Name: Rutvik Redkar

Roll No: 90

Assignment 2

1. Write a program to calculate simple interest in windows applications. Create a class and methods that perform the calculation.

Aim: To write a program to calculate simple interest

Objective: Calculate simple interest in windows applications

Theory:

Window Application:

A Windows forms application is one that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc.

In the Solution Explorer, you will also be able to see the DemoApplication Solution. This solution will contain the below 2 project files

1. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application.
2. The Main program called Program.cs is the default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole.

On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms.

Label Control

Next comes the Label Control. The label control is used to display a text or a message to the user on the form. The label control is normally used along with other controls. Common examples are wherein a label is added along with the textbox control.

The label indicates to the user what is expected to fill up in the textbox. Let’s see how we can implement this with an example shown below. We will add 2 labels, one which will be called ‘name’ and the other called ‘address.’ They will be used in conjunction with the textbox controls which will be added in the later section.

Textbox

A textbox is used for allowing a user to enter some text on the Windows application in C#. Let’s see how we can implement this with an example shown below. We will add 2 textboxes to the form, one for the Name and the other for the address to be entered for the user

Button

A button is used to allow the user to click on a button which would then start the processing of the form. Let's see how we can implement this with an example shown below. We will add a simple button called 'Submit' which will be used to submit all the information on the form.

Code:

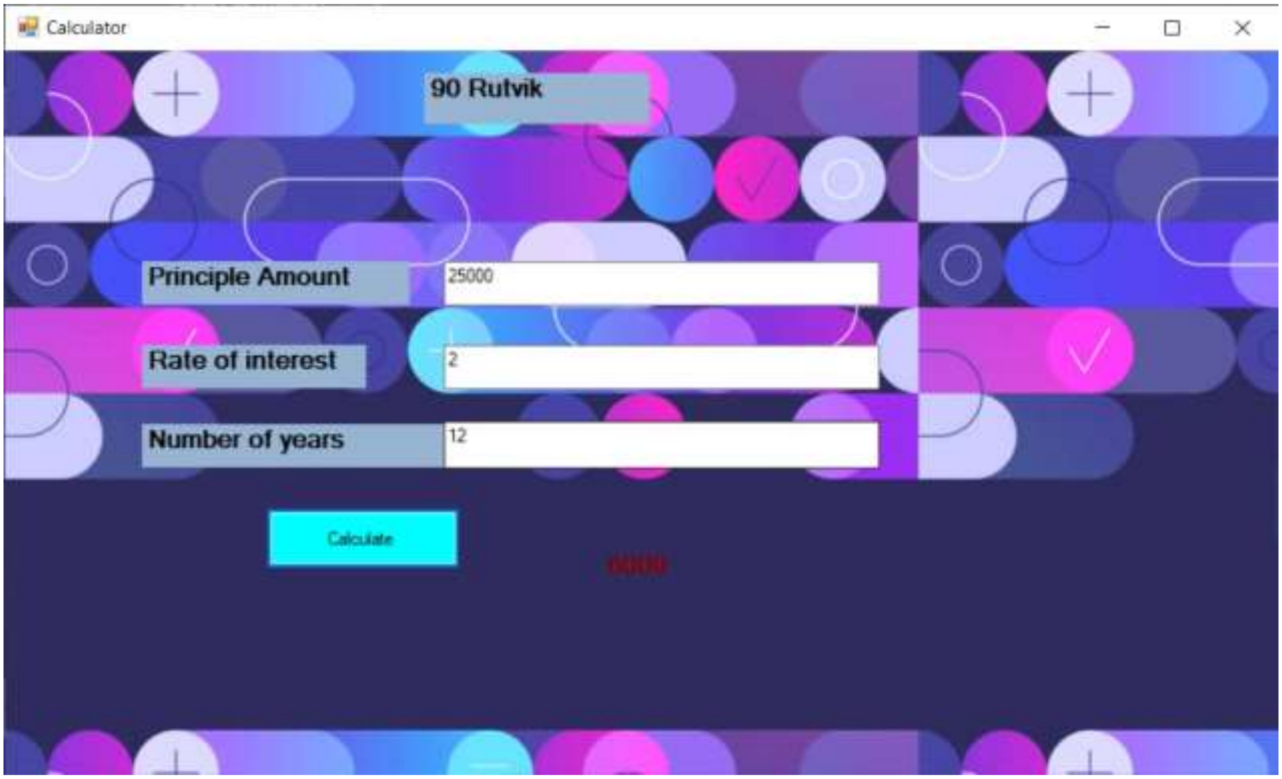
Operation.cs

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

namespace WinFormsApp1
{
     class operation
    {
        public float cal(float p,float r, float n)
        {
            float calsi;
            calsi = (p * r * n) / 100;
            return calsi;
        }

