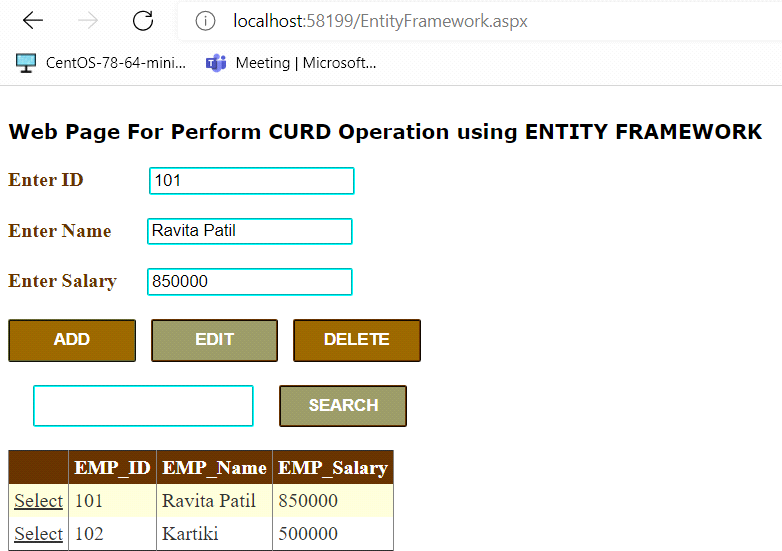
}

## OUTPUT-

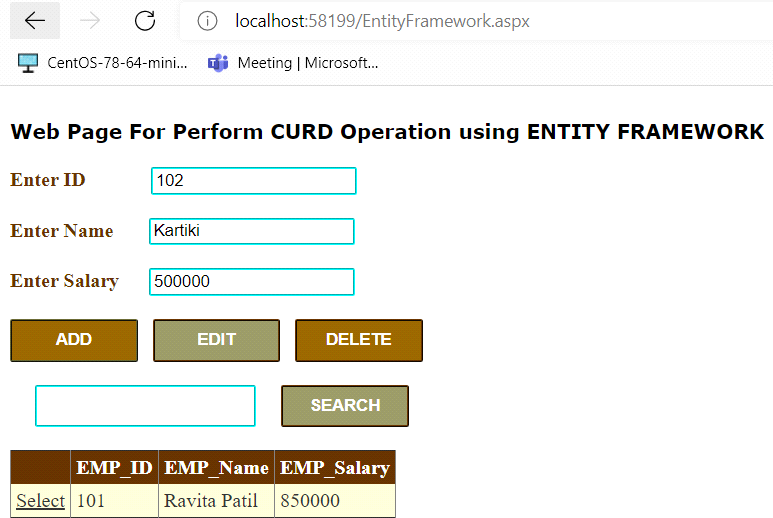
**Click on add button-**



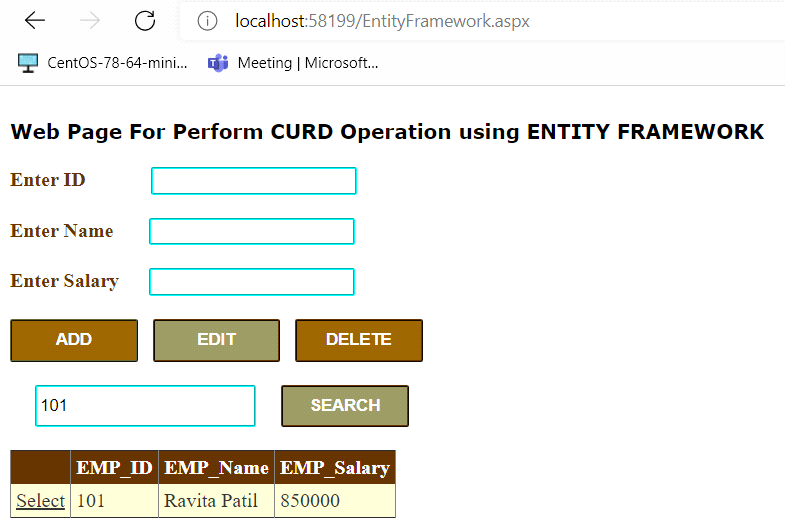
**Click on Edit Button-**



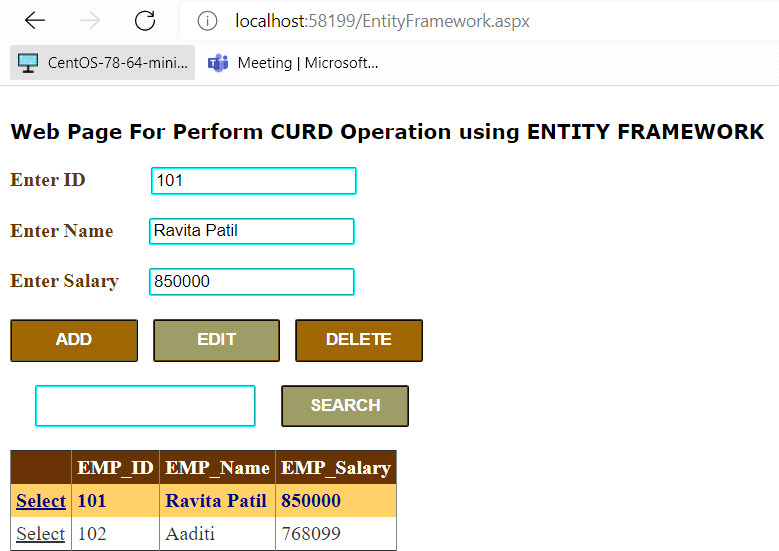
**Click on Delete Button-**



**Click on Search Button-**



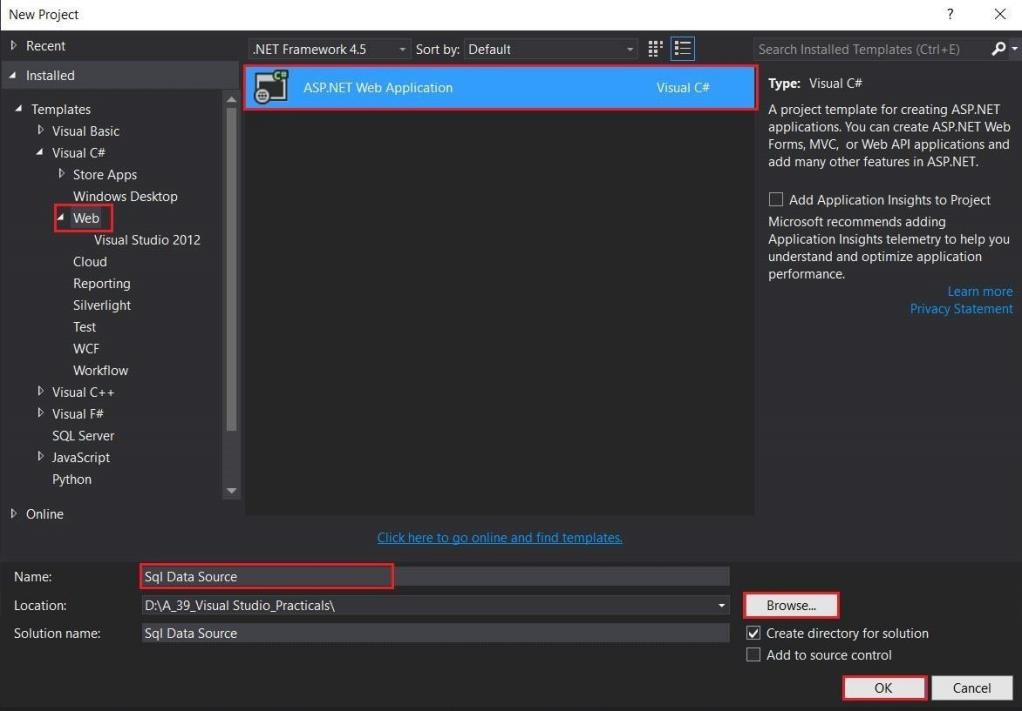
We click on select Row is selected



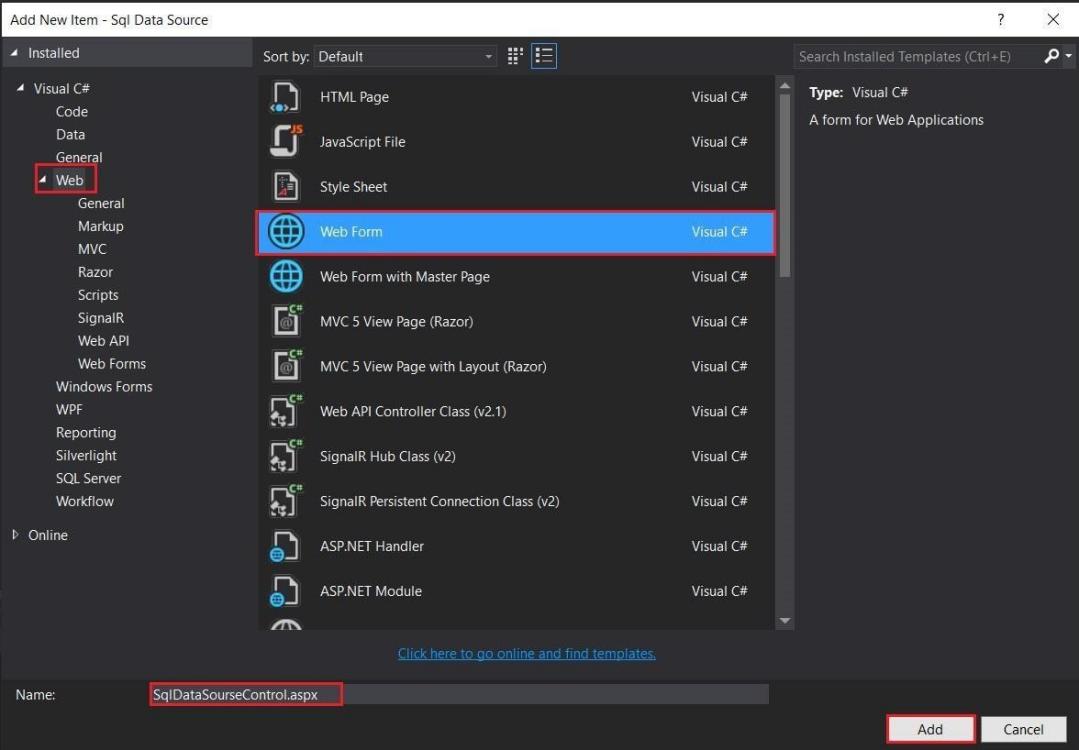
1. **Design a web page to perform CRUD operation using SqlDataSource**

**Steps-**

File new project web Application(c#) Sql Data Source ok

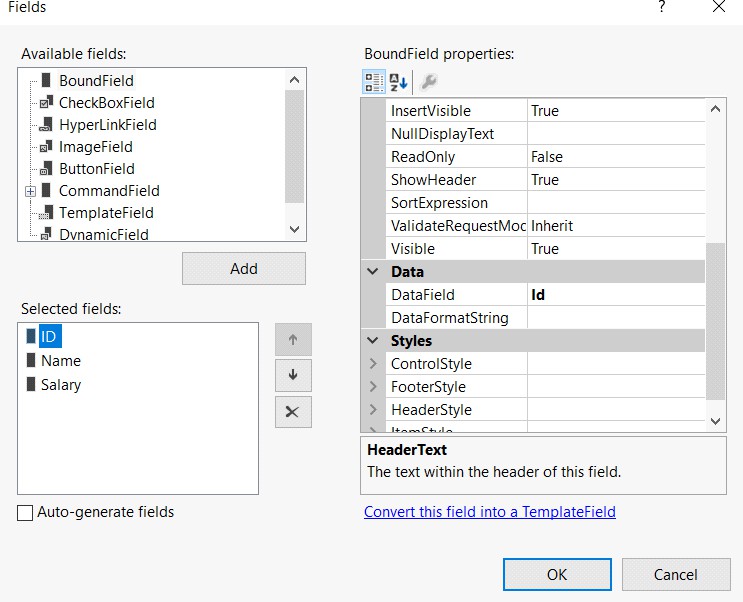


Right click on project add new Web Form SqlDataSource.aspxAdd.

