# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belgaum- 590018**

**A**

# Mini Project Report

**on**

## “AIRLINE MANAGEMENT SYSTEM”

***Submitted in partial fulfillment of Bachelors Engineering Degree** in*

**INFORMATION SCIENCE AND ENGINEERING**

**IV SEMESTER**

**ADVANCED JAVA(BIS402)**

Submitted
by

**Anil Kumar G R[1HK23IS009]**

Under the guidance of

**Prof. Sneha k**

**Assistant Professor**

**Department of Information Science and Engineering**

Accredited by NBA

**2024-2025**

**HKBK COLLEGE OF ENGINEERING**

**22/1,Nagawara,Bengaluru–5600045.**

**E-mail:info@hkbk.edu.in, URL:www.hkbk.edu.in**

# HKBK COLLEGE OF ENGINEERING

**BENGALURU – 560 045**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**A**

## MINI PROJECT REPORT

**on**

**"AIRLINE MANAGEMENT SYSTEM"**

*Submitted in partial fulfillment of the Bachelor of Engineering Degree.*

**IV Semester, Department of Information Science and Engineering**

**2024-25**

**SUBMITTED BY:**

**ANIL KUMAR G R - 1HK23IS009**

# ACKNOWLEDGEMENT

I would like to express my sincere regards and heartfelt acknowledgement to all those who helped make this project possible. Many individuals worked behind the scenes to support and guide me, and the names listed below represent just a few of them.

I take this opportunity to express my deepest gratitude to **Mr. C.M. Ibrahim, Chairman, Mr. C.M. Faiz Mohammed, Director,** and **Dr. Mohammed Riyaz Ahmed, Principal** for providing the necessary infrastructure and support that enabled me to complete the project on time.

I am deeply indebted to Dr. A. Syed Mustafa, Professor & Head, Department of Information **Science and Engineering, and Prof. Sneha K, Assistant Professor,** for their invaluable encouragement and support in the successful completion of my project.

A special and earnest word of thanks goes to the **Department ISE Coordinator, Prof. Sneha K,** for his constant assistance, support, patience, and constructive suggestions that greatly contributed to the betterment of my project.

I am also extremely thankful to the **teaching and non-teaching staff of the Department of Information Science and Engineering** .

**Anil Kumar G R**
**[1HK23IS009]**

# ABSTRACT

The Airline Management System is a software application developed to automate the core functions of an airline's booking and management process. It simplifies the registration of passengers, scheduling of flights, and management of ticket reservations. The system stores and retrieves data such as passenger details, flight information, and booking history, allowing users to access and manage records efficiently. Built using a MySQL database for backend storage, the system supports adding an unlimited number of records. It ensures seamless data handling during ticket reservations, making it easier to update flight schedules and passenger lists in real time. This project is capable of managing a wide range of flights associated with a specific airline, offering a centralized platform to oversee airline operations. By automating these processes, the system reduces manual errors, improves service efficiency, and enhances the overall user experience for both staff and passengers.

# TABLE OF 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.

### 1.1 Problem 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.

The airline industry is one of the foremost dynamic and complex segments within the worldwide economy. It includes the coordination of various forms such as flight planning, ticket reservations, team assignments, client benefit, upkeep operations, and budgetary administration. Customarily, numerous of these errands were dealt with physically or through standalone frameworks, driving to wasteful aspects, expanded chances of mistake, and higher operational costs. As discuss travel has gotten to be more available and request has risen, the require for a centralized and computerized framework to oversee these operations has ended up basic. Scope of the Project.

An **Airline Management System** (AMS) could be a computer program stage planned to address these challenges by coordination all center capacities of an carrier into a single, cohesive framework. It encourages smooth communication between diverse offices, diminishes redundancies, and ensures that both clients and aircraft staff involvement streamlined and error-free forms. The carrier industry could be a foundation of worldwide network, playing a significant part in trade, tourism, and co ordinations. Airlines transport millions of travelers and tons of cargo every day over landmasses, making time efficient and secure travel conceivable. In any case, working an aircraft may be a profoundly complex errand, including endless moving parts each of which must be accurately facilitated to maintain a strategic distance from delays, safety issues, and client disappointment.

As discuss travel proceeds to develop, aircrafts are beneath weight to move forward effectiveness, diminish costs, and upgrade client involvement. These requests have given rise to the selection of computerized frameworks that oversee day-to-day operations. Among these, the **Airline Management System** (AMS) stands out as a center mechanical arrangement for overseeing aircraft operations viably.

An **Airline Management System** may be a centralized computer program arrangement that robotizes and integrates various operational capacities of an carrier. It permits carrier chairmen to oversee flights, bookings, client information, group planning, upkeep, and accounts from a single stage. This not as it were progresses productivity and coordination but moreover empowers real-time decision-making and benefit customization.

# CHAPTER 2

## LITERATURE REVIEW

The advancement of computerized Carrier Administration Frameworks has altogether changed the way discuss travel is overseen and worked. Customarily, carrier ticketing and administration were taken care of physically, which driven to issues such as overbooking, planning clashes, and wasteful client benefit. With headways in computer program advancement and database administration, mechanized frameworks have supplanted obsolete forms to supply real-time upgrades, quicker booking, and way better client information dealing with.

### 2.1 Java Swing for GUI Improvement

Java Swing may be a portion of the Java Establishment Classes (JFC) and gives a set of GUI components for building platform-independent desktop applications. Concurring to Oracle's official documentation, Swing offers a wealthy set of widgets and is built on the Model-View-Controller (MVC) design, permitting for adaptable and responsive client interfacing. In this extend, Swing was utilized to plan shapes for login, client enrollment, flight administration, ticket booking, and cancellations.

### 2.2 MySQL for Backend Database

MySQL is an open-source social database administration framework broadly utilized in scholarly and commercial applications. It bolsters SQL (Organized Inquiry Dialect) and gives highlights such as Corrosive compliance, information security, and versatility. MySQL's compatibility with Java through JDBC (Java Database Network) makes it a appropriate choice for this extend. The database was utilized to store and oversee information related to flights, clients, bookings, and ticketing.

### 2.3 NetBeans IDE for Computer program Improvement

NetBeans is an coordinates advancement environment (IDE) for Java that bolsters quick application advancement through its built-in GUI builder and investigating devices. It permits developers to oversee complex ventures productively and gives consistent integration with databases, making it perfect for building Java Swing applications.

