FLIGHT BOOKING SYSTEM

A MINI-PROJECT REPORT

Submitted by

PRANAV R	240701394
ASHWIN M	240701063

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “FLIGHT BOOKING SYSTEM” is the bonafide work of “PRANAV R,ASHWIN M” who carried out the project work under my supervision.

SIGNATURE	
Dr.J. MANORANJANI	
DEPUTY HEAD OF THE DEPARTMENT	
Dept. of Computer Science and Engg,	
Rajalakshmi Engineering College Chennai	

This mini project report is submitted for the viva voce examination to be held on __________

INTERNAL EXAMINER   EXTERNAL EXAMINER

ABSTRACT

In the modern travel industry, flight booking plays a crucial role in ensuring convenient and efficient air travel for passengers. Even though there are several multinational online platforms that provide flight booking services, many regional travel agencies and customers face difficulties in accessing a simplified and organized system. To address this gap, our team has developed a database-driven application called Flight Booking System to streamline and manage flight reservations effectively.

The main objective of this project is to provide an efficient platform for booking, managing, and tracking flight details based on customer requirements. This system helps maintain comprehensive records of flights, passengers, and ticket availability in an organized manner. By automating these operations, the system reduces manual workload, minimizes errors, and enhances the user experience. The Flight Booking System aims to assist local travel operators and agencies in offering a smooth, reliable, and competitive service in the fast-growing aviation industry.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman MR. S. MEGANATHAN and the chairperson DR. M.THANGAM MEGANATHAN for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal Dr. S.N. MURUGESAN for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department Dr. E.M. MALATHY and our Deputy Head Of The Department Dr. J. MANORANJINI for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide Mrs.Deepa.B, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

PRANAV.R

ASHWIN M

TABLE OF CONTENTS

LIST of figures

CHAPTER 1

INTRODUCTION

INTRODUCTION

The project helps people to know the necessary information and list of bookings and assess the feasibility in the locality. The necessary information about availability of flights and list of bookings will be mentioned according to user convenience.

SCOPE OF THE WORK

The scope of the Flight Booking System includes developing an online platform that allows users to search, book, and manage flight reservations efficiently. It enables customers to view available flights, compare fares, make secure payments, and receive electronic tickets instantly. The system also allows administrators to manage flight schedules, seat availability, pricing, and booking records. By automating the booking process, the system reduces manual effort, minimizes errors, and ensures real-time access to flight information while maintaining data security and transaction integrity.

PROBLEM STATEMENT

The need for this project arises from the fact that, although many airlines and travel companies have expanded globally, flight booking services still remain difficult to access for people in several local and remote areas. Many travelers face challenges such as limited access to reliable booking platforms, lack of real-time flight information, and inefficient manual reservation processes. This often leads to inconvenience, travel delays, and increased costs. Therefore, there is a need for a user-friendly, efficient, and accessible Flight Booking System that enables users to easily search, book, and manage flights from anywhere, ensuring convenience and transparency in the booking process.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main objective of the project is to provide a user-friendly platform for customers to check flight availability, schedules, and fares in real time and maintain and manage flight details, passenger records, and booking information efficiently. Also to automate the booking and ticket generation process to reduce manual errors and delays and to enable secure online payment and confirmation of bookings.To help airline agencies improve their service quality and remain competitive by offering fast, reliable, and accessible booking facilities.

.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
HDD	:	1TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front –End	:	JAVA
Back - End	:	MySql
Language	:	java,SQL

CHAPTER 3

MODULE DESCRIPTION

This application consists of two main modules. When the program runs, it will ask for confirmation at the login window. The person interacting with the system can log in either as an Administrator or as a User. The description of the modules is as follows:

Admin Login
When the person interacting with the system tries to log in as an Admin, they must enter a valid username and password. The administrator has the authority to manage and update flight details, schedules, fares, and booking records. They can also view user information, monitor system activity, and ensure data accuracy in the database.

User Login
When the person tries to log in as a User, they are prompted to enter their credentials to access the booking interface. The user can search for available flights based on origin, destination, and date, view flight details, make bookings, process payments, and receive e-tickets. Users can also manage their reservations by viewing, cancelling, or modifying their bookings as needed.