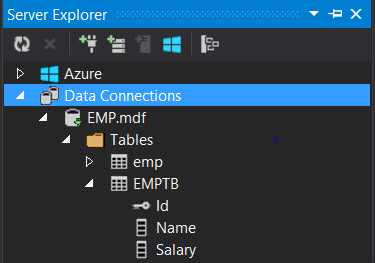


Go to grid view property set Auto Generate Columns- False , Auto Generate Delete Button- True, Auto Generate Select Button-True, Auto Generate Edit Button-Trueok

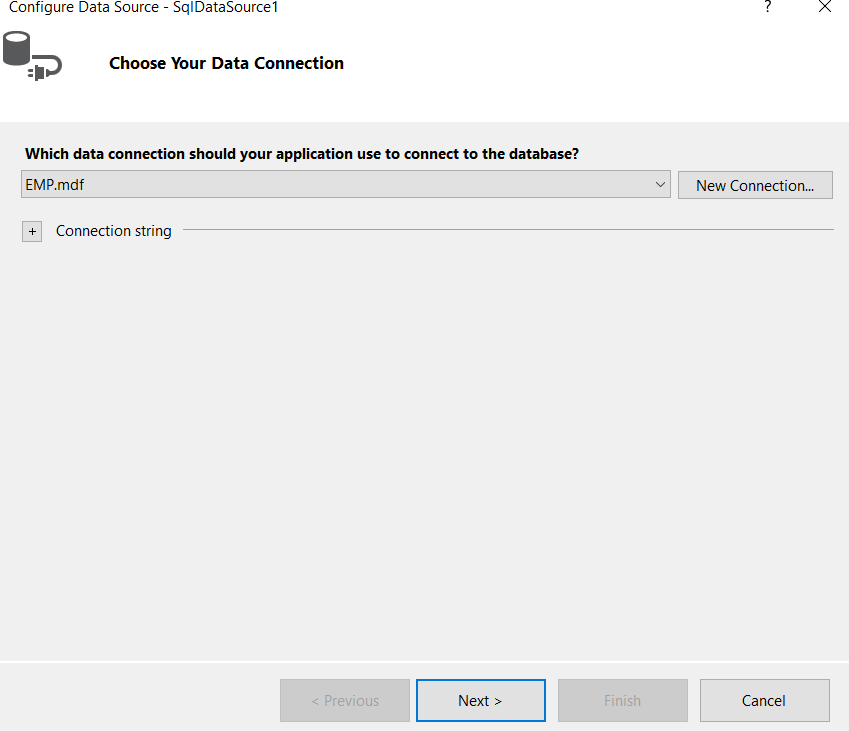
Then go to grid view Edit Column add bound field set data field ok

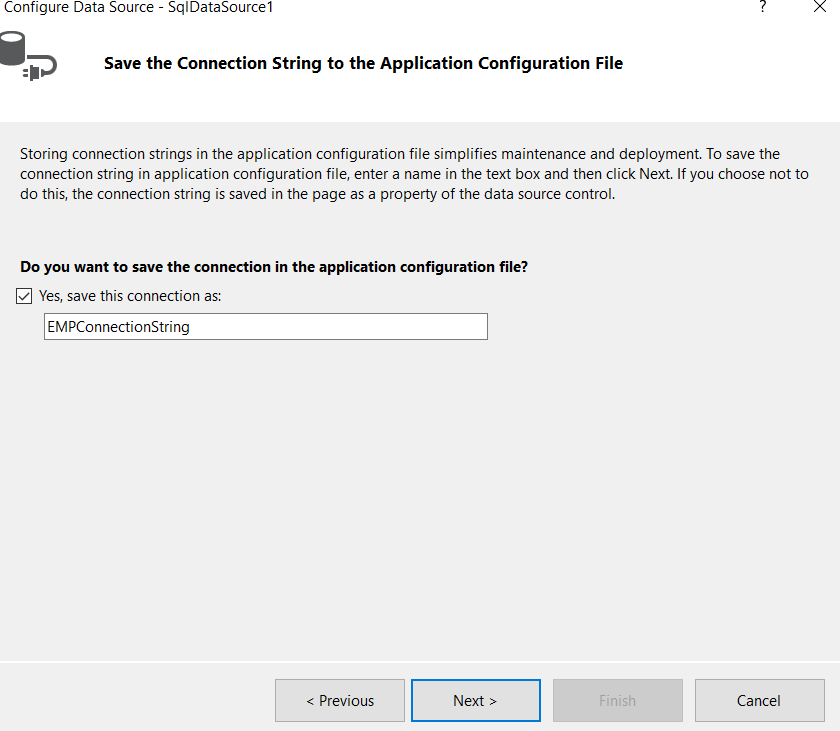


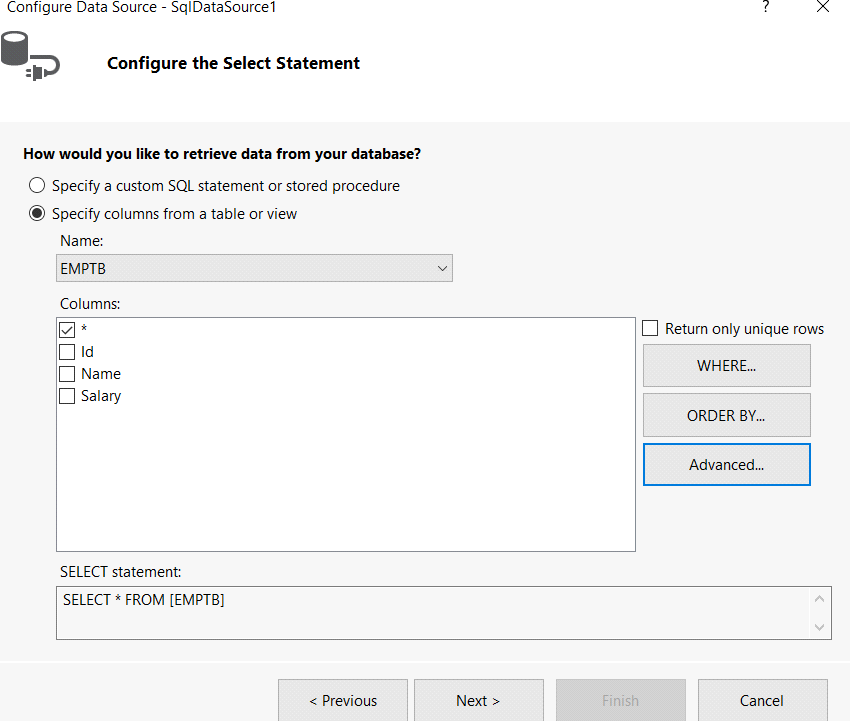
Create Connections EMP.mdf Table name(EMPTB)

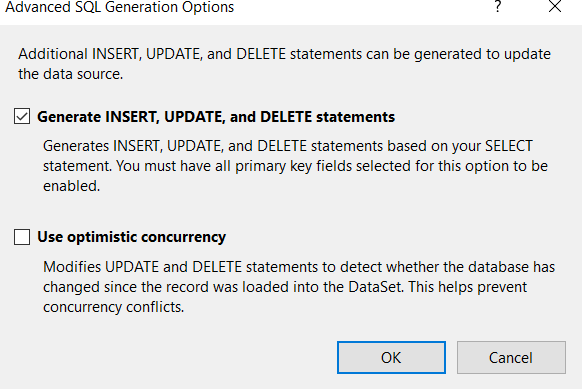


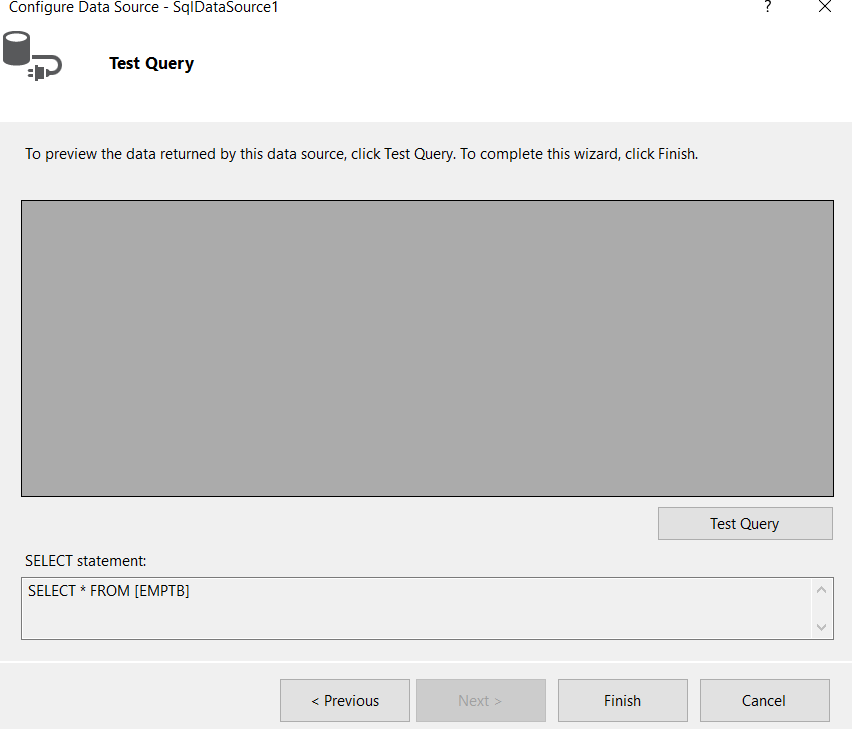
Go to SqlDataSource configure data source







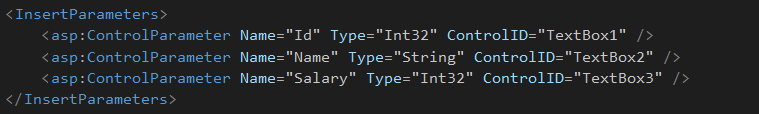




Click on Test Query then Finish

## INPUT-

**Filename-SqlDataSource.aspx**



**Filename- SqlDataSource.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace Sql\_Data\_Source

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

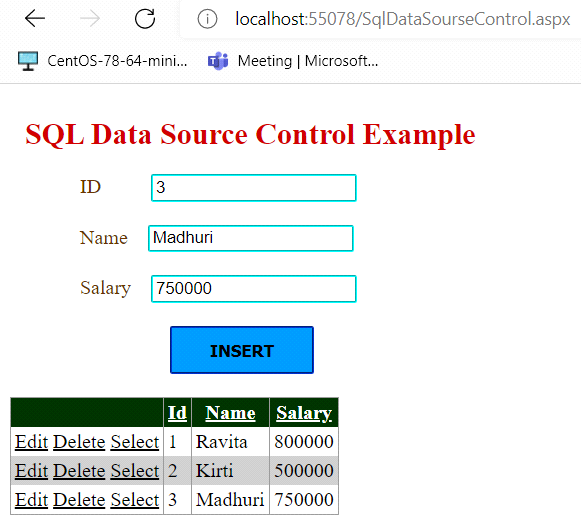
SqlDataSource1.Insert();

