

JAVA AWT PROJECT

DEVELOPING A GUI BASED APPLICATION USING AWT CLASSES

Topic : Ambulance Service Provider System

Roll.No : 22ITR043

Name : KARTHIKEYAN R

Branch : B.TECH

Course : Information Technology

Section : A

Subject : Object Oriented Programming

CourseCode : 22ITC22

Aim:

To Develop a GUI based Ambulance Service Provider Application.

Coding:

```
import java.awt.*;
import java.awt.event.*;
public class AWTPProject {
    private static int currentPage = 1;
    private static Frame frame;
    private static Panel mainPanel;
    private static CardLayout cardLayout;
    private static Panel slidePanel;
    private static Panel loginPanel;
    private static Panel imagePanel;
    private static Panel demoPanel;
    private static Panel statisticPanel;
    private static Panel airPanel;
    private static Panel trainPanel;
    private static Panel icuPanel;
    private static Panel basicPanel;
    private static Panel traPanel;
    private static Panel deadPanel;
    private static Choice choice;
    public static void main(String[] args) {
        frame = new Frame("Ambulance Service Provider");
        cardLayout = new CardLayout();
        mainPanel = new Panel();
        mainPanel.setLayout(cardLayout);
        createSlidePanel();
        createLoginPanel();
        createImagePanel();
```

```

        createDemoPanel();
        createAirAmubulance();
        createStatisticPanel();
        createTrainPanel();
        createIcuPanel();
        createBasic();
        createTransport();
        createDead();
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        frame.add(mainPanel);
        frame.setSize(1280,720);
        frame.setVisible(true);
    }

    private static void createSlidePanel() {
        slidePanel = new Panel();
        slidePanel.setLayout(new BorderLayout());
        Label welcomeLabel = new Label("Welcome to Siren24 AMBULANCE provider
System");
        welcomeLabel.setFont(new Font("Arial", Font.BOLD, 20));
        welcomeLabel.setAlignment(Label.CENTER);
        slidePanel.add(welcomeLabel, BorderLayout.NORTH);
        ImagePanel imagePanel = new ImagePanel("E:/java project/page1.png");
        slidePanel.add(imagePanel, BorderLayout.CENTER);
        Panel buttonPanel = new Panel();
        Button nextPageButton = new Button("Register");
        nextPageButton.setPreferredSize(new Dimension(150, 40));
        nextPageButton.setBackground(Color.PINK);
        Button leftPageButton = new Button("About us");
        leftPageButton.setPreferredSize(new Dimension(150, 40));
        leftPageButton.setBackground(Color.PINK);
        Button rightPageButton = new Button("Get Ambulance");
        rightPageButton.setPreferredSize(new Dimension(150, 40));
        rightPageButton.setBackground(Color.PINK);
    }

```

```

slidePanel.setBackground(Color.LIGHT_GRAY);
nextPageButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.next(mainPanel);
        currentPage++;
        if (currentPage == 2) {
            frame.setSize(1280, 720);
        }
    }
});
leftPageButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "demoPanel");
    }
});
rightPageButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "statisticPanel");
    }
});
buttonPanel.add(leftPageButton);
buttonPanel.add(nextPageButton);
buttonPanel.add(rightPageButton);
slidePanel.add(buttonPanel, BorderLayout.SOUTH);
mainPanel.add(slidePanel, "slidePanel");
}

private static void createDead(){
    deadPanel=new Panel();
    deadPanel.setLayout(null);
    Label bloodgroup=new Label("Blood Group");
    Label welcomeLabel = new Label("Dead Body Ambulance");
    Label firstnameLabel = new Label("Name:");
    Label DepartureLabel = new Label("Departure");
    Label DestinationLabel = new Label("Destination:");
    Label genderLabel = new Label("Gender:");
    Label phoneLabel = new Label("Phone:");

```

```
Label selectLabel = new Label("Select:");
Label addressLabel = new Label("Address:");
TextField firstnameField = new TextField();
TextField DepartureField = new TextField();
TextField DestinationField = new TextField();
TextField phoneField = new TextField();
TextField addressField = new TextField();
Choice choice2=new Choice();
choice2.setBounds(200,460,200,20);
choice2.add("O-");
choice2.add("O+");
choice2.add("A+");
choice2.add("B-");
choice2.add("B+");
choice2.add("AB+");
choice2.add("others");
CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 100, 160, 20);
firstnameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
```

```

maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120,460,80,20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
DestinationField, phoneField, addressField)) {
            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});
cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});
deadPanel.add(firstnameLabel);
deadPanel.add(DepartureLabel);
deadPanel.add(bloodgroup);
deadPanel.add(DestinationLabel);
deadPanel.add(genderLabel);
deadPanel.add(phoneLabel);
deadPanel.add(selectLabel);
deadPanel.add(addressLabel);
deadPanel.add(choice2);
deadPanel.add(firstnameField);
deadPanel.add(DepartureField);
deadPanel.add(DestinationField);
deadPanel.add(phoneField);
deadPanel.add(addressField);
deadPanel.add(submitButton);

```

```

        deadPanel.add(cancelButton);
        deadPanel.add(maleCheckbox);
        deadPanel.add(femaleCheckbox);
        deadPanel.add(doctorCheckbox);
        deadPanel.add(welcomeLabel);

        ImagePanel loginImagePanel = new ImagePanel("E:/java
project/deadamb.46b69303.png");
        loginImagePanel.setBounds(3, 10, 1280, 720);
        deadPanel.add(loginImagePanel);
        mainPanel.add(deadPanel, "deadPanel");
    }

    private static void createTransport(){
        traPanel=new Panel();
        traPanel.setLayout(null);
        Label bloodgroup=new Label("Blood Group");
        Label welcomeLabel = new Label("Patient transport Ambulance");
        Label firstnameLabel = new Label("Name:");
        Label DepartureLabel = new Label("Departure");
        Label DestinationLabel = new Label("Destination:");
        Label genderLabel = new Label("Gender:");
        Label phoneLabel = new Label("Phone:");
        Label selectLabel = new Label("Select:");
        Label addressLabel = new Label("Address:");
        TextField firstnameField = new TextField();
        TextField DepartureField = new TextField();
        TextField DestinationField = new TextField();
        TextField phoneField = new TextField();
        TextField addressField = new TextField();
        Choice choice2=new Choice();
        choice2.setBounds(200,460,200,20);
        choice2.add("O-");
        choice2.add("O+");
        choice2.add("A+");
        choice2.add("B-");
        choice2.add("B+");
        choice2.add("AB+");
        choice2.add("others");
    }

```

```

CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Doctor", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 100, 160, 20);
firstnameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120,460,80,20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
DestinationField, phoneField, addressField)) {
            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
}

```



```

        }
    });
    cancelButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            cardLayout.show(mainPanel, "slidePanel");
        }
    });
    traPanel.add(firstnameLabel);
    traPanel.add(DepartureLabel);
    traPanel.add(bloodgroup);
    traPanel.add(DestinationLabel);
    traPanel.add(genderLabel);
    traPanel.add(phoneLabel);
    traPanel.add(selectLabel);
    traPanel.add(addressLabel);
    traPanel.add(choice2);
    traPanel.add(firstnameField);
    traPanel.add(DepartureField);
    traPanel.add(DestinationField);
    traPanel.add(phoneField);
    traPanel.add(addressField);
    traPanel.add(submitButton);
    traPanel.add(cancelButton);
    traPanel.add(maleCheckbox);
    traPanel.add(femaleCheckbox);
    traPanel.add(doctorCheckbox);
    traPanel.add(patientCheckbox);
    traPanel.add(welcomeLabel);

    ImagePanel loginImagePanel = new ImagePanel("E:/java
project/patamb.4baffc27.png");
    loginImagePanel.setBounds(3, 10, 1280, 720);
    traPanel.add(loginImagePanel);
    mainPanel.add(traPanel, "traPanel");
}

private static void createBasic(){
    basicPanel=new Panel();
    basicPanel.setLayout(null);

```

```
Label bloodgroup=new Label("Blood Group");
Label welcomeLabel = new Label("Basic life support Ambulance");
Label firstnameLabel = new Label("Name:");
Label DepartureLabel = new Label("Departure");
Label DestinationLabel = new Label("Destination:");
Label genderLabel = new Label("Gender:");
Label phoneLabel = new Label("Phone:");
Label selectLabel = new Label("Select:");
Label addressLabel = new Label("Address:");
TextField firstnameField = new TextField();
TextField DepartureField = new TextField();
TextField DestinationField = new TextField();
TextField phoneField = new TextField();
TextField addressField = new TextField();
Choice choice2=new Choice();
choice2.setBounds(200,460,200,20);
choice2.add("O-");
choice2.add("O+");
choice2.add("A+");
choice2.add("B-");
choice2.add("B+");
choice2.add("AB+");
choice2.add("others");
CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Doctor", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 100, 180, 20);
firstnameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
```

```

phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120, 460, 80, 20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
DestinationField, phoneField, addressField)) {
            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});
cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});
basicPanel.add(firstnameLabel);
basicPanel.add(DepartureLabel);
basicPanel.add(bloodgroup);
basicPanel.add(DestinationLabel);
basicPanel.add(genderLabel);
basicPanel.add(phoneLabel);