CHAPTER 4

SAMPLE CODING

```
CREATE DATABASE flightdb;USE  flightdb;

CREATE TABLE flights (

    flight_no VARCHAR(10) PRIMARY KEY,

    source VARCHAR(30),

    destination VARCHAR(30),

    seats INT,

    price DOUBLE);

import java.sql.*;

import java.util.*;


public class FlightBookingSystem {
```

```
static final String URL = "jdbc:mysql://localhost:3306/flightdb";
static final String USER = "root";          // your MySQL username
static final String PASSWORD = "password";  // your MySQL password

static Scanner sc = new Scanner(System.in);

public static void main(String[] args) {
    int choice;
    do {
        System.out.println("\n==== FLIGHT BOOKING SYSTEM ====");
        System.out.println("1. Admin Login");
        System.out.println("2. User Login");
        System.out.println("3. Exit");
        System.out.print("Enter your choice: ");
        choice = sc.nextInt();
    sc.nextLine();

    switch (choice) {
        case 1:
            adminLogin();
            break;
        case 2:
            userMenu();
            break;
        case 3:
            System.out.println("Exiting... Thank you!");
            break;
        default:
            System.out.println("Invalid choice!");
    }
} while (choice != 3);
```

```
}


// ---------- ADMIN MODULE ----------
static void adminLogin() {
    System.out.print("Enter Admin Username: ");
    String username = sc.nextLine();
    System.out.print("Enter Password: ");
    String password = sc.nextLine();

    if (username.equals("admin") && password.equals("1234")) {
        adminMenu();
    } else {
        System.out.println("Invalid credentials!");
    }
}
```

```java
static void adminMenu() {
    int choice;
    do {
        System.out.println("\n--- ADMIN MENU ---");
        System.out.println("1. Add Flight");
        System.out.println("2. View Flights");
        System.out.println("3. Delete Flight");
        System.out.println("4. Logout");
        System.out.print("Enter choice: ");
        choice = sc.nextInt();
        sc.nextLine();

        switch (choice) {
            case 1:
                addFlight();
                break;
            case 2:
                viewFlights();
                break;
            case 3:
                deleteFlight();
                break;
            case 4:
                System.out.println("Admin logged out.");
                break;
            default:
                System.out.println("Invalid option!");
        }
    } while (choice != 4);
}

static void addFlight() {
        try (Connection con = DriverManager.getConnection(URL, USER,
PASSWORD)) {
            System.out.print("Enter Flight Number: ");
            String fn = sc.nextLine();
            System.out.print("Enter Source: ");
            String src = sc.nextLine();
            System.out.print("Enter Destination: ");
            String dest = sc.nextLine();
            System.out.print("Enter Seats: ");
            int seats = sc.nextInt();
            System.out.print("Enter Price: ");
            double price = sc.nextDouble();
            sc.nextLine();

            String query = "INSERT INTO flights VALUES (?, ?, ?, ?, ?)";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setString(1, fn);
            ps.setString(2, src);
            ps.setString(3, dest);
            ps.setInt(4, seats);
            ps.setDouble(5, price);

            ps.executeUpdate();
            System.out.println("Flight added successfully!");

        } catch (SQLIntegrityConstraintViolationException e) {
            System.out.println("Flight already exists!");
        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    static void viewFlights() {
        try (Connection con = DriverManager.getConnection(URL, USER,
PASSWORD);

             Statement st = con.createStatement()) {



            ResultSet rs = st.executeQuery("SELECT * FROM flights");



            System.out.println("\nAvailable Flights:");


System.out.println("---------------------------------------------------------");

            System.out.printf("%-10s %-15s %-15s %-10s %-10s\n",

                    "FlightNo", "Source", "Destination", "Seats", "Price");


System.out.println("---------------------------------------------------------");



            while (rs.next()) {

                System.out.printf("%-10s %-15s %-15s %-10d %-10.2f\n",

                        rs.getString("flight_no"),

                        rs.getString("source"),
                        rs.getString("destination"),
                        rs.getInt("seats"),
                        rs.getDouble("price"));
            }

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    static void deleteFlight() {
        try (Connection con = DriverManager.getConnection(URL, USER,
PASSWORD)) {
            System.out.print("Enter Flight Number to delete: ");
            String fn = sc.nextLine();

            String query = "DELETE FROM flights WHERE flight_no = ?";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setString(1, fn);

            int rows = ps.executeUpdate();
            if (rows > 0)
                System.out.println("Flight deleted successfully!");
            else
                System.out.println("Flight not found!");

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    // ---------- USER MODULE ----------
    static void userMenu() {
```

```
int choice;
do {
    System.out.println("\n--- USER MENU ---");
    System.out.println("1. View Flights");
    System.out.println("2. Book a Flight");
    System.out.println("3. Logout");
    System.out.print("Enter choice: ");
    choice = sc.nextInt();
    sc.nextLine();

    switch (choice) {
        case 1:
            viewFlights();
            break;
        case 2:
            bookFlight();
```

```
                    break;
                case 3:
                    System.out.println("User logged out.");
                    break;
                default:
                    System.out.println("Invalid option!");
            }
        } while (choice != 3);
    }

    static void bookFlight() {
        try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
            System.out.print("Enter Flight Number to book: ");
            String fn = sc.nextLine();

            String checkQuery = "SELECT * FROM flights WHERE flight_no = ?";
            PreparedStatement checkPs = con.prepareStatement(checkQuery);

            checkPs.setString(1, fn);

            ResultSet rs = checkPs.executeQuery();


            if (rs.next()) {

                int seats = rs.getInt("seats");

                double price = rs.getDouble("price");


                if (seats > 0) {

                    String updateQuery = "UPDATE flights SET seats = seats - 1
WHERE flight_no = ?";

                    PreparedStatement ps = con.prepareStatement(updateQuery);

                    ps.setString(1, fn);

                    ps.executeUpdate();


                    System.out.println("Booking confirmed for Flight " + fn);

                    System.out.println("Ticket Price: ₹" + price);
                } else {
                    System.out.println("No seats available!");
                }
            } else {
                System.out.println("Flight not found!");
            }

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Sample 1

Sample 1 depicts the display code, that gets the data from the database i.e. being stored there and represented on users demand with the layout and measurements i.e. being already specified.