    }
}
```

Form1.cs

```
namespace WinFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            operation op = new operation();
            float result;
            result = op.cal(float.Parse(txtprincipal.Text), float.Parse(txtrate.Text),
float.Parse(txtyear.Text));
            lblcal.Text = result.ToString();
        }
    }
}
```

Output:



Calculator

90 Rutvik

Principle Amount 25000

Rate of interest 2

Number of years 12

[illegible]

2.Write a program to design a form that contains student information (Id, Name, Percentage, Gender). By clicking on save it will save the data and by clicking on view data is displayed in the list box. Create a Student class which has different properties.

Aim: To write a program to generate a form to save and view data

Objective: design a form that contains student information and display in list box

Theory:

window Application:

A Windows forms application is one that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc.

In the Solution Explorer, you will also be able to see the Demo Application Solution. This solution will contain the below 2 project files

1. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application.
2. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole.

On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms.

Label Control:

Next comes the Label Control. The label control is used to display a text or a message to the user on the form. The label control is normally used along with other controls. Common examples are wherein a label is added along with the textbox control.

The label indicates to the user on what is expected to fill up in the textbox. Let's see how we can implement this with an example shown below. We will add 2 labels, one which will be called 'name' and the other called 'address.' They will be used in conjunction with the textbox controls which will be added in the later section.

Textbox:

A textbox is used for allowing a user to enter some text on the Windows application in C#. Let's see how we can implement this with an example shown below. We will add 2 textboxes to the form, one for the Name and the other for the address to be entered for the user.

Button:

A button is used to allow the user to click on a button which would then start the processing of the form. Let's see how we can implement this with an example shown below. We will add a simple button called 'Submit' which will be used to submit all the information on the form.

Radio Button:

A Radiobutton is used to showcase a list of items out of which the user can choose one. Let's see how we can implement this with an example shown below. We will add a radio button for a male/female option.

List box:

A Listbox is used to showcase a list of items on the Windows form. Let's see how we can implement this with an example shown below. We will add a list box to the form to store some city locations.

POJO:

POJO in Java stands for Plain Old Java Object. It is an ordinary object, which is not bound by any special restriction. The POJO file does not require any special classpath. It increases the readability & re-usability of a Java program.

POJOs are now widely accepted due to their easy maintenance. They are easy to read and write. A POJO class does not have any naming convention for properties and methods. It is not tied to any Java Framework; any Java Program can use it.

The term POJO was introduced by Martin Fowler (An American software developer) in 2000. it is available in Java from the EJB 3.0 by sun microsystem.

Generally, a POJO class contains variables and their Getters and Setters.

Properties of POJO class

Below are some properties of the POJO class:

- The POJO class must be public.
- It must have a public default constructor.
- It may have the arguments constructor.
- All objects must have some public Getters and Setters to access the object values by other Java Programs.
- The object in the POJO Class can have any access modifies such as private, public, protected. But, all instance variables should be private for improved security of the project.
- A POJO class should not extend predefined classes.
- It should not implement prespecified interfaces.
- It should not have any prespecified annotation.

Code:

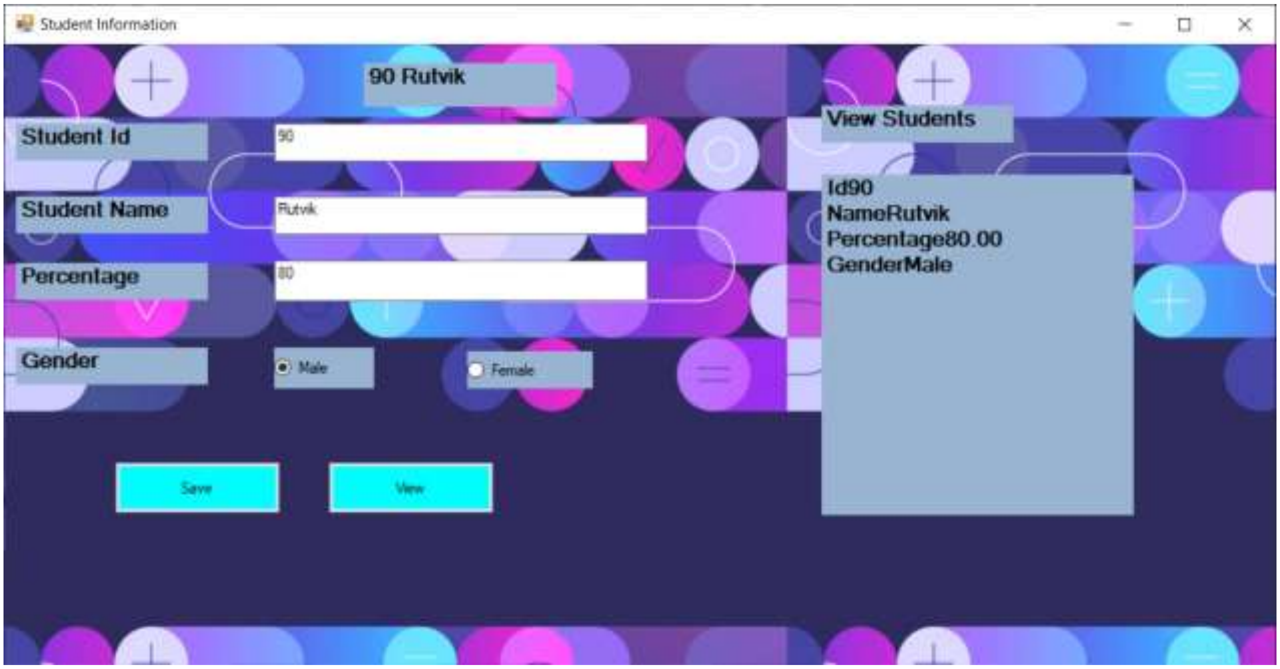
Student.cs

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

namespace WindowsFormsAppStudent
{
    internal class student
    {
        private int id;
        private string name;
        private double percentage;
        private string gender;


        public int Id { get => id; set => id = value; }
        public string Name { get => name; set => name = value; }
        public double Percentage { get => percentage; set => percentage = value; }
        public string Gender { get => gender; set => gender = value; }
    }
}
```