```

```

        basicPanel.add(selectLabel);
        basicPanel.add(addressLabel);
        basicPanel.add(choice2);
        basicPanel.add(firstnameField);
        basicPanel.add(DepartureField);
        basicPanel.add(DestinationField);
        basicPanel.add(phoneField);
        basicPanel.add(addressField);
        basicPanel.add(submitButton);
        basicPanel.add(cancelButton);
        basicPanel.add(maleCheckbox);
        basicPanel.add(femaleCheckbox);
        basicPanel.add(doctorCheckbox);
        basicPanel.add(patientCheckbox);
        basicPanel.add(welcomeLabel);

        ImagePanel loginImagePanel = new ImagePanel("E:/java
project/basicamb.c1743505.png");

        loginImagePanel.setBounds(3, 10, 1280, 720);
        basicPanel.add(loginImagePanel);
        mainPanel.add(basicPanel, "basicPanel");
    }

    private static void createIcuPanel(){
        icuPanel=new Panel();
        icuPanel.setLayout(null);

        Label bloodgroup=new Label("Blood Group");
        Label welcomeLabel = new Label("ICU care Ambulance");
        Label firstnameLabel = new Label("Name:");
        Label DepartureLabel = new Label("Departure");
        Label DestinationLabel = new Label("Destination:");
        Label genderLabel = new Label("Gender:");
        Label phoneLabel = new Label("Phone:");
        Label selectLabel = new Label("Select:");
        Label addressLabel = new Label("Address:");
        TextField firstnameField = new TextField();
        TextField DepartureField = new TextField();
        TextField DestinationField = new TextField();
        TextField phoneField = new TextField();
    }

```

```
TextField addressField = new TextField();
Choice choice2=new Choice();
choice2.setBounds(200,460,200,20);
choice2.add("O-");
choice2.add("O+");
choice2.add("A+");
choice2.add("B-");
choice2.add("B+");
choice2.add("AB+");
choice2.add("others");
CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Doctor", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 100, 120, 20);
firstnameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
```

```

cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120,460,80,20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
DestinationField, phoneField, addressField)) {

            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});
cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});
icuPanel.add(firstnameLabel);
icuPanel.add(DepartureLabel);
icuPanel.add(bloodgroup);
icuPanel.add(DestinationLabel);
icuPanel.add(genderLabel);
icuPanel.add(phoneLabel);
icuPanel.add(selectLabel);
icuPanel.add(addressLabel);
icuPanel.add(choice2);
icuPanel.add(firstnameField);
icuPanel.add(DepartureField);
icuPanel.add(DestinationField);
icuPanel.add(phoneField);
icuPanel.add(addressField);
icuPanel.add(submitButton);
icuPanel.add(cancelButton);
icuPanel.add(maleCheckbox);
icuPanel.add(femaleCheckbox);

```

```

        icuPanel.add(doctorCheckbox);
        icuPanel.add(patientCheckbox);
        icuPanel.add(welcomeLabel);

        ImagePanel loginImagePanel = new ImagePanel("E:/java
project/icuamb.7309b35d.png");

        loginImagePanel.setBounds(3, 10, 1280, 720);
        icuPanel.add(loginImagePanel);
        mainPanel.add(icuPanel, "icuPanel");
    }

    private static void createAirAmubulance(){
        airPanel=new Panel();
        airPanel.setLayout(null);

        Label bloodgroup=new Label("Blood Group");
        Label welcomeLabel = new Label("Air Ambulance");
        Label firstnameLabel = new Label("Name:");
        Label DepartureLabel = new Label("Departure");
        Label DestinationLabel = new Label("Destination:");
        Label genderLabel = new Label("Gender:");
        Label phoneLabel = new Label("Phone:");
        Label selectLabel = new Label("Select:");
        Label addressLabel = new Label("Address:");
        TextField firstnameField = new TextField();
        TextField DepartureField = new TextField();
        TextField DestinationField = new TextField();
        TextField phoneField = new TextField();
        TextField addressField = new TextField();
        Choice choice2=new Choice();
        choice2.setBounds(200,460,200,20);
        choice2.add("O-");
        choice2.add("O+");
        choice2.add("A+");
        choice2.add("B-");
        choice2.add("B+");
        choice2.add("AB+");
        choice2.add("others");
        CheckboxGroup genderGroup = new CheckboxGroup();
        Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);

```

```

Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Doctor", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 80, 80, 50);
firstnameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120, 460, 80, 20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
            DestinationField, phoneField, addressField)) {
            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});

```



```

cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});

airPanel.add(firstnameLabel);
airPanel.add(DepartureLabel);
airPanel.add(bloodgroup);
airPanel.add(DestinationLabel);
airPanel.add(genderLabel);
airPanel.add(phoneLabel);
airPanel.add(selectLabel);
airPanel.add(addressLabel);
airPanel.add(choice2);
airPanel.add(firstnameField);
airPanel.add(DepartureField);
airPanel.add(DestinationField);
airPanel.add(phoneField);
airPanel.add(addressField);
airPanel.add(submitButton);
airPanel.add(cancelButton);
airPanel.add(maleCheckbox);
airPanel.add(femaleCheckbox);
airPanel.add(doctorCheckbox);
airPanel.add(patientCheckbox);
airPanel.add(welcomeLabel);

ImagePanel loginImagePanel = new ImagePanel("E:/java
project/airamb.be5b3c86.png");

loginImagePanel.setBounds(3, 10, 1280, 720);
airPanel.add(loginImagePanel);
mainPanel.add(airPanel, "airPanel");
}

private static void createTrainPanel(){
    trainPanel=new Panel();
    trainPanel.setLayout(null);
    Label bloodgroup=new Label("Blood Group");
    Label welcomeLabel = new Label("Train Ambulance");

```

```
Label firstNameLabel = new Label("Name:");
Label DepartureLabel = new Label("Departure");
Label DestinationLabel = new Label("Destination:");
Label genderLabel = new Label("Gender:");
Label phoneLabel = new Label("Phone:");
Label selectLabel = new Label("Select:");
Label addressLabel = new Label("Address:");
TextField firstNameField = new TextField();
TextField DepartureField = new TextField();
TextField DestinationField = new TextField();
TextField phoneField = new TextField();
TextField addressField = new TextField();
Choice choice2=new Choice();
choice2.setBounds(200,460,200,20);
choice2.add("O-");
choice2.add("O+");
choice2.add("A+");
choice2.add("B-");
choice2.add("B+");
choice2.add("AB+");
choice2.add("others");
CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Doctor", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Patient", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");
welcomeLabel.setBounds(190, 80, 100, 50);
firstNameLabel.setBounds(120, 140, 80, 20);
DepartureLabel.setBounds(120, 180, 80, 20);
DestinationLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
```

```

addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
DepartureField.setBounds(200, 180, 120, 20);
DestinationField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120,460,80,20);
submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, DepartureField,
DestinationField, phoneField, addressField)) {
            cardLayout.show(mainPanel, "slidePanel");
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});
cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});
trainPanel.add(firstnameLabel);
trainPanel.add(DepartureLabel);
trainPanel.add(bloodgroup);
trainPanel.add(DestinationLabel);
trainPanel.add(genderLabel);
trainPanel.add(phoneLabel);
trainPanel.add(selectLabel);
trainPanel.add(addressLabel);

```

```

        trainPanel.add(choice2);
        trainPanel.add(firstnameField);
        trainPanel.add(DepartureField);
        trainPanel.add(DestinationField);
        trainPanel.add(phoneField);
        trainPanel.add(addressField);
        trainPanel.add(submitButton);
        trainPanel.add(cancelButton);
        trainPanel.add(maleCheckbox);
        trainPanel.add(femaleCheckbox);
        trainPanel.add(doctorCheckbox);
        trainPanel.add(patientCheckbox);
        trainPanel.add(welcomeLabel);

        ImagePanel loginImagePanel = new ImagePanel("E:/java
project/trainamb.35f27b63.png");

        loginImagePanel.setBounds(3, 10, 1280, 720);
        trainPanel.add(loginImagePanel);
        mainPanel.add(trainPanel, "trainPanel");
    }

    private static void createLoginPanel() {
        loginPanel = new Panel();
        loginPanel.setLayout(null);

        Label bloodgroup=new Label("Blood Group");
        Label welcomeLabel = new Label("Registration Page");
        Label firstnameLabel = new Label("Name:");
        Label lastnameLabel = new Label("Email:");
        Label cityLabel = new Label("City");
        Label genderLabel = new Label("Gender:");
        Label phoneLabel = new Label("Phone:");
        Label selectLabel = new Label("Select:");
        Label addressLabel = new Label("Address:");
        TextField firstnameField = new TextField();
        TextField lastnameField = new TextField();
        TextField emailField = new TextField();
        TextField phoneField = new TextField();
        TextField addressField = new TextField();
        Choice choice2=new Choice();
    }

