UNIVERSITY OF MUMBAI

PRACTICAL ON

ADVANCED WEB TECHNOLOGY

SUBMITTED BY

DALVI TEJAS RAJENDRA

Application Id – 170021



D.T.S.S College of Commerce

Subject Faculty Name

Ms. Haneena Soney

MCA – I [ 2022 – 23 ]



UNIVERSITY OF MUMBAI

**INSTITUTE OF DISTANCE AND OPEN LEARNING (IDOL)**

**CERTIFICATE**

**THE EXPERIMENTS DULY SIGNED IN THIS PROJECT REPORT REPRESENT THE BONAFIDE**

**WORK BY MR. DALVI TEJAS RAJENDRA APPLICATION ID / SEAT NO. 170021 IN SEMESTER 2 OF FIRST YEAR OF MASTER OF COMPUTER APPLICATION (FYMCA - 1YR) OF PCP CENTER DTSS COLLEGE MALAD (EAST) FOR Advanced Web Technologies PRACTICAL DURING THE ACADEMIC YEAR 2022-2023.**

**LECTURE IN CHARGE HEAD OF DEPARTMENT COURSE IN CHARGE**

**Examiner \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_\_\_\_\_\_\_\_\_\_ DTSS College Seal**

INDEX

|  |  |  |
| --- | --- | --- |
| Module No : | Topics | Remarks |
| 1 | Write AWT program using console to Add Two Number |  |
| 2 | Write AWT program using console to implement string |  |
| 3 | Write AWT program using console to implement  Structure |  |
| 4 | Design a Window Application form |  |
| 5 | Design UI based application using basic Web  Application Forms Controls |  |
| 6 | Design UI based application using basic validation controls |  |
| 7 | Write an AWT program to create a database |  |
| 8 | Write an AWT program to create a database using methods |  |

# Aim : Write AWT program using console to Add Two Number

Program :

using System;

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

using System.Text;

using System.Threading.Tasks;

namespace AddTwoNumber

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the first number"); double num1 = Convert.ToDouble(Console.ReadLine());

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

double sum = num1 + num2;

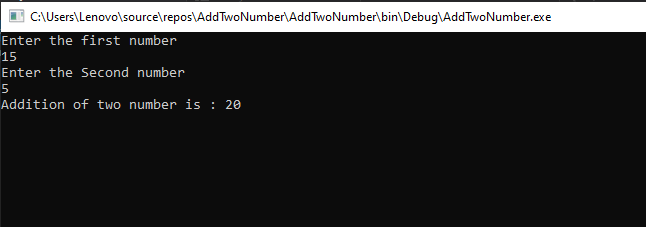
Console.WriteLine($"Addition of two number is : {sum}"); Console.ReadLine();

}

}

}

OutPut :



# Aim : Write AWT program using console to implement string

Code :

using System;

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

using System.Text;

using System.Threading.Tasks;

namespace StringManipulation

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter a string :"); string inputString = Console.ReadLine();

Console.WriteLine($"String is : {inputString}");

Console.WriteLine($"UpperCase : {inputString.ToUpper()}"); Console.WriteLine($"LowerCase : {inputString.ToLower()}"); Console.WriteLine($"Length of string : {inputString.Length}");

Console.WriteLine($"First Character : {inputString[0]}");

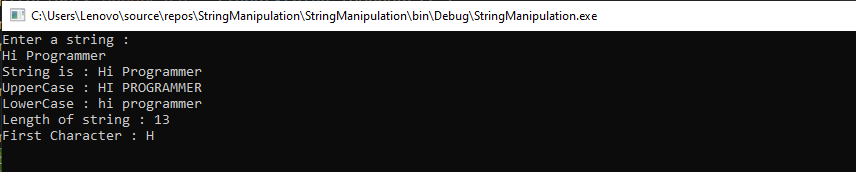
Console.ReadLine();

}

}

}

Output :



# Aim : Write AWT program using console to implement Structure

Code :

using System;

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

using System.Text;

using System.Threading.Tasks;

namespace ImplementStructure

{

struct Person

{

string Name; int Age;

public Person(String name,int age)

{

Name = name; Age = age;

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter Name :"); string name = Console.ReadLine();

Console.WriteLine("Enter Age : "); if(int.TryParse(Console.ReadLine(), out int age))

{

Person p = new Person(name, age);

}

else

{

Console.WriteLine($"Person name is : {name}"); Console.WriteLine($"Person age is : {age} "); Console.ReadLine();

Console.WriteLine("Invalid age"); Console.ReadLine();

