

# CHAPTER 1

## INTRODUCTION

### 1.1 Topic Introduction

This project on Airline Management System is the automation of registration process to airline system. The system is able to provide much information like passenger's details, flight details and the booking details. The system allows us to add records when a passenger reserves a ticket. It also allows deleting and updating the records when based on passenger's requirements. For data storage and retrieval, we use the MySQL database. It enables us to add any number of records in our database from the frontend which is Java core. Any changes made in the frontend will be reflected at the backend.

### 1.2 Objective of the Project

The main objective of the proposed electronically handling of flight's record to enhance the accuracy, flexibility, reliability and to remove the human's error. An airline provides air transport services for passengers, generally with a recognize operating. To provide accurate information about the addition, deletion and modified record. To provide, efficient, accurate, reliable, fast, and robust structure that can handle number of records. The global airline industry continues to grow rapidly, but consistent and robust profitability is elusive.

### 1.3 Motivation for the Project

1. Operational Efficiency: Airlines handle thousands of bookings and cancellations requests daily. An automated system can streamline these processes, reducing errors and improving response times.
  
2. Customer Convenience: A digital reservation system allows passengers to search for flights, check seat availability, book tickets, and manage bookings without needing physical assistance, thus enhancing customer experience.

3. Cost Reduction: By automating many of the manual tasks involved in ticket reservation and management, airlines can reduce their operational costs, particularly customer service and support areas.
4. Data Management and Reporting: An effective reservation system can collect valuable data on booking patterns, peak times, and customer preferences. These insights can inform business decisions, improve flight schedules, and tailor services to customer needs.
5. Error Reduction: Manual systems are prone to errors in bookings and scheduling. A computerized system reduces the chances of double bookings, missed schedules, and other errors that could impact both the airline's revenue and customer satisfaction.

## **1.4 Project Overview**

Airline Reservation System contains the details about flight schedules and its fare tariffs, passenger reservations and ticket records. Air Alliance operates flights to 6 destinations in India namely Mysore, Bangalore, Goa, Mumbai, Pune, Delhi.

## **1.5 Literature Survey**

Every Software development requires the survey process. The Survey process is needed to get the requirement for the software. The Survey also consists of studying the present system and also studying about the tools needed for the development of the software. A proper understanding of the tools is very much essential. Following is an extract of the information of the material collected during literature survey.

## Airline Reservation Are Here to Help

The Airline Reservation System (ARS) is designed to streamline the process of booking flights, managing customer reservations, and improving the overall travel experience. By automating tasks such as scheduling, seat selection, and ticketing, ARS simplifies the booking process for both customers and airline staff.

The system aims to:

- Provide Efficiency: Automate routine tasks to reduce booking times and improve service speed.
- Enhance Accessibility: Make booking and schedule information readily available to customers, allowing for seamless flight reservations.
- Ensure Data Security: Protect customer information through robust security protocols, ensuring that sensitive data remains confidential.
- Improve Customer Experience: Enable customers to easily view available flights, select seats and manage their travel plans online.

## Designing for Airline Reservation

Literature suggests a three-tier architecture, including the presentation layer, business logic layer, and data layer, as the most effective approach for modularity and scalability. The presentation layer, often a web or desktop interface, enables user interactions; the business logic layer manages core functions like booking, ticketing, and scheduling; while the data layer interfaces with a relational database to store and manage data.

## CHAPTER 2

# SYSTEM ANALYSIS

### 2.1 Introduction

The airline reservation system is a crucial software application designed to facilitate the reservation and management of flights for an airline. This project, implemented in Java, aims to simulate the core functionalities of a real-world airline booking system. The system allows users to search for flights, book tickets, manage bookings, and view flight information, providing a streamlined and efficient process for both passengers and administrators.

### 2.2 Scope of Project

#### Objective:

- To develop an efficient airline reservation system that allows users to books and cancels the tickets in real time.
- The system should be user-friendly, secure, and reliable, with a focus on key functionalities such as ticket booking, customer management, and flight management.

#### Key Functionalities:

- **User Authentication:**
  - Implement login and registration for passengers.
  - Use secure authentication mechanisms for user identity verification.
- **Flight Search and Booking:**
  - Allow users to search for available flights by criteria such as date, time, source, and destination.
  - Display flight details, such as departure/arrival times, and ticket prices.
  - Booking confirmation.

- **Ticket Management:**

- Provide options for viewing and cancelling booked tickets.
- Generate a unique ticket ID (PNR) for each booking.
- Include payment simulation for completing a booking.

- **Customer Profile Management:**

- Allow users to view their profile details.

## **2.3 Hardware / Software Requirements**

### **2.3.1 Hardware Requirement:**

- Processor : i3+
- RAM : 4GB
- Hard Disk : 80GB
- Speed : 1.2 GHz+

### **2.3.2 Software Requirement:**

- **Operating System Front End Back End Server Documentation:**  
Windows 10
- **Frontend Software:** Eclipse, JDK 8.
- **Backend Software:** MySQL.

## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 Introduction

The Airline Reservation System will be designed to allow users to view and book available seats, manage reservations, and allow admins to update routes and schedules. This involves structuring both high-level architecture and detailed design elements to ensure efficiency, maintainability, and scalability.

#### 3.2 Description:

##### Registration:

- Registration if the customer wants to book the flight ticket, then he/she must be registered.
- Unregistered users will not be allowed to access the site.

##### Login:

- The passenger who has registered can login to the system by entering the valid user id and password.
- If admin logins, they can add or remove flights, can change the price and the timings.
- If users logins, they can book the ticket from their particular location to their desired destination.

#### 3.3 Selection of Source and destination:

The passenger will be able to choose the source and destination.

##### Available Flights:

After the selection of the source, destination, and date the available buses for the specific route along with the time will be shown to the user.

##### Ticket Booking:

The passenger will be able to book the tickets as per the destination, source, time and date.

[Type here]

### **Logout:**

After the payment of the ticket, the customer will be logged out.

### **Report Generation:**

After all transactions, the system will generate the online ticket and it will be stored in the system database to maintain the records of the passengers along with the traveling details.

## **3.4 Proposed System**

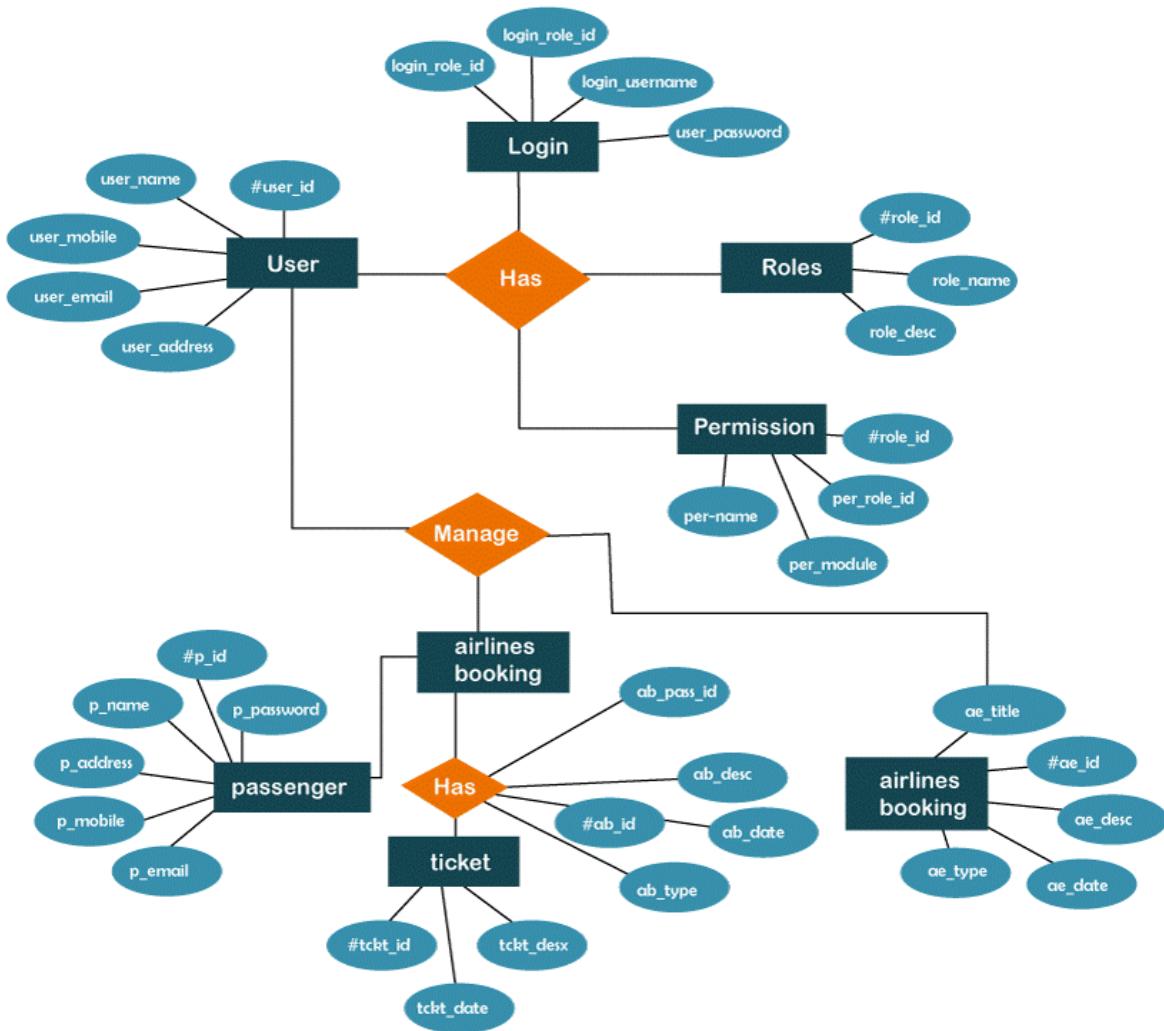
The proposed airline reservation system is an automated, user-friendly platform that allows users to book flight tickets online and manage reservations. For administrators, it provides tools for managing schedules and flight details, streamlining operations and reducing manual errors.

**Enhanced User Experience:** Provide an intuitive interface where users can easily search for available flights, view seat availability, make reservations, and receive immediate booking confirmations.

**Real-Time Data:** Ensure that seat availability and booking statuses are updated in real-time to avoid double bookings and improve user satisfaction.

**Security:** Safeguard user information and payment data by implementing secure login and encrypted data storage.

### 3.5 ER Diagram

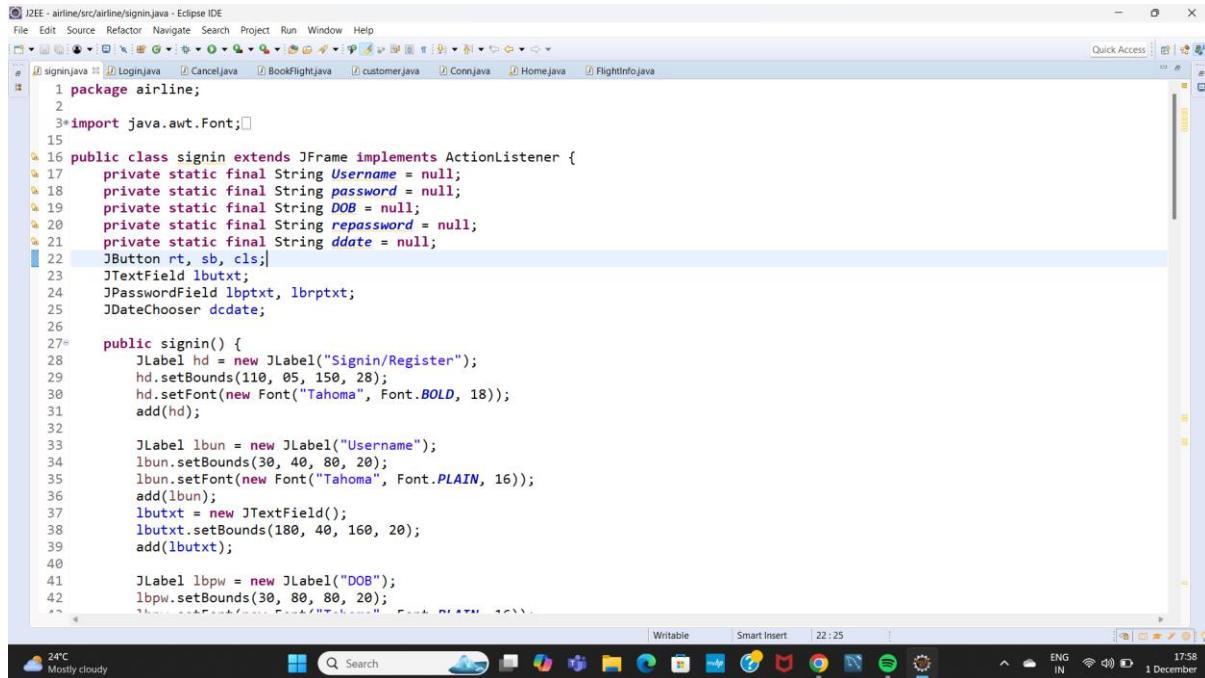


# CHAPTER 4

## System Implementation

### 4.1 Pseudo code

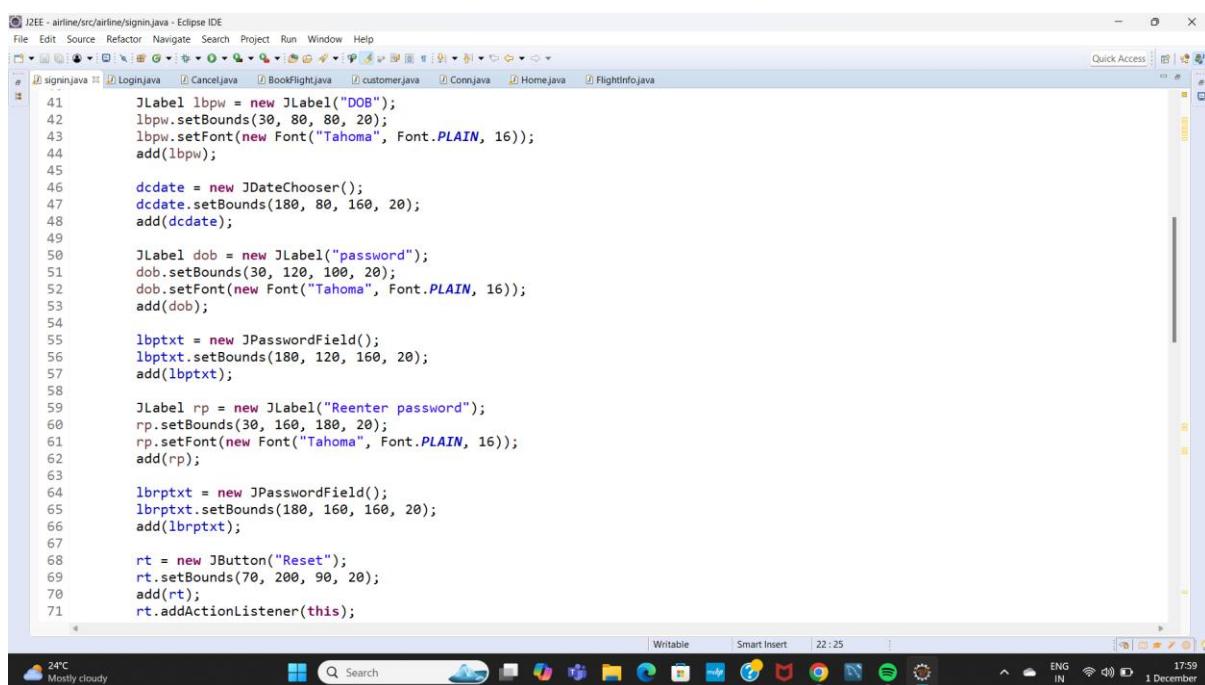
#### 1. Sign in page



```

J2EE - airline/src/airline/signin.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
signin.java Loginjava Canceljava BookFlightJava customer.java Connjava Home.java FlightInfo.java
1 package airline;
2
3 *import java.awt.Font;*
4
5
6 public class signin extends JFrame implements ActionListener {
7     private static final String Username = null;
8     private static final String password = null;
9     private static final String DOB = null;
10    private static final String repassword = null;
11    private static final String ddate = null;
12    JButton rt, sb, cls;
13    JTextField lbutxt;
14    JPasswordField lbptxt, lbrptxt;
15    JDateChooser dcdate;
16
17    public signin() {
18        JLabel hd = new JLabel("Signin/Register");
19        hd.setBounds(110, 05, 150, 28);
20        hd.setFont(new Font("Tahoma", Font.BOLD, 18));
21        add(hd);
22
23        JLabel l bun = new JLabel("Username");
24        l bun.setBounds(30, 40, 80, 20);
25        l bun.setFont(new Font("Tahoma", Font.PLAIN, 16));
26        add(l bun);
27        l butxt = new JTextField();
28        l butxt.setBounds(180, 40, 160, 20);
29        add(l butxt);
30
31        JLabel l bpw = new JLabel("DOB");
32        l bpw.setBounds(30, 80, 80, 20);
33
34        l brptxt = new JPasswordField();
35        l brptxt.setBounds(180, 80, 160, 20);
36
37        l rptxt = new JPasswordField();
38        l rptxt.setBounds(180, 120, 160, 20);
39        add(l rptxt);
40
41        JButton rt = new JButton("Reset");
42        rt.setBounds(70, 200, 90, 20);
43        add(rt);
44        rt.addActionListener(this);
45
46        dcdate = new JDateChooser();
47        dcdate.setBounds(180, 80, 160, 20);
48        add(dcdate);
49
50        JLabel dob = new JLabel("password");
51        dob.setBounds(30, 120, 100, 20);
52        dob.setFont(new Font("Tahoma", Font.PLAIN, 16));
53        add(dob);
54
55        l brptxt = new JPasswordField();
56        l brptxt.setBounds(180, 120, 160, 20);
57        add(l brptxt);
58
59        JLabel rp = new JLabel("Reenter password");
60        rp.setBounds(30, 160, 180, 20);
61        rp.setFont(new Font("Tahoma", Font.PLAIN, 16));
62        add(rp);
63
64        l brptxt = new JPasswordField();
65        l brptxt.setBounds(180, 160, 160, 20);
66        add(l brptxt);
67
68        rt = new JButton("Reset");
69        rt.setBounds(70, 200, 90, 20);
70        add(rt);
71        rt.addActionListener(this);