Form1.cs

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

namespace WindowsFormsAppStudent
{
    public partial class StudentDetail : Form
    {
        student s = new student();
        public StudentDetail()
        {
            InitializeComponent();
        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {

            if(radioButton1.Checked)
            {
                s.Gender = radioButton1.Text;
            }
            else if(radioButton2.Checked)
            {
                s.Gender = radioButton2.Text;
            }
            s.Id = Int32.Parse(textBox1.Text);
            s.Name = textBox3.Text;
            s.Percentage = double.Parse(textBox2.Text);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            radioButton1.Checked = false;
            radioButton2.Checked = false;

            listBox1.Items.Add("Id" + s.Id);
            listBox1.Items.Add("Name" + s.Name);
            listBox1.Items.Add("Percentage" + s.Percentage);
            listBox1.Items.Add("Gender" + s.Gender);
        }
    }
}
```

Output:



Student Information

90 Rutvik

Student Id: 90

Student Name: Rutvik

Percentage: 80

Gender: ● Male ○ Female

View Students

```
Id90
NameRutvik
Percentage80.00
GenderMale
```

3. Write a program to design and perform following operations.
A. Create a login form
B. Once a user login to the application, it will redirect to the student information form which takes the following information.
C. After clicking on the View button it displays all the information in another form

Aim: To create 3 login form and display data

Objective: create login page and after login redirect to another page

Theory:

window Application:

A Windows forms application is one that runs on the desktop computer. A Windows forms application will normally have a collection of controls such as labels, textboxes, list boxes, etc.

In the Solution Explorer, you will also be able to see the Demo Application Solution. This solution will contain the below 2 project files

1. A Form application called Forms1.cs. This file will contain all of the code for the Windows Form application.
2. The Main program called Program.cs is default code file which is created when a new application is created in Visual Studio. This code will contain the startup code for the application as a whole.

On the left-hand side of Visual Studio, you will also see a ToolBox. The toolbox contains all the controls which can be added to a Windows Forms. Controls like a text box or a label are just some of the controls which can be added to a Windows Forms.

Label Control:

Next comes the Label Control. The label control is used to display a text or a message to the user on the form. The label control is normally used along with other controls. Common examples are wherein a label is added along with the textbox control.
The label indicates to the user on what is expected to fill up in the textbox. Let's see how we can implement this with an example shown below. We will add 2 labels, one which will be called 'name' and the other called 'address.' They will be used in conjunction with the textbox controls which will be added in the later section.

Textbox:

A textbox is used for allowing a user to enter some text on the Windows application in C#. Let's see how we can implement this with an example shown below. We will add 2 textboxes to the form, one for the Name and the other for the address to be entered for the user.

Button:

A button is used to allow the user to click on a button which would then start the processing of the form. Let's see how we can implement this with an example shown below. We will add a simple button called 'Submit' which will be used to submit all the information on the form.

Radio Button:

A Radiobutton is used to showcase a list of items out of which the user can choose one. Let's see how we can implement this with an example shown below. We will add a radio button for a male/female option.

List box:

A Listbox is used to showcase a list of items on the Windows form. Let’s see how we can implement this with an example shown below. We will add a list box to the form to store some city locations.

Code:

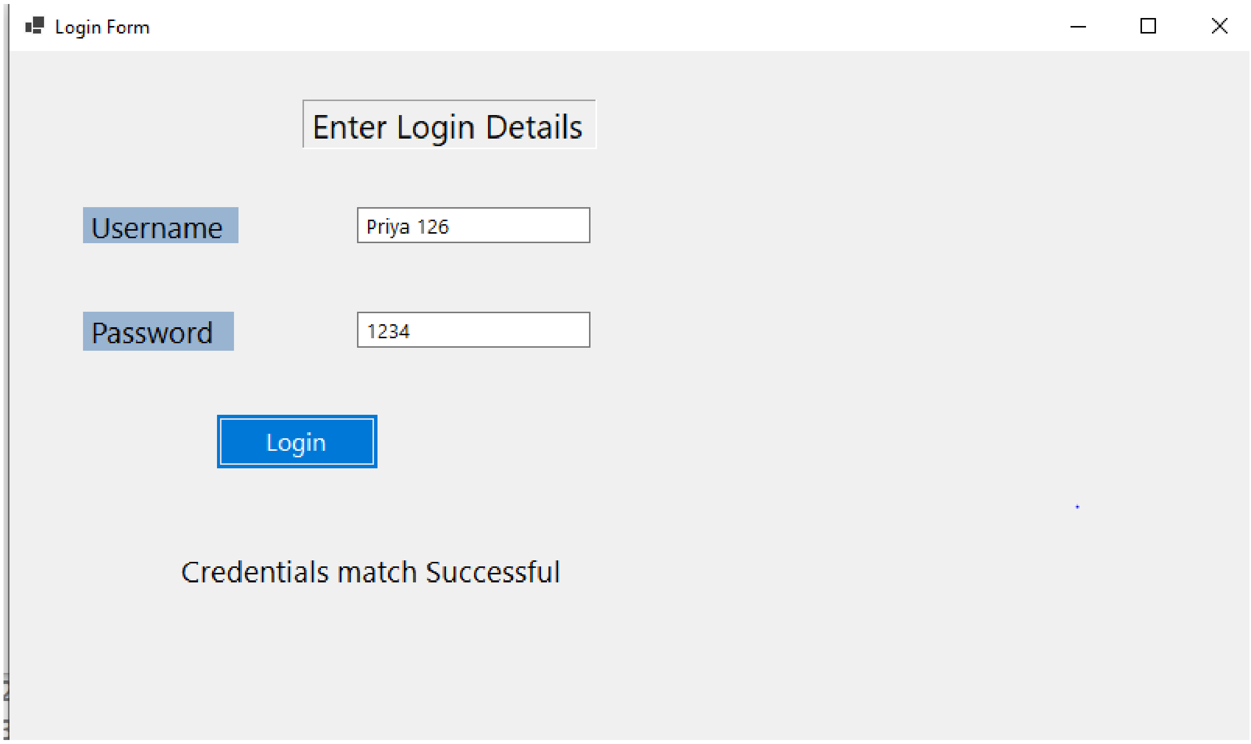
```
Form 1 .cs
namespace Login_Application
{
    public partial class Form1 : Form
    {
        Student_Inf sf = new Student_Inf();
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "Shubham" && textBox2.Text == "123")
            {
                label3.Text = "Credentials match Successful";

