MINI PROJECT

AIRLINE RESERVATION SYSTEM

Aim:

To construct a database for the Airline Reservation system and connect it with my SQL using java.

Algorithm:

1. Start the System
    - Display a menu with the following options:
        - 1. Book a flight
        - 2. Cancel a flight
        - 3. View available flights
        - 4. Exit the system
2. Book a Flight
    - Input the user’s details (name, contact information).
    - Display a list of available flights (Flight number, Date, Destination, Available seats).
    - User selects a flight.
    - Check if seats are available for the selected flight.
        - If seats are available:
            - Reserve a seat for the user.
            - Store booking details (Passenger name, flight, seat number).
            - Provide a booking confirmation with ticket details.
        - If no seats are available, inform the user and ask to select another flight.
3. Cancel a Flight
    - Input the ticket number or booking reference.
    - Search for the ticket in the system.
        - If the ticket exists:
            - Ask for confirmation to cancel the booking.
            - Cancel the booking and free up the seat.
            - Provide confirmation of cancellation.
        - If the ticket does not exist, display an error message.

4. View Available Flights
    - Display a list of available flights (with details such as date, time, available seats).
    - Optionally, allow the user to filter by destination or date.
5. Exit the System
    - Exit the program with a farewell message.

PROGRAM:

Main Class

```
package com.airline;

import com.airline.ui.Menu;

public class Main {
    public static void main(String[] args) {
        Menu.showMainMenu();
    }
}
```

Flight Class

```
package com.airline.models;

public class Flight {
    private String flightId;
    private String origin;
    private String destination;
    private String departureTime;
    private String arrivalTime;
    private int seatsAvailable;
    private double price;

    // Constructor
    public Flight(String flightId, String origin, String destination,
                  String departureTime, String arrivalTime, int seatsAvailable, double price) {
        this.flightId = flightId;
        this.origin = origin;
        this.destination = destination;
        this.departureTime = departureTime;
        this.arrivalTime = arrivalTime;
        this.seatsAvailable = seatsAvailable;
        this.price = price;
    }

    // Getters and Setters
    // ...
}
```

Database Connection

```
package com.airline.database;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/airline";
    private static final String USER = "root";
    private static final String PASSWORD = "password";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

Flight Management

```
package com.airline.services;

import com.airline.database.DatabaseConnection;
import com.airline.models.Flight;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class FlightService {

    public List<Flight> getAllFlights() {
        List<Flight> flights = new ArrayList<>();
        try (Connection connection = DatabaseConnection.getConnection()) {
            String query = "SELECT * FROM flights";
            PreparedStatement statement = connection.prepareStatement(query);
            ResultSet rs = statement.executeQuery();

            while (rs.next()) {
                flights.add(new Flight(
                        rs.getString("flight_id"),
                        rs.getString("origin"),
                        rs.getString("destination"),
                        rs.getString("departure_time"),
                        rs.getString("arrival_time"),
                        rs.getInt("seats_available"),
                        rs.getDouble("price")
                ));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return flights;
    }
```

```java
    // Additional methods for adding, updating, and deleting flights
}
```

Booking Logic

```java
package com.airline.services;

import com.airline.models.Booking;

public class BookingService {

    public boolean bookFlight(String userId, String flightId, int seatNumber) {
        // Logic to check seat availability, update database, and confirm booking
        return true; // Return success or failure
    }

    public boolean cancelBooking(String bookingId) {
        // Logic to cancel booking and update seats
        return true; // Return success or failure
    }
}
```

Menu (Command Line Interface Example)

```java
package com.airline.ui;

import com.airline.services.FlightService;

public class Menu {
    public static void showMainMenu() {
        System.out.println("Welcome to Airline Reservation System");
        System.out.println("1. View Flights");
        System.out.println("2. Book a Flight");
        System.out.println("3. Cancel Booking");
        System.out.println("4. Admin Login");
        System.out.println("5. Exit");