}

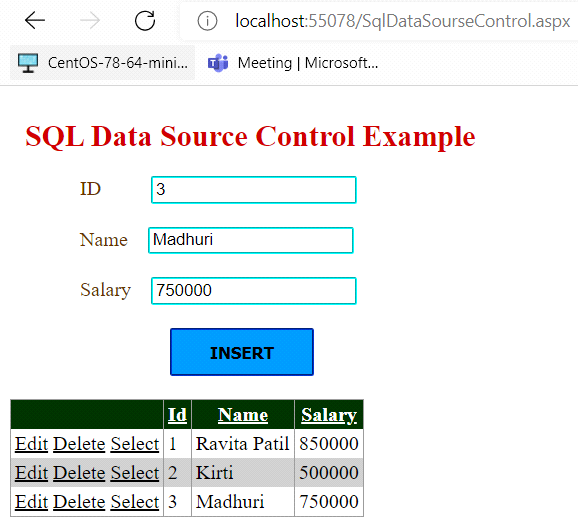
}

}

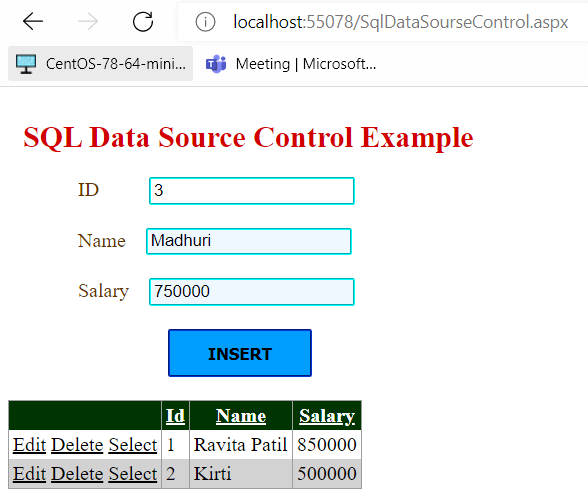
## OUTPUT-



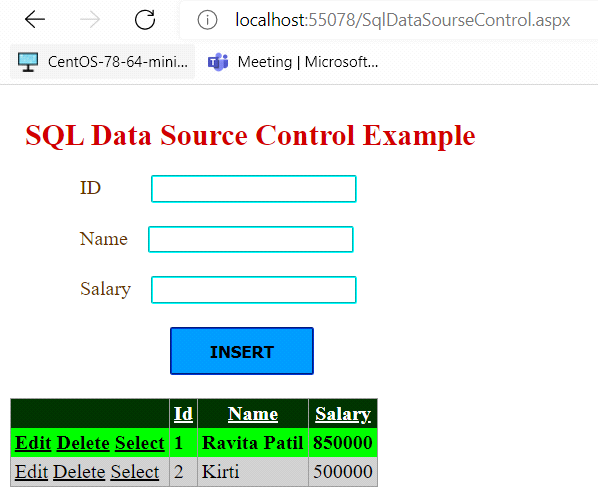
Click on Edit-



Click on Delete-



Click on Select –

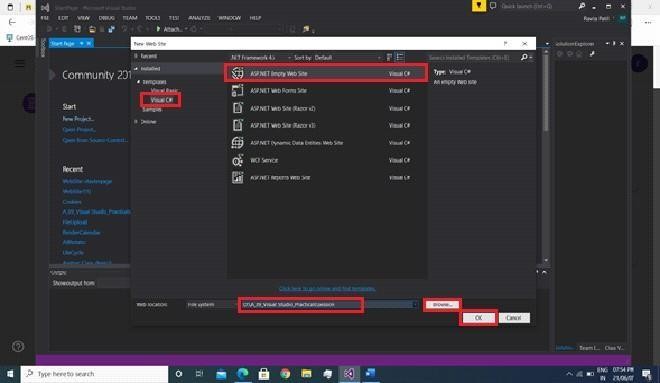


1. **Module: Session Management and AJAX**
   1. **Display the number of times current page is visited using View State, Session, Cookies and Application State.**

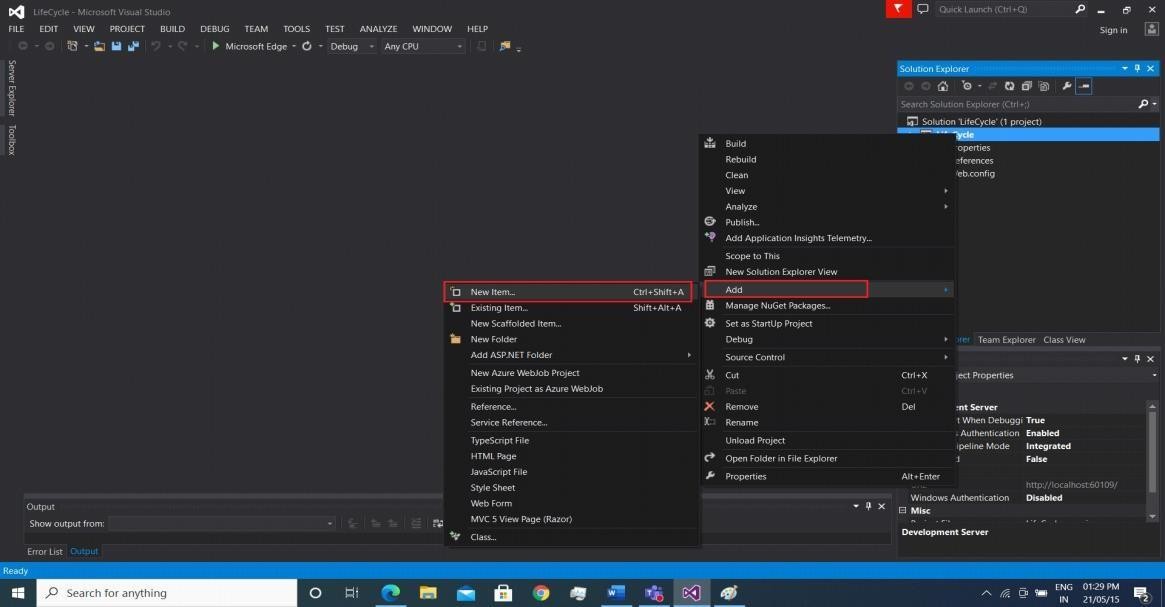
**Steps-**

First open your visual studio-->File-->New-->Website-->Select ASP.NET Empty Website-- >Ok





open solution explorer-->Add New Web Form-->Drag and Drop from Toolbox



## INPUT-

**Filename-Session.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack == true)

{

Label1.Text = "0";

Label2.Text = "0";

Label3.Text = "0";

Label4.Text = "0";

}

}

protected void ButCookies\_Click(object sender, EventArgs e)

{

int cook;

if (Request.Cookies["cook"] == null)

{

cook = 0;

}

else

{

cook = int.Parse(Request.Cookies["cook"].Value);

}

cook++;

Label1.Text = cook.ToString();

}

protected void ButSession\_Click(object sender, EventArgs e)

{

if (Session["count"] != null)

{

Session["count"] = Convert.ToInt32(Session["count"]) + 1;

}

else

{

Session["count"] = Convert.ToInt32(Session["count"] = 1);

}

Label2.Text = Session["count"].ToString();

}

protected void Butviewstate\_Click(object sender, EventArgs e)

{

if (ViewState["num"] != null)//When the page is again posted back, the \_VIEWSTATE field is sent to the server with the HTTP request.

{

ViewState["num"] = Convert.ToInt32(ViewState["num"]) + 1;

}

else

{

ViewState["num"] = Convert.ToInt32(ViewState["num"] = 1);

}

Label3.Text = ViewState["num"].ToString();

}

protected void Butapplication\_Click(object sender, EventArgs e)

{

if (Application["num"] != null)

{

Application["num"] = Convert.ToInt32(Application["num"]) + 1;

}

else

{

Application["num"] = Convert.ToInt32(Application["num"] = 1);

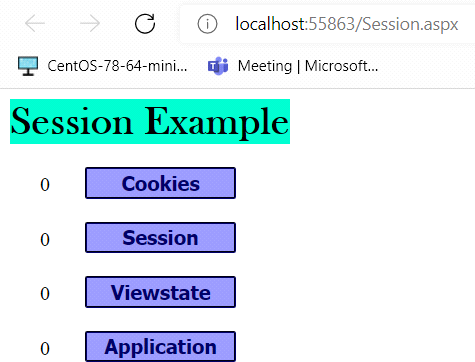
}

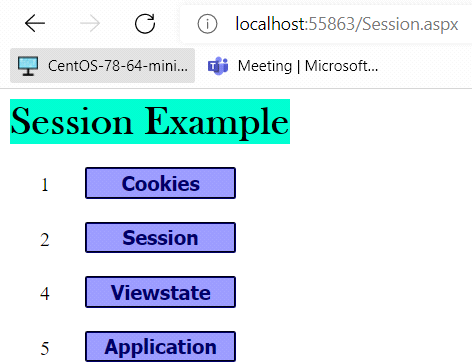
Label4.Text = Application["num"].ToString();

}

}

## OUTPUT-

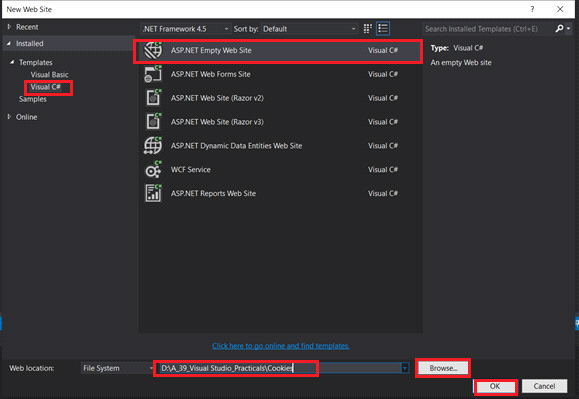




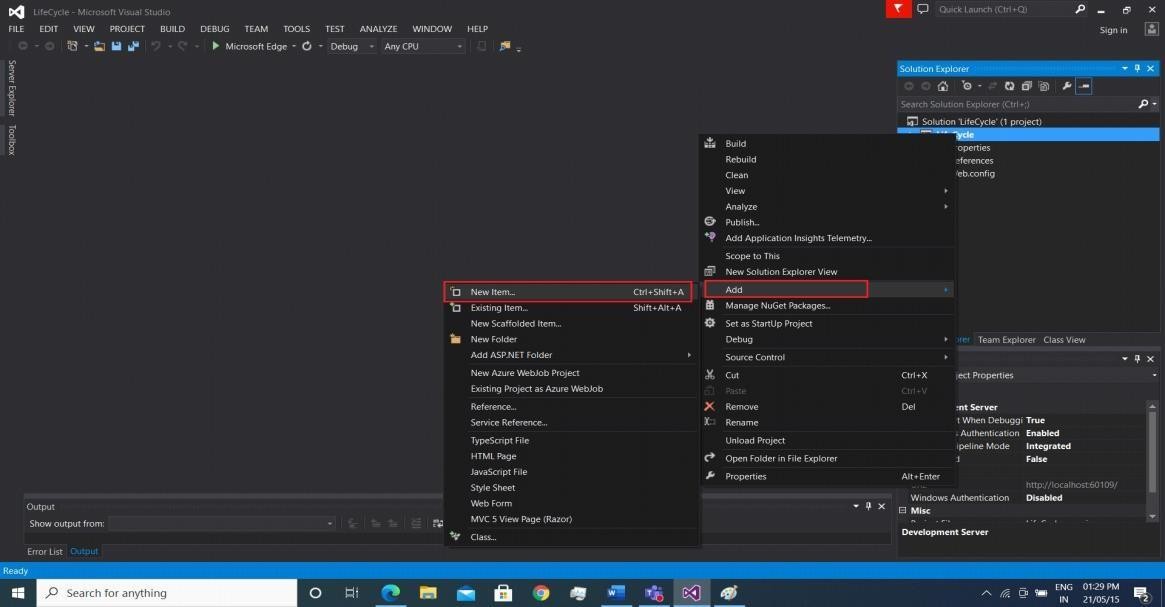
* 1. **Design a webpage to read, write and remove cookies. Steps-**

