PRACTICAL 1A : 28 August 2020

Aim : Create an application that obtains four int value from the user and display the product.

Solution :

using System;

namespace Sigma.Practical

{

class OneA

{

static void Main(string[] args)

{

int num1,num2,num3,num4,prod;

Console.WriteLine("Abhijeet Sharma");

Console.Write("Enter the Number 1 : ");

num1 = Int32.Parse(Console.ReadLine());

Console.Write("Enter the Number 2 : ");

num2 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the Number 3 : ");

num3 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter the Number 4 : ");

num4 = Convert.ToInt32(Console.ReadLine());

prod = num1 \* num2 \* num3 \* num4;

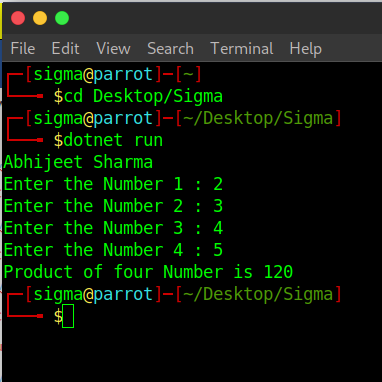
Console.WriteLine("Product of four Number is " + prod);

}

}

}

Output :



Practical 1B : 28 August 2020

Aim : Create an application to demonstrate string operations.

Solution :

using System;

namespace Sigma.Practical

{

class OneB

{

static void Main(string[] args)

{

string str1 = "", str2 = "DalmiaLionsCollege", new\_str = "";

int n = 0;

Console.WriteLine("Abhijeet Sharma");

Console.WriteLine("In Uppercase: "+str2.ToUpper());

Console.WriteLine("In Lowercase: " + str2.ToLower());

n = str1.CompareTo(str2);

if(n==0)

{

Console.WriteLine("Both is equal");

}

else

{

Console.WriteLine("Both is not equal");

}

new\_str = str2.Replace(" ", "");

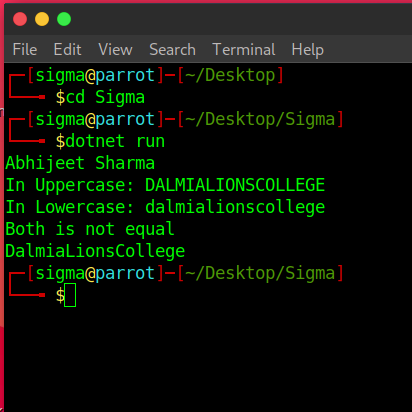
Console.WriteLine(new\_str);

}

}

}

Output :



Practical 1C : 28 August 2020

Aim : Create an application that receives the (Student Id, Name, Course, Date of Birth) information from a set of students. The application should also display the information of all student once the data entered.

Solution :

using System;

namespace Sigma.Practical

{

class OneC

{

struct Student

{

public string studid, name, cname;

public int day, month, year;

}

static void Main(string[] args)

{

Student[] s = new Student[2];

int i;

Console.WriteLine("Abhijeet Sharma");

for (i = 0; i < 2; i++)

{

Console.Write("Enter student Id : ");

s[i].studid = Console.ReadLine();

Console.Write("Enter student name : ");

s[i].name = Console.ReadLine();

Console.Write("Enter Course name : ");

s[i].cname = Console.ReadLine();

Console.Write("Enter date of birth \t Enter day(1=31) : ");

s[i].day = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter month(1-12) : ");

s[i].month = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter year : ");

s[i].year = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine("\n\nStudent's List\n");

for(i=0;i<2;i++)

{

Console.WriteLine("Student ID : "+ s[i].studid);

Console.WriteLine("Student Name : " + s[i].name);

Console.WriteLine("Course Name : " + s[i].cname);

Console.WriteLine("Date of birth(dd-mm-yy) : " + s[i].day+ "-" +s[i].month +"-"+ s[i].year);

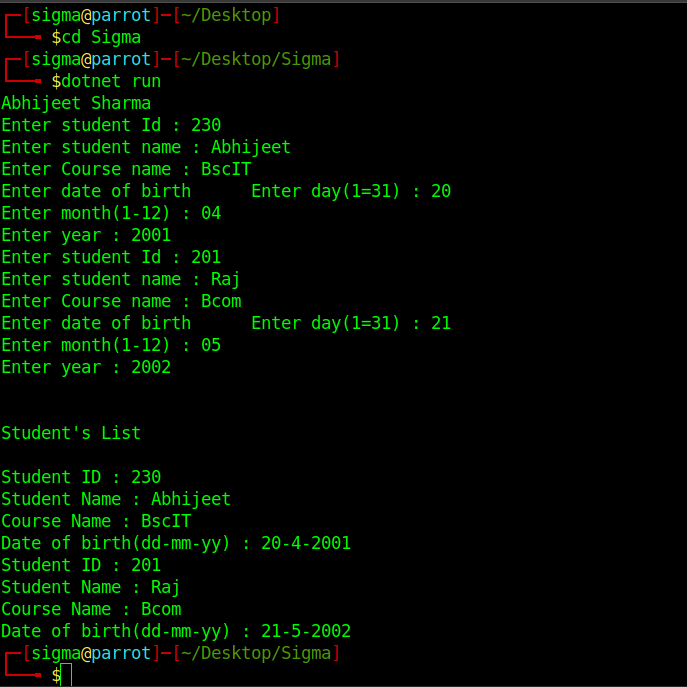
}

}

}

}

Output :



Practical 1D - 3 sept 2020

Aim 1 : Generate Fibonacci series.

Solution :

using System;

namespace Sigma

{

class Program

{

static void Main(string[] args)

{

int n1=0,n2=1,n3,i,number;

Console.Write("Enter the number of elements : ");

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

Console.Write(n1+" "+n2+" ");

for(i=2;i<number;++i)

{

n3=n1+n2;

Console.Write(n3+" ");

n1=n2;

n2=n3;

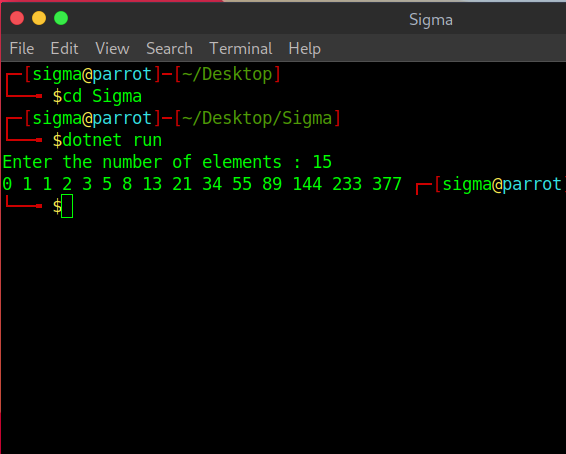
}

}

}

}

Output :



Aim 2 : prime numbers

Solution :

using System;

namespace Sigma

{

class Program

{

static void Main(string[] args)

{

int n, i, m=0, flag=0;

Console.Write("Enter the Number to check Prime: ");

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

m=n/2;

for(i = 2; i <= m; i++)

{

if(n % i == 0)

{

Console.Write("Number is not Prime.");

flag=1;

break;

}

}

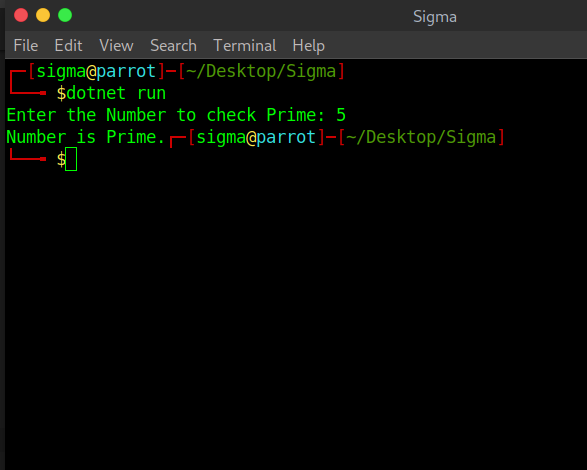
if (flag==0)