        // Handle user input and call relevant methods
        FlightService flightService = new FlightService();
        flightService.getAllFlights().forEach(System.out::println);
    }
}
```

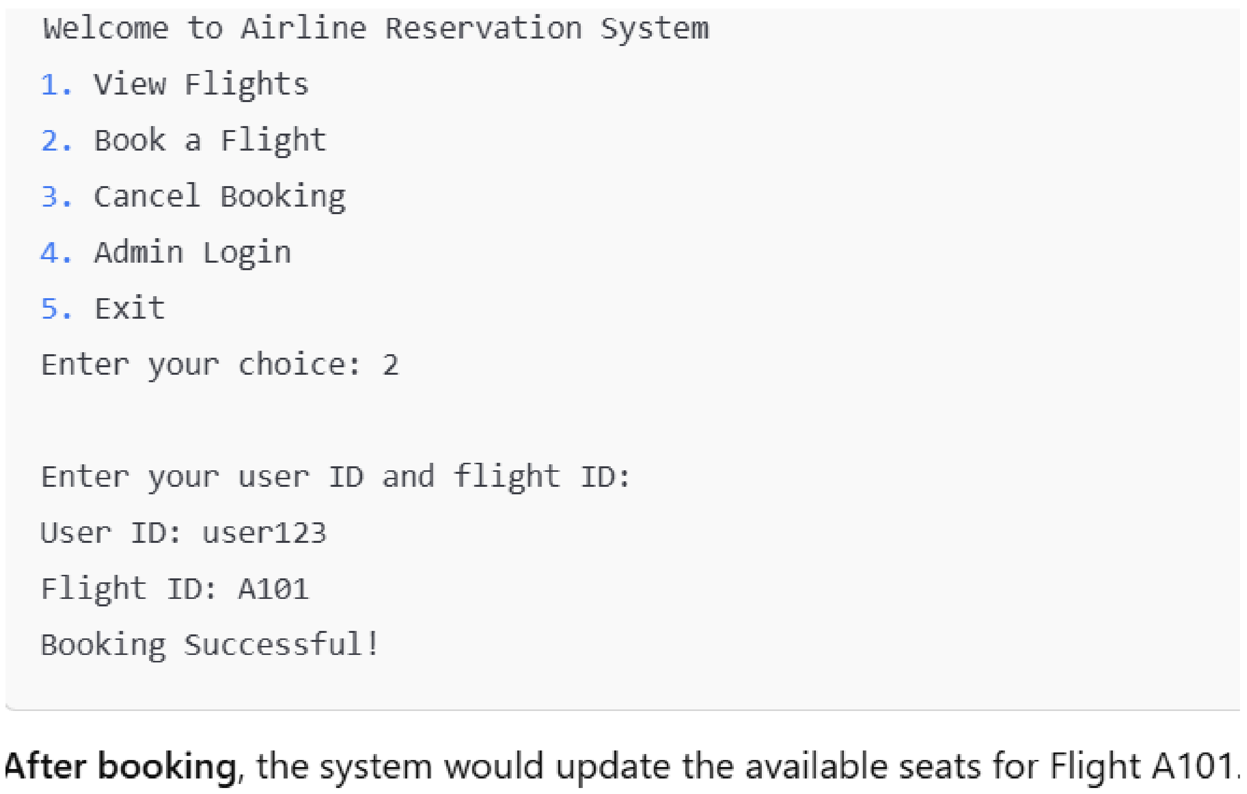
OUTPUT:

VIEW FLIGHT:

```
    Welcome to Airline Reservation System
        1. View Flights
        2. Book a Flight
        3. Cancel Booking
        4. Admin Login
        5. Exit
Enter your choice: 1