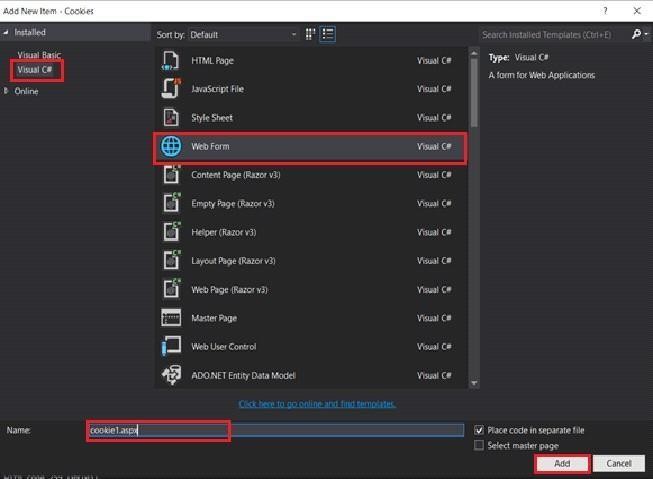
First open your visual studio-->File-->New-->Website-->Select ASP.NET Empty Website-- >Ok





open solution explorer-->Add New Web Form-->Drag and Drop from Toolbox





## INPUT-

**Filename-cookie1.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

public partial class cookie1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

HttpCookie cook1 = new HttpCookie("username"); cook1.Values["username"] = txtcookie.Text; cook1.Expires = DateTime.Now.AddSeconds(10); Response.Cookies.Add(cook1);

Label3.Text = "Cookie " + txtcookie.Text + " is Created";

}

protected void Button2\_Click(object sender, EventArgs e)

{

if (Request.Cookies["username"] != null)

{

Label3.Text = "Cookie Read : " + Request.Cookies["username"].Value;

}

else

{

Label3.Text = "Cookie Empty";

}

}

protected void Button3\_Click(object sender, EventArgs e)

{

Response.Cookies["username"].Expires = DateTime.Now.AddHours(-1); Label3.Text = "Cookie Deleted";

}

}

**OUTPUT-**





**Click on button Create**

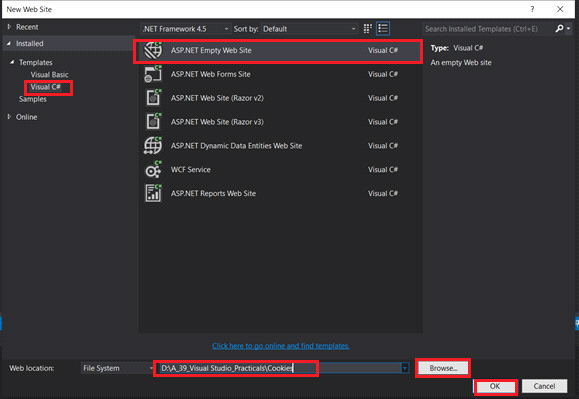
**Click on button Read**

**click on button Delete Cookie is deleted. After delete Cookie then click on Read button**

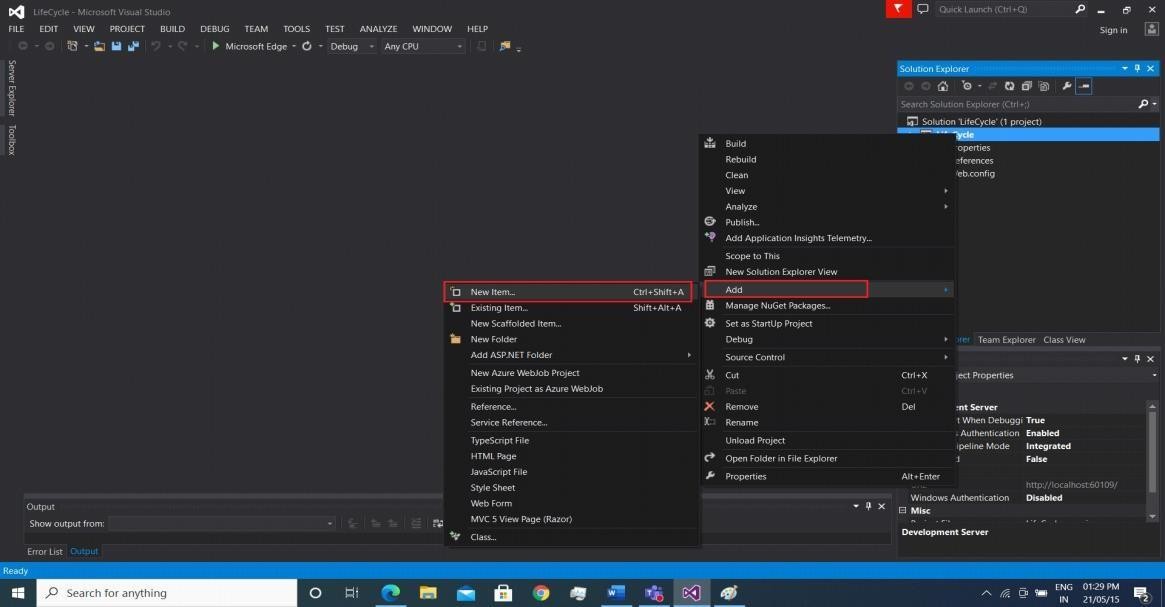
**B) Multiple Cookies Steps-**

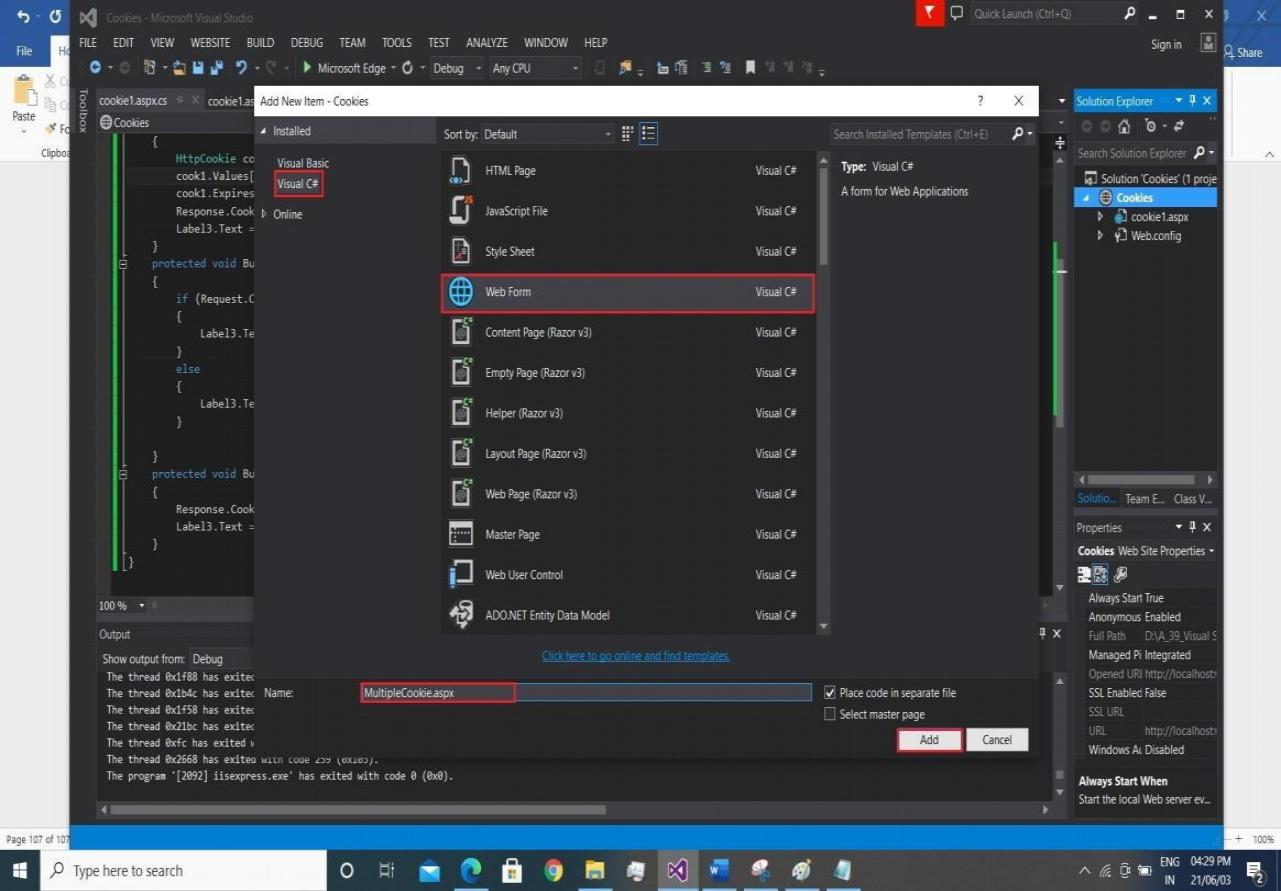
First open your visual studio-->File-->New-->Website-->Select ASP.NET Empty Website-- >Ok





open solution explorer-->Add New Web Form-->Drag and Drop from Toolbox





## INPUT-

**Filename- MultipleCookies.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

if(Request.Cookies["preferences"] != null)

{

lblFirstName.Text = Request.Cookies["preferences"]["firstName"]; lblLastName.Text = Request.Cookies["preferences"]["lastName"]; lblFavoriteColor.Text = Request.Cookies["preferences"]["favoriteColor"];

}

}

protected void btnSubmit\_Click(object sender, EventArgs e)

{

Response.Cookies["preferences"]["firstName"] = txtFirstName.Text; lblFirstName.Text = "First Name is : " + txtFirstName.Text; Response.Cookies["preferences"]["lastName"] = txtLastName.Text; lblLastName.Text = "Last Name is : " + txtLastName.Text; Response.Cookies["preferences"]["favoriteColor"] = txtFavoriteColor.Text; lblFavoriteColor.Text = "fav Colour is : " + txtFavoriteColor.Text; Response.Cookies["preferences"].Expires = DateTime.MaxValue;

