JAVA PROJECT

NAME : DINESHKUMAR V

ROLL NO : 22ITR023

BRANCH : B.TECH

DEPARTMENT : INFORMATION TECHNOLOGY

SECTION : A

PROJECT : DOCTOR APPOINTMENT SYSTEM

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

class Doctor extends Frame {
  Image backgroundImage;

  Doctor() {
    setTitle("Doctor Login");

    Label l1 = new Label("USERNAME:");
    Label l2 = new Label("PASSWORD:");

    TextField t1 = new TextField(15);
    TextField t2 = new TextField(15);

    Button b1 = new Button("LOGIN");
    Button b2 = new Button("CANCEL");

    b1.setBackground(Color.GRAY);
    b2.setBackground(Color.GRAY);

    l1.setBounds(680, 480, 90, 20);
    t1.setBounds(780, 480, 110, 20);
    l2.setBounds(680, 520, 90, 20);
    t2.setBounds(780, 520, 110, 20);
    b1.setBounds(700, 560, 70, 20);
```

```
b2.setBounds(780, 560, 70, 20);

add(l1);
add(t1);
add(l2);
add(t2);
add(b1);
add(b2);

setSize(400, 300);
setLocationRelativeTo(null);
setLayout(null);
setVisible(true);

Toolkit toolkit = Toolkit.getDefaultToolkit();
backgroundImage = toolkit.getImage("sphoto.jpg");

b1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (t1.getText().isEmpty() || t2.getText().isEmpty()) {
            Dialog dialog = new Dialog(Doctor.this, "Error", true);
            dialog.setLayout(new FlowLayout());
            dialog.add(new Label("Please enter username and password."));
            dialog.setSize(300, 100);
            dialog.setLocationRelativeTo(Doctor.this);

            Button closeButton = new Button("Close");
            closeButton.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    dialog.dispose();
                }
```

```
                    });
                    dialog.add(closeButton);
                    dialog.setVisible(true);
                } else {
                    dispose();
                    DoctorAppointment nextpage = new DoctorAppointment();
                    nextpage.setVisible(true);
                }
            }
        });

        b2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dispose();
            }
        });

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
    }
}

class DoctorAppointment extends Frame {
```

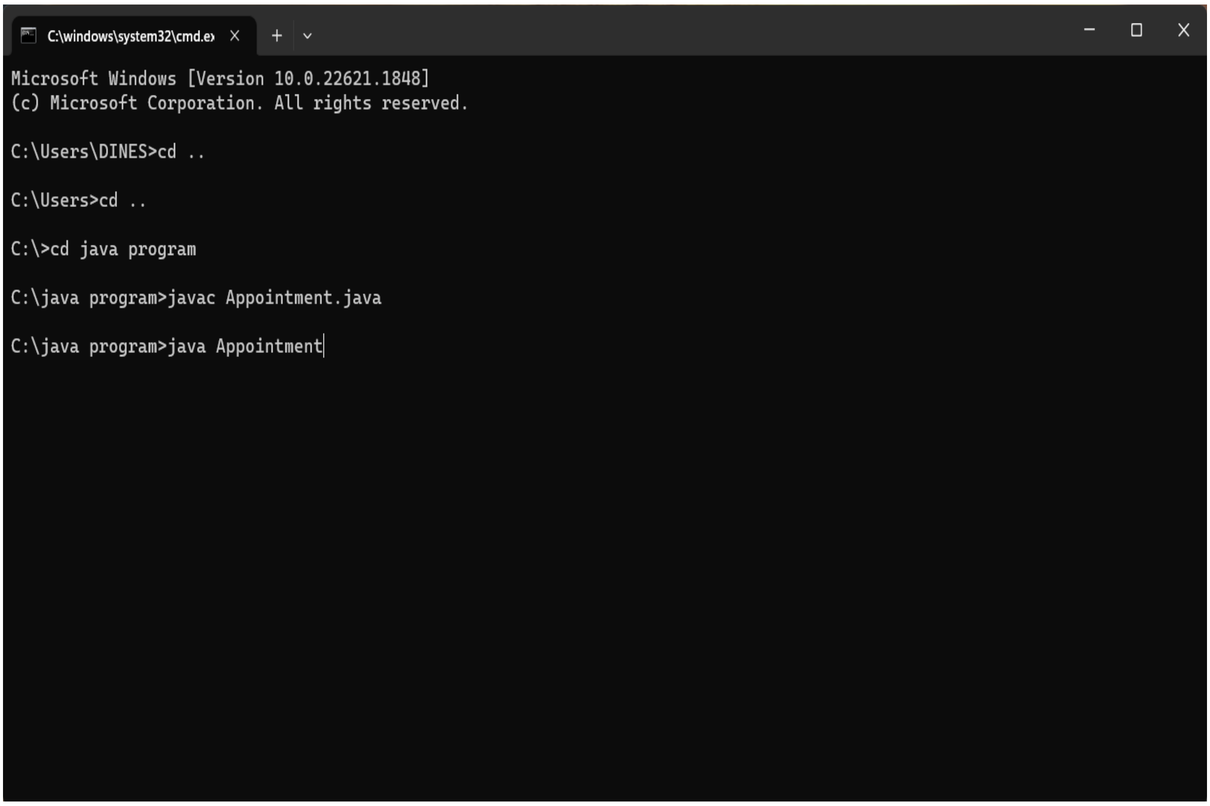
```
Image backgroundImage;