### 2.4 Related Works

Previous scholarly ventures and real-world aircraft reservation frameworks have illustrated the significance of mechanized frameworks in diminishing human mistake, making strides reaction times, and upgrading client fulfillment. Inquire about has appeared that user-friendly interfacing combined with dependable backend frameworks essentially move forward operational effectiveness in transportation administration frameworks.

### 2.5 Rundown

This audit highlights the hypothetical and specialized establishments upon which the Aircraft Administration Framework was built. The utilize of Java Swing for the interface, MySQL for information taking care of, and NetBeans for advancement guarantees that the framework is vigorous, adaptable, and user-friendly. The writing moreover bolsters the utilize of such frameworks in both scholarly and commercial settings for making strides carrier benefit operations.

# CHAPTER 3

## SYSTEM REQUIREMENTS

### 3.1 Software Requirement Specifications

- Operating System Front End Back End Server Documentation: Windows 10
- **Frontend Software:** Java NetBeans 8.2: JDK 8
- **Backend Software:** MySQL

### 3.2Hardware Requirement Specifications

- Computer Processor Core i5 Processor Speed 2.3 GHz Processor Hard Disk 400 GB or more RAM Min 2GB

# CHAPTER 4

## SYSTEM DESIGN

### 4.1 Introduction to EDR

An Entity Relationship Diagram (ERD) is a conceptual blueprint that illustrates the logical structure of a database. It shows the key entities involved in a system and the relationships between them. In the context of an Airline Management System (AMS), the ERD plays a crucial role in modeling the data components required to support core operations such as flight scheduling, booking, passenger management, crew assignment, and aircraft maintenance. It shapes the establishment for database plan and makes a difference guarantee information judgment, consistency, and productive recovery.

Framework investigation and plan are basic stages within the program advancement life cycle, which guarantee that the ultimate program item fulfills the expecting necessities, is solid, viable, and versatile. In this extend, the JDBC Course Enrollment Framework is planned to streamline and robotize the scholastic course enlistment prepare, decreasing manual blunders and moving forward client encounter for both understudies and regulatory staff. This chapter dives into a careful examination of the framework necessities, achievability considers, point by point plan models, and engineering contemplations. Each viewpoint is created to form a strong establishment for the effective execution and arrangement of the framework.

### 4.2 Requirement Analysis

The method of prerequisite examination started with point by point discourses and overviews including key partners:clients who will utilize the stage to book lodging, property proprietors who list their rentals, and chairmen dependable for managing stage operations. The accumulated prerequisites are categorized into utilitarian and nonfunctional sorts.

Utilitarian prerequisites portray what the framework must do. The framework ought to permit clients to enlist and log in safely. Clients must be able to browse accessible postings together with subtle elements such as property title, area, price, availability dates, and comforts. The framework ought to empower clients to create bookings for their chosen housing and see booking affirmations and history. It must too give clients with the capacity to cancel bookings inside a characterized time period. From the property owner's viewpoint, the framework ought to permit for the expansion, adjustment, or evacuation of property postings. Directors must be able to screen client action, oversee posting information, and handle client or proprietor questions as required.

**Non-functional** necessities characterize how the framework performs beneath particular conditions. The application must be responsive and convey fast stack times indeed amid crest hours. It ought to keep up tall accessibility and guarantee secure dealing with of client information. The client interface must be natural, permitting indeed non-technical clients to explore effectively. Furthermore, the framework ought to be versatile to back future development and able of joining with third-party APIs for highlights like installment preparing and notices.

### 4.3 Feasibility Study

A comprehensive achievability consider was conducted to assess the project's common sense from numerous viewpoints.

Specialized achievability surveyed the appropriateness of utilizing Java, JDBC, and a social database such as MySQL or Prophet. These advances are develop, well-supported, and broadly utilized in instructive educate. The improvement group has satisfactory mastery in these instruments, guaranteeing smooth execution. The venture engineering bolsters secluded plan and future adaptability, which encourage includes to its specialized viability. A feasibility consider could be a basic step within the arranging and advancement of any program extend. It makes a difference decide whether the proposed framework is down to earth, cost-effective, and achievable inside the accessible assets. For an Carrier Administration Framework (AMS), which includes overseeing complex operations such as flight planning, traveler bookings, ticketing, installment preparing, team assignments, and air ship upkeep, a comprehensive achievability consider is fundamental. This think about analyzes the framework from different points of view counting specialized, financial, operational, lawful, and plan possibility to survey the project's in general reasonability.

### 4.4 Technical Feasibility

Specialized possibility surveys whether the desired innovation and foundation are accessible to create and actualize the Carrier Administration Framework. Given the development of current advances such as social databases (e.g., MySQL, PostgreSQL), web-based client interfacing, and cloud framework, creating an AMS is actually reasonable. The framework can be built utilizing existing systems and devices, and it can be facilitated on cloud stages like AWS or Purplish blue to guarantee adaptability and unwavering quality. Besides, framework integration with third-party administrations such as installment portals, SMS/email notice frameworks, and outside APIs (e.g., for climate or air terminal information) is in fact achievable. Hence, from a specialized angle, there are no major imperatives that seem anticipate the advancement and arrangement of the framework.

### 4.5 Economic Feasibility

Financial possibility, or cost-benefit examination, assesses whether the monetary benefits of the framework legitimize the speculation. In spite of the fact that beginning advancement and setup costs may be high—especially for foundation, talented faculty, and licensing—the long-term benefits exceed these costs. By robotizing schedule operations such as flight bookings, installment preparing, and staff planning, the framework essentially diminishes operational costs and manual mistakes. Also, the framework can produce expanded income through way better asset utilization, real-time information analytics, and improved client involvement. By and large, the return on speculation (ROI) is anticipated to be positive, making the extend financially doable.

### 4.6 Operational Feasibility

Operational possibility looks at whether the framework will work successfully within the planning environment and whether clients will accept and receive it. Within the case of the AMS, the framework adjusts well with the day-to-day operations of carrier staff and administration. With a user-friendly interface and computerized highlights, it streamlines complex assignments and underpins speedier decision-making. Workers can effortlessly adjust to the modern framework with negligible preparing, and travelers will advantage from smoother booking and check-in forms. Moreover, the framework improves client fulfillment by lessening delays and giving exact flight and booking data.