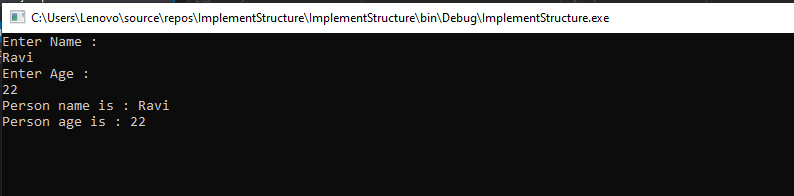
}

}

}

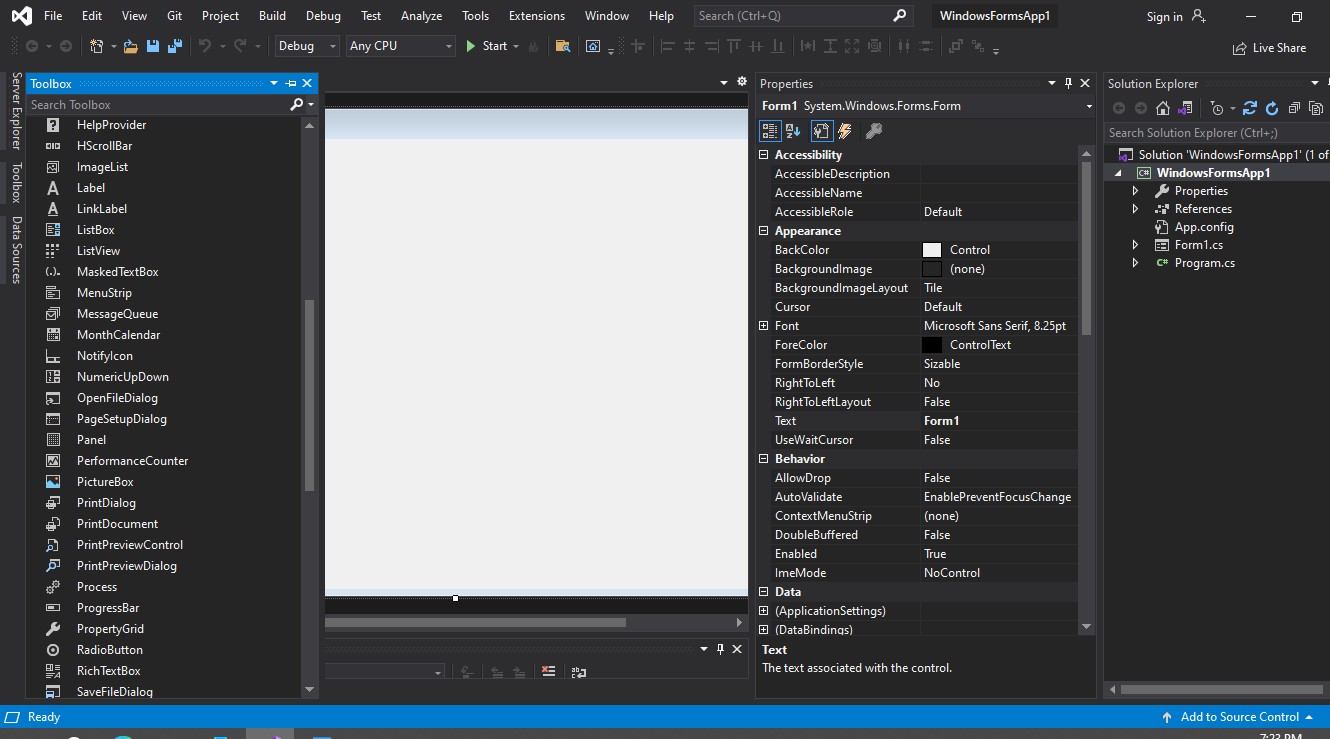
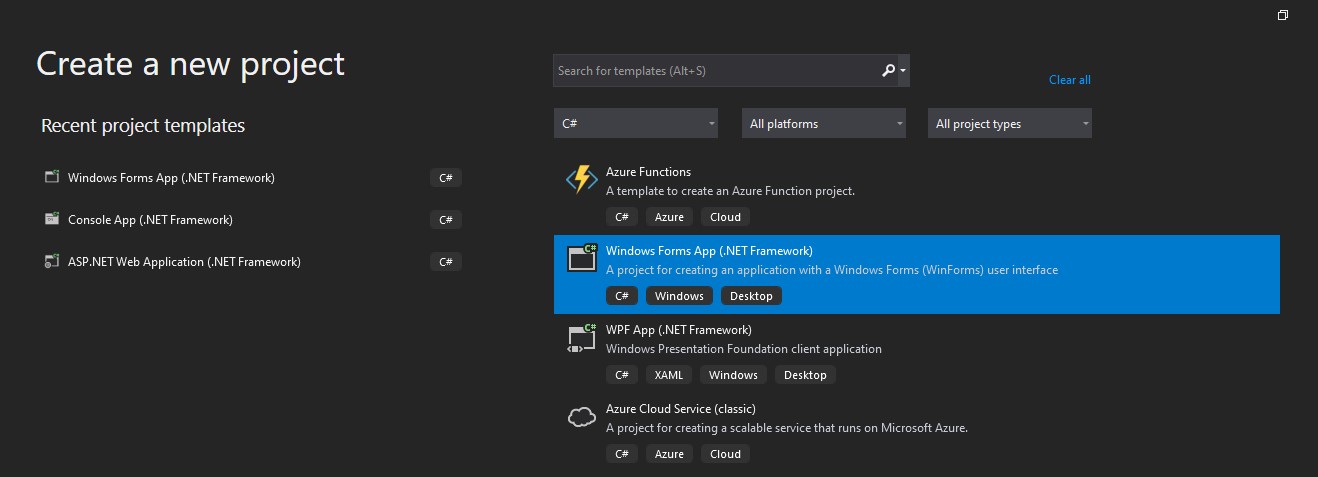
}