Console.Write("Number is Prime.");

}

}

}

Output : 

Aim 3 : test for vowels

Solution :

using System;

namespace Sigma

{

class Program

{

static void Main(string[] args)

{

char ch;

Console.WriteLine("Enter an alphabet");

ch = Convert.ToChar(Console.ReadLine());

switch(Char.ToLower(ch))

{

case 'a':

Console.WriteLine("Vowel");

break;

case 'e':

Console.WriteLine("Vowel");

break;

case 'i':

Console.WriteLine("Vowel");

break;

case 'o':

Console.WriteLine("Vowel");

break;

case 'u':

Console.WriteLine("Vowel");

break;

default:

Console.WriteLine("Not a vowel");

break;

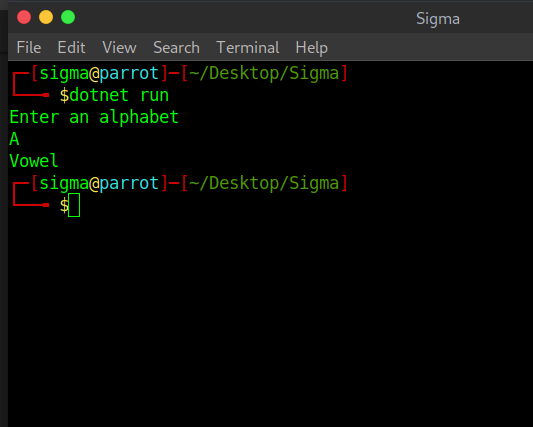
}

}

}

}

Output



Aim 4 : Foreach loop

Solution :

using System;

namespace Sigma

{

class Program

{

static void Main(string[] args)

{

char[] myArray = {'H','e','l','l','o'};

foreach(char ch in myArray)

{

Console.WriteLine(ch);

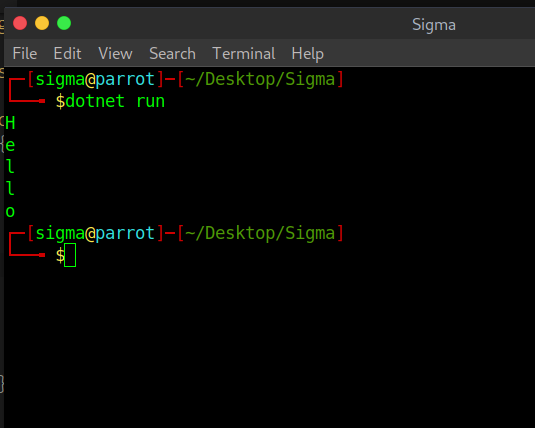
}

}

}

}

Output :



Aim 5 : reverse of a number and sum of digit

Solution :

using System;

namespace Sigma

{

class Program

{

static void Main(string[] args)

{

int num = 1234;

int rev\_num = 0,digits = 0;

while(num > 0)

{

rev\_num = rev\_num \* 10 + num % 10;

digits = digits + num % 10;

num = num / 10;

}

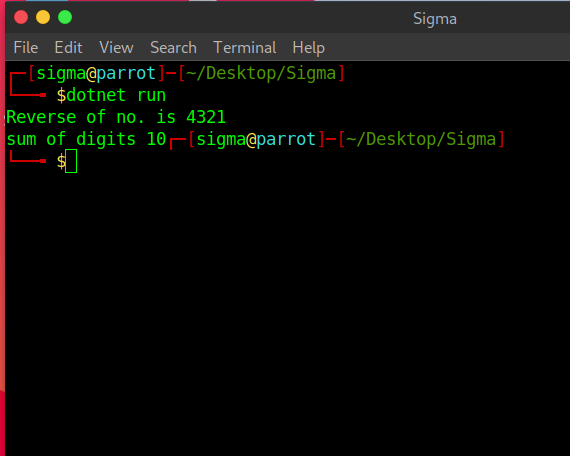
Console.WriteLine("Reverse of no. is "+ rev\_num);

Console.Write("sum of digits "+ digits);

}

}

}

Output : 

**Practical 2(a)**

Aim:- Create simple application to perform following operations.

(i) Finding Factorial Value

using System;

public class FactorialExample

{

public static void Main(string[] args)

{

int i,fact=1,number;

Console.Write("Abhijeet");

Console.WriteLine("Enter any Number: ");

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

for(i=1;i<=number;i++){

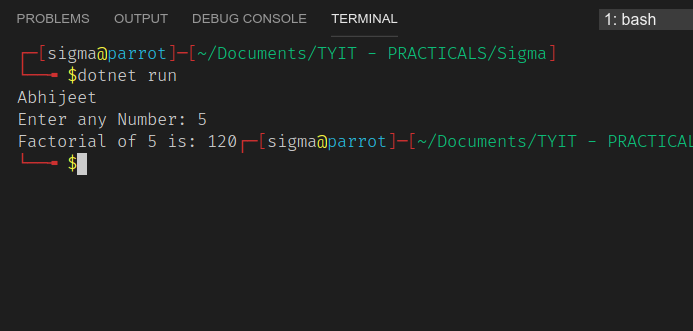
fact=fact\*i;

}

Console.Write("Factorial of " +number+" is: "+fact);

}

}



(ii) Money Conversion

using System;

namespace money

{

public class Program

{

public static void Main(string[] args)

{

Double usd, inr, val;

Console.WriteLine("Abhijeet");

Console.Write("Enter any Number: ");

usd= double.Parse(Console.ReadLine());

val = 69;

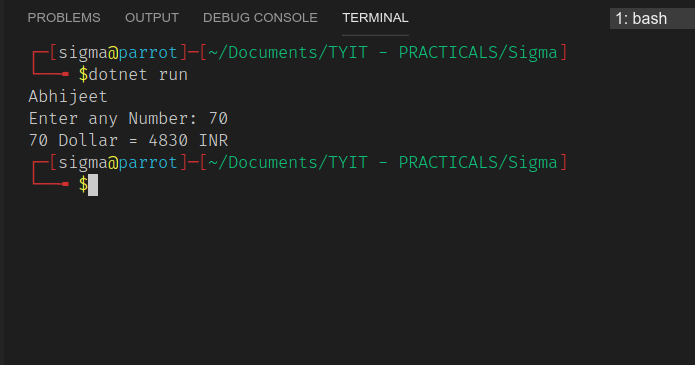
inr = usd \* val;

Console.WriteLine("{0} Dollar = {1} INR", usd, inr);

}

}

}



(iv) Temperature Conversion

using System;

namespace temperature {

class MyApplication

{

static void Main(string[] args)

{

double fahrenheit;

double celsius;

Console.WriteLine("Abhijeet");

Console.Write("Enter any Number: ");

celsius= double.Parse(Console.ReadLine());

fahrenheit = (celsius \* 9) / 5 + 32;

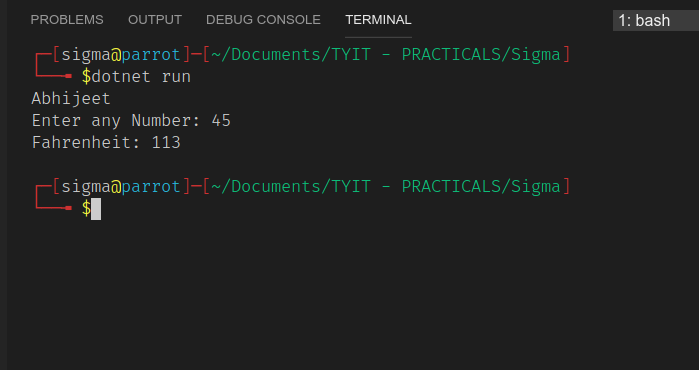
Console.WriteLine("Fahrenheit: " + fahrenheit);

Console.ReadLine();

}

}

}



**Practical 2(b)**

Aim:- Create simple application to demonstrate use of following concepts.