Flight ID: A101 | Origin: New York | Destination: Los Angeles | Departure: 2024-12-01
08:00 | Arrival: 2024-12-01 11:00 | Seats Available: 50 | Price: $300
Flight ID: A102 | Origin: Chicago | Destination: Miami | Departure: 2024-12-02 09:00 |
Arrival: 2024-12-02 12:00 | Seats Available: 30 | Price: $200
Flight ID: A103 | Origin: San Francisco | Destination: Boston | Departure: 2024-12-05
07:00 | Arrival: 2024-12-05 10:00 | Seats Available: 20 | Price: $350
...
```

BOOK A FLIGHT:

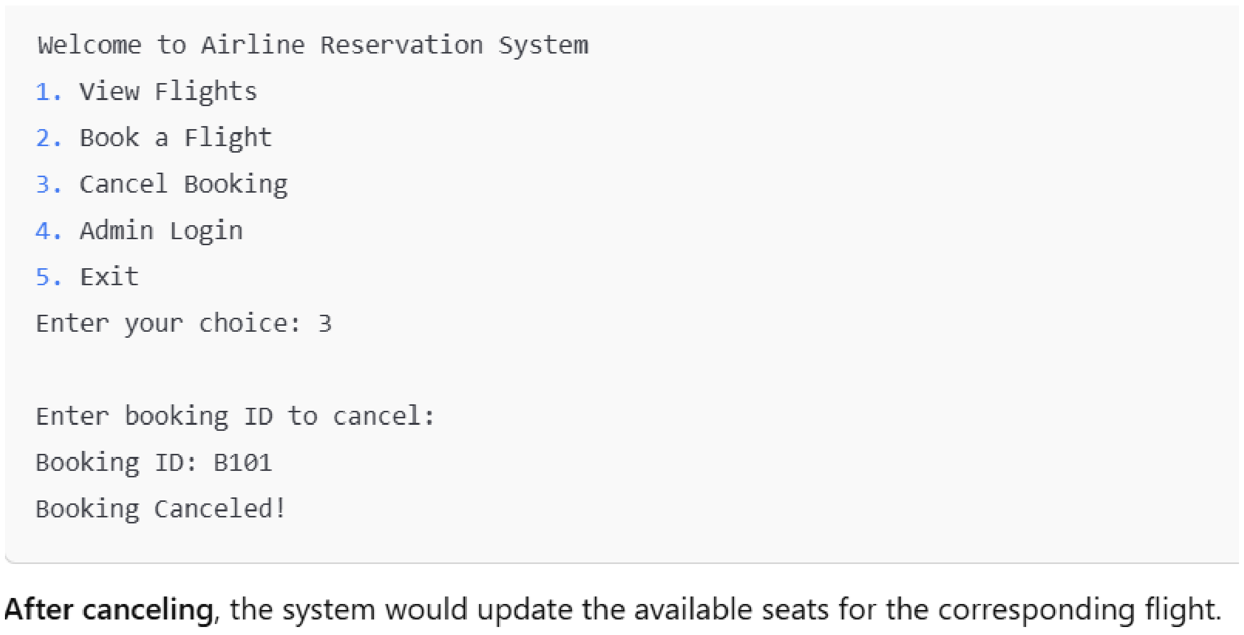


```
 Welcome to Airline Reservation System
1. View Flights
2. Book a Flight
3. Cancel Booking
4. Admin Login
5. Exit
Enter your choice: 2

Enter your user ID and flight ID:
User ID: user123
Flight ID: A101
Booking Successful!
```

After booking, the system would update the available seats for Flight A101.

CANCEL A BOOKING:

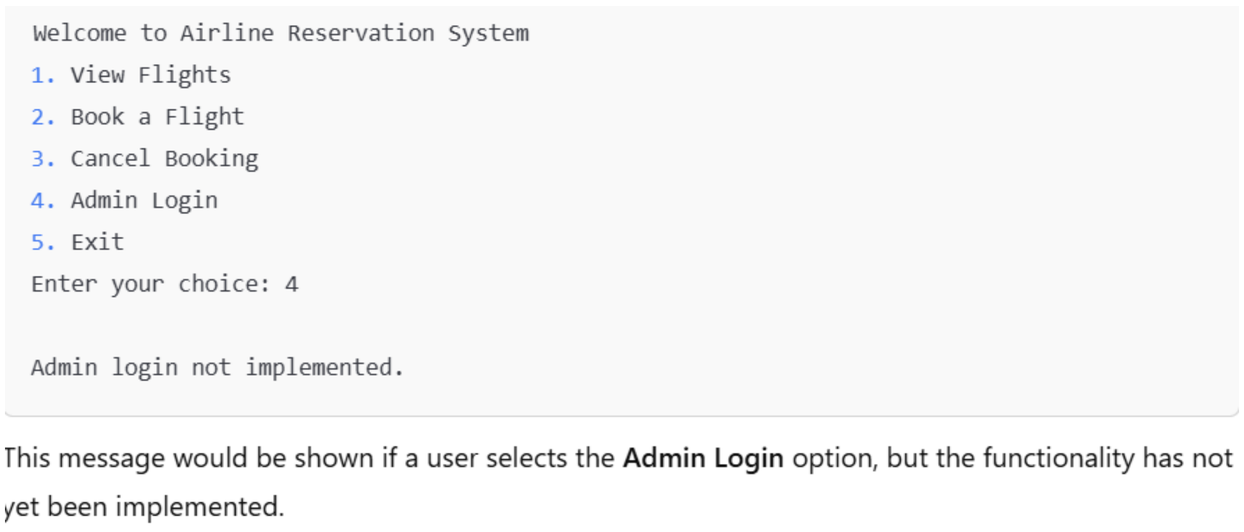


