DOCUMENTATION REPORT

On

FLIGHT MANAGEMENT SYSTEM

By

AeroBlasters

Submitted to:

Mr. Suramya Biswas

Submitted By:

1. Khushmn Sangha
2. Vanshika Rani
3. Soumik Saha
4. Vijay Yadav

CONTENT

1. INTRODUCTION: The Flight Management Project is a comprehensive web application designed using Spring Boot, Eclipse, and MySQL. This project aims to streamline the processes involved in flight booking, managing flights, routes, and airport details while providing an efficient and user-friendly experience for both customers and administrators.

1.1 Key Features:

- **User Roles:** The system distinguishes between customer and admin roles. Both can access basic functionalities like searching for airports, routes, and flights, booking of tickets.
- **Customer Interface:** Customers can browse available flights, view details, and conveniently book tickets. The system facilitates ticket cancellation as well.
- **Admin Interface:** In addition to customer functionalities, administrators have the privilege to add new airports, routes, and flight schedules. They can also access comprehensive passenger and booking details, providing valuable insights for flight management.

1.2 Technology Stack:

- **Spring Boot**: Provides a robust and efficient framework for building web applications.
- **Eclipse IDE:** Offers a user-friendly development environment for coding and managing the project.
- **MySQL Database:** Stores critical flight information, including airports, routes, flights, bookings, and passengers.

1.3 Benefits:

- **Enhanced User Experience:** Separate interfaces for customers and admins streamline the user experience for each role.
- **Simplified Flight Management:** Customers can easily book and manage their flights, while admins have comprehensive control over flight operations.
- **Data-Driven Insights:** Comprehensive passenger and booking details empower admins with data-driven insights to optimize flight management strategies.

1.4 Customer Interface: -

Customers have access to a dedicated interface where they can:

- **View Airports, Routes, and Flights**: Customers can browse detailed information about available airports, flight routes, and scheduled flights.
- **Book Tickets**: Users can seamlessly search for and book flights based on their preferences.
- **Cancel Tickets**: If plans change, customers can easily cancel their booked tickets through the interface.

1.5 Admin Interface: -The admin interface is tailored to the needs of the administrators, offering additional functionalities to ensure the smooth operation of the flight management system:

- **Manage Airports, Routes, and Flights**: Admins can add new airports, define routes, and schedule flights, ensuring up-to-date and accurate information for customers.
- **View Detailed Ticket and Passenger Information**: Administrators have access to detailed views of all tickets and passenger information, enabling efficient management and support.

The Flight Management Project thus offers a holistic solution for both customers and administrators, enhancing the efficiency of flight management operations and improving the overall user experience.

2. Technologies Used:

The Flight Management Project leverages a variety of modern technologies to provide a seamless and efficient experience for both customers and administrators.

2.1 Backend:

- **Java**: The core programming language used for developing the business logic and server-side functionalities of the application.
- **Spring Boot**: A robust framework for building Java-based web applications, providing a streamlined development process with pre-configured templates and reducing boilerplate code.
- **Hibernate**: An ORM (Object-Relational Mapping) tool used to facilitate the interaction between Java applications and the MySQL database, ensuring smooth data persistence and retrieval.

2.2 Frontend:

- **JSP (JavaServer Pages)**: Utilized for creating dynamic web pages on the server side, allowing seamless integration with backend logic.
- **HTML, CSS, JavaScript**: Standard web technologies used to design and structure web pages, ensuring a responsive and user-friendly interface.
- **Bootstrap**: A popular front-end framework that aids in creating responsive and visually appealing web designs, ensuring compatibility across various devices and screen sizes.

2.3 Database:

- **MySQL**: A reliable and widely-used relational database management system. MySQL is employed to store and manage all data related to airports, routes, flights, tickets, and passengers, ensuring efficient data handling and integrity.

2.4 Build Tool:

- **Maven**: A build automation tool used to manage project dependencies, build the project, and streamline the development process, making it easier to manage external libraries and plugins.

2.5 Server:

- **Apache Tomcat**: A widely-used web server and servlet container that deploys and serves Java web applications. Tomcat provides a robust environment for running the Spring Boot application, handling HTTP requests, and ensuring efficient server-side processing.

2.6 Version Control:

- **Git**: A distributed version control system that tracks changes in the source code, facilitating collaboration among team members and maintaining a history of code changes, ensuring efficient project management.