```
CREATE DATABASE flightdb;


USE flightdb;


CREATE TABLE users (
   username VARCHAR(30) PRIMARY KEY,
   password VARCHAR(30)
);
import java.sql.*;
import java.util.*;

public class UserLoginSystem {
   static final String URL = "jdbc:mysql://localhost:3306/flightdb";
   static final String USER = "root";          // your MySQL username
   static final String PASSWORD = "password";   // your MySQL password

   static Scanner sc = new Scanner(System.in);


   public static void main(String[] args) {
       int choice;
       do {
           System.out.println("\n=== USER LOGIN SYSTEM ===");
           System.out.println("1. Register");
           System.out.println("2. Login");
           System.out.println("3. Exit");
           System.out.print("Enter your choice: ");
```

```
choice = sc.nextInt();
sc.nextLine();

switch (choice) {
  case 1:
    registerUser();
    break;
  case 2:
    loginUser();
    break;
  case 3:
    System.out.println("Exiting... Thank you!");
    break;
  default:
    System.out.println("Invalid choice! Try again.");
```

```
        }
      } while (choice != 3);
    }

    // Register new user
    static void registerUser() {
      try (Connection con = DriverManager.getConnection(URL, USER, PASSWORD)) {
        System.out.print("Enter new username: ");
        String uname = sc.nextLine();
        System.out.print("Enter new password: ");
        String pass = sc.nextLine();

        String query = "INSERT INTO users VALUES (“Ashwin”,”Mashwin_06”)";
```

```
            PreparedStatement ps = con.prepareStatement(query);

            ps.setString(1, uname);

            ps.setString(2, pass);


            ps.executeUpdate();

            System.out.println("User registered successfully!");


        } catch (SQLIntegrityConstraintViolationException e) {

            System.out.println("Username already exists! Try another
one.");

        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        }

    }
    // Login existing user

    static void loginUser() {

        try (Connection con = DriverManager.getConnection(URL,
USER, PASSWORD)) {

            System.out.print("Enter username: ");

            String uname = sc.nextLine();

            System.out.print("Enter password: ");

            String pass = sc.nextLine();



            String query = "SELECT * FROM users WHERE username =
“Pranav” AND password = “Rpranav_06";