```

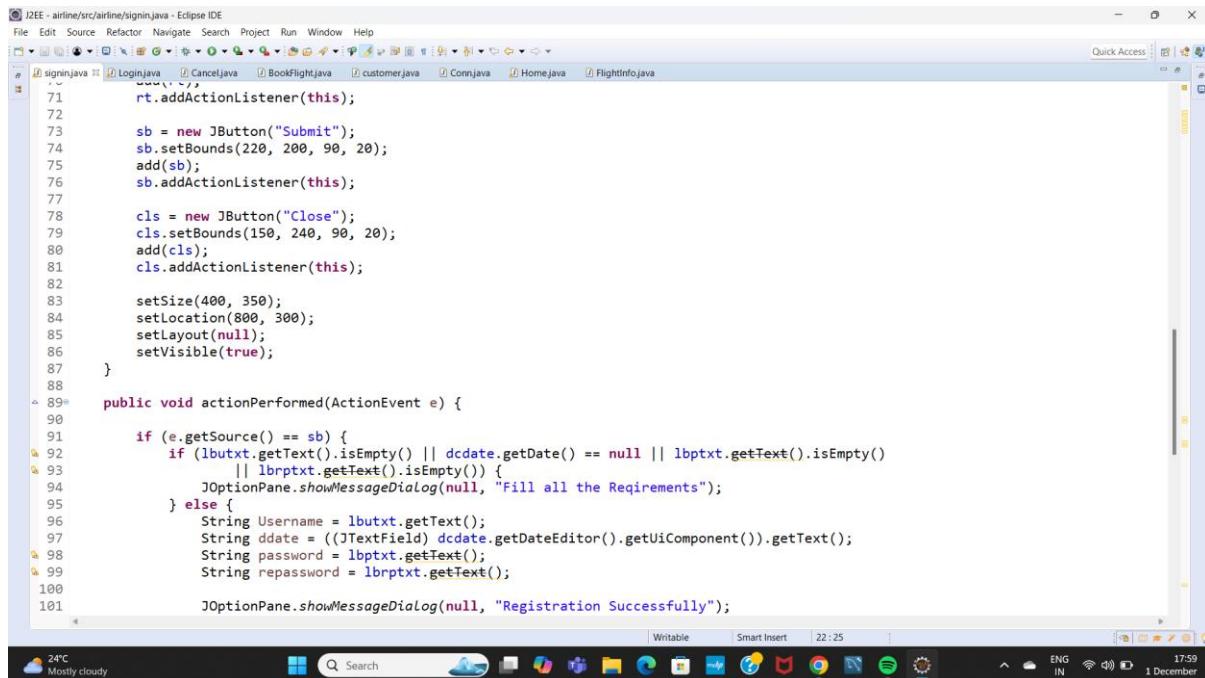


```

J2EE - airline/src/airline/signin.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
signin.java Loginjava Canceljava BookFlightJava customer.java Connjava Home.java FlightInfo.java
1 package airline;
2
3 *import java.awt.Font;*
4
5
6 public class signin extends JFrame implements ActionListener {
7     private static final String Username = null;
8     private static final String password = null;
9     private static final String DOB = null;
10    private static final String repassword = null;
11    private static final String ddate = null;
12    JButton rt, sb, cls;
13    JTextField lbutxt;
14    JPasswordField lbptxt, lbrptxt;
15    JDateChooser dcdate;
16
17    public signin() {
18        JLabel hd = new JLabel("Signin/Register");
19        hd.setBounds(110, 05, 150, 28);
20        hd.setFont(new Font("Tahoma", Font.BOLD, 18));
21        add(hd);
22
23        JLabel l bun = new JLabel("Username");
24        l bun.setBounds(30, 40, 80, 20);
25        l bun.setFont(new Font("Tahoma", Font.PLAIN, 16));
26        add(l bun);
27        l butxt = new JTextField();
28        l butxt.setBounds(180, 40, 160, 20);
29        add(l butxt);
30
31        JLabel l bpw = new JLabel("DOB");
32        l bpw.setBounds(30, 80, 80, 20);
33
34        l brptxt = new JPasswordField();
35        l brptxt.setBounds(180, 80, 160, 20);
36
37        l rptxt = new JPasswordField();
38        l rptxt.setBounds(180, 120, 160, 20);
39        add(l rptxt);
40
41        JButton rt = new JButton("Reset");
42        rt.setBounds(70, 200, 90, 20);
43        add(rt);
44        rt.addActionListener(this);
45
46        dcdate = new JDateChooser();
47        dcdate.setBounds(180, 80, 160, 20);
48        add(dcdate);
49
50        JLabel dob = new JLabel("password");
51        dob.setBounds(30, 120, 100, 20);
52        dob.setFont(new Font("Tahoma", Font.PLAIN, 16));
53        add(dob);
54
55        l brptxt = new JPasswordField();
56        l brptxt.setBounds(180, 120, 160, 20);
57        add(l brptxt);
58
59        JLabel rp = new JLabel("Reenter password");
60        rp.setBounds(30, 160, 180, 20);
61        rp.setFont(new Font("Tahoma", Font.PLAIN, 16));
62        add(rp);
63
64        l brptxt = new JPasswordField();
65        l brptxt.setBounds(180, 160, 160, 20);
66        add(l brptxt);
67
68        rt = new JButton("Reset");
69        rt.setBounds(70, 200, 90, 20);
70        add(rt);
71        rt.addActionListener(this);

```

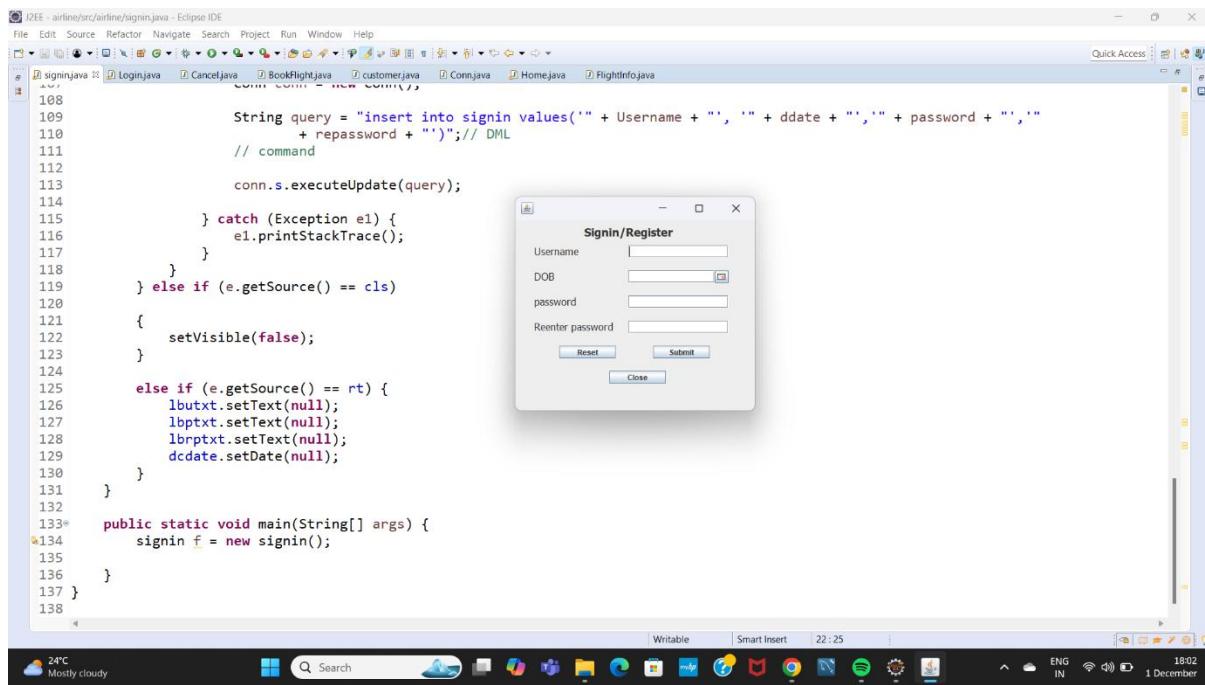
[Type here]



I2EE - airline/src/airline/signin.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
signin.java Loginjava Canceljava BookFlightjava customerjava Connjava Homejava FlightInfojava
71 rt.addActionListener(this);
72
73 sb = new JButton("Submit");
74 sb.setBounds(220, 200, 90, 20);
75 add(sb);
76 sb.addActionListener(this);
77
78 cls = new JButton("Close");
79 cls.setBounds(150, 240, 90, 20);
80 add(cls);
81 cls.addActionListener(this);
82
83 setSize(400, 350);
84 setLocation(800, 300);
85 setLayout(null);
86 setVisible(true);
87
88
89 public void actionPerformed(ActionEvent e) {
90
91     if (e.getSource() == sb) {
92         if (lbutxt.getText().isEmpty() || ddate.getDate() == null || lbrptxt.getText().isEmpty()
93             || lbrptxt.getText().isEmpty()) {
94             JOptionPane.showMessageDialog(null, "Fill all the Requirements");
95         } else {
96             String Username = lbutxt.getText();
97             String ddate = ((JTextField) ddate.getDateEditor().getUiComponent()).getText();
98             String password = lbrptxt.getText();
99             String repassword = lbrptxt.getText();
100
101             JOptionPane.showMessageDialog(null, "Registration Successfully");
102         }
103     }
104 }
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
```

24°C Mostly cloudy    Search    Writable    Smart Insert    22:25    ENG IN    17:59    1 December



I2EE - airline/src/airline/signin.java - Eclipse IDE

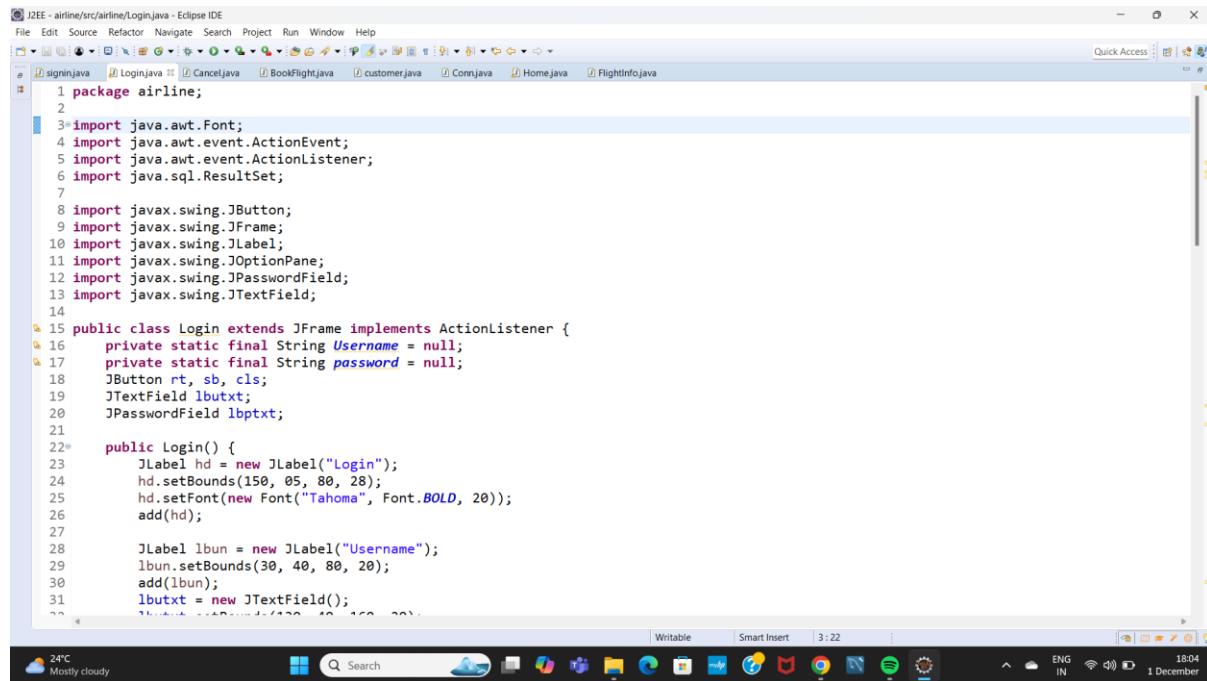
```
File Edit Source Refactor Navigate Search Project Run Window Help
signin.java Loginjava Canceljava BookFlightjava customerjava Connjava Homejava FlightInfojava
109
110     String query = "insert into signin values('"+Username+"','"+ddate+"','"+password+"','"+repassword+"')"; // DML
111     // command
112
113     conn.s.executeUpdate(query);
114
115     } catch (Exception e1) {
116         e1.printStackTrace();
117     }
118 } else if (e.getSource() == cls)
119 {
120     setVisible(false);
121 }
122 else if (e.getSource() == rt) {
123     lbutxt.setText(null);
124     lbrptxt.setText(null);
125     lbrptxt.setText(null);
126     ddate.setDate(null);
127 }
128
129
130
131
132
133
134
135
136
137
138
```

24°C Mostly cloudy    Search    Writable    Smart Insert    22:25    ENG IN    18:02    1 December

The screenshot shows the Java code for the `signin` class. The code handles user input from four buttons: `Submit`, `Close`, and two `Reset` buttons. It checks for empty fields and invalid dates. If successful, it inserts data into a `signin` table. A `JOptionPane` dialog box is used for both success and failure messages. The code also includes logic to clear form fields and reset the date editor.

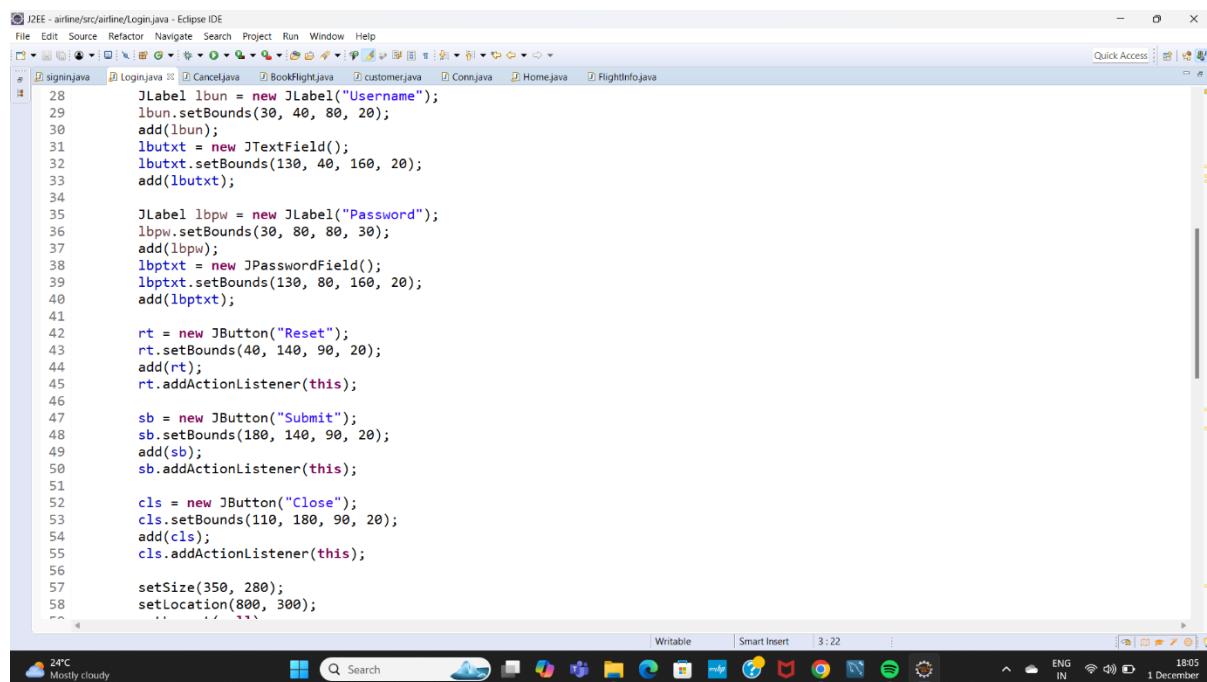
[Type here]

## 2.Login page



J2EE - airline/src/airline/Login.java - Eclipse IDE

```
1 package airline;
2
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.sql.ResultSet;
7
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JPasswordField;
13 import javax.swing.JTextField;
14
15 public class Login extends JFrame implements ActionListener {
16     private static final String Username = null;
17     private static final String password = null;
18     JButton rt, sb, cls;
19     JTextField lbutxt;
20     JPasswordField lpptxt;
21
22     public Login() {
23         JLabel hd = new JLabel("Login");
24         hd.setBounds(150, 05, 80, 28);
25         hd.setFont(new Font("Tahoma", Font.BOLD, 20));
26         add(hd);
27
28         JLabel lbun = new JLabel("Username");
29         lbun.setBounds(30, 40, 80, 20);
30         add(lbun);
31         lbutxt = new JTextField();
32         lbutxt.setBounds(130, 40, 160, 20);
33         add(lbutxt);
34
35         JLabel lbpw = new JLabel("Password");
36         lbpw.setBounds(30, 80, 80, 30);
37         add(lbpw);
38         lpptxt = new JPasswordField();
39         lpptxt.setBounds(130, 80, 160, 20);
40         add(lpptxt);
41
42         rt = new JButton("Reset");
43         rt.setBounds(40, 140, 90, 20);
44         add(rt);
45         rt.addActionListener(this);
46
47         sb = new JButton("Submit");
48         sb.setBounds(180, 140, 90, 20);
49         add(sb);
50         sb.addActionListener(this);
51
52         cls = new JButton("Close");
53         cls.setBounds(110, 180, 90, 20);
54         add(cls);
55         cls.addActionListener(this);
56
57         setSize(350, 280);
58         setLocation(800, 300);
59     }
60 }
```



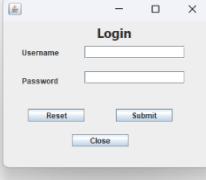
J2EE - airline/src/airline/Login.java - Eclipse IDE

```
1 package airline;
2
3 import java.awt.Font;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.sql.ResultSet;
7
8 import javax.swing.JButton;
9 import javax.swing.JFrame;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JPasswordField;
13 import javax.swing.JTextField;
14
15 public class Login extends JFrame implements ActionListener {
16     private static final String Username = null;
17     private static final String password = null;
18     JButton rt, sb, cls;
19     JTextField lbutxt;
20     JPasswordField lpptxt;
21
22     public Login() {
23         JLabel lbun = new JLabel("Username");
24         lbun.setBounds(30, 40, 80, 20);
25         add(lbun);
26         lbutxt = new JTextField();
27         lbutxt.setBounds(130, 40, 160, 20);
28         add(lbutxt);
29
30         JLabel lbpw = new JLabel("Password");
31         lbpw.setBounds(30, 80, 80, 30);
32         add(lbpw);
33         lpptxt = new JPasswordField();
34         lpptxt.setBounds(130, 80, 160, 20);
35         add(lpptxt);
36
37         rt = new JButton("Reset");
38         rt.setBounds(40, 140, 90, 20);
39         add(rt);
40         rt.addActionListener(this);
41
42         sb = new JButton("Submit");
43         sb.setBounds(180, 140, 90, 20);
44         add(sb);
45         sb.addActionListener(this);
46
47         cls = new JButton("Close");
48         cls.setBounds(110, 180, 90, 20);
49         add(cls);
50         cls.addActionListener(this);
51
52         setSize(350, 280);
53         setLocation(800, 300);
54     }
55 }
```

[Type here]

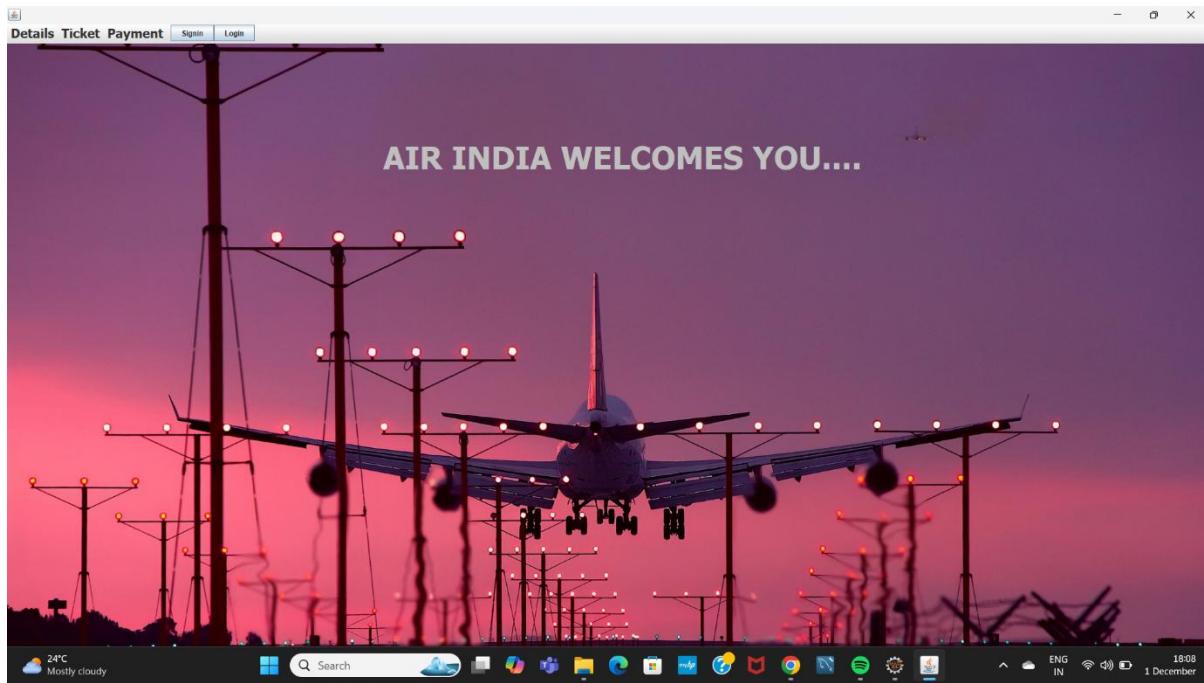
```
J2EE - airline/src/airline/Login.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
signIn.java Login.java CancelJava BookFlightJava customerJava Conn.java HomeJava FlightInfoJava
55
56     setSize(350, 280);
57     setLocation(800, 300);
58     setLayout(null);
59     setVisible(true);
60 }
61
62 public void actionPerformed(ActionEvent e) {
63     if (e.getSource() == sb) {
64         if (lbutxt.getText().isEmpty() || lbptxt.getText().isEmpty()) {
65             JOptionPane.showMessageDialog(null, "Fill all the Requirements");
66         } else {
67             String Username = lbutxt.getText();
68             String password = lbptxt.getText();
69
70             try {
71                 Conn c = new Conn();
72
73                 String query = "select * from signin where username='" + Username + "' and password='" + password
74                         + "'";
75                 ResultSet rs = c.s.executeQuery(query);
76
77                 if (rs.next()) {
78                     new Home();
79                     setVisible(false);
80                 } else {
81                     setVisible(false);
82                     JOptionPane.showMessageDialog(null, "Invalid user password");
83                     setVisible(false);
84                 }
85             } catch (Exception e2) {
86
87         }
88     }
89 }
90 } else if (e.getSource() == cls) {
91     setVisible(false);
92 }
93
94 else if (e.getSource() == rt) {
95     lbutxt.setText(null);
96     lbptxt.setText(null);
97 }
98
99
100
101 public static void main(String[] args) {
102     Login f = new Login();
103
104 }
105 }
```