3. Project Structure: Diagram represents the Structured format and outlines of Flight Management System project:

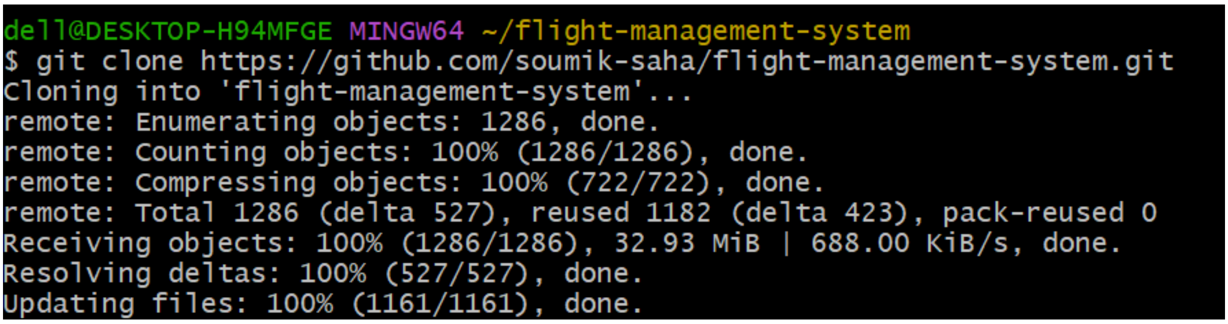
```
aeroBlasters/
├── src/main/java/com/aeroBlasters/
│   ├── controller/
│   │   ├── AirportController.java
│   │   ├── GlobalExceptionHandler.java
│   │   ├── LoginController.java
│   │   ├── RouteFlightController.java
│   │   ├── TicketController.java
│   ├── bean/
│   │   ├── Airport.java
│   │   ├── Flight.java
│   │   ├── FlightUser.java
│   │   ├── Passenger.java
│   │   ├── Route.java
│   │   ├── Ticket.java
│   │   ├── TicketPassengerEmbed.java
│   ├── config/
│   │   ├── EncoderConfig.java
│   │   ├── SecurityConfig.java
│   ├── dao/
│   │   ├── AirportDao.java
│   │   ├── AirportDaoImpl.java
│   │   ├── AirportRepository.java
│   │   ├── FlightDao.java
│   │   ├── FlightDaoImpl.java
│   │   ├── FlightRepository.java
│   │   ├── FlightUserDao.java
│   │   ├── FlightUserRepository.java
│   │   ├── PassengerDao.java
│   │   ├── PassengerDaoImpl.java
│   │   ├── PassengerRepository.java
│   │   ├── RouteDao.java
│   │   ├── RouteDaoImpl.java
│   │   ├── RouteRepository.java
│   │   ├── TicketDao.java
│   │   ├── TicketDaoImpl.java
│   │   ├── TicketRepository.java
│   ├── exception/
│   │   ├── AirportException.java
│   │   ├── FlightException.java
│   │   ├── RouteException.java
│   │   ├── TicketException.java
│   ├── service/
│   │   ├── FlightService.java
│   │   ├── FlightUserService.java
│   │   ├── RouteService.java
│   │   ├── TicketService.java
│   ├── AeroblastersApplication.java
├── src/main/resources/
│   ├── static/
│   ├── application.properties
├── src/main/webapp/WEB-INF/
│   ├── views/
│   │   ├── adminIndex.jsp
│   │   ├── airportEntryPage.jsp
│   │   ├── airportErrorPage.jsp
│   │   ├── airportReportPage.jsp
│   │   ├── airportSelectPage.jsp
│   │   ├── airportShowPage.jsp
│   │   ├── customerIndex.jsp
│   │   ├── errorPage.jsp
│   │   ├── flightEntryPage.jsp
│   │   ├── flightErrorPage.jsp
│   │   ├── flightReportPage.jsp
│   │   ├── index_2.jsp
│   │   ├── index.jsp
│   │   ├── loginErrorPage.jsp
│   │   ├── loginPage.jsp
│   │   ├── newUserEntry.jsp
│   │   ├── passengerReportPage.jsp
│   │   ├── routeEntryPage.jsp
│   │   ├── routeFlightShowPage.jsp
│   │   ├── routeErrorPage.jsp
│   │   ├── routeReportPage.jsp
│   │   ├── routeSelectPage.jsp
│   │   ├── showTicketPage.jsp
│   │   ├── ticketBookingPage.jsp
│   │   ├── ticketBookingPage copy.jsp
│   ├── web.xml
├── pom.xml
├── README.md
```

4. Flight Management System - Setup Instructions:

Follow these steps to set up and run the Flight Management System project on local machine.

4.1 Clone the Repository:

First, clone the repository to your local machine and navigate into the project directory.

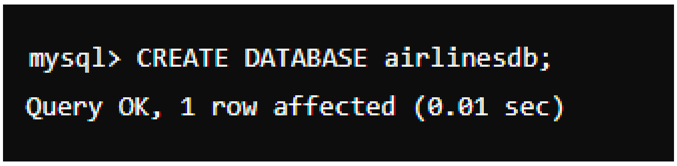


```
dell@DESKTOP-H94MFGE MINGW64 ~/flight-management-system
$ git clone https://github.com/soumik-saha/flight-management-system.git
Cloning into 'flight-management-system'...
remote: Enumerating objects: 1286, done.
remote: Counting objects: 100% (1286/1286), done.
remote: Compressing objects: 100% (722/722), done.
remote: Total 1286 (delta 527), reused 1182 (delta 423), pack-reused 0
Receiving objects: 100% (1286/1286), 32.93 MiB | 688.00 KiB/s, done.
Resolving deltas: 100% (527/527), done.
Updating files: 100% (1161/1161), done.
```