            PreparedStatement ps = con.prepareStatement(query);

            ps.setString(1, uname);

            ps.setString(2, pass);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                System.out.println("Login successful! Welcome, " + uname
+ "!");
                // You can redirect the user to the flight booking menu here
            } else {
                System.out.println("Invalid username or password!");
            }

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Sample 2

Sample 2 depicts the booking part of the code, where it displays booking details and enter user data and store it in database

CHAPTER 5

SCREEN SHOTS

Fig 5.1 Introduction page

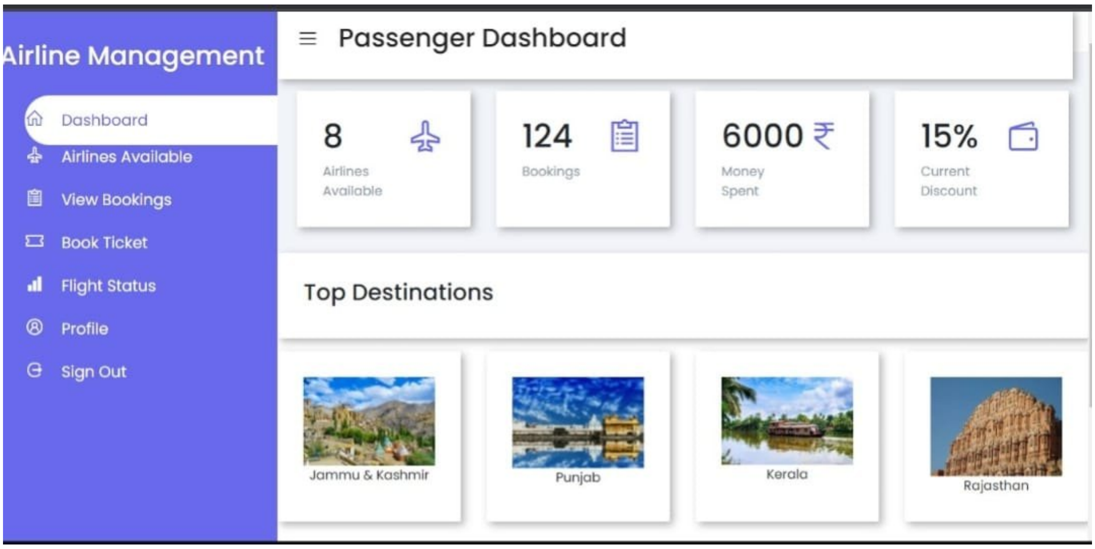
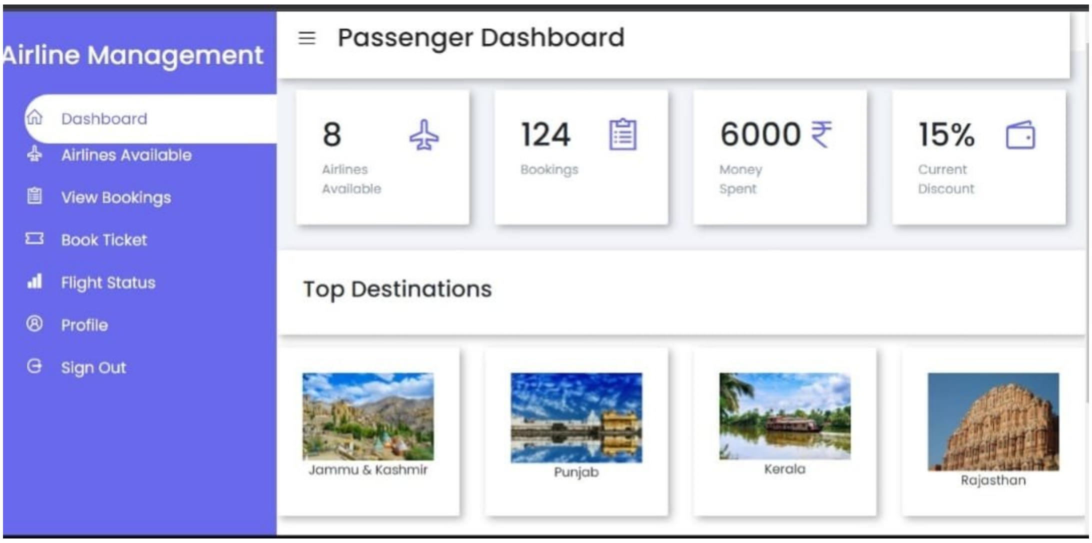