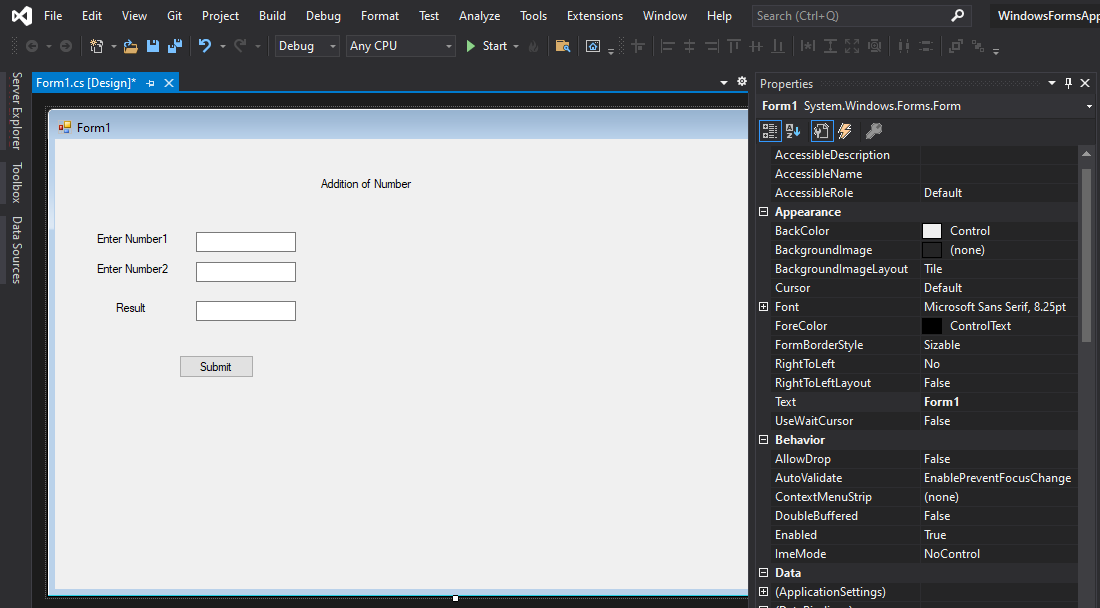
Output :



# Aim : Design a Window Application form

Steps:





Code :

using System;

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

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

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

namespace WindowsFormsApplication

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

int a, b, c;

a = Convert.ToInt32(textBox1.Text); b = Convert.ToInt32(textBox2.Text);

c = a + b;

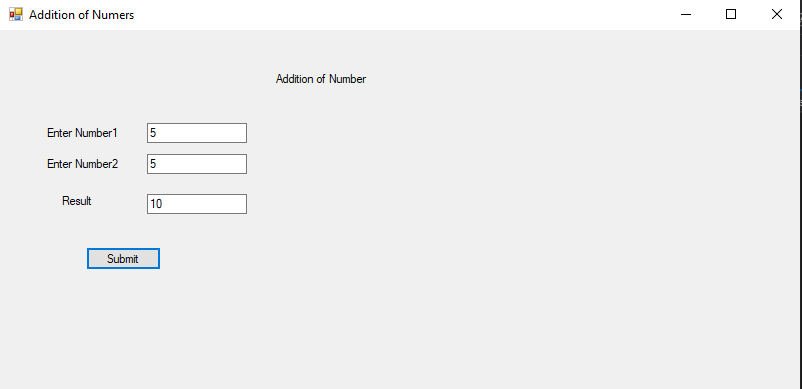
textBox3.Text = c.ToString();