(i) Function Overloading

using System;

public class Function

{

public static int mulDisplay(int one, int two)

{

return one \* two;

}

public static int mulDisplay(int one, int two, int three)

{

return one \* two \* three;

}

}

public class Program

{

public static void Main()

{

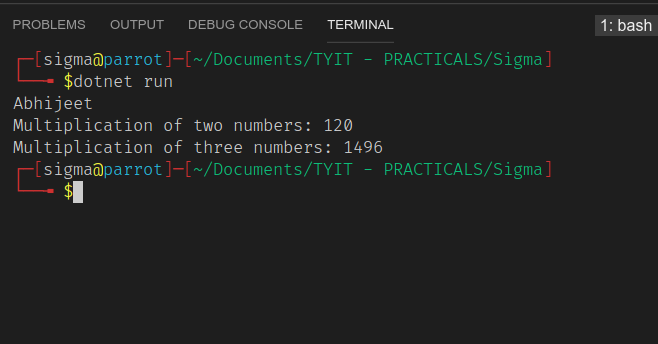
Console.WriteLine("Abhijeet");

Console.WriteLine("Multiplication of two numbers: "+Function.mulDisplay(10, 12));

Console.WriteLine("Multiplication of three numbers: "+Function.mulDisplay(11, 8, 17));

}

}



(ii) Inheritance (All Type)

* **Single inheritance**

using System;

namespace Application

{

class A

{

public void show()

{

Console.WriteLine("Hiii my name is Abhijeet");

}

}

class B : A

{

public void display()

{

Console.WriteLine("Hello");

}

}

class single

{

public static void Main()

{

B obj = new B();

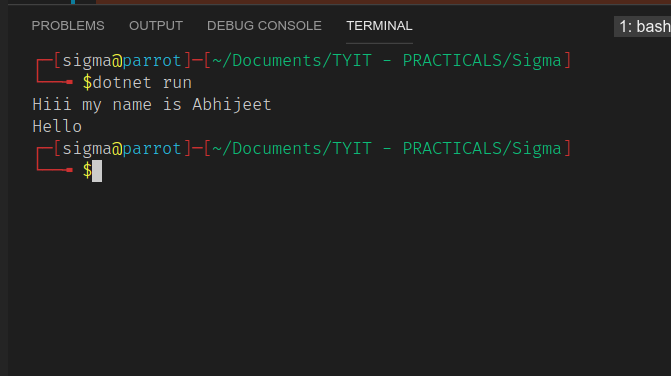
obj.show();

obj.display();

}

}

}



* **Hierarchical Inheritance**

using System;

namespace Application

{

class A

{

public void show()

{

Console.WriteLine("Hey my name is abhijeet");

}

}

class B : A

{

public void display()

{

Console.WriteLine("Hello");

}

}

class C : A

{

public void show1()

{

Console.WriteLine("What's your age ?");

}

}

class hierarchical

{

public static void Main()

{

B objl = new B();

C obj2 = new C();

objl.show();

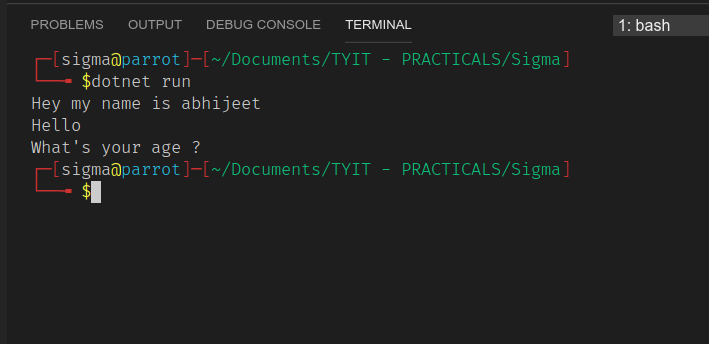
objl.display();

obj2.show1();

}

}

}



* **Multilevel inheritance**

using System;

public class RBI

{

public float rate\_rbi=6.5f;

public float calculateBalRBI(int amount,int years)

{

return

amount+((amount\*rate\_rbi\*years)/100);

}

}

public class SBI:RBI

{

public float rate\_sbi=7f;

public float calculateBalSBI(int amount,int years)

{

return

amount+((amount\*rate\_sbi\*years)/100);

}

}

public class Canera:SBI

{

public float rate\_canera=7.2f;

public float calculateBalCanera(int amount,int

years)

{

return

amount+((amount\*rate\_canera\*years)/100);

}

}

public class InheritanceExample

{

public static void Main (string[] args)

{

Canera c=new Canera();

Console.WriteLine("Abhijeet sharma");

Console.WriteLine("Enter the amount you want to

invest??");

int amount=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("for how many time you want to

invest (in years)?");

int years=Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Your amount according to

canera bank policies after "+years+" years will be

"+c.calculateBalCanera(amount,years));

Console.WriteLine("Your amount according to State

bank of India policies after "+years+" years will be

"+c.calculateBalSBI(amount,years));

Console.WriteLine("Your amount according to

Reserve bank of India policies after "+years+" years will be

"+c.calculateBalRBI(amount,years));

}

}

**(iii)** Constructor overloading

using System;

class ADD

{

int x, y;

double f;

string s;

public ADD(int a, double b)

{

x = a;

f = b;

}

public ADD(int a, string b)

{

y = a;

s = b;

}

public void show()

{

Console.WriteLine("Abhijeet");

Console.WriteLine("1st constructor (int + float): {0} ",(x + f));

}

public void show1()

{

Console.WriteLine("2nd constructor (int + string): {0}",(s + y));

}

}

class c

{

static void Main()

{

ADD g = new ADD(10, 20.2);

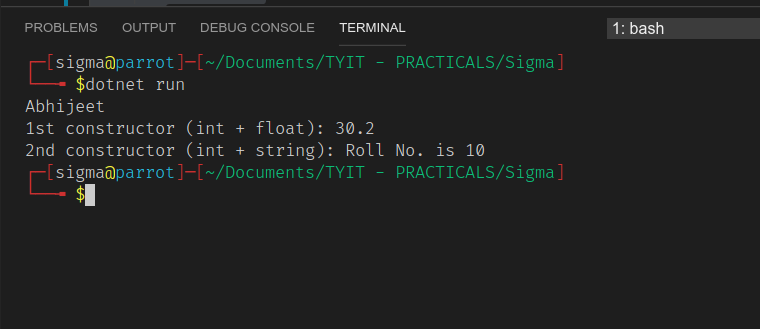
g.show();

ADD q = new ADD(10, "Roll No. is ");

q.show1();

}

}



**(iv)** Interface

using System;

public interface Drawable

{

void draw();

}

public class Rectangle : Drawable

{

public void draw()

{

Console.WriteLine("drawing rectangle...");

}

}

public class Circle : Drawable

{

public void draw()

{

Console.WriteLine("drawing circle...");

}

}

class Interface

{

public static void Main()

{

Console.WriteLine("abhijeet");

Drawable d;

d = new Rectangle();

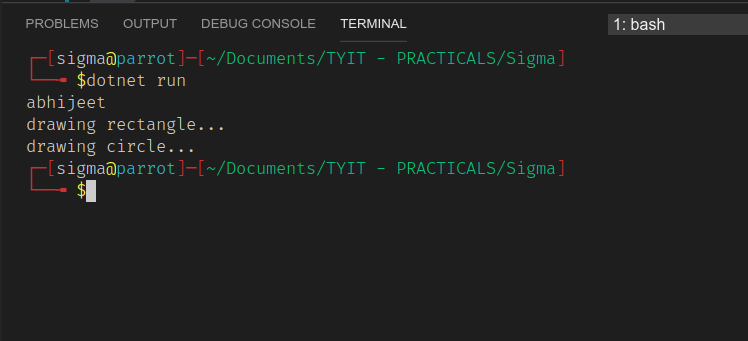
d.draw();

d = new Circle();

d.draw();