                sf.Show();
            }

            else { label3.Text = "Wrong Credentials"; }
        }
    }
}

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

namespace Login_Application
{
    public partial class view : Form
    {
        public view()
        {
            InitializeComponent();
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void view_Load(object sender, EventArgs e)
        {
            Student_Inf std = new Student_Inf();
            label2.Text = "Roll no: " + Student_Inf.roll_no;
            label3.Text = "Name : " + Student_Inf.name;
            label4.Text = "Dob : " + Student_Inf.dob;
            label5.Text = "hobbies: " + Student_Inf.hobbies;
            label6.Text = "Gender : " + Student_Inf.gender;
            label7.Text = "city: " + Student_Inf.city;
        }
    }
}
```

Studentinf.cs

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

namespace Login_Application
{
    public partial class Student_Inf : Form
    {
        public static string name, gender, hobbies, city;
        public static DateTime dob;
        public static int roll_no;
        public Student_Inf()
        {
            InitializeComponent();
        }

        private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (radioButton1.Checked)
            { gender = radioButton1.Text; }
            else if (radioButton2.Checked)
            { gender = radioButton2.Text; }

            roll_no = Int32.Parse(textBox1.Text);
            name = textBox2.Text;
            dob = dateTimePicker1.Value.Date;

            if (checkBox1.Checked) { hobbies = hobbies + checkBox1.Text; }
            else if (checkBox2.Checked) { hobbies = hobbies + checkBox2.Text; }
            else if (checkBox3.Checked) { hobbies = hobbies + checkBox3.Text; }
            city = comboBox1.SelectedItem.ToString();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            view v = new view();
            v.Show();
            this.Close();
        }
    }
}
```

Output:

Login Form:

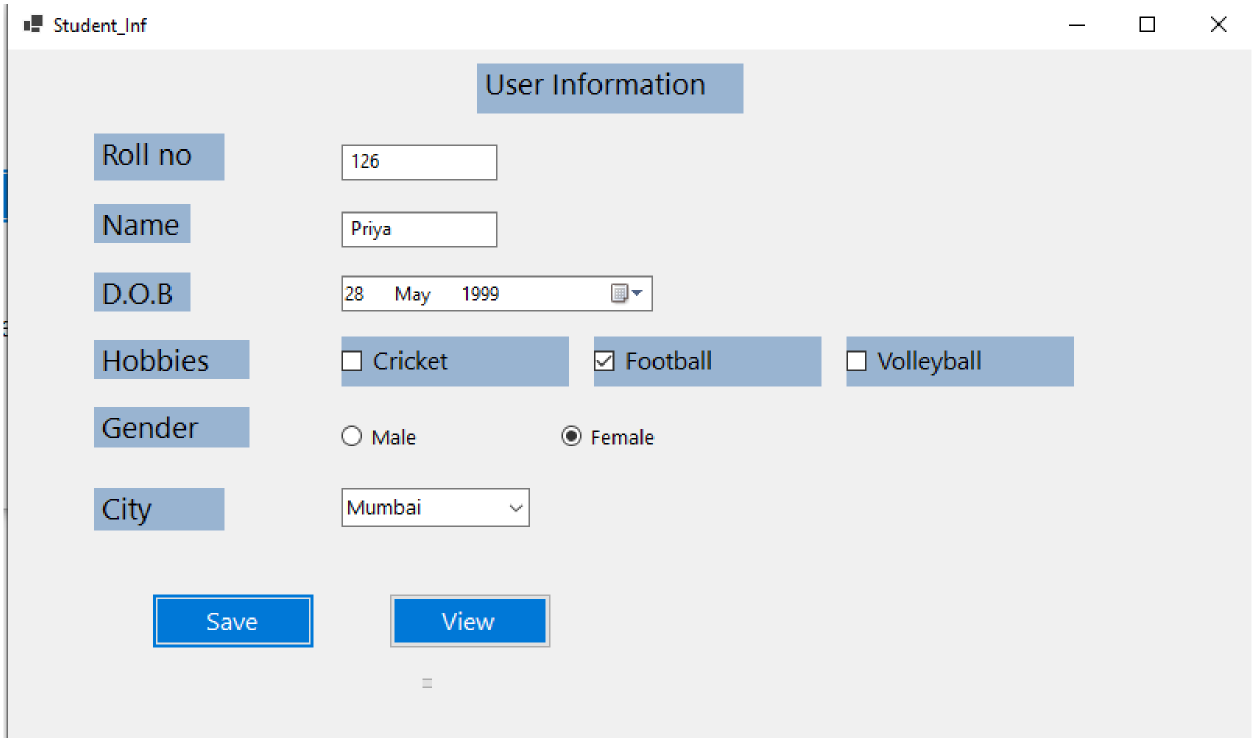


Username: Priya 126

Password: 1234

Credentials match Successful

Student_Inf:



Student_Inf

User Information

Roll no 126

Name Priya

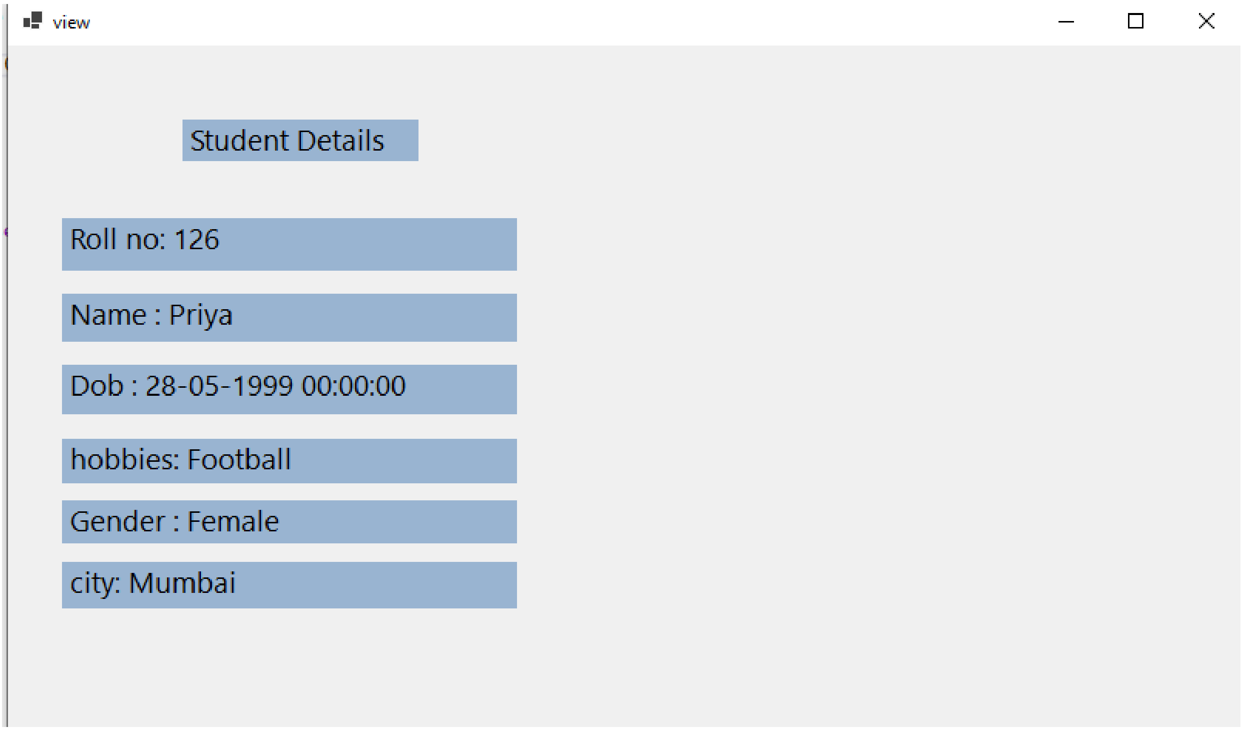
D.O.B 28 May 1999

Hobbies □ Cricket ☑ Football □ Volleyball

Gender ○ Male ● Female

City Mumbai

View:



view

Student Details

Roll no: 126

Name : Priya

Dob : 28-05-1999 00:00:00

hobbies: Football

Gender : Female

city: Mumbai