Fig 5.2 User details



8 Airlines Available

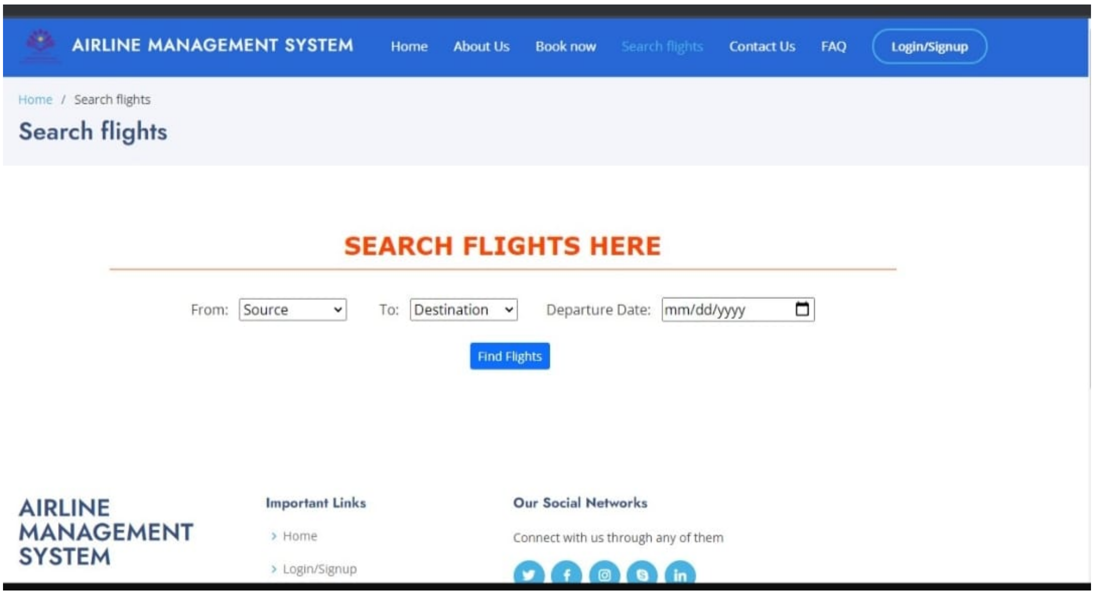
124 Bookings

6000 ₹ Money Spent

15% Current Discount

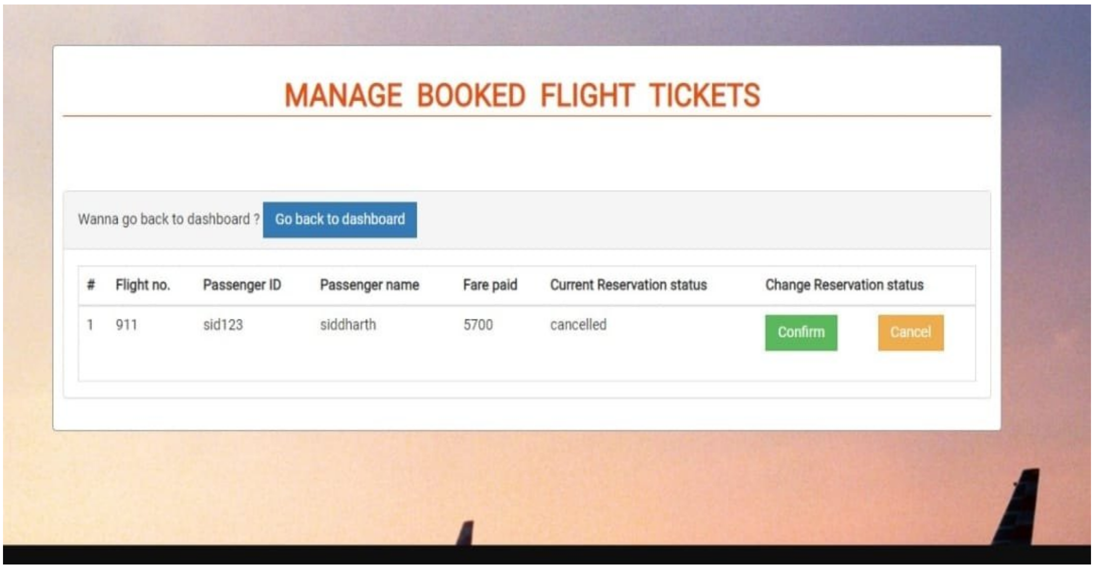
Top Destinations

Fig 5.3 Flight Searching log



From: Source To: Destination Departure Date: mm/dd/yyyy