}

}



**Date:-09/10/2020**

**Pratical :-3A**

**Title:- create a simple web page with various server control to demonstrate setting and use of the properties**.

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO** | **CONTROL** | **PROPERTY** | **VALUE** |
| 1 | Label | ID | lblName |
| 2 | Label | Text | TextEnter Your Name |
| 3 | Textbox | ID | txtName |
| 4 | Label | ID | lblLocation |
| 5 | Label | Text | Location |
| 6 | Listbox | ID | lstLocation: |
| 7 | Listbox | ITEM LIST | Mumbai,Chennai,Delhi,Bangalore |
| 8 | Label | ID | lblGender |
| 9 | Label | Text | Gender |
| 10 | Radiobutton | ID | rdMale |
| 11 | Label | Text | Male |
| 12 | Radiobutton | ID | rdFemale |
| 13 | Label | Text | Female |
| 14 | Button | ID | btnSubmit |
| 15 | Label | Text | Submit |

NOTE:- DOUBLE click on submit button and the following code in the click Event protected void

bh8Button1\_click (object ,sender,EventArg e)

{

Response.Write(txtName.Text +"</br>"+ lstLocation.selectedItem.Text + " < /br>");

txtName.visible=false;

lstLocation.visible=false;

chkC.visible=false;

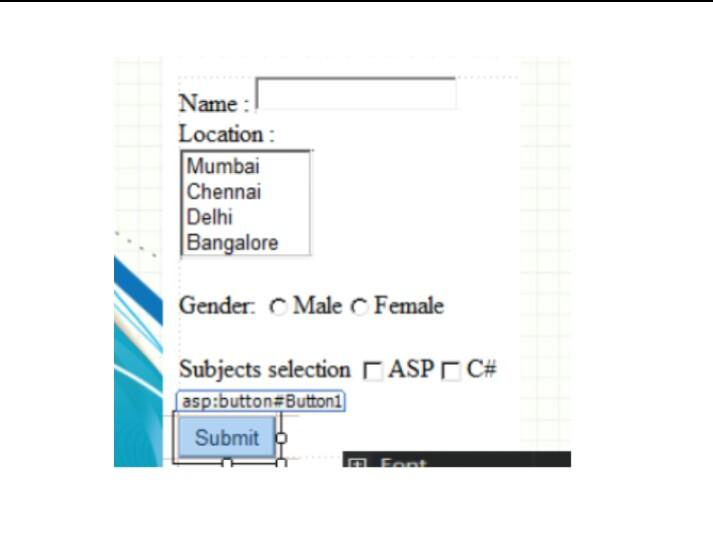
chkASP.visible=false;

rdmale.visible=false;

rdFemale.visible=false;

btnSubmit.visible=false;

}



Practical 3b

Date:-16/10/2020

Aim:- Demonstrate the use of Calendar control to perform following operations

1. Display message in a calendar control.
2. Display vacation in a calendar control.
3. Select a day in a calendar control using style.
4. Difference between two calendar dates.

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO** | **Control** | **Properties** | **Value** |
| 1 | Calendar | Id | Calendar1 |
| 2 |  | BackColor | #FFFFCC |
| 3 |  | DayNameFormat | Shortest |
| 4 |  | NextPrevFormat | ShortMonth |
| 5 |  | SelectedDayStyle-BackColor | Blue |
| 6 |  | TodayDayStyle-  ForeColor | White |
| 7 | Button | Id | btnResult |
| 8 |  | Text | Submit |
| 9 | Button | Id | btnReset |
| 10 |  | Text | Reset |
| 11 | Label | Id | Label1 |
| 12 | Label | Id | Label2 |
| 13 | Label | Id | Label3 |
| 14 | Label | Id | Label4 |
| 15 | Label | Id | Label5 |

Code:-

Calendar\_Control.aspx.cs

Protected void Calendar1\_DayRender(object sender, System.Web.UI.WebControls.DayRenderEventArgs e)

{

if(e.Day.Date.Day == 5 && e.Day.Date.Month == 9)

{

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

Label lbl = new Label();

lbl.Text = "<br>Teachers Day! ";

e.Cell.Controls.Add(lbl);

Image g1 = new image();

g1.ImageUrl = "Untitled.jpg";

g1.Height = 20;

g1.Width = 20;

e.Cell.Controls.Add(g1);

}

if(e.Day.Date.Day == 17 && e.Day.Date.Month == 10)

{

Calendar1.SelectedDate = new DateTime(2020, 10, 16);

Calendar1.SelectedDates.SelectRange(Calendar1.SelectedDate, Calendar1.SelectedDate.AddDays(9));

Label lbl1 = new Label();

lbl1.Text = "<br> Navratri! ";

e.Cell.Controls.Add(lbl1);

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

{

Label1.Text = "";

Label2.Text = "";

Label3.Text = "";

Label4.Text = "";

Label5.Text = "";

Calendar1.SelectedDates.Clear();

}

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

{

Calendar1.Caption = "TechnoLytics Learning ";

Label2.Text = "Today's Date:"+Calendar1.TodaysDate.ToShortDateString();

Label3.Text = " Navarro Festival Starts on :10-17-2020";

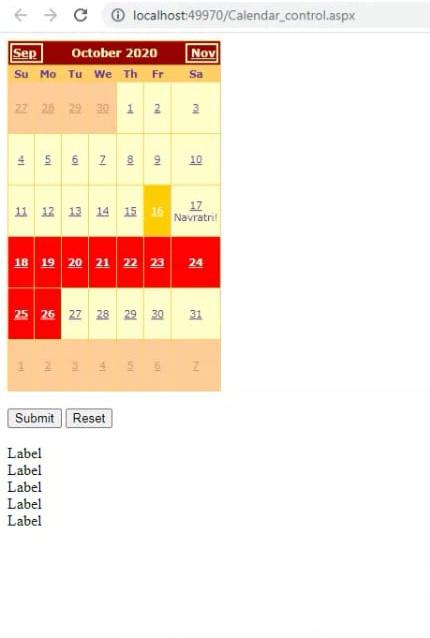
TimeSpan d = new DateTime(2020, 10, 17)-DateTime.Now;

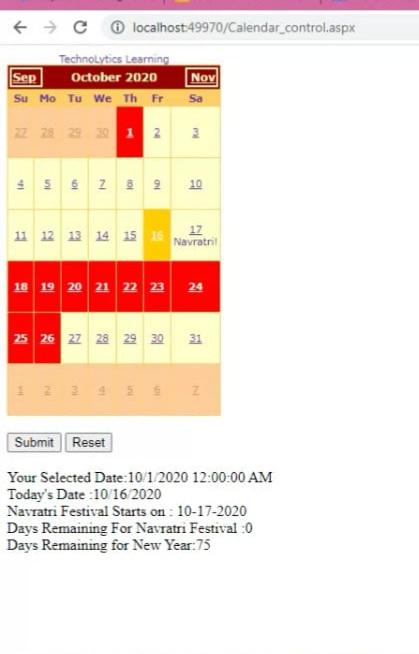
Label4.Text="Days Remaining For Navratri Festival :"+d.Days.ToString();

TimeSpan d1 = new DateTime(2020, 12, 31)-DateTime.Now;

Label5.Text = "Days Remaining for New Year:" + d1.Days.ToString();