### 4.7 Legal Feasibility

Lawful achievability includes guaranteeing that the framework complies with all pertinent laws, controls, and information assurance approaches. An AMS must handle delicate traveler information such as names, contact data, and installment points of interest, which makes compliance with information protection laws just like the Common Information Assurance Control (GDPR) or neighborhood cybersecurity acts basic. The framework ought to actualize solid encryption, secure verification, and information get to control to protect data. As long as the framework is outlined with lawful compliance in intellect and proper measures are actualized, it'll not confront legitimate or administrative obstructions, affirming its legitimate possibility.

### 4.8 Schedule Feasibility

Schedule feasibility determines whether the system can be developed within the available time frame. Considering the complexity of the Airline Management System, the project may require several months of development, including requirement gathering, system design, development, testing, and deployment. However, with a skilled team and a welldefined project plan, the system can be delivered on time using agile methodologies that prioritize key features and allow iterative improvements. Delays can be minimized through

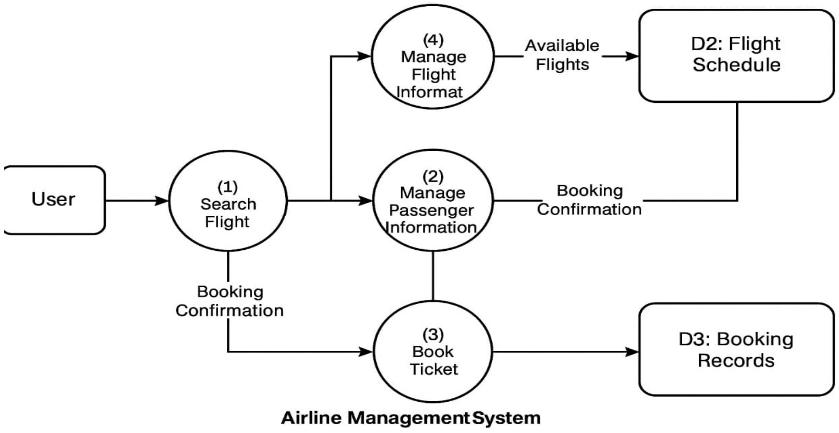


Fig 3.1 Data Flow Diagram

## 4.9 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

### 5.1 Backend Implementation

**MYSQL**

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

**Table cancellation:**

```
create table cancellation(pnr_no varchar(10), cancellation_no varchar(10), cancellation_date
DATE, fli_code varchar(15));
```

**Table flight:**

```
create table flight(f_code varchar(10), f_name varchar(20), src varchar(30), dst varchar(30));
```

**Table login:**

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

**Table passenger:**

```
create table passenger(pnr_no varchar(10), address varchar(30), nationality varchar(15),
name varchar(20), gender varchar(10), ph_no varchar(15), passport_no varchar(20), fl_code
varchar(10));
```

**Table payment:**

```
create table payment(pnr_no varchar(10), ph_no varchar(15), cheque_no varchar(15),
card_no varchar(20), paid_amt varchar(10), pay_date DATE);
```

**Table reservation:**

```
create table reservation(pnr_no varchar(10), ticket_id varchar(10), f_code varchar(10),
jny_date DATE, jny_time varchar(10), src varchar(20), dst varchar(20));
```

**Table sector:**

```
create table sector(flight_code varchar(20), capacity varchar(10), class_code varchar(5),
    class_name varchar(20));
```

### 5.2 Frontend Implementation

### 5.2.1 Java Core

Core Java is the part of Java programming language that is used for creating or developing a general-purpose application. It uses only one tier architecture that is why it is called as ‘stand alone’ application. Core java programming covers the swings, socket, awt, thread concept, collection object and classess.

### 5.2.2 Swings

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

### 5.3 Creating mainframe class

```
package arilinemanagmentsystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Home extends JFrame implements ActionListener{
public Home(){
setLayout(null);
ImageIconi1=New
ImageIcon(ClassLoader.getSystemResource("arilinemanagmentsystem/Icons/front.jpg"));
JLabel image = new JLabel(i1);
image.setBounds(0,0,1600,800);
add(image);

JLabel heading = new JLabel("WELCOMES TO AIR INDIA ARILINES");
heading.setBounds(400,60,1000,60);
heading.setForeground(Color.RED);
heading.setFont(new Font("Tahoma",Font.CENTER_BASELINE,40));
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 ticketcancellationDetails = new JMenuItem("Cancel Ticket");
          ticketcancellationDetails.addActionListener(this);
           details.add(ticketcancellationDetails);

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

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



       setExtendedState(JFrame.MAXIMIZED_BOTH);
       setLocation(600,250);
       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();
}
}
```

# CHAPTER 6

## TEST CASES

### 6.1 Test Case 1: User Login

- **Description:** Verify user login with valid credentials
- **Steps:**
    1. Navigate to the login page
    2. Enter valid username and password
    3. Click "Login"
- **Expected Result:** User is redirected to the dashboard

### 6.2 Test Case 2: Flight Search Functionality

- **Description:** Verify flight search with valid input
- **Steps:**
    1. Select source and destination
    2. Select travel date
    3. Click "Search Flights"
- **Expected Result:** List of available flights is displayed

### 6.3 Test Case 3: Booking a Flight

- **Description:** Verify booking flow with valid passenger details
- **Steps:**
    1. Select a flight from search results
    2. Enter passenger details
    3. Confirm booking
- **Expected Result:** Booking is successful and ticket is generated

### 6.4 Test Case 4: Add Passenger Validation

- **Description:** Ensure passenger form validates missing data

- **Steps:**
    1. Leave required fields empty
    2. Click "Submit"
- **Expected Result:** Error message is shown for missing fields

### 6.5 Test Case 5: Data Storage in Database

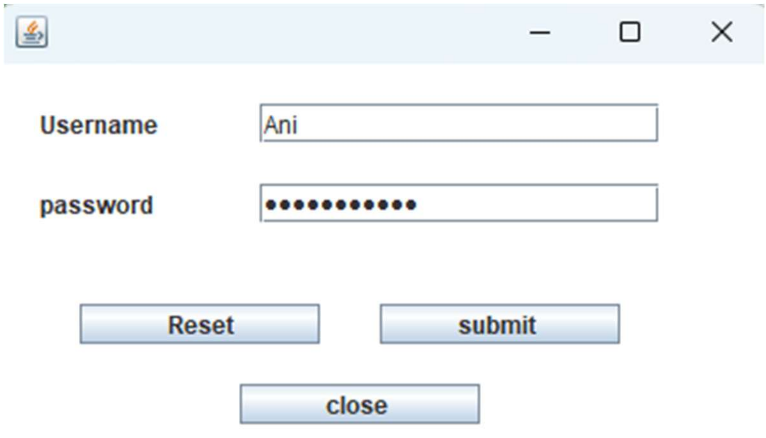
- **Description:** Ensure booking details are stored in MySQL database
- **Steps:**
    1. Complete a ticket booking
    2. Check the database for new entry
- **Expected Result:** Passenger and booking info is saved correctly