DoctorAppointment() {
    setTitle("Doctor Appointment");

    Label nameLabel = new Label("Name:");
    Label sexLabel = new Label("Sex:");
    Label dobLabel = new Label("Date of Birth:");
    Label bloodGroupLabel = new Label("Blood Group:");
    Label contactNumberLabel = new Label("Contact Number:");
    Label addressLabel = new Label("Address:");
    Label dateLabel = new Label("Date:");
    Label timeLabel = new Label("Time:");
    Label specialistLabel = new Label("Specialist:");
    Label doctorListLabel = new Label("Doctor List:");

    TextField nameTextField = new TextField();
    TextField contactNumberTextField = new TextField();
    TextField addressTextField = new TextField();

    CheckboxGroup sexCheckboxGroup = new CheckboxGroup();
    Checkbox maleCheckbox = new Checkbox("Male", sexCheckboxGroup, true);
    Checkbox femaleCheckbox = new Checkbox("Female", sexCheckboxGroup, false);
    Checkbox othersCheckbox = new Checkbox("Others", sexCheckboxGroup, false);

    Choice bloodGroupChoice = new Choice();
    bloodGroupChoice.add("  ");
    bloodGroupChoice.add("A+");
    bloodGroupChoice.add("A-");
    bloodGroupChoice.add("B+");
    bloodGroupChoice.add("B-");
bloodGroupChoice.add("O+");

bloodGroupChoice.add("O-");

bloodGroupChoice.add("AB+");

bloodGroupChoice.add("AB-");


Choice dateChoice = new Choice();

dateChoice.add("  ");

dateChoice.add("July 7, 2023");

dateChoice.add("July 13, 2023");

dateChoice.add("July 20, 2023");

dateChoice.add("August 1, 2023");

dateChoice.add("August 19, 2023");

dateChoice.add("August 30, 2023");


Choice timeChoice = new Choice();

timeChoice.add("  ");

timeChoice.add("10:00 AM to 01:00 PM");

timeChoice.add("02:00 PM to 05:00 PM");

timeChoice.add("07:00 PM to 09:00 PM");


Choice specialistChoice = new Choice();

specialistChoice.add("  ");

specialistChoice.add("Cardiologist");

specialistChoice.add("Dermatologist");

specialistChoice.add("Endocrinologist");

specialistChoice.add("Gastroenterologist");

specialistChoice.add("Neurologist");

specialistChoice.add("Oncologist");

specialistChoice.add("Ophthalmologists");


Choice doctorList = new Choice();
```

```
doctorList.add(" ");

doctorList.add("Dr. Aswin");

doctorList.add("Dr. ArunKumar");

doctorList.add("Dr. Santhiya");

doctorList.add("Dr. Subashini");

doctorList.add("Dr. David");

doctorList.add("Dr. Isai");


TextField dobTextField = new TextField();

TextField dateTextField = new TextField();

TextField timeTextField = new TextField();


Button submitButton = new Button("Submit");


nameLabel.setBounds(650, 150, 80, 20);

nameTextField.setBounds(750, 150, 150, 20);


sexLabel.setBounds(650, 190, 80, 20);

maleCheckbox.setBounds(750, 190, 60, 20);

femaleCheckbox.setBounds(810, 190, 70, 20);

othersCheckbox.setBounds(880, 190, 80, 20);


dobLabel.setBounds(650, 220, 90, 20);

dobTextField.setBounds(750, 220, 150, 20);


bloodGroupLabel.setBounds(650, 250, 90, 20);

bloodGroupChoice.setBounds(750, 250, 90, 20);


contactNumberLabel.setBounds(650, 280, 90, 20);

contactNumberTextField.setBounds(750, 280, 150, 20);
```

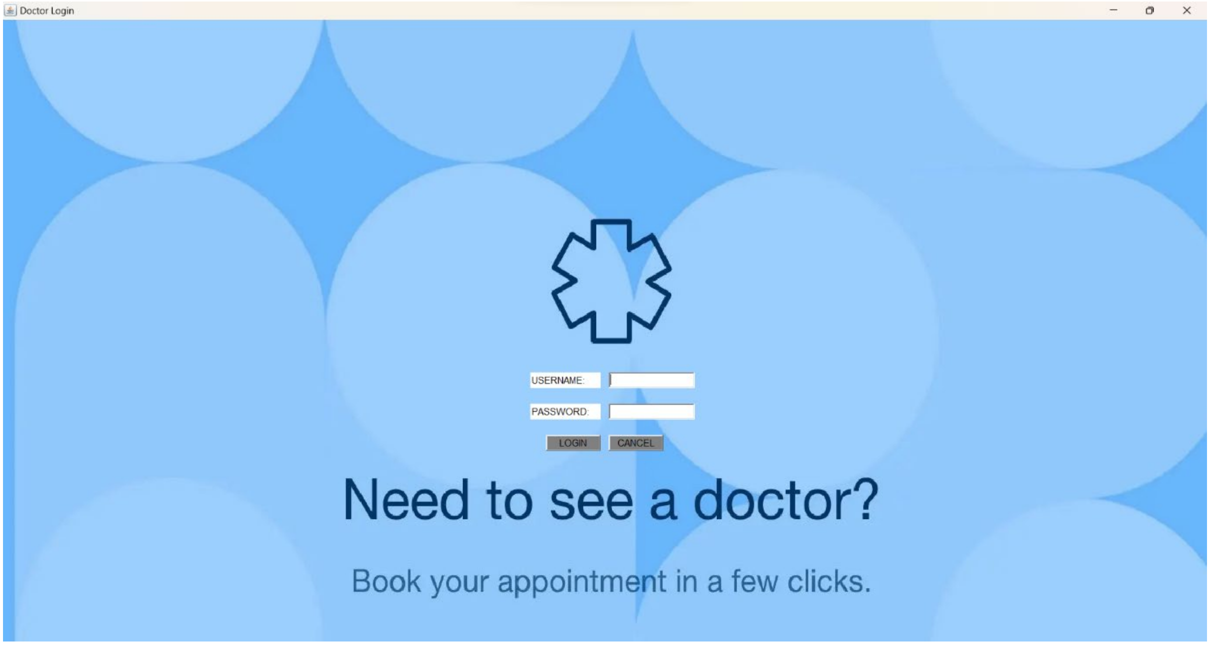
```
addressLabel.setBounds(650, 310, 90, 20);
addressTextField.setBounds(750, 310, 200, 100);