}

****

****

****

DATE :- 23/10/2020

Name:- Abhijeet Sharma

ROLL no :- 330

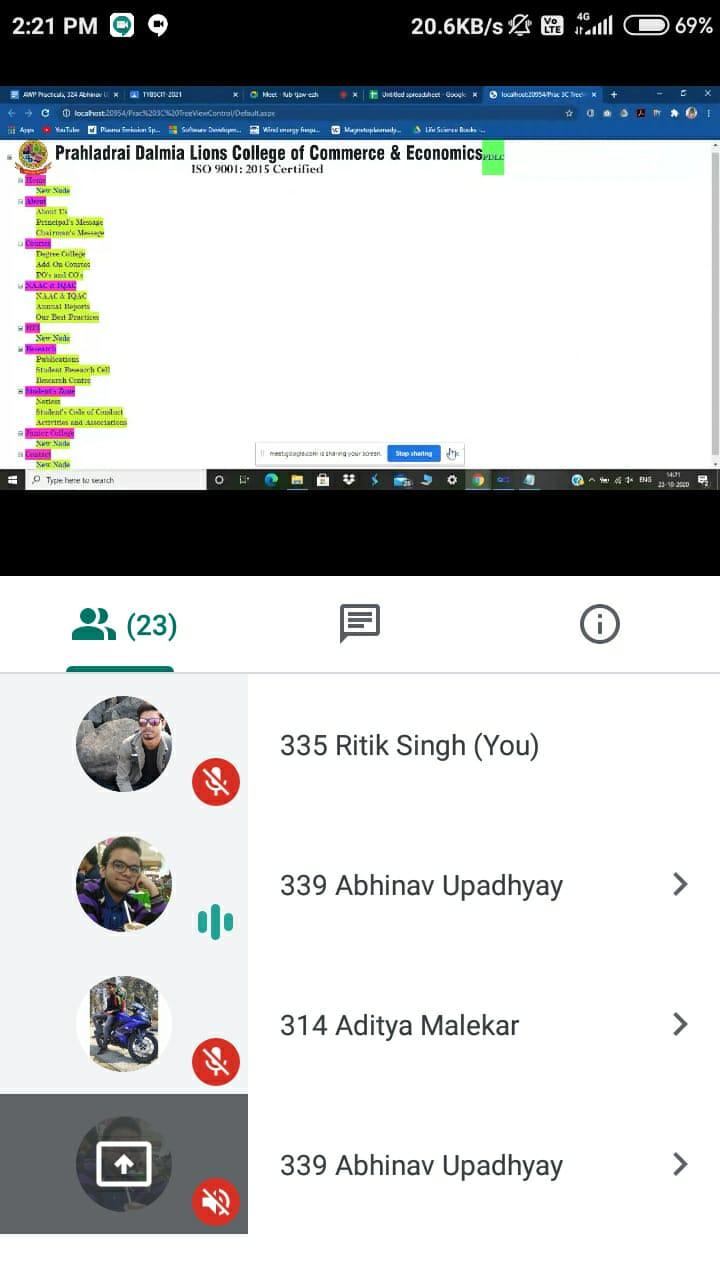
PRATICAL :- 3c

Aim :- create a Treeview control navigation website of pdlc maximum node should be not more than 3.also format TreeView control.

code:-

|  |  |  |  |
| --- | --- | --- | --- |
| sr.no | control | property | value |
| 1 | TreeView | ID | TreeView1 |
| 2 | TreeView | TreeNode Navigate url | https://www.dalmialionscollege.ac.in |
| 3 | TreeView | TreeNode Navigate url | URL Link |
| 4 | TreeView | TreeNode Navigate url | URL Link |
| 5 | TreeView | TreeNode Navigate url | URL Link |
| 6 | TreeView | TreeNode Navigate url | URL Link |
| 7 | TreeView | TreeNode Text | Home |
| 8 | TreeView | HoverNodeStyle | #FF523B |
| 9 | TreeView | LeafNodeStyle | #CCFF33 |
| 10 | TreeView | ParentNodeStyle | #FF33CC |
| 11 | TreeView | RootNodeStyle | #66FF66 |
| 12 | TreeView | TreeNode ImageUrl | URL Link |

output:



DATE :-24/10/2020

PRATICAL no.3c.2

Name :- Abhijeet

Rollno :- 330

Aim:- let's Bind ASP.net TreeView control to and xml file.

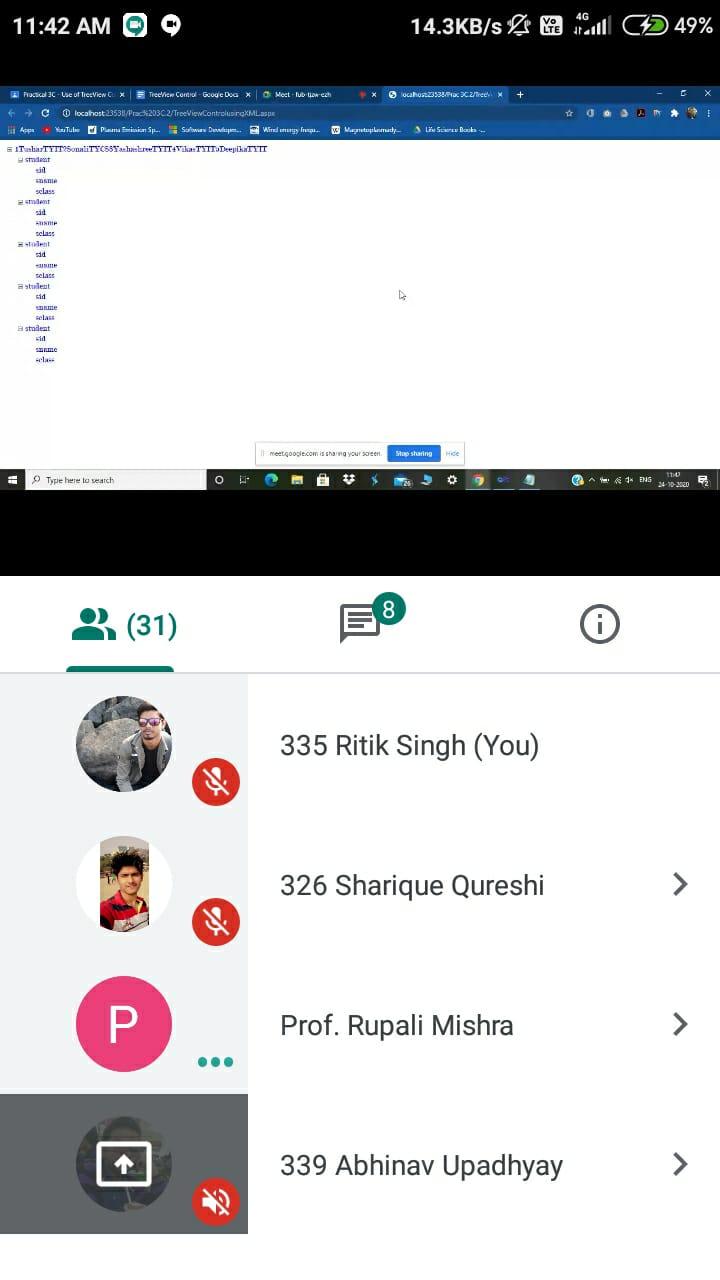
steps :-

1. Add an Xml file and name it XmlFile.xml
2. Drag and drop an Xml DataSource control on the webform. set DataFile attribute should point to the Xml File that we added in step 1
3. Drag and drop TreeView control and set DataSource control we create in step 2.also,set DataBinding.