}

}

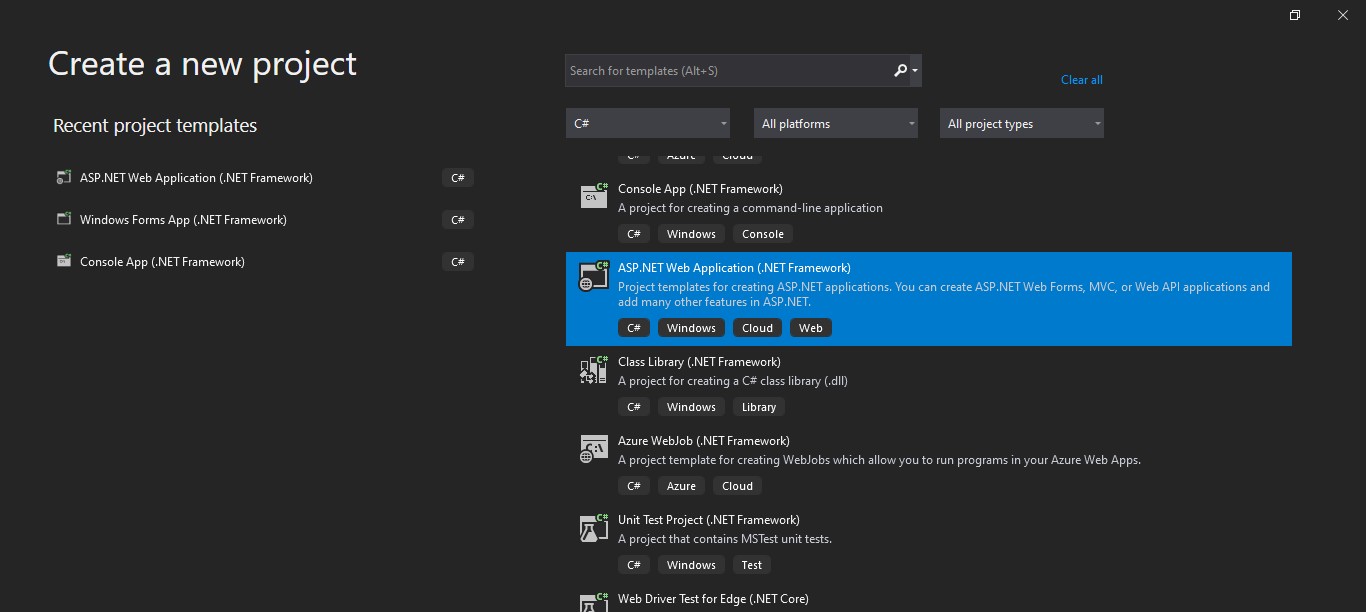
}

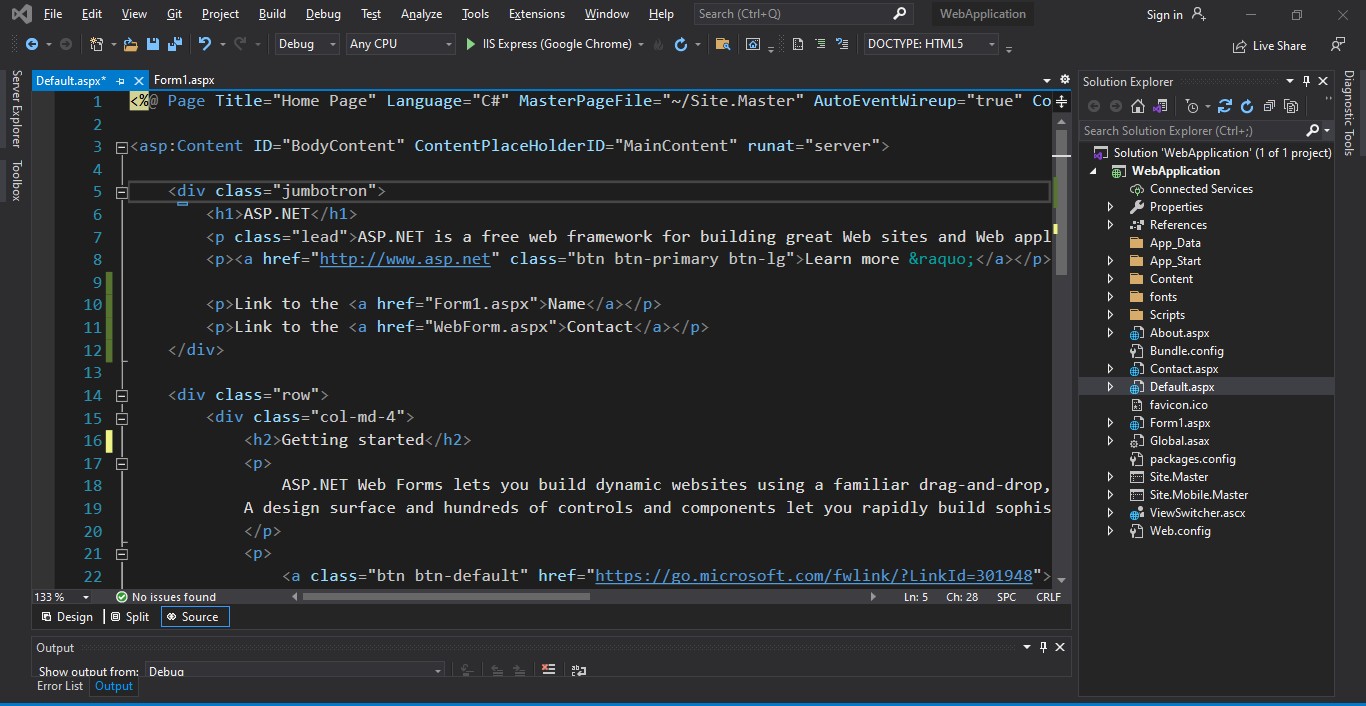
Output :