### 6.6 Test Case 6: View All Passengers List

- **Description:** Verify admin can view all passenger records
- **Steps:**
    1. Log in as admin
    2. Navigate to "Passenger List"
- **Expected Result:** Full list of passengers is displayed

# CHAPTER 7

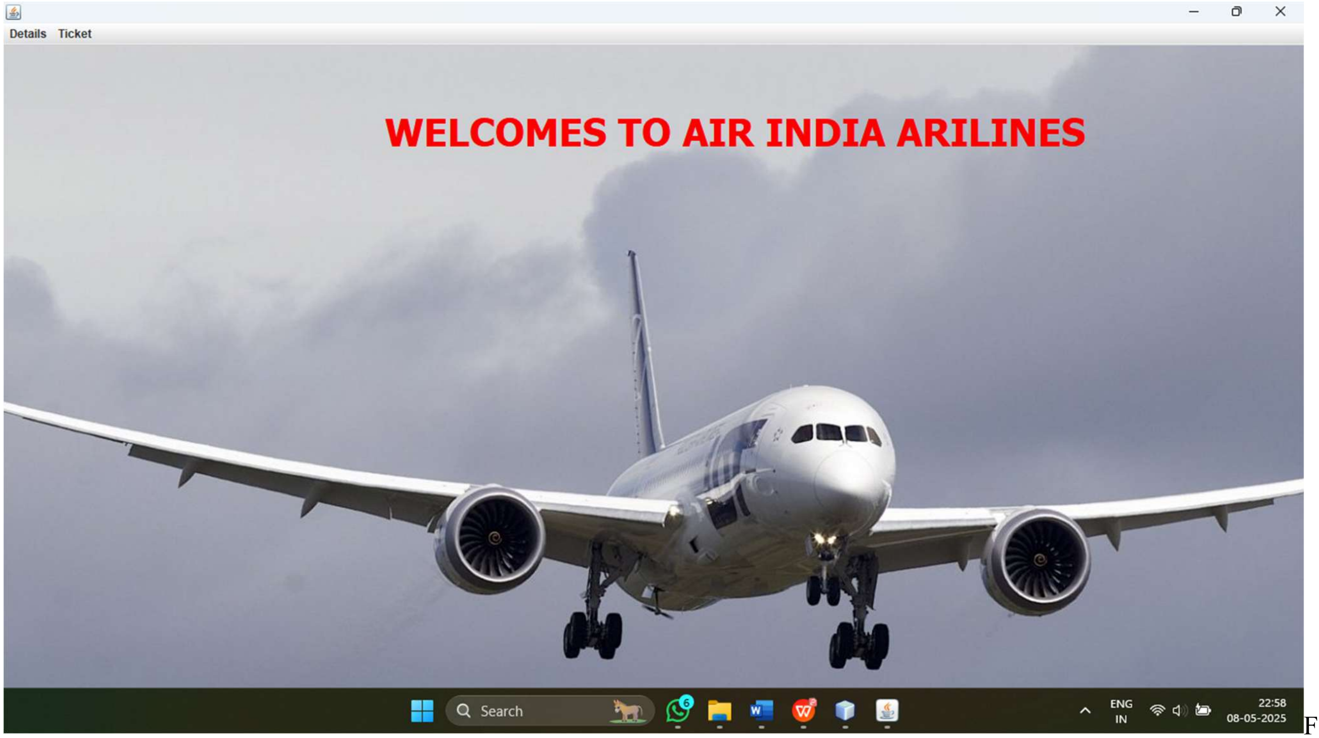
## SCREENSHOTS



Username Ani

password ●●●●●●●●●●●●

Fig 6.1: Login Operation



WELCOMES TO AIR INDIA ARILINES

Fig

Fig 6.2: Mainframe

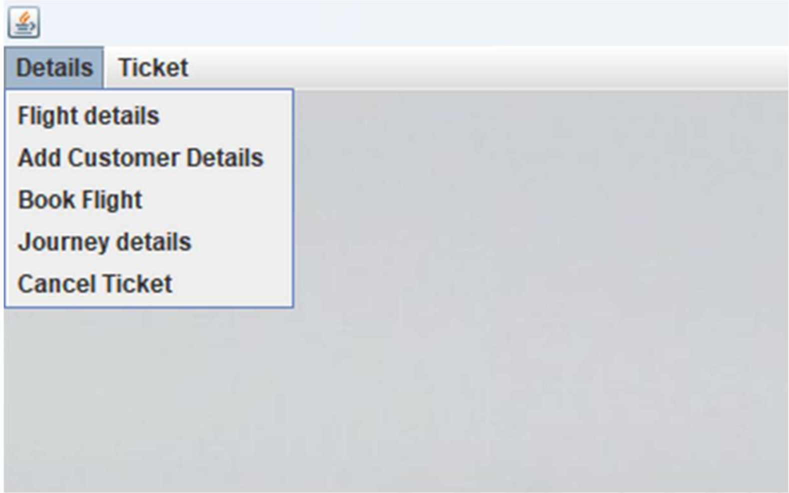
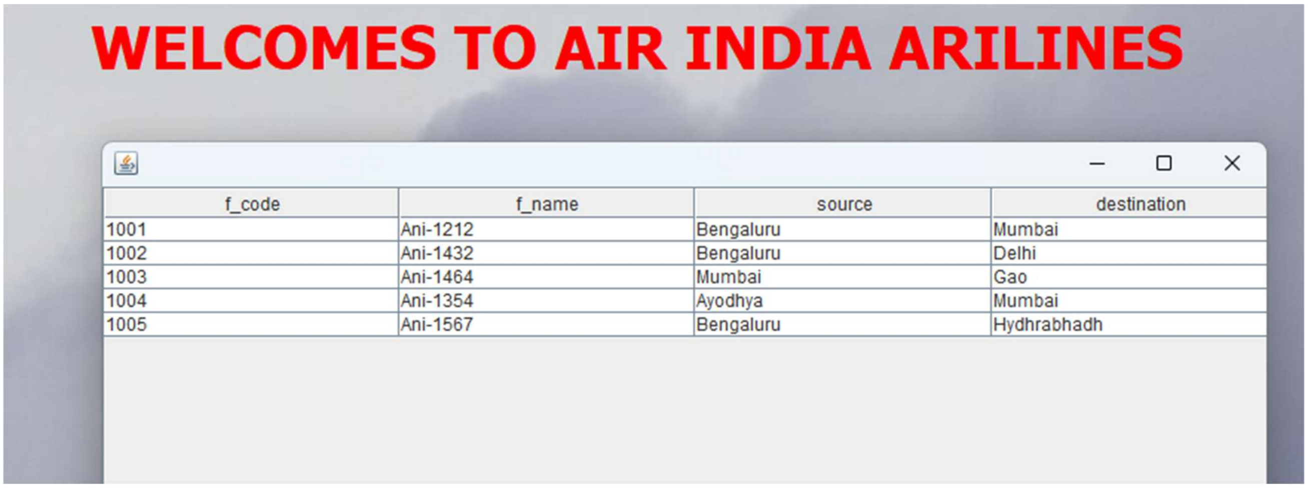


