

ADVANCED WEB TECHNOLOGIES LAB

Subject Code: MCAL24

A Project Report Submitted in Fulfillment of the Degree of

MASTER

In COMPUTER APPLICATION

Year 2023-2024

By

Mr. KETAN RAVINDRA DAKI

(Application Id-78749)

Semester-II

Under the Guidance of

Prof. Dnyaneshwar Deore



Institute of Distance and Open Learning
Vidya Nagari, Kalina, Santacruz East – 400098.
University of Mumbai

PCP Center

[Satish Pradhan Dyanasadhana College, Thane]



Institute of Distance and Open Learning Vidya
Nagari, Kalina, Santacruz East – 400098.

CERTIFICATE

This to certify that, "**KETAN RAVINDRA DAKI**" appearing **Master's in Computer Application (Semester II) Application ID: 78749** has satisfactorily completed the prescribed practical of **MCAL24 –Advanced Web Technologies Lab** as laid down by the University of Mumbai for the academic year 2023-24.

Teacher In Charge

External Examiner

Coordinator – M.C.A

Index

Practical	AIM	Signature
1	Design UI based applications using basic Windows forms Controls	
2	Design Applications using Classes and Objects	
3	Design Applications using Inheritance and Abstract Classes.	
4	Design online registration form using Master Page and Validations.	
5	Design simple angular web application	
6	Design web applications to demonstrate connected and disconnected architecture in Ado.net	
7	Design web applications to demonstrate the use of data bound controls and procedures	
8	Design web applications to demonstrate the use of LINQ to SQL and ADO.NET Entity Framework	
9	Design web applications to demonstrate Client and Server side State Management Techniques.	
10	Design Web Application to produce and consume a web Service	
11	Design Web Application to produce and Consume a web Service	
12	Design Web Applications to demonstrate Model View Controller	

Practical : 1

AIM: Design UI based applications using basic Windows forms Controls

A) WAP in C# that ask the user to enter a month, a day and a two digit year. The program should then determine whether the month times a day is equal to the year. If so, it should display the message saying the date is magic. Otherwise not a magic.

Code :

```
using System;
using System.Collections.Generic;
using System.ComponentModel; using
System.Data; using System.Drawing;
using System.Linq; using
System.Text; using
System.Windows.Forms;
namespace WindowsFormsApplication1
{ public partial class Form1 : Form
    { public Form1()
        {
            InitializeComponent(); label1.Visible
            = false;
        }
    private void button1_Click(object sender, EventArgs e)
        { int d = Int32.Parse(textBox1.Text);
            int m = Int32.Parse(textBox2.Text);
            int y = Int32.Parse(textBox3.Text);
            label1.Visible = true; if(d*m==y)
            {
                label1.Visible = true;      label1.Text      =
"THE DATE IS MAGIC";
            } else
            {
                label1.Visible = true;
            }
        }
    }
}
```

OUTPUT:

 ADDITION

ENTER DATE

ENTER MONTH

ENTER YEAR

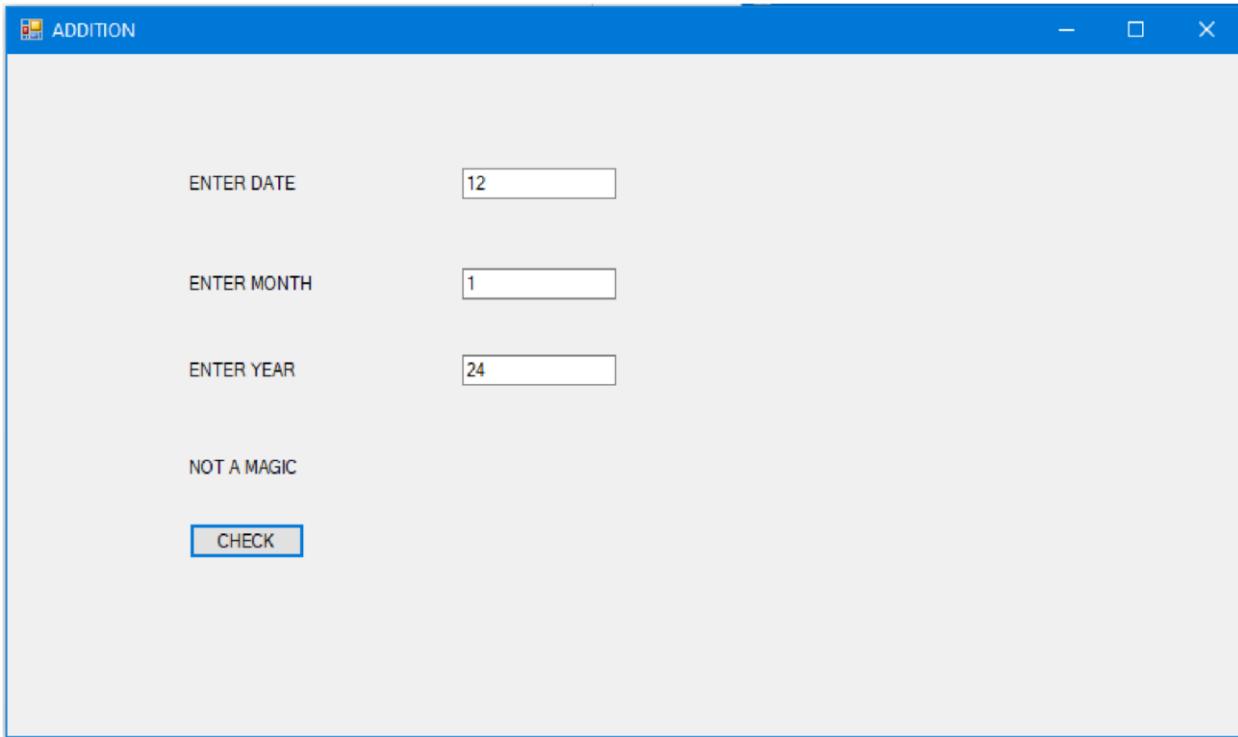
 ADDITION

ENTER DATE

ENTER MONTH

ENTER YEAR

THE DATE IS MAGIC



B) Write a Program to perform Money Conversion.

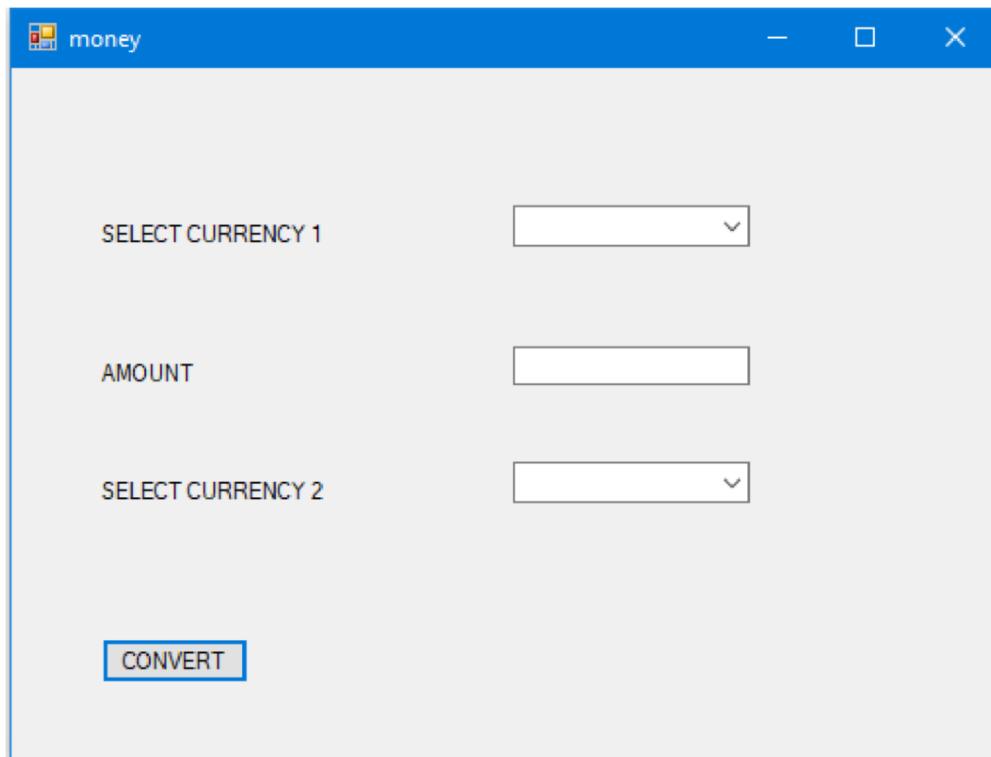
CODE:

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using System.Drawing;
using System.Linq; using
System.Text; using
System.Windows.Forms;
namespace WindowsFormsApplication1
{ public partial class money : Form
    { public money()
        {
            InitializeComponent(); label2.Visible
            = false;
        }

        private void button1_Click(object sender, EventArgs e) {
double inr = 72.91; double dol
            = 0.014;
            double val = double.Parse(textBox1.Text);
            String s1 = comboBox1.Text;
            String s2 = comboBox2.Text;
            if (s1 == s2)
            { label2.Visible = true;
                label2.Text = "Please select two different currencies";
            } else if (s1 == "INR")
```

```
        {
            label2.Visible = true;
            label2.Text = "Value in dollar is: $" + val*dol;
        } else
        {
            label2.Visible     =
true;
        }
    }
}
```

OUTPUT:



money

SELECT CURRENCY 1

AMOUNT

SELECT CURRENCY 2

CONVERT Please select two different currencies

money

SELECT CURRENCY 1

AMOUNT

SELECT CURRENCY 2

CONVERT Value in dollar is: \$4.97

C) To convert temperature from Fahrenheit to Celsius or vice versa.

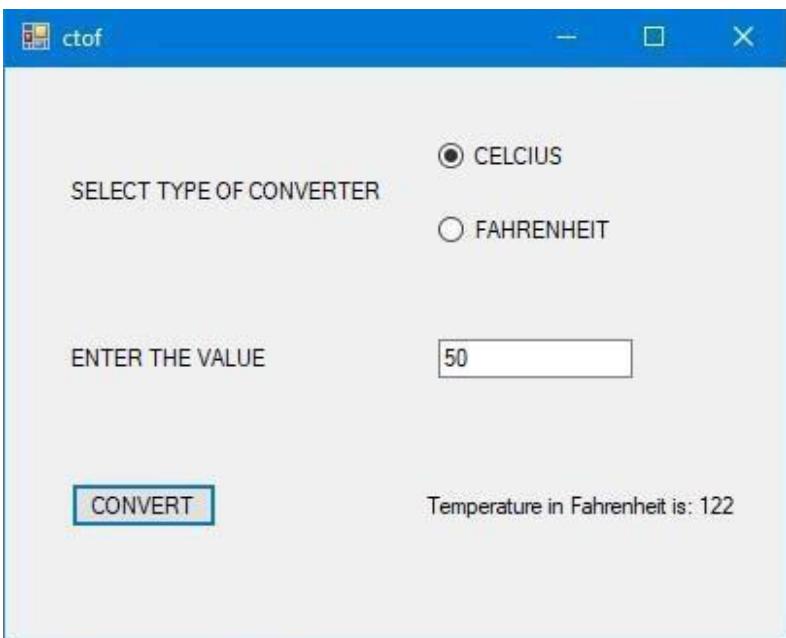
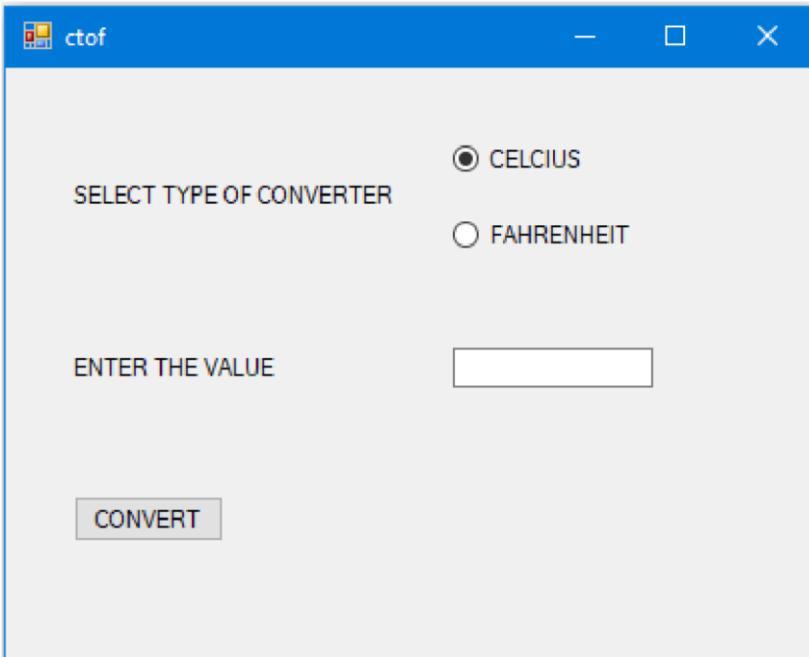
CODE:

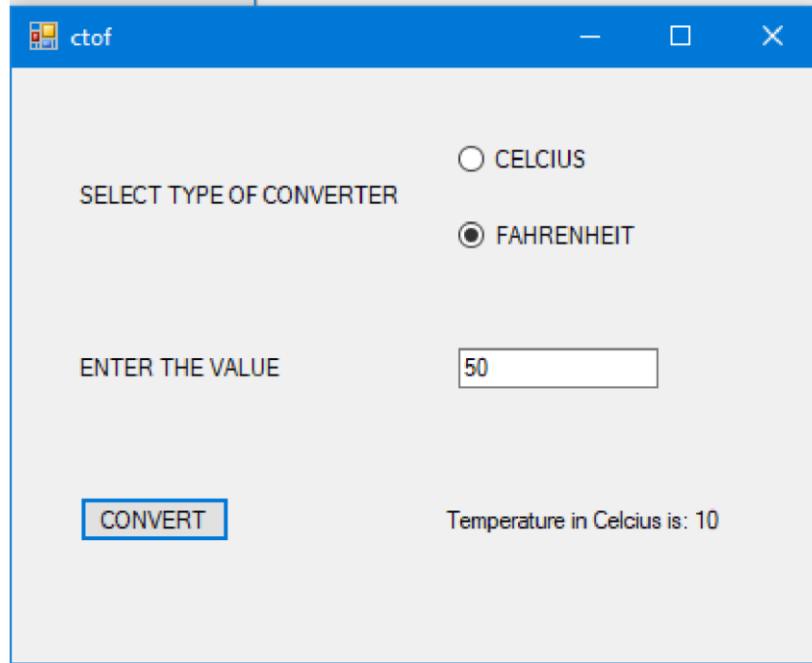
```
using System;           using
System.Collections.Generic;   using
System.ComponentModel;       using
System.Data;  using System.Drawing;
using System.Linq;         using
System.Text;            using
System.Windows.Forms;
namespace WindowsFormsApplication1
{ public partial class ctof : Form
    { public ctof()
        {
            InitializeComponent(); label3.Visible
            = false;
        }