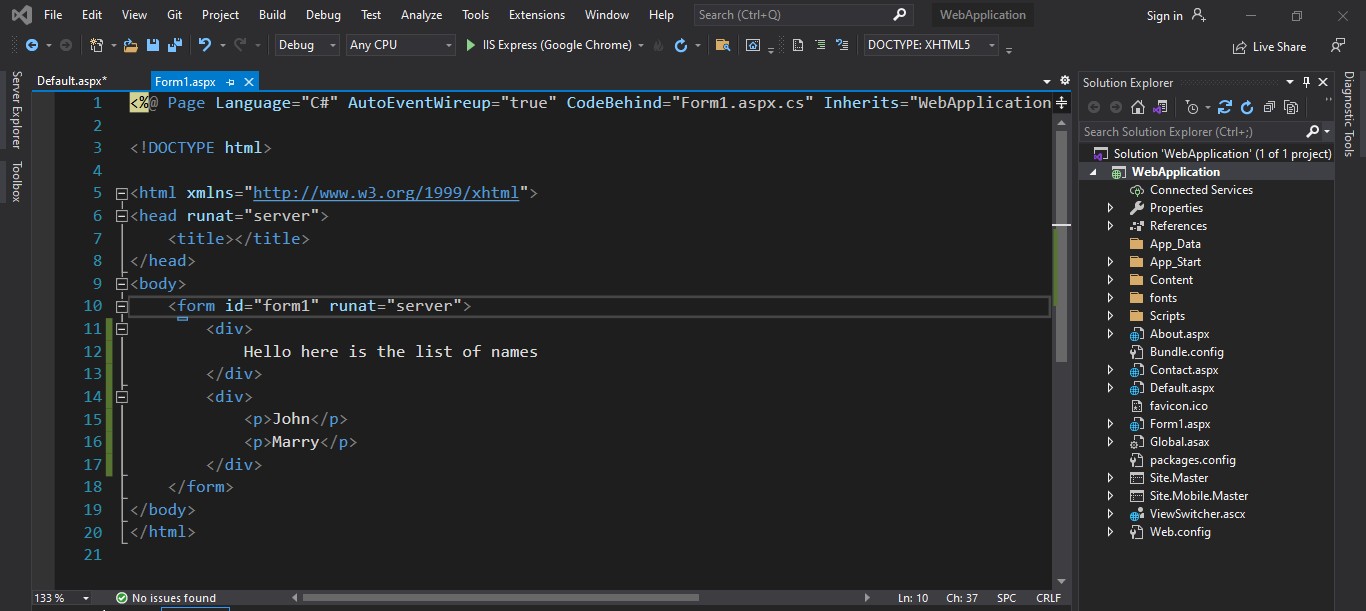


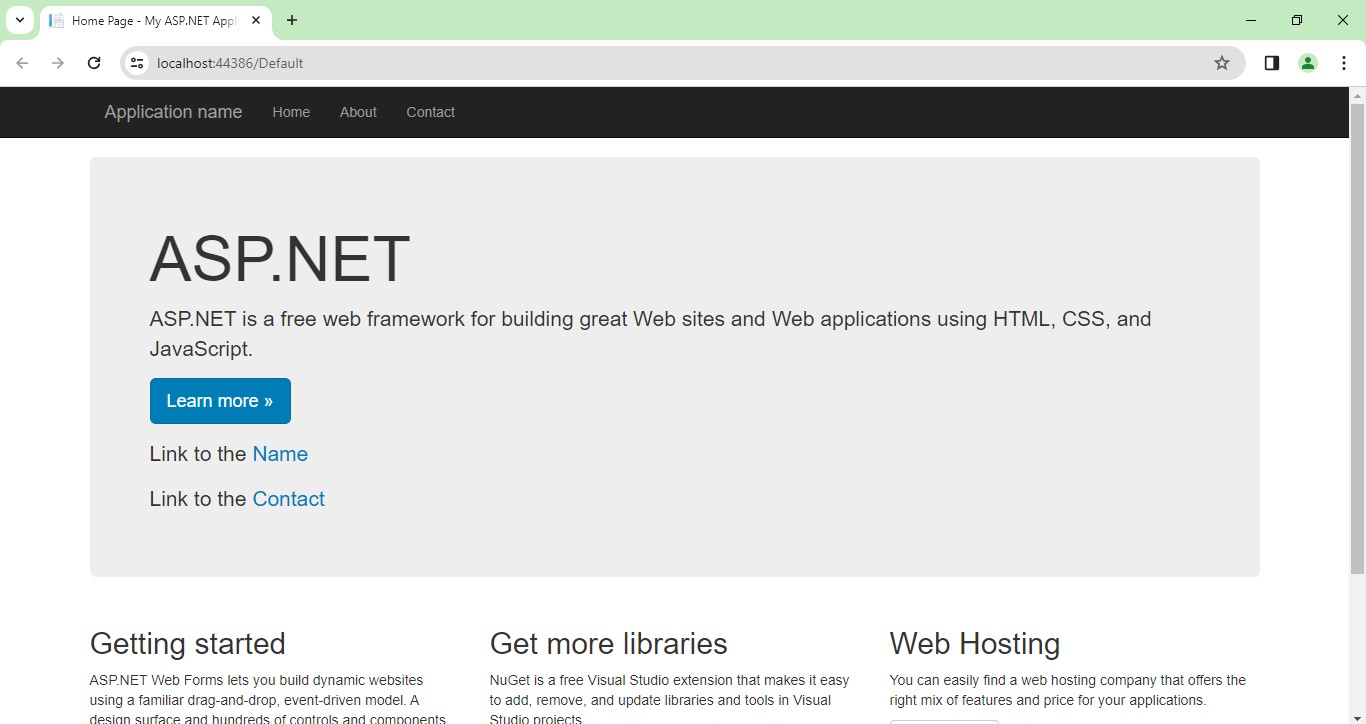
# Aim : Design UI based application using basic Web Application Forms Controls