```

```
choice2.setBounds(200,460,200,20);
choice2.add("O-");
choice2.add("O+");
choice2.add("A+");
choice2.add("B-");
choice2.add("B+");
choice2.add("AB+");
choice2.add("others");

CheckboxGroup genderGroup = new CheckboxGroup();
Checkbox maleCheckbox = new Checkbox("Male", genderGroup, false);
Checkbox femaleCheckbox = new Checkbox("Female", genderGroup, false);
CheckboxGroup selectGroup = new CheckboxGroup();
Checkbox doctorCheckbox = new Checkbox("Driver", selectGroup, false);
Checkbox patientCheckbox = new Checkbox("Helper", selectGroup, false);
Button submitButton = new Button("Submit");
Button cancelButton = new Button("Cancel");

welcomeLabel.setBounds(190, 80, 120, 50);
firstnameLabel.setBounds(120, 140, 80, 20);
lastnameLabel.setBounds(120, 180, 80, 20);
cityLabel.setBounds(120, 220, 80, 20);
genderLabel.setBounds(120, 260, 80, 20);
phoneLabel.setBounds(120, 300, 80, 20);
selectLabel.setBounds(120, 340, 80, 20);
addressLabel.setBounds(120, 380, 80, 20);
firstnameField.setBounds(200, 140, 120, 20);
lastnameField.setBounds(200, 180, 120, 20);
emailField.setBounds(200, 220, 120, 20);
phoneField.setBounds(200, 300, 120, 20);
addressField.setBounds(200, 380, 120, 60);
maleCheckbox.setBounds(200, 260, 60, 20);
femaleCheckbox.setBounds(260, 260, 70, 20);
doctorCheckbox.setBounds(200, 340, 60, 20);
patientCheckbox.setBounds(260, 340, 60, 20);
submitButton.setBounds(140, 500, 55, 35);
cancelButton.setBounds(200, 500, 55, 35);
bloodgroup.setBounds(120,460,80,20);
```

```

submitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (allDetailsEntered(firstnameField, lastnameField, emailField,
phoneField, addressField)) {
            cardLayout.next(mainPanel);
            currentPage++;
            if (currentPage == 3) {
                cardLayout.show(mainPanel, "imagePanel");
            }
            showDialog("Details Submitted Successfully!");
        } else {
            showDialog("Please enter all details.");
        }
    }
});

cancelButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});

loginPanel.add(firstnameLabel);
loginPanel.add(lastnameLabel);
loginPanel.add(bloodgroup);
loginPanel.add(cityLabel);
loginPanel.add(genderLabel);
loginPanel.add(phoneLabel);
loginPanel.add(selectLabel);
loginPanel.add(addressLabel);
loginPanel.add(choice2);
loginPanel.add(firstnameField);
loginPanel.add(lastnameField);
loginPanel.add(emailField);
loginPanel.add(phoneField);
loginPanel.add(addressField);
loginPanel.add(submitButton);
loginPanel.add(cancelButton);
loginPanel.add(maleCheckbox);

```

```

        loginPanel.add(femaleCheckbox);
        loginPanel.add(doctorCheckbox);
        loginPanel.add(patientCheckbox);
        loginPanel.add(welcomeLabel);
        ImagePanel loginImagePanel = new ImagePanel("E:/java project/front.png");
        loginImagePanel.setBounds(3, 10, 1280, 720);
        loginPanel.add(loginImagePanel);
        mainPanel.add(loginPanel, "loginPanel");
    }

    private static void createImagePanel() {
        imagePanel = new Panel();
        imagePanel.setLayout(new BorderLayout());
        imagePanel.setBackground(Color.LIGHT_GRAY);
        Label successLabel = new Label("CONGRATS!");
        successLabel.setFont(new Font("Arial", Font.BOLD, 20));
        successLabel.setAlignment(Label.CENTER);
        imagePanel.add(successLabel, BorderLayout.NORTH);
        ImagePanel image = new ImagePanel("E:/java project/register.png");
        imagePanel.add(image, BorderLayout.CENTER);
        mainPanel.add(imagePanel, "imagePanel");
    }

    private static void createDemoPanel(){
        demoPanel=new Panel();
        demoPanel.setBackground(Color.PINK);
        ImagePanel leftImage = new ImagePanel("E:/java project/Aboutus.png");
        demoPanel.setLayout(new BorderLayout());
        demoPanel.add(leftImage, BorderLayout.CENTER);
        Button backButton = new Button("Back");
        backButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                cardLayout.show(mainPanel, "slidePanel");
            }
        });
        demoPanel.add(backButton, BorderLayout.SOUTH);
        mainPanel.add(demoPanel, "demoPanel");
    }
}

```

```

private static void createStatisticPanel() {
    statisticPanel = new Panel();
    statisticPanel.setBackground(Color.CYAN);
    statisticPanel.setLayout(new BorderLayout());
    Label topLabel = new Label("Select an option:");
    topLabel.setFont(new Font("Arial", Font.BOLD, 20));
    topLabel.setAlignment(Label.CENTER);
    statisticPanel.add(topLabel, BorderLayout.NORTH);
    Choice choice = new Choice();
    choice.add("ICU Care Ambulance");
    choice.add("Basic Life Support Ambulance");
    choice.add("Dead Body Ambulance");
    choice.add("Air Ambulance");
    choice.add("Patient Transport Ambulance");
    choice.add("Train Ambulance");
    statisticPanel.add(choice, BorderLayout.CENTER);
    Panel buttonPanel = new Panel();
    Button submitButton = new Button("Submit");
    submitButton.setPreferredSize(new Dimension(150, 40));
    submitButton.setBackground(Color.PINK);
    Button backButton = new Button("Back");
    backButton.setPreferredSize(new Dimension(150, 40));
    backButton.setBackground(Color.PINK);
    submitButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String selectedOption = choice.getSelectedItem();
            if (selectedOption.equals("ICU Care Ambulance")) {
                cardLayout.show(mainPanel, "icuPanel");
                showDialog("Purpose: to transport patients who requires more advanced
medical supervision.");
            } else if (selectedOption.equals("Basic Life Support Ambulance")) {
                cardLayout.show(mainPanel, "basicPanel");
                showDialog("Purpose: to transport patients who requires more basic
medical supervision.");
            } else if (selectedOption.equals("Dead Body Ambulance")) {
                cardLayout.show(mainPanel, "deadPanel");
            }
        }
    });
}

```



```

        showDialog("Purpose: to transport the dead body of a person to the
place of last rites.");
    }else if (selectedOption.equals("Train Ambulance")) {
        cardLayout.show(mainPanel, "trainPanel");
        showDialog("Purpose: to tranport patients from one place to another
far away place.");
    }else if (selectedOption.equals("Air Ambulance")) {
        cardLayout.show(mainPanel, "airPanel");
        showDialog("Purpose: to transport patients from one palce to another
far away place through air.");
    }else if (selectedOption.equals("Patient Transport Ambulance")) {
        cardLayout.show(mainPanel, "traPanel");
        showDialog("Purpose: to transport patients who have to transport from
one hospital to another.");
    }
}

});

backButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        cardLayout.show(mainPanel, "slidePanel");
    }
});

buttonPanel.add(submitButton);
buttonPanel.add(backButton);
statisticPanel.add(buttonPanel, BorderLayout.SOUTH);
mainPanel.add(statisticPanel, "statisticPanel");
}

private static void showDialog(String message) {
    final Dialog dialog = new Dialog(frame, "Success", true);
    dialog.setSize(500, 400);
    dialog.setLocationRelativeTo(frame);
    final Label label = new Label(message);
    label.setAlignment(Label.CENTER);
    dialog.setLayout(new BorderLayout());
    dialog.add(label, BorderLayout.CENTER);
    new Thread(new Runnable() {
        @Override