        private void button1_Click(object sender, EventArgs e)
        { if (radioButton1.Checked)
            { label3.Visible = true;
label3.Text = "Temperature in Fahrenheit is:
"+Math.Round(((float.Parse(textBox1.Text) * 1.8) + 32),2).ToString(); }
if(radioButton2.Checked)
            { label3.Visible = true;
                label3.Visible = true;

                label3.Text = "Temperature in Celcius is:
"+Math.Round(((float.Parse(textBox1.Text)-32)*0.5556),2).ToString();
            }
        }
    }
}
```

OUTPUT:





D) Create a Window application to calculate age of a person by providing input as birth date and current date .Current date and Birth date must be in long string format and display the age in terms of years:

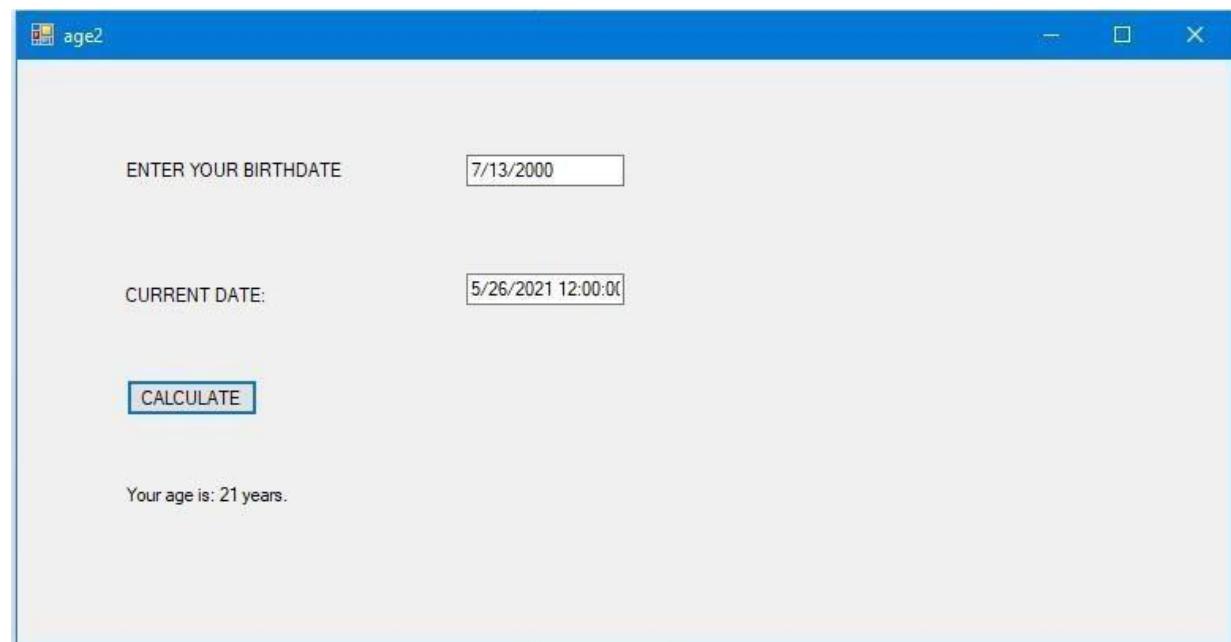
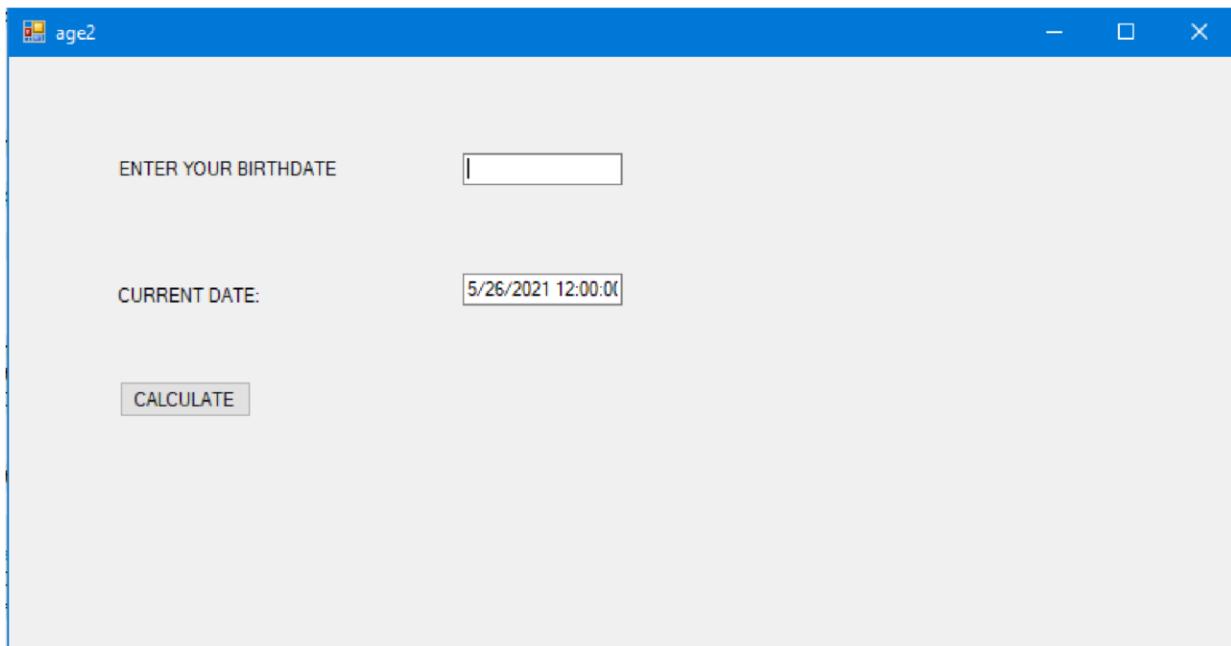
CODE:

```
using System;
using System.Collections.Generic;
using System.ComponentModel; using
System.Data; using System.Drawing;
using System.Linq; using
System.Text; using
System.Windows.Forms;
namespace WindowsFormsApplication1
{ public partial class age2 : Form
{ public age2()
{
    InitializeComponent();
    textBox2.Text = DateTime.Today.ToString(); label2.Visible
    = false;
}

        private void button1_Click(object sender, EventArgs e)
{
    DateTime d = DateTime.Parse(textBox1.Text);
    var yr = DateTime.Today.Year - d.Year;
    label2.Visible = true;
    label2.Text = "Your age is: " + yr.ToString() + " years." ;
}
```

```
    }  
}
```

OUTPUT:



Practical : 2

AIM: Design Applications using Classes and Objects

A) Write a program to declare a class “staff” having data members as name and post. Accept this data 5 staffs and display names of staff who are HOD.

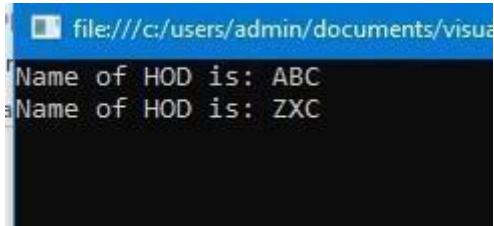
CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Text;
namespace ConsoleApplication4
{ class staff
{
    String name, post;
public staff(String n , String p)
    { this.name = n;
        this.post = p;
    }

void dis()
    { if (post == "HOD")
    {
        Console.WriteLine("Name of HOD is: "+name); }

    } static void Main(string[]
args)
{ staff[] s = new staff[5]; s[0] = new
    staff("ABC", "HOD"); s[1] = new
    staff("XYZ", "TEACHER"); s[2] = new
    staff("JKL", "LAB ASSISTANT"); s[3] = new
    staff("ZXC", "HOD"); s[4] = new
    staff("QWE", "DEP.HOD");
for (int i = 0; i <= 4; i++)
    { s[i].dis();
    }
    Console.ReadKey();
}
}
```

OUTPUT:



```
file:///c:/users/admin/documents/visual studio 2010/Projects/ConsoleApplication3/ConsoleApplication3.cs
Name of HOD is: ABC
Name of HOD is: ZXC
```

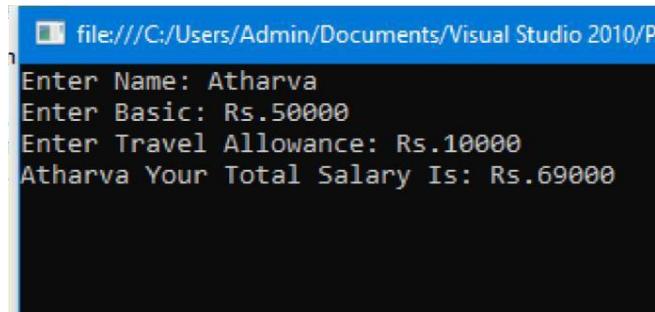
B) Define a class “salary” which will contain member variable Basic, TA, DA, HRA.
Write a program using Constructor with default values for DA and HRA and calculate the salary of the employee.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Text;
namespace ConsoleApplication3
{ class salary
    { int da, hra;
        public salary()
        { da = 5000;
            hra = 4000;
        }
    public static void Main(string[] args)
    {
        Console.Write("Enter Name: ");
        String name = Console.ReadLine();
        Console.Write("Enter Basic: Rs."); int basic =
Int32.Parse(Console.ReadLine());
        Console.Write("Enter Travel Allowance: Rs.");
        int ta = Int32.Parse(Console.ReadLine());
```