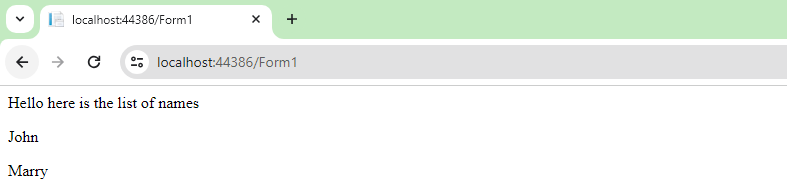
Steps :



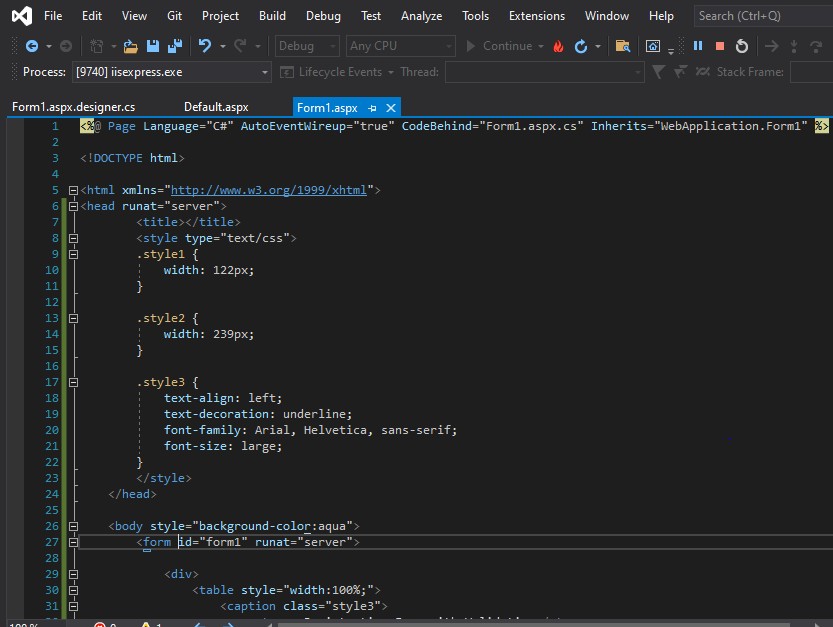








# AIM : Design UI based application using basic validation controls



Code :

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Form1.aspx.cs" Inherits="WebApplication.Form1" %>

<!DOCTYPE html>

<html xmlns="<http://www.w3.org/1999/xhtml>">

<head runat="server">

<title></title>

<style type="text/css">

.style1 {

width: 122px;

}

.style2 {

width: 239px;

}

.style3 {

text-align: left;

text-decoration: underline;

font-family: Arial, Helvetica, sans-serif; font-size: large;

}

</style>

</head>

<body style="background-color:aqua">

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

<div>

<table style="width:100%;">

<caption class="style3">

<strong>Registration Form with Validation</strong>

</caption>

<tr>

<td class="style1">

</td>

<td class="style2">

</td>

<td>

</td>

</tr>

<tr>

<td class="style1">

<asp:Label ID="Label1" runat="server" Text="FirstName:"></asp:Label>

</td>

<td class="style2">

<asp:TextBox ID="TextBox1" runat="server" Height="22px" MaxLength="20" Width="158px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server" ControlToValidate="TextBox1" ErrorMessage="Please Enter your First Name" ForeColor="#CC0000"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td class="style1">

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

</td>

<td class="style2">

<asp:TextBox ID="TextBox2" runat="server" Height="22px" MaxLength="10" Width="158px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="TextBox2" ErrorMessage="Please Enter your Last Name" ForeColor="#CC0000"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td class="style1">

<asp:Label ID="Label3" runat="server"

Text="Email:"></asp:Label>

</td>

<td class="style2">

<asp:TextBox ID="TextBox3" runat="server" Height="22px" MaxLength="15" Width="158px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="TextBox3" ErrorMessage="Please Enter your Email ID" ForeColor="#CC0000"></asp:RequiredFieldValidator>

</td>

<td>

<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server" ControlToValidate="TextBox3" ValidationExpression="\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*" ErrorMessage="Invalid Email Id"></asp:RegularExpressionValidator>

<td>

</td>

</td>

</tr>

<tr>

<td class="style1">

<asp:Label ID="Label4" runat="server" Text="Phone No.

