**Part 1 :- Console Application**

**Program 1 :**

**AIM** **:-** Write a program using Console Applications (Use ReadLine() & WriteLine() functions).

WAP to enter Employee Name, Age, Joining Date, BASIC. Calculate DA, HRA, PF,

Gross pay & Net Pay and display it with Employee information.

**Program :-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication\_1

{

class Program

{

string name,date;

int age;

double basis,DA,HRA,PF,gross\_pay,net\_pay;

public void getdata()

{

Console.WriteLine("\n----- Enter Employee details -----");

Console.Write("\nEnter name : ");

name = Convert.ToString(Console.ReadLine());

Console.Write("\nEnter age : ");

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

Console.Write("\nEnter date : ");

date=Convert.ToString(Console.ReadLine());

Console.Write("\nEnter salary : ");

basis=Convert.ToDouble(Console.ReadLine());

}

public void display()

{

Console.WriteLine("\n---------- Employee details ----------");

Console.WriteLine("\nName : " + name);

Console.WriteLine("\nAge : " + age);

Console.WriteLine("\nJoing date : " + date);

Console.WriteLine("\nSalary : " + basis);

DA = (1.26 \* basis);

HRA = (0.2 \* basis);

PF = (0.12 \* basis);

gross\_pay = (basis + DA + HRA);

net\_pay = (gross\_pay - PF);

Console.WriteLine("\nDA : " + DA);

Console.WriteLine("\nHRA : " + HRA);

Console.WriteLine("\nPF : " + PF);

Console.WriteLine("\nGross pay : " + gross\_pay);

Console.WriteLine("\nNet pay : " + net\_pay);

}

}

class prac1\_demo

{

static void Main(string []a)

{

Program e = new Program();

e.getdata();

e.display();

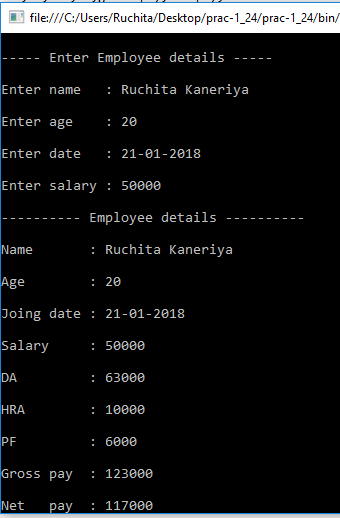
Console.Read();

}

}

}

**Output :-**



**Program 2 :**

**AIM** **:-** Write C# Menu-Based program to do the following:

* Convert binary to decimal
* Convert decimal to hexadecimal
* Convert decimal to binary
* Convert decimal to octal.
* Create a separate class for each functionality and put each class in a separate namespace in the same program.

**Program :-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace numberconversion

{

class Program

{

static void Main(string[] args)

{

int choice;

String ch = "y";

Console.Write("----- Number Conversion -----\n");

while (ch.Equals("y"))

{

Console.WriteLine("\n1. Binary to Decimal\n2. Decimal to HexaDecimal\n3.

Decimal to Binary\n4. Decimal to Octal");

Console.Write("\nEnetr your choice : ");

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

switch(choice)

{

case 1: Console.WriteLine("\n----- Binary to Decimal Conversion -----");

binary\_decimal.c1 ob1=new binary\_decimal.c1();

ob1.bin\_dec();

break;

case 2: Console.WriteLine("\n----- Decimal to Hexadecimal Conversion ------");

decimal\_hexadecimal.c2 ob2 = new decimal\_hexadecimal.c2();

ob2.dec\_hexadec();

break;

case 3: Console.WriteLine("\n----- Decimal to Binary Conversion -----");

decimal\_binary.c3 ob3=new decimal\_binary.c3();

ob3.dec\_bin();

break;

case 4: Console.WriteLine("\n----- Decimal to Octal Conversion -----");

decimal\_octal.c4 ob4=new decimal\_octal.c4();

ob4.bin\_oct();

break;

}

Console.Write("\n\nDo you want to Continue ?(y/n) : ");

ch = Console.ReadLine();

}

}

}

namespace binary\_decimal

{

class c1

{

public void bin\_dec()

{

int n, decimal\_no = 0, rem, base\_no = 1;

Console.WriteLine("\nEnter a Binary number : (0/1)");

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

while (n > 0)

{

rem = n % 10;

decimal\_no = decimal\_no + rem \* base\_no;

n = n / 10;

base\_no = base\_no \* 2;

}

Console.Write("Decimal equivalent is : " + decimal\_no);

}

}

}

namespace decimal\_hexadecimal

{

class c2

{

public void dec\_hexadec()

{

int n, i=1, rem = 0;

char[] hex = new char[100];

char rem1;

Console.Write("\nEnter a Decimal number : ");

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

while(n!=0)

{

rem=n%16;

if(rem<10)

rem=rem+48;

else

rem=rem+55;

rem1=Convert.ToChar(rem);

for(i=0;n>0;i++)

{

hex[i++] = rem1;

n = n / 16;

}

}

Console.Write("Hexadecimal equivalent is : ");

for (i = i - 1; i >= 0; i--)

{

Console.Write(hex[i]);

}

}

}

}

namespace decimal\_binary

{

class c3

{

public void dec\_bin()

{

int i, n;

int[] bin = new int[15];

Console.Write("\nEnter a Decimal number : ");

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

for (i = 0; n > 0; i++)

{

bin[i] = n % 2;

n = n / 2;

}

Console.Write("Binary equivalent is : ");

for (i = i - 1; i >= 0; i--)

{

Console.Write(bin[i]);

}

}

}

}

namespace decimal\_octal

{

class c4

{

public void bin\_oct()

{

int n, i;

int[] oct = new int[100];

Console.Write("\nEnter a Decimal Number : ");

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

for (i = 0; n > 0; i++)

{

oct[i] = n % 8;

n = n / 8;

}

Console.Write("Octal equivalent is : ");

for (i = i - 1; i >= 0; i--)

{

Console.Write(oct[i]);

}

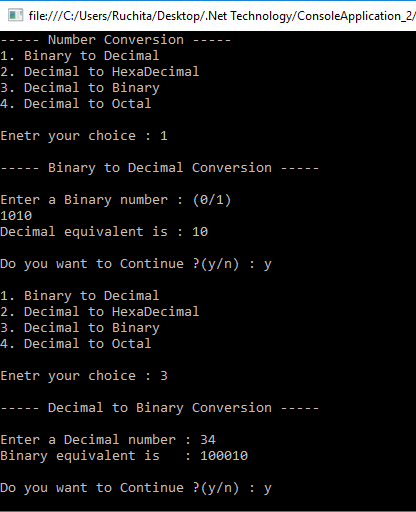
}

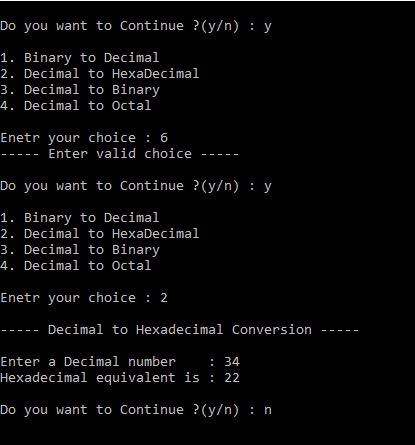
}

}

}

**Output :-**





**Program 3 :**

**AIM** **:-** Create Console Applications to implement following C# Concepts:

* Constructor & Destructor
* Method Overloading
* Inheritance
* Properties and Indexer

**Program :-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication\_3

{

class student

{

public student()

{

Console.WriteLine("Inside Constructor");

}

~student()

{

Console.WriteLine("Inside Destructor");

Console.Read();

}

}

class marksheet : student

{

double sub1,sub2,sub3;

public marksheet()

{

sub1 = 0; sub2 = 0; sub3 = 0;

}

public marksheet(double s1, double s2, double s3)

{

sub1 = s1; sub2 = s2; sub3 = s3;

}

public double get\_total()

{

return (sub1 + sub2 + sub3);

}

public void get\_total(double s4,double s5)

{

Console.WriteLine("\n---------- Method overloading ----------");

Console.WriteLine("Total :" +(s4+s5));

}

public static marksheet operator +(marksheet m1, marksheet m2)

{

marksheet m = new marksheet();

m.sub1 = m1.sub1 + m2.sub1;

m.sub2 = m1.sub2 + m2.sub2;

m.sub3 = m1.sub3 + m2.sub3;

return m;

}

}

class studentdemo

{

static void Main(string[] args)

{

marksheet ob = new marksheet();

Console.WriteLine("Total marks : "+ob.get\_total());

ob.get\_total(90, 90);

marksheet ob1 = new marksheet(99, 99, 99);

marksheet ob2 = new marksheet(99, 99, 99);

marksheet ob3 = new marksheet();

Console.WriteLine("\n---------- Operator Overloading ----------");

ob3 = ob1 + ob2;

Console.WriteLine("Total of ob3 : "+ob3.get\_total());

Console.WriteLine("Before main ends");

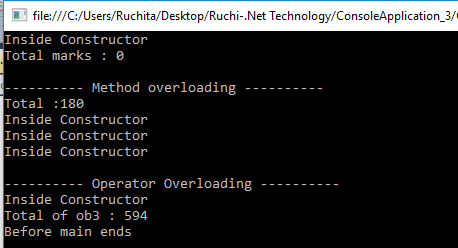
Console.ReadLine();

}

}

}

**Output :-**



**Program 4 :**

**AIM** **:-** Create a class to demonstrate *static property* by counting number of objects created.

**Program :-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication\_4

{

class circle

{

static int count = 0;

public circle()

{

count++;

}

public static int cnt

{

get

{ return count;}

}

}

class circle\_demo

{

static void Main(string[] args)

{

circle c1 = new circle();

Console.WriteLine("Count of object : "+circle.cnt);

circle c2 = new circle();

circle c3 = new circle();

Console.WriteLine("Count of object : "+circle.cnt);

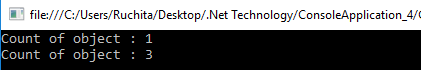
Console.ReadLine();

}

}

}

**Output :-**



**Program 5 :**

**AIM** **:-** Create a class MyStringIndexer with data members title, author, subject.

It should contain

- Constructor with arguments

- Indexer with string index

* Create a class IndexerDemo for Main function. Write code to get and set values of any data members of MyStringIndexer class.