dateLabel.setBounds(650, 430, 90, 20);
dateChoice.setBounds(750, 430, 160, 30);

timeLabel.setBounds(650, 460, 90, 20);
timeChoice.setBounds(750, 460, 160, 20);

specialistLabel.setBounds(650, 490, 90, 20);
specialistChoice.setBounds(750, 490, 160, 30);

doctorListLabel.setBounds(650, 520, 90, 20);
doctorList.setBounds(750, 520, 160, 30);

submitButton.setBounds(740, 600, 90, 40);

add(nameLabel);
add(nameTextField);
add(sexLabel);
add(maleCheckbox);
add(femaleCheckbox);
add(othersCheckbox);
add(dobLabel);
add(dobTextField);
add(bloodGroupLabel);
add(bloodGroupChoice);
add(contactNumberLabel);
add(contactNumberTextField);
add(addressLabel);
add(addressTextField);
add(dateLabel);
add(dateChoice);
add(timeLabel);
add(timeChoice);
add(specialistLabel);
add(specialistChoice);
add(doctorListLabel);
add(doctorList);
add(submitButton);

submitButton.setBackground(Color.CYAN);

submitButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    if (nameTextField.getText().isEmpty() ||
        dobTextField.getText().isEmpty() ||
        contactNumberTextField.getText().isEmpty() ||
        addressTextField.getText().isEmpty() ||
        dateChoice.getSelectedItem().equals("  ") ||
        timeChoice.getSelectedItem().equals("  ") ||
        specialistChoice.getSelectedItem().equals("  ") ||
        doctorList.getSelectedItem().equals(" ")) {

      Dialog dialog = new Dialog(DoctorAppointment.this, "Error", true);
      dialog.setLayout(new FlowLayout());
      dialog.add(new Label("Please enter all the required information."));
      dialog.setSize(400, 100);
      dialog.setLocationRelativeTo(DoctorAppointment.this);

      Button closeButton = new Button("Close");
      closeButton.addActionListener(new ActionListener() {
```

```
                    public void actionPerformed(ActionEvent e) {
                        dialog.dispose();
                    }
                });
                dialog.add(closeButton);
                dialog.setVisible(true);
            } else {
                String selectedDate = dateChoice.getSelectedItem();
                String selectedTime = timeChoice.getSelectedItem();