:"></asp:Label>

</td>

<td class="style2">

<asp:TextBox ID="TextBox4" runat="server" Height="22px"

MaxLength="10" Width="158px"></asp:TextBox>

</td>

<td>

<asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ControlToValidate="TextBox4" ErrorMessage="Please Enter your Phone No" ForeColor="#CC0000"></asp:RequiredFieldValidator>

</td>

</tr>

<tr>

<td class="style1">

</td>

<td class="style2">

</td>

<td>

</td>

</tr>

<tr>

<td class="style1">

</td>

<td class="style2">

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

</td>

<td>

</td>

</tr>

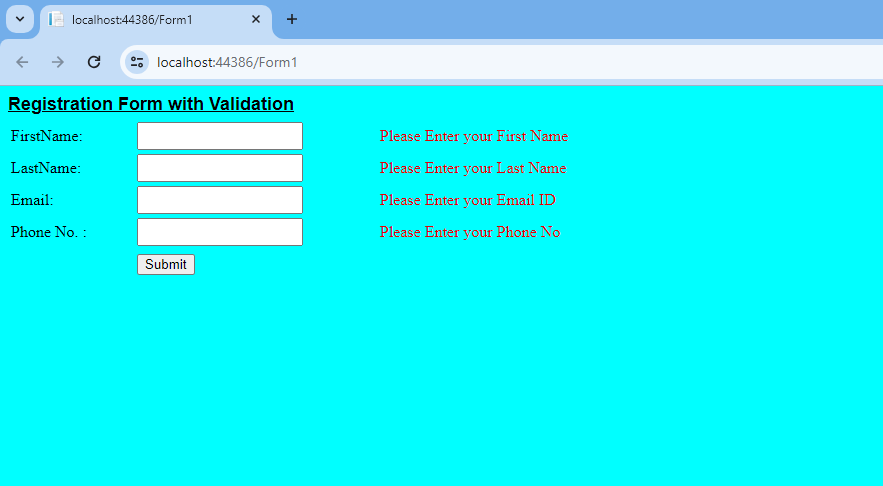
</table>

</div>

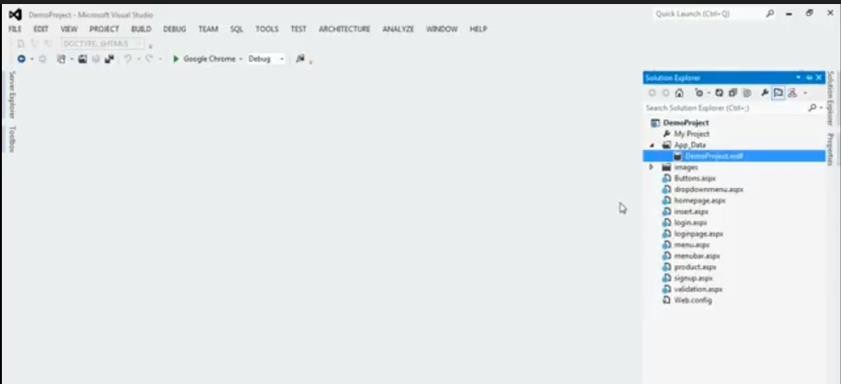
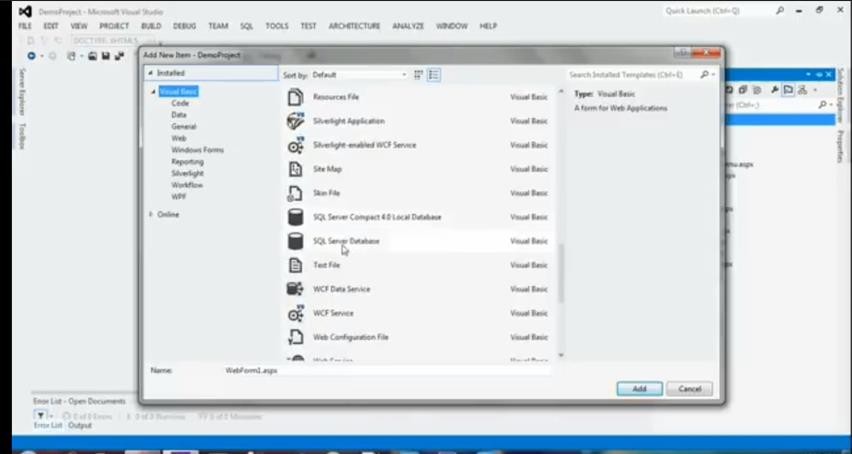
</form>