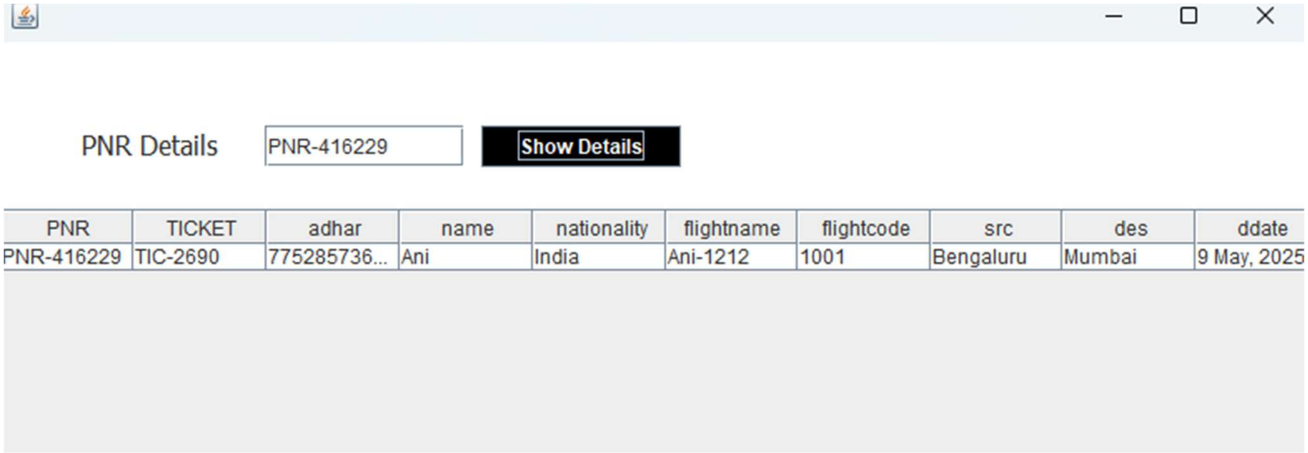
Fig 6.3: Drop-down Menu



WELCOMES TO AIR INDIA ARILINES

| f_code | f_name | source | destination |
|---|---|---|---|
| 1001 | Ani-1212 | Bengaluru | Mumbai |
| 1002 | Ani-1432 | Bengaluru | Delhi |
| 1003 | Ani-1464 | Mumbai | Gao |
| 1004 | Ani-1354 | Ayodhya | Mumbai |
| 1005 | Ani-1567 | Bengaluru | Hydhrabhadh |

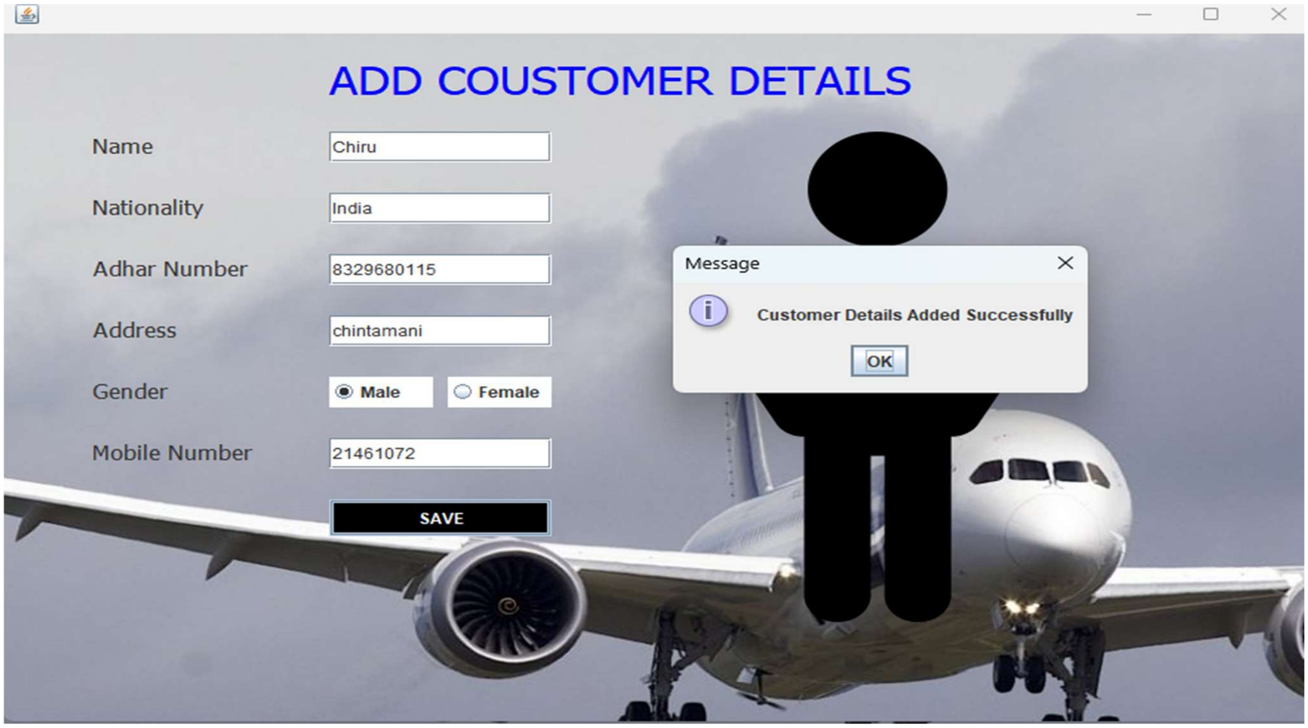
Fig 6.4: Flight Information



PNR Details PNR-416229

| PNR | TICKET | adhar | name | nationality | flightname | flightcode | src | des | ddate |
|---|---|---|---|---|---|---|---|---|---|
| PNR-416229 | TIC-2690 | 775285736... | Ani | India | Ani-1212 | 1001 | Bengaluru | Mumbai | 9 May, 2025 |