Fig 5.4 Booking creation

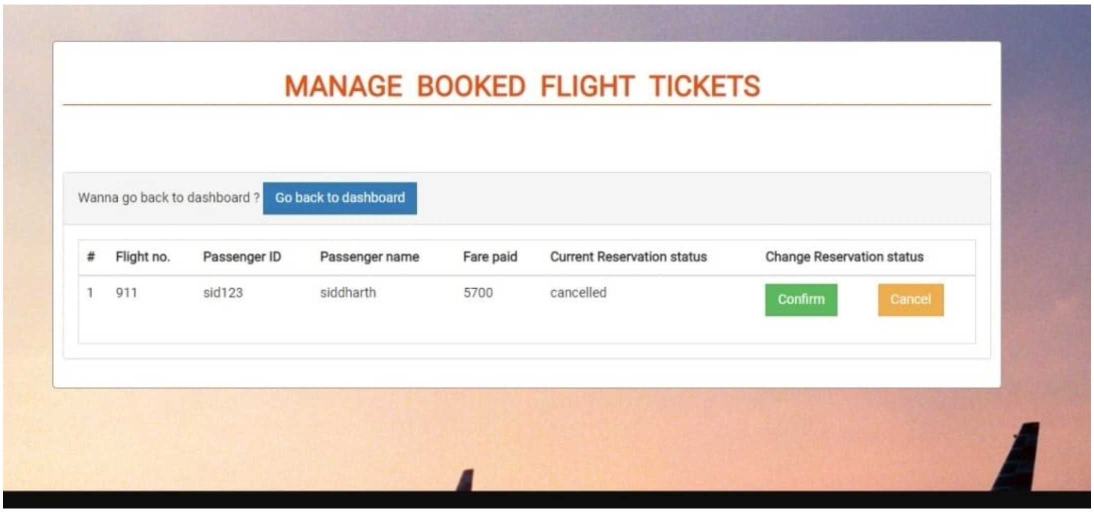


MANAGE BOOKED FLIGHT TICKETS

Wanna go back to dashboard ?

#	Flight no.	Passenger ID	Passenger name	Fare paid	Current Reservation status	Change Reservation status
1	911	sid123	siddharth	5700	cancelled	Confirm Cancel