}

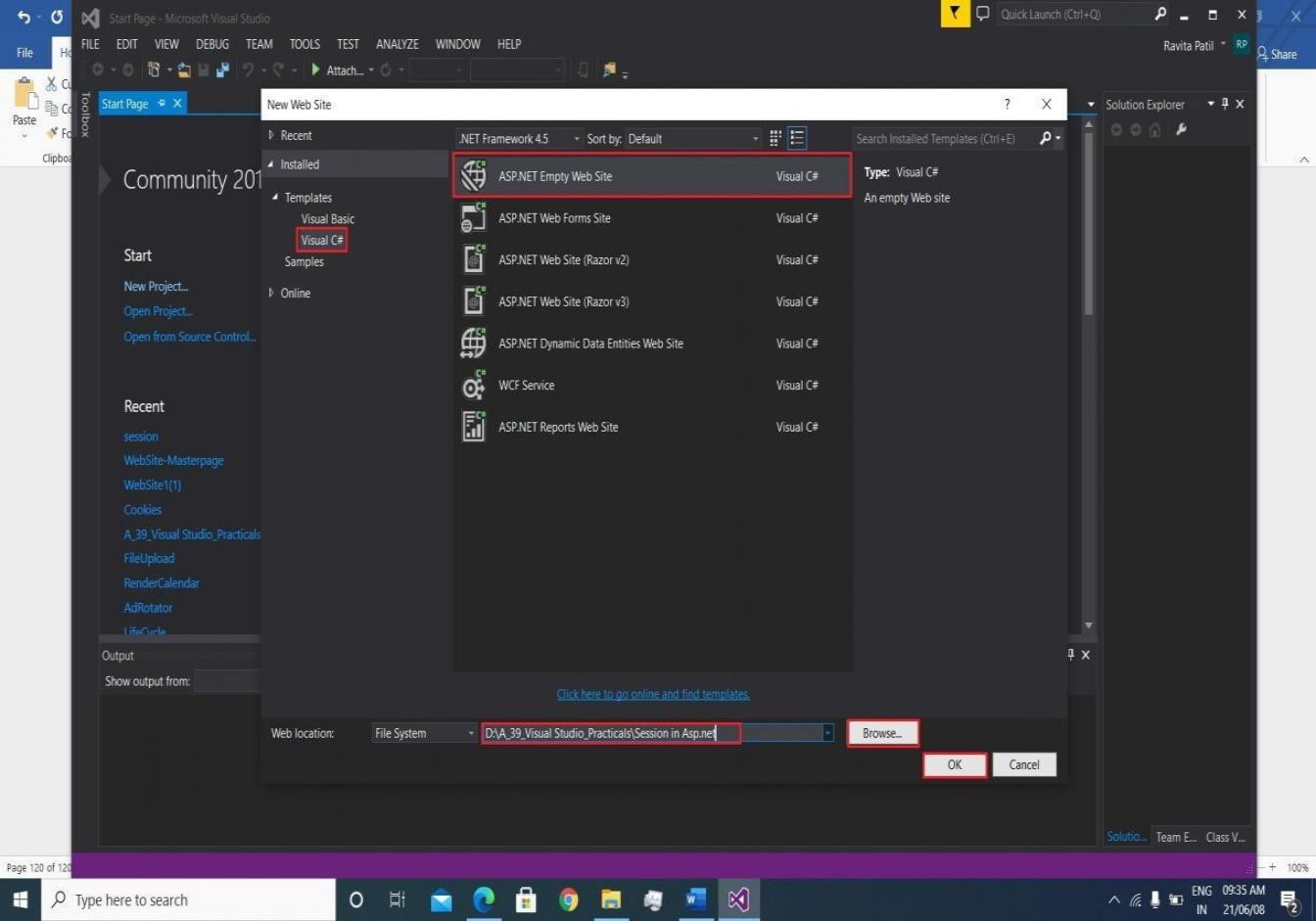
}

## OUTPUT-

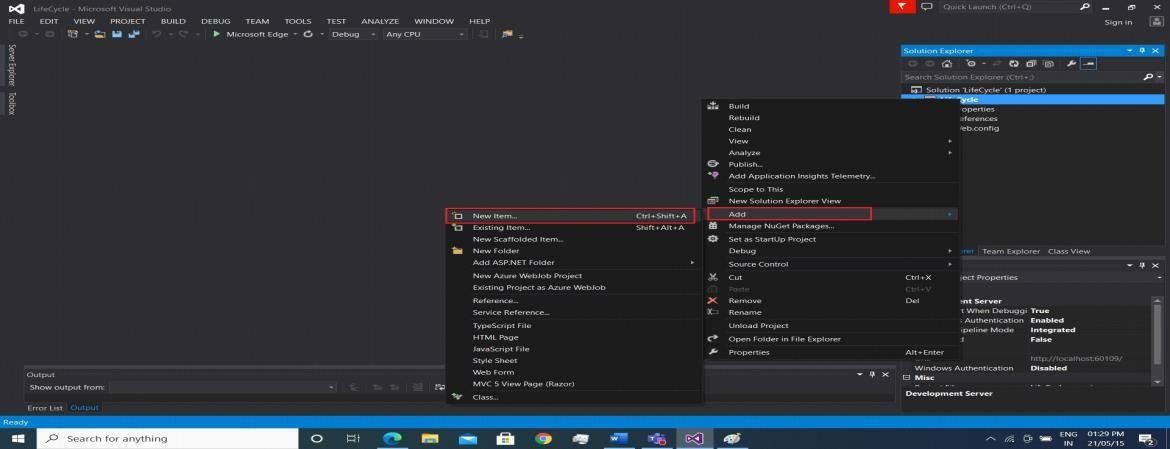
* 1. **Design Web Application using Server Side Session Management Techniques(login) Steps-**

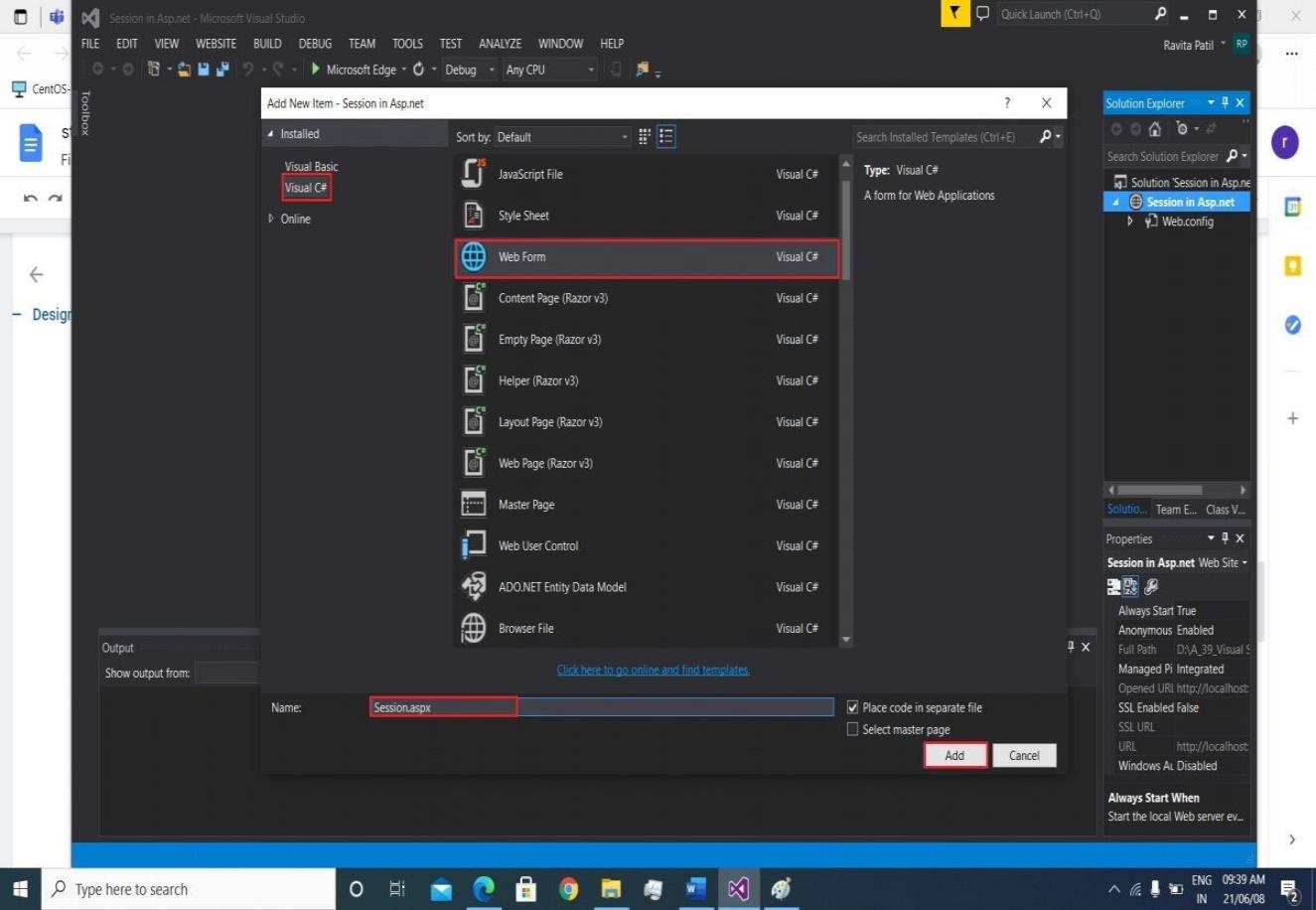
First open your visual studio-->File-->New-->Website-->Select ASP.NET Empty Website-- >Ok





open solution explorer-->Add New 2 Web Form-->Drag and Drop from Toolbox WebForm Session.aspx and Usesession.aspx





## INPUT-

**Filename-Session.aspx.cs**

using System;

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

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

Session["id"] = TextBox1.Text; Session["Pass"] = TextBox2.Text; Response.Redirect("Usesession.aspx"); Session.RemoveAll();

}

}

**Filename- UserSession.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

Label1.Text = Session["id"].ToString();

}

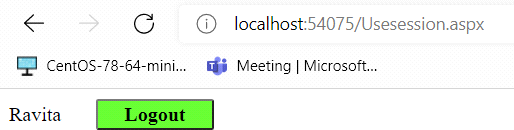
protected void Button1\_Click(object sender, EventArgs e)

{

Session.RemoveAll(); Response.Redirect("Session.aspx");

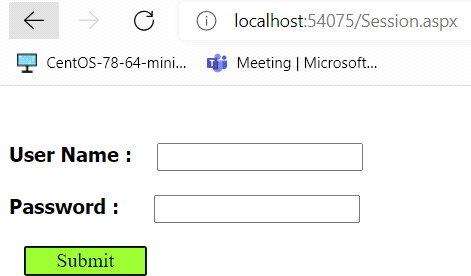
}

}



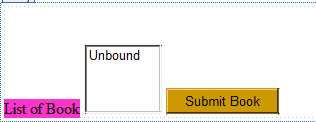
## OUTPUT-

**After click on Logout button**



1. **Design Web Application to show a list of books.on click of submit button selected book should be added in session and book details should be displayed on another page.**

**Input:**





**book.cs:**

using System;

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

using System.Web; using System.Web.UI;

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

using System.Web.UI.WebControls;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

ListBox1.Items.Add("Advanced .net"); ListBox1.Items.Add("ASP.NET"); ListBox1.Items.Add("Advance JAVA"); ListBox1.Items.Add("ADBMS"); ListBox1.Items.Add("Python");

}

protected void Button1\_Click(object sender, EventArgs e)

{

Session["bookname"] = ListBox1.SelectedItem; if (Session["bookname"] == null)

{

Response.Write("<script>alert('Book NOT selected')</script>");

}

else

{

Server.Transfer("~/Book2.aspx");

}

}

}

**Book2.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Book2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

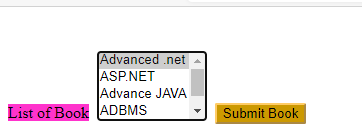
{

string book = Session["bookname"].ToString(); Label1.Text = book;

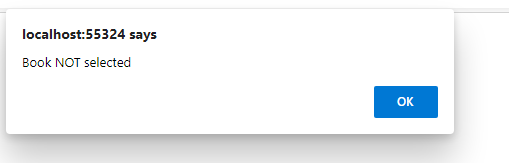
}

}

**Output:**

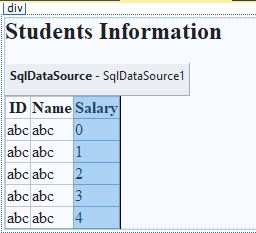






1. **Design Web Application to read 10 students info from database table into a gridview.cache this output for 10 sec.use cache object.**

**Input:**



**.CS File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Collections;

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

{

protected void Page\_Load(object sender, EventArgs e)

{

ArrayList student;

if (Cache["stud"] == null)

{

student = new ArrayList(); student.Add(DateTime.Now.AddSeconds(10)); Cache.Add("stud", student, null,

System.Web.Caching.Cache.NoAbsoluteExpiration, new TimeSpan(0, 0, 60), System.Web.Caching.CacheItemPriority.Default, null);

}

else

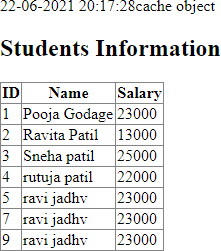
student = (ArrayList)Cache["stud"]; foreach(DateTime dt in student)

Response.Write(dt.ToString()+"cache object<br />");