**Program :-**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication\_5

{

class MyStringIndexer

{

string subject, title, author;

public MyStringIndexer(string x, string y, string z)

{

subject = x;

title = y;

author = z;

}

public string this[string str]

{

get

{

if(str=="subject")

return subject;

if (str == "title")

return title;

if (str == "author")

return author;

return "";

}

set

{

if (str == "subject")

subject = value;

if (str == "title")

title = value;

if (str == "author")

author = value;

}

}

}

class IndexDemo

{

static void Main(string[] args)

{

MyStringIndexer r = new MyStringIndexer("Advance Java","JDBC Connection","XYZ");

Console.WriteLine("Subject : "+r["subject"]);

Console.WriteLine("Title : "+r["title"]);

Console.WriteLine("Author : "+r["author"]);

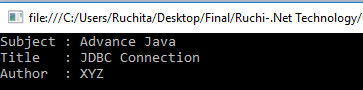
Console.ReadLine();

}

}

}

**Output :-**



**Part 2 :- Windows Forms**

**Program 1 :**

**AIM** **:-** Create a Windows Form to Convert following currency conversion.

Rupees to dollar,frank, euro.

**Program :-**

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 WindowsFormsApplication\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

double rs = Convert.ToInt32(textBox1.Text);

double dollar = (rs \* 0.016);

double frank = (rs \* 0.015);

double euro = (rs \* 0.013);

textBox2.Text = dollar.ToString("0.000");

textBox3.Text = frank.ToString("0.000");

textBox4.Text = euro.ToString("0.000");

}

else

{

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

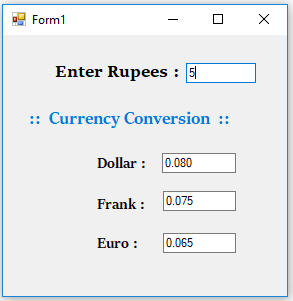
}

}

}

}

**Output :-**



**Program 2 :**

**AIM** **:-** Create a Windows Form to convert infix notation to postfix notation.

Ex: a+b-c => ab+c

**Program :-**

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 WindowsFormsApplication\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

label3.Text = "";

int priority = 0;

Stack<char> s = new Stack<char>();

for (int i = 0; i < textBox1.Text.Length; i++)

{

char ch = textBox1.Text[i];

if (ch == '+' || ch == '-' || ch == '\*' || ch == '/')

{

if (s.Count <= 0)

s.Push(ch);

else

{

if (s.Peek() == '\*' || s.Peek() == '/')

priority = 1;

else

priority = 0;

if (priority == 1)

{

if (ch == '+' || ch == '-')

{

label3.Text = label3.Text + s.Pop();

i--;

}

else

{

label3.Text = label3.Text + s.Pop();

i--;

}

}

else

{

if (ch == '+' || ch == '-')

{

label3.Text = label3.Text + s.Pop();

s.Push(ch);

}

else

s.Push(ch);

}

}

}

else

label3.Text = label3.Text + ch;

}

int l = s.Count;

for (int j = 0; j < l; j++)

{

label3.Text = label3.Text + s.Pop();

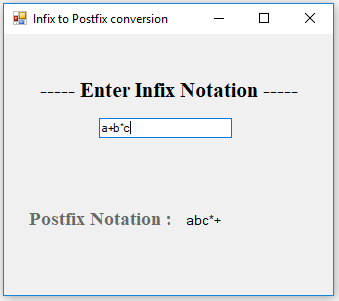
}

}

}

}

**Output :-**



**Program 3 :**

**AIM** **:-** Create a Windows Form to convert digits to words.

Ex: 123 => One Hundred and Twenty-three

**Program :-**

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication\_5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

int num = Convert.ToInt32(textBox1.Text);

if (num == 0)

{

label3.Text = "Zero";

}

string[] unit1 = new[] { "Zero", "One", "Two", "Three", "Four", "Five",

"Six", "Seven", "Eight", "Nine", "Ten" };

label3.Text = "";

if ((num / 1000000) > 0)

{

label3.Text = label3.Text + unit1[num / 10000000] + "million";

num = num % 10000000;

}

if ((num / 1000) > 0)

{

label3.Text = label3.Text + unit1[num / 1000] + " thousand ";

num = num % 1000;

}

if ((num / 100) > 0)

{

label3.Text = label3.Text + unit1[num / 100] + " hundred ";

num = num % 100;

}

if (num > 0)

{

if (label3.Text != "")

{

label3.Text = label3.Text + " and ";

}

string[] unit2 = new[] { "Zero", "One", "Two", "Three", "Four", "Five", "Six",

Seven", "Eight", "Nine","Ten","Eleven","Twelve","

Thirteen","Fourteen","Fifteen","Sixteen","Seventeen",

“Eightteen","Nineteen" };

string[] unit3 = new[] { "Zero", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty",

"Seventy", "Eighty", "Ninety", "Ten" };

if (num < 20)

label3.Text = label3.Text + unit2[num];

else

{

label3.Text = label3.Text + unit3[num / 10];

if((num%10)>0)

{

label3.Text=label3.Text+" "+unit2[num%10];

}

}

}

}

else

{

label3.Text="Enter number first : ";

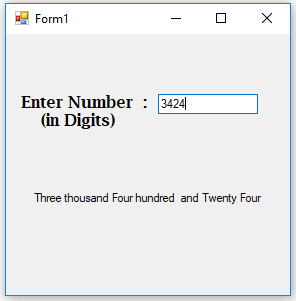
}

}

}

}

**Output :-**



**Program 4 :**

**AIM** **:-** Create a Windows Form to implement Calculator Application.

**Program :-**

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 WindowsFormsApplication\_4

{

public partial class Calculator : Form

{

double ans = 0;

String str;

bool opration\_performed = false;

public Calculator()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (textBox1.Text == "0")

{

textBox1.Text = "";

}

}

private void textbox1\_keypress(object sender, KeyPressEventArgs e)

{ }

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + zero.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + one.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + two.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + three.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + four.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + five.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + six.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + seven.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + eight.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

textBox1.Text = textBox1.Text + nine.Text;

opration\_performed = false;

}

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

{

if (opration\_performed)

textBox1.Clear();

if (textBox1.Text.Contains("."))

{

textBox1.Text = textBox1.Text;

}

else

{

textBox1.Text=textBox1.Text+dot.Text;

}

opration\_performed=false;

}

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

{

textBox1.Text = "0";

ans = 0;

}

private void operation(object sender, EventArgs e)

{

Button btn = (Button)sender;

str = btn.Text;

ans = Double.Parse(textBox1.Text);

label1.Text = ans + " " + str;

opration\_performed = true;

}

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

{

switch (str)

{

case "+": textBox1.Text = (ans + Double.Parse(textBox1.Text)).ToString();

break;

case "-": textBox1.Text = (ans - Double.Parse(textBox1.Text)).ToString();

break;

case "\*": textBox1.Text = (ans \* Double.Parse(textBox1.Text)).ToString();

break;

case "/": textBox1.Text = (ans / Double.Parse(textBox1.Text)).ToString();

break;

}

ans = Double.Parse(textBox1.Text);

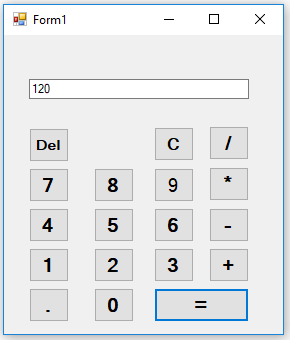
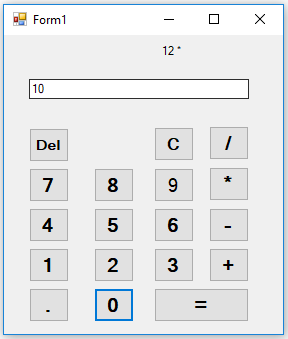
label1.Text = "";

}

}

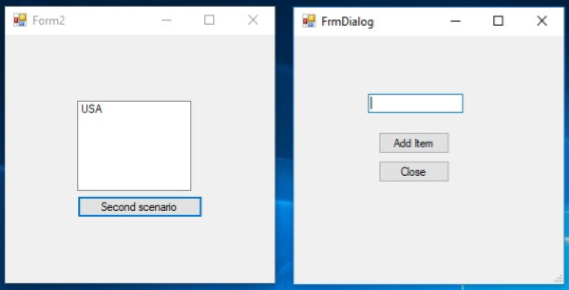
}

**Output :-**



**Program 5 :**

**AIM** **:-** Implement the following using Delegate.



**Program :-**

**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 WindowsFormsApplication\_5

{

public partial class Form1 : Form

{

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

{

}

public Form1()

{

InitializeComponent();

}

public void mydelegate\_fun(string s)

{

listBox1.Items.Add(s);

}

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

{

Form2 f = new Form2();

f.r=mydelegate\_fun;

f.Show();

f.Focus();

}

}

}

**Form2.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 WindowsFormsApplication\_5

{

public delegate void mydelegate(string s);

public partial class Form2 : Form

{

public mydelegate r = null;

public Form2()

{

InitializeComponent();

}

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

{

r(textBox1.Text);

}

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

{

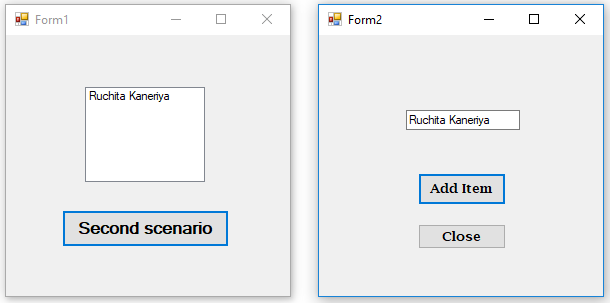
this.Close();

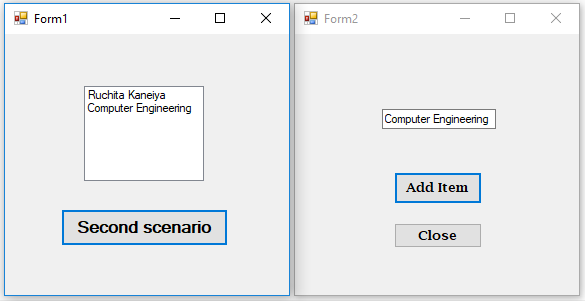
}

}

}

**Output :-**

****

****

**Program 6 :**

**AIM** **:-** create adows form which implements Menus (File, Color) and a Text Box. File menu item

has 4 options Font, Open, Save, Print(Print Preview and Print), Exit. Color menu opens a

Color Dialog box. Clicking on the colors sets the background of the form with respective

color. Open menu item opens a file, reads the contents and displays in the Text Box

while Save menu items saves the content of Text Box in to a file. Also provide a Context

Menu of your choice.