Fig 6.5:PNR Details



ADD COUSTOMER DETAILS

Name: Chiru

Nationality: India

Adhar Number: 8329680115

Address: chintamani

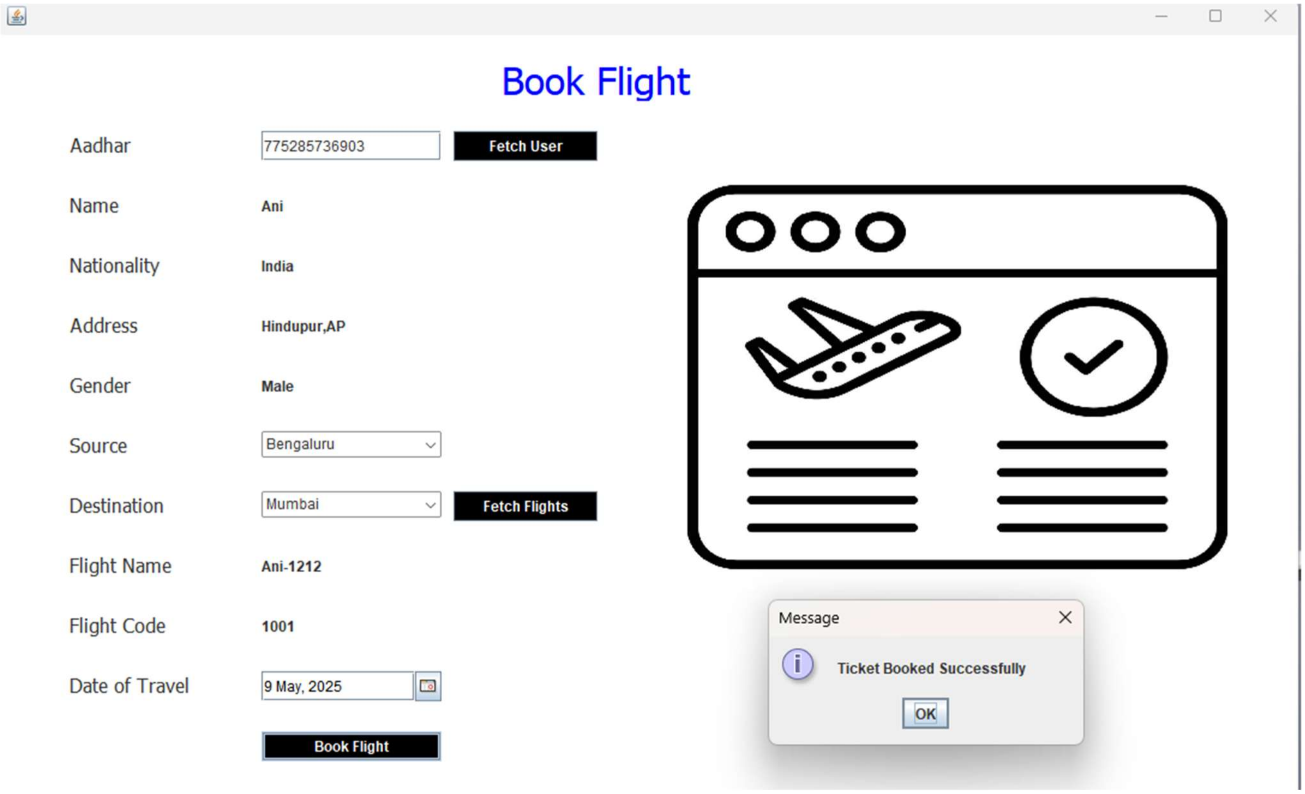
Gender: ● Male ○ Female

Mobile Number: 21461072

Message

Customer Details Added Successfully

Fig 6.6: Adding Customer



Book Flight

Aadhar: 775285736903

Name: Ani

Nationality: India

Address: Hindupur,AP

Gender: Male

Source: Bengaluru

Destination: Mumbai

Flight Name: Ani-1212

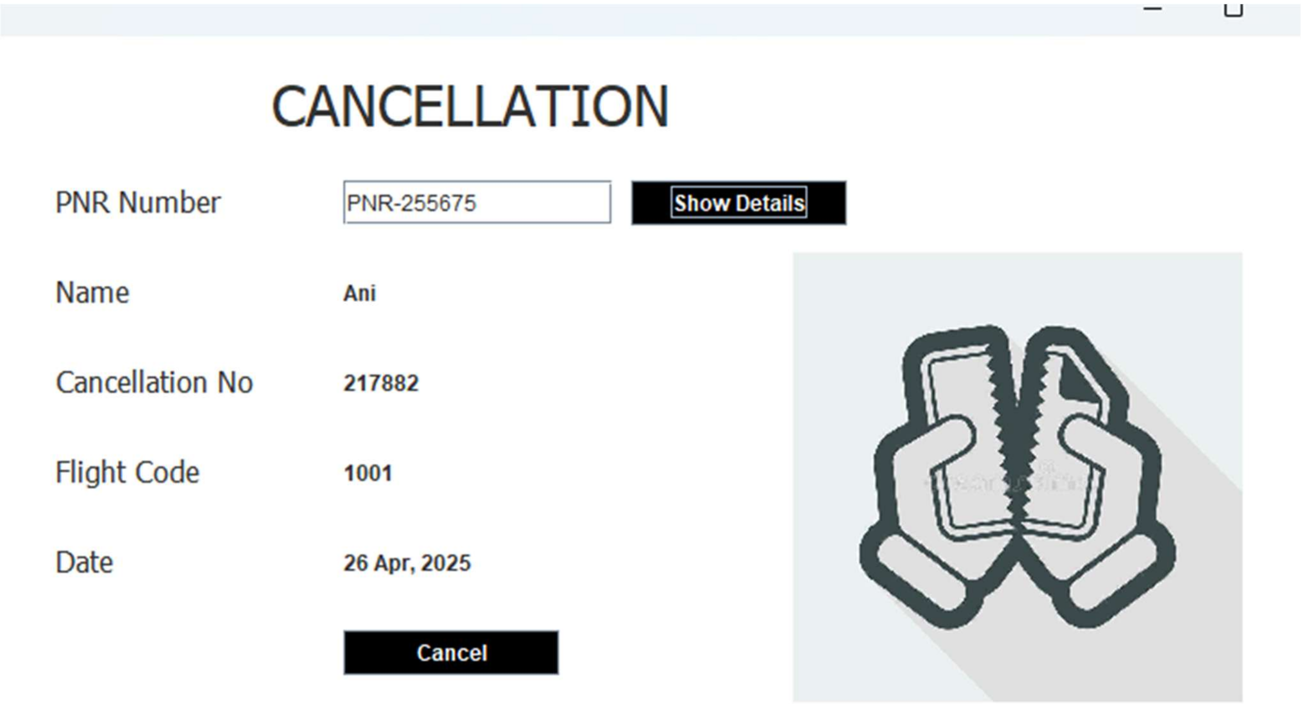
Flight Code: 1001

Date of Travel: 9 May, 2025

Message

Ticket Booked Successfully

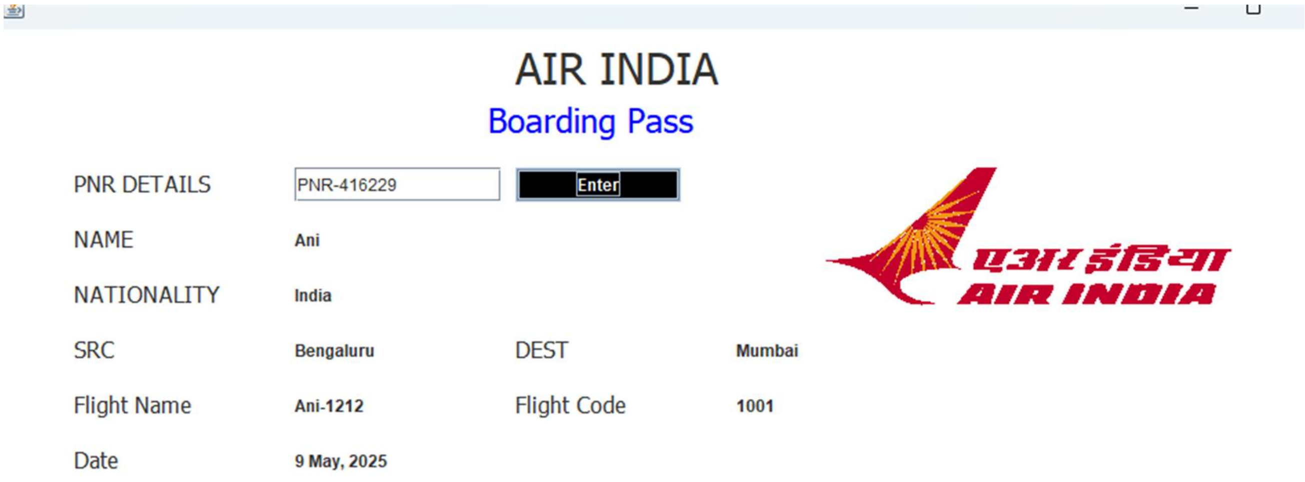
Fig 6.7: Journey Details



CANCELLATION

| | |
|---|---|
| PNR Number | PNR-255675 |
| Name | Ani |
| Cancellation No | 217882 |
| Flight Code | 1001 |