**Properties** :-

|  |  |  |  |
| --- | --- | --- | --- |
| sr.no | control | property | value |
| 1 | XmlDataSource | ID | XmlDataSource1 |
| 2 | XmlDataSource | DataFile | ~/XmlFile.xml |
| 3 | TreeView | ID | TreeView1 |
| 4 | TreeView | DataSourceID | XmlDataSource1 |
| 5 | TreeNodeBinding | DataMember | TreeViewItem |
| 6 | TreeNodeBinding | NavigateUrl Field | Navigator1 |

**Output** :-



## XMLFile.xmls

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

<studentdetail>

<student>

<sid>1</sid>

<sname>siddharth</sname>

<sclass>TYIT</sclass>

</student>

<student>

<sid>2</sid>

<sname>Adarsh</sname>

<sclass>TYBCOM</sclass>

</student>

<student>

<sid>3</sid>

<sname>Ranjeet </sname>

<sclass>TYBAF</sclass>

</student>

<student>

<sid>4</sid>

<sname>Yash</sname>

<sclass>TYBsc</sclass>

</student>

<student>

<sid>4</sid>

<sname>Vikas</sname>

<sclass>TYINTERIOR</sclass>

</student>

<student>

<sid>5</sid>

<sname>Sonu</sname>

<sclass>MCOM</sclass>

</student>

</studentdetail>

Date:5/11/2020

Name: Abhijeet

Rollno:330

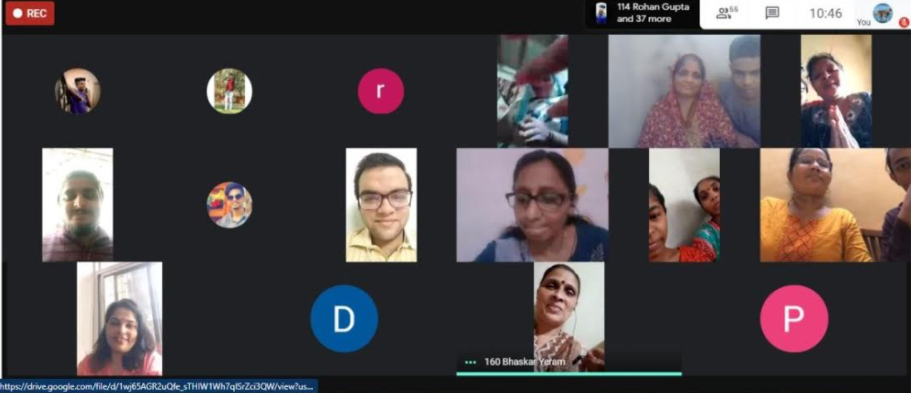
Practical 4A

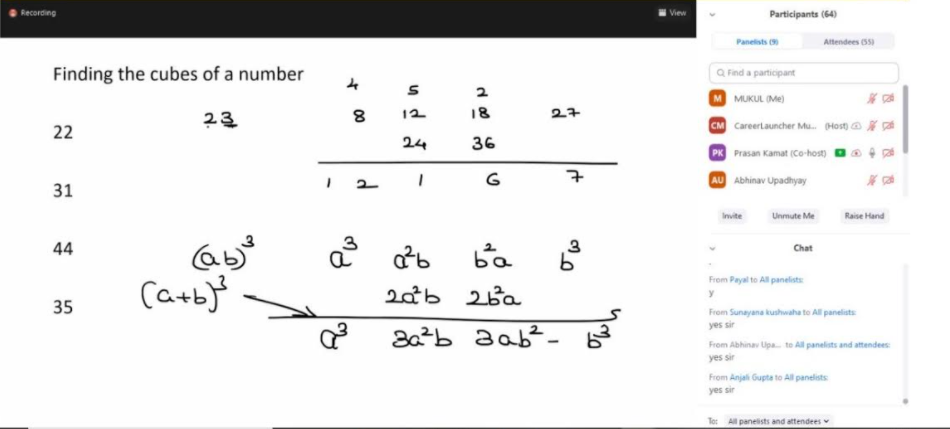
Aim:Create Web Form to demonstrate use of AdRotator Control.

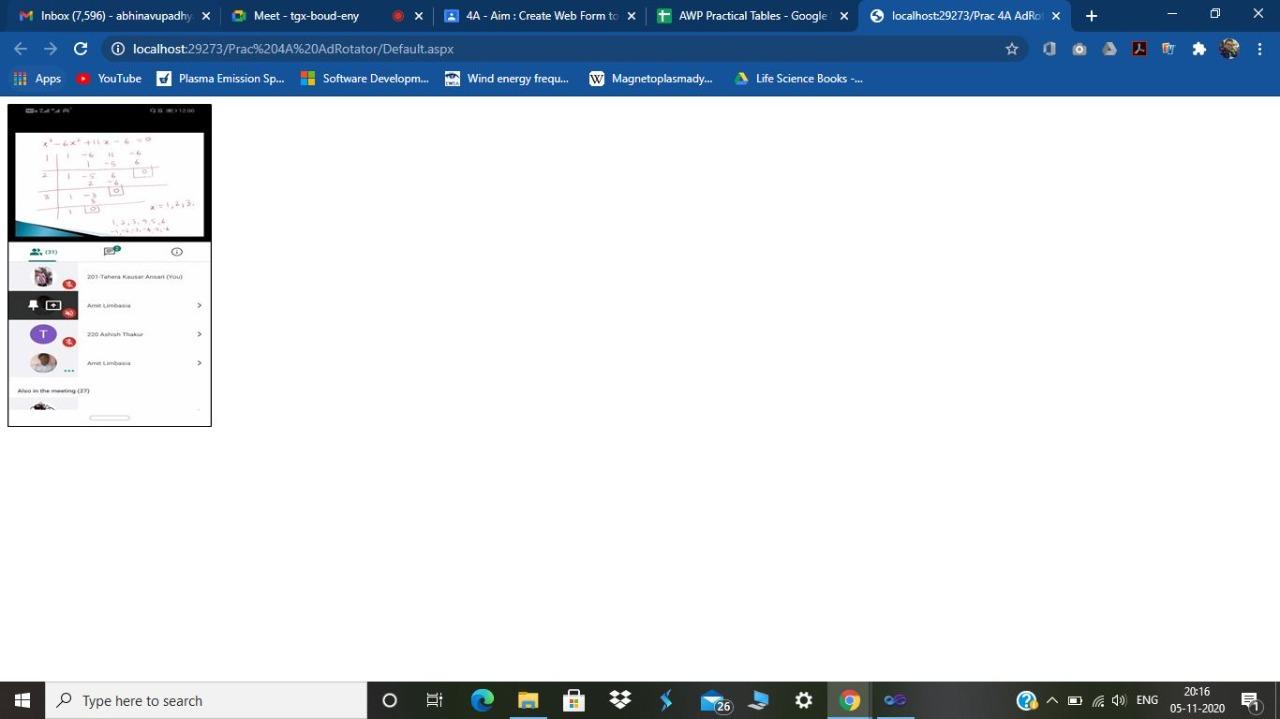
Property Table:

|  |  |  |  |
| --- | --- | --- | --- |
| srno | control | property | value |
| 1 | AdRotator | ID | AdRotator1 |
| 2 | AdRotator | DataSourceID | XmlDataSource1 |
| 3 | XmlDataSource | ID | XmlDataSource1 |
| 4 | XmlDataSource | DataFile | ~/XMLFile.xml |

Output:







Date:5/11/2020

Practical 4B

Name: Abhijeet

Rollno: 330

Aim: Create a web application to demonstrate use of Master Page.

Create a website for Dlita using the concept of Master page. Include the following objects in your master page.