**Program :-**

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 WindowsFormsApplication6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

FontDialog fd = new FontDialog();

if (fd.ShowDialog() == DialogResult.OK)

{

textBox1.Font = fd.Font;

}

}

private void openToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "txt files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

ofd.CheckFileExists = true;

if (ofd.ShowDialog() == DialogResult.OK)

{

textBox1.Text = ofd.FileName;

}

}

private void saveToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "txt files(\*.txt)|\*.txt";

if (sfd.ShowDialog() == DialogResult.OK)

{

MessageBox.Show("File saved Successfully !!!");

}

}

private void printToolStripMenuItem1\_Click(object sender, EventArgs e)

{

PrintDialog pdlg = new PrintDialog();

pdlg.Document = printDocument11;

if (pdlg.ShowDialog() == DialogResult.OK)

{

printDocument11.Print();

}

}

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

{

PrintPreviewDialog ppdlg = new PrintPreviewDialog();

ppdlg.Document = printDocument11;

ppdlg.ShowDialog();

}

private void exitToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

this.Close();

}

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

{

ColorDialog cd = new ColorDialog();

cd.AllowFullOpen = true;

if (cd.ShowDialog() == DialogResult.OK)

{

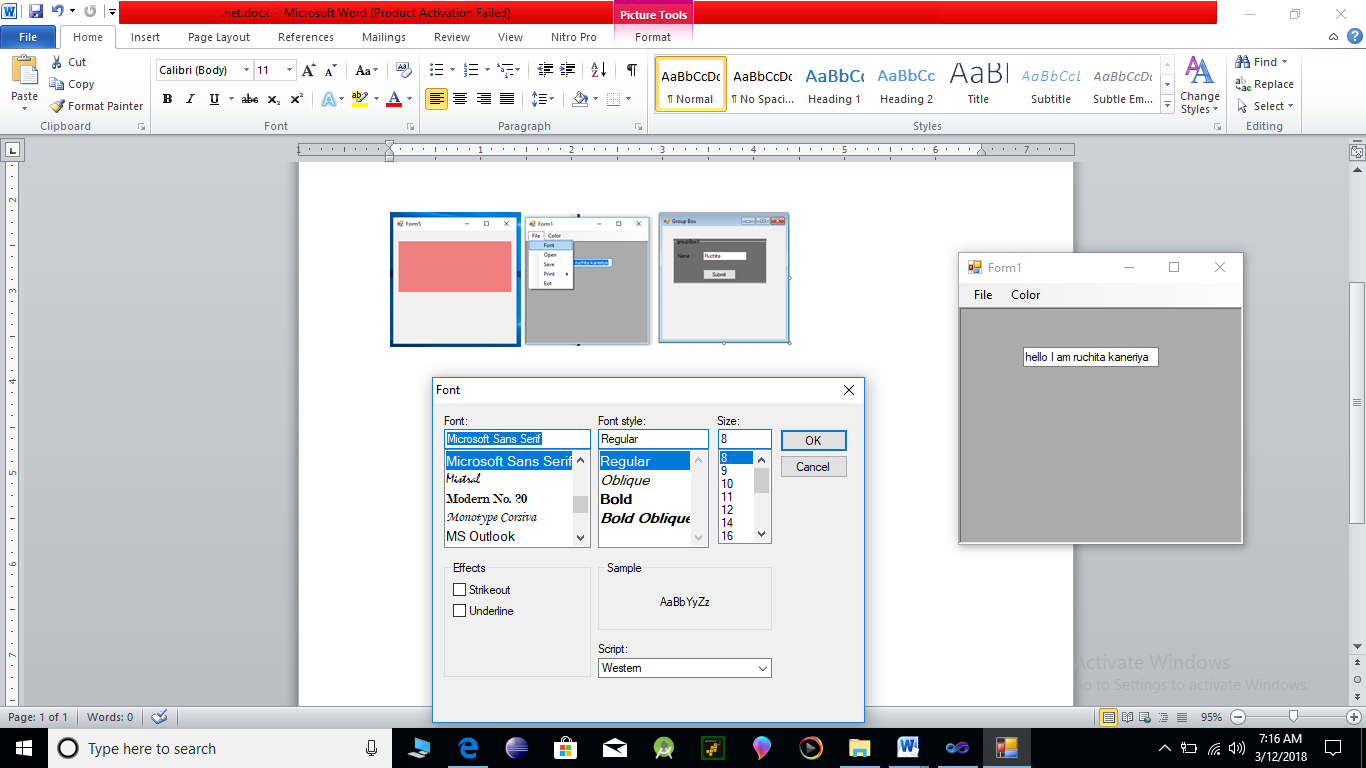
this.BackColor = cd.Color;

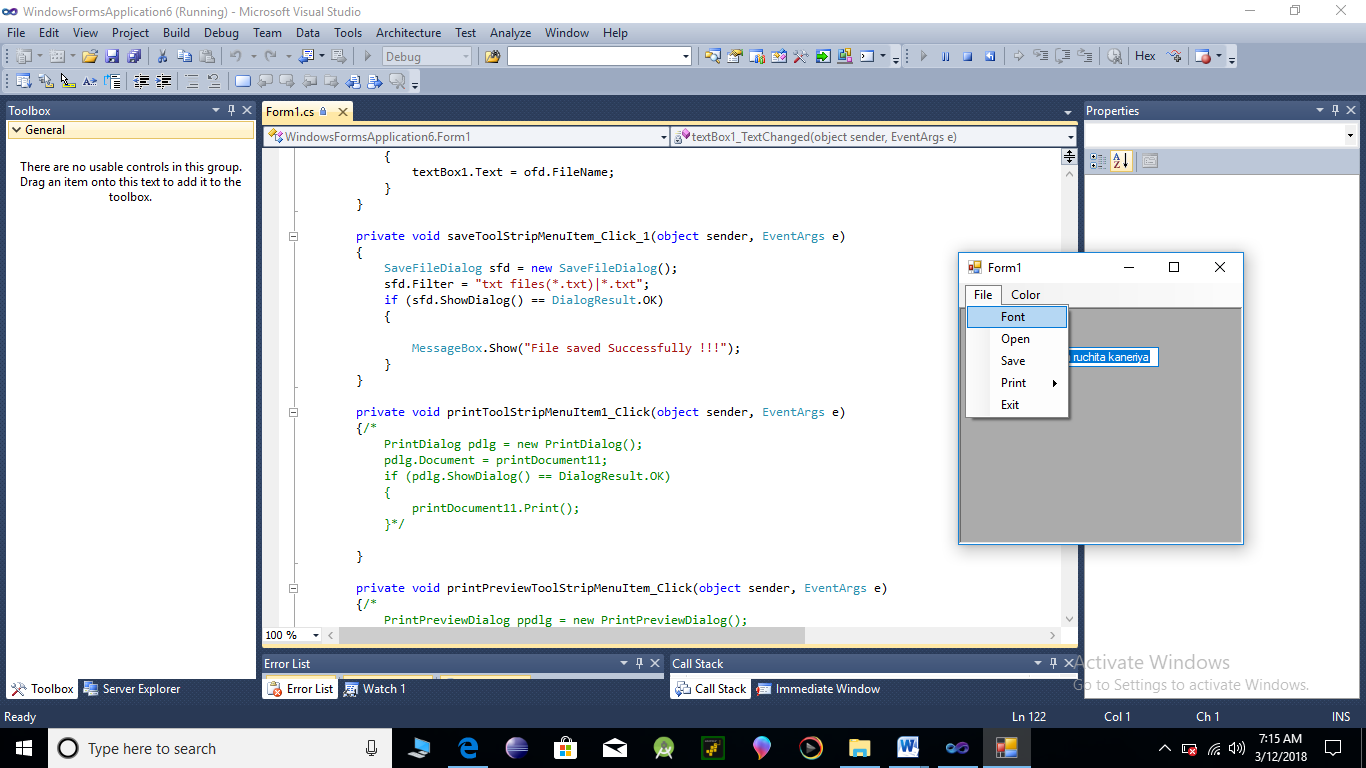
textBox1.BackColor = cd.Color;