| Date | 26 Apr, 2025 |

Fig 6.8: Cancellation

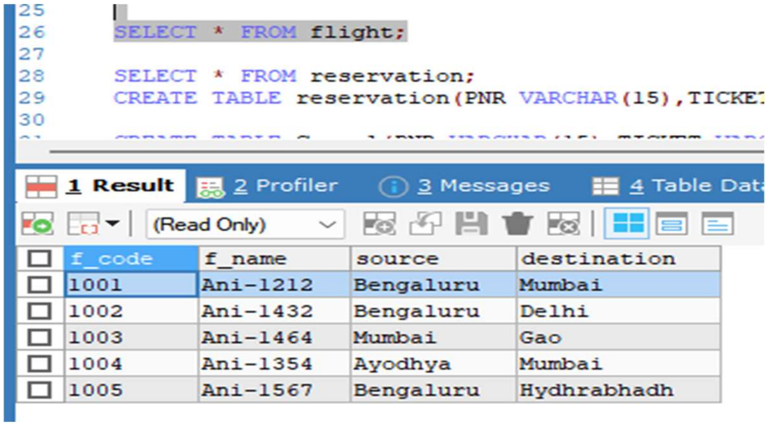


AIR INDIA

Boarding Pass

| | | | |
|---|---|---|---|
| PNR DETAILS | PNR-416229 | | |
| NAME | Ani | | |
| NATIONALITY | India | | |
| SRC | Bengaluru | DEST | Mumbai |
| Flight Name | Ani-1212 | Flight Code | 1001 |
| Date | 9 May, 2025 | | |

Fig 6.9: Boarding Pass

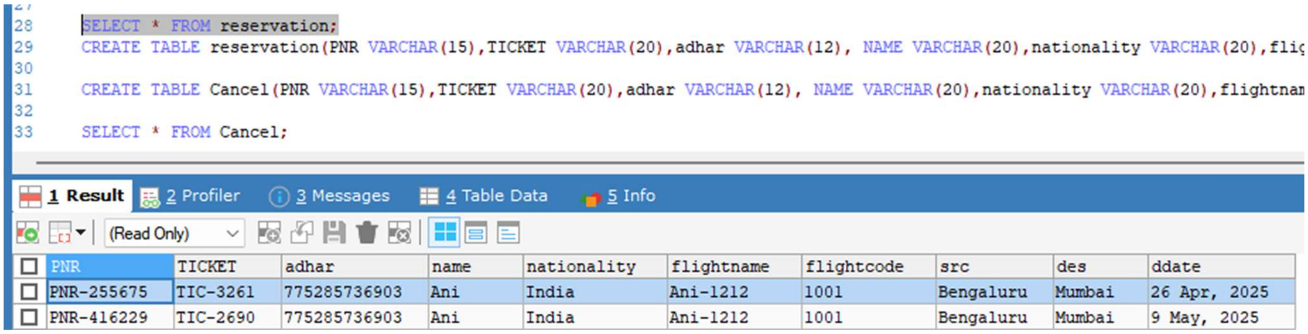


```
SELECT * FROM flight;

SELECT * FROM reservation;
CREATE TABLE reservation(PNR VARCHAR(15),TICKE
```

| □ | f_code | f_name | source | destination |
|---|---|---|---|---|
| □ | 1001 | Ani-1212 | Bengaluru | Mumbai |
| □ | 1002 | Ani-1432 | Bengaluru | Delhi |
| □ | 1003 | Ani-1464 | Mumbai | Gao |
| □ | 1004 | Ani-1354 | Ayodhya | Mumbai |
| □ | 1005 | Ani-1567 | Bengaluru | Hydhrabhadh |