                if (selectedDate.equals("July 7, 2023") && selectedTime.equals("02:00 PM to 05:00 PM"))
{
                    Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Conflict", true);
                    dialog.setLayout(new FlowLayout());
                    dialog.add(new Label("PLEASE PREFER OTHER DAY!!"));
                    dialog.setSize(300, 100);
                    dialog.setLocationRelativeTo(DoctorAppointment.this);
                    dialog.setVisible(true);
                } else if (selectedDate.equals("July 20, 2023") && selectedTime.equals("07:00 PM to
09:00 PM")) {
                    Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Conflict", true);
                    dialog.setLayout(new FlowLayout());
                    dialog.add(new Label("PLEASE PREFER OTHER DAY!!"));
                    dialog.setSize(300, 100);
                    dialog.setLocationRelativeTo(DoctorAppointment.this);
                    dialog.setVisible(true);
                } else if (selectedDate.equals("July 13, 2023") && selectedTime.equals("07:00 PM to
09:00 PM")) {
                    Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Booked", true);
                    dialog.setLayout(new FlowLayout());
                    dialog.add(new Label("Your appointment is booked. Visit the doctor on July 13."));
                    dialog.setSize(500, 100);
```

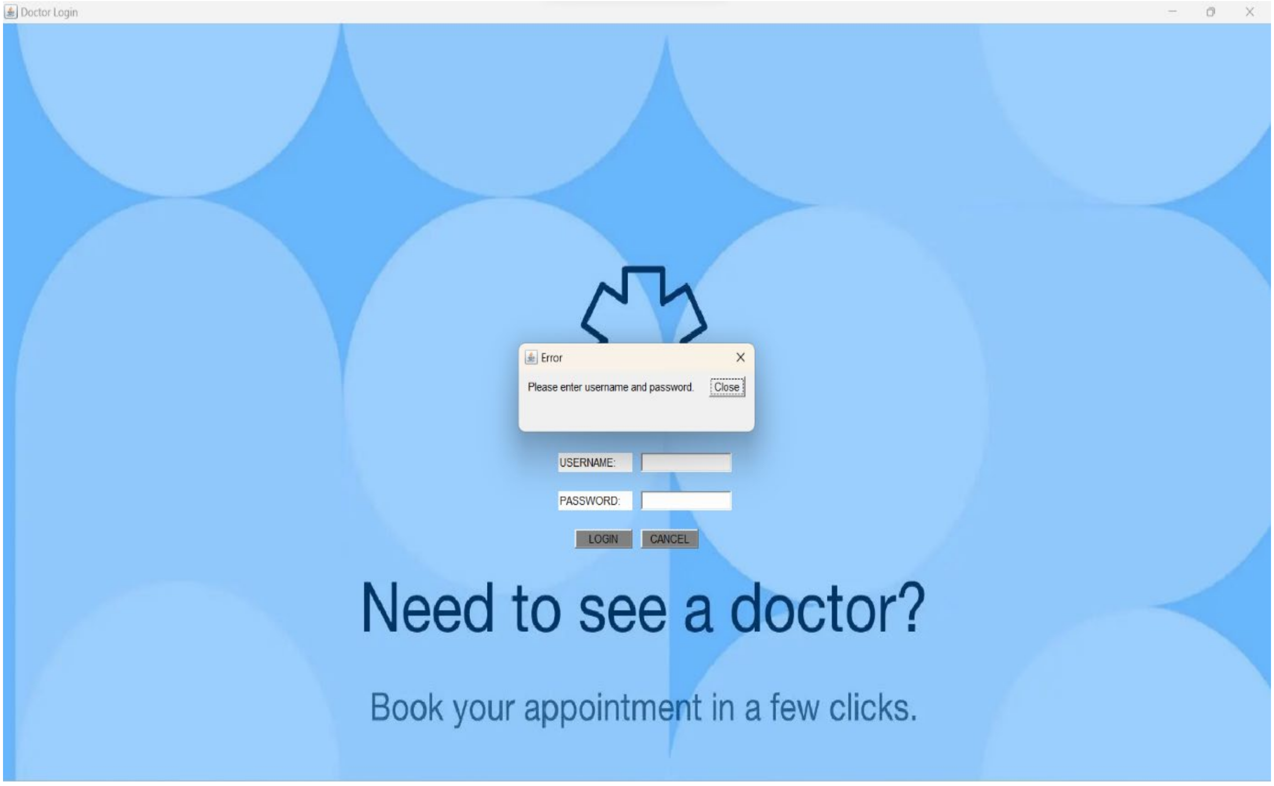
```
                        dialog.setLocationRelativeTo(DoctorAppointment.this);
                        dialog.setVisible(true);
                    } else if (selectedDate.equals("August 1, 2023") && selectedTime.equals("07:00 PM to
09:00 PM")) {
                        Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Booked", true);
                        dialog.setLayout(new FlowLayout());
                        dialog.add(new Label("Your appointment is booked. Visit the doctor on August 1."));
                        dialog.setSize(500, 100);
                        dialog.setLocationRelativeTo(DoctorAppointment.this);
                        dialog.setVisible(true);
                    } else if (selectedDate.equals("August 19, 2023") && selectedTime.equals("07:00 PM to
09:00 PM")) {
                        Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Booked", true);
                        dialog.setLayout(new FlowLayout());
                        dialog.add(new Label("Your appointment is booked. Visit the doctor on August 19."));
                        dialog.setSize(500, 100);
                        dialog.setLocationRelativeTo(DoctorAppointment.this);
                        dialog.setVisible(true);
                    } else if (selectedDate.equals("August 30, 2023") && selectedTime.equals("07:00 PM to
09:00 PM")) {
                        Dialog dialog = new Dialog(DoctorAppointment.this, "Appointment Conflict", true);
                        dialog.setLayout(new FlowLayout());
                        dialog.add(new Label("PLEASE PREFER OTHER DAY!!"));
                        dialog.setSize(400, 100);
                        dialog.setLocationRelativeTo(DoctorAppointment.this);
                        dialog.setVisible(true);
                    }
                }
            }
        });

        setSize(400, 500);
        setLocationRelativeTo(null);
        setLayout(null);
        setVisible(true);

        Toolkit toolkit = Toolkit.getDefaultToolkit();
        backgroundImage = toolkit.getImage("fn.jpg");
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), null);
    }
}

public class Appointment{
    public static void main(String[] args) {
        Doctor obj = new Doctor();
    }
}
```



```
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DINES>cd ..

C:\Users>cd ..