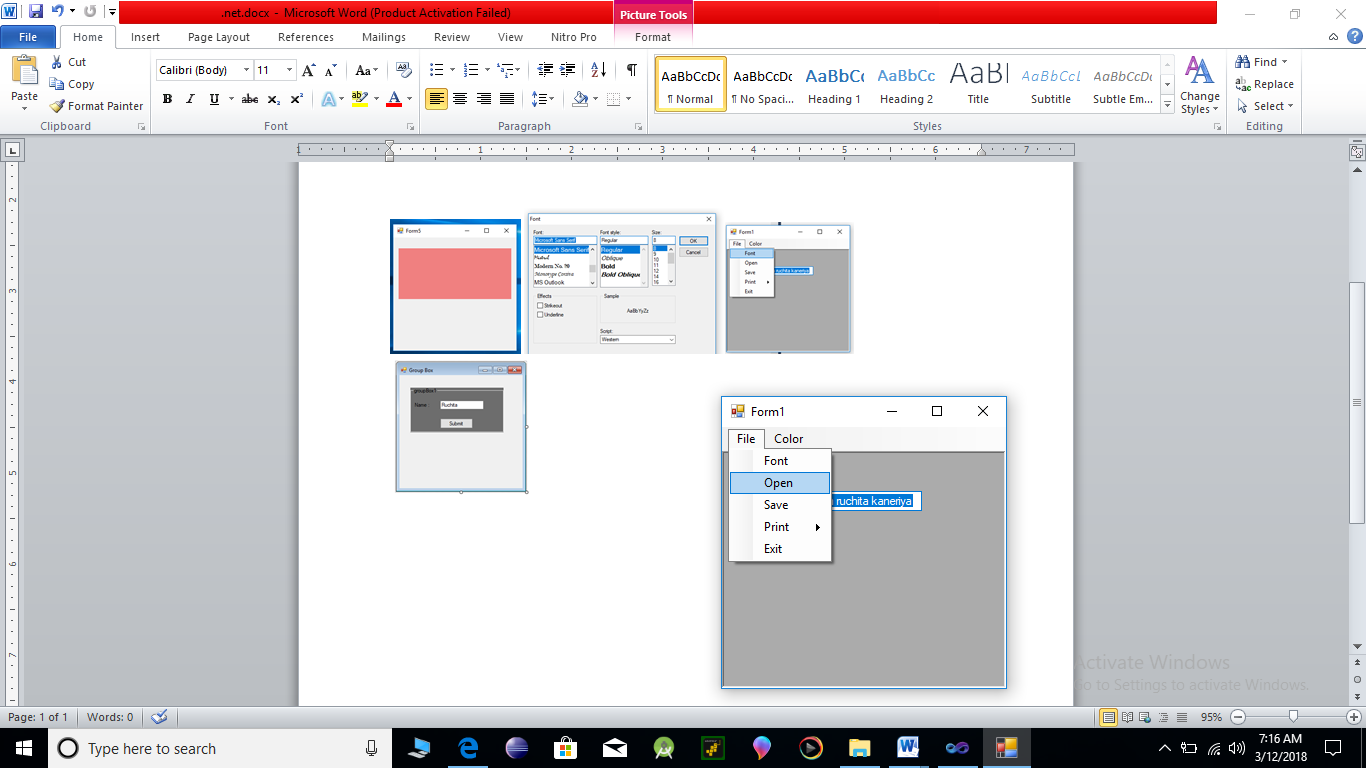
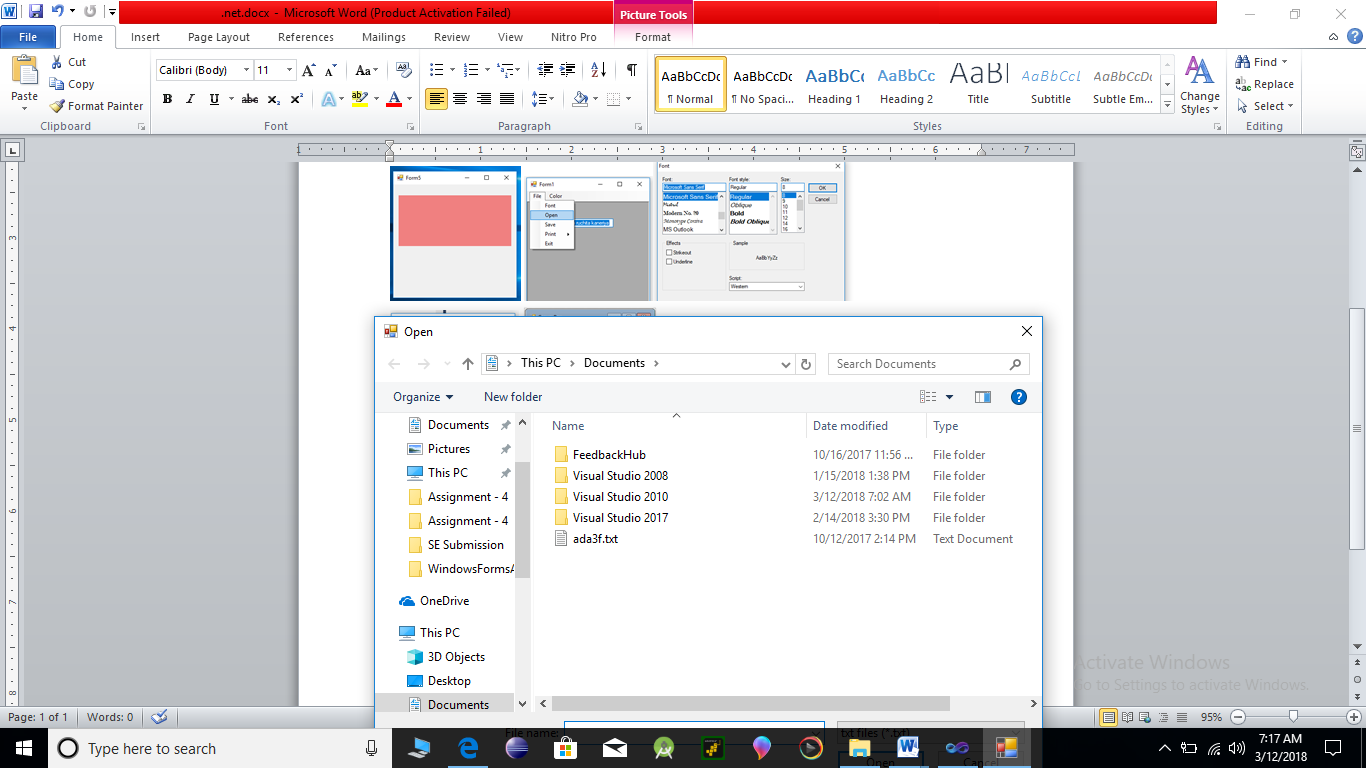
}