```
    salary s = new salary();
    Console.WriteLine(name+ " Your Total Salary Is:
Rs."+(s.da+s.hra+basic+ta)); Console.ReadKey(); }
}
}
```

OUTPUT:



Practical : 3

AIM: Design Applications using Inheritance and Abstract Classes.

A) Write a program to implement multilevel inheritance from following figure.

Accept and display data for one student.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Text;
namespace Student
{

class Student
{ public String name;
  public int roll_no;
public void get_data()
{
  Console.WriteLine("Enter Your Name: "); name
= Console.ReadLine();
  Console.WriteLine("Enter Your Roll No: "); roll_no
= Int32.Parse(Console.ReadLine());

}

}

class Test : Student
{ public int marks1,marks2;
  public void get_data2()
  {

    Console.WriteLine("Enter Marks 1: "); marks1 =
Int32.Parse(Console.ReadLine());
    Console.WriteLine("Enter Marks 2: "); marks2 =
Int32.Parse(Console.ReadLine()); }
}

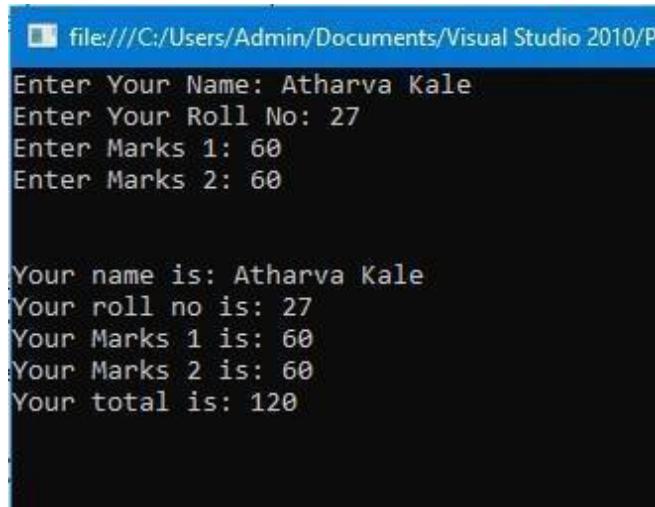
class Result : Test
{ public int total;
  public void add()
  { total = marks1 + marks2;
  }

  public void dis()
{ get_data();
  get_data2();
  add();
  Console.WriteLine("\n\nYour name is: "+name); }
```

```
        Console.WriteLine("Your roll no is: "+roll_no.ToString());
        Console.WriteLine("Your Marks 1 is: "+marks1.ToString());
        Console.WriteLine("Your Marks 2 is: "+marks2.ToString());
        Console.WriteLine("Your total is: "+total.ToString()); }
    }

class Program
{ static void Main(string[] args)
    {
        Result r = new Result();
        r.dis(); Console.ReadKey();
    }
}
}
```

OUTPUT:



```
file:///C:/Users/Admin/Documents/Visual Studio 2010/P
Enter Your Name: Atharva Kale
Enter Your Roll No: 27
Enter Marks 1: 60
Enter Marks 2: 60

Your name is: Atharva Kale
Your roll no is: 27
Your Marks 1 is: 60
Your Marks 2 is: 60
Your total is: 120
```

B) Create Super Class Student and two sub class of it, Graduate and Under Graduate. The members of the Student are name, id, grade, age and address and one method : boolean method IsPassed which takes in the parameter integer grade(0-100) and return true.The two sub classes over ride the method, for UG its 70% for passing and for G its 80% as passing grade.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Text;
namespace ConsoleApplication5
{
    class Student
    {
        public String name, address;
        public int id, age, grade;
        public void getdata()
        {
            Console.Write("Enter Your Name: "); name
            = Console.ReadLine();

            Console.Write("Enter your ID"); id=Int32.Parse(Console.ReadLine());

            Console.Write("Enter your Age"); age =
            Int32.Parse(Console.ReadLine());

            Console.Write("Enter your Grade"); grade
            = Int32.Parse(Console.ReadLine());

            Console.Write("Enter Your Address"); address
            = Console.ReadLine();
        }

        public void dis()
        {
            Console.WriteLine("\nYour Name: "+name); Console.WriteLine("Your
            ID: "+id);
            Console.WriteLine("Your Age: "+age);
            Console.WriteLine("Your Grade: "+grade);
            Console.WriteLine("Your Address: "+address);

        }
        public virtual Boolean check()
        {
            return true;
        }
    }

    class Undergraduate : Student
    {
        public override Boolean check()
        {
            getdata(); dis();
            if (grade >= 70)
            {

```

```

        Console.WriteLine("YOU ARE PASS");
    } else
    {
        Console.WriteLine("YOU ARE FAIL");
    }

    return true;
}

}

class Postgraduate : Student
{
    public override Boolean check()
    { getdata(); dis();
        if (grade >= 80)
        {
            Console.WriteLine("YOU ARE PASS");
        } else
        {
            Console.WriteLine("YOU ARE FAIL");
        }

        return true;
    }
}

class Program
{
    static void Main(string[] args)
    { int c;
        Console.WriteLine("Select Graduation:\n1.Undergraduate\n2.Postgraduate");
        c=Int32.Parse(Console.ReadLine()); switch(c)
        { case 1:
            Undergraduate u = new Undergraduate();
            u.check(); break;
        case 2:
            Postgraduate p = new Postgraduate();
            p.check(); break;

        }

        Console.ReadKey();
    }
}

```

OUTPUT:

```
file:///C:/Users/Admin/Documents/Visual Studio 201  
Select Graduation:  
1.Undergraduate  
2.Postgraduate  
  
1  
  
Enter Your Name: Atharva Kale  
Enter your ID: 27  
Enter your Age: 21  
Enter your Grade: 75  
Enter Your Address: Kalyan  
  
Your Name: Atharva Kale  
Your ID: 27  
Your Age: 21  
Your Grade: 75  
Your Address: Kalyan  
YOU ARE PASS
```

```
file:///C:/Users/Admin/Documents/Visual Studio 201  
Select Graduation:  
1.Undergraduate  
2.Postgraduate  
  
2  
  
Enter Your Name: Atharva Kale  
Enter your ID: 27  
Enter your Age: 21  
Enter your Grade: 50  
Enter Your Address: Kalyan  
  
Your Name: Atharva Kale  
Your ID: 27  
Your Age: 21  
Your Grade: 50  
Your Address: Kalyan  
YOU ARE FAIL
```

C) Program to calculate to find the area of various shape: Rectangle, Circle, Ellipse, Square and Triangle using abstract class and abstract method.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Text;
namespace ConsoleApplication6
{ abstract class Shape
    { public abstract void calc(); }

    class Rectangle : Shape
    {
        public override void calc()
        {
            Console.Write("Enter Length of Rectangle: ");
            float l = float.Parse(Console.ReadLine());
            Console.Write("Enter Breadth of Rectangle: "); float
            w = float.Parse(Console.ReadLine());
            Console.WriteLine("Area of Rectangle is: "+w*l); }

    }
    class Circle : Shape
    {
        public override void calc()
        {
            Console.Write("Enter Radius of Circle: "); float
            r = float.Parse(Console.ReadLine());
            Console.WriteLine("Area of Circle is: " + Math.Round(Math.PI*r*r),2); }

    }
    class Ellipse : Shape
    {
        public override void calc()
        {
            Console.Write("Enter Minor Radii of Ellipse: ");
            float r1 = float.Parse(Console.ReadLine());
            Console.Write("Enter Major Radii of Ellipse: ");
            float r2 = float.Parse(Console.ReadLine());
            Console.WriteLine("Area of Ellipse is: " + Math.Round(Math.PI*r1*r2),2); }

    }
    class Triangle : Shape
    {
        public override void calc()
        {
            Console.Write("Enter Base of Triangle: "); float ba =
            float.Parse(Console.ReadLine()); Console.Write("Enter
            Height of Triangle: "); float he =
            float.Parse(Console.ReadLine()); Console.WriteLine("Area of
            Triangle is: " + ba*he*0.5); }

    }
    class Square : Shape
    {
```

```

public override void calc()
{
    Console.Write("Enter Side of Square: "); float side =
float.Parse(Console.ReadLine()); Console.WriteLine("Area
of Square is: " + side*side); }

}

class ok
{ static void Main(string[] args)
{
    Rectangle r = new Rectangle();
    r.calc();

    Circle c = new Circle();
    c.calc();

    Ellipse e = new Ellipse();
    e.calc();

    Square s = new Square();
    s.calc();

    Triangle t = new Triangle();
    t.calc(); Console.ReadKey();
}
}

```

OUTPUT:

```

file:///c:/users/admin/documents/visual studio 2010/Projects/ConsoleApplication1/ConsoleApplication1/bin/Debug/ConsoleApplication1.exe

Enter Length of Rectangle: 4
Enter Breadth of Rectangle: 5
Area of Rectangle is: 20
Enter Radius of Circle: 9
Area of Circle is: 254
Enter Minor Radii of Ellipse: 4
Enter Minor Radii of Ellipse: 2
Area of Ellipse is: 25
Enter Side of Square: 6
Area of Square is: 36
Enter Base of Triangle: 12
Enter Height of Triangle: 3
Area of Triangle is: 18

```

Practical : 4

AIM: Design online registration form using Master Page and Validations.

A) Design online registration form for the participation of technical events (use HTML Controls, validation controls) and display all the data on the other page. Also design a layout using Master Page.

CODE:

1 MasterPage.master:

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

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<asp:ContentPlaceHolder id="head" runat="server">
</asp:ContentPlaceHolder>
<style type="text/css">
.style1 {
width: 100%;
}
.style2 {
height: 23px;
}
.style3 {
height: 467px;
}
.style4 {
width: 100%;
height: 100%;
}
.style5 {
width: 120px; }
</style>
</head>
<body>
<form id="form1" runat="server">
<div>

<table class="style1">
<tr>
```

```

<td bgcolor="#3333FF" class="style2" style="text-align: center">
<asp:Label ID="Label2" runat="server" Font-Bold="True" Font-Size="30pt"
ForeColor="#DEFC83" Text="TECHNOFEST 2021"></asp:Label>
</td>
</tr>
<tr>
<td class="style3">
<table class="style4">
<tr>
<td align="justify" class="style5"
style="text-align: center; vertical-align: top" bgcolor="Orange">
<br />
<asp:Label ID="Label1" runat="server" BackColor="Yellow" BorderColor="Black"
BorderStyle="Dashed" Text="E V E N T S" Width="120px" Font-Bold="True"
Font-Italic="True" Font-Names="Bodoni MT Condensed" Font-Size="25pt"
ForeColor="#CC0000" Height="38px" style="margin-top: 0px"></asp:Label>
<br />
<br />
<asp:Label ID="Button2" runat="server" Text="HACKATHON" Width="120px"
BackColor="Yellow" BorderStyle="Solid" BorderWidth="3px" Font-Bold="True"
ForeColor="#CC0000" Height="30px" Enabled="False" />
<br />
<br />
<asp:Label ID="Button3" runat="server" Text="LAN GAMING" Width="120px"
BackColor="Yellow" BorderStyle="Solid" BorderWidth="3px" Font-Bold="True"
ForeColor="#CC0000" Height="30px" Enabled="False" />
<br />
<br />
<asp:Label ID="Button4" runat="server" Text="WEBINAR" Width="120px"
BackColor="Yellow" BorderStyle="Solid" BorderWidth="3px" Font-Bold="True"
ForeColor="#CC0000" Height="30px" Enabled="False" />
<br />
<br />
<asp:Label ID="Button5" runat="server" Text="ROBO WARS" Width="120px"
BackColor="Yellow" BorderStyle="Solid" BorderWidth="3px" Font-Bold="True"
ForeColor="#CC0000" Height="30px" Enabled="False" />
</td>
<td style="vertical-align: top">
<asp:ContentPlaceHolder ID="mainstuff" runat="server">

STUFFS AND ALLS

</asp:ContentPlaceHolder>

<asp:ContentPlaceHolder ID="prints" runat="server"> STUFFS

AND ALLS

</asp:ContentPlaceHolder>

</td>
</tr>

```

```

</table>
</td>
</tr>
<tr>
<td class="style2" bgcolor="#3333FF" style="text-align: center">
<asp:Label ID="Label3" runat="server" Font-Bold="True" Font-Size="30pt"
ForeColor="#DEFC83" Text="COME INCREASE THE KNOWLEDGE AND ENJOY!!!"></asp:Label>
</td>
</tr>
</table>

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

```

2. First.aspx:

```

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

<asp:Content ID="Content1" runat="server" contentplaceholderid="mainstuff">

<p style="height: 60px" align="center">
    <asp:Label ID="Label4" runat="server" Font-Bold="True" Font-Italic="False"
    Font-Size="25pt" Font-Underline="False" ForeColor="Orange"
    Text="REGISTRATION FORM"></asp:Label><br />
    <asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Italic="False"
    Font-Size="15pt" Font-Underline="False" ForeColor="Orange"
    Text="PARTICIPATE AND CONQUER"></asp:Label>
    <br />
</p>
<p style="height: 23px" align="center">
    <asp:Label ID="Label5" runat="server" Text="Name: " ForeColor="Orange"
></asp:Label>
    &nbsp;
    <asp:TextBox ID="name" runat="server" ></asp:TextBox>
    &nbsp;&nbsp;
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ControlToValidate="name" ErrorMessage="*" Font-Bold="True"
        ForeColor="Red"></asp:RequiredFieldValidator> </p>
<p style="height: 25px" align="center">
    <asp:Label ID="Label6" runat="server" Text="Roll No: " ForeColor="Orange"
></asp:Label>
    &nbsp;
    <asp:TextBox ID="roll" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
        ControlToValidate="roll" ErrorMessage="*"
        ForeColor="Red"></asp:RequiredFieldValidator>

```