}

}

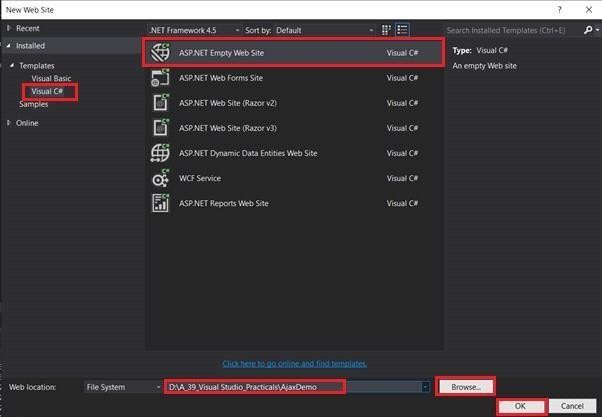
**Output:**



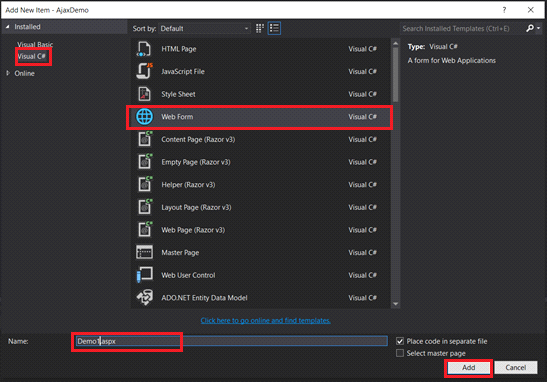
**6. Design Web Application using ASP.NET Ajax Controls Steps-**

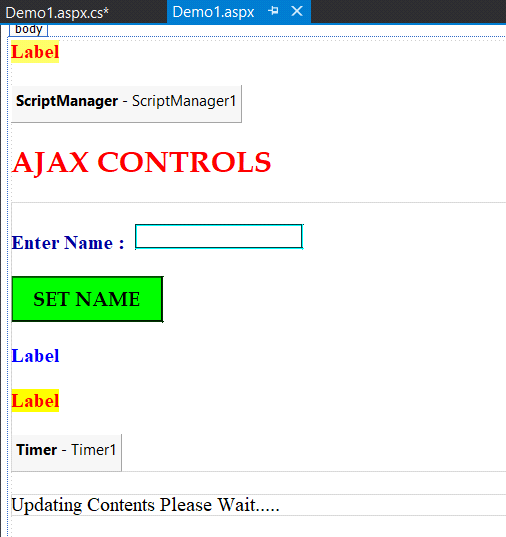
First open your visual studio-->File-->New-->Website-->Select ASP.NET Empty Website-- >Ok





open solution explorer-->Add New Web Form-->Drag and Drop from Toolbox





## INPUT-

**Filename- Demo1.aspx.cs**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Demo1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Label1.Text = "Page loaded time : " + DateTime.Now.ToString("T");

}

protected void Button1\_Click(object sender, EventArgs e)

{

System.Threading.Thread.Sleep(3000); Label3.Text = "Name Updated : "+ TextBox1.Text;

}

protected void Timer1\_Tick(object sender, EventArgs e)

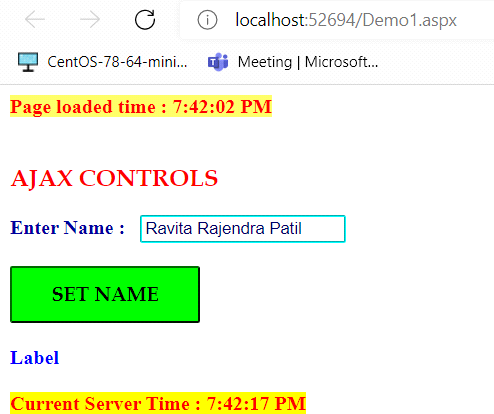
{

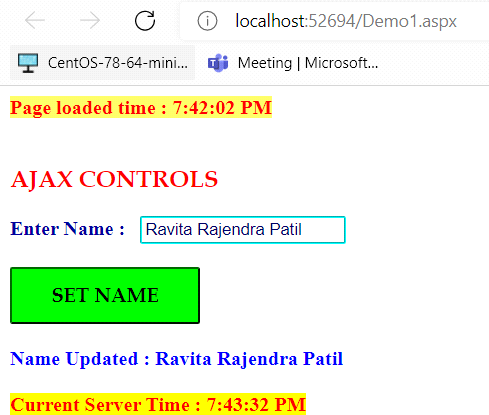
Label4.Text = "Current Server Time : "+ DateTime.Now.ToString("T");

}

}

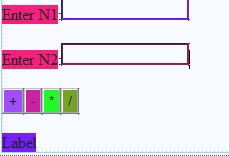
## OUTPUT-



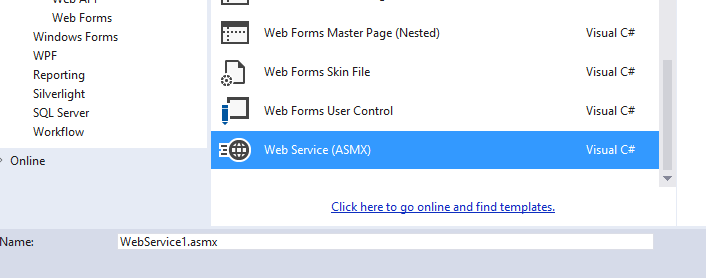


**After click on Set Name button**

1. **Module: Web Services and WCF**
2. **Create xml based webservice to create calculator and consume it in website.**



**Add WebService File:**



**.Asmx File:**

using System;

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

using System.Web;

using System.Web.Services;

namespace WebApplication3

{

/// <summary>

/// Summary description for WebService1

/// </summary>

[WebService(Namespace = "<http://tempuri.org/>")] [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)] [System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class WebService1 : System.Web.Services.WebService

{

[WebMethod]

public string HelloWorld()

{

return "Hello World";

}

[WebMethod]

public double add(double i, double j)

{

return i + j;

}

[WebMethod]

public double substract(double i, double j)

{

return i - j;

}

[WebMethod]

public double multiply(double i, double j)

{

return i \* j;

}

[WebMethod]

public double Division(double i, double j)

{

return i / j;

}

}

}

**.aspx File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication3

{

public partial class WebForm1 : System.Web.UI.Page

{

WebService1 obj = new WebService1();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

double sum = obj.add(Convert.ToDouble(TextBox1.Text), Convert.ToDouble(TextBox2.Text));

Label3.Text = "Your Addition Result=" + sum.ToString();

}

protected void Button2\_Click(object sender, EventArgs e)

{

double sub = obj.substract(Convert.ToDouble(TextBox1.Text), Convert.ToDouble(TextBox2.Text));

Label3.Text = "Your substraction Result=" + sub.ToString();

}

protected void Button3\_Click(object sender, EventArgs e)

{

double mul = obj.multiply(Convert.ToDouble(TextBox1.Text), Convert.ToDouble(TextBox2.Text));

Label3.Text = "Your multiplication Result=" + mul.ToString();

}

protected void Button4\_Click(object sender, EventArgs e)

{

double div = obj.Division((Convert.ToDouble(TextBox1.Text)), Convert.ToDouble(TextBox2.Text));

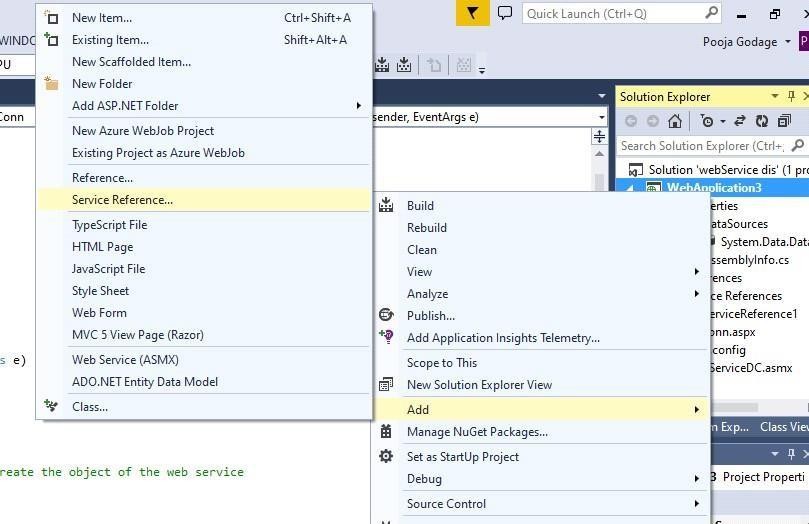
Label3.Text = "Your Division Result=" + div.ToString();

}

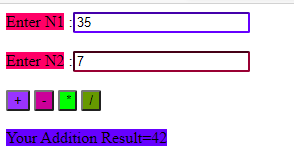
}

}

**Add Service Reference:**

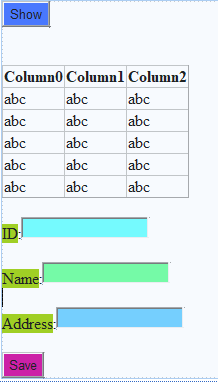


**Output:**

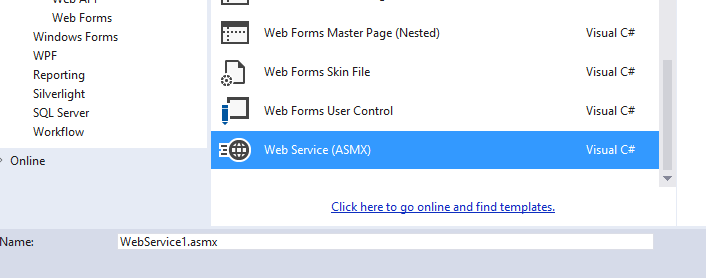


1. **Design a Web Service to Fetch & insert Details of Students Table using ADO.NET. Design a Web Client to show contents of table in a Grid View.**

**Input:**



**Add WebService File:**



**.asmx file:**

using System;

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

using System.Web;

using System.Web.Services; using System.Data.SqlClient; using System.Data;

namespace WebApplication3

{

/// <summary>

/// Summary description for WebServiceDC

/// </summary>

[WebService(Namespace = "<http://tempuri.org/>")] [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)] [System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class WebServiceDC : System.Web.Services.WebService

{

SqlConnection con; SqlDataAdapter adap; DataSet ds; SqlCommand cmd;

[WebMethod]

public DataSet GetSmember()

{

con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\webse rvice.mdf;Integrated Security=True;Connect Timeout=30");

adap = new SqlDataAdapter("select \* from EMP", con); ds = new DataSet();

adap.Fill(ds, "EMP"); return ds;

}

[WebMethod]

public int SaveData(int id, string na, string ad)

{

con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\webse rvice.mdf;Integrated Security=True;Connect Timeout=30");

con.Open();

cmd = new SqlCommand("insert into EMP values(" + id + ",'"

+ na + "','" + ad + "')", con);

int temp = cmd.ExecuteNonQuery(); return temp;

}

}

}

**.Aspx File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApplication3

{

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

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void TextBox1\_TextChanged(object sender, EventArgs

e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

WebServiceDC mys = new WebServiceDC(); // you need to create the object of the web service

mys.GetSmember();

GridView1.DataSource = mys.GetSmember(); GridView1.DataBind();

}

protected void Button2\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(TextBox1.Text); string na = TextBox2.Text;

string ad = TextBox3.Text;

WebServiceDC myservice = new WebServiceDC(); // you need to create the object of the web service

int roweffected = myservice.SaveData(id, na, ad); if (roweffected == 1)

{

Label1.Text = "Record is saved";

}

else

{

Label1.Text = "sorry record is not saved try again";