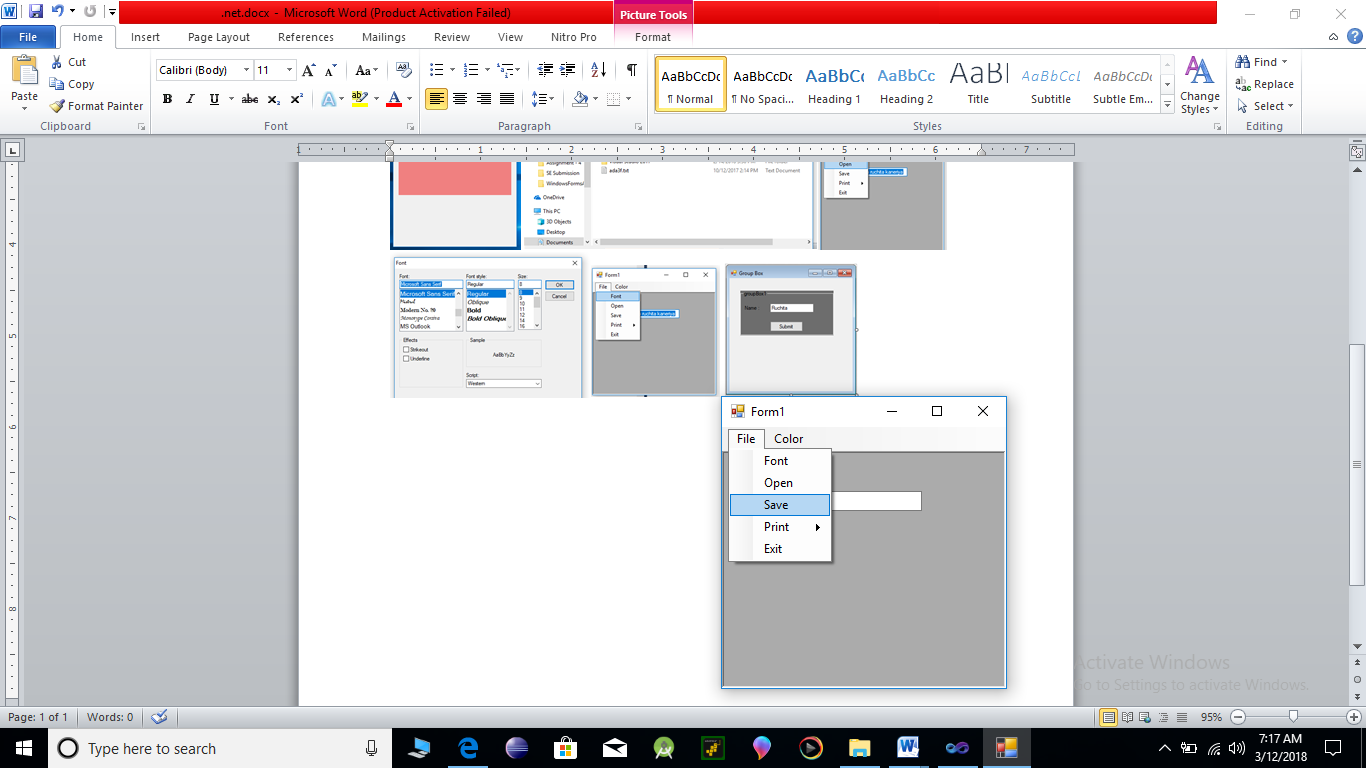
}

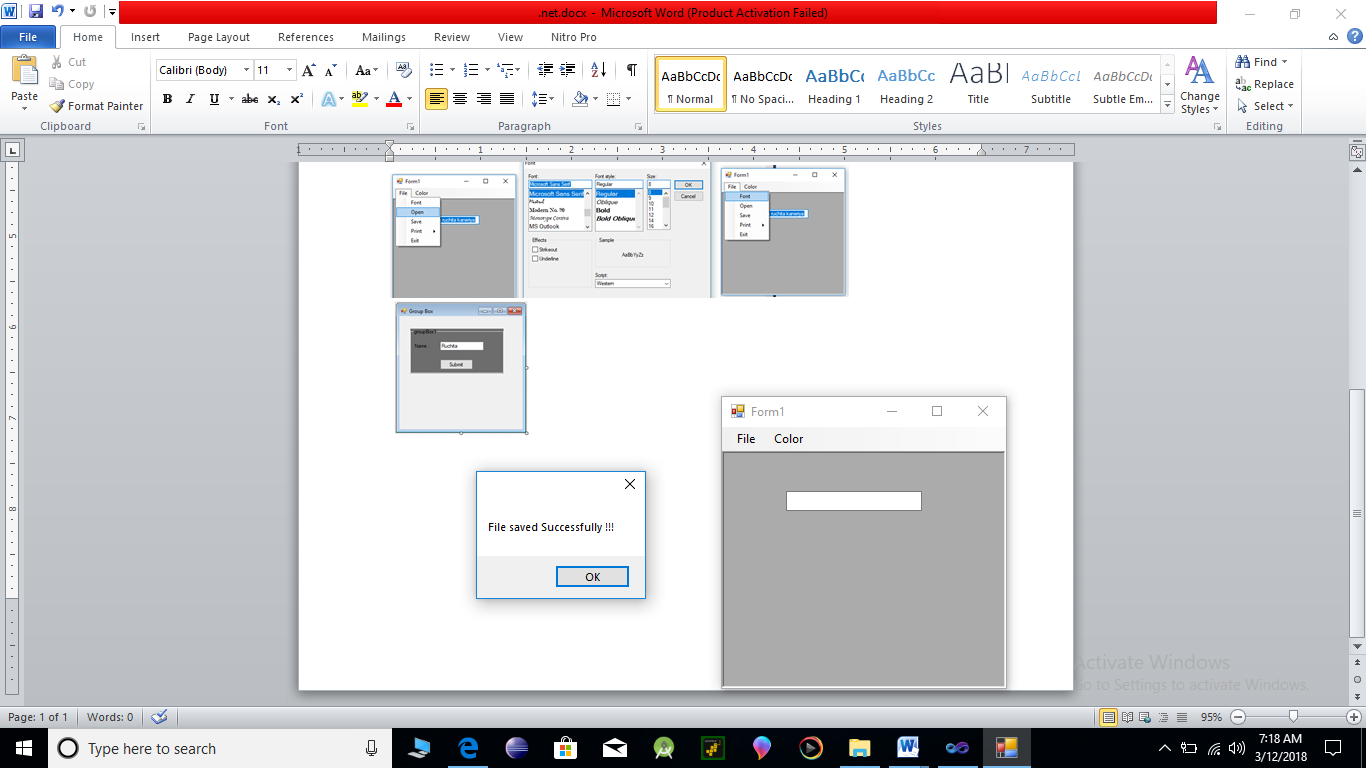
**Output :-**

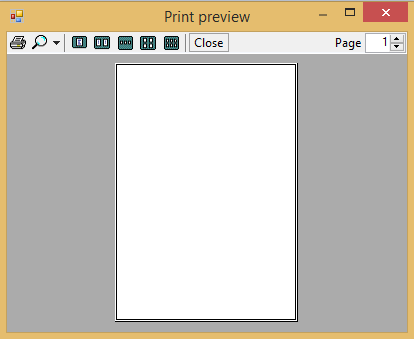


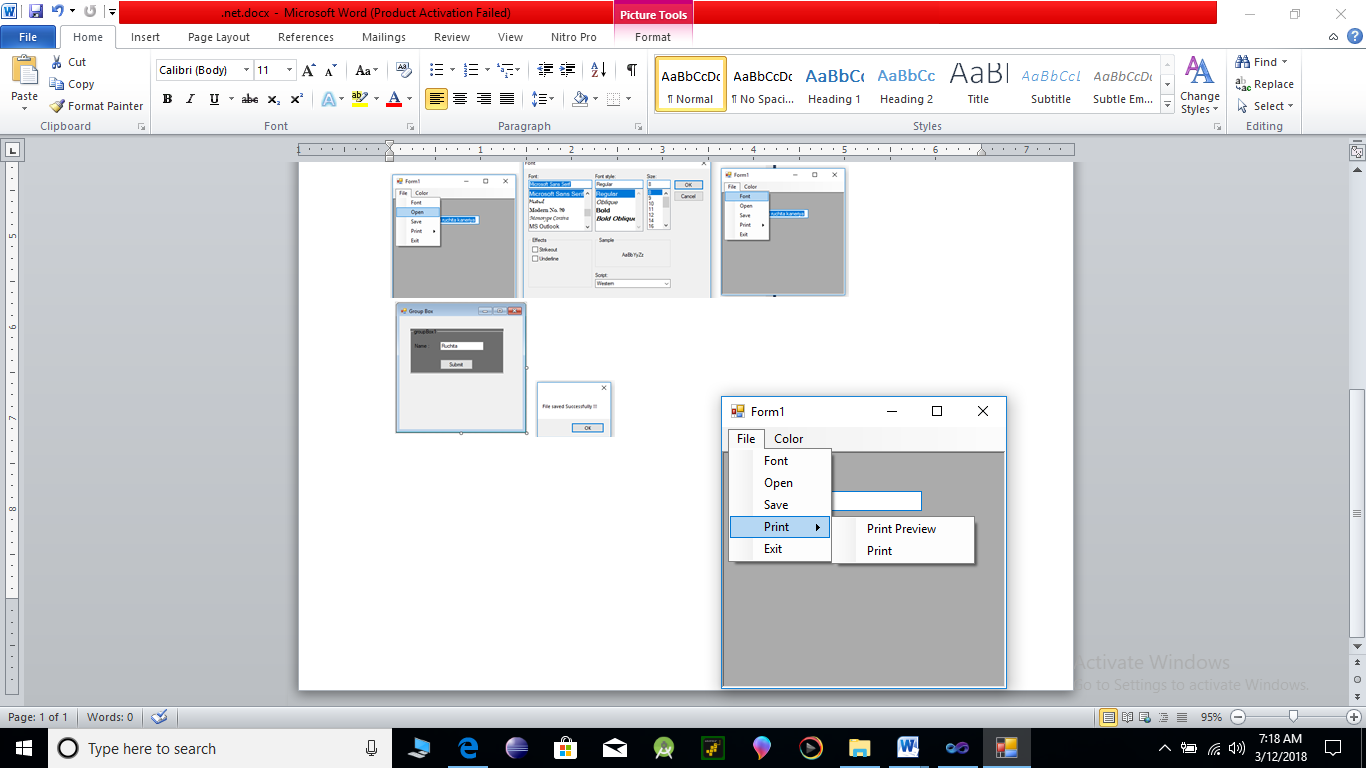


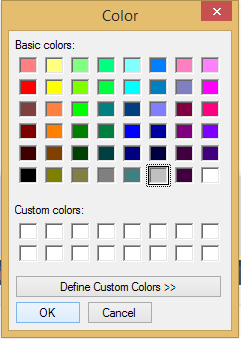


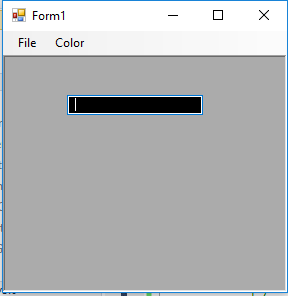










****

**Program 7 :**

**AIM** **:-** Implement an MDI form with Menu item “Add”, clicking on Add generates a new child

form(blank). And Menu item “Child Forms” which contains sub menu for implementing

Error Provider, Tool tip, Panel and Group Box controls each on a separate form.

**Program :-**

**Form 1 :**

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 WindowsFormsApplication7

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

Form2 f = new Form2();

f.Show();

}

}

}

**Form 2 :**

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 WindowsFormsApplication7

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

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

{

Form3 f3 = new Form3();

f3.Show();

}

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

{

Form4 f4 = new Form4();

f4.Show();

}

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

{

Form5 f5 = new Form5();

f5.Show();

}

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

{

Form6 f6 = new Form6();

f6.Show();

}

}

}

**Form 3 :**

sing System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Text.RegularExpressions;

namespace WindowsFormsApplication7

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void text\_name\_Validating(object sender, CancelEventArgs e)

{

if (text\_name.Text == string.Empty)

{

errorProvider1.SetError(text\_name, "Please Enter Name"); //Warning

}

else

{

errorProvider3.SetError(text\_name, "Correct");

}

}

private void text\_age\_Validated(object sender, EventArgs e)

{

if (text\_age.Text == string.Empty)

{

errorProvider1.SetError(text\_age, "Please Enter Age");

}

else

{

Regex numcheck = new Regex(@"^([0-9]\*|\d\*)$");

if (numcheck.IsMatch(text\_age.Text))

{

errorProvider3.SetError(text\_age, "Correct");

}

else

{

errorProvider2.SetError(text\_age, "Wrong Format");

}

}

}

}

}

**Form 4:**

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 WindowsFormsApplication7

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void Form4\_Load(object sender, EventArgs e)

{

ToolTip tt = new ToolTip();

tt.SetToolTip(button1, "Click me to execute");

}

}

}

**Form 5:**

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 WindowsFormsApplication7

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

}

}

**Form 6:**

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 WindowsFormsApplication7

{

public partial class Form6 : Form

{

public Form5()

{

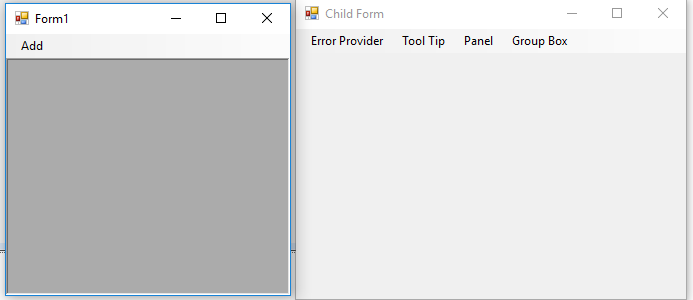
InitializeComponent();