```
Welcome to Airline Reservation System
1. View Flights
2. Book a Flight
3. Cancel Booking
4. Admin Login
5. Exit
Enter your choice: 3

Enter booking ID to cancel:
Booking ID: B101
Booking Canceled!
```

After canceling, the system would update the available seats for the corresponding flight.

ADMIN LOGIN:

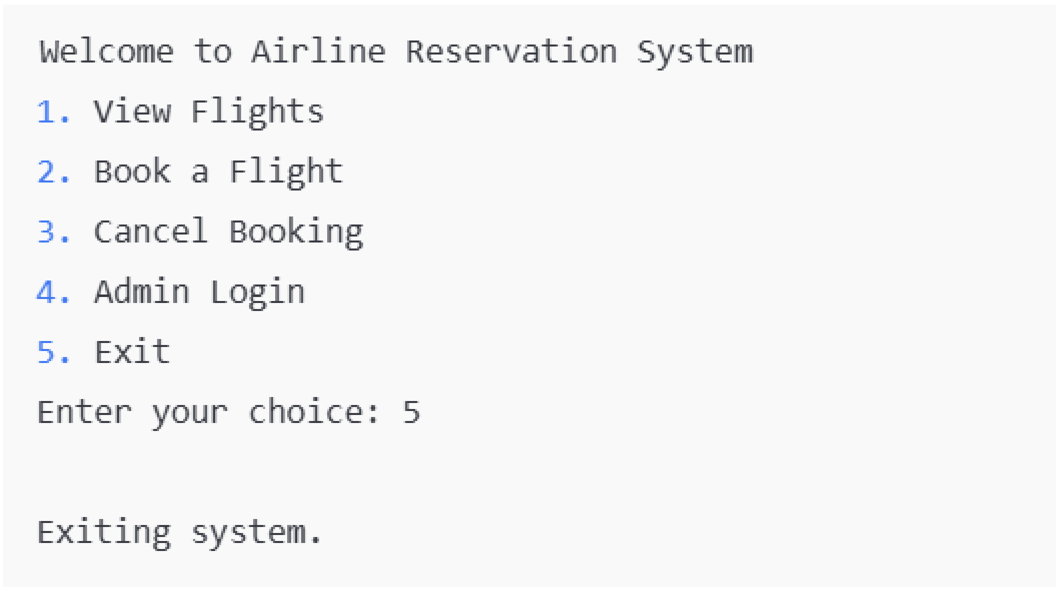


```
Welcome to Airline Reservation System
1. View Flights
2. Book a Flight
3. Cancel Booking
4. Admin Login
5. Exit
Enter your choice: 4

Admin login not implemented.
```

This message would be shown if a user selects the **Admin Login** option, but the functionality has not yet been implemented.

EXIT SYSTEM:



```
Welcome to Airline Reservation System
1. View Flights
2. Book a Flight
3. Cancel Booking
4. Admin Login
5. Exit
Enter your choice: 5

Exiting system.
```

RESULT:

The database construction for the Airline reservation system has been successfullycomplected and connected with mySQL using java.