```

    &nbsp;<asp:RangeValidator ID="RangeValidator2" runat="server"
        ControlToValidate="roll" ErrorMessage="*Please Enter Valid Roll No"
        Font-Bold="True" ForeColor="Red" MaximumValue="120" MinimumValue="1"
        Type="Integer"></asp:RangeValidator>
    &nbsp;&nbsp;
    </p>
    <p style="height: 25px" align="center">
        <asp:Label ID="Label8" runat="server" ForeColor="Orange" Text="Age:></asp:Label>
    &nbsp;<asp:TextBox ID="age" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server"
            ControlToValidate="age" ErrorMessage="">
            <br/>
            <asp:RangeValidator ID="RangeValidator1" runat="server"
                ControlToValidate="age" ErrorMessage="*Age Not In Range" Font-Bold="True"
                ForeColor="Red" MaximumValue="50" MinimumValue="17"
                Type="Integer"></asp:RangeValidator>
        </p>
        <p style="height: 25px" align="center">
            <asp:Label ID="Label9" runat="server" ForeColor="Orange" Text="E-Mail:></asp:Label>
            <asp:TextBox ID="email" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
                ControlToValidate="email" ErrorMessage="">
                <br/>
                <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
                    runat="server" ControlToValidate="email"
                    ErrorMessage="*Please Enter Valid E-Mail" Font-Bold="True" ForeColor="Red"
                    ValidationExpression="\w+([-.\'])\w+@\w+([-.\'])\w+\.\w+([-.\'])\w+*></asp:RegularExpressionValidator>
            </p>
            <p style="height: 25px" align="center">
                <asp:Label ID="Label7" runat="server" ForeColor="Orange" Text="Department:></asp:Label>
                <asp:DropDownList ID="dep" runat="server">
                    <asp:ListItem>Engineering</asp:ListItem>
                    <asp:ListItem>MCA</asp:ListItem>
                    <asp:ListItem>INXT</asp:ListItem>
                </asp:DropDownList>
            </p>
            <p style="height: 22px" align="center">
                <asp:Label ID="Label10" runat="server" ForeColor="Orange" Text="Select Events: " Font-Size="15pt"></asp:Label>
                <asp:CheckBoxList ID="eve" runat="server" style="margin-left:800px">
                    <asp:ListItem>Hackathon</asp:ListItem>
                    <asp:ListItem>Lan Gaming</asp:ListItem>
                    <asp:ListItem>Webinar</asp:ListItem>
                    <asp:ListItem>Robo Wars</asp:ListItem>
                </asp:CheckBoxList>
                <br />
            </p>

```

```

<p style="height: 25px" align="center">
    <asp:Button ID="Button2" runat="server" Text="SUBMIT"
        Width="120px" BackColor="Yellow" BorderStyle="Solid" Font-Bold="True"
        ForeColor="Orange" Height="30px" PostBackUrl="~/Print.aspx"
        onclick="Button2_Click" />
</p>

</asp:Content>

<asp:Content ID="Content2" runat="server" contentplaceholderid="prints">
</asp:Content>

```

3. print.aspx:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Print.aspx.cs" Inherits="Print" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="mainstuff" Runat="Server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="prints" Runat="Server">
    <p align="center">
        <asp:Label ID="Label4" runat="server" Font-Bold="True" Font-Italic="False" Font-
            Size="25pt" Font-Underline="False" ForeColor="Orange"
            Text="Registered Successfully!!!"/><br />
        <asp:Label ID="Label15" runat="server" Font-Bold="True" Font-Italic="False" Font-
            Size="15pt" Font-Underline="False" ForeColor="Orange"
            Text="Please Validate Information"/></asp:Label>
        <br />
    </p>
    <p align="center">
        <asp:Label ID="Label5" runat="server" Text="Name: " ForeColor="Orange"
            Font-Bold="True" Font-Size="15pt"/></asp:Label>
        <asp:Label ID="Label10" runat="server" Font-Bold="True" Font-Size="15pt"
            Text="Label"/></asp:Label>
    </p>
    <p align="center">
        <asp:Label ID="Label6" runat="server" Text="Roll No: " ForeColor="Orange"
            Font-Bold="True" Font-Size="15pt"/></asp:Label>
        <asp:Label ID="Label11" runat="server" Font-Bold="True" Font-Size="15pt"
            Text="Label"/></asp:Label>
    </p>
    <p align="center">
        <asp:Label ID="Label8" runat="server" ForeColor="Orange" Text="Age: "

```

```

        Font-Bold="True" Font-Size="15pt">></asp:Label>
<asp:Label ID="Label12" runat="server" Font-Bold="True" Font-Size="15pt"
    Text="Label"></asp:Label>
</p>
<p align="center">
    <asp:Label ID="Label16" runat="server" ForeColor="Orange" Text="E-Mail: "
        Font-Bold="True" Font-Size="15pt"></asp:Label>
<asp:Label ID="Label17" runat="server" Font-Bold="True" Font-Size="15pt"
    Text="Label"></asp:Label>
</p>
<p align="center">
    <asp:Label ID="Label7" runat="server" ForeColor="Orange"
        Text="Department: " Font-Bold="True" Font-Size="15pt"></asp:Label>
    <asp:Label ID="Label13" runat="server" Font-Bold="True" Font-Size="15pt"
        Text="Label"></asp:Label>
</p>
<p align="center">
    <asp:Label ID="Label18" runat="server" ForeColor="Orange"
        Text="Selected Events:" Font-Bold="True" Font-Size="15pt"></asp:Label><br />
    <asp:Label ID="Label19" runat="server" Font-Bold="True" Font-Size="15pt"
        Text="Label"></asp:Label>
</p>
<p align="center">
    &ampnbsp</p>
</asp:Content>
```

4. print.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 Print :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
{ if (PreviousPage != null)
{
    ContentPlaceHolder place =
(ContentPlaceHolder)PreviousPage.Master.FindControl("mainstuff");
    TextBox n = (TextBox)place.FindControl("name");
    TextBox r = (TextBox)place.FindControl("roll"); TextBox
    a = (TextBox)place.FindControl("age");
    TextBox email = (TextBox)place.FindControl("email");
    DropDownList d = (DropDownList)place.FindControl("dep");
    CheckBoxList c = (CheckBoxList)place.FindControl("eve");
    Label10.Text = n.Text;
    Label11.Text = r.Text;
    Label12.Text = a.Text;
    Label17.Text = email.Text;
    Label13.Text = d.SelectedItem.Text;
    string item = "";
    for (int i = 0; i < c.Items.Count; i++)
    { if (c.Items[i].Selected == true)
```

```

        { item += c.Items[i].Text + "<br>";
    }

    if (item != null)
    {
        Label19.Text = item;
    }

}

```

OUTPUT:

AFTER LOADING:

TECHNOFEST 2021

REGISTRATION FORM
PARTICIPATE AND CONQUER

Name:

Roll No:

Age:

E-Mail:

Department:

Select Events:

Hackathon
 Lan Gaming
 Webinar
 Robo Wars

COME INCREASE THE KNOWLEDGE AND ENJOY!!!

○ Required Field Validator:

REGISTRATION FORM

PARTICIPATE AND CONQUER

Name: *

Roll No: *

Age: *

E-Mail: *

Department:

Select Events:

Hackathon
 Lan Gaming
 Webinar
 Robo Wars

Other Validations:

REGISTRATION FORM

PARTICIPATE AND CONQUER

Name:

Roll No: *Please Enter Valid Roll No

Age: *Age Not In Range

E-Mail: *Please Enter Valid E-Mail

Department:

Select Events:

Hackathon
 Lan Gaming
 Webinar
 Robo Wars

Entering Correct Data:

REGISTRATION FORM

PARTICIPATE AND CONQUER

Name: Atharva Kale

Roll No: 27

Age: 21

E-Mail: abc@gmail.com

Department: MCA

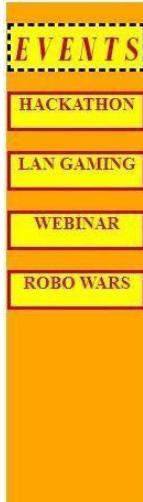
Select Events:

- Hackathon
- Lan Gaming
- Webinar
- Robo Wars

SUBMIT

Data Shown After Successful registration:

TECHNOFEST 2021



Registered Successfully!!!

Please Validate Information

Name: Atharva Kale

Roll No: 27

Age: 21

E-Mail: abc@gmail.com

Department: MCA

Selected Events:

Hackathon

Lan Gaming

Webinar

COME INCREASE THE KNOWLEDGE AND ENJOY!!!

Practical : 5

AIM: Design simple angular web application.

A) Build a simple angular web application.

CODE:

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

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

<html xmlns="http://www.w3.org/1999/xhtml">
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div ng-app="myapp" ng-controller="myctrl">

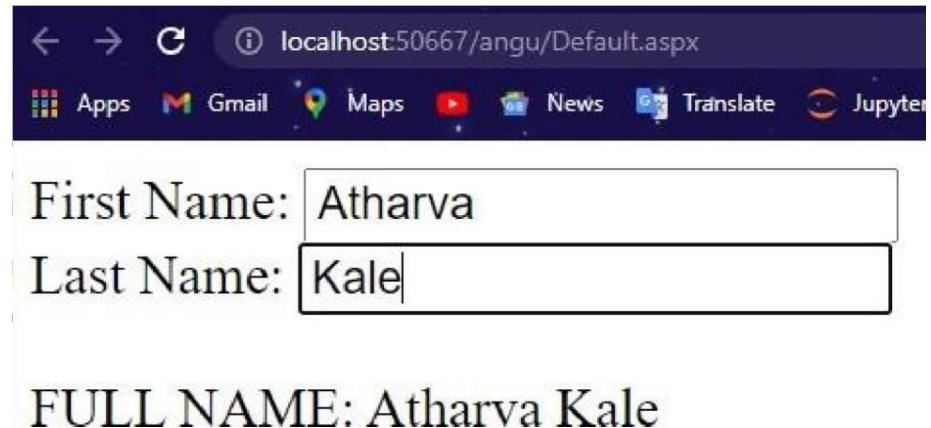
            First Name: <input type="text" ng-model="fname" /><br />
            Last Name: <input type="text" ng-model="lname" />

            <br />
            <br />
            FULL NAME: {{fname+" "+lname}}
        </div>

        <script type="text/javascript">    var app =
            angular.module('myapp',
                []);
            app.controller('myctrl', function ($scope) {

                $scope.fname = "";
                $scope.lname = "";
            });
        </script>
    </form>
</body>
</html>
```

OUTPUT:



Practical : 6

AIM: Design web applications to demonstrate connected and disconnected architecture in Ado.net.

- A) Design a webpage to demonstrate a connection oriented architecture. Fetch Student details from database such as Roll no, Name, Programme (e.g. MCA), Course (e.g. AWT) etc.

CODE:

1. Default.aspx:

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

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:GridView ID="GridView1" runat="server" Height="248px" Width="371px">
            </asp:GridView>

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

2. Default.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.SqlClient;
```

```

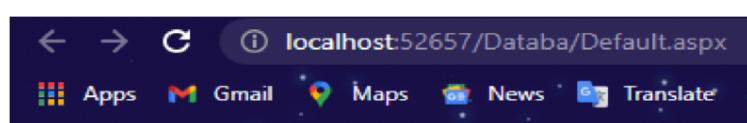
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender,
EventArgs e)
    {
        SqlConnection con = new SqlConnection("data
source=COOLBOY\\SQLEXPRESS;Database=college; integrated security=true;");
        SqlCommand query = new SqlCommand("select * from student",con); con.Open();
        SqlDataReader r = query.ExecuteReader();
        GridView1.DataSource = r;
        GridView1.DataBind(); con.Close();
    }
}

```

OUTPUT:

DATA STORED IN DATABASE AT SQL SERVER MANAGEMENT STUDIO

	name	roll_no	programme	course
	Atharva	27	MCA	AWT
	B	2	MCA	AIML
	C	3	ENG	C++
	D	4	INXT	Deep learning
	E	5	BSC	Database
▶*	NULL	NULL	NULL	NULL



name	roll_no	programme	course
Atharva	27	MCA	AWT
B	2	MCA	AIML
C	3	ENG	C++
D	4	INXT	Deep learning
E	5	BSC	Database

B) Design a webpage to demonstrate a disconnected architecture. Fetch Student details from database such as Roll no, Name, Programme (e.g. MCA), Course (e.g.AWT) etc.

CODE:

1. Default.aspx:

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

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server" Height="297px" Width="386px">
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

2. Default.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.SqlClient; using
System.Data;
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("data
source=COOLBOY\SQLEXPRESS;Database=college; integrated security=true;");
        SqlDataAdapter a = new SqlDataAdapter("select * from student", con);
        con.Open();
        DataSet ds = new DataSet();
        a.Fill(ds, "student");
```

```

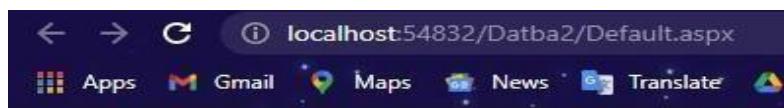
        GridView1.DataSource = ds.Tables[0];
        GridView1.DataBind(); con.Close();
    }
}

```

OUTPUT:

DATA STORED IN DATABASE AT SQL SERVER MANAGEMENT STUDIO

	name	roll_no	programme	course
	Atharva	27	MCA	AWT
	B	2	MCA	AIML
	C	3	ENG	C++
	D	4	INXT	Deep learning
	E	5	BSC	Database
»*	NULL	NULL	NULL	NULL



The screenshot shows a web browser window with the URL `localhost:54832/Datba2/Default.aspx`. The page displays a table with the same data as the SQL Server Management Studio table, showing student information: name, roll number, programme, and course.

name	roll_no	programme	course
Atharva	27	MCA	AWT
B	2	MCA	AIML
C	3	ENG	C++
D	4	INXT	Deep learning
E	5	BSC	Database

Practical : 7

AIM: Design web applications to demonstrate the use of data bound controls and procedures.