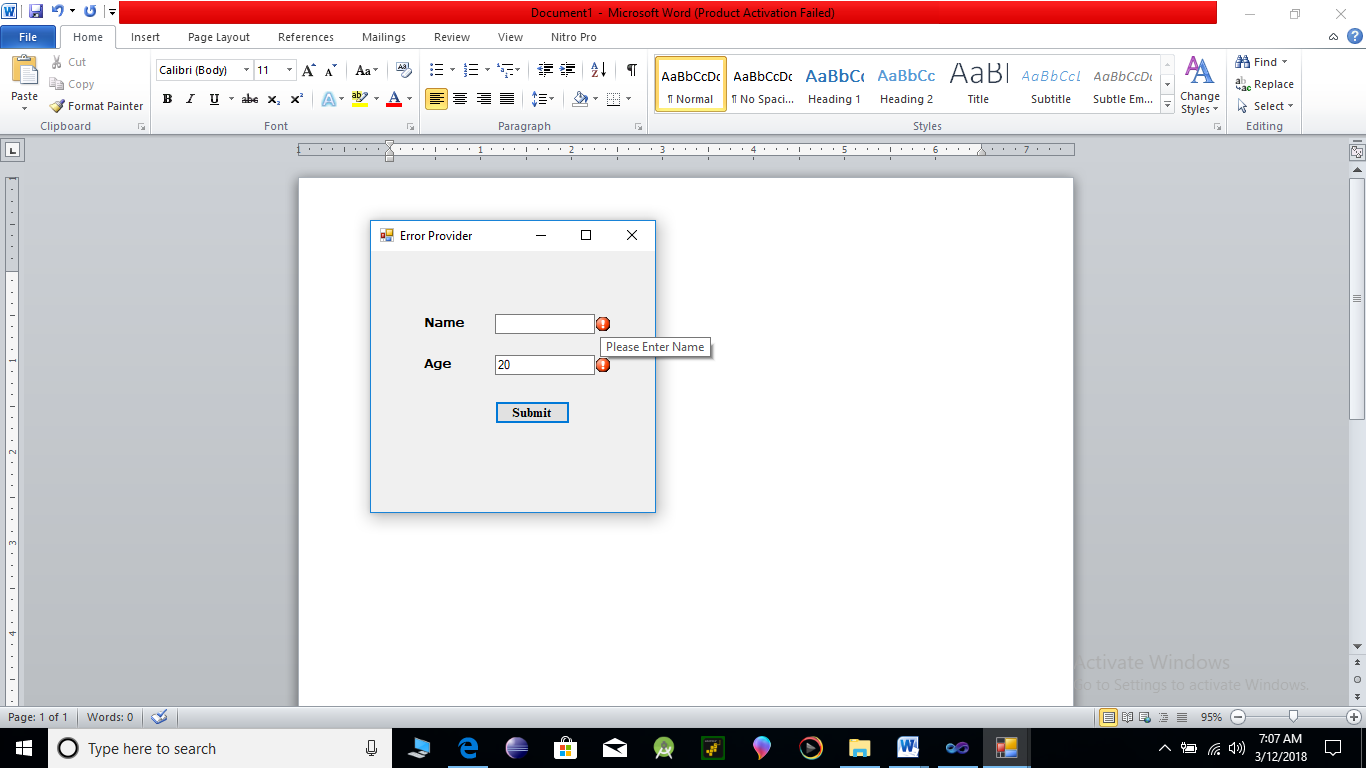
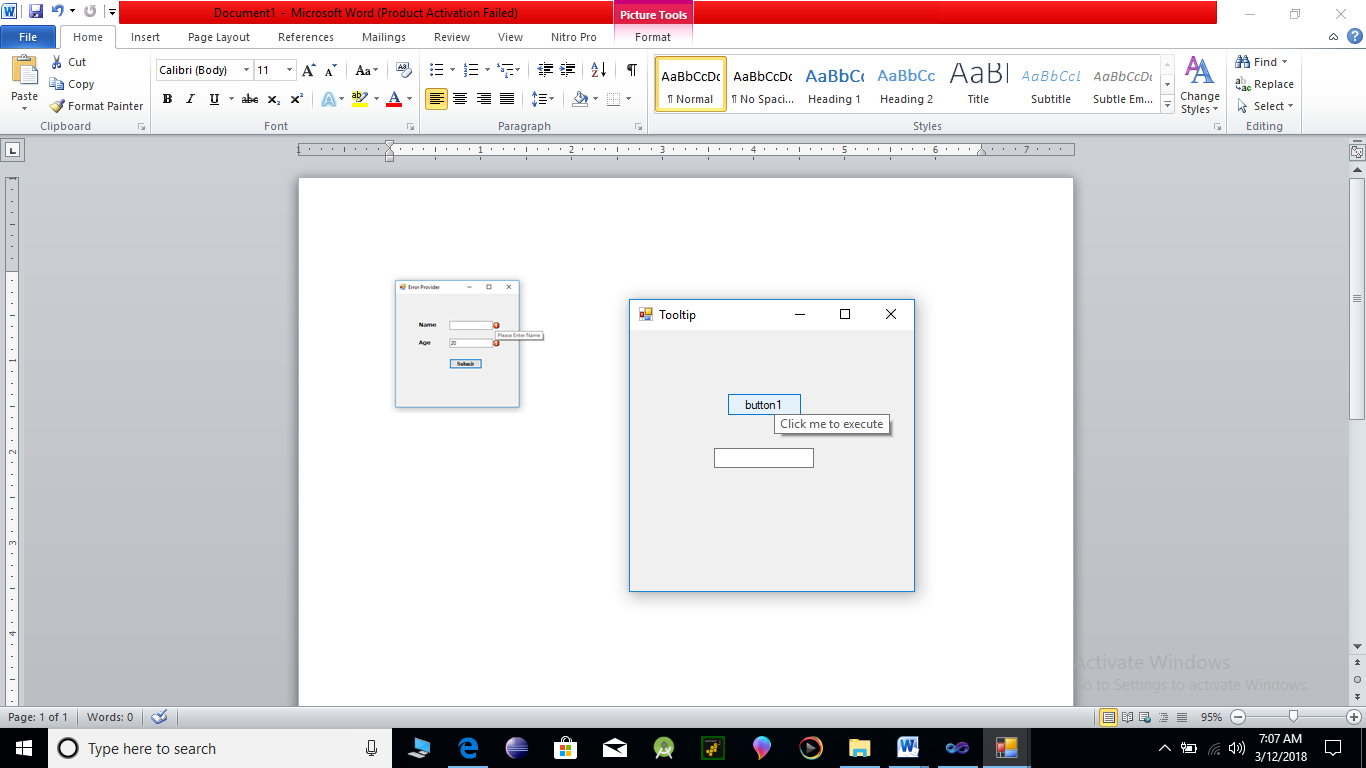
}

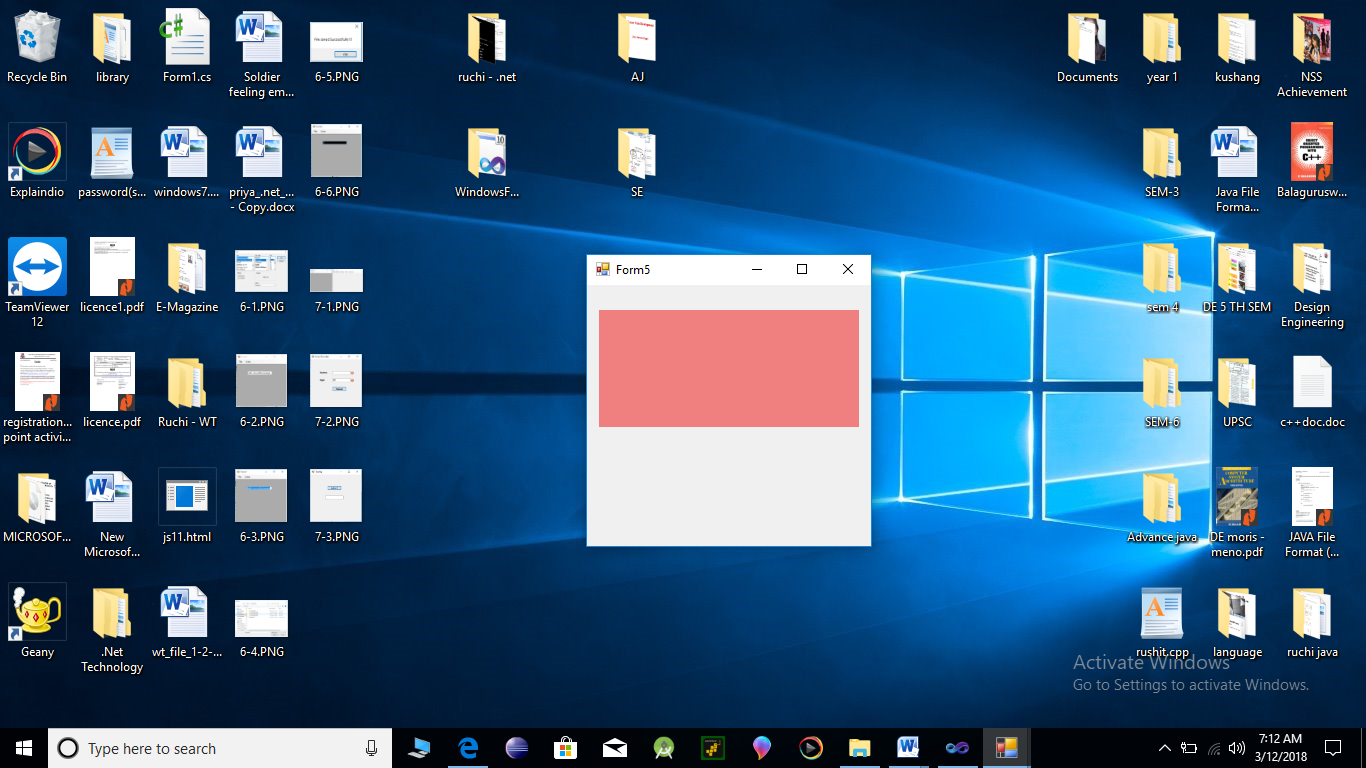
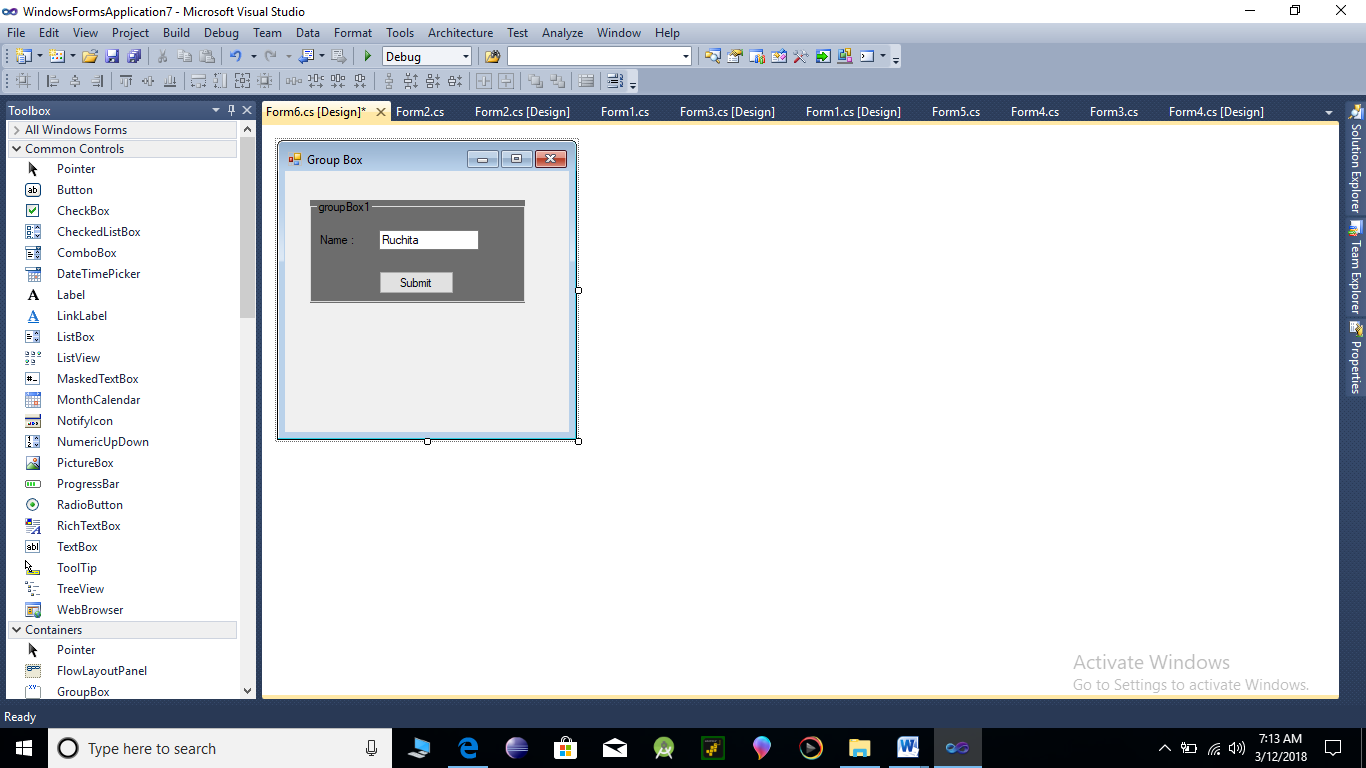
}