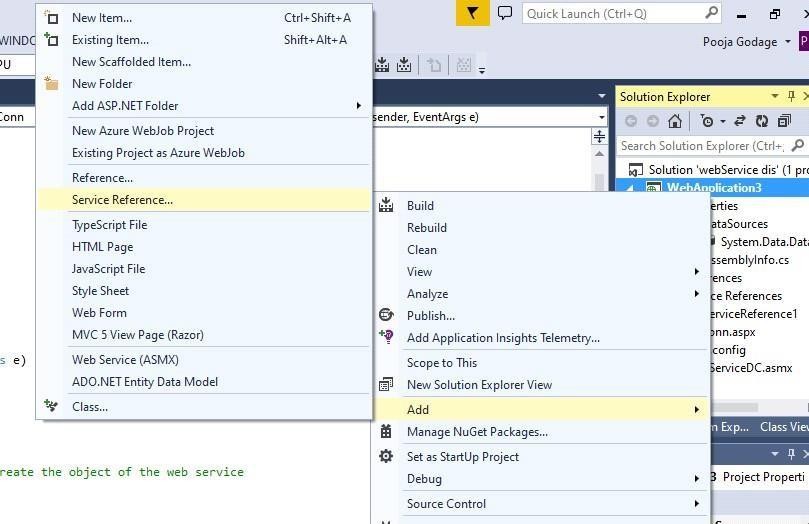
}

}

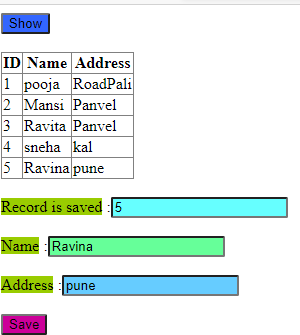
}

}

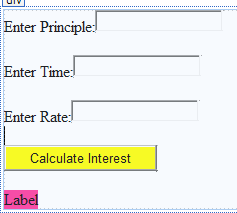
**Add Service Reference:**



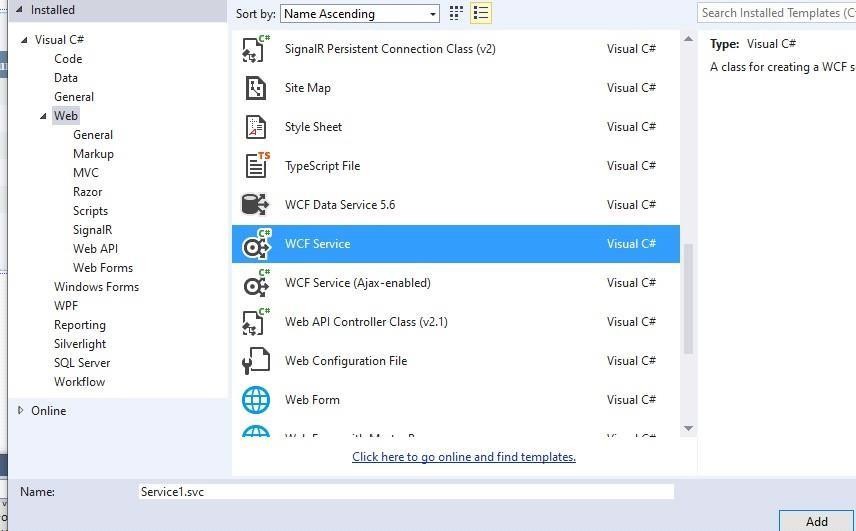
**Output:**



1. **Create and Consume the WCF Web Service to Calculate Simple Interest Input:**



**Add WCF Service:**



**Service1.cs:**

using System;

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

using System.Runtime.Serialization; using System.ServiceModel;

using System.Text;

namespace wcf\_service

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IService1" in both code and config file together.

[ServiceContract]

public interface IService1

{

[OperationContract]

string GetName(string name);

[OperationContract]

double getSI(double p, double t, double r);

}

}

**Service1svc.cs:**

using System;

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

using System.Runtime.Serialization; using System.ServiceModel;

using System.Text;

namespace wcf\_service

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "Service1" in code, svc and config file together.

// NOTE: In order to launch WCF Test Client for testing this service, please select Service1.svc or Service1.svc.cs at the Solution Explorer and start debugging.

public class Service1 : IService1

{

public string GetName(string name)

{

return string.Format(name);

}

public double getSI(double p, double t, double r)

{

return ((p \* t \* r) / 100);

}

}

}

**.CS File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls;

namespace wcf\_service

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

ServiceReference1.Service1Client ob = new ServiceReference1.Service1Client();

protected void Button1\_Click(object sender, EventArgs e)

{

string name = ob.GetName("Pooja!!"); Response.Write("Hello " + name);

double p = Convert.ToDouble(TextBox1.Text); double t = Convert.ToDouble(TextBox2.Text); double r = Convert.ToDouble(TextBox3.Text); double si = ob.getSI(p, t, r);

Label4

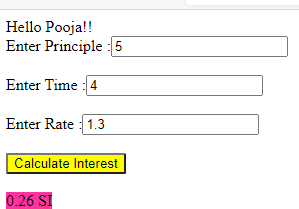
.Text = si.ToString() + " SI ";

}

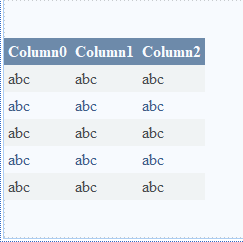
}

}

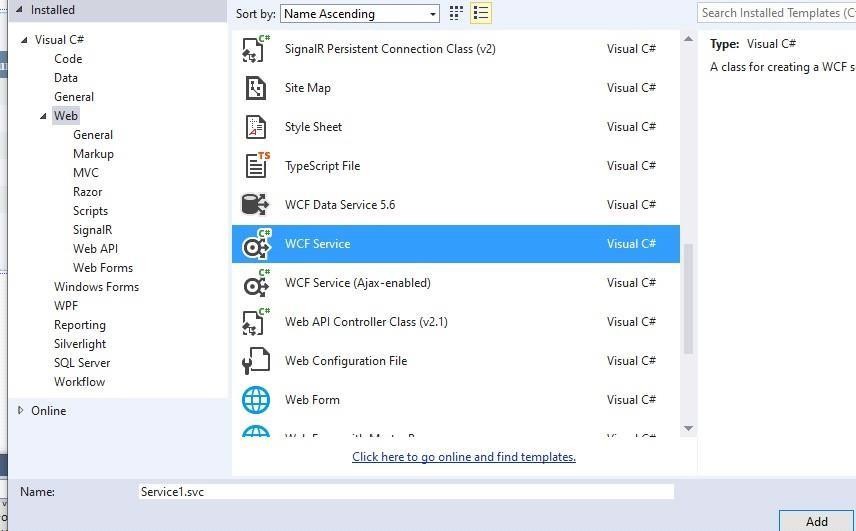
**Output:**



1. **Create a WCF Web Service Using Database Input:**



**Add WCF Service:**



**ServiceDB.cs:**

using System;

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

using System.Runtime.Serialization; using System.ServiceModel;

using System.Text;

using System.Data;

namespace WCF\_DB

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the interface name "IServiceDB" in both code and config file together.

[ServiceContract]

public interface IServiceDB

{

[OperationContract] student GetStudent();

}

[DataContract] public class student

{

[DataMember]

public DataTable StudentTable

{

get; set;

}

}

}

**ServiceDB.svc.cs:**

using System;

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

using System.Runtime.Serialization; using System.ServiceModel;

using System.Text; using System.Data;

using System.Data.SqlClient;

namespace WCF\_DB

{

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class name "ServiceDB" in code, svc and config file together.

// NOTE: In order to launch WCF Test Client for testing this service, please select ServiceDB.svc or ServiceDB.svc.cs at the Solution Explorer and start debugging.

public class ServiceDB : IServiceDB

{

SqlConnection con; SqlDataAdapter ad; SqlCommand cmd; DataTable dt;

student st = new student();

public student GetStudent()

{

con = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Admin\Documents\webse rvice.mdf;Integrated Security=True;Connect Timeout=30");

cmd = new SqlCommand("SELECT \* FROM [EMP]",con); ad = new SqlDataAdapter(cmd);

dt = new DataTable("e"); ad.Fill(dt); st.StudentTable = dt; return st;

}

}

}

**.CS File:**

using System;

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

using System.Web; using System.Web.UI;

using System.Web.UI.WebControls; using System.Data;

namespace WCF\_DB

{

public partial class WebForm1 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

ServiceReference3.ServiceDBClient ob = new ServiceReference3.ServiceDBClient();

ServiceReference3.student st = new ServiceReference3.student();

st = ob.GetStudent();

DataTable dt = new DataTable();

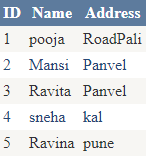
dt = st.StudentTable; GridView1.DataSource = dt.DefaultView; GridView1.DataBind();

}

}

}

**Output:**



1. **Module: ASP.NET MVC**

**1)MVC Application With Model view controller display information(students info)**

**Add->new project->asp.net web application->MVC->ok. Model->ADD->Class**

**Controller->ADD**

**Add Class File: Student.cs**

using System;

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

using System.Web;

namespace MvcStudInformation.Models

{

public class Student

{

public int studentID { get; set; } public string studentName { get; set; } public int studentAge { get; set; }

}

}

**Add Controller StudentController.cs**

using MvcStudInformation.Models; using System;

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

using System.Web; using System.Web.Mvc;

namespace MvcStudInformation.Controllers

{

public class StudentController : Controller

{

// GET: Student

public ActionResult Index()

{

ViewBag.Itemlist = "student information page"; IList<Student> studentList = new List<Student>

{

new Student(){studentID=1,studentName="Pooja",studentAge=22},

new Student(){studentID=2,studentName="Mansi",studentAge=20},

new Student(){studentID=3,studentName="Ravita",studentAge=21}

};

return View(studentList);

}

}

}

**Index.html:**

@foreach(var i in Model)

{

<h1>@ViewBag.ItemList</h1>

<b>Student ID:</b>@i.studentID

<br />

<b>Student Name</b>@i.studentName

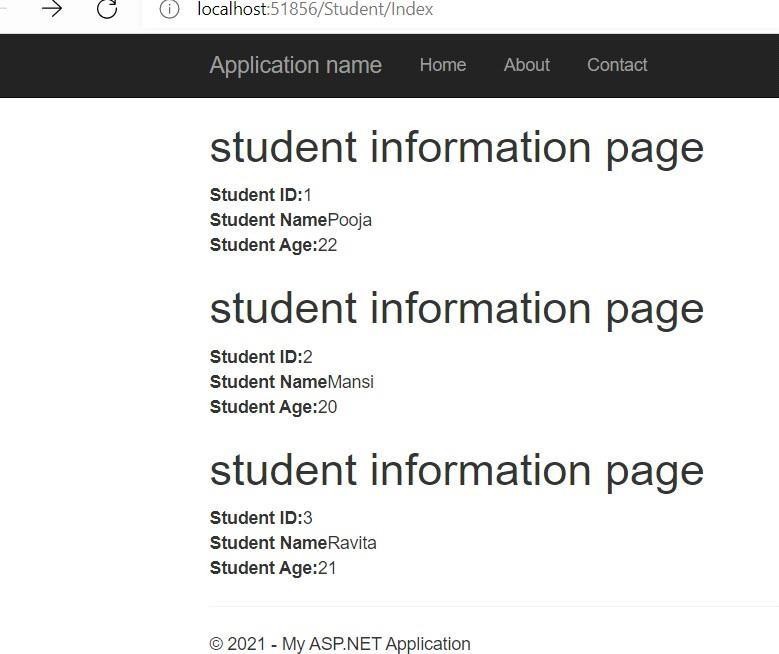
<br />

<b>Student Age:</b>@i.studentAge

<br />

}

Output:



1. **MVC Application foí customeí data entíy using HľML helpeí and validation**

**Add Class File Customer.cs:**

using System;