A) Create a webpage that demonstrates the use of data bound controls of ASP.NET.

CODE:

1. Default.aspx:

```
@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
<%
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
                DataSourceID="SqlDataSource1">
                <Columns>
                    <asp:BoundField DataField="name" HeaderText="name" SortExpression="name"
                />
                    <asp:BoundField DataField="roll_no" HeaderText="roll_no"
                        SortExpression="roll_no" />
                    <asp:BoundField DataField="programme" HeaderText="programme"
                        SortExpression="programme" />
                    <asp:BoundField DataField="course" HeaderText="course"
                        SortExpression="course" />
                </Columns>
            </asp:GridView>
            <br />
            <asp:SqlDataSource ID="SqlDataSource1" runat="server"
                ConnectionString="<%$ ConnectionStrings:collegeConnectionString %>"
```

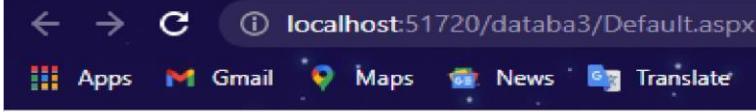
```

        SelectCommand="SELECT [name], [roll_no], [programme], [course] FROM
[student]">
    </asp:SqlDataSource>

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

```

OUTPUT:



A screenshot of a web browser window titled "localhost:51720/databa3/Default.aspx". The browser interface includes standard navigation buttons (back, forward, search), a tab bar with "Apps", "Gmail", "Maps", "News", and "Translate", and a status bar at the bottom.

name	roll_no	programme	course
Atharva	27	MCA	AWT
B	2	MCA	AIML
C	3	ENG	C++
D	4	INXT	Deep learning
E	5	BSC	Database
ASFG	45	MCA	AWT
Atharva	36	BSC	AI

B) Design a webpage to demonstrate the working of a simple stored procedure.

CODE:

1. Default.aspx:

```

@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default"
<%
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title></title>
</head>
<body>

```

```

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

    <asp:GridView ID="GridView1" runat="server">
    </asp:GridView>
    <br />
    <br />

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

```

2. Default.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.SqlClient; using
System.Data;
public partial class _Default :
System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection("data source=COOLBOY\\SQLEXPRESS;
Database=College; integrated security = true"); con.Open();
    SqlCommand cmd = new SqlCommand("enterdata", con); cmd.CommandType
= CommandType.StoredProcedure;

    SqlDataReader rdr = cmd.ExecuteReader();
    GridView1.DataSource = rdr;
    GridView1.DataBind();
}
}

```

3. Stored Procedure “Enterdata”:

```

CREATE PROCEDURE enterdata
AS
BEGIN
SELECT * FROM student
END;

```

OUTPUT:



name	roll_no	programme	course
Atharva	27	MCA	AWT
B	2	MCA	AIML
C	3	ENG	C++
D	4	INXT	Deep learning
E	5	BSC	Database
ASFG	45	MCA	AWT
Atharva	36	BSC	AI

- C) Design a webpage to demonstrate the working of parameterized stored procedure.

CODE:

1. Default.aspx:

```
@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits
      ="_Default"
<%
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:Label ID="Label2" runat="server" Text="Enter Name: "></asp:Label>
            &nbsp;&nbsp;

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

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

```

        <asp:Label ID="Label3" runat="server" Text="Enter Roll No: "></asp:Label>
&nbsp;&nbsp;
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
<br />
<br />
        <asp:Label ID="Label4" runat="server" Text="Enter Programme: "></asp:Label>
        <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
<br />
<br />
        <asp:Label ID="Label5" runat="server" Text="Enter Course: "></asp:Label>
        <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
<br />
<br />
<br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="SUBMIT" />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

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

```

2. Default.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.SqlClient; using
System.Data;

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

protected void Button1_Click(object sender, EventArgs e)
{
    SqlConnection con = new SqlConnection("data source=COOLBOY\\SQLEXPRESS;
Database=college; integrated security = true");
    SqlParameter rn, name, br, prog, cor;
    SqlCommand cmd = new SqlCommand("spd", con);

//Specifying that we are dealing with stored procedure cmd.CommandType
= CommandType.StoredProcedure;
//Creating parameter rn = new SqlParameter("@roll_no",
SqlDbType.Int); name = new SqlParameter("@name",
SqlDbType.VarChar); prog = new SqlParameter("@programme",
SqlDbType.VarChar); cor = new SqlParameter("@course",
SqlDbType.VarChar);

//Add parameters with sqlCommand object
cmd.Parameters.Add(rn);
cmd.Parameters.Add(name);
cmd.Parameters.Add(prog);
cmd.Parameters.Add(cor); //passing
}

```

```

parameter Values rn.Value =
Int32.Parse(TextBox2.Text); name.Value =
TextBox1.Text; prog.Value =
TextBox3.Text; cor.Value =
TextBox4.Text; con.Open();

try
{
cmd.ExecuteNonQuery();
Label1.Text = "success";
}

catch (Exception)
{
    Label1.Text = "fail";
}
}
}

```

3. Parameterized Stored Procedure “spd”:

```

CREATE PROCEDURE spd(
@name varchar(50),
@roll_no int,
@programme varchar(50),
@course varchar(50)
)
AS
BEGIN
INSERT INTO student(
[name],
[roll_no],
[programme],
[course]
)
VALUES (
@name,
@roll_no,
@programme,
@course
)
END

```

OUTPUT:

DATA SUCCESSFULLY INSERTED

A screenshot of a web browser window titled "localhost:52025/daba5/Default.aspx". The page contains four input fields:

- Enter Name:
- Enter Roll No:
- Enter Programme:
- Enter Course:

Below the inputs are two buttons: a "SUBMIT" button and a "success" button.

TABLE AFTER INSERTION OF DATA

	name	roll_no	programme	course
1	Atharva	27	MCA	AWT
2	B	2	MCA	AIML
3	C	3	ENG	C++
4	D	4	INXT	Deep learning
5	E	5	BSC	Database
6	ASFG	45	MCA	AWT
7	Atharva	36	BSC	AI
8	Elon	44	ENG	SPACE

Practical : 8

AIM: Design web applications to demonstrate the use of LINQ to SQL and ADO.NET Entity Framework.

A) Design a web page to display the employee information from table to grid control. Use LINQ to SQL.

CODE:

1. Default.aspx:

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:GridView ID="GridView1" runat="server" Height="224px" Width="285px">
                </asp:GridView>

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

2. Default.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 _Default : System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
{
```

```

        DataClassesDataContext dbContext = new DataClassesDataContext();
        //Retrieve data from StudentsInfo table
        GridView1.DataSource = (from a in dbContext.employees where a.EMP_CITY ==
"Kalyan" select a);

        GridView1.DataBind();
    }
}

```

3. Adding LINQ TO SQL class and drop table employee on it:



OUTPUT:

EMP_ID	EMP_NAME	EMP_CITY	EMP_SAL
1	Atharva	Kalyan	10000
3	AGA	Kalyan	15000
4	GASE	Kalyan	30000

B) Design a library system in ASP.NET and show all the book details in a Gridview dynamically using ADO.NET Entity Framework.

CODE:

1. Default.aspx:

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

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

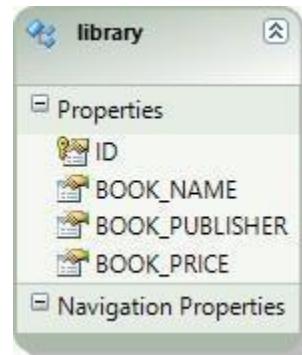
            <asp:GridView ID="GridView1" runat="server" Height="278px" Width="343px">
                </asp:GridView>

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

2. Default.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;  using System.Web.UI;
using System.Web.UI.WebControls;
using collegeModel;
public partial class _Default : System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
    { collegeEntities2 ctx = new collegeEntities2();
        GridView1.DataSource = (from s in ctx.libraries select s);
        // GridView1.DataSource = ctx.Students;
        GridView1.DataBind();
    }
}
```

3. Model.edmx:



OUTPUT:

← → ⌂ ⓘ localhost:50254/WebSite8/

ID	BOOK_NAME	BOOK_PUBLISHER	BOOK_PRICE
1	Beginning ASP.NET	John Wiley & Sons,	1000
2	Professional ASP.NET 3.5 SP1 Edition	John Wiley & Sons,	2000
3	Pro asp.netmvC 5 platform	Berkeley	3000
4	Professional ASP.NET MVC 5	Wrox Press	4000
5	ASP. Net 4.5 Unleashed	Pearson Education India	5000

Practical : 9

AIM: Design web applications to demonstrate Client and Server side State Management Techniques.

A) WAP to implement Client side state management techniques and Server side state management techniques on the form design attached with the assignment.

1) ViewState:

CODE:

○ Default.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 _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
{
    Label4.Visible = false;
}
protected void Button1_Click(object sender, EventArgs e)
{
    ViewState["name"] = TextBox1.Text;
    ViewState["age"] = TextBox2.Text;
    ViewState["branch"] = TextBox3.Text;
    TextBox1.Text = TextBox2.Text = TextBox3.Text = null; }

protected void Button2_Click(object sender, EventArgs e)
{ if (ViewState != null)
{
    Label4.Visible = true;
    Label4.Text = "AFTER RETRIEVE";
    TextBox1.Text = Convert.ToString(ViewState["name"]);
    TextBox2.Text = Convert.ToString(ViewState["age"]);
    TextBox3.Text = Convert.ToString(ViewState["branch"]); }
}
```

OUTPUT:

- **SUBMIT:**

A screenshot of a web browser window titled "localhost:50560/view/". The address bar also shows "localhost:50560/view/". Below the address bar is a navigation bar with icons for Apps, Gmail, Maps, YouTube, News, and others. The main content area contains three text input fields: "NAME: Atharva", "AGE: 36", and "BRANCH: BSC". Below these fields are two buttons: "SUBMIT" and "RETRIEVE".

- **RETRIEVE:**

A screenshot of a web browser window titled "localhost:50560/view/default.aspx". The address bar shows "localhost:50560/view/default.aspx". Below the address bar is a navigation bar with icons for Apps, Gmail, Maps, YouTube, News, and others. The main content area contains three text input fields: "NAME: Atharva", "AGE: 36", and "BRANCH: BSC". Below these fields are two buttons: "SUBMIT" and "RETRIEVE".

AFTER RETRIEVE

2) Hidden Field:

CODE:

- **Default.aspx:**

```
@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits  
=_Default"  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<%
%>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:Label ID="Label1" runat="server" Text="NAME: "></asp:Label>
            &nbsp;
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <asp:Label ID="Label2" runat="server" Text="AGE: "></asp:Label>
            &nbsp;
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            <asp:Label ID="Label3" runat="server" Text="BRANCH: "></asp:Label>
            &nbsp;
            <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
            <br />
            <br />
            <br />
            <br />
            <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="SUBMIT" />
            &nbsp;&nbsp;

            <asp:Button ID="Button2" runat="server" onclick="Button1_Click" Text="RETRIEVE"
               PostBackUrl="~/Default2.aspx" />

            <br />
            <br />
            <asp:HiddenField ID="HiddenField1" runat="server" />
            <asp:HiddenField ID="HiddenField2" runat="server" />
            <asp:HiddenField ID="HiddenField3" runat="server" />

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

```

Default.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 _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)

```

```
        {
            HiddenField1.Value = TextBox1.Text;
            HiddenField2.Value = TextBox2.Text;
            HiddenField3.Value = TextBox3.Text;
        }
    }
```

Default2.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 Default2 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string na=Request.Form["HiddenField1"];
        string age = Request.Form["HiddenField2"];
        string branch = Request.Form["HiddenField3"];
        Label1.Text += na;
        Label2.Text += age;
        Label3.Text += branch;
    }
}
```

- **CODE:**

- SUBMIT:**

A screenshot of a web browser window titled "localhost:62303/hidd/Default.aspx". The address bar shows the same URL. Below the title bar, there are navigation buttons (back, forward, search) and a toolbar with links to "Apps", "Gmail", "Maps", "News", and "Translate". The main content area contains three text input fields:

- NAME:
- AGE:
- BRANCH:

Below these fields are two buttons: "SUBMIT" and "RETRIEVE".