```

```

        public void run() {
            try {
                Thread.sleep(2500);
                dialog.dispose();
            } catch (InterruptedException ex) {
                ex.printStackTrace();
            }
        }
    }).start();
    dialog.setVisible(true);
}

private static boolean allDetailsEntered(TextField firstnameField, TextField
lastnameField, TextField emailField,
                                           TextField phoneField, TextField
addressField) {
    String firstname = firstnameField.getText();
    String lastname = lastnameField.getText();
    String email = emailField.getText();
    String phone = phoneField.getText();
    String address = addressField.getText();

    return !firstname.isEmpty() && !lastname.isEmpty() && !email.isEmpty()
        && !phone.isEmpty() && !address.isEmpty();
}

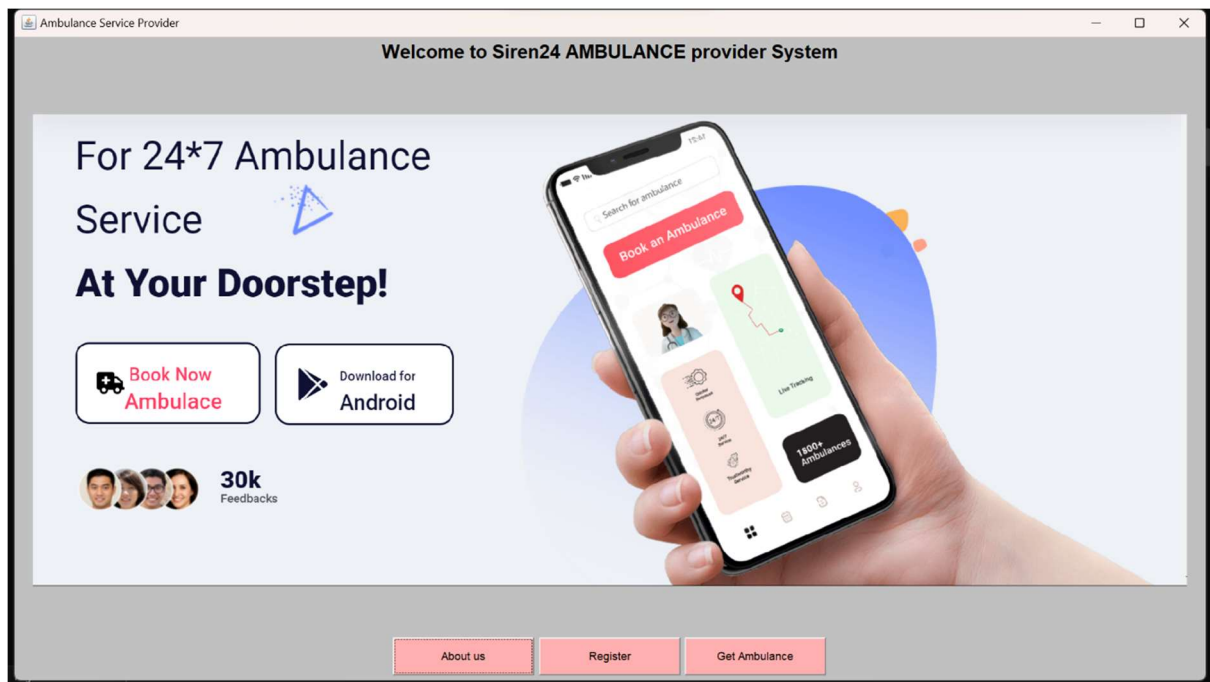
class ImagePanel extends Panel {
    private Image image;
    private int scrollPosition;
    public ImagePanel(String imagePath) {
        image = Toolkit.getDefaultToolkit().getImage(imagePath);
        scrollPosition = 0;
    }
    public void setScrollPosition(int position) {
        scrollPosition = position;
        repaint();
    }
    public void paint(Graphics g) {

```

```
int panelWidth = getWidth();
int panelHeight = getHeight();
int imageWidth = image.getWidth(this);
int imageHeight = image.getHeight(this);
double aspectRatio = (double) imageWidth / imageHeight;
int newWidth, newHeight;
if (panelWidth / aspectRatio > panelHeight) {
    newHeight = panelHeight - 40;
    newWidth = (int) (newHeight * aspectRatio);
} else {
    newWidth = panelWidth - 40;
    newHeight = (int) (newWidth / aspectRatio);
}
int x = (panelWidth - newWidth) / 2;
int y = (panelHeight - newHeight) / 2;
g.drawImage(image, x, y, newWidth, newHeight, this);
}
}
```

Output:

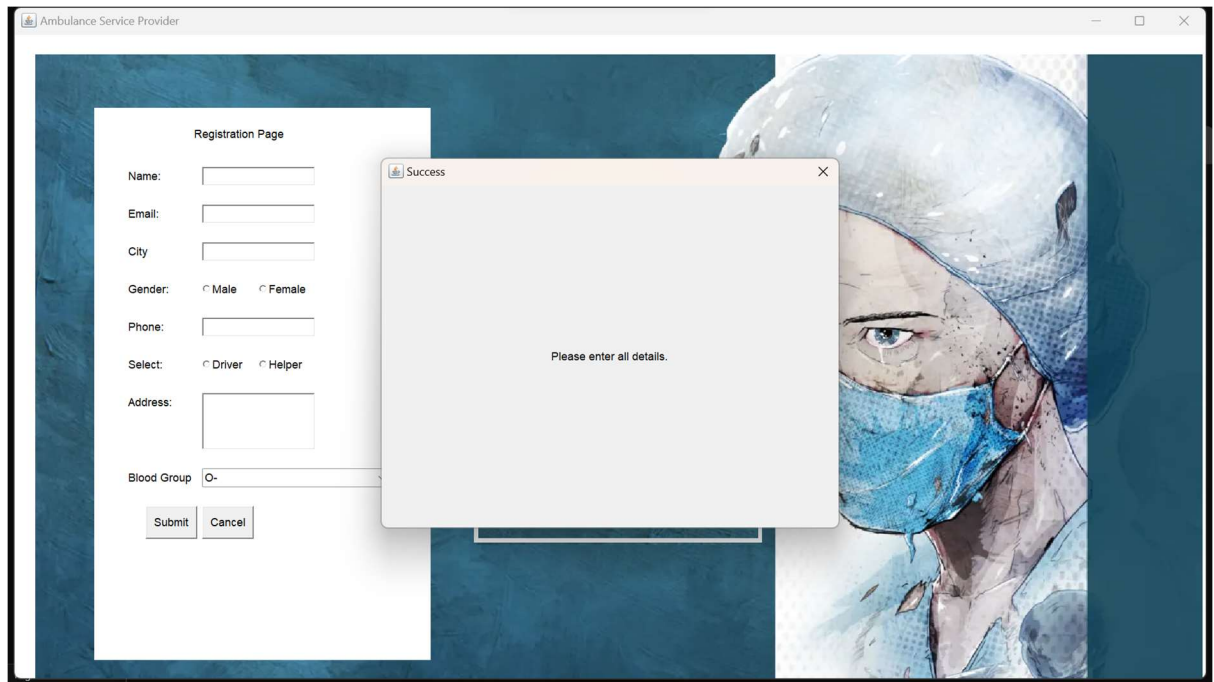
I. Welcome Page..



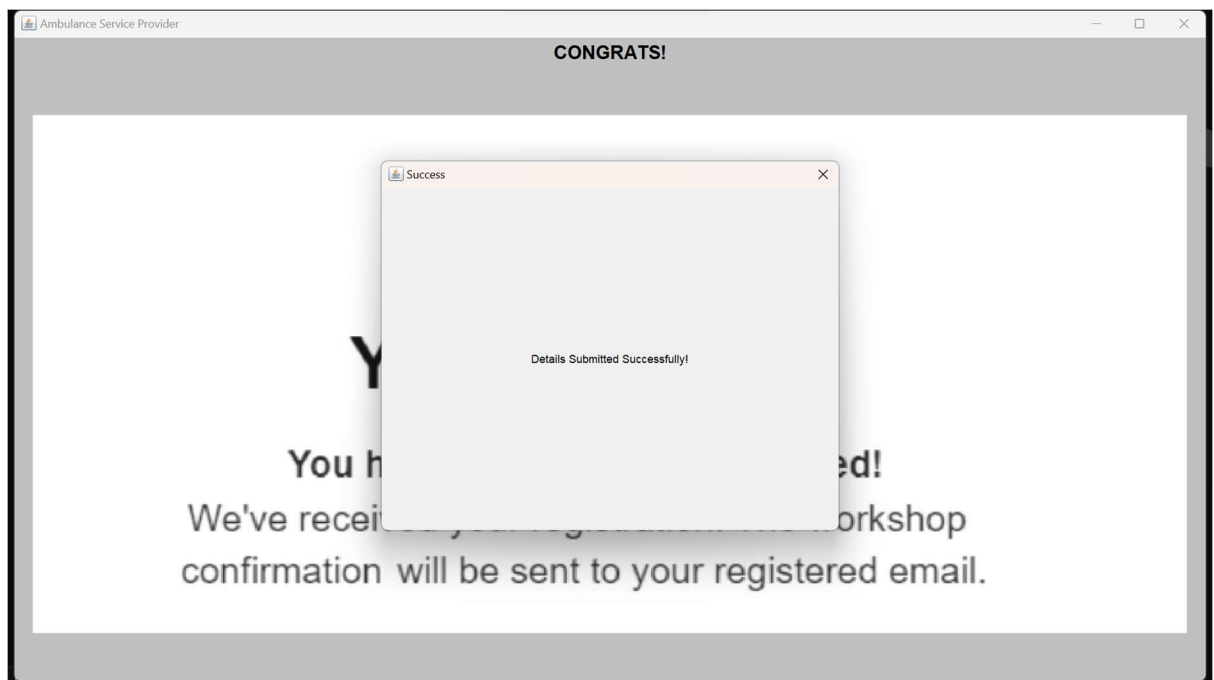
II. Register page..

A screenshot of a web browser window titled "Ambulance Service Provider". The page is titled "Registration Page" and features a registration form on the left and a large illustration of a healthcare worker in a blue scrubs, white cap, and blue face mask on the right. The illustration has the text "FRONTLINE HEROES" in a white box. The registration form includes the following fields: "Name:" (text input), "Email:" (text input), "City:" (text input), "Gender:" with radio buttons for "Male" and "Female", "Phone:" (text input), "Select:" with radio buttons for "Driver" and "Helper", "Address:" (text input), and "Blood Group" with a dropdown menu showing "O-". At the bottom of the form are "Submit" and "Cancel" buttons.