C:\>cd java program

C:\java program>javac Appointment.java

C:\java program>java Appointment
```



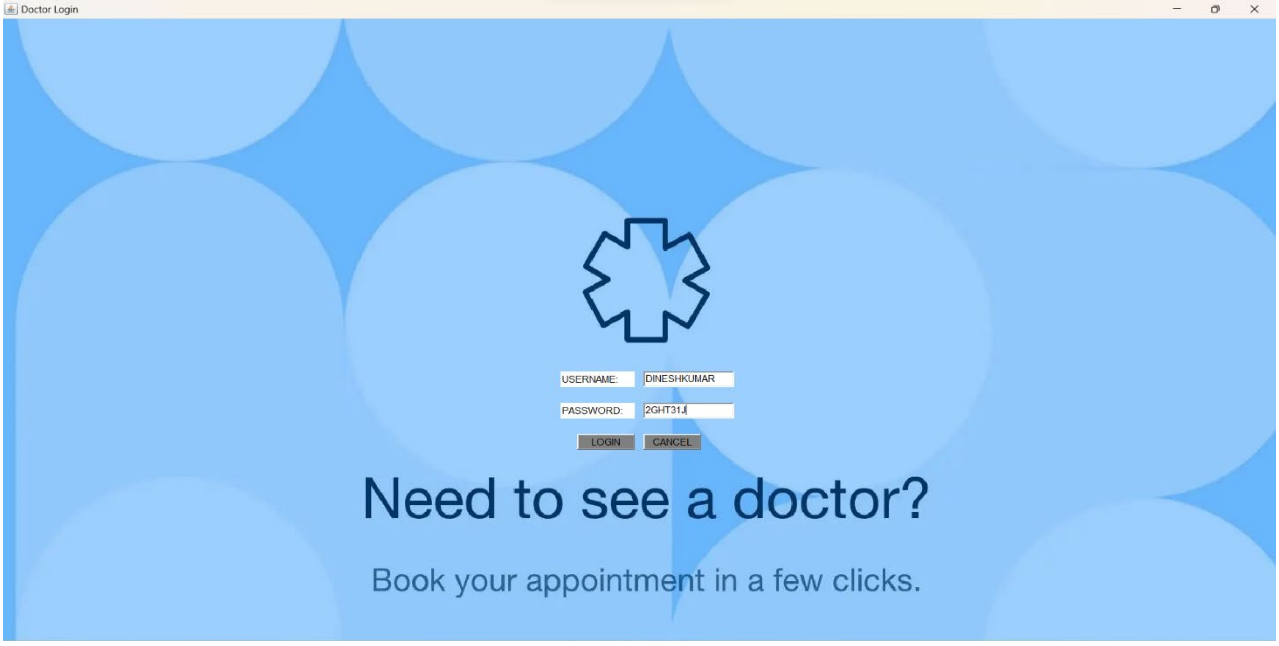
Doctor Login
USERNAME:
PASSWORD:
LOGIN
CANCEL
Need to see a doctor?
Book your appointment in a few clicks.



Please enter username and password.

Need to see a doctor?

Book your appointment in a few clicks.

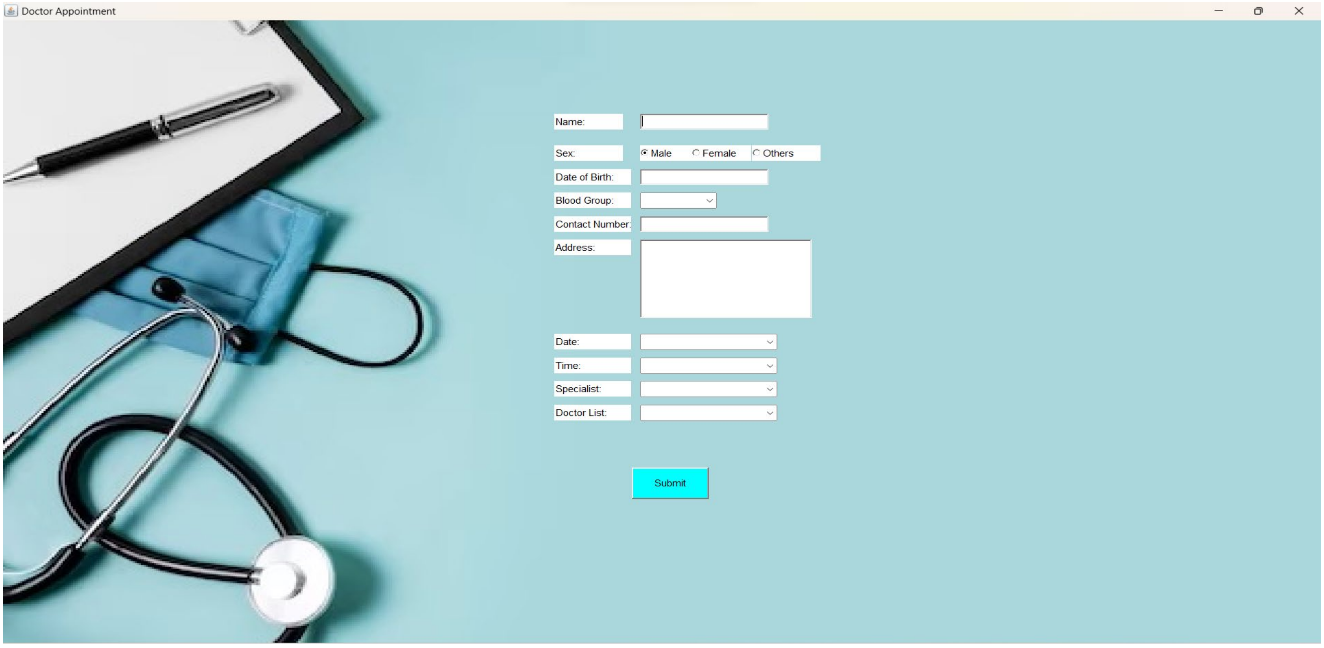


USERNAME: DINESHKUMAR

PASSWORD: 2GHT31J

Need to see a doctor?

Book your appointment in a few clicks.



Doctor Appointment

Name:

Sex: ● Male ○ Female ○ Others