Fig 5.5 Deletion of Booking

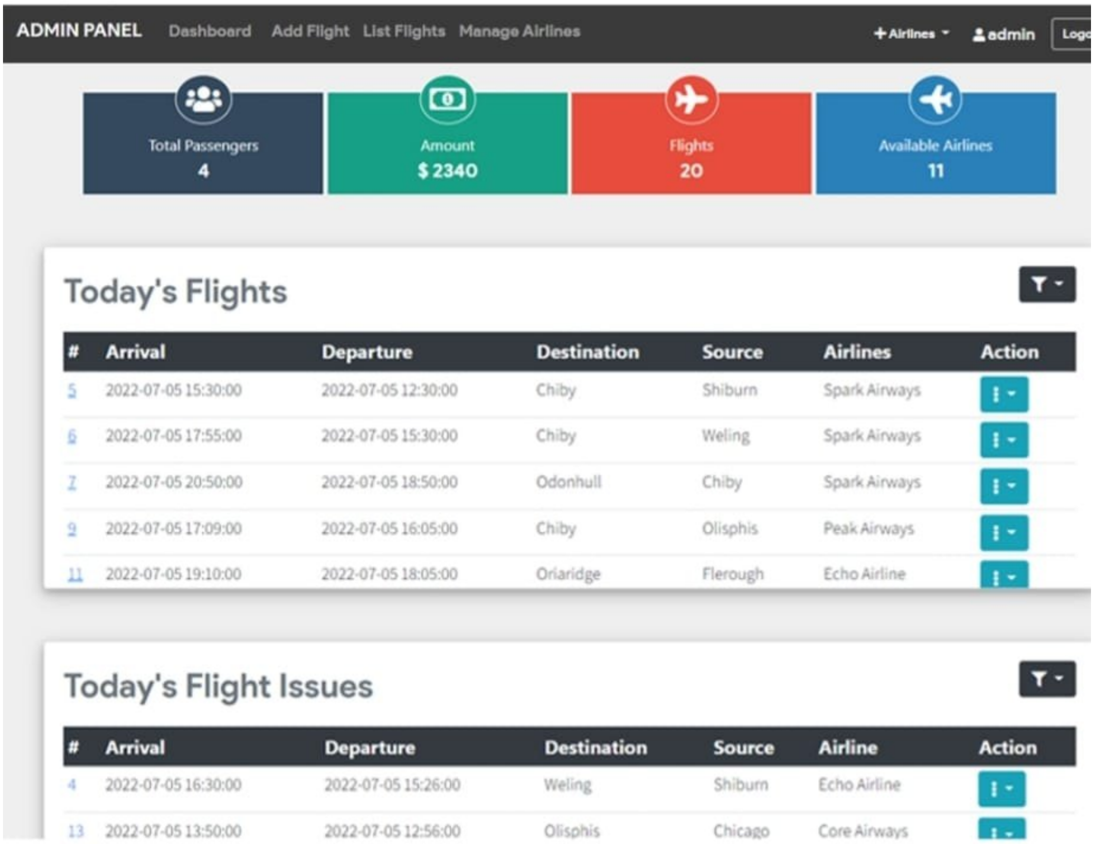


MANAGE BOOKED FLIGHT TICKETS

Wanna go back to dashboard ?

#	Flight no.	Passenger ID	Passenger name	Fare paid	Current Reservation status	Change Reservation status
1	911	sid123	siddharth	5700	cancelled	Confirm Cancel

Fig 5.6 Database creation



Total Passengers
4

Amount
$ 2340

Flights
20

Available Airlines
11

Today's Flights

#	Arrival	Departure	Destination	Source	Airlines	Action
5	2022-07-05 15:30:00	2022-07-05 12:30:00	Chiby	Shiburn	Spark Airways	⋮ ▾
6	2022-07-05 17:55:00	2022-07-05 15:30:00	Chiby	Weling	Spark Airways	⋮ ▾
7	2022-07-05 20:50:00	2022-07-05 18:50:00	Odonhull	Chiby	Spark Airways	⋮ ▾
9	2022-07-05 17:09:00	2022-07-05 16:05:00	Chiby	Olisphis	Peak Airways	⋮ ▾
11	2022-07-05 19:10:00	2022-07-05 18:05:00	Oriaridge	Flerough	Echo Airline	⋮ ▾

Today's Flight Issues

#	Arrival	Departure	Destination	Source	Airline	Action
4	2022-07-05 16:30:00	2022-07-05 15:26:00	Weling	Shiburn	Echo Airline	⋮ ▾
13	2022-07-05 13:50:00	2022-07-05 12:56:00	Olisphis	Chicago	Core Airways	⋮ ▾

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

In such a way, with the help of our project, customers will be able to check the list of bookings and can register themselves to avail a cab. The booking system clearly represents the available data of the customers for booking and the number of bookings using a booking log and management becomes easier. In future people will be able to book cabs according to the data available in the system and with respect to the availability. Hence this project makes the user and other advantages to be benefitted in all possible ways.

REFERENCES

Oracle MySQL Documentation – MySQL Database Reference Manual
https://dev.mysql.com/doc/

Java Tutorials by Oracle – Official Java Programming Guide
https://docs.oracle.com/javase/tutorial/

W3Schools – JDBC and Database Connectivity in Java
https://www.w3schools.com/java/java_mysql.asp

GeeksforGeeks – Flight Reservation System Project in Java
https://www.geeksforgeeks.org/

JavaTpoint – Java Database Connectivity (JDBC) and CRUD Operations
https://www.javatpoint.com/java-jdbc

TutorialsPoint – Software Engineering Project Design and Implementation
https://www.tutorialspoint.com/software_engineering/index.htm