- **RETRIEVE:**

A screenshot of a web browser window titled "localhost:62303/hidd/Default2.aspx". The address bar shows the same URL. Below the title bar, there are navigation buttons (back, forward, search) and a toolbar with links to "Apps", "Gmail", "Maps", "News", and "Translate". The main content area displays the data submitted earlier:

Name: Atharva
Age: 36
Branch: BSC

3) Cookies:

- **Default.aspx.cs:**

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

- **OUTPUT:**

```
using System.Web.UI;      using
System.Web.UI.WebControls;
public partial class _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
{
}
protected void Button1_Click(object sender, EventArgs e)
{
    HttpCookie ck = new HttpCookie("userInfo");
    ck["Name"] = TextBox1.Text; ck["Age"] =
    TextBox2.Text; ck["Branch"] =
    TextBox3.Text; Response.Cookies.Add(ck);
    Response.Redirect("Default2.aspx");
}
}
```

- **Default2.aspx.cs:**

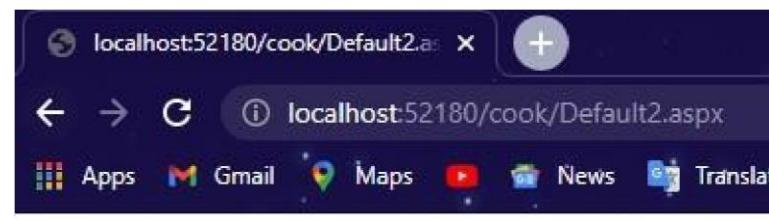
```
using System;      using
System.Collections.Generic;  using
System.Linq;  using System.Web;
using System.Web.UI;      using
System.Web.UI.WebControls;
public partial class Default2 :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
{
    HttpCookie cook = Request.Cookies["userInfo"];
    string name = cook["Name"].ToString(); string
    age = cook["Age"].ToString(); string branch =
    cook["Branch"].ToString();
    Label1.Text += name;
    Label2.Text += age;
    Label3.Text += branch;
} }
```

- **SUBMIT:**

- **CODE:**

The screenshot shows a web browser window with the URL `localhost:52180/cook/Default.aspx`. Below the address bar, there are navigation buttons (back, forward, search), a refresh button, and a plus sign for new tabs. The main content area contains a form with three text input fields. The first field is labeled "NAME" and contains the value "Atharva". The second field is labeled "AGE" and contains the value "36". The third field is labeled "BRANCH" and contains the value "BSC". Below the form is a "SUBMIT" button.

- **RETRIEVE:**



4) Query String:

- **Default.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 _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e) {
}
protected void Button1_Click(object sender, EventArgs e)
{
    Response.Redirect("Default2.aspx?Name=" + TextBox1.Text + "&Age=" + TextBox2.Text
}
```

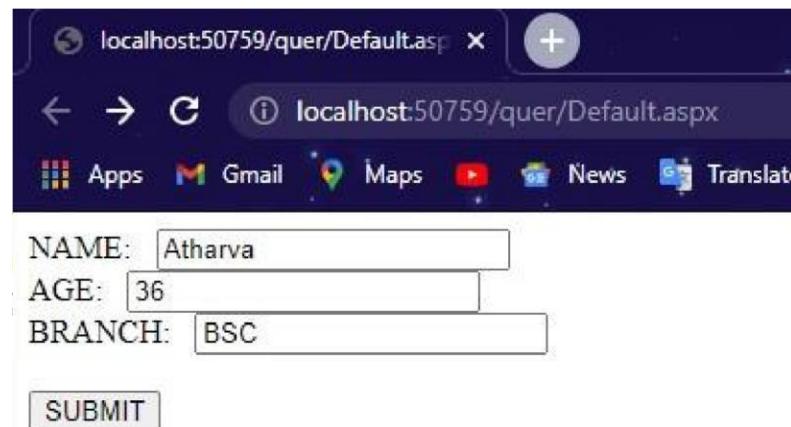
- **OUTPUT:**

```
+ "&Branch=" + TextBox3.Text); }  
}
```

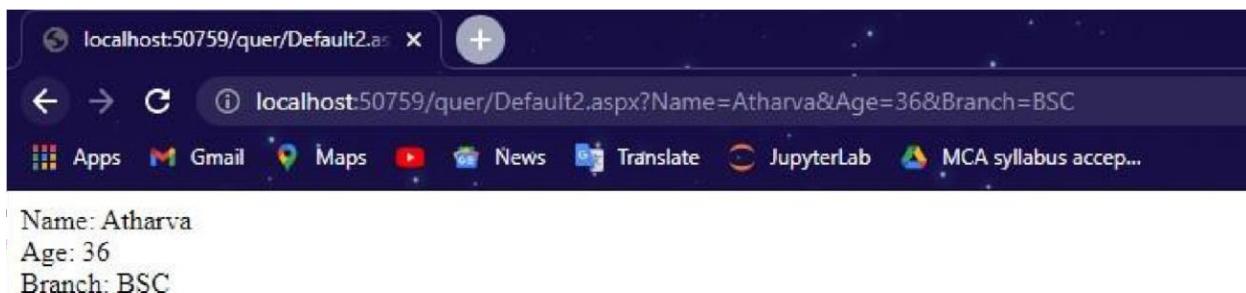
- **Default2.aspx.cs:**

```
using System;  
using System.Collections.Generic;  
using System.Linq; using  
System.Web; using System.Web.UI;  
using System.Web.UI.WebControls;  
public partial class Default2 :  
System.Web.UI.Page  
{ protected void Page_Load(object sender, EventArgs e)  
{ string name = Request.QueryString["Name"]; string  
age = Request.QueryString["Age"]; string branch  
= Request.QueryString["Branch"];  
Label1.Text += name;  
Label2.Text += age;  
Label3.Text += branch;  
} }
```

- **SUBMIT:**



- **RETRIEVE:**



- **CODE:**

5) Session State:

- **Default.aspx:**

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;  using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e) {
}
protected void Button1_Click(object sender, EventArgs e)
{
    Session["name"] = TextBox1.Text;
    Session["age"] = TextBox2.Text;
    Session["branch"] = TextBox3.Text;
    TextBox1.Text = TextBox2.Text = TextBox3.Text = null; }

protected void Button2_Click(object sender, EventArgs e)
{ if (ViewState != null)
{
    TextBox1.Text = Convert.ToString(Session["name"]);
    TextBox2.Text = Convert.ToString(Session["age"]);
    TextBox3.Text = Convert.ToString(Session["branch"]); }
}
```

- **OUTPUT:**

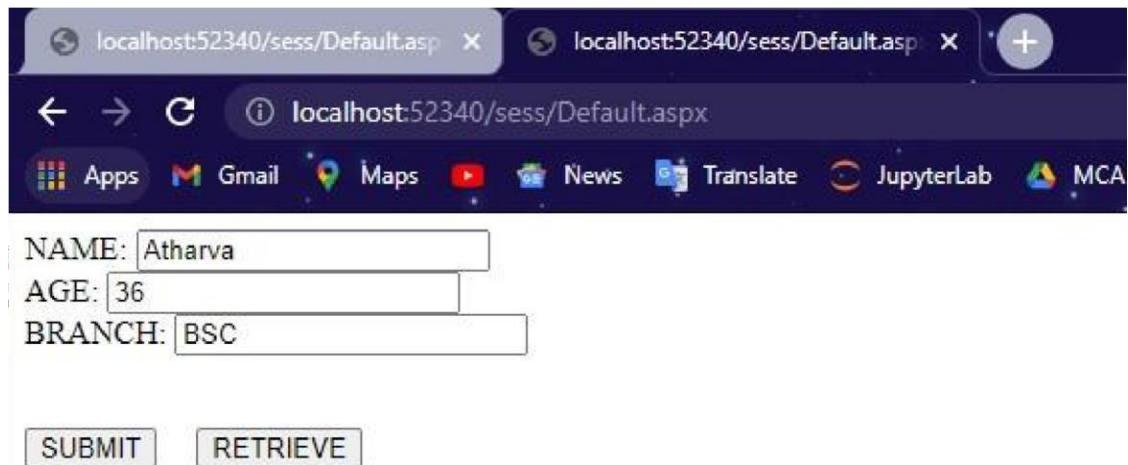
- **SUBMIT:**

The screenshot shows a web browser window with the URL `localhost:52340/sess/Default.aspx` in the address bar. Below the address bar is a standard navigation toolbar with icons for back, forward, search, and other links like Apps, Gmail, Maps, News, and Translate. The main content area contains a form with three text input fields labeled NAME, AGE, and BRANCH, each containing the values Atharva, 36, and BSC respectively. Below the form are two buttons: SUBMIT and RETRIEVE.

NAME:	Atharva
AGE:	36
BRANCH:	BSC

SUBMIT RETRIEVE

RETRIEVE ON ANOTHER TAB:



6) Application State:

CODE:

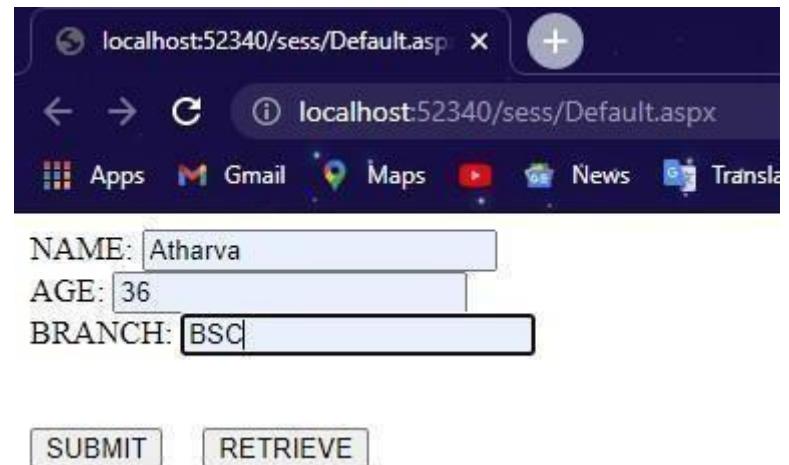
- Default.aspx:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;  using System.Web.UI;
using System.Web.UI.WebControls;
public partial class _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e) {
}
protected void Button1_Click(object sender, EventArgs e)
{
    Application["name"] = TextBox1.Text;
    Application["age"] = TextBox2.Text;
    Application["branch"] = TextBox3.Text;
    TextBox1.Text = TextBox2.Text = TextBox3.Text = null; }

protected void Button2_Click(object sender, EventArgs e)
{ if (ViewState != null)
{
    TextBox1.Text = Convert.ToString(Application["name"]);
    TextBox2.Text = Convert.ToString(Application["age"]);
    TextBox3.Text = Convert.ToString(Application["branch"]); }
}
```

- OUTPUT:

- SUBMIT:

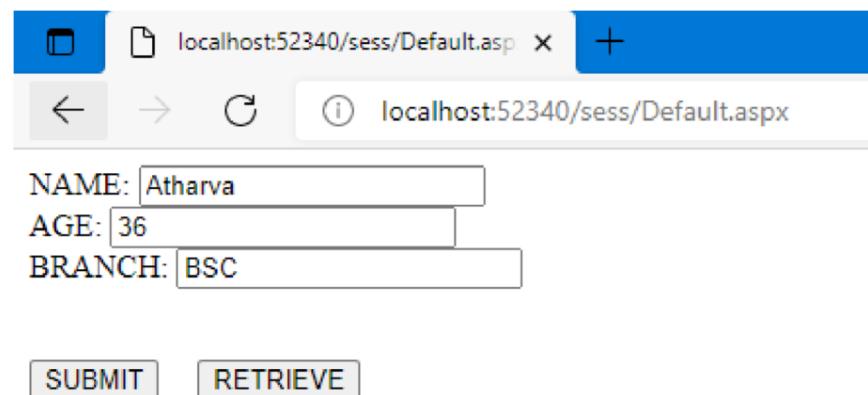


A screenshot of a Microsoft Edge browser window. The address bar shows the URL `localhost:52340/sess/Default.aspx`. Below the address bar, there are several browser navigation and utility icons. The main content area displays a form with three text input fields. The first field is labeled "NAME:" and contains the value "Atharva". The second field is labeled "AGE:" and contains the value "36". The third field is labeled "BRANCH:" and contains the value "BSC". Below the form are two buttons: "SUBMIT" on the left and "RETRIEVE" on the right.

NAME: Atharva
AGE: 36
BRANCH: BSC

SUBMIT RETRIEVE

- **RETRIEVE ON ANOTHER BROWSER:**



A screenshot of a Microsoft Edge browser window, similar to the one above. The address bar shows the URL `localhost:52340/sess/Default.aspx`. The main content area displays the same form with the same data: "NAME: Atharva", "AGE: 36", and "BRANCH: BSC". The "RETRIEVE" button is highlighted with a blue glow, indicating it is the active button.

NAME: Atharva
AGE: 36
BRANCH: BSC

SUBMIT RETRIEVE

Practical : 10

AIM: Design Web Application to produce and consume a web Service

A) Create an XML web service that returns all the student details from the student table.

CODE:

1. Default.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 Default2 :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e)
    { }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Visible = false;
        localhost2.WebService2 wb = new localhost2.WebService2();
        GridView1.DataSource = wb.takedata();
        GridView1.DataBind();
    }
}
```

2. Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:GridView ID="GridView1" runat="server">
            </asp:GridView>
            <br />
            <asp:Label ID="Label1" runat="server" Text="PRESS BUTTON TO GETDATA"></asp:Label>
            <br />
            <br />

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

```

<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />

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

```

3. WebService.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;           using
System.Web.Services;  using
System.Data;          using
System.Data.SqlClient;

/// <summary>
/// Summary description for WebService2
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// 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
WebService2 : System.Web.Services.WebService { public
WebService2 () {

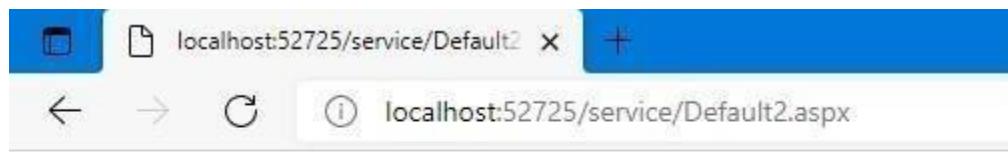
    //Uncomment the following line if using designed components
    //InitializeComponent();
}

[WebMethod]
public string HelloWorld() {
    return "Hello World";
}

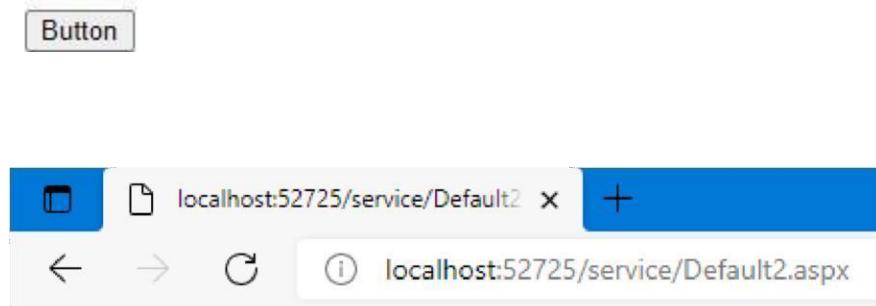
[WebMethod] public
DataTable takedata()
{
    SqlConnection         con         =         new         SqlConnection("data
source=COOLBOY\\SQLEXPRESS;Database=college; integrated security=true;");
    SqlCommand cmd = new SqlCommand("SELECT * FROM student");
    SqlDataAdapter sda = new SqlDataAdapter();
    cmd.Connection = con; sda.SelectCommand =
    cmd; DataTable dt = new DataTable();
    dt.TableName = "student"; sda.Fill(dt);
    return dt;
}
}

```

OUTPUT :



PRESS BUTTON TO GET DATA



name	roll_no	programme	course
B	2	MCA	AIML
C	3	ENG	C++
D	4	INXT	Deep learning
E	5	BSC	Database
Atharva	27	MCA	AWT
Atharva	36	BSC	AI
Elon	44	ENG	SPACE
ASFG	45	MCA	AWT

Button

- B)** Create an XML web service that is used to insert a row with the values given by the user.

1. Default.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 _Default :
System.Web.UI.Page
{ protected void Page_Load(object sender, EventArgs e) {
}
protected void Button1_Click(object sender, EventArgs e)
{
try
{
string name, programme,
course;
int roll_no;
localhost.add a = new localhost.add(); name =
TextBox1.Text; roll_no =
Int32.Parse(TextBox2.Text); programme =
TextBox3.Text;
course = TextBox4.Text;

a.data_add(name, roll_no, programme, course);

Label5.Text = "DATA ADDED SUCCESSFULLY!";
}
catch (Exception)
{
Label5.Text = "OOPS! SOME ISSUE!!";
}
}
}
```

2. Default.aspx:

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title> </head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Enter Name: "></asp:Label>
            &nbsp;&nbsp;&nbsp;
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Label ID="Label2" runat="server" Text="Enter Roll No: "></asp:Label>
            &nbsp;&nbsp;
            <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Label ID="Label3" runat="server" Text="Enter Programme: "></asp:Label>
            &nbsp;&nbsp;
```

```

<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
<br />
<br />
<asp:Label ID="Label4" runat="server" Text="Enter Course: "></asp:Label>
 &nbsp;&nbsp;
<asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
<br />
<br />
<br />
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="SUBMIT" />
<br />
<br />
<br />
<asp:Label ID="Label5" runat="server"></asp:Label>

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

```

3. add.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;           using
System.Web.Services; using
System.Data.SqlClient;

/// <summary>
/// Summary description for add
/// </summary>
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsProfiles.BasicProfile1_1)]
// 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 add : System.Web.Services.WebService {
    public add () {
        //Uncomment the
        following line
        if using
        designed
        components
            //InitializeComponent();
    }

    [WebMethod]
    public void data_add(string name, int roll, string programme, string course)
    {
        SqlConnection con = new SqlConnection("data
source=COOLBOY\\SQLEXPRESS;Database=college; integrated security=true;");
        SqlCommand cmd = new SqlCommand("INSERT INTO student VALUES
(@name,@roll,@course)",con);
        cmd.Parameters.AddWithValue("@name",name);
        cmd.Parameters.AddWithValue("@roll",roll);
        cmd.Parameters.AddWithValue("@programme",programme);
    }
}

```

```
        cmd.Parameters.AddWithValue("@course", course);
        con.Open(); cmd.ExecuteNonQuery(); con.Close();
    }

}
```

- **OUTPUT:**

The screenshot shows a web browser window with the URL `localhost:56787/service2/Default.aspx`. The page contains four input fields: "Enter Name:" with value "Stephan Strange", "Enter Roll No." with value "1", "Enter Programme:" with value "MCA", and "Enter Course:" with value "Quantum Computing". Below the form is a "SUBMIT" button.

DATA ADDED SUCCESSFULLY!

	name	roll_no	programme	course
1	Stephan Strange	1	MCA	Quantum Computing
2	B	2	MCA	AIML
3	C	3	ENG	C++
4	D	4	INXT	Deep learning
5	E	5	BSC	Database
6	Atharva	27	MCA	AWT
7	Atharva	36	BSC	AI
8	Elon	44	ENG	SPACE
9	ASFG	45	MCA	AWT

Practical : 11

AIM: Design Web Application to produce and Consume a web Service.

A) Create a WCF web service "Calculator" with methods as Addition, Subtraction, Multiplication & Division.

CODE:

1. IMathService.cs:

```
using System;
using System.Collections.Generic; using
System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;   using
System.Text;
namespace MathService
{
    // NOTE: You can use the "Rename" command on the "Refactor" menu to change the
    interface name "IMathService" in both code and config file together.
    [ServiceContract]
    public interface IMathService
    {
        [OperationContract]           int
        Addition(int num1, int num2);

        [OperationContract]
        int Subtraction(int num1, int num2);

        [OperationContract]
        int Multiplication(int num1, int num2);

        [OperationContract]
        int Division(int num1, int num2);
    }
}
```

2. MathService.cs:

```
using      System;      using
System.Collections.Generic;  using
System.Linq;      using
System.Runtime.Serialization;  using
System.ServiceModel;      using
System.Text;
namespace MathService
{
```

```

// NOTE: You can use the "Rename" command on the "Refactor" menu to change the class
name "MathService" in both code and config file together.
public class MathService : IMathService
{ public int Addition(int num1, int num2)
    { return num1 + num2;
    }
    public int Subtraction(int num1, int num2)
    { return num1 - num2;
    }
    public int Multiplication(int num1, int num2)
    { return num1 * num2;
    }
    public int Division(int num1, int num2)
    { return num1 / num2;
    }

}
}

```

3. App.config:

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <system.serviceModel>
        <behaviors>
            <serviceBehaviors>
                <behavior name="mexBehaviour">
                    <serviceMetadata httpGetEnabled="true" />
                </behavior>
            </serviceBehaviors>
        </behaviors>
        <services>
            <service name="MathService.MathService" behaviorConfiguration="mexBehaviour">
                <endpoint address="MathService" binding="basicHttpBinding"
contract="MathService.IMathService">
                    </endpoint>
                <endpoint address="MathService" binding="netTcpBinding"
contract="MathService.IMathService">
                    </endpoint>
                <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange" />
                <host>
                    <baseAddresses>
                        <add baseAddress="http://localhost:8080/" />
                        <add baseAddress="net.tcp://localhost:8090" />
                    </baseAddresses>
                </host>
            </service>
        </services>
    </system.serviceModel>
</configuration>

```

4. Program.cs:

```

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

```

```

System.ServiceModel;      namespace
MathService_Host
{ class Program
    { static void Main(string[] args)
        {
            ServiceHost host = new ServiceHost(typeof(MathService.MathService));
            host.Open();
            Console.WriteLine("Service Hosted Sucessfully"); Console.Read();
        }
    }
}

```

5. 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.Windows.Forms;
namespace MachService_Client
{ public partial class Form1 : Form
    { public Form1()
        {
            InitializeComponent(); label1.Visible
            = false;
        }

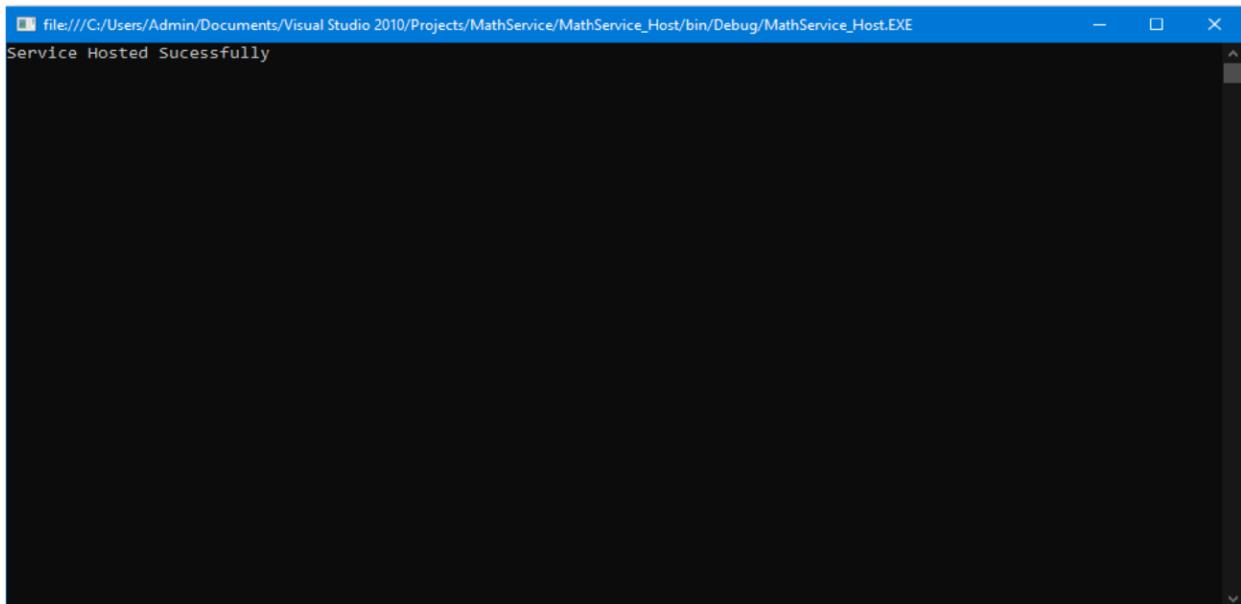
        private void button1_Click(object sender, EventArgs e)
        {
            ServiceReference1.MathServiceClient _obj = new
ServiceReference1.MathServiceClient("NetTcpBinding_IMathService");
            string val = comboBox1.Text;
            label1.Visible = true; label1.Text =
                "Value is: "; if(val=="ADD")
            {
                label1.Text += _obj.Addition(Convert.ToInt32(textBox1.Text),
Convert.ToInt32(textBox2.Text)).ToString();
            }
            else if(val=="SUB")
            {
                label1.Text += _obj.Subtraction(Convert.ToInt32(textBox1.Text),
Convert.ToInt32(textBox2.Text)).ToString();
            }
            else if(val=="MUL")
            {
                label1.Text += _obj.Multiplication(Convert.ToInt32(textBox1.Text),
Convert.ToInt32(textBox2.Text)).ToString();
            }
            else if(val=="DIV")
            {
                label1.Text += _obj.Division(Convert.ToInt32(textBox1.Text),
Convert.ToInt32(textBox2.Text)).ToString();
            }
        }
    }
}

```

```
}
```

OUTPUT:

SERVICE HOSTED SUCCESSFULLY:



○ **ADDITION:**

A screenshot of a Windows application window titled "Form1". The window contains four text input fields and one dropdown menu:

- FIRST NUMBER:
- SECOND NUMBER:
- SELECT OPERATION:
- SUBMIT

Below the window, the text "Value is: 30" is displayed.