Date of Birth:

Blood Group:

Contact Number:

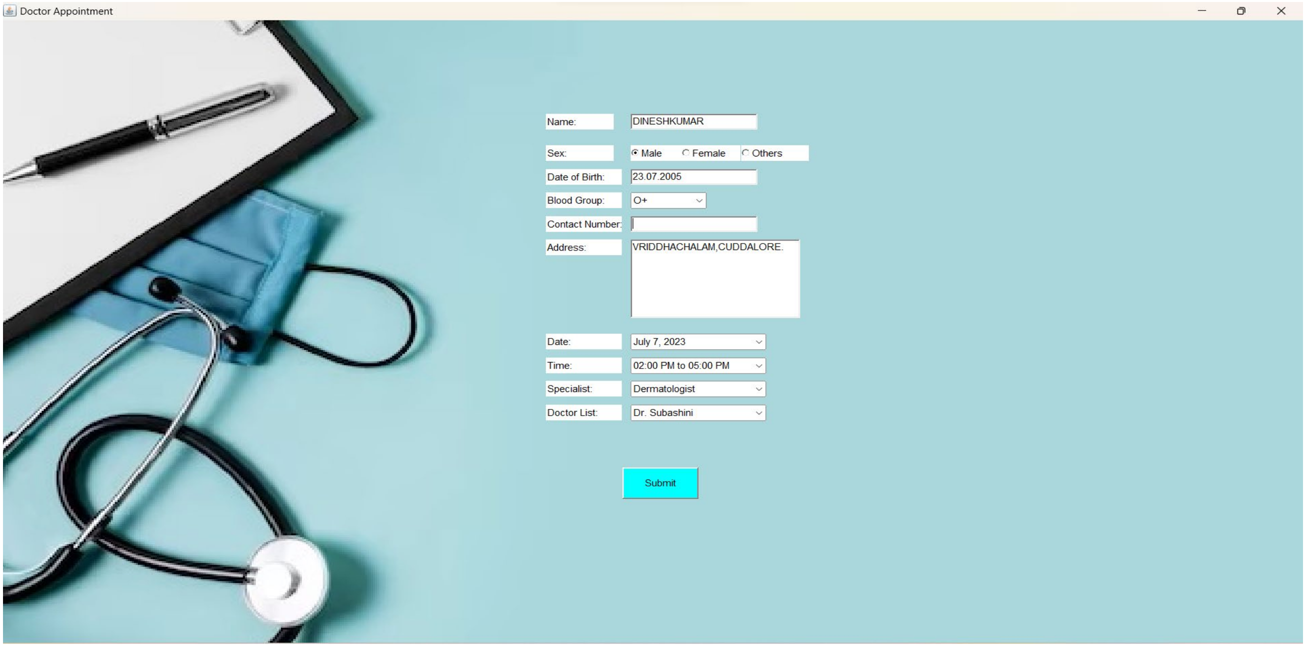
Address:

Date:

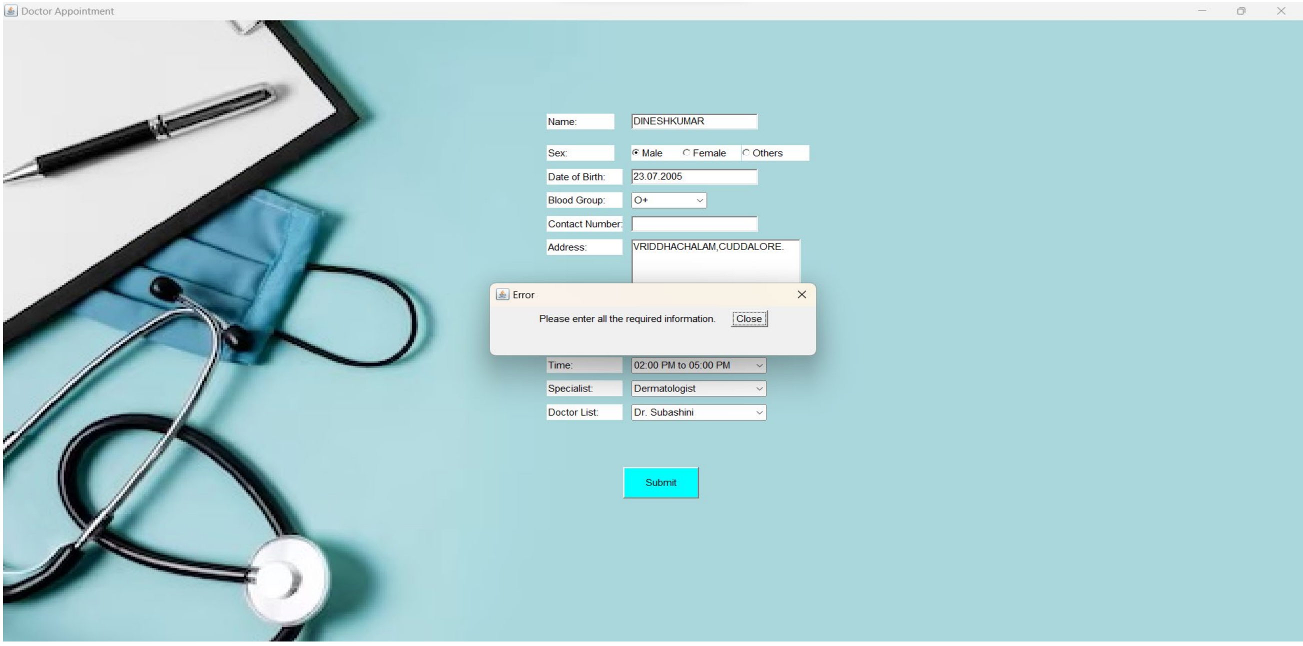
Time:

Specialist:

Doctor List:



Name:	DINESHKUMAR
Sex:	● Male ○ Female ○ Others
Date of Birth:	23.07.2005
Blood Group:	O+
Contact Number:	
Address:	VRIDDHACHALAM,CUDDALORE.
Date:	July 7, 2023
Time:	02:00 PM to 05:00 PM
Specialist:	Dermatologist
Doctor List:	Dr. Subashini

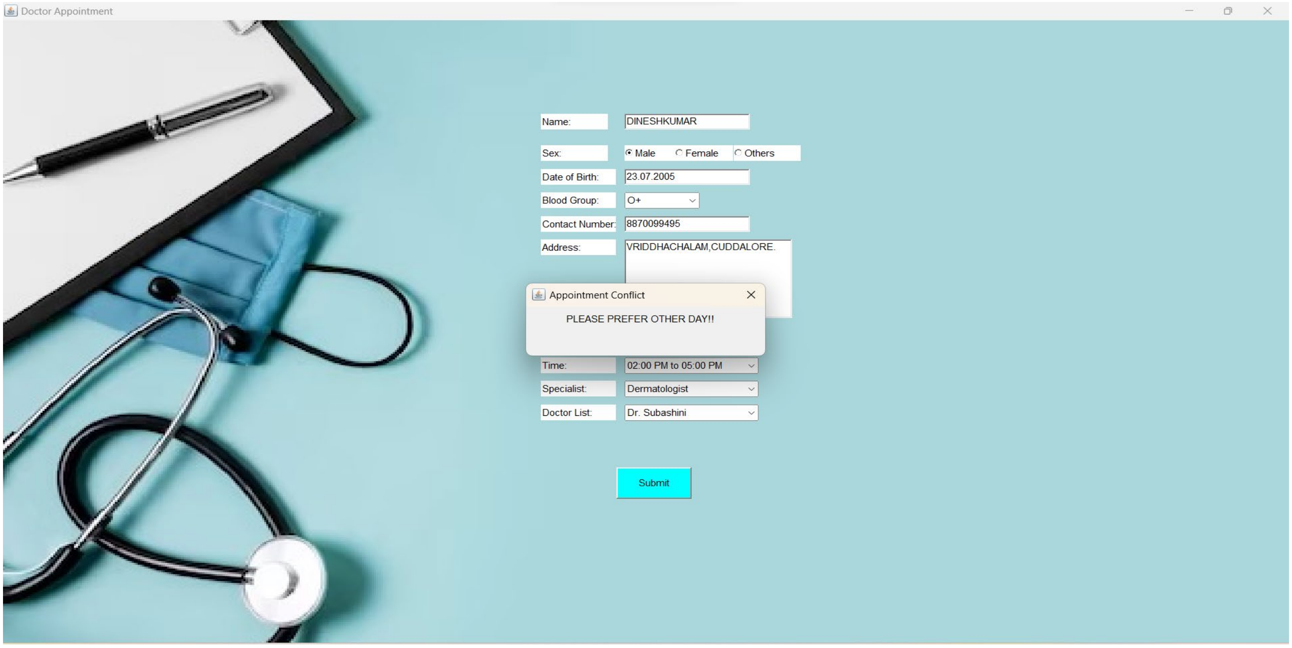


Doctor Appointment

Name:	DINESHKUMAR
Sex:	● Male ○ Female ○ Others
Date of Birth:	23.07.2005