}

**Output :**

****





**PART -3 ADO,NET**

**Aim:** Implement the Windows Application.

**Program:**

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;

using System.Data.SqlClient;

namespace WindowsFormsApplication\_ado.net

{

public partial class Form1 : Form

{

SqlConnection cn;

SqlDataAdapter adp;

DataSet ds;

SqlCommand cmd;

SqlDataReader dr;

public Form1()

{

InitializeComponent();

}

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

{

cn = new SqlConnection();

cn.ConnectionString = @"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Ruchita\Documents\s15cos124.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True";

if(cn.State!=ConnectionState.Open)

{

cn.Open();

MessageBox.Show("After Connection Establishment");

}

BindComboBox();

bindgrid();

}

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

{

//BindComboBox();

//bindgrid();

}

public void BindComboBox()

{

adp = new SqlDataAdapter("select \* from Department",cn);

ds=new DataSet();

adp.Fill(ds);

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

{

MessageBox.Show("No Records Found");

}

{

dept.DataSource=ds.Tables[0];

dept.ValueMember=ds.Tables[0].Columns["Dept\_code"].ToString();

dept.DisplayMember=ds.Tables[0].Columns["Dept\_name"].ToString();

}

}

public void bindgrid()

{

adp.Dispose();

ds.Dispose();

adp = new SqlDataAdapter("select \* from Employee",cn);

ds = new DataSet();

adp.Fill(ds);

this.dataGridView1.DataSource = ds.Tables[0];

}

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

{

this.Close();

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

{

checkBox1.Text = "Active";

}

else

{

checkBox1.Text = "InActive";

}

}

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

{

int gender, status, count;

String hobbies = " ";

if (rb1\_male.Checked == true && rb2\_female.Checked == false)

{

gender = 1;

}

else

{

gender = 2;

}

if (checkBox1.Checked == true)

{

status = 0;

}

else

{

status = 1;

}

for (int i = 0; i <= checkedListBox1.CheckedItems.Count - 1; i++)

{

if (i == 0)

{

hobbies = checkedListBox1.CheckedItems[i].ToString();

}

else

{

hobbies += "," + checkedListBox1.CheckedItems[i].ToString();

}

}

if (this.code.Text != "")

{

cmd = new SqlCommand();

cmd.CommandText = "insert into Employee(Emp\_Code,Emp\_Name,Address,Emp\_Dept,Hobbies,Gender,Status) values (" +

"'" + code.Text + "','" + name.Text + "','" + address.Text + "','" + dept.SelectedValue + "','" + hobbies + "'," + gender + "," + status + ")";

cmd.Connection = cn;

cmd.CommandType = CommandType.Text;

count = cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("Record inserted successfully", "Employee", MessageBoxButtons.OK);

bindgrid();

Resetform();

}

else

{

MessageBox.Show("Error", "Employee", MessageBoxButtons.OK);

}

}

}

public void Resetform()

{

code.Text = "";

name.Text= "";

address.Text = "";

dept.SelectedIndex = 0;

checkBox1.Checked = false;

rb1\_male.Checked = false;

rb2\_female.Checked= false;

for (int i = 0; i < checkedListBox1.Items.Count; i++)

{

checkedListBox1.SetItemCheckState(i, CheckState.Unchecked);

}

}

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

{ }

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

{

string str, Emp\_code = "";

Emp\_code = code.Text;

Resetform();

str = "select \* from Employee where Employee Code='" + Emp\_code + "'";

cmd = new SqlCommand();

cmd.Connection = cn;

cmd.CommandType = CommandType.Text;

cmd.CommandText = str;

dr = cmd.ExecuteReader();

if (dr.Read())

{

code.Text = Convert.ToString(dr.GetValue(0));

name.Text = Convert.ToString(dr.GetValue(1));

dept.SelectedValue = Convert.ToString(dr.GetValue(4));

address.Text = Convert.ToString(dr.GetValue(3));

if (Convert.ToUInt16(dr.GetValue(6)) == 1)

{

rb1\_male.Checked = true;

}

else

{

rb2\_female.Checked= true;

}

if (Convert.ToInt16(dr.GetValue(7)) == 1)

{

checkBox1.Checked = true;

}

else

{

checkBox1.Checked = false;

}

string tmp = dr.GetValue(5).ToString();

string[] arr = tmp.Split(',');

for (int i = 0; i < checkedListBox1.Items.Count; i++)

{

for (int j = 0; j <= arr.Length - 1; j++)

{

if (arr[j] == checkedListBox1.Items[i].ToString())

{

checkedListBox1.SetItemCheckState(i, CheckState.Checked);

}

}

}

}

else

{

MessageBox.Show("Employee does not exist", "Employee", MessageBoxButtons.OK);

}

dr.Close();

}

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

{

cmd = new SqlCommand();

cmd.Connection = cn;

string Emp\_code = "";

Emp\_code = code.Text;

int gender,status;

string hobbies = "";

if (rb1\_male.Checked== true && rb2\_female.Checked == false)

{

gender = 1;

}

else

{

gender = 2;

}

if (checkBox1.Checked == true)

{

status = 0;

}

else

{

status = 1;

}

for (int i = 0; i <= checkedListBox1.CheckedItems.Count - 1; i++)

{

if (i == 0)

{

hobbies = checkedListBox1.CheckedItems[i].ToString();

}

else

{

hobbies += "," + checkedListBox1.CheckedItems[i].ToString();

}

}

cmd.CommandText="update Employee SET Name='"+name.Text+"',Department='"+dept.SelectedValue.ToString()+"',Gender="+gender+",Status="+status+",Address='"+address.Text+"',Hobbies='"+hobbies+"' where Employee\_Code='" + Emp\_code + "'";

cmd.CommandType = CommandType.Text;

DialogResult dlr = new DialogResult();

dlr = MessageBox.Show("Are you sure that you want to update record ? ", "Employee", MessageBoxButtons.YesNo);

if (dlr == DialogResult.Yes)

{

int count = cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("record updated successfully", "Employee", MessageBoxButtons.OK);

Resetform();

bindgrid();

}

}

}

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

{

string Emp\_Code = "";

Emp\_Code = code.Text;

cmd = new SqlCommand();

cmd.Connection = cn;

cmd.CommandText = "delete from EMPLOYEE where Employee\_Code='" + Emp\_Code + "'";

cmd.CommandType = CommandType.Text;

DialogResult dlr = new DialogResult();

dlr =MessageBox.Show("Are you sure you want to delete record?", "Employee", MessageBoxButtons.YesNo);

if (dlr == DialogResult.Yes)

{

int count = cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("record deleted successfully", "Employee", MessageBoxButtons.OK);

Resetform();

bindgrid();