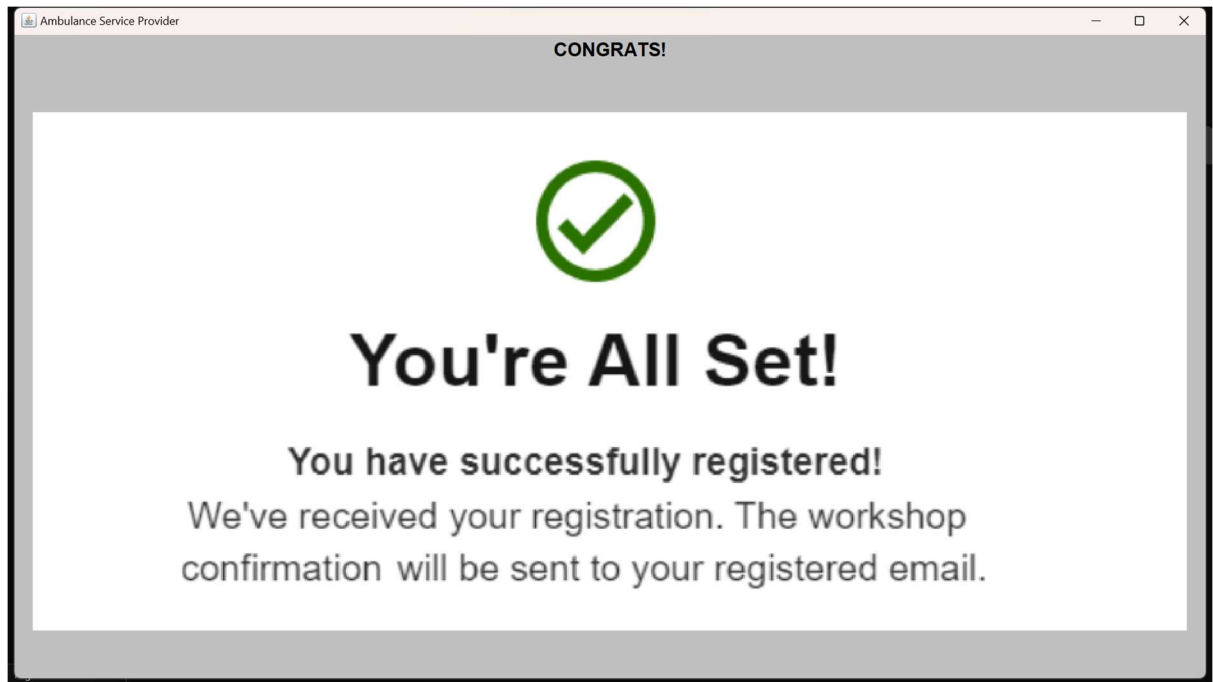
III. If not entering any details and submitting details .



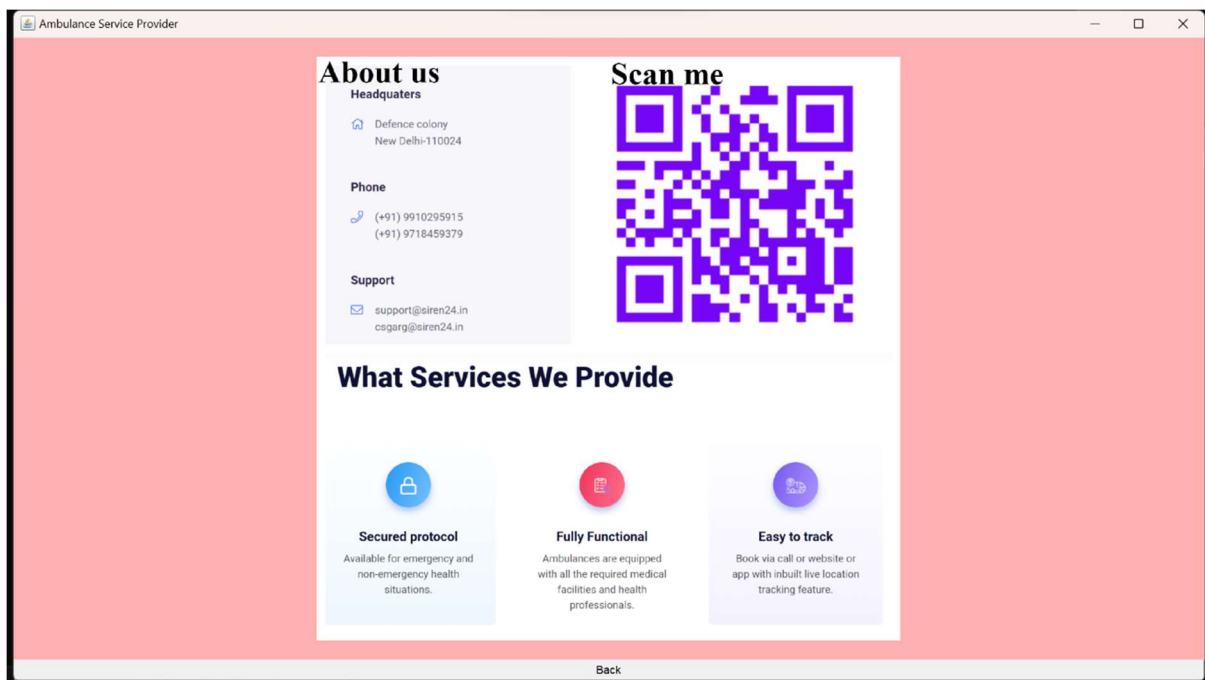
IV. After submitting details in registration page.