1. Header -

2. Footer -

3. Menu including all events till date (minimum 3) FY Orientation, Speed Mathematics, Bridge Course

# 4. Content page Notice / Show Flow

Steps:

1. **Add Master page**
2. **Add Insert table**
3. **Add Header Image with Menu (Link to the respective web page)**
4. **Add Footer image**
5. **Add ContentPlaceHolder**
6. **Add 3 web forms with respective image (Link to the master page)**

**Code:**

**FY Orientation.aspx:**

<@ Page Title=”” Language=”C#”MasterPageFile=”~/MasterPage.master” AutoeventWireup=”true” CodeFile=”FY Orientation.aspx.cs” Inherits=”\_Default” %>

<asp:Content ID=”Content1” ContentPlaceHolder ID=”head” Runat=”Server”>

</asp:Content>

<asp:Content ID=”Content 3” ContentPlaceHolder ID=”ContentPlaceHolder2”

Runat=”Server”>

<asp:Image ID=”Image3” runat=”Server” Height=”400px”

ImageUrl=”~/FY Orientation.jpg” Width=”250px” />

<asp:Content>

**Speed Mathematics.aspx**

<@ Page Title=”” Language=”C#”MasterPageFile=”~/MasterPage.master” AutoeventWireup=”true” CodeFile=”Speed Mathematics” Inherits=”\_Default” %>

<asp:Content ID=”Content1” ContentPlaceHolder ID=”head” Runat=”Server”>

</asp:Content>

<asp:Content ID=”Content 3” ContentPlaceHolder ID=”ContentPlaceHolder2”

Runat=”Server”>

<asp:Image ID=”Image3” runat=”Server” Height=”400px”

ImageUrl=”~/Speed Mathematics.jpg” Width=”400px” />

<asp:Content>

**Bridge Course.aspx**

<@ Page Title=”” Language=”C#”MasterPageFile=”~/MasterPage.master” AutoeventWireup=”true” CodeFile=”Bridge Course.aspx.cs” Inherits=”\_Default” %>

<asp:Content ID=”Content1” ContentPlaceHolder ID=”head” Runat=”Server”>

</asp:Content>

<asp:Content ID=”Content 3” ContentPlaceHolder ID=”ContentPlaceHolder2”

Runat=”Server”>

<asp:Image ID=”Image3” runat=”Server” Height=”400px”

ImageUrl=”~/Bridge Course.jpg” Width=”400px” />

<asp:Content>

**MasterPage.master:**

<asp:Image ID=”Image1” runat=”Server” Height=”100px”

ImageUrl=”~/IMG.jpg Width=”400px” />

<asp:Menu iD=”Menu1 runat=”Server” Orientation=”Horizontal”>

<Items>

<asp:MenuItem NavigateUrl=”~/FY Orientation.aspx” Text=”FY Orientation”

Value=”FY Orientation”><asp:MenuItem>

<asp:MenuItem NavigateUrl=”~/Speed Mathematics.aspx” Text=”FY Orientation”

Value=”Speed Mathematics”><asp:MenuItem>

<asp:MenuItem NavigateUrl=”~/Bridge Course.aspx” Text=”FY Orientation”

Value=”Bridge Course”><asp:MenuItem>

</Items>

</asp:Menu>

<asp:ContentPlaceHolder ID=”ContentPlaceHolder2” runat=”Server”>

<asp:ContentPlaceHolder>

<asp:ImageUrl ID=”Image2” runat=”server” Height=”60px” ImageUrl=”~/dlita\_footer.jpg width=”400px” />

**Property table:**

|  |  |  |  |
| --- | --- | --- | --- |
| srno | control | property | value |
| 1 | Image | ID | Image1 |
| 2 | Image | Height | 100px |
| 3 | Image | width | 400px |
| 4 | Image | ImageURL | Header |
| 5 | Menu | ID | Menu1 |
| 6 | MenuItem | NavigateURL | FY Orientation.aspx |
| 7 | MenuItem | NavigateURL | Speed Mathematics. aspx |
| 8 | MenuItem | NavigateURL | Bridge Course.aspx |
| 9 | ConentPlacHolder | ID | ContentPlaceHolder2 |
| 10 | Image | ID | Image2 |
| 11 | Image | Height | 60px |
| 12 | Image | Width | 400px |
| 13 | Image | ImageURL | Footer |

**Output:**







Date:7/11/2020

Practical 4c

Name:Abhijeet

Roll no : 330

Aim:Create a Registration form to demonstrate use of various Validation controls.