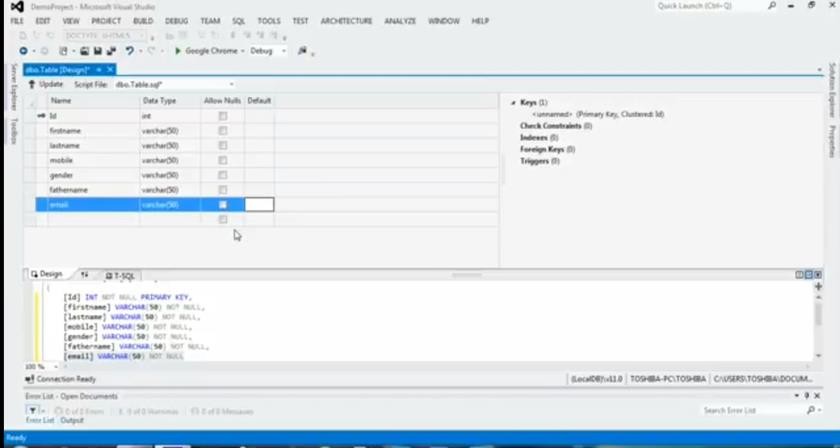
</body>

</html>



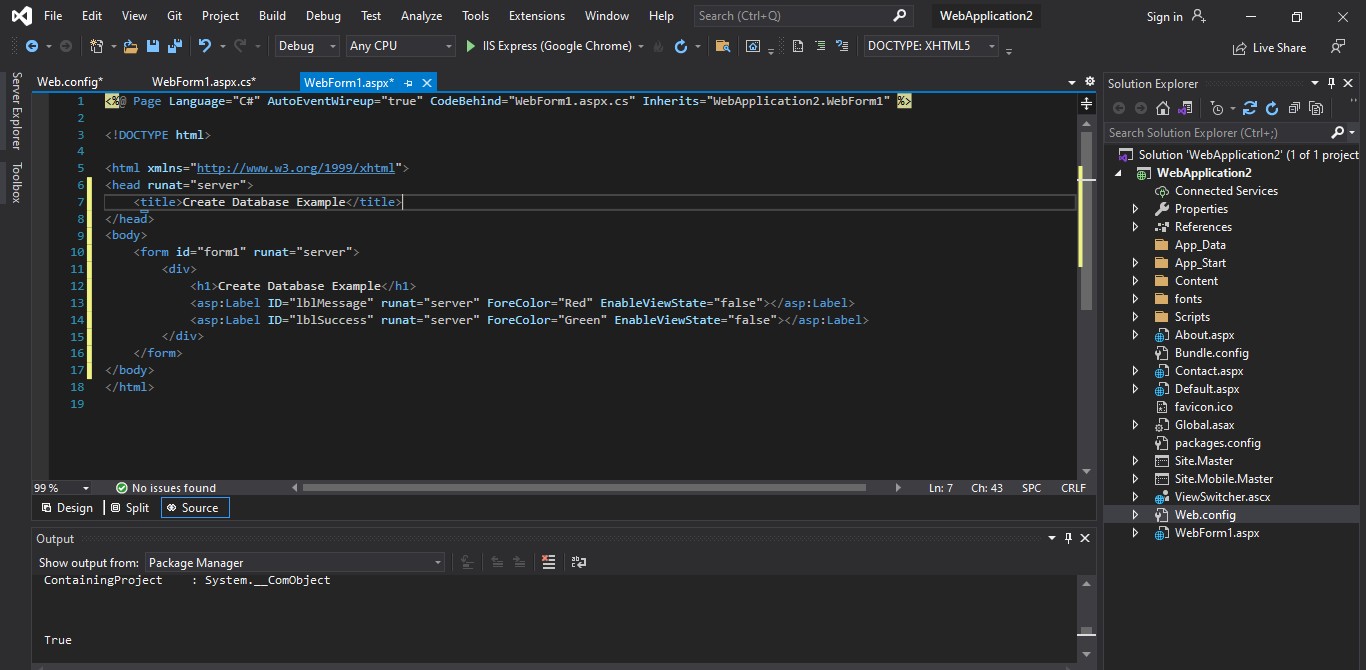
# AIM : Write an AWT program to create a database :





AIM : Write an AWT program to create a database using methods

Step



Webform.aspx.cs code :

using System;

using System.Collections.Generic; using System.Configuration;

using System.Data.SqlClient; using System.Linq;

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

using System.Web.UI.WebControls;

namespace WebApplication2

{

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

{

string connectionString = ConfigurationManager.ConnectionStrings["ConnectionStringName"].ConnectionString;

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

{

if (!IsPostBack)

{

bool success = CreateDatabase(); if (success)

{

lblSuccess.Text = "Database created successfully!";

}

}

}

private bool CreateDatabase()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection))

NVARCHAR(50))";

connection))

try

{

}

connection.Open();

string createDatabaseQuery = "CREATE DATABASE Employee";

using (SqlCommand command = new SqlCommand(createDatabaseQuery,

{

command.ExecuteNonQuery();

}

connection.ChangeDatabase("Employee");

string createTableQuery = "CREATE TABLE EMP (ID INT PRIMARY KEY, Name using (SqlCommand command = new SqlCommand(createTableQuery,

{

command.ExecuteNonQuery();

}

lblMessage.Text = ""; return true;

catch (Exception ex)

{

lblMessage.Text = "Error: " + ex.Message; return false;

}

finally

{

connection.Close();

}

}

}

}

}