Blood Group:	O+
Contact Number:	
Address:	VRIDDHACHALAM,CUDDALORE.
Time:	02:00 PM to 05:00 PM
Specialist:	Dermatologist
Doctor List:	Dr. Subashini

Error

Please enter all the required information.



Name: DINESHKUMAR

Sex: ● Male ○ Female ○ Others

Date of Birth: 23.07.2005

Blood Group: O+

Contact Number: 8870099495

Address: VRIDDHACHALAM,CUDDALORE.

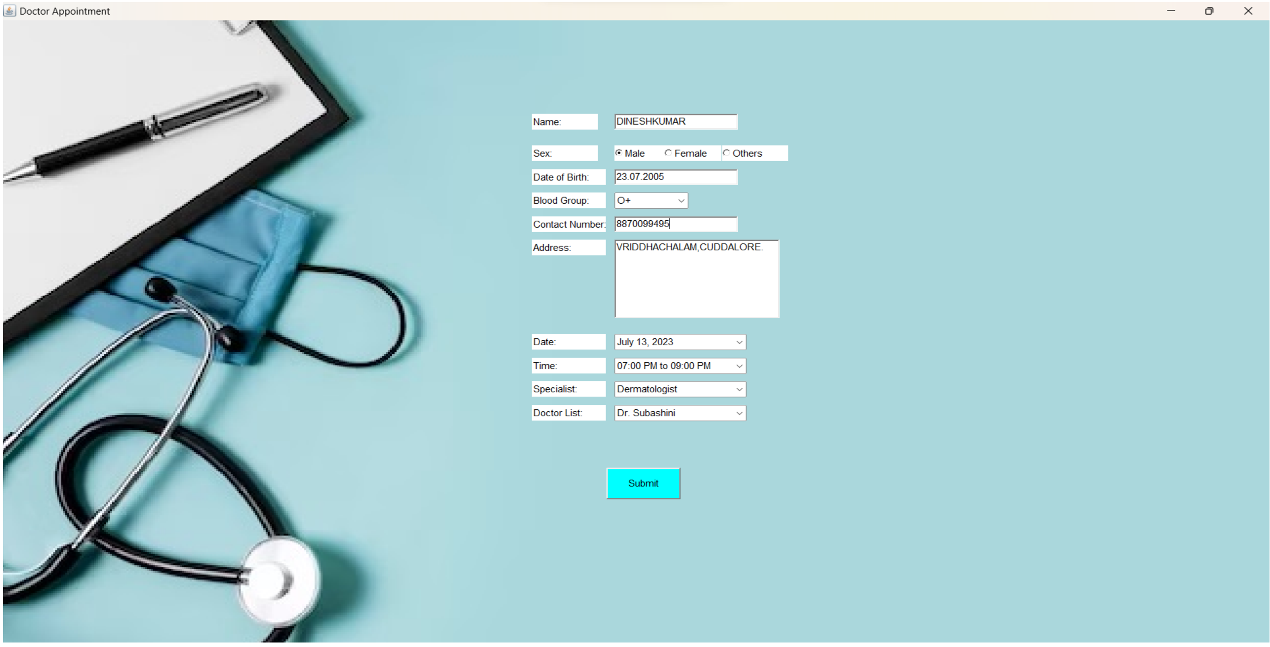
Appointment Conflict

PLEASE PREFER OTHER DAY!!

Time: 02:00 PM to 05:00 PM

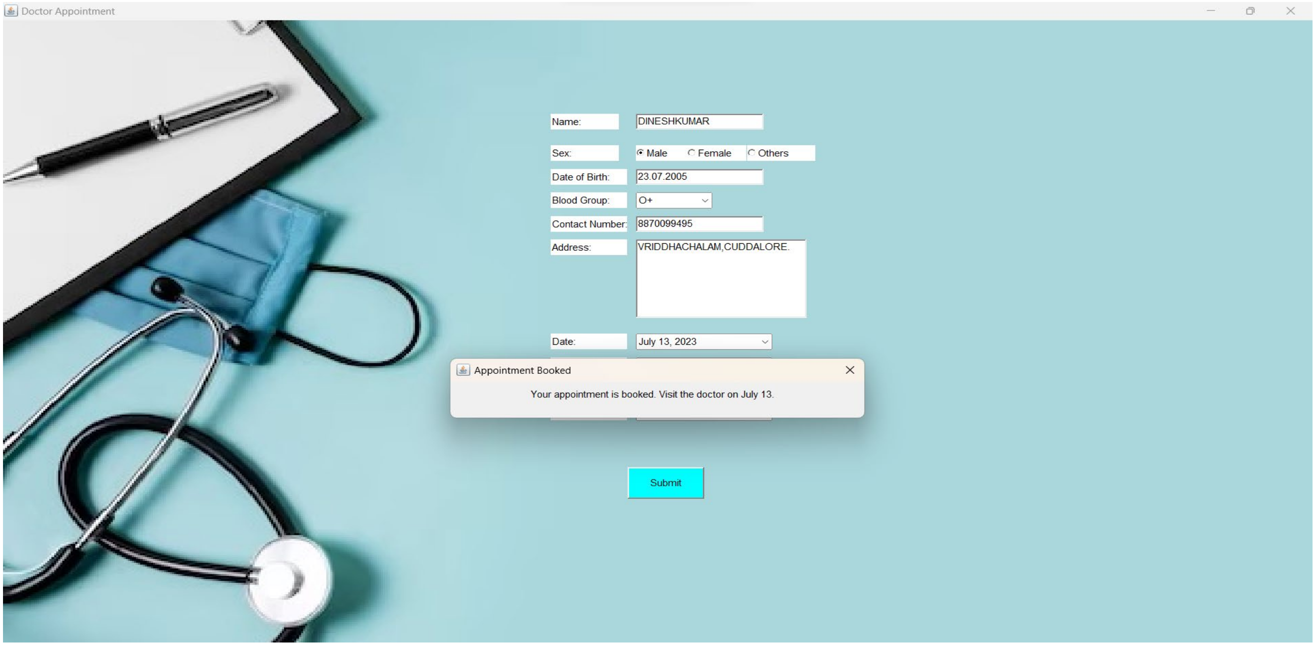
Specialist: Dermatologist

Doctor List: Dr. Subashini



Doctor Appointment

Name:	DINESHKUMAR
Sex:	● Male ○ Female ○ Others
Date of Birth:	23.07.2005
Blood Group:	O+
Contact Number:	8870099495
Address:	VRIDDHACHALAM,CUDDALORE.
Date:	July 13, 2023
Time:	07:00 PM to 09:00 PM
Specialist:	Dermatologist
Doctor List:	Dr. Subashini



Name:	DINESHKUMAR
Sex:	● Male ○ Female ○ Others
Date of Birth:	23.07.2005
Blood Group:	O+
Contact Number:	8870099495
Address:	VRIDDHACHALAM,CUDDALORE.
Date:	July 13, 2023

Appointment Booked

Your appointment is booked. Visit the doctor on July 13.