```
J2EE - airline/src/airline/Login.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
signIn.java Login.java CancelJava BookFlightJava customerJava Conn.java HomeJava FlightInfoJava
76     ResultSet rs = c.s.executeQuery(query);
77
78     if (rs.next()) {
79         new Home();
80         setVisible(false);
81     } else {
82         setVisible(false);
83         JOptionPane.showMessageDialog(null, "Invalid user password");
84         setVisible(false);
85     }
86 } catch (Exception e2) {
87     e2.printStackTrace();
88 }
89
90 } else if (e.getSource() == cls) {
91     setVisible(false);
92 }
93
94 else if (e.getSource() == rt) {
95     lbutxt.setText(null);
96     lbptxt.setText(null);
97 }
98
99
100
101 public static void main(String[] args) {
102     Login f = new Login();
103
104 }
105 }
```



[Type here]

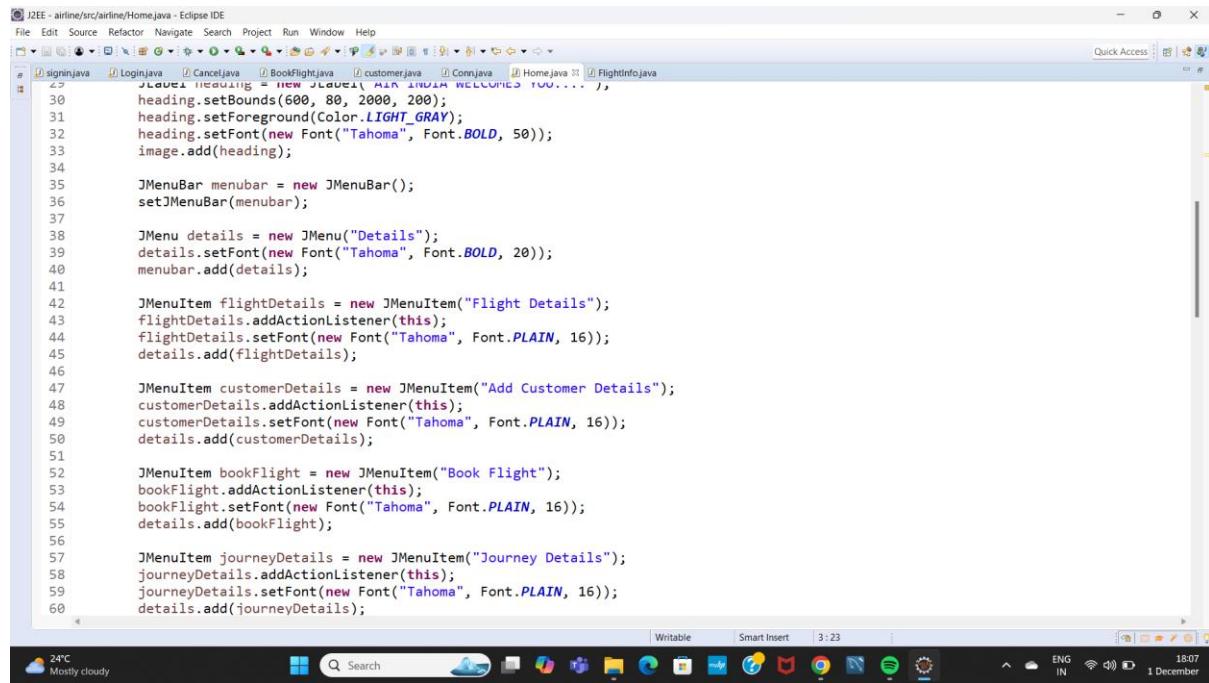
### 3. Home page

A screenshot of the Eclipse IDE showing the Java code for the "Home.java" class. The code imports various Java Swing components and defines a class "Home" that extends "JFrame" and implements "ActionListener". It includes code to set up a background image of an aircraft and a welcome message label.

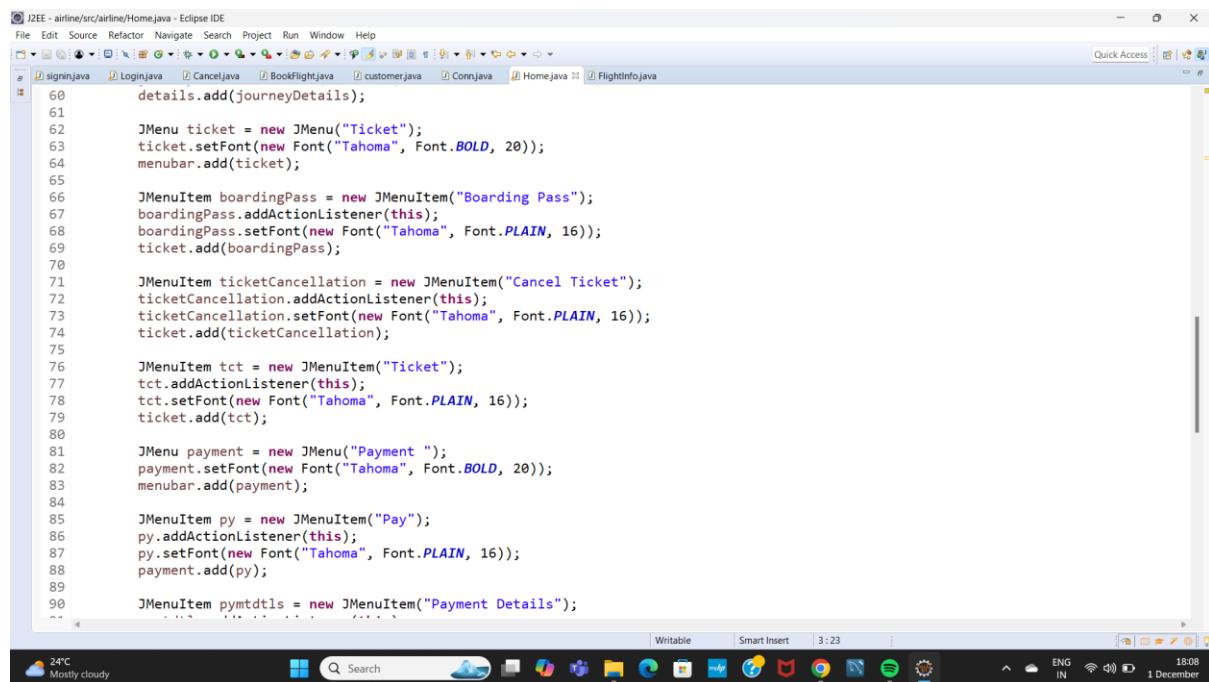
```
1 package airline;
2
3 import java.awt.Color;
4 import java.awt.Font;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 import javax.swing.ImageIcon;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JMenu;
13 import javax.swing.JMenuBar;
14 import javax.swing.JMenuItem;
15
16 public class Home extends JFrame implements ActionListener {
17
18     JButton lg, si;
19
20     public Home() {
21         setLayout(null);
22
23         ImageIcon ii = new ImageIcon(getClass().getResource("aircraft.jpg"));
24         JLabel image = new JLabel(ii);
25         image.setBounds(0, 0, 1920, 980);
26         add(image);
27         setVisible(true);
28
29         JLabel heading = new JLabel("AIR INDIA WELCOMES YOU....");
30         heading.setBounds(600, 80, 2000, 200);
31         heading.setForeground(Color.LIGHT_GRAY);
32 }
```

The taskbar at the bottom shows various application icons and the date and time as 1 December 18:07.

[Type here]



```
29     JLabel heading = new JLabel("AIR INDIA WELCOMES YOU....");
30     heading.setBounds(600, 80, 2000, 200);
31     heading.setForeground(Color.LIGHT_GRAY);
32     heading.setFont(new Font("Tahoma", Font.BOLD, 50));
33     image.add(heading);
34
35     JMenuBar menubar = new JMenuBar();
36     setJMenuBar(menubar);
37
38     JMenu details = new JMenu("Details");
39     details.setFont(new Font("Tahoma", Font.BOLD, 20));
40     menubar.add(details);
41
42     JMenuItem flightDetails = new JMenuItem("Flight Details");
43     flightDetails.addActionListener(this);
44     flightDetails.setFont(new Font("Tahoma", Font.PLAIN, 16));
45     details.add(flightDetails);
46
47     JMenuItem customerDetails = new JMenuItem("Add Customer Details");
48     customerDetails.addActionListener(this);
49     customerDetails.setFont(new Font("Tahoma", Font.PLAIN, 16));
50     details.add(customerDetails);
51
52     JMenuItem bookFlight = new JMenuItem("Book Flight");
53     bookFlight.addActionListener(this);
54     bookFlight.setFont(new Font("Tahoma", Font.PLAIN, 16));
55     details.add(bookFlight);
56
57     JMenuItem journeyDetails = new JMenuItem("Journey Details");
58     journeyDetails.addActionListener(this);
59     journeyDetails.setFont(new Font("Tahoma", Font.PLAIN, 16));
60     details.add(journeyDetails);
```



```
60     details.add(journeyDetails);
61
62     JMenu ticket = new JMenu("Ticket");
63     ticket.setFont(new Font("Tahoma", Font.BOLD, 20));
64     menubar.add(ticket);
65
66     JMenuItem boardingPass = new JMenuItem("Boarding Pass");
67     boardingPass.addActionListener(this);
68     boardingPass.setFont(new Font("Tahoma", Font.PLAIN, 16));
69     ticket.add(boardingPass);
70
71     JMenuItem ticketCancellation = new JMenuItem("Cancel Ticket");
72     ticketCancellation.addActionListener(this);
73     ticketCancellation.setFont(new Font("Tahoma", Font.PLAIN, 16));
74     ticket.add(ticketCancellation);
75
76     JMenuItem tct = new JMenuItem("Ticket");
77     tct.addActionListener(this);
78     tct.setFont(new Font("Tahoma", Font.PLAIN, 16));
79     ticket.add(tct);
80
81     JMenu payment = new JMenu("Payment ");
82     payment.setFont(new Font("Tahoma", Font.BOLD, 20));
83     menubar.add(payment);
84
85     JMenuItem py = new JMenuItem("Pay");
86     py.addActionListener(this);
87     py.setFont(new Font("Tahoma", Font.PLAIN, 16));
88     payment.add(py);
89
90     JMenuItem pymtdtls = new JMenuItem("Payment Details");
```

[Type here]

```
J2EE - airline/src/airline/Home.java - Eclipse IDE
File Edit Source Refactor Navigate Search Run Window Help
Quick Access
# signin.java LoginJava CancelJava BookFlightJava customer.java Connjava Home.java FlightInfo.java
90 JMenuItem pymtdtls = new JMenuItem("Payment Details");
91 pymtdtls.addActionListener(this);
92 pymtdtls.setFont(new Font("Tahoma", Font.PLAIN, 16));
93 payment.add(pymtdtls);
94
95 si = new JButton("Signin");
96 menubar.add(si);
97 si.addActionListener(this);
98
99 lg = new JButton("Login");
100 menubar.add(lg);
101 lg.addActionListener(this);
102
103 setExtendedState(JFrame.MAXIMIZED_BOTH);
104 setVisible(true);
105 }
106
107 public void actionPerformed(ActionEvent ae) {
108 if (ae.getSource() == si) {
109 new signin();
110 }
111 if (ae.getSource() == lg) {
112 new Login();
113 }
114 String text = ae.getActionCommand();
115
116 if (text.equals("Add Customer Details")) {
117 new customer();
118 } else if (text.equals("Flight Details")) {
119 new FlightInfo();
120 } else if (text.equals("Book Flight")) {
```

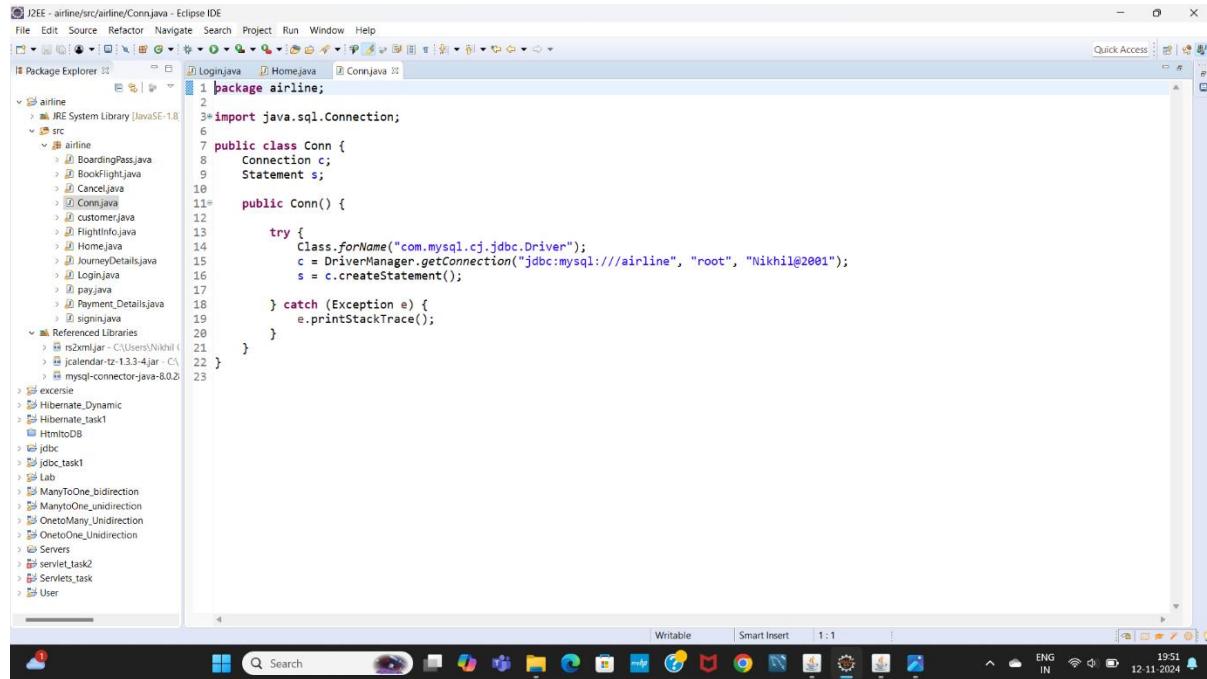
J2EE - airline/src/airline/Home.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
signIn.java Login.java CancelJava BookFlightJava customerJava ConnJava Home.java FlightInfo.java

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == si) {
        new signIn();
    }
    if (ae.getSource() == lg) {
        new Login();
    }
    String text = ae.getActionCommand();
    if (text.equals("Add Customer Details")) {
        new customer();
    } else if (text.equals("Flight Details")) {
        new FlightInfo();
    } else if (text.equals("Book Flight")) {
        new BookFlight();
    } else if (text.equals("Journey Details")) {
        new JourneyDetails();
    } else if (text.equals("Cancel Ticket")) {
        new Cancel();
    } else if (text.equals("Boarding Pass")) {
        new BoardingPass();
    } else if (text.equals("Payment Details")) {
        new Payment_Details();
    }
}
public static void main(String[] args) {
    new Home();
}
```

[Type here]

## 4. Connection page



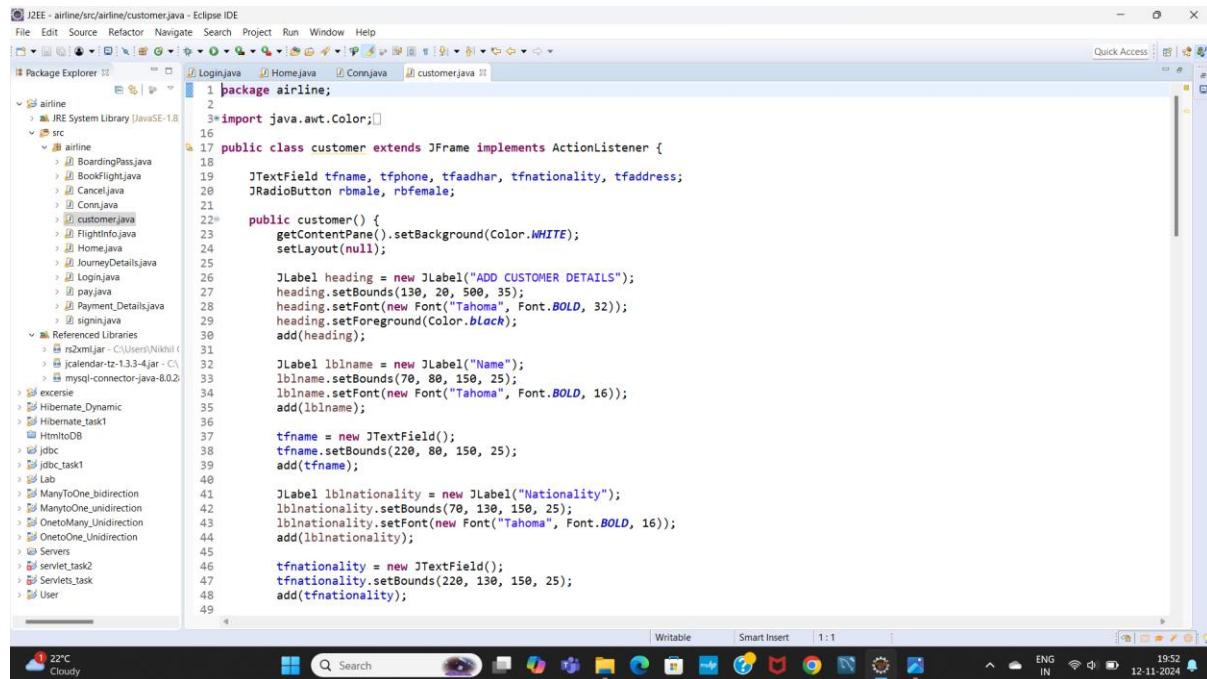
```
J2EE - airline/src/airline/Conn.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Conn.java Homejava Connjava
1 package airline;
2
3 import java.sql.Connection;
4
5 public class Conn {
6     Connection c;
7     Statement s;
8
9     public Conn() {
10
11         try {
12             Class.forName("com.mysql.cj.jdbc.Driver");
13             c = DriverManager.getConnection("jdbc:mysql://airline", "root", "Nikhil@2001");
14             s = c.createStatement();
15
16         } catch (Exception e) {
17             e.printStackTrace();
18         }
19     }
20 }
21
22 }

Referenced Libraries
> n2oMljar - C:\Users\Nikhil\...
> calendar-tz-1.3.4.jar - C...
> mysql-connector-java-8.0.2

excercises
> Hibernate_Dynamic
> Hibernate_task1
HmmitoDB
> jdbc
> jdbctask1
> Lab
> ManyToOne_bidirection
> ManyToOne_unidirection
> OntoMany_Unidirection
> OneToOne_Unidirection
> Servers
> servlet_task2
> Servlets_task
> User

Writable Smart Insert 1:1
19:51 ENG IN 12-11-2024
```

## 5. Customer page



```
J2EE - airline/src/airline/customer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Conn.java Homejava Connjava customer.java
1 package airline;
2
3 import java.awt.Color;
4
5 public class customer extends JFrame implements ActionListener {
6
7     JTextField tfname, tfphone, tfaadhar, tfnationality, tfaddress;
8     JRadioButton rbmale, rbfemale;
9
10    public customer() {
11        getContentPane().setBackground(Color.WHITE);
12        setLayout(null);
13
14        JLabel heading = new JLabel("ADD CUSTOMER DETAILS");
15        heading.setBounds(130, 20, 500, 35);
16        heading.setFont(new Font("Tahoma", Font.BOLD, 32));
17        heading.setForeground(Color.black);
18        add(heading);
19
20        JLabel lblname = new JLabel("Name");
21        lblname.setBounds(70, 80, 150, 25);
22        lblname.setFont(new Font("Tahoma", Font.BOLD, 16));
23        add(lblname);
24
25        JTextField tfname = new JTextField();
26        tfname.setBounds(220, 80, 150, 25);
27        add(tfname);
28
29        JLabel lblnationality = new JLabel("Nationality");
30        lblnationality.setBounds(70, 130, 150, 25);
31        lblnationality.setFont(new Font("Tahoma", Font.BOLD, 16));
32        add(lblnationality);
33
34        JTextField tfnationality = new JTextField();
35        tfnationality.setBounds(220, 130, 150, 25);
36        tfnationality.setFont(new Font("Tahoma", Font.BOLD, 16));
37        add(tfnationality);
38
39        JButton b1 = new JButton("Submit");
40        b1.setBounds(200, 180, 100, 30);
41        add(b1);
42
43        JButton b2 = new JButton("Cancel");
44        b2.setBounds(300, 180, 100, 30);
45        add(b2);
46
47        JButton b3 = new JButton("Back");
48        b3.setBounds(400, 180, 100, 30);
49        add(b3);
50
51    }
52
53    public void actionPerformed(ActionEvent e) {
54        if (e.getSource() == b1) {
55            String name = tfname.getText();
56            String nationality = tfnationality.getText();
57
58            JOptionPane.showMessageDialog(null, "Customer Added Successfully!");
59        }
60    }
61
62 }

Referenced Libraries
> n2oMljar - C:\Users\Nikhil\...
> calendar-tz-1.3.4.jar - C...
> mysql-connector-java-8.0.2

excercises
> Hibernate_Dynamic
> Hibernate_task1
HmmitoDB
> jdbc
> jdbctask1
> Lab
> ManyToOne_bidirection
> ManyToOne_unidirection
> OntoMany_Unidirection
> OneToOne_Unidirection
> Servers
> servlet_task2
> Servlets_task
> User

Writable Smart Insert 1:1
22°C Cloudy 19:52 ENG IN 12-11-2024
```