V. Successfully registered message.

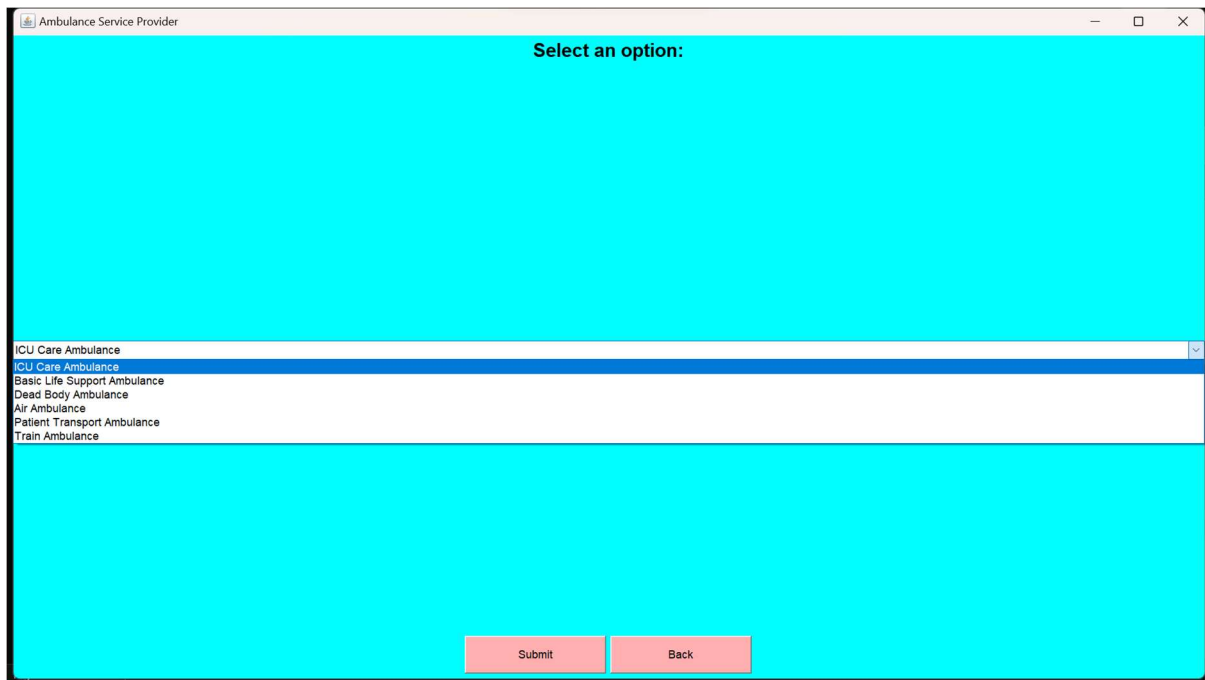


VI. Clicking aboutus button in Welcome page.



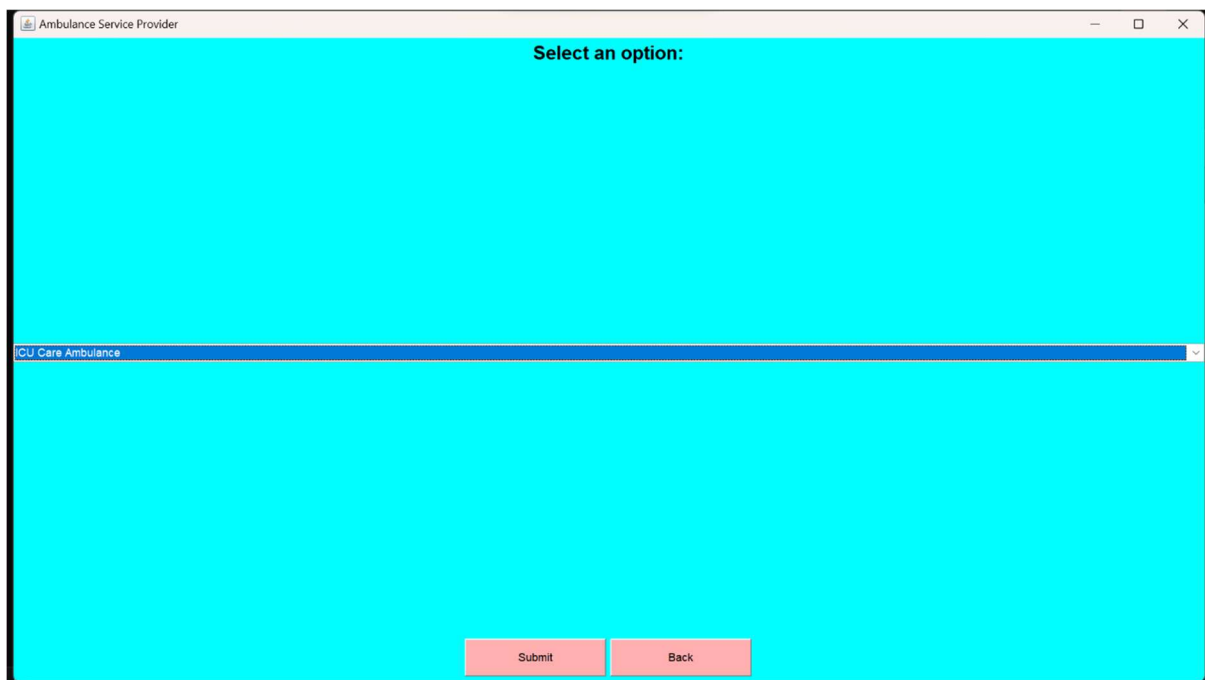
VII. To get ambulance click 'get ambulance'

Button..

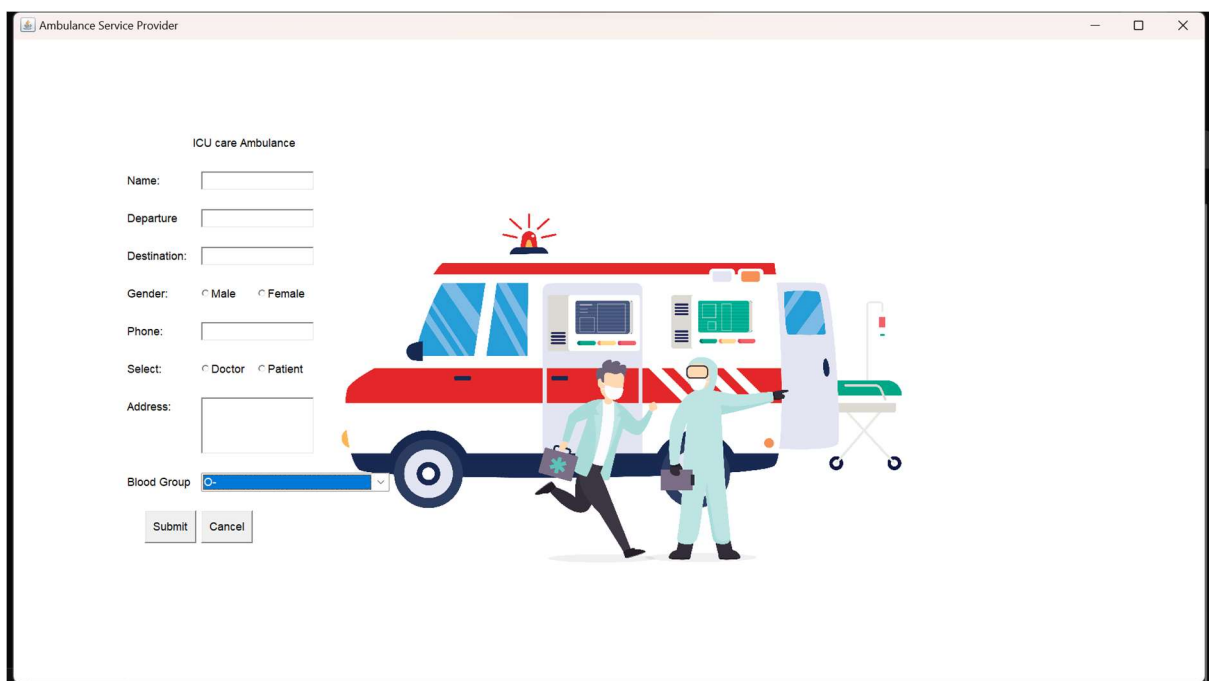
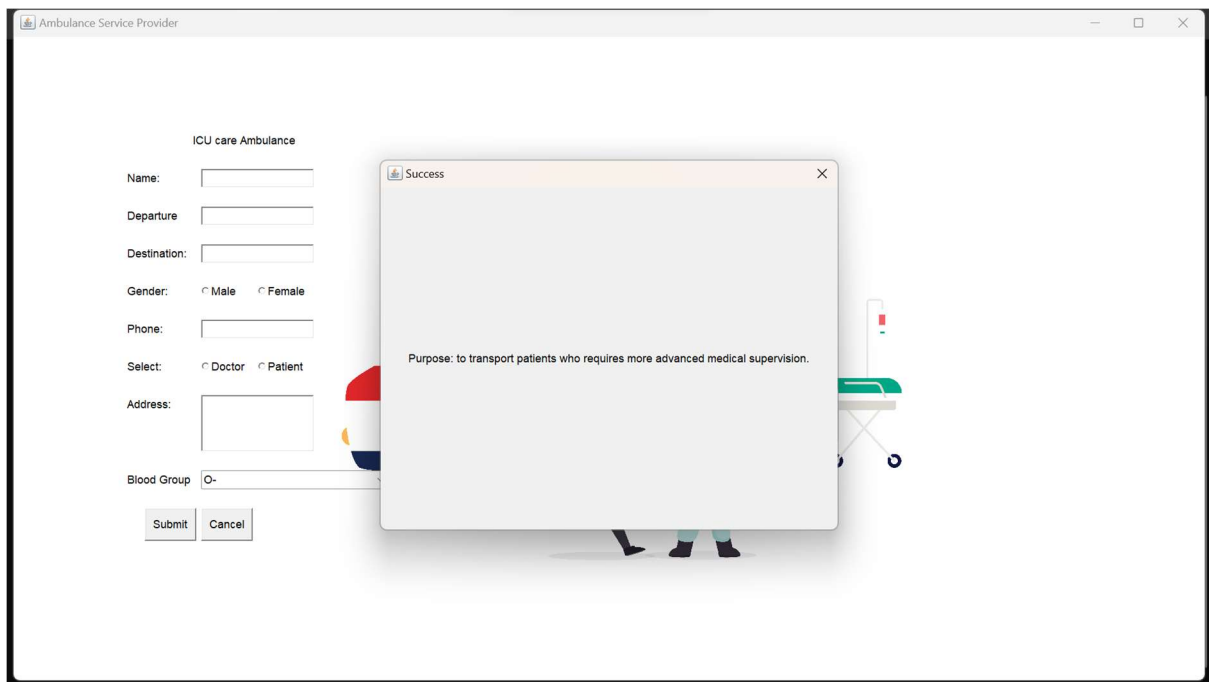


A screenshot of a web application window titled "Ambulance Service Provider". The window has a light blue header bar with the title and standard window controls (minimize, maximize, close). Below the header, the main content area is white. At the top of the content area, the text "Select an option:" is displayed. Below this text is a dropdown menu. The dropdown menu is open, showing a list of ambulance types: "ICU Care Ambulance", "Basic Life Support Ambulance", "Dead Body Ambulance", "Air Ambulance", "Patient Transport Ambulance", and "Train Ambulance". The "ICU Care Ambulance" option is highlighted with a blue background. At the bottom of the window, there are two red buttons: "Submit" and "Back".