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

using System.Web;

using System.ComponentModel.DataAnnotations; namespace MVCCustomer.Models

{

public class Customer

{

[Required(ErrorMessage = "Please Insert Customer ID!")] public int CustID { get; set; }

[Required(ErrorMessage = "Please Insert Customer Name!")] public String CustName { get; set; }

[Required(ErrorMessage = "Please Insert Customer Address!")] public String CustAdd { get; set; }

}

}

**Add Controller**

using System;

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

using System.Web; using System.Web.Mvc;

using MVCCustomer.Models;

namespace MVCCustomer.Controllers

{

public class CustomerController : Controller

{

// GET: Customer

public ActionResult Index()

{

return View();

}

[HttpGet]

public ViewResult CustomerInput()

{

return View();

}

[HttpPost]

public ViewResult CustomerInput(Customer c1)

{

if (ModelState.IsValid)

{

return View("CustomerDisplay", c1);

}

else

{

return View();

}

}

}

**Index.HTML:**

<html>

<body>

<div class="text-center">

<p>

<h1/>

<div class="text-center"/>

<p><h1>Customer Information System </h1><br /> </p>

<div class="btn btn-success"> @Html.ActionLink("Add Customer Details",

"CustomerInput")

</div>

</div>

</body>

</html>

**CustomerInput.html:**

@model MVCCustomer.Models.Customer @{

ViewBag.Title = "CustomerDisplay";

}

@using (Html.BeginForm())

{

@Html.ValidationSummary()

<div class="form-group">

<label>Customer ID :</label>@Html.TextBoxFor(x => x.CustID, new { @class = "form-control" })

</div>

<div class="form-group">

<label>Customer Name :</label>@Html.TextBoxFor(x => x.CustName, new { @class = "form-control " })

</div><div class="form-group">

<label>Customer Address:</label>@Html.TextBoxFor(x => x.CustAdd, new { @class = "form-control" })

</div>

<div class="text-center">

<input class="btn btn-success" type="submit" value="Submit Customer Data" />

</div>

}

**CustomerDisplay.html:**

@model MVCCustomer.Models.Customer

@{

ViewBag.Title = "CustomerDisplay";

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>CustomerDisplay</title>

</head>

<body>

<div class="text-center">

<h1>Customer Information System</h1>

<p>Customer ID : @Model.CustID</p>

<p>Customer Name : @Model.CustName</p>

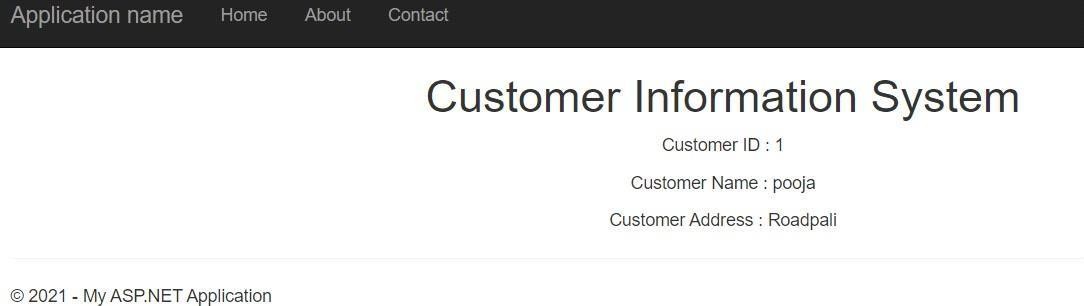
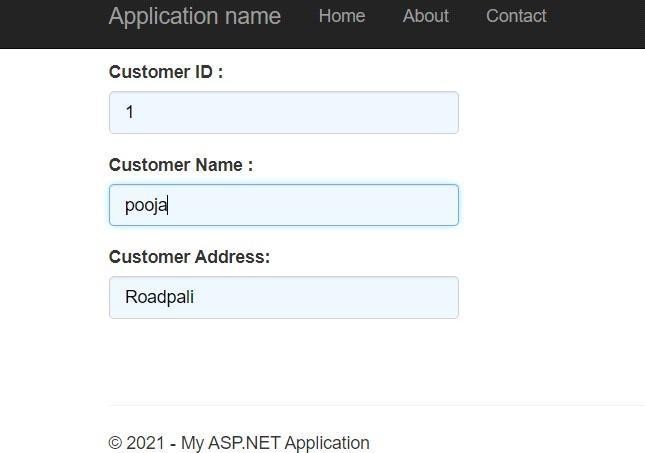
<p>Customer Address : @Model.CustAdd</p>

</div>

</body>

</html>

**Output:**



1. **Build MVC Application to fíom CURD opeíations using EÏ.**

**Emp.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations; using System.Linq;

using System.Web;

namespace WebApplication1.Models

{

public class Emp

{

[Key]

public virtual int EId { get; set; }

public virtual int Experience { get; set; } [Display(Name = "EmpName")]

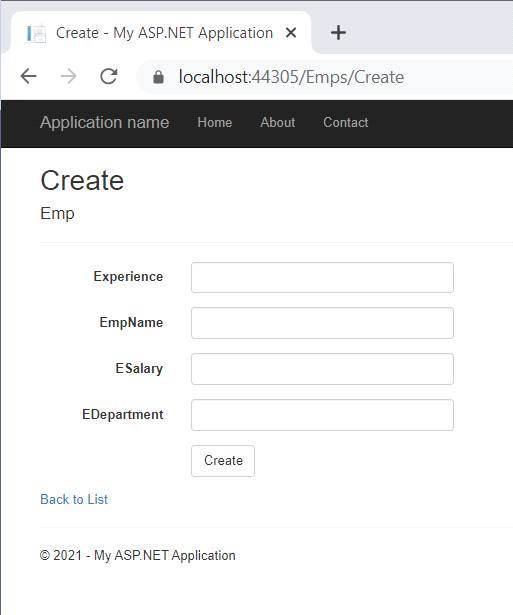
public virtual string EmpName { get; set; } public virtual int ESalary { get; set; }

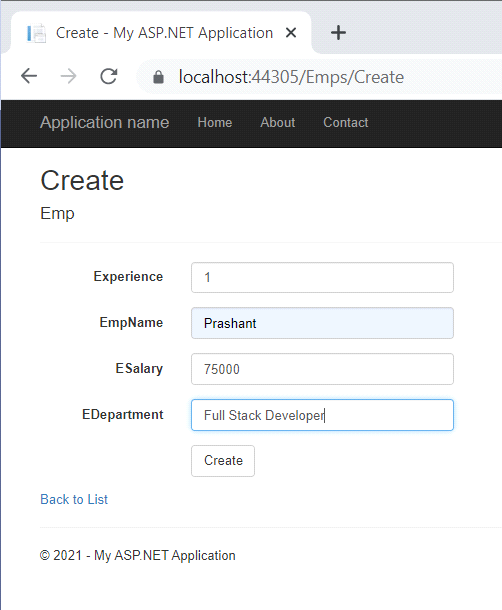
public virtual string EDepartment { get; set; }

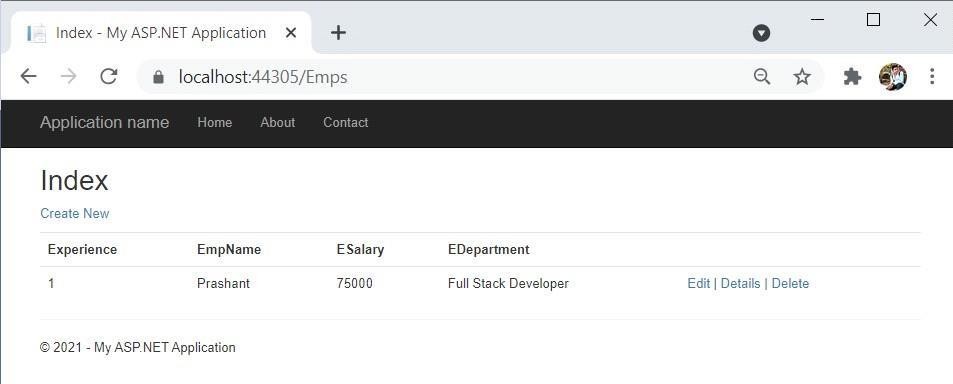
}

}

**Output:**







**Angular n JS Query**

1. **Build an application Using JQuery.(Table even odd row format,Filter())**

**Add->project->c#->ASP.net web Application->Web form->then**

**Right Click->project->ADD->Web form.**

.aspx Page

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="oddeven.aspx.cs" Inherits="jquery.oddeven" %>

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml)

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<h3>jQuery to cahnge background color of odd even rows</h3>

<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script type="text/javascript">

$(function ()

{

$("table#table1 tr:even").css("background-color", "YELLOW");

$("table#table1 tr:odd").css("background-color", "PINK");

$("table").css("width", "50%");

$("table").attr("border", "true");

});

</script>

<div>

<table id="table1">

<tr><td>Book Number</td>

<td>1001</td></tr>

<tr><td>Isbn</td>

<td>AA</td></tr>

<tr><td>Author</td>

<td>suyash</td></tr>

<tr><td>Publisher</td>

<td>DreamTech</td></tr>

<tr><td>cost</td>

<td>1000</td></tr>

<tr><td>Copies</td>

<td>10</td></tr>

</table>

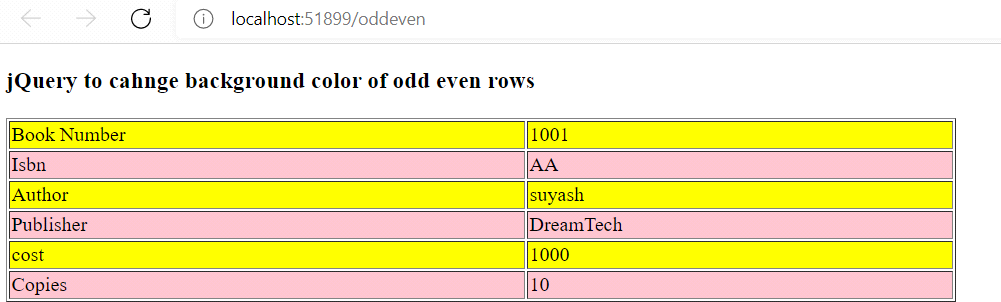
</div>

</form>

</body>

</html>

Output:



1. **Build an angular web application.(basic calculator)**

**.html page**

<!DOCTYPE html>

<html xmlns[="http://www.w3.org/1999/xhtml">](http://www.w3.org/1999/xhtml)

<head>

<title></title>

<script src='https://ajax.googleapis.com/ajax/libs/angularjs/1.5.0- rc.0/angular.min.js'></script>

</head>

<body ng-app>

<h2>Arithmetic Expression In Angularjs</h2> <br />

<h3>numeric</h3>

10 + 20 = **{{**10+20**}}**

<br /><h3>Subtraction</h3>

30 - 20 = **{{**30-20**}}**

<br /><h3>Multiply</h3>

10 \* 20 = **{{**10\*20**}}**

<br /><h3>Division</h3>

20 / 10 = **{{**20/10**}}**

<br />Enter Numbers for Addition:

<input type="text" ng-model=Num1 /> + <input type="text" ng-model=Num2 />

= <span>**{{**Num1 + Num2**}}**</span>

<br />Enter Numbers for Substracion:

<input type="text" ng-model="Num3" /> - <input type="text" ng-model="Num4" />

= <span>**{{**Num3 - Num4**}}**</span>

<br />Enter Numbers for Multiplication:

<input type="text" ng-model="Num5" /> \* <input type="text" ng-model="Num6" />

= <span>**{{**Num5 \* Num6**}}**</span>

<br />Enter Numbers for Division:

<input type="text" ng-model="Num1" /> / <input type="text" ng-model="Num2" />

= <span>**{{**Num1 / Num2**}}**</span>

</body>

</html>

**Output:**