}

}

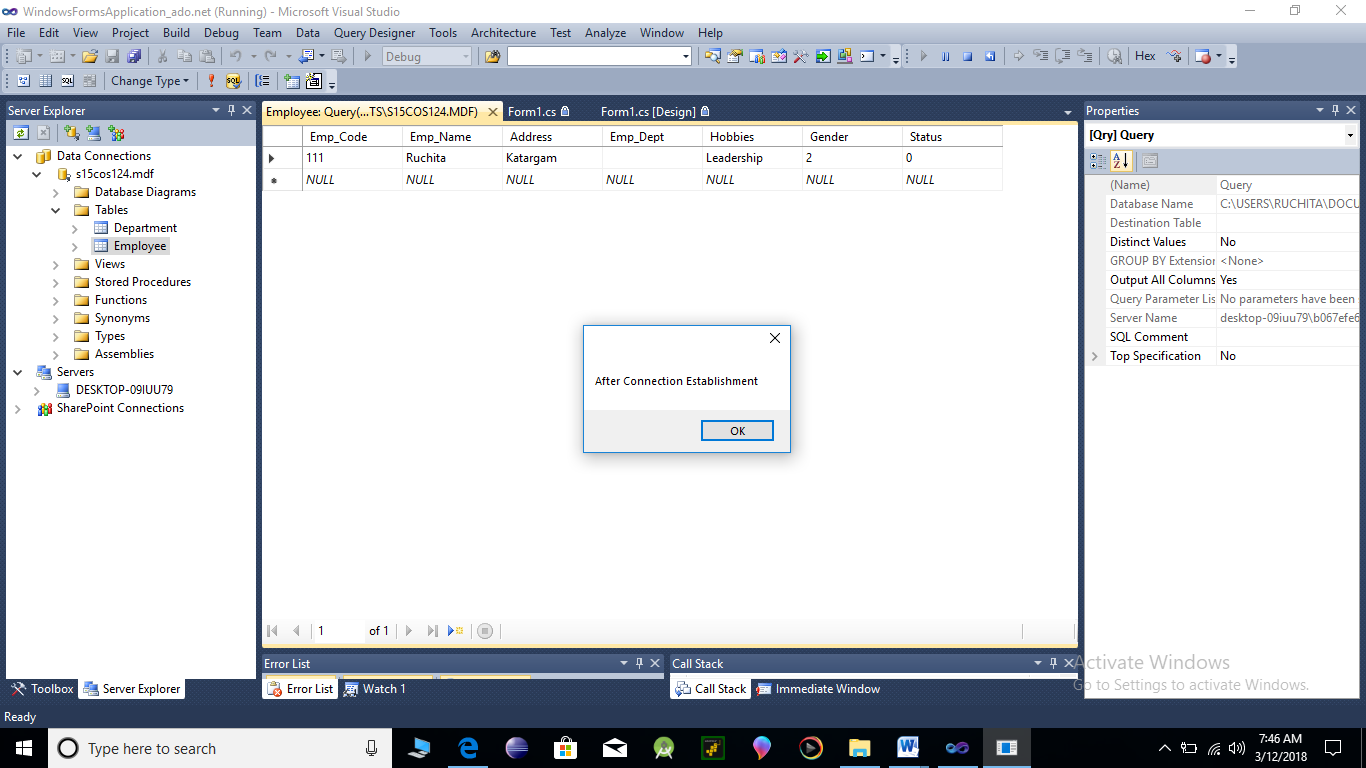
}

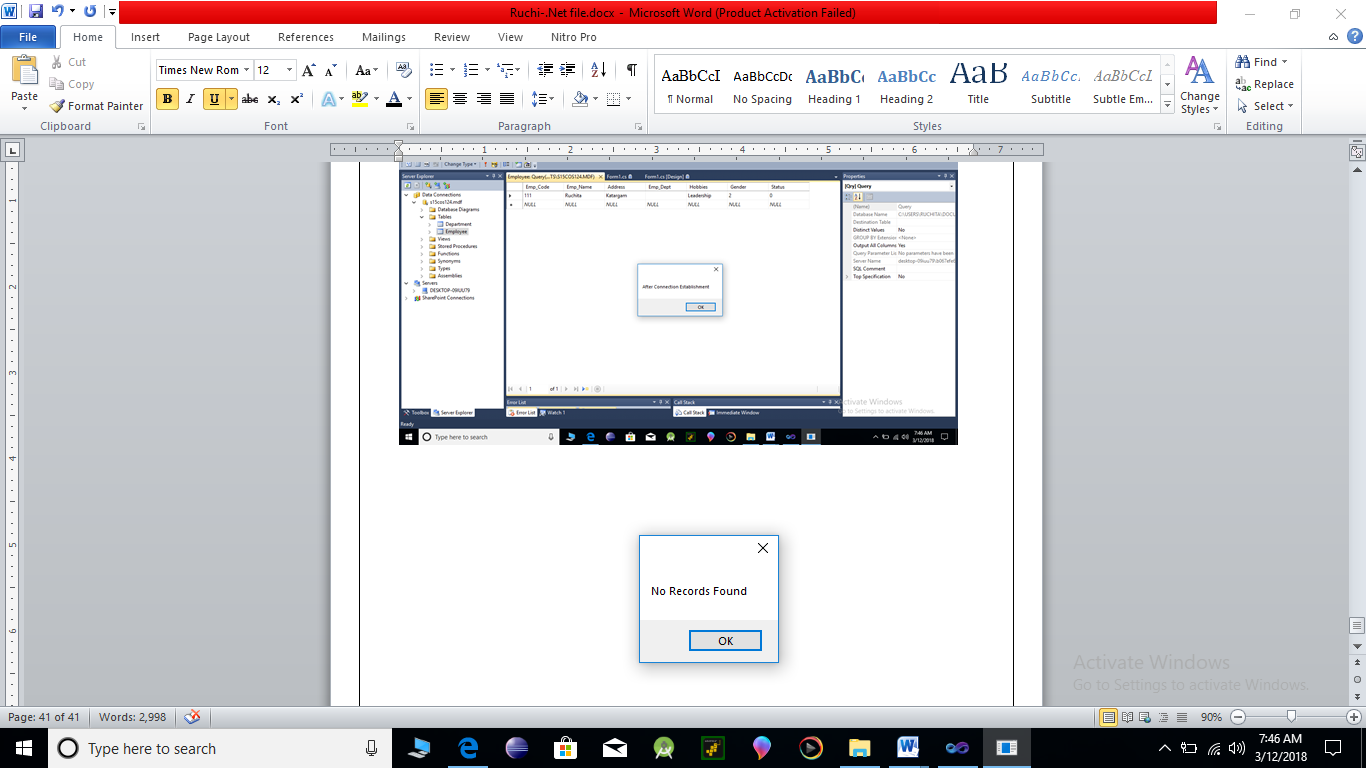
}

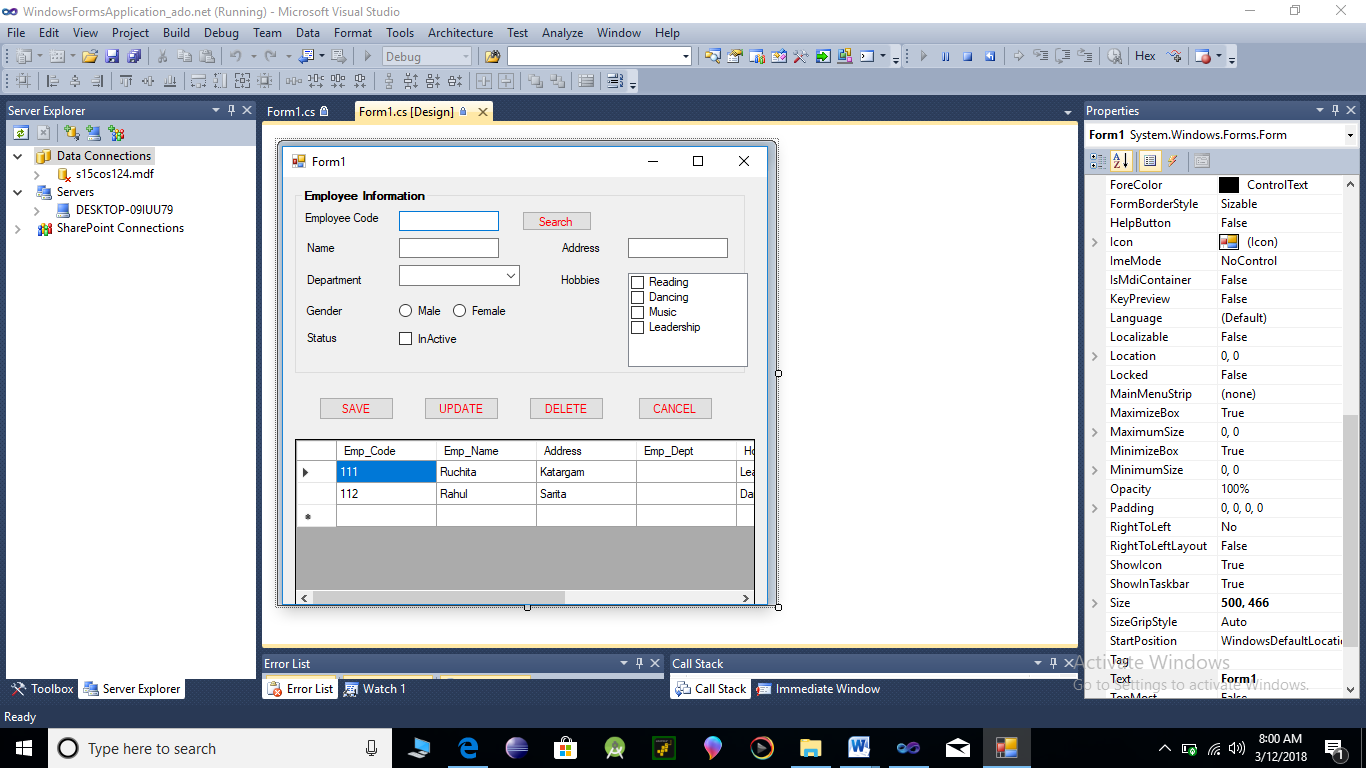
}

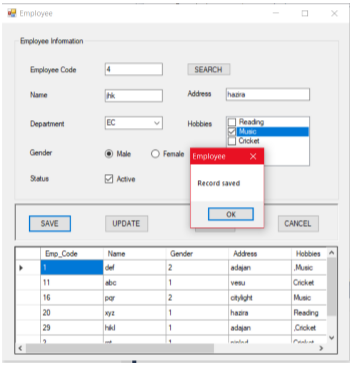
**Output:**

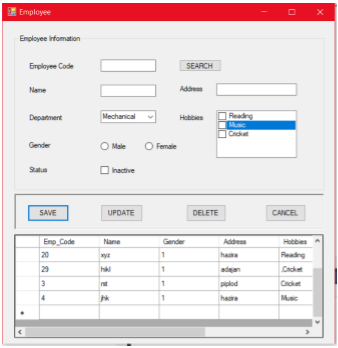
After Connection Establishment :





After Inserted Record:

**Save:**



**Update :**