VIII. Select the type of ambulance u need, and click 'submit' button
1. ICU care ambulance.



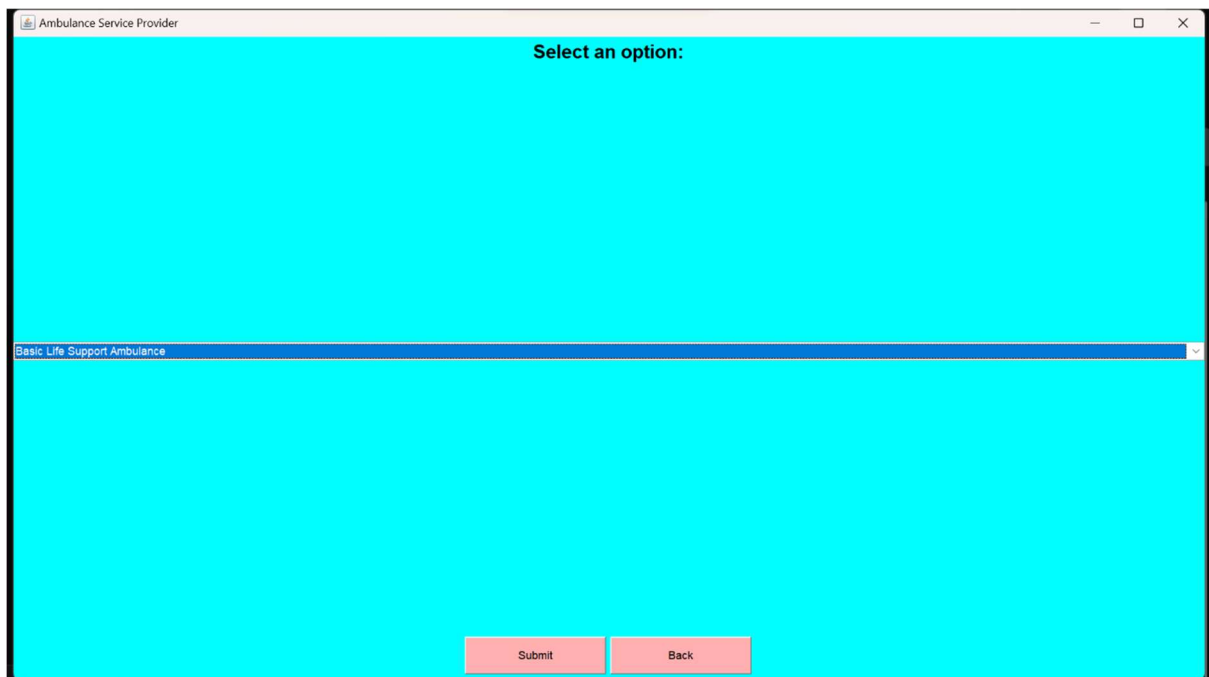
A screenshot of the same web application window titled "Ambulance Service Provider". The window has a light blue header bar with the title and standard window controls. Below the header, the main content area is white. At the top of the content area, the text "Select an option:" is displayed. Below this text is a dropdown menu. The dropdown menu is open, showing a list of ambulance types: "ICU Care Ambulance", "Basic Life Support Ambulance", "Dead Body Ambulance", "Air Ambulance", "Patient Transport Ambulance", and "Train Ambulance". The "ICU Care Ambulance" option is highlighted with a blue background. At the bottom of the window, there are two red buttons: "Submit" and "Back".

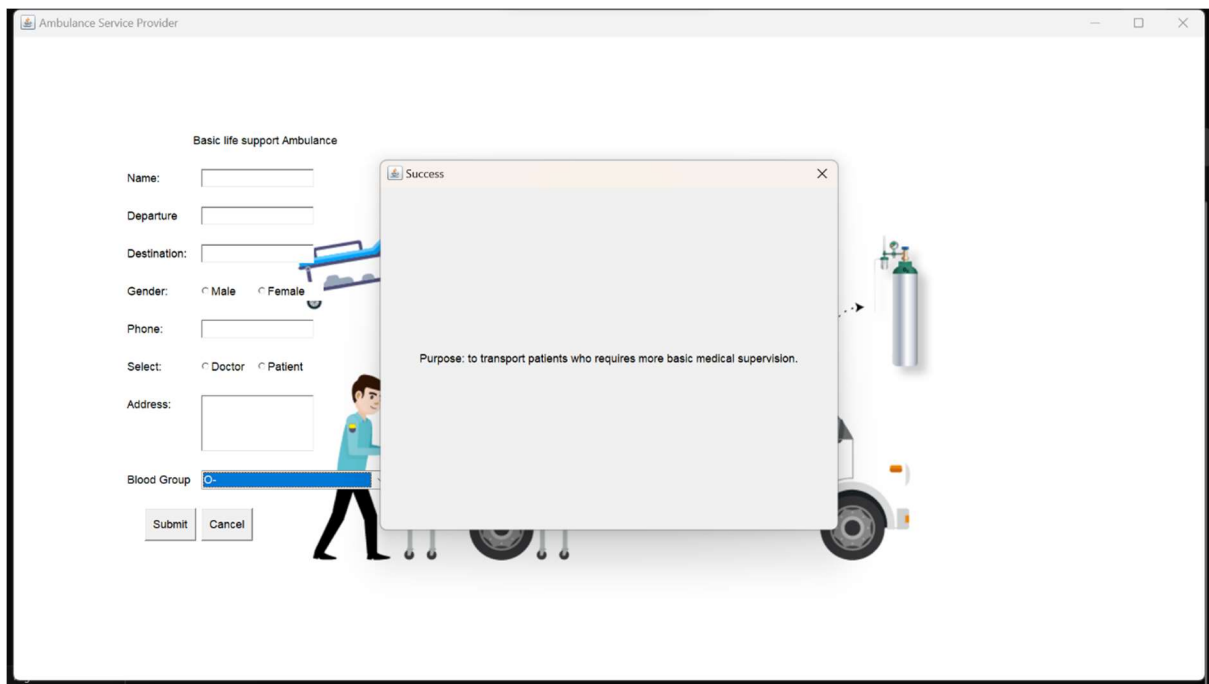


After submitting details.



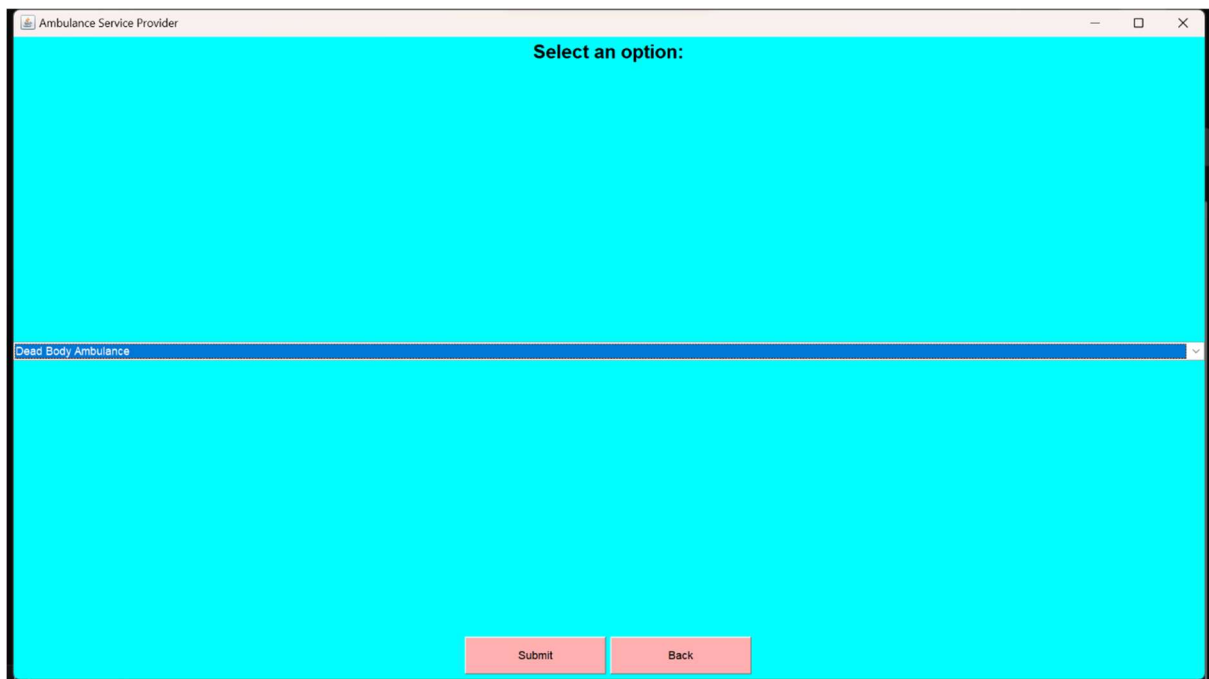
2. Basic Life Support Ambulance.







3. Dead Body Ambulance.



Ambulance Service Provider

Dead Body Ambulance

Name:

Departure

Destination:

Gender:

Male

Female

Phone:

Select:

Patient

Address:

Blood Group

O-

Submit

Cancel

Success

Purpose: to transport the dead body of a person to the place of last rites.

Ambulance Service Provider

Dead Body Ambulance

Name:

Departure

Destination:

Gender:

Male

Female

Phone:

Select:

Patient


Address:

Blood Group

O-

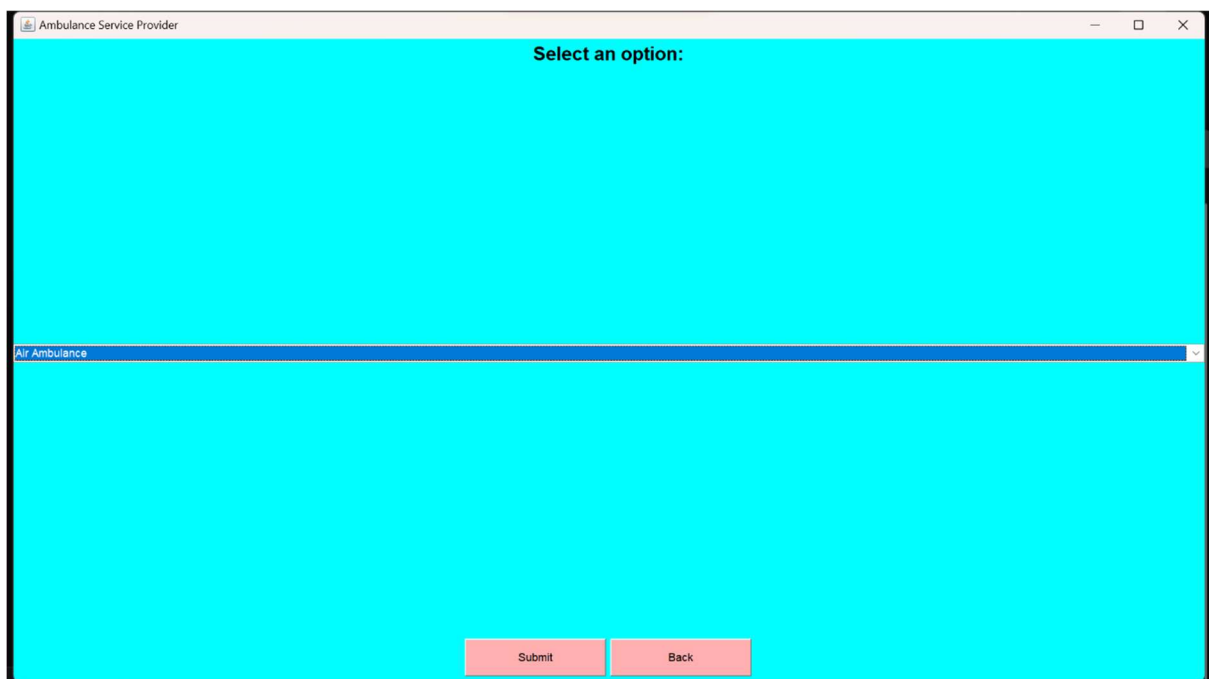
Submit

Cancel





4. Air ambulance.



Ambulance Service Provider

Air Ambulance

Name:

Departure

Destination:

Gender:

Male

Female

Phone:

Select:

Doctor

Patient

Address:

Blood Group

O-

Submit

Cancel

Success

Purpose: to transport patients from one palce to another far away place through air.

Ambulance Service Provider

Air Ambulance

Name:

Departure

Destination:

Gender:

Male

Female

Phone:

Select:

Doctor

Patient

Address:

Blood Group

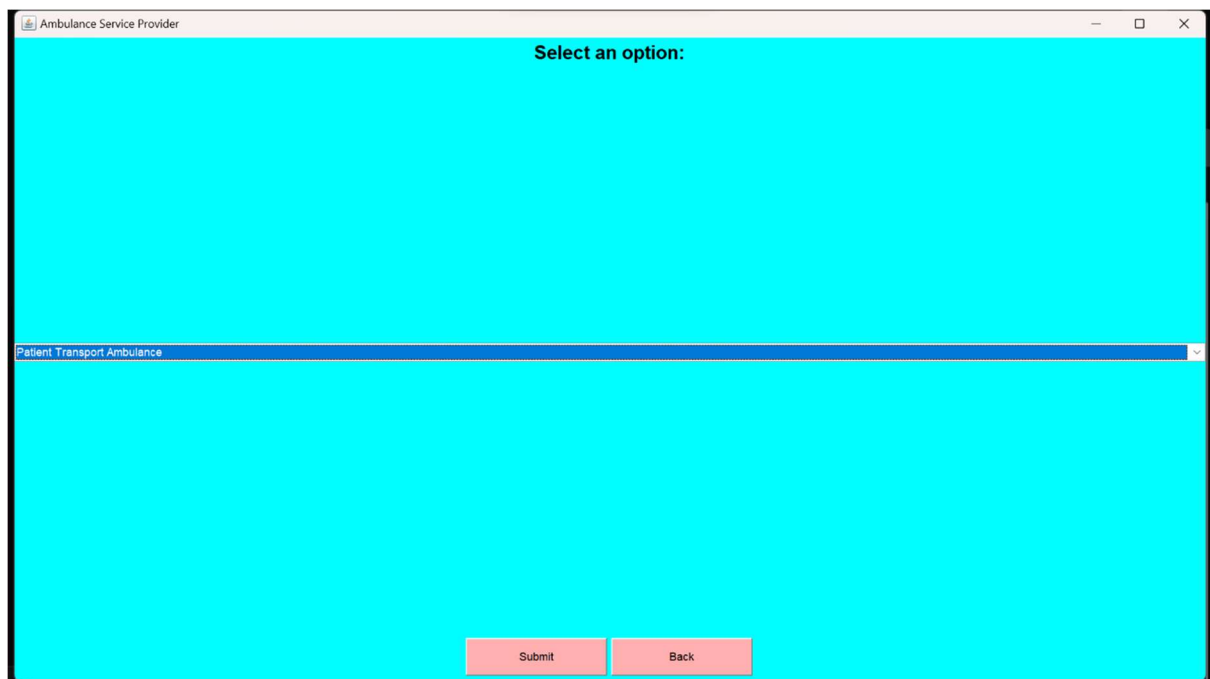
O-

Submit

Cancel



5. Patient Transport Ambulance.



Ambulance Service Provider

Patient transport Ambulance

Name:

Departure:

Destination:

Gender: ☐ Male ☐ Female

Phone:

Select: ☐ Doctor ☐ Patient

Address:

Blood Group:

Success

Purpose: to transport patients who have to transport from one hospital to another.

Ambulance Service Provider

Patient transport Ambulance

Name:

Departure:

Destination:

Gender: ☐ Male ☐ Female

Phone:

Select: ☐ Doctor ☐ Patient

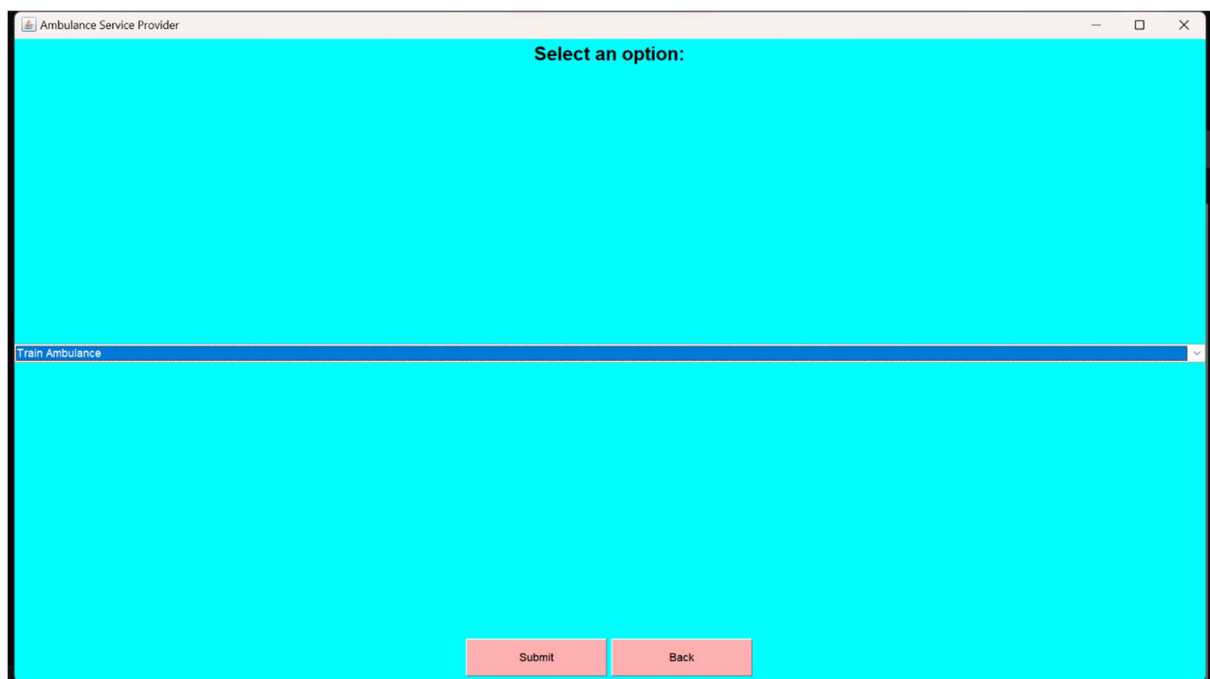
Address:

Blood Group:





6. Train Ambulance.



Ambulance Service Provider

Train Ambulance

Name:

Departure

Destination:

Gender:

☐ Male

☐ Female

Phone:

Select:

☐ Doctor

☐ Patient

Address:

Blood Group

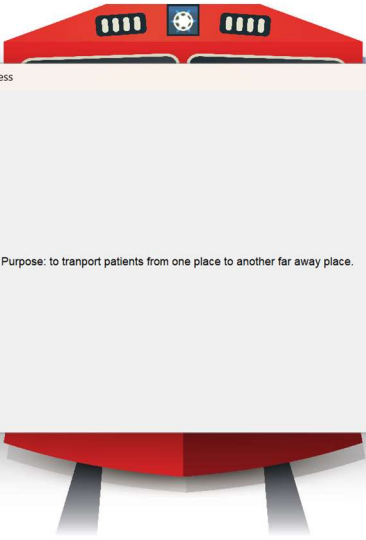
O-

Submit

Cancel

Success

Purpose: to tranport patients from one place to another far away place.



Ambulance Service Provider

Train Ambulance

Name:

Departure

Destination:

Gender:

☒ Male

☐ Female

Phone:

Select:

☐ Doctor

☐ Patient

Address:

Blood Group

O-

Submit

Cancel