[Type here]

J2EE - airline/src/airline/customer.java - Eclipse IDE

```
45 tfnationality = new JTextField();
46 tfnationality.setBounds(220, 130, 150, 25);
47 add(tfnationality);
48
49 JLabel lblaadhar = new JLabel("Aadhar Number");
50 lblaadhar.setBounds(70, 180, 150, 25);
51 lblaadhar.setFont(new Font("Tahoma", Font.BOLD, 16));
52 add(lblaadhar);
53
54 JTextField tfaadhar = new JTextField();
55 tfaadhar.setBounds(220, 180, 150, 25);
56 add(tfaadhar);
57
58 JLabel lbladdress = new JLabel("Address");
59 lbladdress.setBounds(70, 230, 150, 25);
60 lbladdress.setFont(new Font("Tahoma", Font.BOLD, 16));
61 add(lbladdress);
62
63 JTextField tfaddress = new JTextField();
64 tfaddress.setBounds(220, 230, 150, 25);
65 add(tfaddress);
66
67 JLabel lblgender = new JLabel("Gender");
68 lblgender.setBounds(70, 280, 150, 25);
69 lblgender.setFont(new Font("Tahoma", Font.BOLD, 16));
70 add(lblgender);
71
72 ButtonGroup gendergroup = new ButtonGroup(); // for selecting only one button
73 JRadioButton rbmale = new JRadioButton("Male"); // for round shape & check box for box shape
74 rbmale.setBounds(220, 280, 70, 25);
75 rbmale.setBackground(Color.WHITE);
76 add(rbmale);
77
78 JRadioButton rbfemale = new JRadioButton("Female");
79 rbfemale.setBounds(300, 280, 70, 25);
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
```

Writable Smart Insert 1:1

22°C Cloudy 19:52 12-11-2024 ENG IN

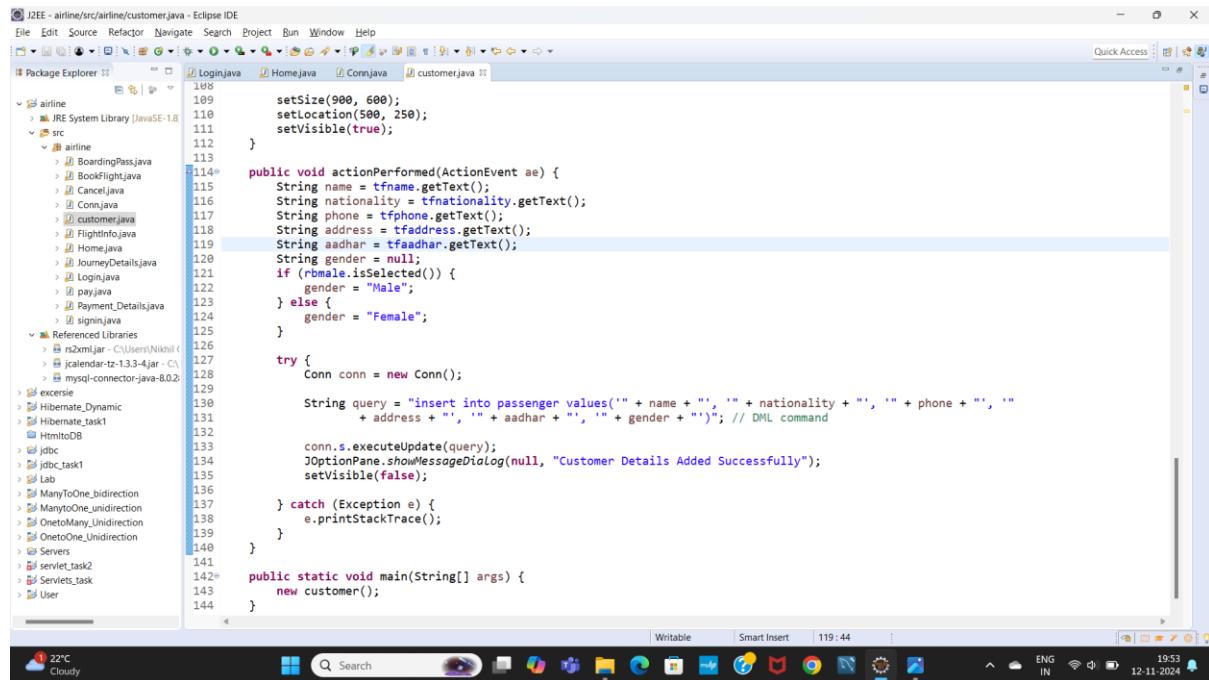
J2EE - airline/src/airline/customer.java - Eclipse IDE

```
80 rbfemale = new JRadioButton("Female");
81 rbfemale.setBounds(300, 280, 70, 25);
82 rbfemale.setBackground(Color.WHITE);
83 add(rbfemale);
84
85 gendergroup.add(rbmale);
86 gendergroup.add(rbfemale);
87
88 JLabel lblphone = new JLabel("Phone");
89 lblphone.setBounds(70, 330, 150, 25);
90 lblphone.setFont(new Font("Tahoma", Font.BOLD, 16));
91 add(lblphone);
92
93 JTextField tfphone = new JTextField();
94 tfphone.setBounds(220, 330, 150, 25);
95 add(tfphone);
96
97 JButton save = new JButton("Submit");
98 save.setBackground(Color.BLACK);
99 save.setForeground(Color.WHITE);
100 save.setBounds(220, 380, 150, 30);
101 save.addActionListener(this);
102 add(save);
103
104 ImageIcon i1 = new ImageIcon(getClass().getResource("custm.PNG"));
105 JLabel image = new JLabel(i1);
106 image.setBounds(430, 10, 200, 300);
107 add(image);
108
109 setSize(900, 600);
110 setLocation(500, 250);
111 setVisible(true);
112
113
114
115
116
```

Writable Smart Insert 1:1

22°C Cloudy 19:52 12-11-2024 ENG IN

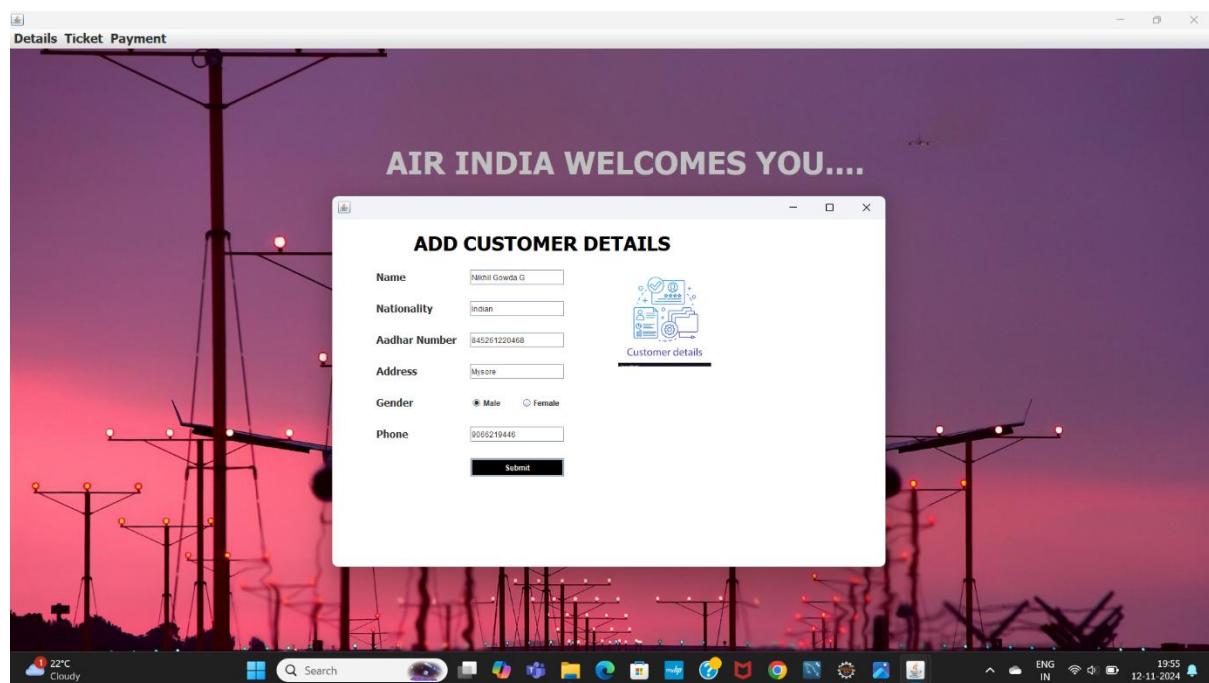
[Type here]



The screenshot shows the Eclipse IDE interface with the project 'J2EE - airline/src/airline/customer.java' open. The code in the editor is as follows:

```
108     setSize(900, 600);
109     setLocation(500, 250);
110     setVisible(true);
111 }
112 }
113
114 public void actionPerformed(ActionEvent ae) {
115     String name = fname.getText();
116     String nationality = tfnationality.getText();
117     String phone = tfphone.getText();
118     String address = tfaddress.getText();
119     String aadhar = tfaadhar.getText();
120     String gender = null;
121     if (rbmale.isSelected()) {
122         gender = "Male";
123     } else {
124         gender = "Female";
125     }
126     try {
127         Conn conn = new Conn();
128         String query = "insert into passenger values('"+name+"','"+nationality+"','"+phone+"','"+address+"','"+aadhar+"','"+gender+"')"; // DML command
129         conn.s.executeUpdate(query);
130         JOptionPane.showMessageDialog(null, "Customer Details Added Successfully");
131         setVisible(false);
132     } catch (Exception e) {
133         e.printStackTrace();
134     }
135 }
136
137 public static void main(String[] args) {
138     new customer();
139 }
140
141
142
143 }
144 }
```

The code implements a Java Swing application for adding customer details. It uses JDBC to insert data into a 'passenger' table. The application has fields for Name, Nationality, Aadhar Number, Address, Gender, and Phone. A 'Conn' class is used for database connectivity.



[Type here]

## 6. Flight Details

J2EE - airline/src/airline/FlightInfo.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Package Explorer

I Package Explorer

1 package airline;

2

3 import java.awt.Color;

4

5 public class FlightInfo extends JFrame {

6

7 public FlightInfo() {

8

9 getContentPane().setBackground(Color.WHITE);

10 setLayout(null);

11

12 JTable table = new JTable();

13

14 try {

15 Conn conn = new Conn();

16

17 ResultSet rs = conn.s.executeQuery("select \* from f\_details");

18 table.setModel(dbutils.resultSetToTableModel(rs));

19 } catch (Exception e) {

20 e.printStackTrace();

21 }

22

23 JScrollPane jsp = new JScrollPane(table);

24 jsp.setBounds(0, 0, 800, 500);

25 add(jsp);

26

27 setSize(800, 180);

28 setLocation(600, 400);

29 setVisible(true);

30 }

31

32 public static void main(String[] args) {

33 new FlightInfo();

34 }

35 }

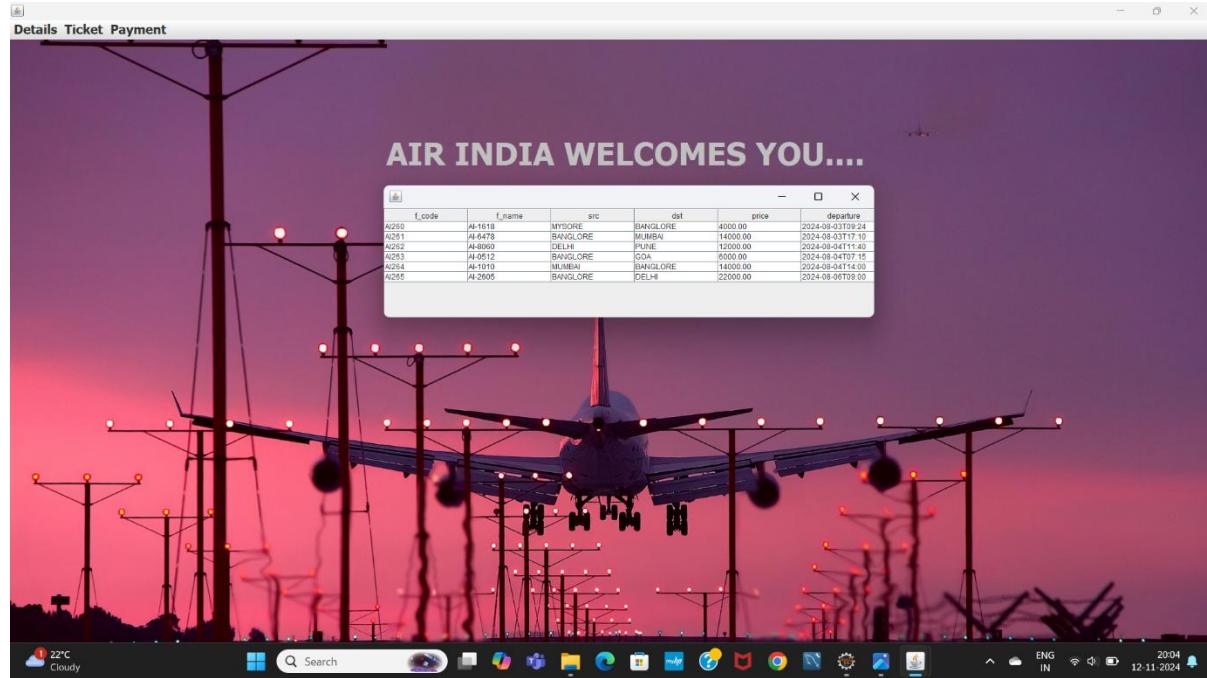
Writable Smart Insert 1:1

22°C Cloudy

Search

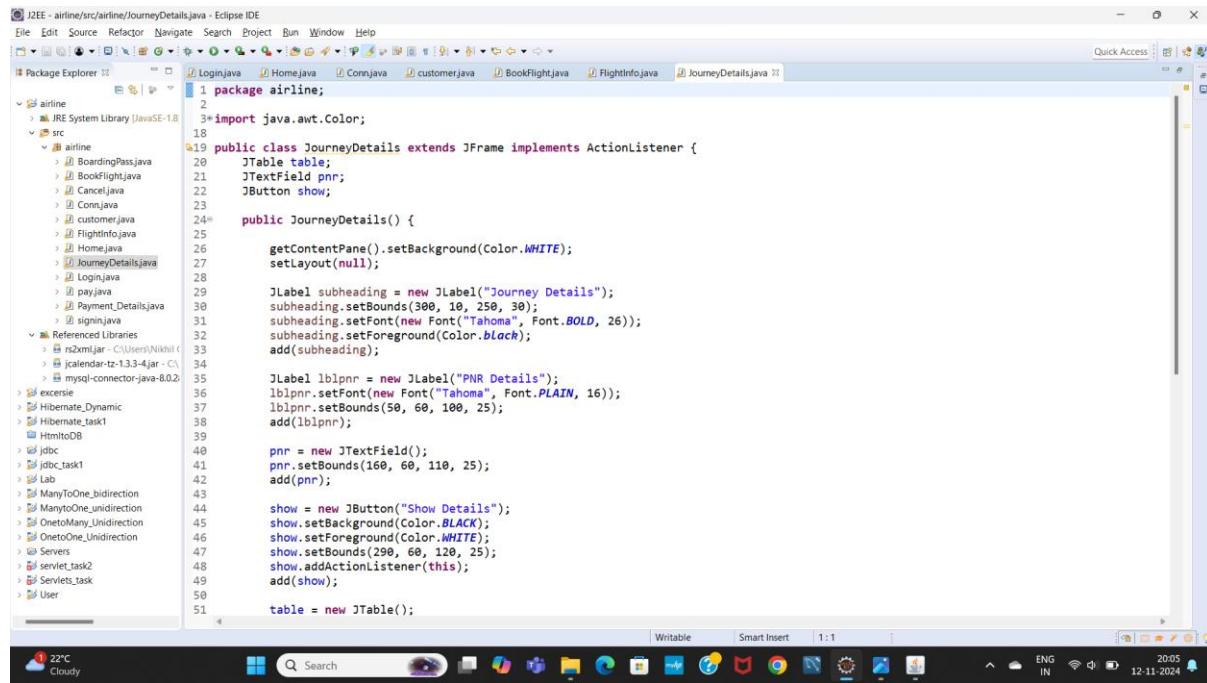
ENG IN

2001 12-11-2024

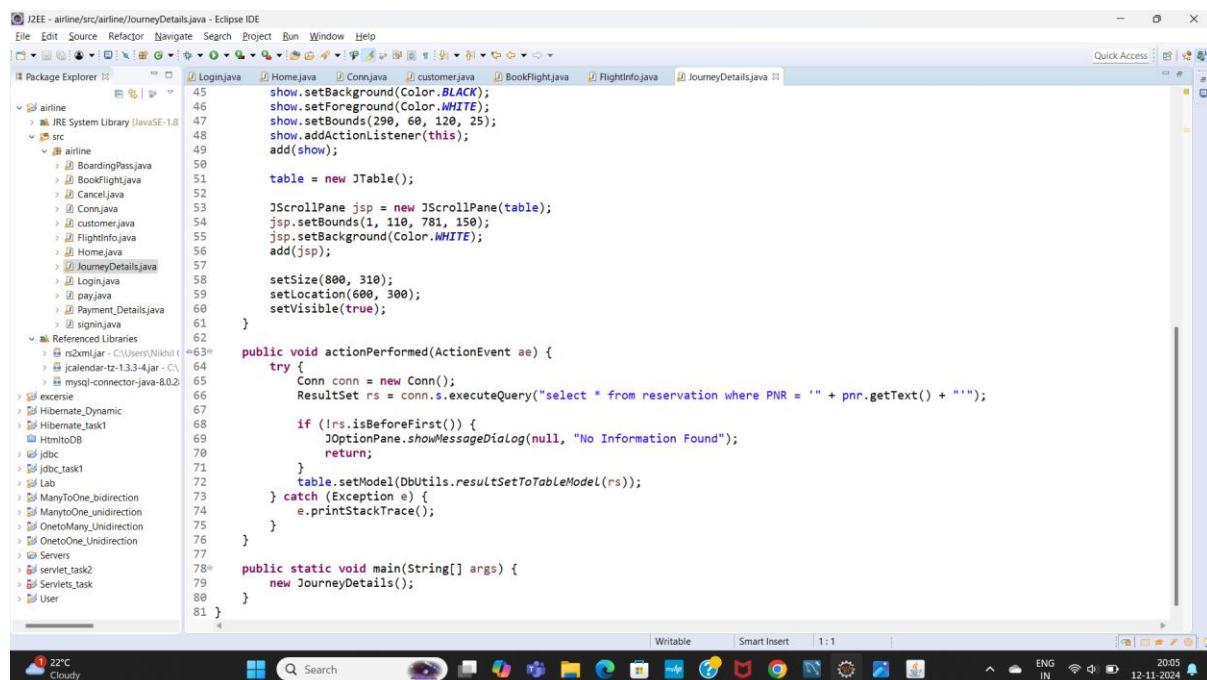


[Type here]

## 7. Journey Details

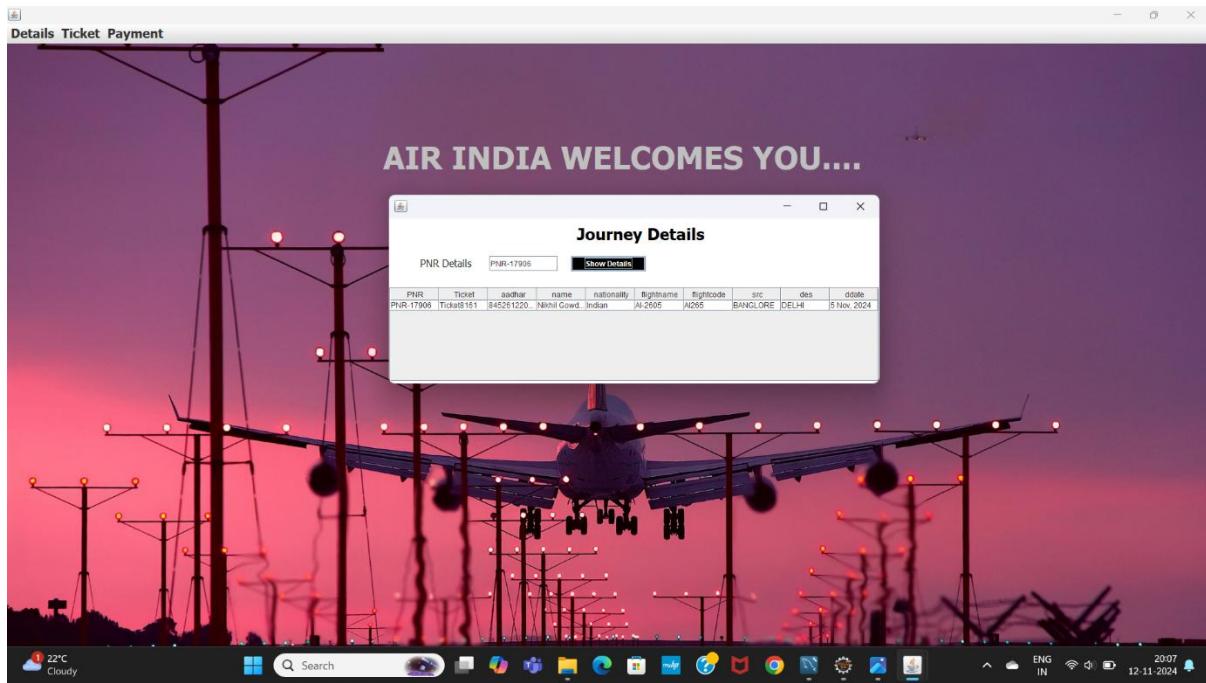


```
J2EE - airline/src/airline/JourneyDetails.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JLoginJava HomeJava ConnJava customerJava BookFlightJava FlightInfoJava JourneyDetails.java
1 package airline;
2
3*import java.awt.Color;
4
5 public class JourneyDetails extends JFrame implements ActionListener {
6
7     JTable table;
8     JTextField pnr;
9     JButton show;
10
11    public JourneyDetails() {
12
13        getContentPane().setBackground(Color.WHITE);
14        setLayout(null);
15
16        JLabel subheading = new JLabel("Journey Details");
17        subheading.setBounds(300, 10, 250, 30);
18        subheading.setFont(new Font("Tahoma", Font.BOLD, 26));
19        subheading.setForeground(Color.black);
20        add(subheading);
21
22        JLabel lblpnr = new JLabel("PNR Details");
23        lblpnr.setFont(new Font("Tahoma", Font.PLAIN, 16));
24        lblpnr.setBounds(50, 60, 100, 25);
25        add(lblpnr);
26
27        JTextField pnr = new JTextField();
28        pnr.setBounds(160, 60, 110, 25);
29        add(pnr);
30
31        JButton show = new JButton("Show Details");
32        show.setBackground(Color.BLACK);
33        show.setForeground(Color.WHITE);
34        show.setBounds(290, 60, 120, 25);
35        show.addActionListener(this);
36        add(show);
37
38        table = new JTable();
39
40        JScrollPane jsp = new JScrollPane(table);
41        jsp.setBounds(1, 110, 781, 150);
42        jsp.setBackground(Color.WHITE);
43        add(jsp);
44
45        setSize(800, 310);
46        setLocation(600, 300);
47        setVisible(true);
48
49    }
50
51    public void actionPerformed(ActionEvent ae) {
52        try {
53            Conn conn = new Conn();
54            ResultSet rs = conn.s.executeQuery("select * from reservation where PNR = '" + pnr.getText() + "'");
55
56            if (!rs.isBeforeFirst()) {
57                JOptionPane.showMessageDialog(null, "No Information Found");
58                return;
59            }
60            table.setModel(DbUtils.resultSetToTableModel(rs));
61        } catch (Exception e) {
62            e.printStackTrace();
63        }
64    }
65
66    public static void main(String[] args) {
67        new JourneyDetails();
68    }
69
70}
```

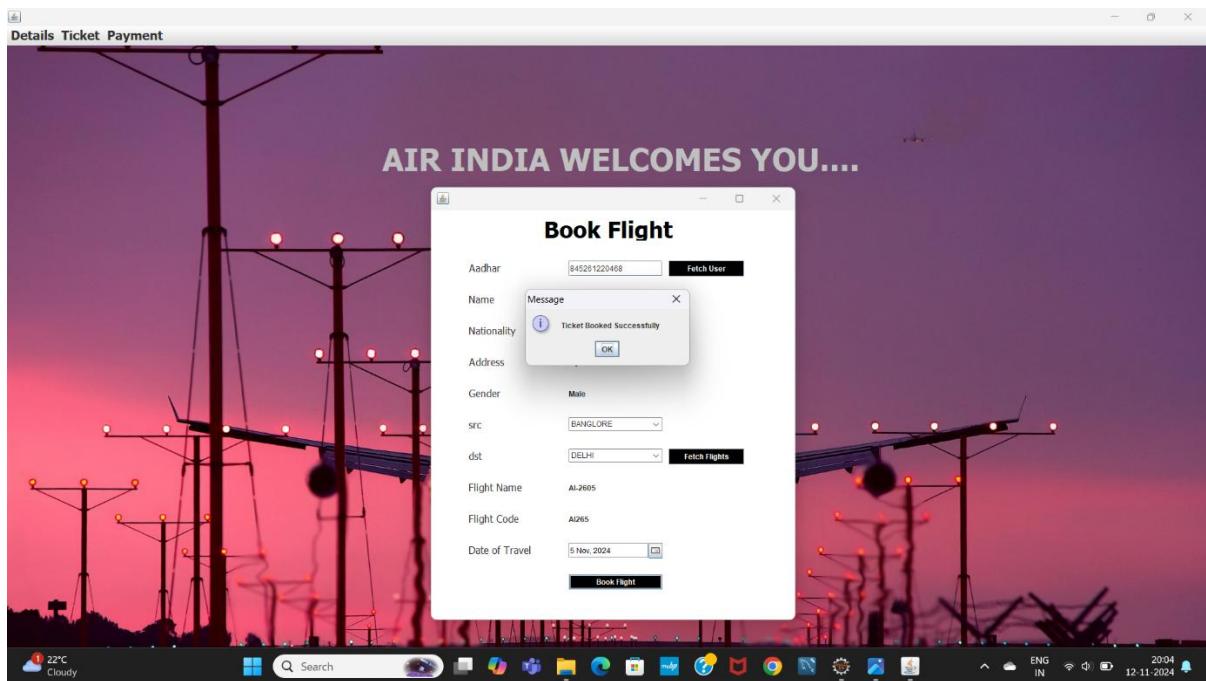


```
J2EE - airline/src/airline/JourneyDetails.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JLoginJava HomeJava ConnJava customerJava BookFlightJava FlightInfoJava JourneyDetails.java
1 package airline;
2
3*import java.awt.Color;
4
5 public class JourneyDetails extends JFrame implements ActionListener {
6
7     JTable table;
8     JTextField pnr;
9     JButton show;
10
11    public JourneyDetails() {
12
13        getContentPane().setBackground(Color.BLACK);
14        show.setForeground(Color.WHITE);
15        show.setBounds(290, 60, 120, 25);
16        show.addActionListener(this);
17        add(show);
18
19        table = new JTable();
20
21        JScrollPane jsp = new JScrollPane(table);
22        jsp.setBounds(1, 110, 781, 150);
23        jsp.setBackground(Color.WHITE);
24        add(jsp);
25
26        setSize(800, 310);
27        setLocation(600, 300);
28        setVisible(true);
29
30    }
31
32    public void actionPerformed(ActionEvent ae) {
33        try {
34            Conn conn = new Conn();
35            ResultSet rs = conn.s.executeQuery("select * from reservation where PNR = '" + pnr.getText() + "'");
36
37            if (!rs.isBeforeFirst()) {
38                JOptionPane.showMessageDialog(null, "No Information Found");
39                return;
40            }
41            table.setModel(DbUtils.resultSetToTableModel(rs));
42        } catch (Exception e) {
43            e.printStackTrace();
44        }
45    }
46
47    public static void main(String[] args) {
48        new JourneyDetails();
49    }
50
51}
```

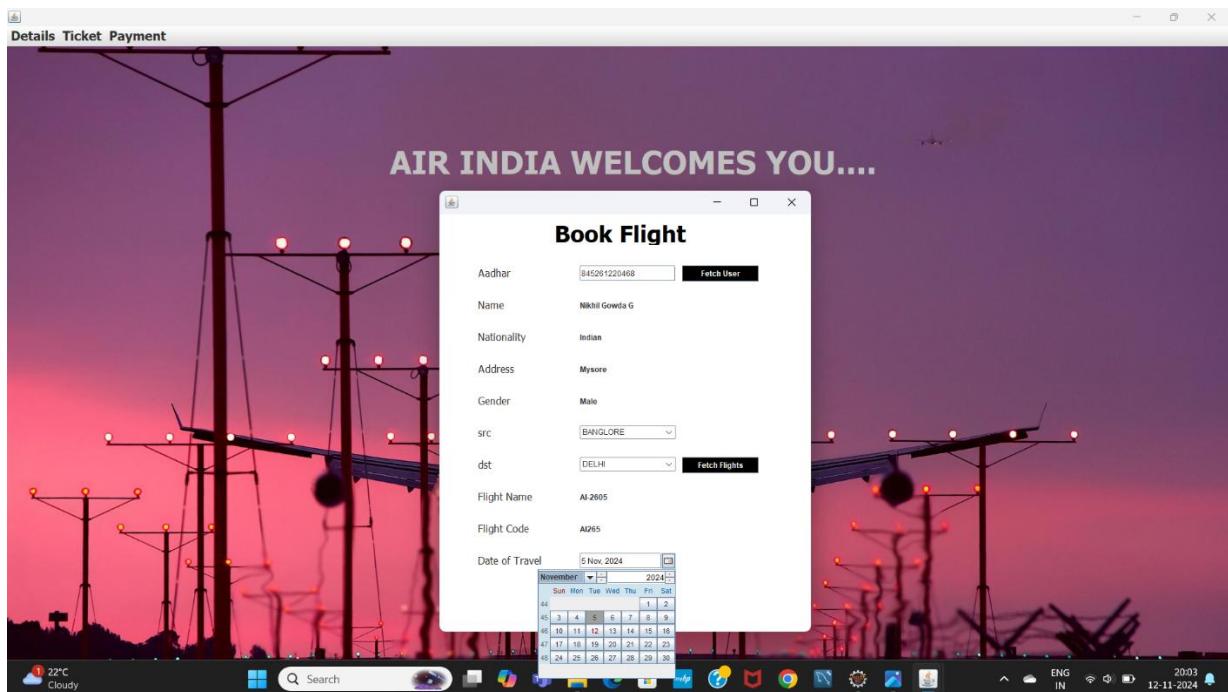
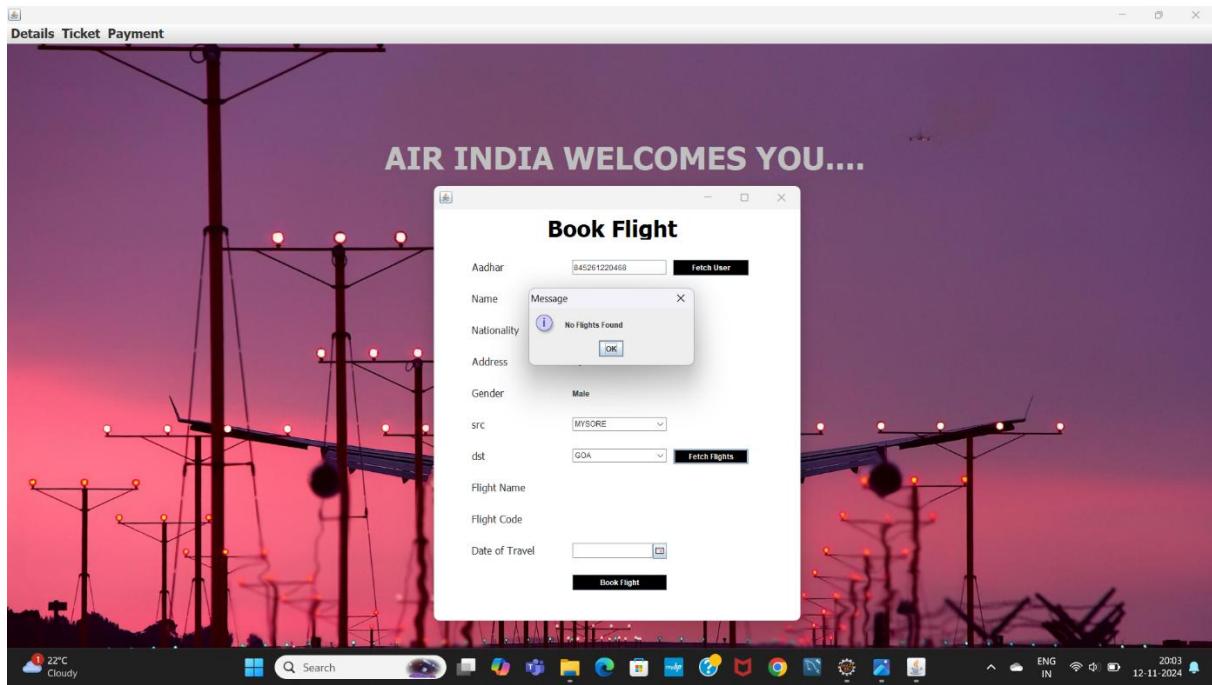
[Type here]



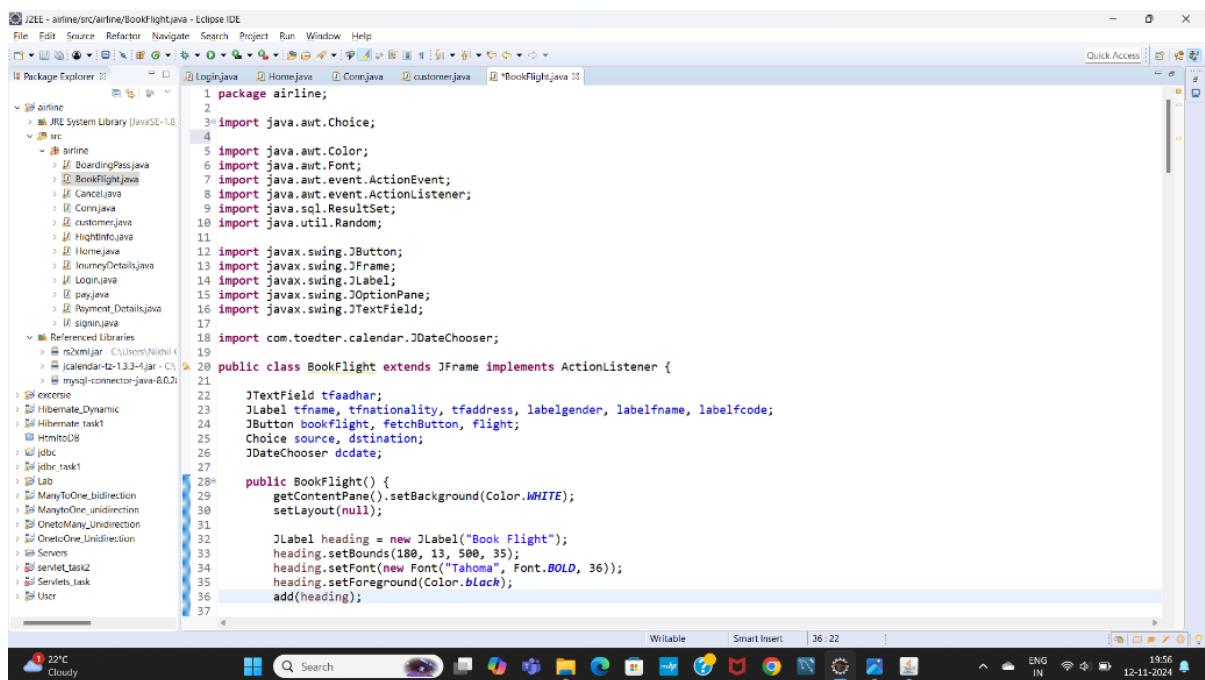
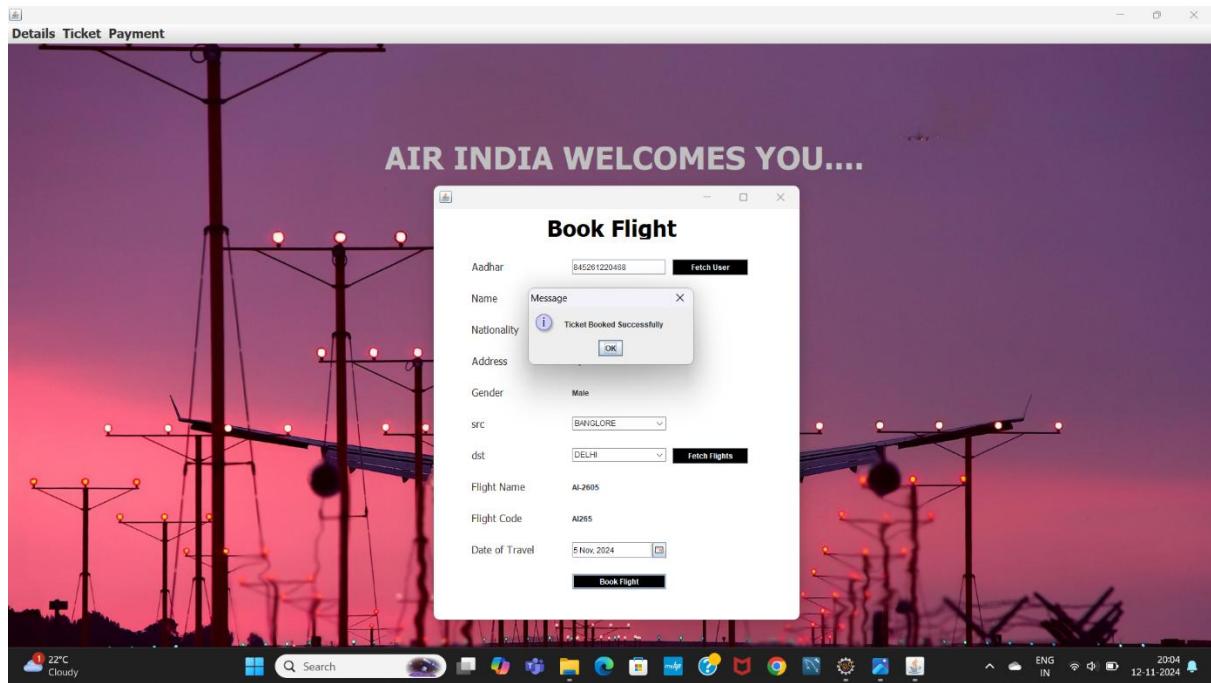
## 8. Book Flight



[Type here]



[Type here]



[Type here]

J2EE - airline/src/airline/BookFlight.java - Eclipse IDE

```
38     JLabel lblaadhar = new JLabel("Aadhar");
39     lblaadhar.setBounds(60, 80, 150, 25);
40     lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
41     add(lblaadhar);
42
43     JTextField tfaadhar = new JTextField();
44     tfaadhar.setBounds(220, 80, 150, 25);
45     add(tfaadhar);
46
47     JButton fetchButton = new JButton("Fetch User");
48     fetchButton.setBackground(Color.BLACK);
49     fetchButton.setForeground(Color.WHITE);
50     fetchButton.setBounds(380, 80, 120, 25);
51     fetchButton.addActionListener(this);
52     add(fetchButton);
53
54     JLabel lblname = new JLabel("Name");
55     lblname.setBounds(60, 130, 150, 25);
56     lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
57     add(lblname);
58
59     JLabel tfname = new JLabel();
60     tfname.setBounds(220, 130, 150, 25);
61     add(tfname);
62
63     JLabel lblnationality = new JLabel("Nationality");
64     lblnationality.setBounds(60, 180, 150, 25);
65     lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
66     add(lblnationality);
67
68     JLabel tfnationality = new JLabel();
69     tfnationality.setBounds(220, 180, 150, 25);
70     add(tfnationality);
71
72     JLabel lbladdress = new JLabel("Address");
73     lbladdress.setBounds(60, 230, 150, 25);
74     lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
75
76     JLabel tfaddress = new JLabel();
77     tfaddress.setBounds(220, 230, 150, 25);
78     add(tfaddress);
79
80     JLabel lblgender = new JLabel("Gender");
81     lblgender.setBounds(60, 280, 150, 25);
82     lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
83     add(lblgender);
84
85     JLabel labelgender = new JLabel("Gender");
86     labelgender.setBounds(220, 280, 150, 25);
87     add(labelgender);
88
89     JLabel lblsource = new JLabel("src");
90     lblsource.setBounds(60, 330, 150, 25);
91     lblsource.setFont(new Font("Tahoma", Font.PLAIN, 16));
92     add(lblsource);
93
94     Choice source = new Choice(); // for selection purpose
95     source.setBounds(220, 330, 150, 25);
96     // source.add("Mysore");
97     // source.add("Bnglr"); // selection
98     add(source);
99
100    JLabel lbdst = new JLabel("dst");
101    lbdst.setBounds(60, 380, 150, 25);
102    lbdst.setFont(new Font("Tahoma", Font.PLAIN, 16));
103    add(lbdst);
104
105    Choice dstination = new Choice();
106    dstination.setBounds(220, 380, 150, 25);
107
```

J2EE - airline/src/airline/BookFlight.java - Eclipse IDE

```
71
72     JLabel lbladdress = new JLabel("Address");
73     lbladdress.setBounds(60, 230, 150, 25);
74     lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
75     add(lbladdress);
76
77     JTextField tfaddress = new JTextField();
78     tfaddress.setBounds(220, 230, 150, 25);
79     add(tfaddress);
80
81     JLabel lblgender = new JLabel("Gender");
82     lblgender.setBounds(60, 280, 150, 25);
83     lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
84     add(lblgender);
85
86     JLabel labelgender = new JLabel("Gender");
87     labelgender.setBounds(220, 280, 150, 25);
88     add(labelgender);
89
90     JLabel lblsource = new JLabel("src");
91     lblsource.setBounds(60, 330, 150, 25);
92     lblsource.setFont(new Font("Tahoma", Font.PLAIN, 16));
93     add(lblsource);
94
95     Choice source = new Choice(); // for selection purpose
96     source.setBounds(220, 330, 150, 25);
97     // source.add("Mysore");
98     // source.add("Bnglr"); // selection
99     add(source);
100
101    JLabel lbdst = new JLabel("dst");
102    lbdst.setBounds(60, 380, 150, 25);
103    lbdst.setFont(new Font("Tahoma", Font.PLAIN, 16));
104    add(lbdst);
105
106    Choice dstination = new Choice();
107    dstination.setBounds(220, 380, 150, 25);
108
```

[Type here]

The screenshot shows the Eclipse IDE interface with the BookFlight.java file open in the editor. The code implements a JButton 'flight' which, when clicked, executes a database query to fetch flight details based on source and destination. The IDE's status bar at the bottom indicates it's 19:57 on 12.11.2024.

```
105     destination = new Choice();
106     destination.setBounds(220, 380, 150, 25);
107     add(destination);
108
109     try {
110         Conn c = new Conn();
111         String query = "select * from f_details";
112         ResultSet rs = c.s.executeQuery(query);
113
114         while (rs.next()) {
115             source.add(rs.getString("src"));
116             destination.add(rs.getString("dst"));
117         }
118     } catch (Exception e) {
119         e.printStackTrace();
120     }
121
122     flight = new JButton("Fetch Flights");
123     flight.setBackground(Color.BLACK);
124     flight.setForeground(Color.WHITE);
125     flight.setBounds(380, 380, 120, 25);
126     flight.addActionListener(this);
127     add(flight);
128
129     JLabel lblfname = new JLabel("Flight Name");
130     lblfname.setBounds(60, 430, 150, 25);
131     lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
132     add(lblfname);
133
134     JLabel lblname = new JLabel();
135     lblname.setBounds(220, 430, 150, 25);
136     add(lblname);
137
138     JLabel lblfcode = new JLabel("Flight Code");
139     lblfcode.setBounds(60, 480, 150, 25);
140
141 }
```

This screenshot shows the same Eclipse IDE session, but the BookFlight.java code has been scrolled down significantly, with the scroll bar visible on the right side of the editor window. The status bar at the bottom remains the same.

```
139     JLabel lblfcode = new JLabel("Flight Code");
140     lblfcode.setBounds(60, 480, 150, 25);
141     lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));
142     add(lblfcode);
143
144     JLabel labelcode = new JLabel();
145     labelcode.setBounds(220, 480, 150, 25);
146     add(labelcode);
147
148     JLabel lbdate = new JLabel("Date of Travel");
149     lbdate.setBounds(60, 530, 150, 25);
150     lbdate.setFont(new Font("Tahoma", Font.PLAIN, 16));
151     add(lbdate);
152
153     JDateChooser ddate = new JDateChooser();
154     ddate.setBounds(220, 530, 150, 25);
155     add(ddate);
156
157     bookflight = new JButton("Book Flight");
158     bookflight.setBackground(Color.BLACK);
159     bookflight.setForeground(Color.WHITE);
160     bookflight.setBounds(220, 580, 150, 25);
161     bookflight.addActionListener(this);
162     add(bookflight);
163
164     setSize(600, 700);
165     setLocation(650, 180);
166     setVisible(true);
167
168 }
169
170 public void actionPerformed(ActionEvent ae) {
171     if (ae.getSource() == fetchButton) {
172         String aadhar = tfaadhar.getText();
173
174         try {
175             Conn conn = new Conn();
176
177             String query = "select * from f_details where src = ? and dst = ?";
178             PreparedStatement ps = conn.s.prepareStatement(query);
179
180             ps.setString(1, source.getSelectedItem().toString());
181             ps.setString(2, destination.getSelectedItem().toString());
182
183             ResultSet rs = ps.executeQuery();
184
185             while (rs.next()) {
186                 String fname = rs.getString("fname");
187                 String lname = rs.getString("lname");
188                 String fcode = rs.getString("fcode");
189                 String date = rs.getString("date");
190
191                 JLabel fnameLabel = new JLabel(fname);
192                 fnameLabel.setBounds(60, 100, 150, 25);
193                 add(fnameLabel);
194
195                 JLabel lnameLabel = new JLabel(lname);
196                 lnameLabel.setBounds(220, 100, 150, 25);
197                 add(lnameLabel);
198
199                 JLabel fcodeLabel = new JLabel(fcode);
200                 fcodeLabel.setBounds(60, 150, 150, 25);
201                 add(fcodeLabel);
202
203                 JLabel dateLabel = new JLabel(date);
204                 dateLabel.setBounds(220, 150, 150, 25);
205                 add(dateLabel);
206
207             }
208
209         } catch (Exception e) {
210             e.printStackTrace();
211         }
212     }
213 }
```

[Type here]

J2EE - airline/src/airline/BookFlight.java Eclipse IDE

```
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == fetchButton) {
        String aadhar = tfaadhar.getText();

        try {
            Conn conn = new Conn();

            String query = "select * from passenger where aadhar = '" + aadhar + "'";

            ResultSet rs = conn.s.executeQuery(query);

            if (rs.next()) {
                tfname.setText(rs.getString("name"));
                tfnationality.setText(rs.getString("nationality"));
                tfaddress.setText(rs.getString("address"));
                labelgender.setText(rs.getString("gender"));
            } else {
                JOptionPane.showMessageDialog(null, "Please enter correct aadhar");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else if (ae.getSource() == flight) {
        String src = source.getSelectedItem();
        String dst = destination.getSelectedItem();
        try {
            Conn conn = new Conn();

            String query = "select * from f_details where src = '" + src + "' and dst = '" + dst + "'";

            ResultSet rs = conn.s.executeQuery(query);

            if (rs.next()) {
                labelfname.setText(rs.getString("f_name"));
                labelfcode.setText(rs.getString("f_code"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Writable | Smart Insert | 144:1 | ENG IN 2000 12-11-2024

J2EE - airline/src/airline/BookFlight.java Eclipse IDE

```
        } else {
            JOptionPane.showMessageDialog(null, "No Flights Found");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    Random random = new Random();

    String aadhar = tfaadhar.getText();
    String name = tfname.getText();
    String nationality = fnationality.getText();
    String flightname = labelfname.getText();
    String flightcode = labelfcode.getText();
    String src = source.getSelectedItem();
    String des = destination.getSelectedItem();
    String ddate = ((JTextField) ddate.getDateEditor().getUiComponent()).getText();

    try {
        Conn conn = new Conn();

        String query = "insert into reservation values('PNR-' + random.nextInt(1000000) + "", 'Ticket' +
            + random.nextInt(10000) + "", "' + aadhar + '", "' + name + '", "' + nationality + '", "' +
            + flightname + '", "' + flightcode + '", "' + src + '", "' + des + '", "' + ddate + '")";

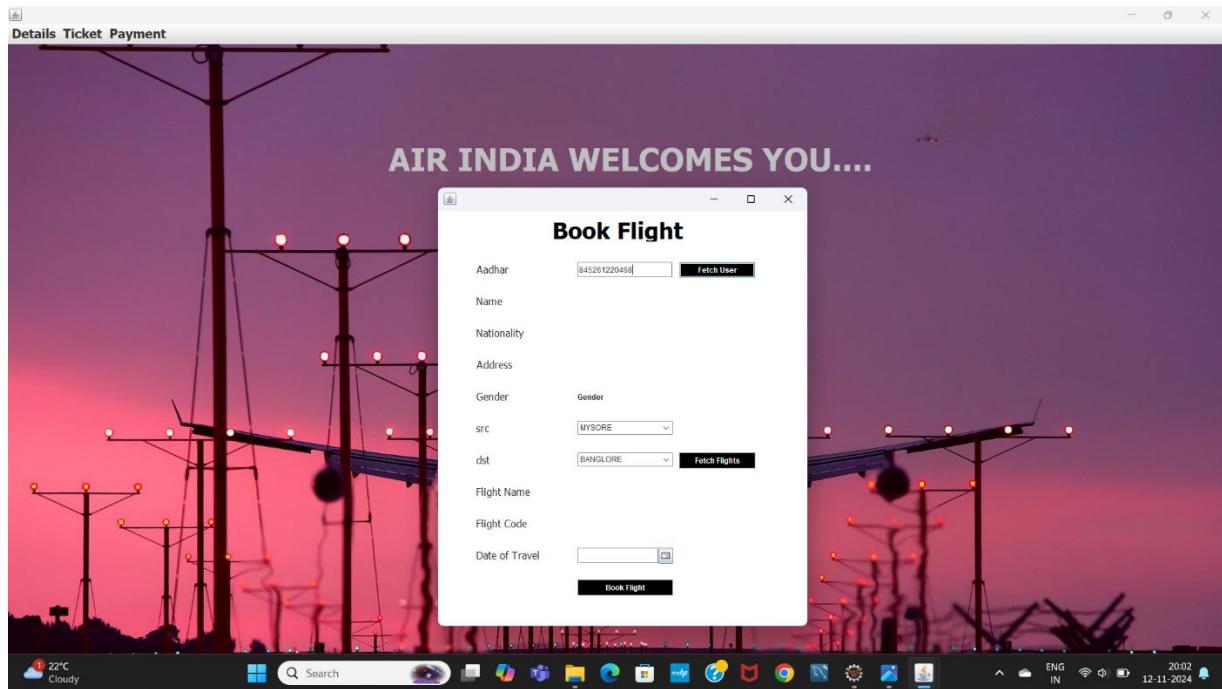
        conn.s.executeUpdate(query);

        JOptionPane.showMessageDialog(null, "Ticket Booked Successfully");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

public static void main(String[] args) {
    new BookFlight();
}
```

Writable | Smart Insert | 143:1 | ENG IN 2001 12-11-2024

[Type here]



## 8. Boarding Pass

```
1 package airline;
2
3 import java.awt.Color;
4
5 public class BoardingPass extends JFrame implements ActionListener {
6
7     JTextField tfpnr;
8     JLabel tfname, tfnationality, lblsrc, lbldest, labelfname, labelfcode, labeldate;
9     JButton fetchButton;
10
11    public BoardingPass() {
12        getContentPane().setBackground(Color.WHITE);
13        setLayout(null);
14
15        JLabel heading = new JLabel("AIR INDIA");
16        heading.setBounds(200, 10, 450, 35);
17        heading.setFont(new Font("Tahoma", Font.BOLD, 32));
18        add(heading);
19
20        JLabel subheading = new JLabel("Boarding Pass");
21        subheading.setBounds(210, 50, 300, 30);
22        subheading.setFont(new Font("Tahoma", Font.PLAIN, 24));
23        subheading.setForeground(Color.BLUE);
24        add(subheading);
25
26        JLabel lblaadhar = new JLabel("PNR DETAILS");
27        lblaadhar.setBounds(60, 100, 150, 25);
28        lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
29        add(lblaadhar);
30
31        tfpnr = new JTextField();
32        tfpnr.setBounds(220, 100, 150, 25);
33        add(tfpnr);
34
35        fetchButton = new JButton("Enter");
36        fetchButton.setBackground(Color.BLACK);
37        fetchButton.setForeground(Color.WHITE);
38        add(fetchButton);
39
40    }
41
42}
```

[Type here]

J2EE - airline/src/airline/BoardingPass.java - Eclipse IDE

```
45     JButton fetchButton = new JButton("Enter");
46     fetchButton.setBackground(Color.BLACK);
47     fetchButton.setForeground(Color.WHITE);
48     fetchButton.setBounds(380, 100, 120, 25);
49     fetchButton.addActionListener(this);
50     add(fetchButton);

51     JLabel lblname = new JLabel("NAME");
52     lblname.setBounds(60, 140, 150, 25);
53     lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
54     add(lblname);

55     JTextField tfname = new JTextField();
56     tfname.setBounds(220, 140, 150, 25);
57     add(tfname);

58     JLabel lbnationality = new JLabel("NATIONALITY");
59     lbnationality.setBounds(60, 180, 150, 25);
60     lbnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
61     add(lbnationality);

62     JTextField tfnationality = new JTextField();
63     tfnationality.setBounds(220, 180, 150, 25);
64     add(tfnationality);

65     JLabel lbladdress = new JLabel("SRC");
66     lbladdress.setBounds(60, 220, 150, 25);
67     lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
68     add(lbladdress);

69     JLabel lblsrc = new JLabel();
70     lblsrc.setBounds(220, 220, 150, 25);
71     add(lblsrc);

72     JLabel lblgender = new JLabel("DEST");
73     lblgender.setBounds(380, 220, 150, 25);
74     add(lblgender);

75     JLabel lblname1 = new JLabel("Flight Name");
76     lblname1.setBounds(60, 260, 150, 25);
77     lblname1.setFont(new Font("Tahoma", Font.PLAIN, 16));
78     add(lblname1);

79     JLabel lbldest = new JLabel("DEST");
80     lbldest.setBounds(380, 260, 150, 25);
81     lbldest.setFont(new Font("Tahoma", Font.PLAIN, 16));
82     add(lbldest);

83     JLabel lblfcode = new JLabel("Flight Code");
84     lblfcode.setBounds(380, 300, 150, 25);
85     lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));
86     add(lblfcode);

87     JLabel lblfdate = new JLabel("Date");
88     lblfdate.setBounds(60, 340, 150, 25);
89     lblfdate.setFont(new Font("Tahoma", Font.PLAIN, 16));
90     add(lblfdate);

91     JLabel lblname2 = new JLabel("Flight Name");
92     lblname2.setBounds(220, 340, 150, 25);
93     lblname2.setFont(new Font("Tahoma", Font.PLAIN, 16));
94     add(lblname2);

95     JLabel lblcode = new JLabel("Flight Code");
96     lblcode.setBounds(220, 380, 150, 25);
97     lblcode.setFont(new Font("Tahoma", Font.PLAIN, 16));
98     add(lblcode);

99     JLabel lbldate = new JLabel("Date");
100    lbldate.setBounds(60, 420, 150, 25);
101    lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));
102    add(lbldate);

103    JButton btnSubmit = new JButton("Submit");
104    btnSubmit.setBounds(220, 460, 150, 25);
105    btnSubmit.addActionListener(this);
106    add(btnSubmit);

107    JButton btnCancel = new JButton("Cancel");
108    btnCancel.setBounds(380, 460, 150, 25);
109    btnCancel.addActionListener(this);
110    add(btnCancel);

111    JButton btnClear = new JButton("Clear");
112    btnClear.setBounds(220, 500, 150, 25);
113    btnClear.addActionListener(this);
114    add(btnClear);

115    // Timer
116    Timer timer = new Timer();
117    timer.schedule(new TimerTask() {
118        public void run() {
119            // Update UI or perform task
120        }
121    }, 0, 1000);
122}
```

J2EE - airline/src/airline/BoardingPass.java - Eclipse IDE

```
79     JLabel lblgender = new JLabel("DEST");
80     lblgender.setBounds(380, 220, 150, 25);
81     lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
82     add(lblgender);

83     JLabel lbldest = new JLabel();
84     lbldest.setBounds(540, 220, 150, 25);
85     add(lbldest);

86     JLabel lblfname = new JLabel("Flight Name");
87     lblfname.setBounds(60, 260, 150, 25);
88     lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
89     add(lblfname);

90     JLabel lblfcode = new JLabel("Flight Code");
91     lblfcode.setBounds(380, 260, 150, 25);
92     lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));
93     add(lblfcode);

94     JLabel lblfdate = new JLabel("Date");
95     lblfdate.setBounds(60, 300, 150, 25);
96     lblfdate.setFont(new Font("Tahoma", Font.PLAIN, 16));
97     add(lblfdate);

98     JLabel lblname = new JLabel();
99     lblname.setBounds(220, 260, 150, 25);
100    add(lblname);

101    JLabel lblcode = new JLabel();
102    lblcode.setBounds(540, 260, 150, 25);
103    add(lblcode);

104    JLabel lbldate = new JLabel();
105    lbldate.setBounds(220, 300, 150, 25);
106    lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));
107    add(lbldate);

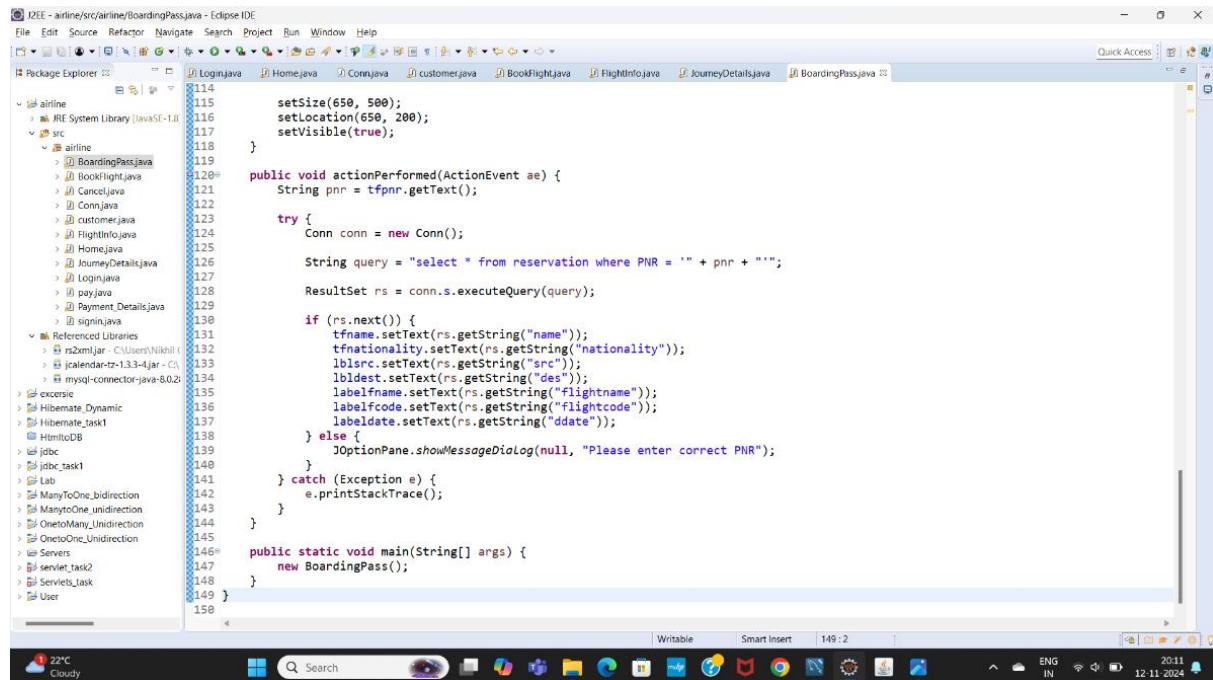
108    JButton btnSubmit = new JButton("Submit");
109    btnSubmit.setBounds(220, 340, 150, 25);
110    btnSubmit.addActionListener(this);
111    add(btnSubmit);

112    JButton btnCancel = new JButton("Cancel");
113    btnCancel.setBounds(380, 340, 150, 25);
114    btnCancel.addActionListener(this);
115    add(btnCancel);

116    JButton btnClear = new JButton("Clear");
117    btnClear.setBounds(220, 380, 150, 25);
118    btnClear.addActionListener(this);
119    add(btnClear);

120    // Timer
121    Timer timer = new Timer();
122    timer.schedule(new TimerTask() {
123        public void run() {
124            // Update UI or perform task
125        }
126    }, 0, 1000);
127}
```

[Type here]

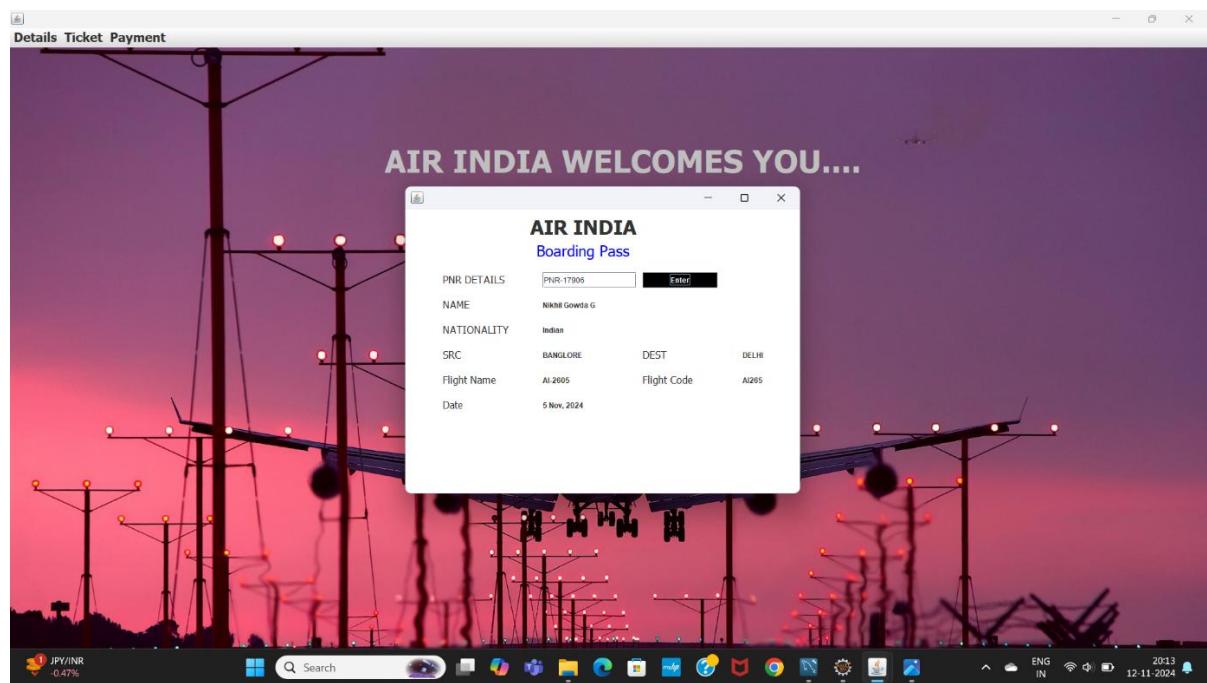


The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** J2EE - airline/src/airline/BoardingPass.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Standard Eclipse toolbar items.
- Package Explorer:** Shows the project structure with packages like airline, src, and Referenced Libraries containing various Java files.
- Code Editor:** Displays the Java code for `BoardingPass.java`. The code handles PNR input, database query execution, and result processing to set labels for flight details. It includes exception handling and a main method.

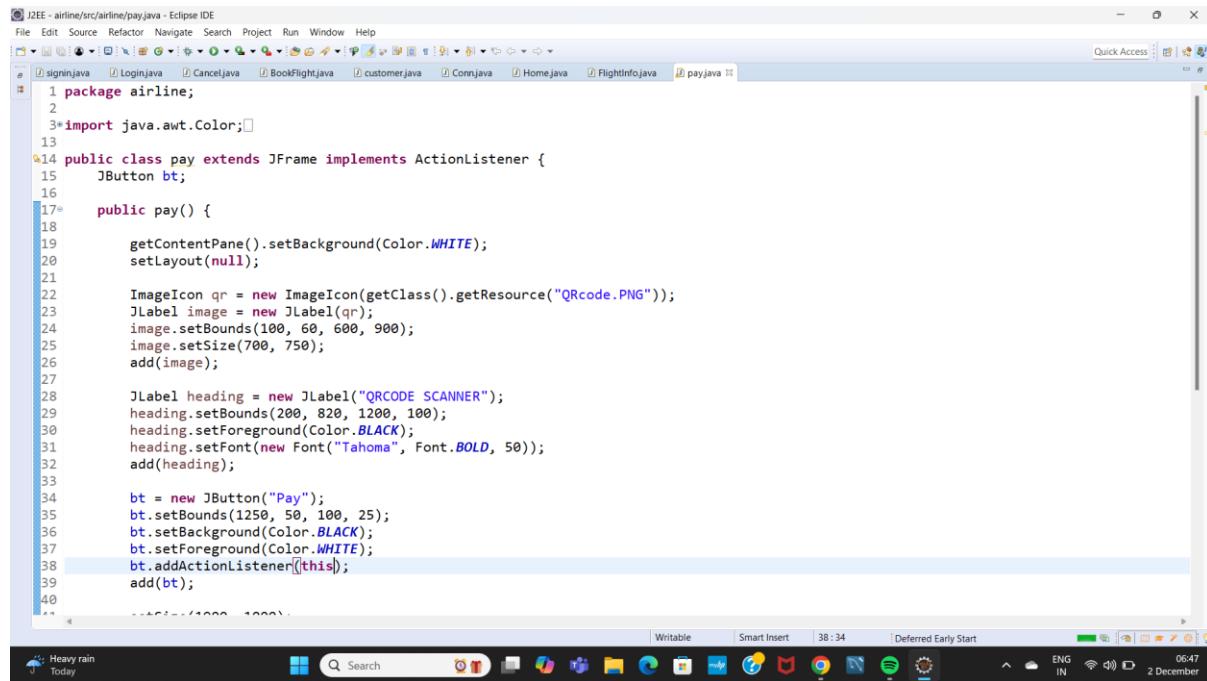
```
114     setSize(650, 500);
115     setLocation(650, 200);
116     setVisible(true);
117 }
118
119 public void actionPerformed(ActionEvent ae) {
120     String pnr = tfpnr.getText();
121
122     try {
123         Conn conn = new Conn();
124
125         String query = "select * from reservation where PNR = '" + pnr + "'";
126         ResultSet rs = conn.s.executeQuery(query);
127
128         if (rs.next()) {
129             tfname.setText(rs.getString("name"));
130             tfnationality.setText(rs.getString("nationality"));
131             lblsrc.setText(rs.getString("src"));
132             lbldest.setText(rs.getString("des"));
133             labelfname.setText(rs.getString("flightname"));
134             labelcode.setText(rs.getString("flightcode"));
135             labeldate.setText(rs.getString("ddate"));
136         } else {
137             JOptionPane.showMessageDialog(null, "Please enter correct PNR");
138         }
139     } catch (Exception e) {
140         e.printStackTrace();
141     }
142 }
143
144 public static void main(String[] args) {
145     new BoardingPass();
146 }
147
148 }
149 }
```

- Bottom Status Bar:** Shows Writable, Smart Insert, 149:2, ENG IN, 2011, 12-11-2024, and a battery icon.



[Type here]

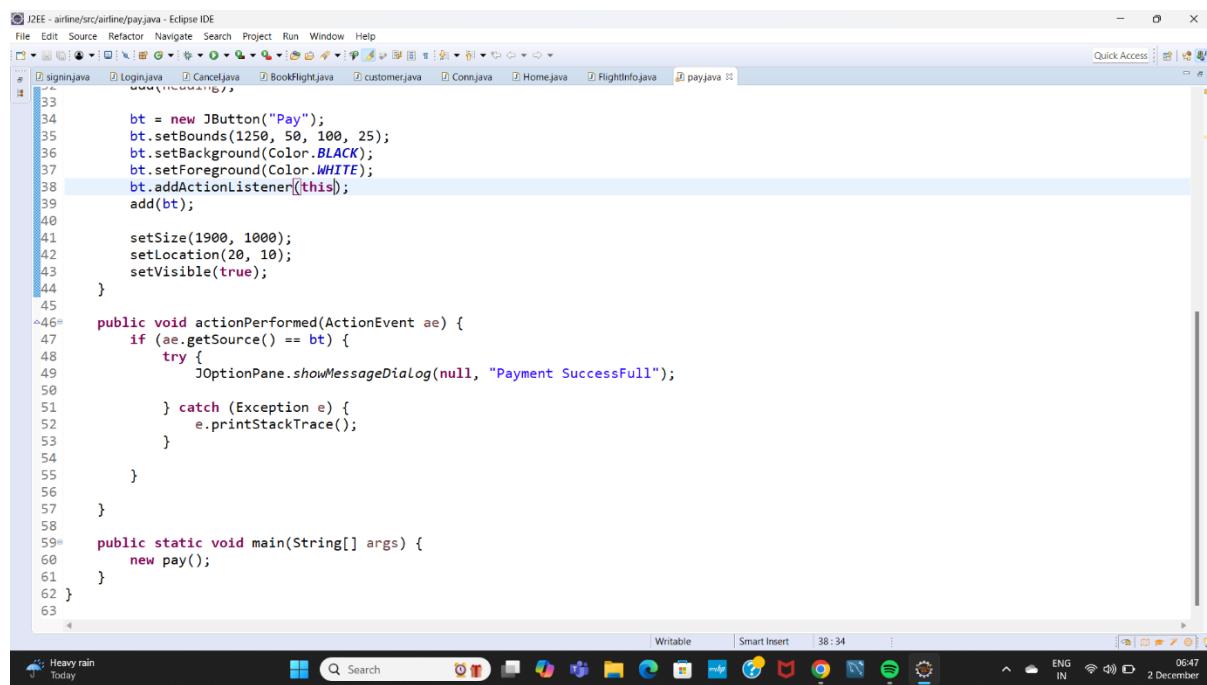
## 9. Payment



J2EE - airline/src/airline/pay.java - Eclipse IDE

```
1 package airline;
2
3 import java.awt.Color;
4
5 public class pay extends JFrame implements ActionListener {
6     JButton bt;
7
8     public pay() {
9         getContentPane().setBackground(Color.WHITE);
10        setLayout(null);
11
12        ImageIcon qr = new ImageIcon(getClass().getResource("QRcode.PNG"));
13        JLabel image = new JLabel(qr);
14        image.setBounds(100, 60, 600, 900);
15        image.setSize(700, 750);
16        add(image);
17
18        JLabel heading = new JLabel("QR CODE SCANNER");
19        heading.setBounds(200, 820, 1200, 100);
20        heading.setForeground(Color.BLACK);
21        heading.setFont(new Font("Tahoma", Font.BOLD, 50));
22        add(heading);
23
24        bt = new JButton("Pay");
25        bt.setBounds(1250, 50, 100, 25);
26        bt.setBackground(Color.BLACK);
27        bt.setForeground(Color.WHITE);
28        bt.addActionListener(this);
29        add(bt);
30
31        setSize(1900, 1000);
32        setLocation(20, 10);
33        setVisible(true);
34    }
35
36    public void actionPerformed(ActionEvent ae) {
37        if (ae.getSource() == bt) {
38            try {
39                JOptionPane.showMessageDialog(null, "Payment SuccessFull");
40            } catch (Exception e) {
41                e.printStackTrace();
42            }
43        }
44    }
45
46    public static void main(String[] args) {
47        new pay();
48    }
49 }
```

Writable Smart Insert 38 : 34 Deferred Early Start ENG IN 06:47 2 December



J2EE - airline/src/airline/pay.java - Eclipse IDE

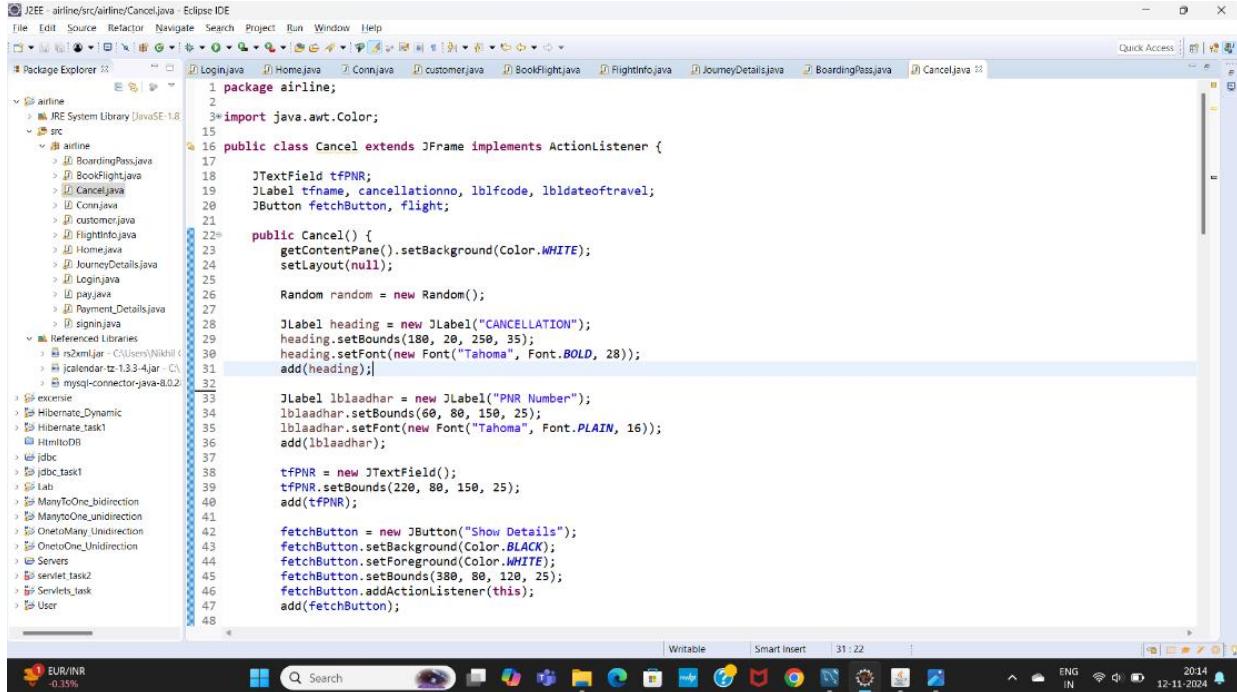
```
1 package airline;
2
3 import java.awt.Color;
4
5 public class pay extends JFrame implements ActionListener {
6     JButton bt;
7
8     public pay() {
9         getContentPane().setBackground(Color.WHITE);
10        setLayout(null);
11
12        ImageIcon qr = new ImageIcon(getClass().getResource("QRcode.PNG"));
13        JLabel image = new JLabel(qr);
14        image.setBounds(100, 60, 600, 900);
15        image.setSize(700, 750);
16        add(image);
17
18        JLabel heading = new JLabel("QR CODE SCANNER");
19        heading.setBounds(200, 820, 1200, 100);
20        heading.setForeground(Color.BLACK);
21        heading.setFont(new Font("Tahoma", Font.BOLD, 50));
22        add(heading);
23
24        bt = new JButton("Pay");
25        bt.setBounds(1250, 50, 100, 25);
26        bt.setBackground(Color.BLACK);
27        bt.setForeground(Color.WHITE);
28        bt.addActionListener(this);
29        add(bt);
30
31        setSize(1900, 1000);
32        setLocation(20, 10);
33        setVisible(true);
34    }
35
36    public void actionPerformed(ActionEvent ae) {
37        if (ae.getSource() == bt) {
38            try {
39                JOptionPane.showMessageDialog(null, "Payment SuccessFull");
40            } catch (Exception e) {
41                e.printStackTrace();
42            }
43        }
44    }
45
46    public static void main(String[] args) {
47        new pay();
48    }
49 }
```

Writable Smart Insert 38 : 34 ENG IN 06:47 2 December

[Type here]

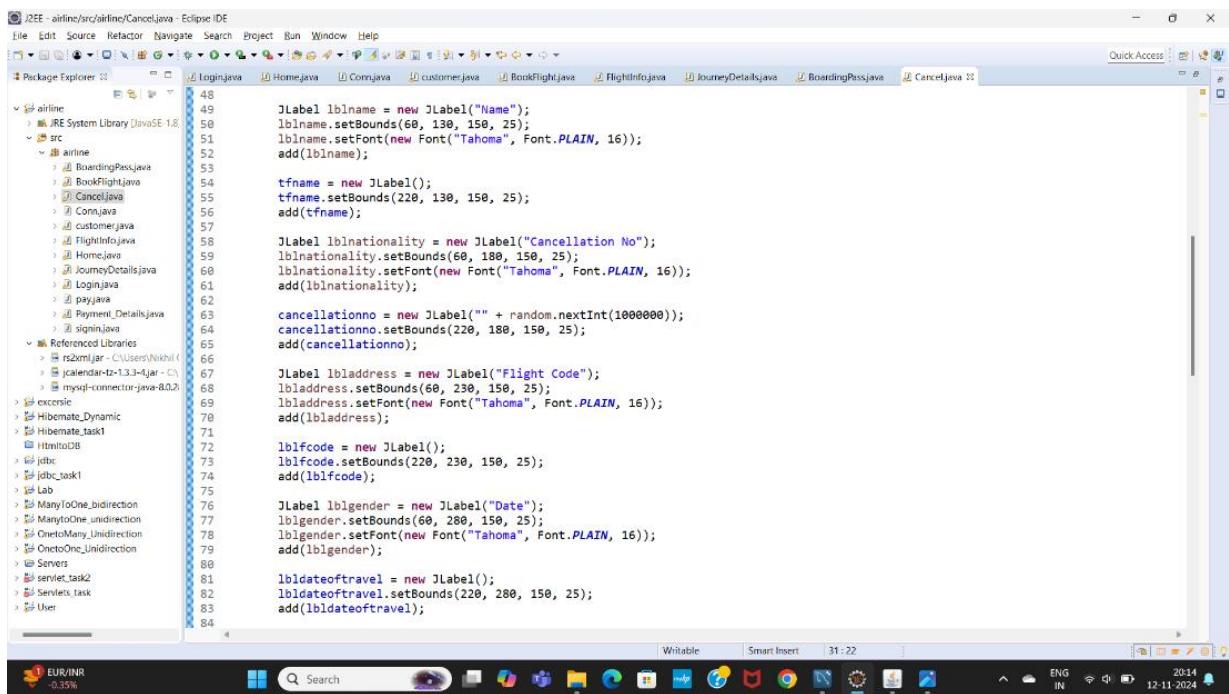
## Add your QR code img

### 10. Cancel



J2EE - airline/src/airline/Cancel.java - Eclipse IDE

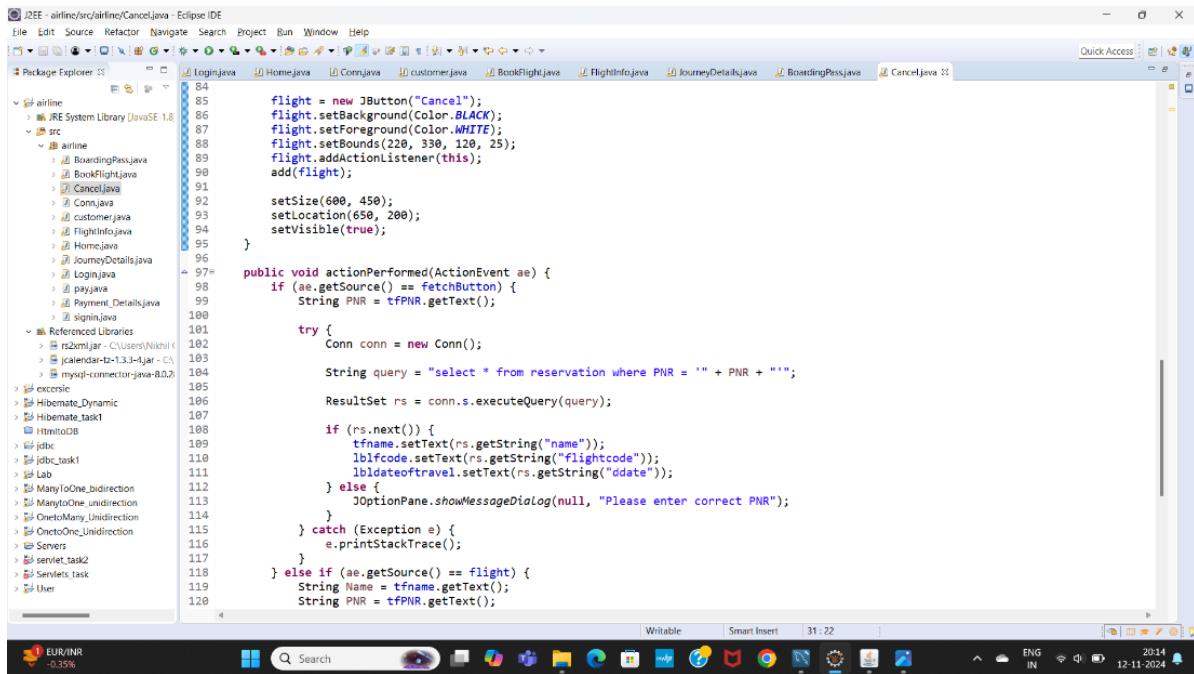
```
1 package airline;
2
3 import java.awt.Color;
4
5 public class Cancel extends JFrame implements ActionListener {
6
7     JTextField tfPNR;
8     JLabel tfname, cancellationno, lblfcode, lbldateoftravel;
9     JButton fetchButton, flight;
10
11    public Cancel() {
12        getContentPane().setBackground(Color.WHITE);
13        setLayout(null);
14
15        Random random = new Random();
16
17        JLabel heading = new JLabel("CANCELLATION");
18        heading.setBounds(180, 20, 250, 35);
19        heading.setFont(new Font("Tahoma", Font.BOLD, 28));
20
21        JLabel lblaadhar = new JLabel("PNR Number");
22        lblaadhar.setBounds(60, 80, 150, 25);
23        lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
24
25        JTextField tfPNR = new JTextField();
26        tfPNR.setBounds(220, 80, 150, 25);
27        add(tfPNR);
28
29        JButton fetchButton = new JButton("Show Details");
30        fetchButton.setBackground(Color.BLACK);
31        fetchButton.setForeground(Color.WHITE);
32        fetchButton.setBounds(380, 80, 120, 25);
33        fetchButton.addActionListener(this);
34        add(fetchButton);
35
36        JButton flight = new JButton("Flight Info");
37        flight.setBackground(Color.BLACK);
38        flight.setForeground(Color.WHITE);
39        flight.setBounds(380, 120, 120, 25);
40        add(flight);
41
42        JButton Conn = new JButton("Conn");
43        Conn.setBackground(Color.BLACK);
44        Conn.setForeground(Color.WHITE);
45        Conn.setBounds(380, 160, 120, 25);
46        add(Conn);
47
48    }
49
50    JLabel lblname = new JLabel("Name");
51    lblname.setBounds(60, 130, 150, 25);
52    lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
53
54    JLabel tfname = new JLabel();
55    tfname.setBounds(220, 130, 150, 25);
56    add(tfname);
57
58    JLabel lblnationality = new JLabel("Cancellation No");
59    lblnationality.setBounds(60, 180, 150, 25);
60    lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
61
62    JLabel cancellationno = new JLabel("+" + random.nextInt(1000000));
63    cancellationno.setBounds(220, 180, 150, 25);
64    add(cancellationno);
65
66    JLabel lbladdress = new JLabel("Flight Code");
67    lbladdress.setBounds(60, 230, 150, 25);
68    lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
69
70    JLabel lblfcode = new JLabel();
71    lblfcode.setBounds(220, 230, 150, 25);
72    add(lblfcode);
73
74    JLabel lblgender = new JLabel("Date");
75    lblgender.setBounds(60, 280, 150, 25);
76    lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
77
78    JLabel lbldateoftravel = new JLabel();
79    lbldateoftravel.setBounds(220, 280, 150, 25);
80    add(lbldateoftravel);
81
82}
```



J2EE - airline/src/airline/Login.java - Eclipse IDE

```
1 package airline;
2
3 import java.awt.Color;
4
5 public class Login extends JFrame implements ActionListener {
6
7     JTextField tfname, tfpassword;
8     JButton Conn, customer, flightinfo, JourneyDetails, BoardingPass, cancel;
9
10    public Login() {
11        getContentPane().setBackground(Color.WHITE);
12        setLayout(null);
13
14        Random random = new Random();
15
16        JLabel heading = new JLabel("AIRLINES");
17        heading.setBounds(180, 20, 250, 35);
18        heading.setFont(new Font("Tahoma", Font.BOLD, 28));
19
20        JLabel tfname = new JLabel("Name");
21        tfname.setBounds(60, 80, 150, 25);
22        tfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
23
24        JTextField tfname1 = new JTextField();
25        tfname1.setBounds(220, 80, 150, 25);
26        add(tfname1);
27
28        JLabel tfpassword = new JLabel("Password");
29        tfpassword.setBounds(60, 120, 150, 25);
30        tfpassword.setFont(new Font("Tahoma", Font.PLAIN, 16));
31
32        JTextField tfpassword1 = new JTextField();
33        tfpassword1.setBounds(220, 120, 150, 25);
34        add(tfpassword1);
35
36        JButton Conn = new JButton("Conn");
37        Conn.setBackground(Color.BLACK);
38        Conn.setForeground(Color.WHITE);
39        Conn.setBounds(380, 80, 120, 25);
40        Conn.addActionListener(this);
41        add(Conn);
42
43        JButton customer = new JButton("Customer");
44        customer.setBackground(Color.BLACK);
45        customer.setForeground(Color.WHITE);
46        customer.setBounds(380, 120, 120, 25);
47        customer.addActionListener(this);
48        add(customer);
49
50        JButton flightinfo = new JButton("Flight Info");
51        flightinfo.setBackground(Color.BLACK);
52        flightinfo.setForeground(Color.WHITE);
53        flightinfo.setBounds(380, 160, 120, 25);
54        flightinfo.addActionListener(this);
55        add(flightinfo);
56
57        JButton JourneyDetails = new JButton("Journey Details");
58        JourneyDetails.setBackground(Color.BLACK);
59        JourneyDetails.setForeground(Color.WHITE);
60        JourneyDetails.setBounds(380, 200, 120, 25);
61        JourneyDetails.addActionListener(this);
62        add(JourneyDetails);
63
64        JButton BoardingPass = new JButton("Boarding Pass");
65        BoardingPass.setBackground(Color.BLACK);
66        BoardingPass.setForeground(Color.WHITE);
67        BoardingPass.setBounds(380, 240, 120, 25);
68        BoardingPass.addActionListener(this);
69        add(BoardingPass);
70
71        JButton cancel = new JButton("Cancel");
72        cancel.setBackground(Color.BLACK);
73        cancel.setForeground(Color.WHITE);
74        cancel.setBounds(380, 280, 120, 25);
75        cancel.addActionListener(this);
76        add(cancel);
77
78    }
79
80    void Conn() {
81        String name = tfname1.getText();
82        String password = tfpassword1.getText();
83
84        if (name.equals("Nikhil") & password.equals("12345")) {
85            JOptionPane.showMessageDialog(null, "Welcome " + name);
86        } else {
87            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
88        }
89    }
90
91    void customer() {
92        String name = tfname1.getText();
93        String password = tfpassword1.getText();
94
95        if (name.equals("Nikhil") & password.equals("12345")) {
96            JOptionPane.showMessageDialog(null, "Customer Details");
97        } else {
98            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
99        }
100    }
101
102    void flightinfo() {
103        String name = tfname1.getText();
104        String password = tfpassword1.getText();
105
106        if (name.equals("Nikhil") & password.equals("12345")) {
107            JOptionPane.showMessageDialog(null, "Flight Info");
108        } else {
109            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
110        }
111    }
112
113    void JourneyDetails() {
114        String name = tfname1.getText();
115        String password = tfpassword1.getText();
116
117        if (name.equals("Nikhil") & password.equals("12345")) {
118            JOptionPane.showMessageDialog(null, "Journey Details");
119        } else {
120            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
121        }
122    }
123
124    void BoardingPass() {
125        String name = tfname1.getText();
126        String password = tfpassword1.getText();
127
128        if (name.equals("Nikhil") & password.equals("12345")) {
129            JOptionPane.showMessageDialog(null, "Boarding Pass");
130        } else {
131            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
132        }
133    }
134
135    void cancel() {
136        String name = tfname1.getText();
137        String password = tfpassword1.getText();
138
139        if (name.equals("Nikhil") & password.equals("12345")) {
140            JOptionPane.showMessageDialog(null, "Cancel");
141        } else {
142            JOptionPane.showMessageDialog(null, "Incorrect Name or Password");
143        }
144    }
145
146}
```

[Type here]



```
flight = new JButton("Cancel");
flight.setBackground(Color.BLACK);
flight.setForeground(Color.WHITE);
flight.setBounds(220, 330, 120, 25);
flight.addActionListener(this);
add(flight);

setSize(600, 450);
setLocation(650, 200);
setVisible(true);

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == fetchButton) {
        String PNR = tfPNR.getText();

        try {
            Conn conn = new Conn();

            String query = "select * from reservation where PNR = '" + PNR + "'";

            ResultSet rs = conn.s.executeQuery(query);

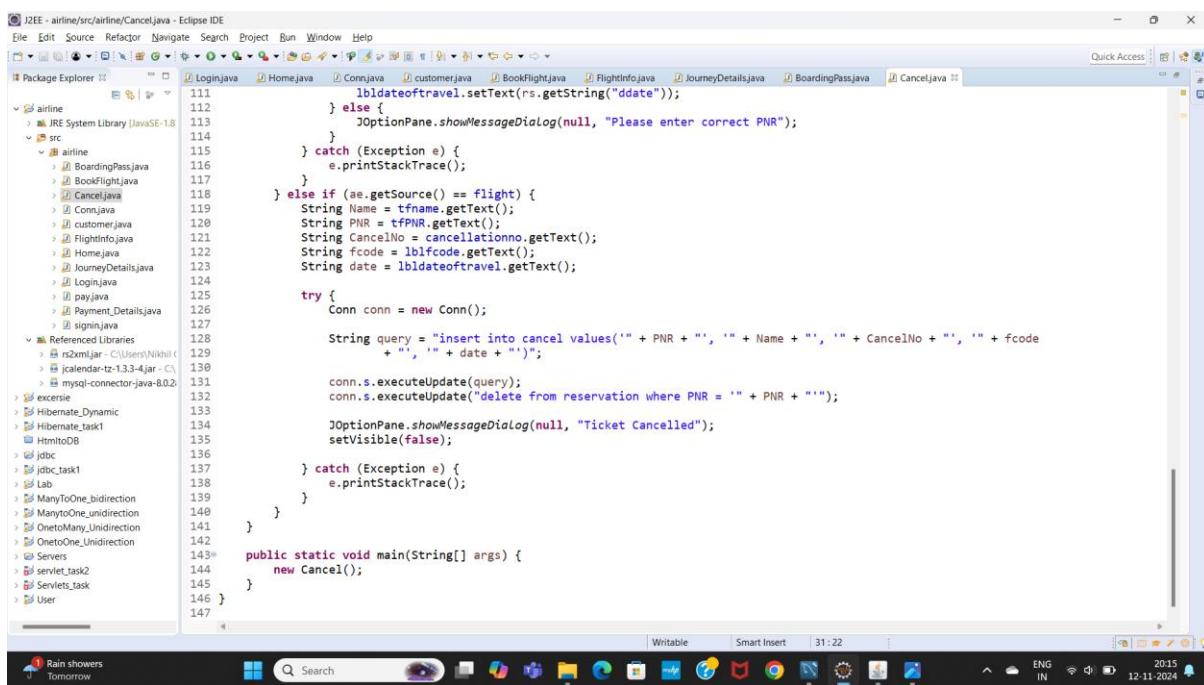
            if (rs.next()) {
                tfname.setText(rs.getString("name"));
                lblfcode.setText(rs.getString("flightcode"));
                lbdateoftravel.setText(rs.getString("ddate"));
            } else {
                JOptionPane.showMessageDialog(null, "Please enter correct PNR");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else if (ae.getSource() == flight) {
        String Name = tfname.getText();
        String PNR = tfPNR.getText();
        String CancelNo = cancellationno.getText();
        String fcode = lblfcode.getText();
        String date = lbdateoftravel.getText();

        try {
            Conn conn = new Conn();

            String query = "insert into cancel values('" + PNR + "', '" + Name + "', '" + CancelNo + "', '" + fcode +
                           "', '" + date + "')";

            conn.s.executeUpdate(query);
            conn.s.executeUpdate("delete from reservation where PNR = '" + PNR + "'");

            JOptionPane.showMessageDialog(null, "Ticket Cancelled");
            setVisible(false);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



```
lbdateoftravel.setText(rs.getString("ddate"));

} else {
    JOptionPane.showMessageDialog(null, "Please enter correct PNR");
}

} catch (Exception e) {
    e.printStackTrace();
}

} else if (ae.getSource() == flight) {
    String Name = tfname.getText();
    String PNR = tfPNR.getText();
    String CancelNo = cancellationno.getText();
    String fcode = lblfcode.getText();
    String date = lbdateoftravel.getText();

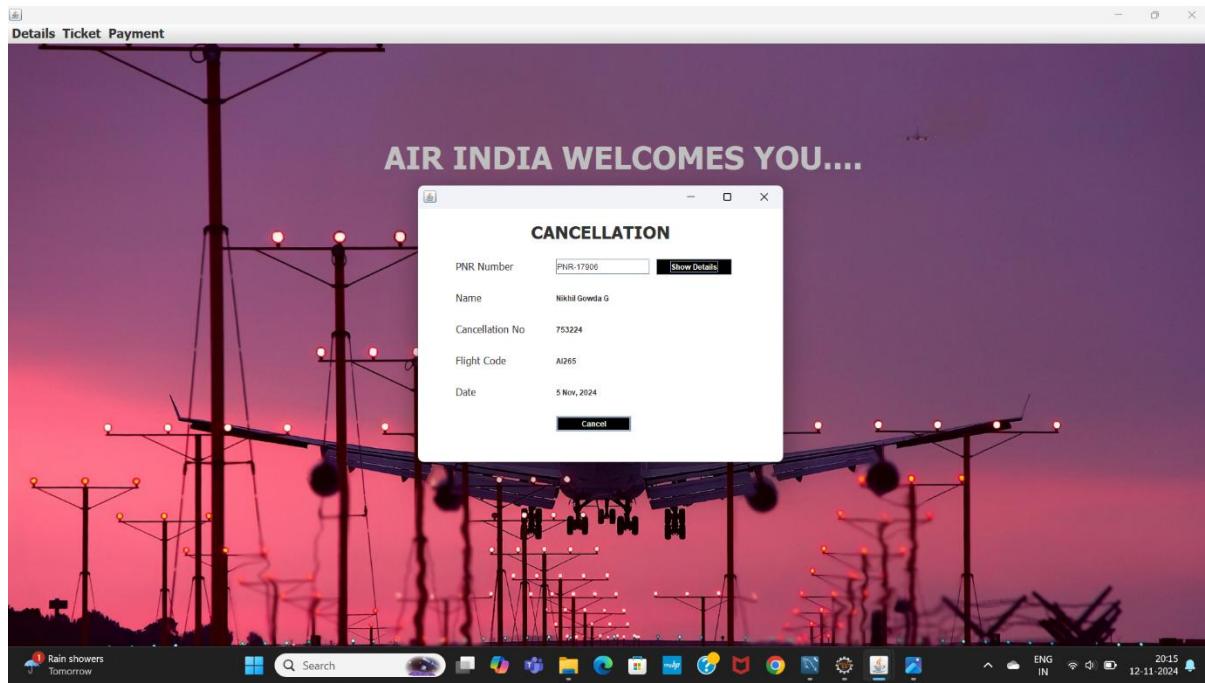
    try {
        Conn conn = new Conn();

        String query = "insert into cancel values('" + PNR + "', '" + Name + "', '" + CancelNo + "', '" + fcode +
                       "', '" + date + "')";

        conn.s.executeUpdate(query);
        conn.s.executeUpdate("delete from reservation where PNR = '" + PNR + "'");

        JOptionPane.showMessageDialog(null, "Ticket Cancelled");
        setVisible(false);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    new Cancel();
}
```



## CHAPTER 5

### TESTING

#### 5.1 Introduction

Testing for Airline Management System includes different levels and types of tests that aim to validate the system's components, performance, and security:

- 1. Unit Testing:** This is the initial level of testing where individual units or components of the system are tested. For example, testing if the login functionality is correctly verifying user credentials or if the booking module is accurately reserving a seat.
- 2. Integration Testing:** After unit testing, integration testing ensures that different components of the system work together correctly. For instance, it validates that the user interface interacts properly with the backend

database and that seat availability is accurately reflected when a user attempts to book a seat.

3. **System Testing:** This phase tests the entire system as a whole. It verifies if the system meets the specified requirements, such as ensuring the correct functioning of the booking process, payment gateway integration, and cancellation of tickets.
4. **Acceptance Testing:** This is typically the final testing phase, where the system is tested against the requirements provided by the stakeholders. In the case of a bus reservation system, it would involve validating that the system meets all the functional and non-functional requirements, such as usability, performance, and security.
5. **Performance Testing:** This type of testing focuses on how well the system performs under stress or load. For example, checking how many users can concurrently make reservations without the system crashing.
6. **Security Testing:** Since airline management systems typically handle sensitive information, including user credentials, payment details, and personal information, security testing ensures that the system is resistant to common vulnerabilities such as SQL injection, data breaches, or unauthorized access.

## 5.2 Types of Testing for Airline Management System

1. **Functional Testing:** Ensures that the core functions of the system, such as user login, bus search, seat booking, and payment processing, work as expected.
2. **Usability Testing:** Ensures the system is user-friendly, intuitive. This could include testing the home page, user profile pages.

3. **Regression Testing:** Whenever new features or fixes are implemented, regression testing ensures that existing functionalities remain unaffected. For example, after implementing the payment gateway, testing ensures that the book flight functionality still works correctly.
4. **End-to-End Testing:** Verifies the entire process flow, from user login to flight booking and payment confirmation to ensure all components work seamlessly together.
5. **Compatibility Testing:** Tests if the system works across different browsers, devices, and operating systems. For instance, ensuring that the booking interface works properly on both desktop and mobile versions of the site.
6. Testing is essential to ensure that Airline management system operates reliably, securely, and efficiently. Proper testing helps ensure that all user-facing features and backend processes perform as expected, offering a smooth experience for users and administrators alike.

### 5.3 Test Case

Test Case ID	Module	Test Description	Input	Expected Output	Actual Output	Pass/Fail
TC_001	Login	Test login with invalid credentials	Username: invalid User Password: invalid Password	Error message: "Invalid login credentials"	Error message: "Invalid login credentials"	pass
TC_002	Cancel Booking	Test attempting to cancel a non-existent booking	PNR: PNR-412482	Error message: "Booking not found"	Error message: "Booking not found"	pass

[Type here]

TC_003	Search Flight	Search for available buses on a valid rout	Route: City A to City B Date: 2023-12-01	List of available buses displayed	List of available buses displayed	pass
TC_004	Login	Test login with invalid credentials (Fail Case)	Username: Wrong User Password: wrong Password	Error message: "Invalid login credentials"	Error message: "Incorrect password" (Fail)	fail
TC_005	Cancel Booking	Test cancel booking with invalid Booking ID (Fail Case)	PNR: PNR-412482	Error message: "Booking not found"	Error message: "Booking ID does not exist" (Fail)	fail

## CHAPTER 6

### CONCLUSION AND FUTURE ENHANCEMENTS

#### 6.1 Conclusion

This project on Airline Management System is the automation of registration process of airline system. The system is able to provide much information like passenger's details, flight details and the booking details. The system allows us to add records when a passenger reserves a ticket. It also allows deleting and updating the records based on passenger's requirements. This project has guided

our path through various aspects of computer science where developing online application plays a major role.

## 6.2 Future Enhancements

As technology evolves and user needs change, there are several future enhancements that could be implemented in airline management system to make it more efficient, user-friendly, and aligned with modern requirements. These enhancements could include improvements in user experience, system scalability, integration with emerging technologies, and more.

Here are some potential future enhancements for Airline Management System:

- Allow customers to submit queries and suggestions
- Create a strong password mechanism.
- AI algorithms that suggest the best available seats based on factors like proximity to the exit, window views, or user preferences.
- Improve the navigational sequence.
- Develop Excel export for bookings.

## 6.3 References

### 1. Academic Journals and Conferences:

- **IEEE Xplore or ACM Digital Library:** You may find papers related to software design, ticketing systems, or Airline Management Systems. You could search for terms like "Airline reservation system," "transportation management software," or "Java-based booking system."
  - [GeeksforGeeks - Online Airline Reservation System](#)
  - [Medium - Airline Ticket Booking System](#)

### 2.Tech Websites & Blogs:

- **Stack Overflow:** While not a traditional news outlet, developers often share solutions, discussions, and code snippets related to building reservation systems.
- **Java World:** A publication focused on Java, where you might find articles about implementing various systems with Java, including transportation systems.

### **3. Books and Texts on Java Programming:**

- "**Java How to Program**" by **Paul Deitel and Harvey Deitel**: Includes exercises related to building various systems in Java.
- "**Core Java Volume I—Fundamentals**" by **Cay S. Horstmann**: Explains core Java concepts with examples that could be adapted to a bus reservation system.

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]