Fig 6.10: Backend flight Records



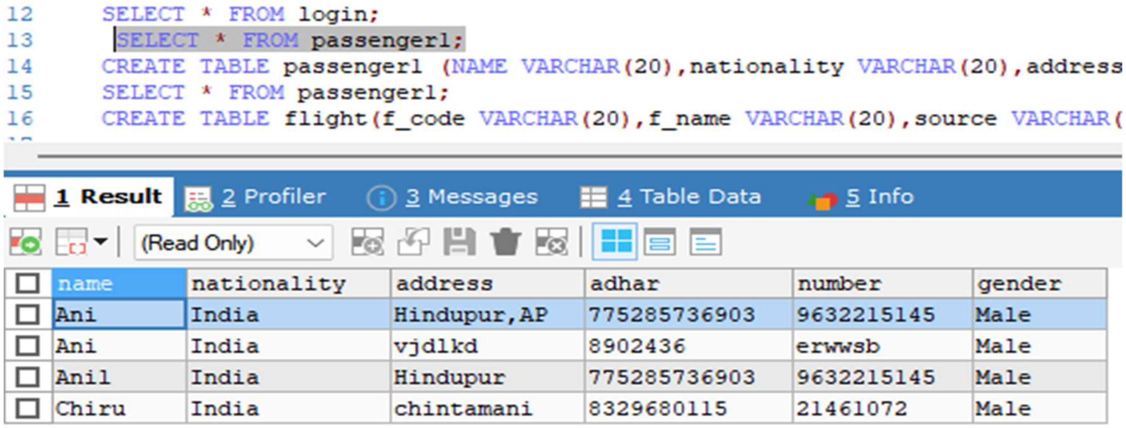
```
SELECT * FROM reservation;
CREATE TABLE reservation(PNR VARCHAR(15),TICKET VARCHAR(20),adhar VARCHAR(12), NAME VARCHAR(20),nationality VARCHAR(20),flig

CREATE TABLE Cancel(PNR VARCHAR(15),TICKET VARCHAR(20),adhar VARCHAR(12), NAME VARCHAR(20),nationality VARCHAR(20),flightnam

SELECT * FROM Cancel;
```

| □ | PNR | TICKET | adhar | name | nationality | flightname | flightcode | src | des | ddate |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | PNR-255675 | TIC-3261 | 775285736903 | Ani | India | Ani-1212 | 1001 | Bengaluru | Mumbai | 26 Apr, 2025 |
| □ | PNR-416229 | TIC-2690 | 775285736903 | Ani | India | Ani-1212 | 1001 | Bengaluru | Mumbai | 9 May, 2025 |

Fig 6.11: Backend Reservation Records



```
SELECT * FROM login;
 SELECT * FROM passengerl;
CREATE TABLE passengerl (NAME VARCHAR(20),nationality VARCHAR(20),address
SELECT * FROM passengerl;
CREATE TABLE flight(f_code VARCHAR(20),f_name VARCHAR(20),source VARCHAR(
```

| □ | name | nationality | address | adhar | number | gender |
|---|---|---|---|---|---|---|
| □ | Ani | India | Hindupur,AP | 775285736903 | 9632215145 | Male |
| □ | Ani | India | vjdlkd | 8902436 | erwwsb | Male |
| □ | Anil | India | Hindupur | 775285736903 | 9632215145 | Male |
| □ | Chiru | India | chintamani | 8329680115 | 21461072 | Male |

Fig 6.12: Backend Passenger Records

# CHAPTER 8

## CONCLUSION

The Airline management System may be a comprehensive course of action laid out to streamline and mechanize the complex operations of the flying industry. By coordination essential capacities such as flight arranging, ticket booking, traveler and gather organization, installment taking care of, and flying machine upkeep, the system basically overhauls operational efficiency and advantage movement. Through the utilize of cutting edge propels, checking social databases, cloud computing, and user-friendly meddle, the system ensures accuracy, speed, and security in supervising broad volumes of data over numerous divisions.

This system not because it were lessens manual workload and human botch but as well makes strides the by and large traveler association by engaging faster bookings, real-time flight overhauls, and straightforward get to to travel information. For carrier staff and chiefs, it gives a centralized organize for checking flights, supervising resources, and making data-driven choices. The system besides supports compliance with authoritative rules and data confirmation laws, ensuring that fragile traveler and operational data is taken care of securely.

Other than, the Airline management System is adaptable and flexible, allowing aircrafts of all sizes to actualize it concurring to their specific needs and operational scale. With mechanization, straightforwardness, and real-time information at its center, the system locks in aircrafts to optimize their organizations, diminish costs, and update client fulfillment. In conclusion, the Carrier Organization System can be a basic imaginative progress that meets the creating demands of the flying division, empowering secure, profitable, and customer friendly airplane operations inside the progressed period. The Airline management System is an compelling device for streamlining carrier operations such as flight arranging, ticket booking, group assignment, and traveler organization.

# REFERENCES

1.https://developers.openshift.com/database/mysql.html

OpenShift developer guide for setting up and managing MySQL databases, useful for airline system backends.

2. https://youtu.be/UbIIFLsEeiM

YouTube tutorial on building an Airline Reservation System using Python and MySQL.

3. https://aws.amazon.com/solutions/airline/

Amazon Web Services' page on cloud solutions for airlines, covering scalability and security.

4. https://www.oracle.com/database/technologies/

Oracle's database solutions and documentation, useful for enterprise-scale AMS development.

5. https://spring.io/projects/spring-boot

Spring Boot documentation, useful for building backend airline service APIs in Java.

6. https://learn.microsoft.com/en-us/dotnet/architecture/microservices/

Guide to building microservices, applicable for modular design of airline systems.