SUBTRACTION:

The screenshot shows a Windows application window titled "Form1". Inside the window, there are three text input fields labeled "FIRST NUMBER:" (containing "10"), "SECOND NUMBER:" (containing "20"), and "SELECT OPERATION:" (containing "SUB"). Below these fields is a blue "SUBMIT" button. At the bottom of the window, the text "Value is: -10" is displayed.

- **MULTIPLICATION:**

The screenshot shows a Windows application window titled "Form1". Inside the window, there are three text input fields labeled "FIRST NUMBER:" (containing "10"), "SECOND NUMBER:" (containing "20"), and "SELECT OPERATION:" (containing "MUL"). Below these fields is a blue "SUBMIT" button. At the bottom of the window, the text "Value is: 200" is displayed.

DIVISION:

The screenshot shows a Windows application window titled "Form1". Inside the window, there are three text input fields labeled "FIRST NUMBER:" and "SECOND NUMBER:", both containing the value "20". Below these is a dropdown menu labeled "SELECT OPERATION:" with "DIV" selected. At the bottom center is a blue "SUBMIT" button. Below the button, the text "Value is: 1" is displayed.

FIRST NUMBER: 20

SECOND NUMBER: 20

SELECT OPERATION: DIV

SUBMIT

Value is: 1

Practical : 12

AIM: Design Web Applications to demonstrate Model View Controller.

A) Create a MVC application to demonstrate the use of Httpget and Httppost.

CODE:

1. Employee.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web;
namespace EMPDEMO.Models
{ public class Employee
    { public string Name{get; set;} public
        string Address { get; set; } public
        int Age { get; set; }

    }
}
```

2. HomeController.cs:

```
using EMPDEMO.Models; using
System;
using System.Collections.Generic; using
System.Linq;
using System.Web; using
System.Web.Mvc;
namespace EMPDEMO.Controllers
{ public class HomeController : Controller
    {
        [HttpGet] //  

        GET: Home
        public ActionResult Index()
        {
            Employee emp = new Employee()
            {
                Address = "KALYAN",
                Name = "ATHARVA",
                Age = 21
            };

            return View(emp);
        }
}
```

```

        [HttpPost]
    public ActionResult Index(Employee emp)
    { return View("DisplayData", emp);
    }

}}

```

3. RouteConfig.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;      using
System.Web; using System.Web.Mvc;
using System.Web.Routing;
namespace EMPDEMO
{ public class RouteConfig
    { public static void RegisterRoutes(RouteCollection routes)
        { routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.MapRoute( name:
                    "Default",
                    url: "{controller}/{action}/{id}",
                    defaults: new { controller = "Home", action = "Index", id =
UrlParameter.Optional }
                );
        }
    }
}

```

4. Index.cshtml:

```

@model EMPDEMO.Models.Employee

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <div>
        @using (Html.BeginForm("MyForm"))
        {
            @Html.TextBoxFor(x => x.Name)<br /><br /><br />
            @Html.TextBoxFor(x => x.Address)<br /><br /><br />
        }
    </div>

```

```

@Html.TextBoxFor(x => x.Age)<br /><br /><br />

        <input type="submit" value="submit" />
    }
</div>
</body>
</html>

```

5. DisplayData.cshtml:

```

@model EMPDEMO.Models.Employee

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>DisplayData</title>
</head>
<body>
    <div>
        <h1> Employee Details: </h1>
        <p> Employee Name: @Model.Name</p>
        <p>Employee Address: @Model.Address</p>
        <p> Employee Age: @Model.Age</p>
    </div>
</body>
</html>

```

OUTPUT:

- **HttpGet:**

A screenshot of a Microsoft Edge browser window. The address bar shows the URL `https://localhost:44332`. The page contains three text input fields: the first field has the value "ATHARVA", the second field has the value "KALYAN", and the third field has the value "21". Below these fields is a blue "submit" button.

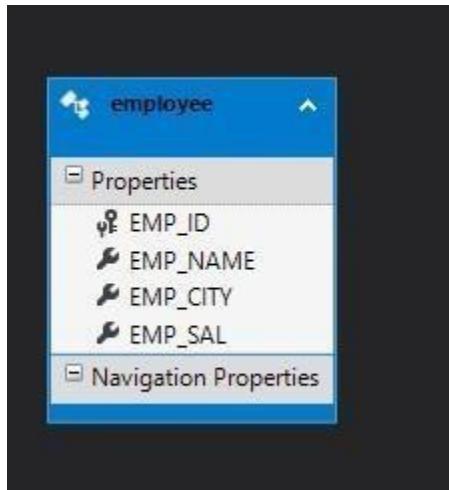
- **HttpPost:**

A screenshot of a Microsoft Edge browser window. The address bar shows the URL `https://localhost:44332/?Length=6`. The page displays the heading "Employee Details:" in large bold text. Below it, there are three lines of text: "Employee Name: ATHARVA", "Employee Address: KALYAN", and "Employee Age: 21".

B) Create a MVC application to Edit,Create, Delete and to display the data from the database using ADO.net Entity Model.

CODE:

1. Model.edmx:



2. employeeController.cs:

```
using System;
using System.Collections.Generic;
using System.Data;      using
System.Data.Entity;    using
System.Linq;    using System.Net;
using System.Web;     using
System.Web.Mvc;   using MVC_DEMO_2;
namespace MVC_DEMO_2.Controllers
{ public class employeesController : Controller
    { private collegeEntities db = new collegeEntities();

        // GET: employees public
        ActionResult Index()
        { return View(db.employees.ToList()); }

        // GET: employees/Details/5 public
        ActionResult Details(int? id)
        { if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            employee employee = db.employees.Find(id); if
            (employee == null)
            { return HttpNotFound();
            }
            return View(employee);
        }
    }
}
```

```

// GET: employees/Create public
ActionResult Create()
{
    return View();
}

// POST: employees/Create
// To protect from overposting attacks, enable the specific properties you want
to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "EMP_ID,EMP_NAME,EMP_CITY,EMP_SAL")]
employee employee)
{
    if (ModelState.IsValid)
    {
        db.employees.Add(employee);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(employee);
}

// GET: employees/Edit/5 public
ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    employee employee = db.employees.Find(id); if
(employee == null)
    { return HttpNotFound(); }
    return View(employee);
}

// POST: employees/Edit/5
// To protect from overposting attacks, enable the specific properties you want
to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "EMP_ID,EMP_NAME,EMP_CITY,EMP_SAL")]
employee employee)
{
    if (ModelState.IsValid)
    {
        db.Entry(employee).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(employee);
}

// GET: employees/Delete/5 public
ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
}

```

```

        employee employee = db.employees.Find(id); if
        (employee == null)
        { return HttpNotFound();
        }
        return View(employee);
    }

    // POST: employees/Delete/5
    [HttpPost, ActionName("Delete")] [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    { employee employee = db.employees.Find(id);
        db.employees.Remove(employee);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    { if (disposing)
        { db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

3. RouteConfig.cs:

```

using System; using
System.Collections.Generic; using
System.Linq; using System.Web;
using System.Web.Mvc; using
System.Web.Routing; namespace
MVC_DEMO_2
{ public class RouteConfig
    { public static void RegisterRoutes(RouteCollection routes)
        { routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
routes.MapRoute( name:
    "Default",
        url: "{controller}/{action}/{id}",
        defaults: new { controller = "employees", action = "Index", id =
UrlParameter.Optional }
    );
    }}}

```

4. Edit.cshtml:

```

@model MVC_DEMO_2.employee

@{
    ViewBag.Title = "Edit";
}

```

```

<h2>Edit</h2>

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

    <div class="form-horizontal">
        <h4>employee</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.EMP_ID)

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

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

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

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

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

```

```
<script src="~/Scripts/jquery-3.4.1.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>
```

5. Details.cshtml:

```
@model MVC_DEMO_2.employee

@{
    ViewBag.Title = "Details";
}



## Details



#### employee



---



<dt>
            @Html.DisplayNameFor(model => model.EMP_NAME) </dt>

        <dd>
            @Html.DisplayFor(model => model.EMP_NAME) </dd>

        <dt>
            @Html.DisplayNameFor(model => model.EMP_CITY)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.EMP_CITY)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.EMP_SAL)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.EMP_SAL)
        </dd>



@Html.ActionLink("Edit", "Edit", new { id = Model.EMP_ID }) |
        @Html.ActionLink("Back to List", "Index")


```

6. Delete.cshtml:

```
@model MVC_DEMO_2.employee

@{
    ViewBag.Title = "Delete";
}



## Delete



### Are you sure you want to delete this?



#### employee



---



@Html.DisplayNameFor(model => model.EMP_NAME)


@Html.DisplayFor(model => model.EMP_NAME)


@Html.DisplayNameFor(model => model.EMP_CITY)


@Html.DisplayFor(model => model.EMP_CITY)


@Html.DisplayNameFor(model => model.EMP_SAL)


@Html.DisplayFor(model => model.EMP_SAL) </dd>


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

    <div class="form-actions no-color">
        <input type="submit" value="Delete" class="btn btn-default" /> |
        @Html.ActionLink("Back to List", "Index")
    </div>
}
</div>
```

7. Create.cshtml:

```
@model MVC_DEMO_2.employee

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>

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

    <div class="form-horizontal">
        <h4>employee</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.EMP_ID, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EMP_ID, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EMP_ID, "", new { @class =
"text-danger" })
            </div>
        </div>

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

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

        <div class="form-group">
            @Html.LabelFor(model => model.EMP_SAL, htmlAttributes: new { @class =
"control-label col-md-2" })
```

```

        <div class="col-md-10">
            @Html.EditorFor(model => model.EMP_SAL, new { htmlAttributes = new {
                @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.EMP_SAL, "", new { @class =
                "text-danger" })
        </div>
    </div>

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

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

```

```

<script src="~/Scripts/jquery-3.4.1.min.js"></script>
<script src="~/Scripts/jquery.validate.min.js"></script>
<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

```

- **OUTPUT:**

- **Create New:**

Application name

Create

employee

EMP_ID
70

EMP_NAME
Dr.Strange

EMP_CITY
New York

EMP_SAL
60000

[Create](#)

[Back to List](#)

© 2021 - My ASP.NET Application

- **Display:**

Index

[Create New](#)

EMP_NAME	EMP_CITY	EMP_SAL	
Atharva	Dadar	10000	Edit Details Delete
DGX	Dadar	20000	Edit Details Delete
AGA	Kalyan	15000	Edit Details Delete
GASE	Kalyan	30000	Edit Details Delete
XYZ	Mumbai	50000	Edit Details Delete
UYI	Kurla	1000000	Edit Details Delete
Dr.Strange	New York	60000	Edit Details Delete

© 2021 - My ASP.NET Application

- **Edit:**

Index

[Create New](#)

EMP_NAME	EMP_CITY	EMP_SAL	
Atharva	Dadar	10000	Edit Details Delete
DGX	Dadar	20000	Edit Details Delete
AGA	Kalyan	15000	Edit Details Delete
GASE	Kalyan	30000	Edit Details Delete
XYZ	Mumbai	50000	Edit Details Delete
UYI	Kurla	1000000	Edit Details Delete
Dr.Strange	Kamartaj	60000	Edit Details Delete

© 2021 - My ASP.NET Application

□ Edit - My ASP.NET Application × +

← → ⌂ https://localhost:44341/employees/Edit/70

Application name

Edit

employee

EMP_NAME

EMP_CITY

EMP_SAL

[Back to List](#)

© 2021 - My ASP.NET Application

○ Delete:

□ Delete - My ASP.NET Application × +

← → ⌂ https://localhost:44341/employees/Delete/70

Application name

Delete

Are you sure you want to delete this?

employee

EMP_NAME
Dr.Strange

EMP_CITY
Kamartaj

EMP_SAL
60000

| [Back to List](#)

© 2021 - My ASP.NET Application



Application name

Index

[Create New](#)

EMP_NAME **EMP_CITY** **EMP_SAL**

Atharva	Dadar	10000	Edit Details Delete
DGX	Dadar	20000	Edit Details Delete
AGA	Kalyan	15000	Edit Details Delete
GASE	Kalyan	30000	Edit Details Delete
XYZ	Mumbai	50000	Edit Details Delete
UYI	Kurla	1000000	Edit Details Delete

© 2021 - My ASP.NET Application