1.RequiredFieldValidation Control

2.RangeValidation Control

3.CompareValidation Control

4.RegularExpressionValidation Control

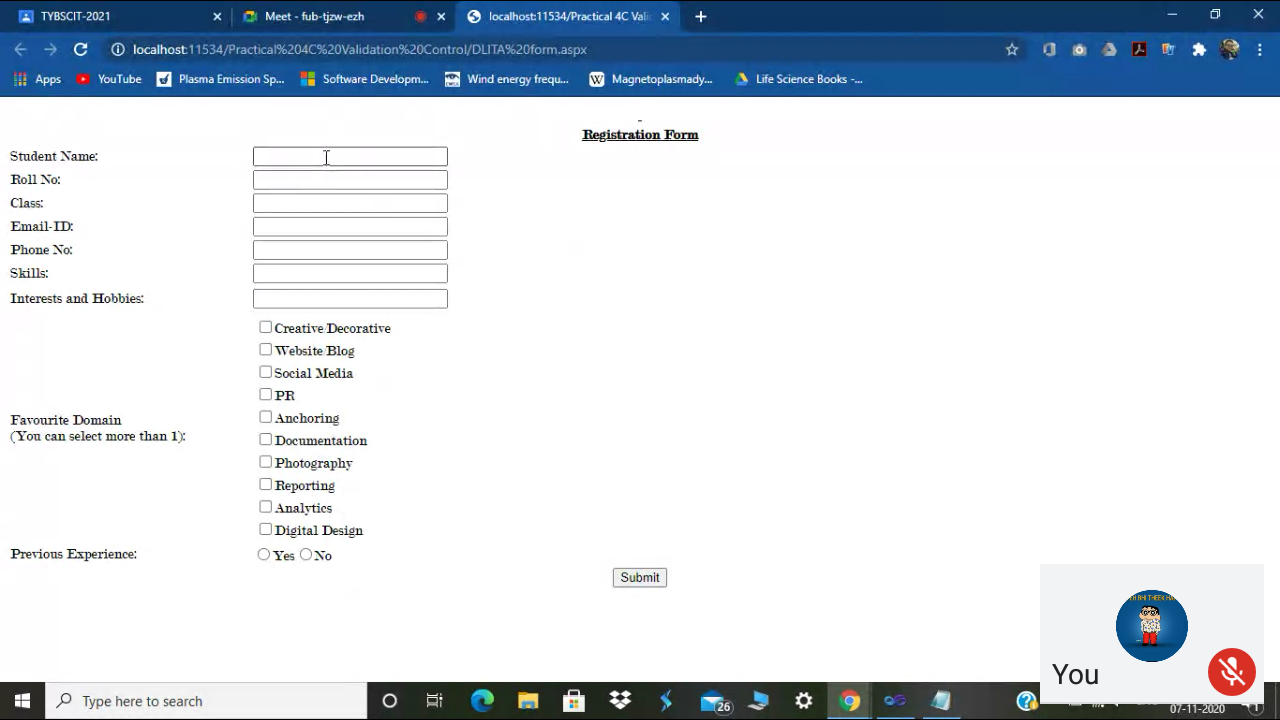
5.CustomValidator Control

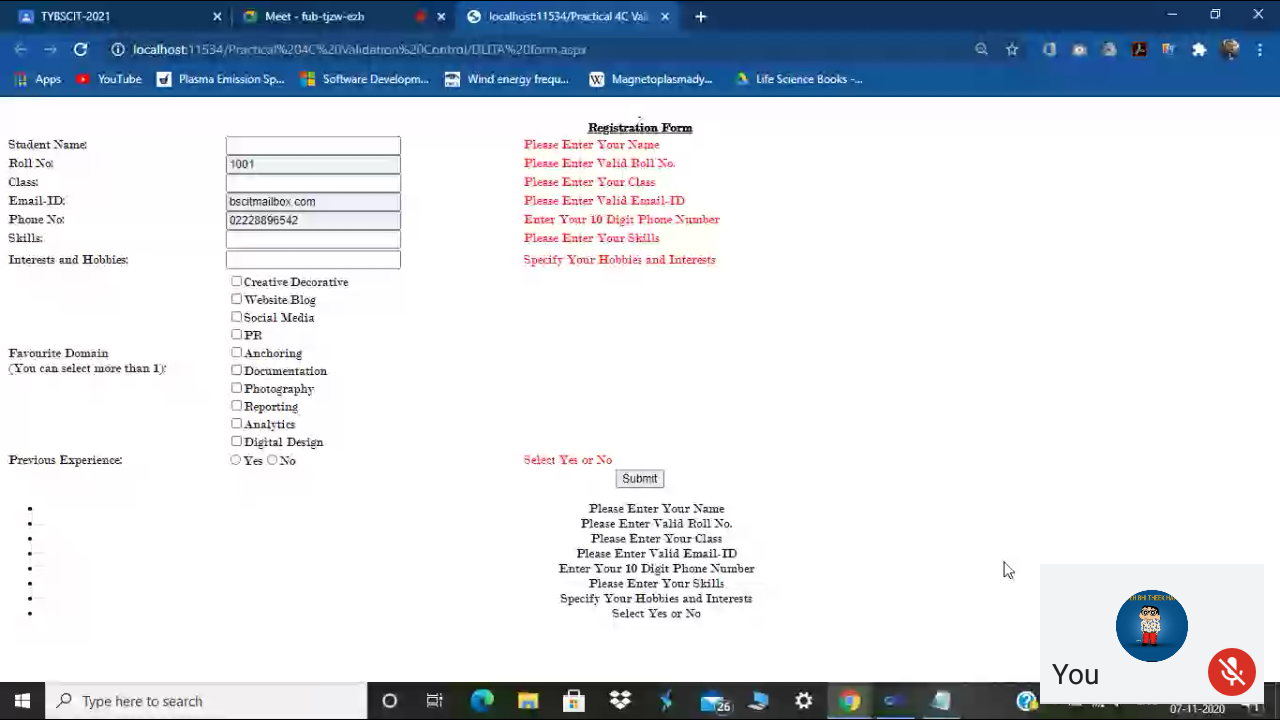
6.ValidationSummary

Property table :

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Control** | **Property** | **Value** |
| 1 | **RequiredFieldValidator** | **ID** | **RequiredFieldValidator2** |
| 2 | **RequiredFieldValidator** | **ControlToValidate** | **TextBox1** |
| 3 | **RequiredFieldValidator** | **ErrorMessage** | **Please Enter your name** |
| 4 | **RequiredFieldValidator** | **Display** | **Dynamic** |
| 5 | **CompareValidator** | **ID** | **CompareValidator1** |
| 6 | **CompareValidator** | **ControlToValidate** | **TextBox2** |
| 7 | **CompareValidator** | **ErrorMessage** | **Please Enter Valid Roll No** |
| 8 | **CompareValidator** | **Operator** | **LessThanEqual** |
| 9 | **RegularExpressionValidator** | **ID** | **RegularExpressionValidator1** |
| 10 | **RegularExpressionValidator** | **ControlToValidate** | **TextBox4** |
| 11 | **RegularExpressionValidator** | **ErrorMessage** | **Please Enter Valid Email-ID** |
| 12 | **RegularExpressionValidator** | **ValidationExpression** | **Expression** |
| 13 | **RangeValidator** | **ID** | **RangeValidator2** |
| 14 | **RangeValidator** | **ControlToValidate** | **TextBox5** |
| 15 | **RangeValidator** | **ErrorMessage** | **Enter Your 10 Digit Phone Number** |
| 16 | **RangeValidator** | **Type** | **Integer** |
| 17 | **ValidationSummary** | **ID** | **ValidationSummary1** |

**Output :**





DATE:-20/11/2020

NAME:-Abhijeet

Roll No:-330

PRACTICAL NO:-5a

Aim:-

CODE:-

Public partial class \_Default : System.web.UI.Page

{

int Counter=0;

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

{

}

Protected void Button\_Click(object sender,EventArgs e)

{

Counter =int.Parse(HiddenField1.Value);

Counter +=1;

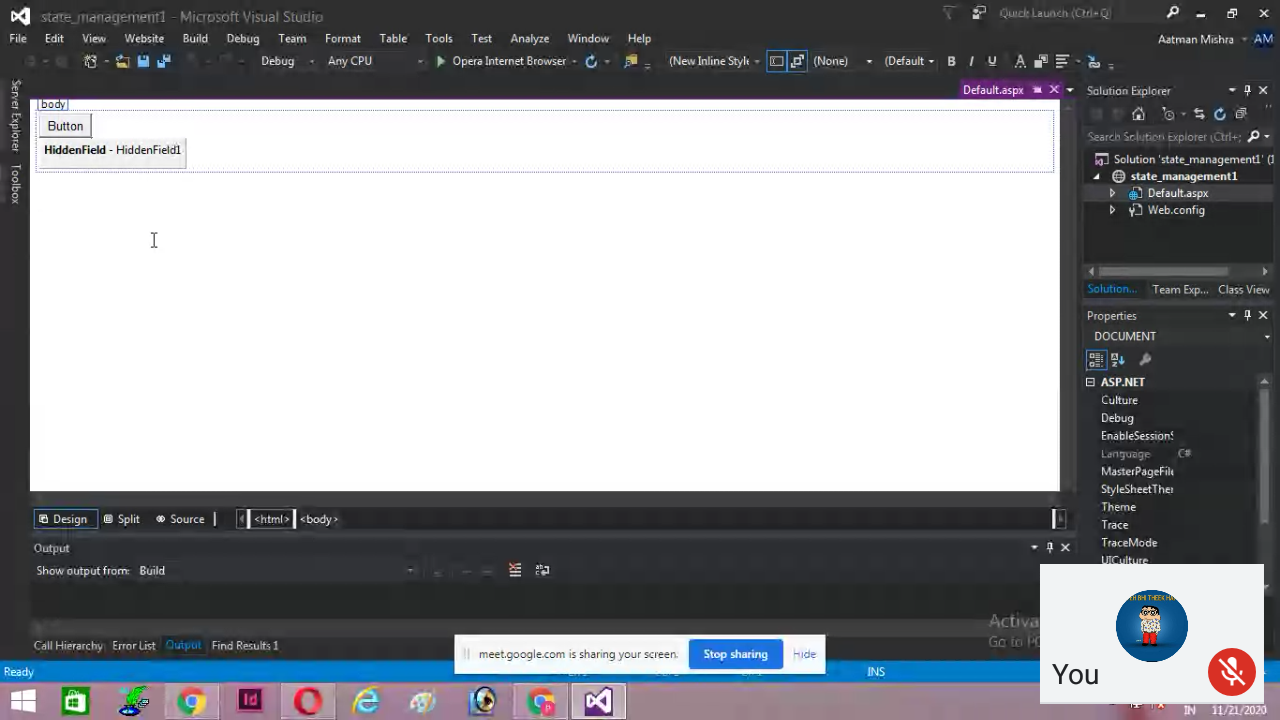
Response.Write(“Hit Count is “ + Counter);

HiddenField1.Value=Counter.ToString();

}

}

Output:



DATE:-21/11/2020

NAME:Abhijeet

Roll No:330

Practical no: 5b

Aim:

Code:-

**Global.asax**

Void Application\_Start(object sender, EventArgs e)

{

Application[“User”]=0;

}

Void Session\_start(object sender EventArgs e)

{

Application.Lock();

Application[“user”]=(int)Application[“user”]+1; *//Casting*

Application.UnLock();

}

Void Session\_End(object sender ,EventArgs e)

{

Application.Lock();

Application[“user”]=(int)Application[“user”]-1; *//Casting*

Application.UnLock();

}

**Web.config**

<configuration>

<system.web>

<sessionState mode=”InProc” timeout=”20” cookieless=”true”></sessionState>

<compilation debug=”true” targetFramework=”4.5.2” />

<httpRuntime targetFramework=”4.5.2”/>

</system.web>

</configuration>

**Default.aspx.cs**

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)

{

Label1.Text=(“THE NUMBER OF USERS ARE: “ +Application[“user”].ToString());

}

}

Output:

