Airline Management System

ABSTRACT

This project on Airline Management System is the automation of registration process of airlines system. The system provides information like passenger's information, flight information, list of all passengers, it allows storing and retrieving data related to the airline industry and make transactions related to air travel etc. The system also allows us to add records when a passenger reserves a ticket. For data storage and retrieval we use MySQL Database. It enables us to add any number of records in our database. The project "Airline Management System" comprises of a large number of flights which belong to a particular airline. The system we have implemented manages different objects viz.

The Airline Management System is a comprehensive Java-based application designed to streamline airline operations. This project includes modules for user login, flight booking, journey details, ticket cancellation, and boarding pass generation. The application uses Java Swing for the GUI and MySQL for the backend database.

· Airline

· Airline Employees

· Customers/Traveller

Each of these accesses a database schema which has corresponding tables.

Language Used - Core Java

Concept Used – Swing

IDE Used – NetBeans

Database Used - MySQL

CONTENTS

CHAPTER 1

INTRODUCTION

Airline Management System is the administration of airports and airlines. It includes the activities of setting the strategy of airports to gather and provide information on airline commercial and operational priorities. It covers a broad overview of the airline management. It is also studied as a branch of study that teaches management of airport and airlines. This provides a broad overview of the airline industry and creates awareness of the underlying marketing, financial, operational, and other factors influencing airline management. This study provides information on airline commercial and operational priorities, along with teaching the key characteristics of aircraft selection and the impact of airport decision making. It provides some amount of automation in airlines management and helps airline system in making their business more efficient. An added attraction for their potential customers. It will also show the attitude of the management that they are aware to the newly introduced technology and ready to adopt them.

1.1Problem Definition

This project on Flight 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 to delete and update the records 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 Need

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 any number of records. The global airline industry continues to grow rapidly, but consistent and robust profitability is elusive. Measured by revenue, the industry has doubled over the past decade, from US$369 billion in 2004 to a projected $746 billion in 2014, according to the International Air Transport Association(IATA).Much of that growth has been driven by low-cost carriers(LCCs), which now control some 25 percent of the worldwide market and which have been expanding rapidly in emerging markets; growth also came from continued gains by carriers in developed markets, the IATA reported. Yet profit margins are still low, less than 3 percent overall. In the commercial aviation sector, just about every group in the aviation industry chain—airports, airplane manufacturers, jet engine makers, travel agents, and service companies, to name a few—turns a profit. It is seemingly ironic that the airline companies that actually move passengers from one place to another, the most crucial link in the chain, struggle to make a profit.

A few factors that directs us to develop a new system are given below -:

1. Faster System
2. Accuracy
3. Reliability

4. Informative
5. Reservations and cancellations from any where to any place.

CHAPTER 2

REQUIREMENTS

2.1 Software Requirement Specifications

Operating System Front End Back End Server Documentation : Windows 10

Frontend Software: Java NetBeans 8.2 : JDK 8

Backend Software: MySQL

2.2 Hardware Requirement Specifications

Computer Processor Core i3 Processor Speed 2.3 GHz Processor Hard Disk 400 GB or more RAM Min 2GB

CHAPTER 3

ENTITY RELATIONSHIP DIAGRAM

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. If the application is primarily a database application, the entity-relationship approach can be used effectively for modeling some parts of the problem. The main focus in ER modeling is the Data Items in the system and the relationship between them. It aims to create conceptual scheme for the Data from the user's perspective. The model thus created is independent of any database model. The ER models are frequently represented as ER diagram. Here we present the ER diagram of the above mentioned project.

CHAPTER 4

SCHEMA DIAGRAM

4.1 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

A database schema can be divided broadly into two categories −

- Physical Database Schema − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- Logical Database Schema − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

CHAPTER 5

IMPLEMENTATION

Login Operation Purpose: To authenticate users by validating their credentials.

Input: - Username: Text field input. - Password: Password field input.

Output: - 'Valid' if the credentials are correct. - Error message if credentials are incorrect.

Code Functionality: This module uses the `Login` class to interact with the MySQL database and validate user credentials. The `Conn` class handles the database connection.

Login.java

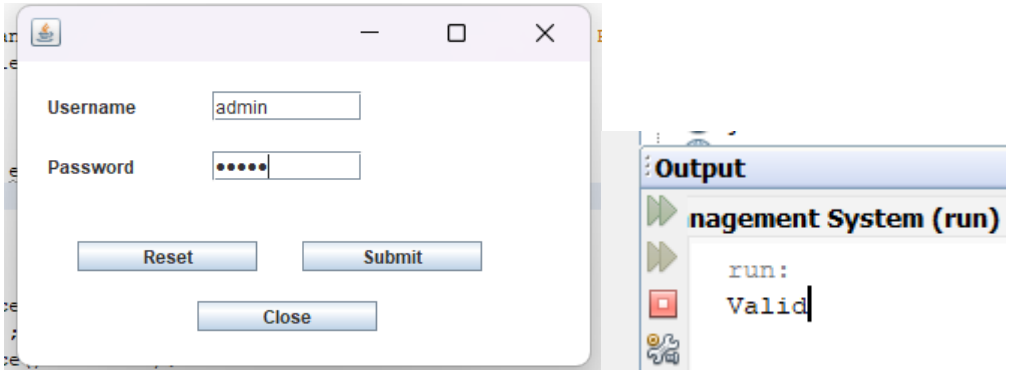
```
package airline.management.system;
import java.awt.Color;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener{
   JButton submit, reset, close;
   JTextField tfusername;
   JPasswordField tfpassword;


   public Login(){

      getContentPane().setBackground(Color.WHITE);
      setLayout(null);


      tfusername = new JTextField();
      tfusername.setBounds(130,20,100,20);
      add(tfusername);

      tfpassword = new JPasswordField();
      tfpassword.setBounds(130,60,100,20);
      add(tfpassword);

      JLabel lblusername = new JLabel("Username");
      lblusername.setBounds(20,20,100,20);
      add(lblusername);

      JLabel lblpassword = new JLabel("Password");
```

```
        lblpassword.setBounds(20,60,100,20);
        add(lblpassword);

        reset = new JButton("Reset");
        reset.setBounds(40, 120, 120, 20);
        reset.addActionListener(this);
        add(reset);

        submit = new JButton("Submit");
        submit.setBounds(190, 120, 120, 20);
        submit.addActionListener(this);
        add(submit);

        close = new JButton("Close");
        close.setBounds(120, 160, 120, 20);
        close.addActionListener(this);
        add(close);


        setSize(400,250);
        setLocation(600,250);
        setVisible(true);

    }

    public void actionPerformed(ActionEvent ae){
        if (ae.getSource()== submit){
            String username = tfusername.getText();
            String password = tfpassword.getText();


            try{
                Conn c = new Conn();

                String query = "select * from login where username = '"+username+"' and password =
'"+password+"'";
                ResultSet rs = c.s.executeQuery(query);

                if (rs.next()){
                    System.out.print("Valid");
                    setVisible(false);

                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Username and Password");
                    setVisible(false);
                }

            }
            catch (Exception e){
```

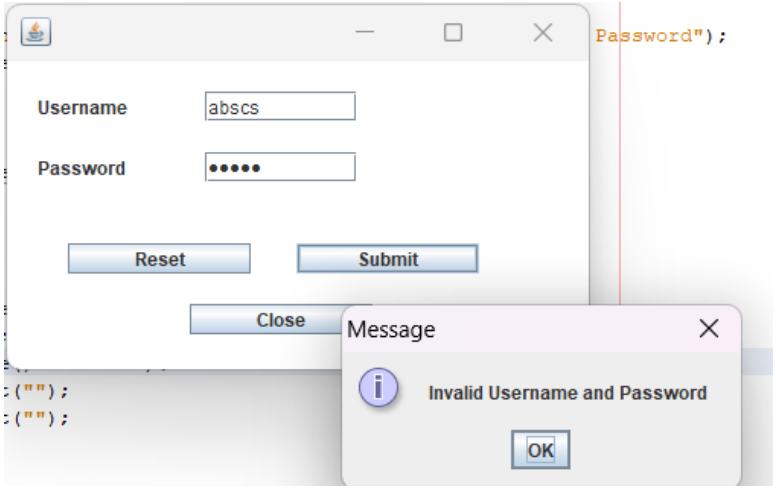
```
        }

      }else if (ae.getSource() == close){
         setVisible(false);
      }else if (ae.getSource() == reset){
         tfusername.setText("");
         tfpassword.setText("");
      }
   }

   public static void main(String [] args){
    new Login();

   }

}
```

Output Login page:

If Password and Username is correct it will print Valid .



If Password and Username is not correct it will print Valid .



JDBC Connection file:

Conn.java

```
package airline.management.system;

import java.sql.*;


public class Conn {

   Connection c;
   Statement s;

   public Conn(){
      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         c = DriverManager.getConnection("jdbc:mysql://localhost:3306/airlinemanagementsystem",
"root", "ujwala123");
          s = c.createStatement();
      } catch (Exception e) {
         e.printStackTrace();

      }

   }
}
```

Database :

```
show databases;
create database airlinemanagementsystem;
use airlinemanagementsystem;

/*Login Table*/

create table login(username varchar(20), password varchar(20));

insert into login values('admin', 'admin');

show tables;
select * from login;

/*Passenger Table*/

create table passenger (name varchar(20), nationality varchar(20), phone varchar(15),
address varchar(50), aadhar varchar(20), gender varchar(20));

select * from passenger;
```

/*flight Table*/

```
create table flight(f_code varchar(20), f_name varchar(20), source varchar(40), destination
varchar(40));

insert into flight values("1001", "AI-1212", "Delhi", "Mumbai");
insert into flight values("1002", "AI-1453", "Delhi", "Goa");
insert into flight values("1003", "AI-1112", "Mumbai", "Chennai");
insert into flight values("1004", "AI-3222", "Delhi", "Amritsar");

select * from flight;

describe flight;
```

/*reservation Table*/

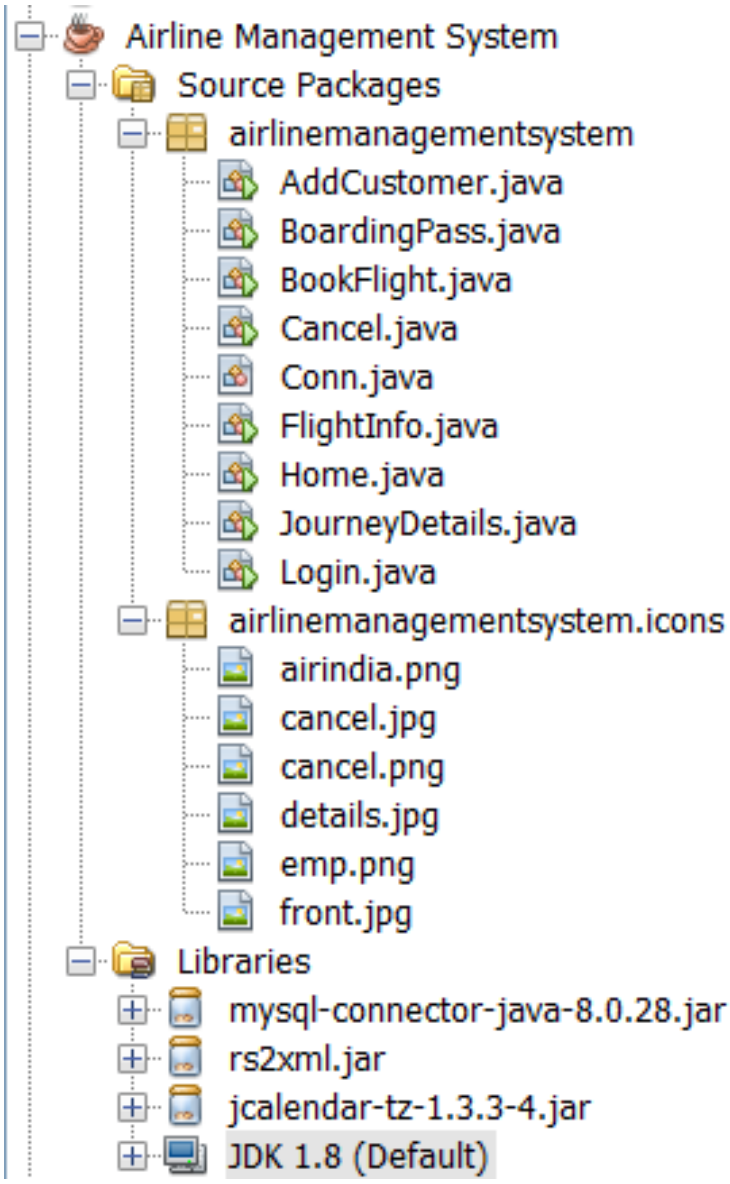
```
create table reservation(PNR varchar(15), TICKET varchar(20), aadhar varchar(20), name
varchar(20), nationality varchar(30), flightname varchar(15), flightcode varchar(20), src
varchar(30), des varchar(30), ddate varchar(30));

select * from reservation;
```

/*cancel Table*/

```
create table cancel(pnr varchar(20), name varchar(40), cancelno varchar(20), fcode
varchar(20), ddate varchar(30));

select * from cancel;
```

	Tables_in_airlinemanagementsystem
▶	cancel
	flight
	login
	passenger
	reservation

Project Hierarchy:



2. Home

Home Page Purpose: Serves as the main navigation hub for accessing different functionalities.

Input: - User clicks on menu items. Output: - Opens respective modules such as 'Flight Details' or 'Boarding Pass'.

Code Functionality: The `Home` class initializes the menu and handles navigation events using the `actionPerformed` method.

Input:

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Home extends JFrame implements ActionListener{

   public Home() {
      setLayout(null);

      ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("airlinemanagementsystem/icons/front.jpg"));
      JLabel image = new JLabel(i1);
      image.setBounds(0, 0, 1925, 900);
      add(image);

      JLabel heading = new JLabel("AIR INDIA WELCOMES YOU");
      heading.setBounds(700, 50, 1500, 60);
      heading.setForeground(Color.BLUE);
      heading.setFont(new Font("Tahoma", Font.PLAIN, 36));
      image.add(heading);

      JMenuBar menubar = new JMenuBar();
      setJMenuBar(menubar);
```

```
    JMenu details = new JMenu("Details");
    menubar.add(details);

    JMenuItem flightDetails = new JMenuItem("Flight Details");
    flightDetails.addActionListener(this);
    details.add(flightDetails);

    JMenuItem customerDetails = new JMenuItem("Add Customer Details");
    customerDetails.addActionListener(this);
    details.add(customerDetails);

    JMenuItem bookFlight = new JMenuItem("Book Flight");
    bookFlight.addActionListener(this);
    details.add(bookFlight);

    JMenuItem journeyDetails = new JMenuItem("Journey Details");
    journeyDetails.addActionListener(this);
    details.add(journeyDetails);

    JMenuItem ticketCancellation = new JMenuItem("Cancel Ticket");
    ticketCancellation.addActionListener(this);
    details.add(ticketCancellation);

    JMenu ticket = new JMenu("Ticket");
    menubar.add(ticket);

    JMenuItem boardingPass = new JMenuItem("Boarding Pass");
    boardingPass.addActionListener(this);
    ticket.add(boardingPass);


    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setVisible(true);
  }

  public void actionPerformed(ActionEvent ae) {
    String text = ae.getActionCommand();

    if (text.equals("Add Customer Details")) {
      new AddCustomer();
    } 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();
    }
    }
    public static void main(String[] args) {
        new Home();
    }
}
```

Output:

3.Add Customer Details

Add Customer Details Purpose: To collect and save customer details in the database.

Input: - Name, Nationality, Address, Gender, Aadhar, Phone: User inputs.

Output: - Confirmation message upon successful data entry.

Code Functionality: The `AddCustomer` class collects inputs via text fields and radio buttons and stores the data in the `passenger` table.

INPUT:

AddCustomer.java

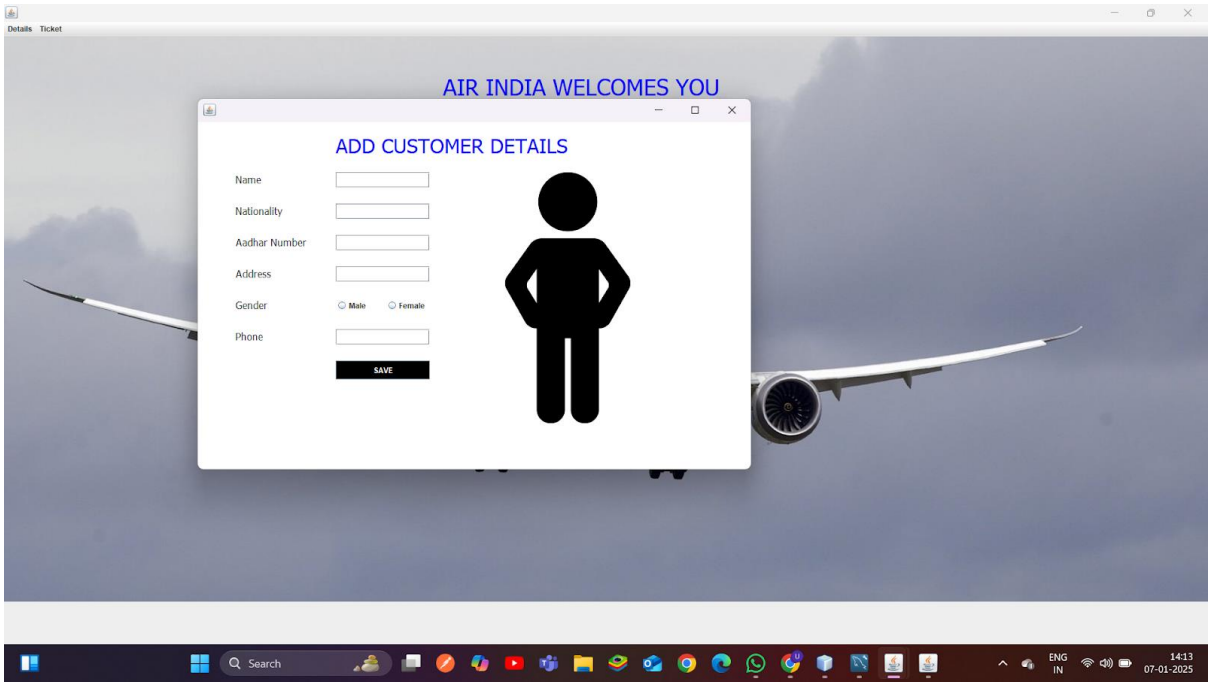
```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class AddCustomer extends JFrame implements ActionListener{

   JTextField tfname, tfphone, tfaadhar, tfnationality, tfaddress;
   JRadioButton rbmale, rbfemale;

   public AddCustomer() {
      getContentPane().setBackground(Color.WHITE);
      setLayout(null);

      JLabel heading = new JLabel("ADD CUSTOMER DETAILS");
      heading.setBounds(220, 20, 500, 35);
      heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
      heading.setForeground(Color.BLUE);
      add(heading);

      JLabel lblname = new JLabel("Name");
      lblname.setBounds(60, 80, 150, 25);
      lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
      add(lblname);

      tfname = new JTextField();
      tfname.setBounds(220, 80, 150, 25);
      add(tfname);
```

```
JLabel lblnationality = new JLabel("Nationality");
lblnationality.setBounds(60, 130, 150, 25);
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblnationality);

tfnationality = new JTextField();
tfnationality.setBounds(220, 130, 150, 25);
add(tfnationality);

JLabel lblaadhar = new JLabel("Aadhar Number");
lblaadhar.setBounds(60, 180, 150, 25);
lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblaadhar);

tfaadhar = new JTextField();
tfaadhar.setBounds(220, 180, 150, 25);
add(tfaadhar);

JLabel lbladdress = new JLabel("Address");
lbladdress.setBounds(60, 230, 150, 25);
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lbladdress);

tfaddress = new JTextField();
tfaddress.setBounds(220, 230, 150, 25);
add(tfaddress);

JLabel lblgender = new JLabel("Gender");
lblgender.setBounds(60, 280, 150, 25);
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblgender);

ButtonGroup gendergroup = new ButtonGroup();

rbmale = new JRadioButton("Male");
rbmale.setBounds(220, 280, 70, 25);
rbmale.setBackground(Color.WHITE);
add(rbmale);

rbfemale = new JRadioButton("Female");
rbfemale.setBounds(300, 280, 70, 25);
rbfemale.setBackground(Color.WHITE);
add(rbfemale);

gendergroup.add(rbmale);
gendergroup.add(rbfemale);
```

```
        JLabel lblphone = new JLabel("Phone");
        lblphone.setBounds(60, 330, 150, 25);
        lblphone.setFont(new Font("Tahoma", Font.PLAIN, 16));
        add(lblphone);

        tfphone = new JTextField();
        tfphone.setBounds(220, 330, 150, 25);
        add(tfphone);

        JButton save = new JButton("SAVE");
        save.setBackground(Color.BLACK);
        save.setForeground(Color.WHITE);
        save.setBounds(220, 380, 150, 30);
        save.addActionListener(this);
        add(save);

        ImageIcon image = new
ImageIcon(ClassLoader.getSystemResource("airlinemanagementsystem/icons/emp.png"));
        JLabel lblimage = new JLabel(image);
        lblimage.setBounds(450, 80, 280, 400);
        add(lblimage);

        setSize(900, 600);
        setLocation(300, 150);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        String name = tfname.getText();
        String nationality = tfnationality.getText();
        String phone = tfphone.getText();
        String address = tfaddress.getText();
        String aadhar = tfaadhar.getText();
        String gender = null;
        if (rbmale.isSelected()) {
            gender = "Male";
        } else {
            gender = "Female";
        }

        try {
            Conn conn = new Conn();

            String query = "insert into passenger values('"+name+"', '"+nationality+"',
'"+phone+"', '"+address+"', '"+aadhar+"', '"+gender+"')";

            conn.s.executeUpdate(query);
```

```
            JOptionPane.showMessageDialog(null, "Customer Details Added Successfully");

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

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

Output:




```
18 •  select * from passenger;
19
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	nationality	phone	address	aadhar	gender
Ujwala	Indian	9967852728	Mumbai	492853319507	Female
Sangeeta	India	9892986527	Uttar Pradesh	290487507799	Female
Sujit	India	9892986772	Banglore	492753319507	Male
Sonam	india	8887252325	Thane	123456789012	Female

4.Flight Details:

INPUT:

FlightInfo.java

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import net.proteanit.sql.DbUtils;

public class FlightInfo extends JFrame{

    public FlightInfo() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JTable table = new JTable();

        try {
            Conn conn = new Conn();

            ResultSet rs = conn.s.executeQuery("select * from flight");
            table.setModel(DbUtils.resultSetToTableModel(rs));
        } catch(Exception e) {
            e.printStackTrace();
        }

        JScrollPane jsp = new JScrollPane(table);
        jsp.setBounds(0, 0, 800, 500);
        add(jsp);

        setSize(800, 500);
        setLocation(400, 200);
        setVisible(true);
    }

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

Output:


```
29 •  select * from flight;
30
31 •  describe flight;
32
33    /*reservation Table*/
34
```

Result Grid | Filter Rows:

f_code	f_name	source	destination
1001	AI-1212	Delhi	Mumbai
1002	AI-1453	Delhi	Goa
1003	AI-1112	Mumbai	Chennai
1004	AI-3222	Delhi	Amritsar
1005	AI-1212	Delhi	Ayodhya

5. Book Flight

INput:

BookFlight.java

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import com.toedter.calendar.JDateChooser;
import java.util.*;

public class BookFlight extends JFrame implements ActionListener{

   JTextField tfaadhar;
   JLabel tfname, tfnationality, tfaddress, labelgender, labelfname, labelfcode;
   JButton bookflight, fetchButton, flight;
   Choice source, destination;
   JDateChooser dcdate;

   public BookFlight() {
      getContentPane().setBackground(Color.WHITE);
      setLayout(null);

      JLabel heading = new JLabel("Book Flight");
      heading.setBounds(420, 20, 500, 35);
      heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
      heading.setForeground(Color.BLUE);
      add(heading);

      JLabel lblaadhar = new JLabel("Aadhar");
      lblaadhar.setBounds(60, 80, 150, 25);
      lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
      add(lblaadhar);

      tfaadhar = new JTextField();
      tfaadhar.setBounds(220, 80, 150, 25);
      add(tfaadhar);

      fetchButton = new JButton("Fetch User");
      fetchButton.setBackground(Color.BLACK);
      fetchButton.setForeground(Color.WHITE);
      fetchButton.setBounds(380, 80, 120, 25);
      fetchButton.addActionListener(this);
      add(fetchButton);

      JLabel lblname = new JLabel("Name");
      lblname.setBounds(60, 130, 150, 25);
```

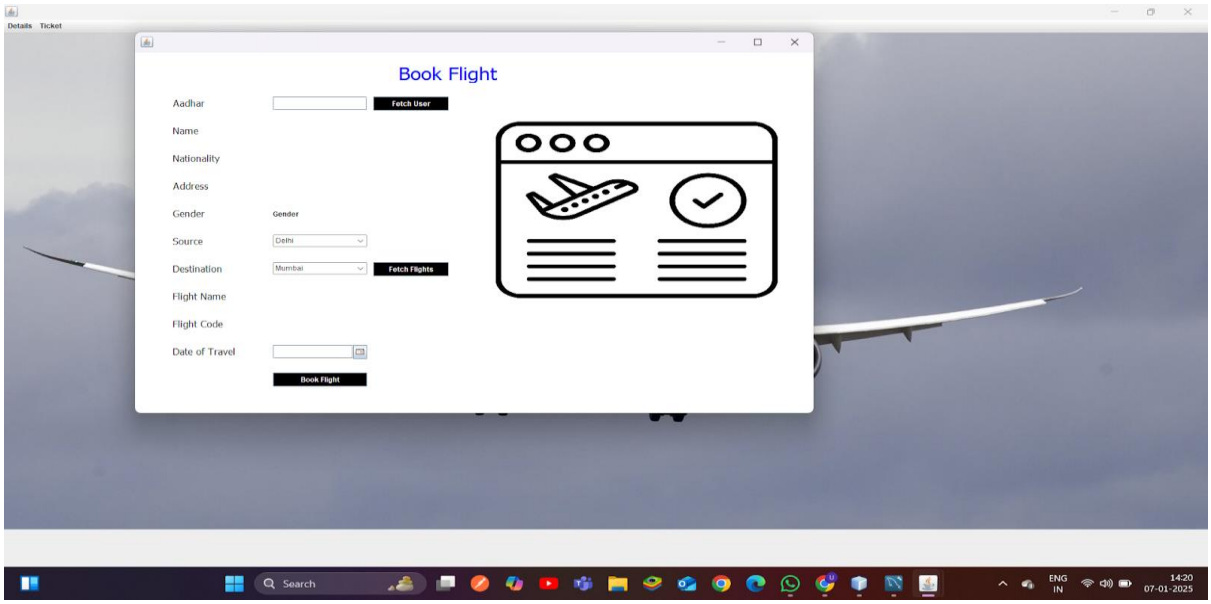
```
lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblname);

tfname = new JLabel();
tfname.setBounds(220, 130, 150, 25);
add(tfname);

JLabel lblnationality = new JLabel("Nationality");
lblnationality.setBounds(60, 180, 150, 25);
lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblnationality);

tfnationality = new JLabel();
tfnationality.setBounds(220, 180, 150, 25);
add(tfnationality);

JLabel lbladdress = new JLabel("Address");
lbladdress.setBounds(60, 230, 150, 25);
lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lbladdress);

tfaddress = new JLabel();
tfaddress.setBounds(220, 230, 150, 25);
add(tfaddress);

JLabel lblgender = new JLabel("Gender");
lblgender.setBounds(60, 280, 150, 25);
lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblgender);

labelgender = new JLabel("Gender");
labelgender.setBounds(220, 280, 150, 25);
add(labelgender);

JLabel lblsource = new JLabel("Source");
lblsource.setBounds(60, 330, 150, 25);
lblsource.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lblsource);

source = new Choice();
source.setBounds(220, 330, 150, 25);
add(source);

JLabel lbldest = new JLabel("Destination");
lbldest.setBounds(60, 380, 150, 25);
lbldest.setFont(new Font("Tahoma", Font.PLAIN, 16));
add(lbldest);

destination = new Choice();
destination.setBounds(220, 380, 150, 25);
add(destination);
```

```
        try {
            Conn c = new Conn();
            String query = "select * from flight";
            ResultSet rs = c.s.executeQuery(query);

            while(rs.next()) {
                source.add(rs.getString("source"));
                destination.add(rs.getString("destination"));
            }

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

        flight = new JButton("Fetch Flights");
        flight.setBackground(Color.BLACK);
        flight.setForeground(Color.WHITE);
        flight.setBounds(380, 380, 120, 25);
        flight.addActionListener(this);
        add(flight);

        JLabel lblfname = new JLabel("Flight Name");
        lblfname.setBounds(60, 430, 150, 25);
        lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
        add(lblfname);

        labelfname = new JLabel();
        labelfname.setBounds(220, 430, 150, 25);
        add(labelfname);

        JLabel lblfcode = new JLabel("Flight Code");
        lblfcode.setBounds(60, 480, 150, 25);
        lblfcode.setFont(new Font("Tahoma", Font.PLAIN, 16));
        add(lblfcode);

        labelfcode = new JLabel();
        labelfcode.setBounds(220, 480, 150, 25);
        add(labelfcode);

        JLabel lbldate = new JLabel("Date of Travel");
        lbldate.setBounds(60, 530, 150, 25);
        lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));
        add(lbldate);

        dcdate = new JDateChooser();
        dcdate.setBounds(220, 530, 150, 25);
        add(dcdate);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("airlinemanagementsystem/icons/details.jpg"));
```

```
    Image i2 = i1.getImage().getScaledInstance(450, 320, Image.SCALE_DEFAULT);
    ImageIcon image = new ImageIcon(i2);
    JLabel lblimage = new JLabel(image);
    lblimage.setBounds(550, 80, 500, 410);
    add(lblimage);

    bookflight = new JButton("Book Flight");
    bookflight.setBackground(Color.BLACK);
    bookflight.setForeground(Color.WHITE);
    bookflight.setBounds(220, 580, 150, 25);
    bookflight.addActionListener(this);
    add(bookflight);

    setSize(1100, 700);
    setLocation(200, 50);
    setVisible(true);
}

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 dest = destination.getSelectedItem();
        try {
            Conn conn = new Conn();

            String query = "select * from flight where source = '"+src+"' and destination = '"+dest+"'";

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

            if (rs.next()) {
```

```
                labelfname.setText(rs.getString("f_name"));
                labelfcode.setText(rs.getString("f_code"));
            } 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 = tfnationality.getText();
        String flightname = labelfname.getText();
        String flightcode = labelfcode.getText();
        String src = source.getSelectedItem();
        String des = destination.getSelectedItem();
        String ddate = ((JTextField) dcdate.getDateEditor().getUiComponent()).getText();

        try {
            Conn conn = new Conn();

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

            conn.s.executeUpdate(query);

            JOptionPane.showMessageDialog(null, "Ticket Booked Successfully");

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

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

Output:



Result Grid | Filter Rows:

f_code	f_name	source	destination
1001	AI-1212	Delhi	Mumbai
1002	AI-1453	Delhi	Goa
1003	AI-1112	Mumbai	Chennai
1004	AI-3222	Delhi	Amritsar
1005	AI-1212	Delhi	Ayodhya

```
37 • select * from reservation;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PNR	TICKET	aadhar	name	nationality	flightname	flightcode	src	des	ddate
PNR-861258	TIC-4430	492753319507	Sujit	India	AI-1112	1003	Mumbai	Chennai	24 Feb, 2025

6. Journey Details:

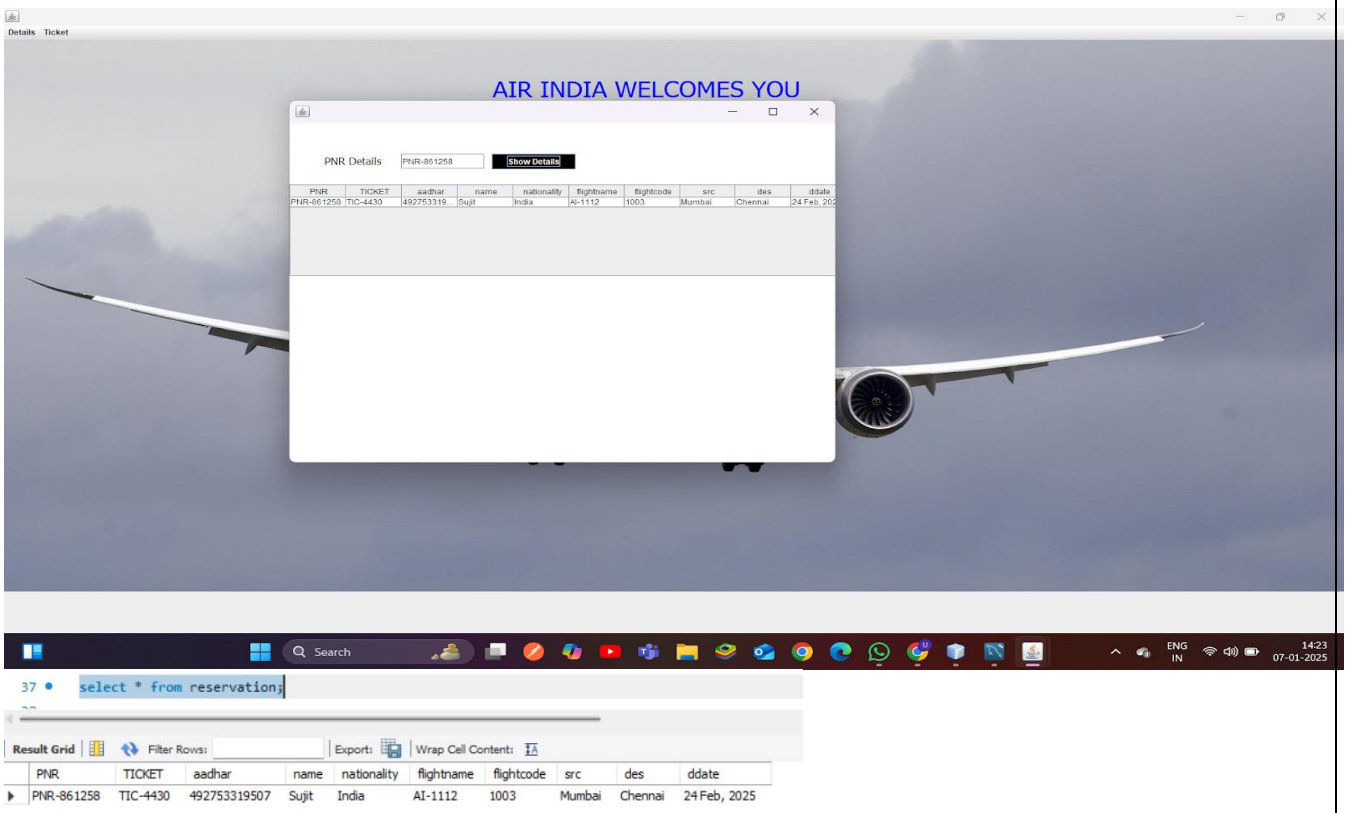
Input:

JourneyDeatils.java

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.sql.*;
import java.awt.event.*;
import net.proteanit.sql.DbUtils;

public class JourneyDetails extends JFrame implements ActionListener{
   JTable table;
   JTextField pnr;
   JButton show;

   public JourneyDetails() {

      getContentPane().setBackground(Color.WHITE);
      setLayout(null);

      JLabel lblpnr = new JLabel("PNR Details");
      lblpnr.setFont(new Font("Tahoma", Font.PLAIN, 16));
      lblpnr.setBounds(50, 50, 100, 25);
      add(lblpnr);

      pnr = new JTextField();
      pnr.setBounds(160, 50, 120, 25);
      add(pnr);

      show = new JButton("Show Details");
      show.setBackground(Color.BLACK);
      show.setForeground(Color.WHITE);
      show.setBounds(290, 50, 120, 25);
      show.addActionListener(this);
      add(show);

      table = new JTable();

      JScrollPane jsp = new JScrollPane(table);
      jsp.setBounds(0, 100, 800, 150);
      jsp.setBackground(Color.WHITE);
      add(jsp);

      setSize(800, 600);
      setLocation(400, 150);
      setVisible(true);
   }
```

```
    public void actionPerformed(ActionEvent ae) {
        try {
            Conn conn = new Conn();
            ResultSet rs = conn.s.executeQuery("select * from reservation where PNR =
'"+pnr.getText()+"'");

            if (!rs.isBeforeFirst()) {
                JOptionPane.showMessageDialog(null, "No Information Found");
                return;
            }
            table.setModel(DbUtils.resultSetToTableModel(rs));
        } catch(Exception e) {
            e.printStackTrace();
        }
    }

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

Output:



6. Cancel Ticket

Input:

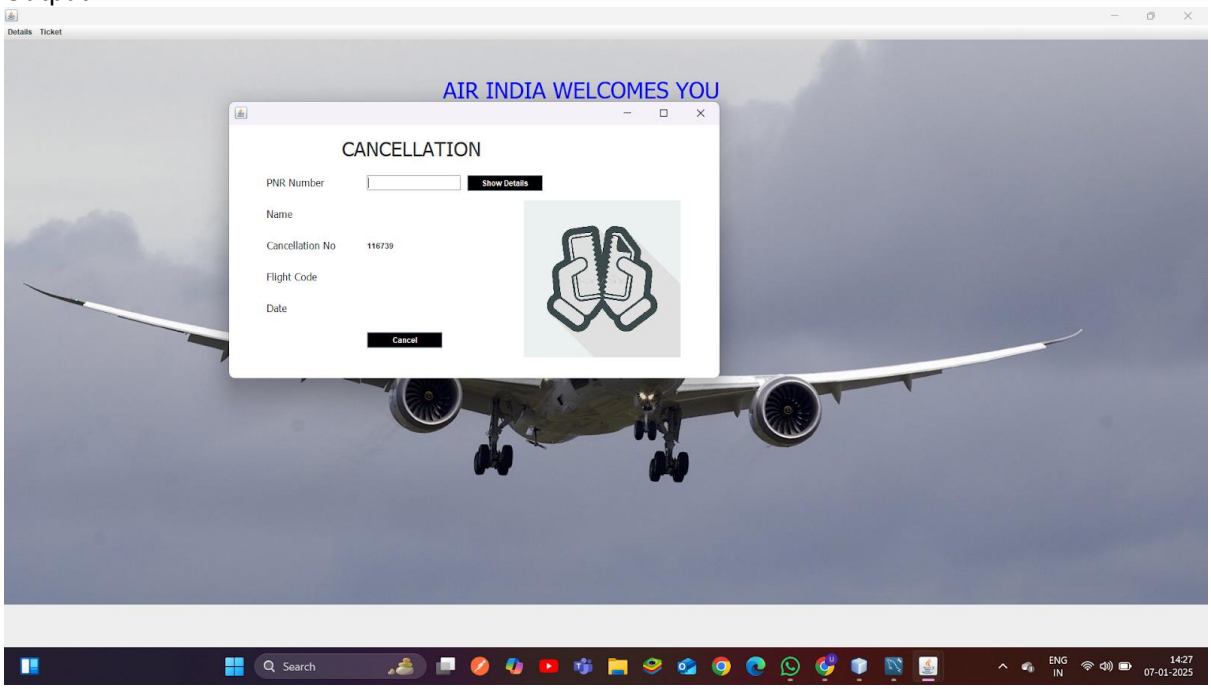
Cancel.java

```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class Cancel extends JFrame implements ActionListener{

   JTextField tfpnr;
   JLabel tfname, cancellationno, lblfcode, lbldateoftravel;
   JButton fetchButton, flight;

   public Cancel() {
      getContentPane().setBackground(Color.WHITE);
      setLayout(null);

      Random random = new Random();

      JLabel heading = new JLabel("CANCELLATION");
      heading.setBounds(180, 20, 250, 35);
      heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
      add(heading);

      ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("airlinemanagementsystem/icons/cancel.jpg"));
      Image i2 = i1.getImage().getScaledInstance(250, 250, Image.SCALE_DEFAULT);
      ImageIcon i3 = new ImageIcon(i2);
      JLabel image = new JLabel(i3);
      image.setBounds(470, 120, 250, 250);
      add(image);

      JLabel lblaadhar = new JLabel("PNR Number");
      lblaadhar.setBounds(60, 80, 150, 25);
      lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
      add(lblaadhar);

      tfpnr = new JTextField();
      tfpnr.setBounds(220, 80, 150, 25);
      add(tfpnr);

      fetchButton = new JButton("Show Details");
      fetchButton.setBackground(Color.BLACK);
      fetchButton.setForeground(Color.WHITE);
```

```
    fetchButton.setBounds(380, 80, 120, 25);
    fetchButton.addActionListener(this);
    add(fetchButton);

    JLabel lblname = new JLabel("Name");
    lblname.setBounds(60, 130, 150, 25);
    lblname.setFont(new Font("Tahoma", Font.PLAIN, 16));
    add(lblname);

    tfname = new JLabel();
    tfname.setBounds(220, 130, 150, 25);
    add(tfname);

    JLabel lblnationality = new JLabel("Cancellation No");
    lblnationality.setBounds(60, 180, 150, 25);
    lblnationality.setFont(new Font("Tahoma", Font.PLAIN, 16));
    add(lblnationality);

    cancellationno = new JLabel("" + random.nextInt(1000000));
    cancellationno.setBounds(220, 180, 150, 25);
    add(cancellationno);

    JLabel lbladdress = new JLabel("Flight Code");
    lbladdress.setBounds(60, 230, 150, 25);
    lbladdress.setFont(new Font("Tahoma", Font.PLAIN, 16));
    add(lbladdress);

    lblfcode = new JLabel();
    lblfcode.setBounds(220, 230, 150, 25);
    add(lblfcode);

    JLabel lblgender = new JLabel("Date");
    lblgender.setBounds(60, 280, 150, 25);
    lblgender.setFont(new Font("Tahoma", Font.PLAIN, 16));
    add(lblgender);

    lbldateoftravel = new JLabel();
    lbldateoftravel.setBounds(220, 280, 150, 25);
    add(lbldateoftravel);

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

    setSize(800, 450);
    setLocation(350, 150);
    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"));
                    lbldateoftravel.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 = lbldateoftravel.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();
    }
}
```

Output:




```
43 •  select * from cancel;
```

Result Grid | Filter Rows: | Export:

pnr	name	cancelno	fcode	ddate
PNR-63187	Sonam	234	1001	15 Jan, 2025

7. Boarding Pass

Boarding Pass Purpose: To generate and display a boarding pass for a specific PNR.

Input: - PNR: User inputs the unique PNR number.

Output: - Displays passenger details, source, destination, flight name, and date.

Code Functionality: The `BoardingPass` class retrieves data from the `reservation` table based on the entered PNR. If no matching record is found, an error message is shown.

Input:

BoardingPass.java

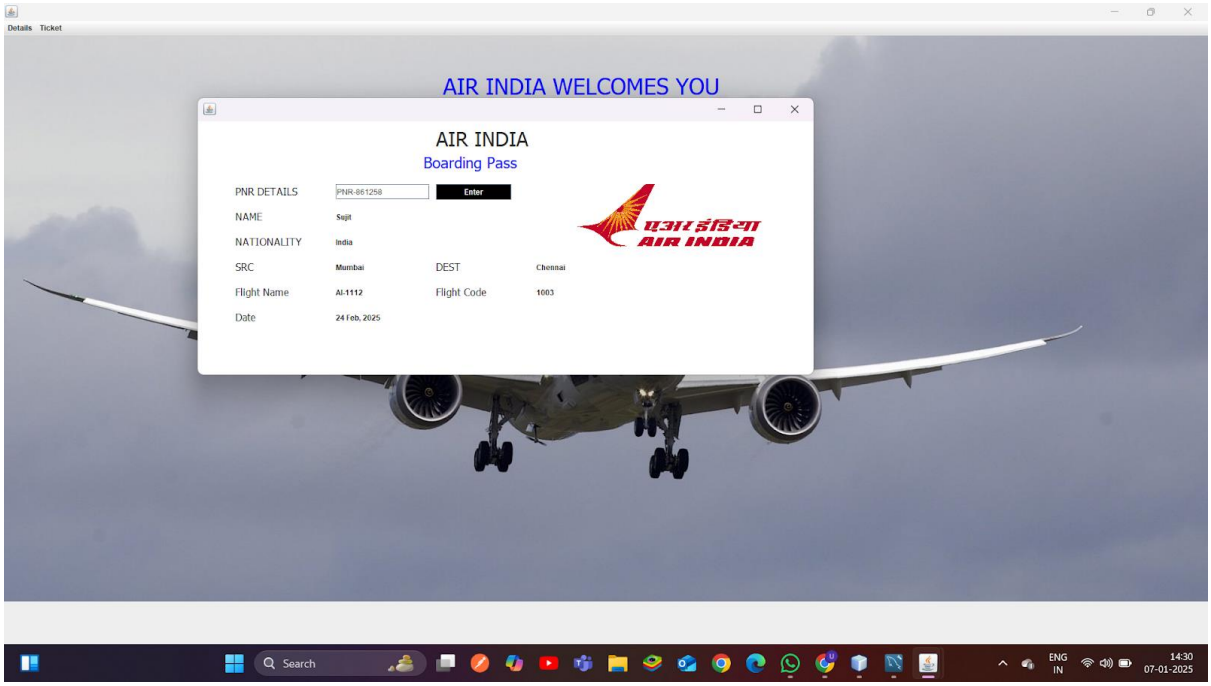
```
package airlinemanagementsystem;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.*;

public class BoardingPass extends JFrame implements ActionListener{

   JTextField tfpnr;
   JLabel tfname, tfnationality, lblsrc, lbldest, labelfname, labelfcode, labeldate;
   JButton fetchButton;

   public BoardingPass() {
      getContentPane().setBackground(Color.WHITE);
      setLayout(null);

      JLabel heading = new JLabel("AIR INDIA");
      heading.setBounds(380, 10, 450, 35);
      heading.setFont(new Font("Tahoma", Font.PLAIN, 32));
      add(heading);

      JLabel subheading = new JLabel("Boarding Pass");
      subheading.setBounds(360, 50, 300, 30);
      subheading.setFont(new Font("Tahoma", Font.PLAIN, 24));
      subheading.setForeground(Color.BLUE);
      add(subheading);

      JLabel lblaadhar = new JLabel("PNR DETAILS");
      lblaadhar.setBounds(60, 100, 150, 25);
      lblaadhar.setFont(new Font("Tahoma", Font.PLAIN, 16));
      add(lblaadhar);
```

```
tfpnr = new JTextField();
tfpnr.setBounds(220, 100, 150, 25);
add(tfpnr);

fetchButton = new JButton("Enter");
fetchButton.setBackground(Color.BLACK);
fetchButton.setForeground(Color.WHITE);
fetchButton.setBounds(380, 100, 120, 25);
fetchButton.addActionListener(this);
add(fetchButton);

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

tfname = new JLabel();
tfname.setBounds(220, 140, 150, 25);
add(tfname);

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

tfnationality = new JLabel();
tfnationality.setBounds(220, 180, 150, 25);
add(tfnationality);

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

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

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

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

JLabel lblfname = new JLabel("Flight Name");
lblfname.setBounds(60, 260, 150, 25);
lblfname.setFont(new Font("Tahoma", Font.PLAIN, 16));
```

```
        add(lblfname);

        labelfname = new JLabel();
        labelfname.setBounds(220, 260, 150, 25);
        add(labelfname);

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

        labelfcode = new JLabel();
        labelfcode.setBounds(540, 260, 150, 25);
        add(labelfcode);

        JLabel lbldate = new JLabel("Date");
        lbldate.setBounds(60, 300, 150, 25);
        lbldate.setFont(new Font("Tahoma", Font.PLAIN, 16));
        add(lbldate);

        labeldate = new JLabel();
        labeldate.setBounds(220, 300, 150, 25);
        add(labeldate);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("airlinemanagementsystem/icons/airindia.png"));
        Image i2 = i1.getImage().getScaledInstance(300, 230, Image.SCALE_DEFAULT);
        ImageIcon image = new ImageIcon(i2);
        JLabel lblimage = new JLabel(image);
        lblimage.setBounds(600, 0, 300, 300);
        add(lblimage);

        setSize(1000, 450);
        setLocation(300, 150);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        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"));
                tfnationality.setText(rs.getString("nationality"));
                lblsrc.setText(rs.getString("src"));
```

```
            lbldest.setText(rs.getString("des"));
            labelfname.setText(rs.getString("flightname"));
            labelfcode.setText(rs.getString("flightcode"));
            labeldate.setText(rs.getString("ddate"));
          } else {
            JOptionPane.showMessageDialog(null, "Please enter correct PNR");
          }
        } catch (Exception e) {
          e.printStackTrace();
        }
    }

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

Output:



Conclusion

The Airline Management System is a robust solution for managing airline operations, offering intuitive modules for different user needs. It demonstrates effective use of Java Swing and MySQL, showcasing a blend of frontend and backend programming. The modular design ensures scalability and easy maintenance.

Thank You…