4.2 Setup the Database:

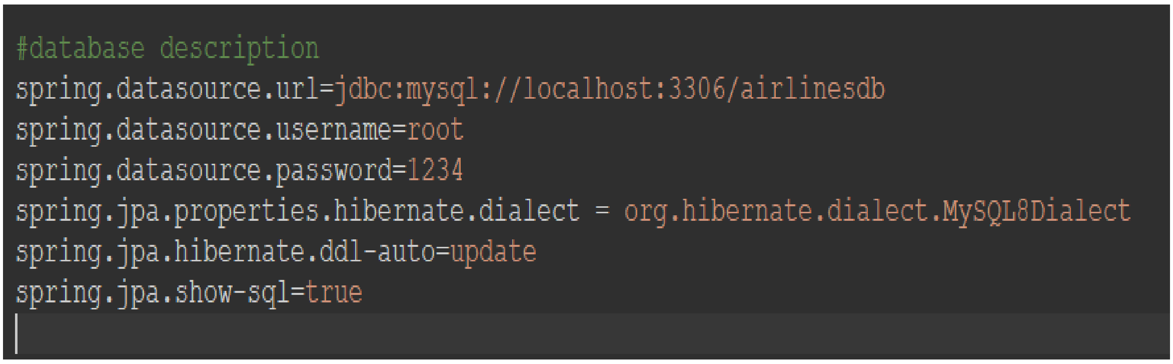
Install MySQL and create a database named `**airlinesdb**`. Update the database configuration in `src/main/resources/application.properties` with MySQL credentials.

4.2.1 Create the Database:



```
mysql> CREATE DATABASE airlinesdb;
Query OK, 1 row affected (0.01 sec)
```

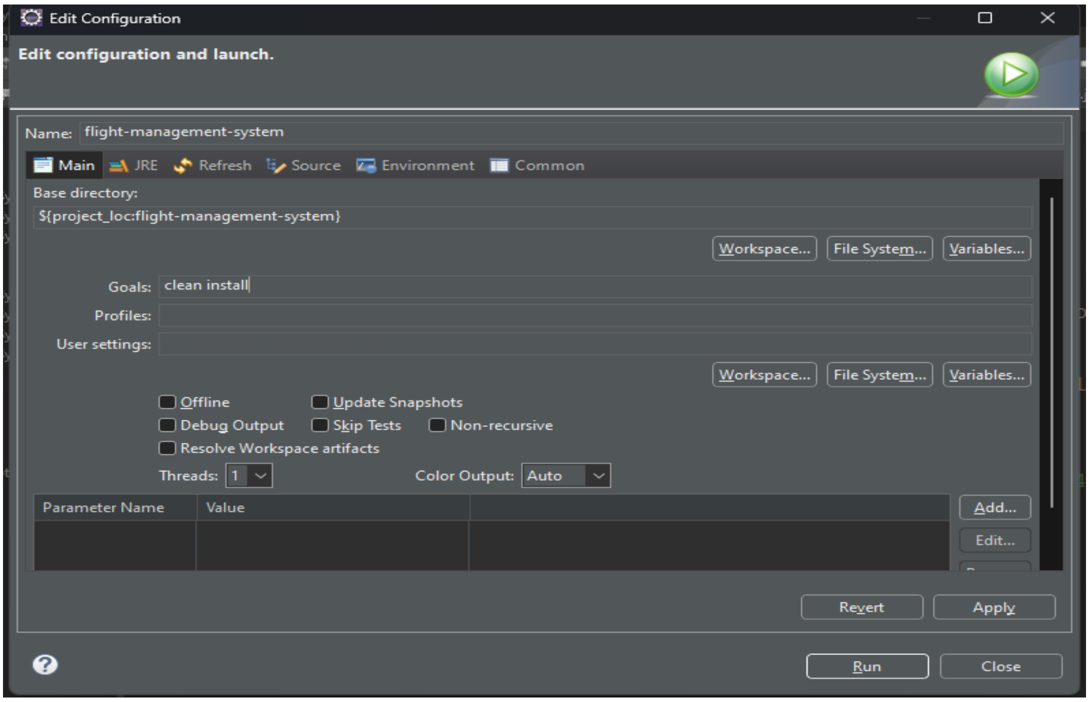
4.2.2 Update application.properties :



```
#database description
spring.datasource.url=jdbc:mysql://localhost:3306/airlinesdb
spring.datasource.username=root
spring.datasource.password=1234
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

4.3 Build the Project:

Use Maven to clean and build the project. This will compile the code, run tests, and package the application.



Name: flight-management-system

Base directory: ${project_loc:flight-management-system}

Goals: clean install

Profiles:

User settings:

□ Offline □ Update Snapshots

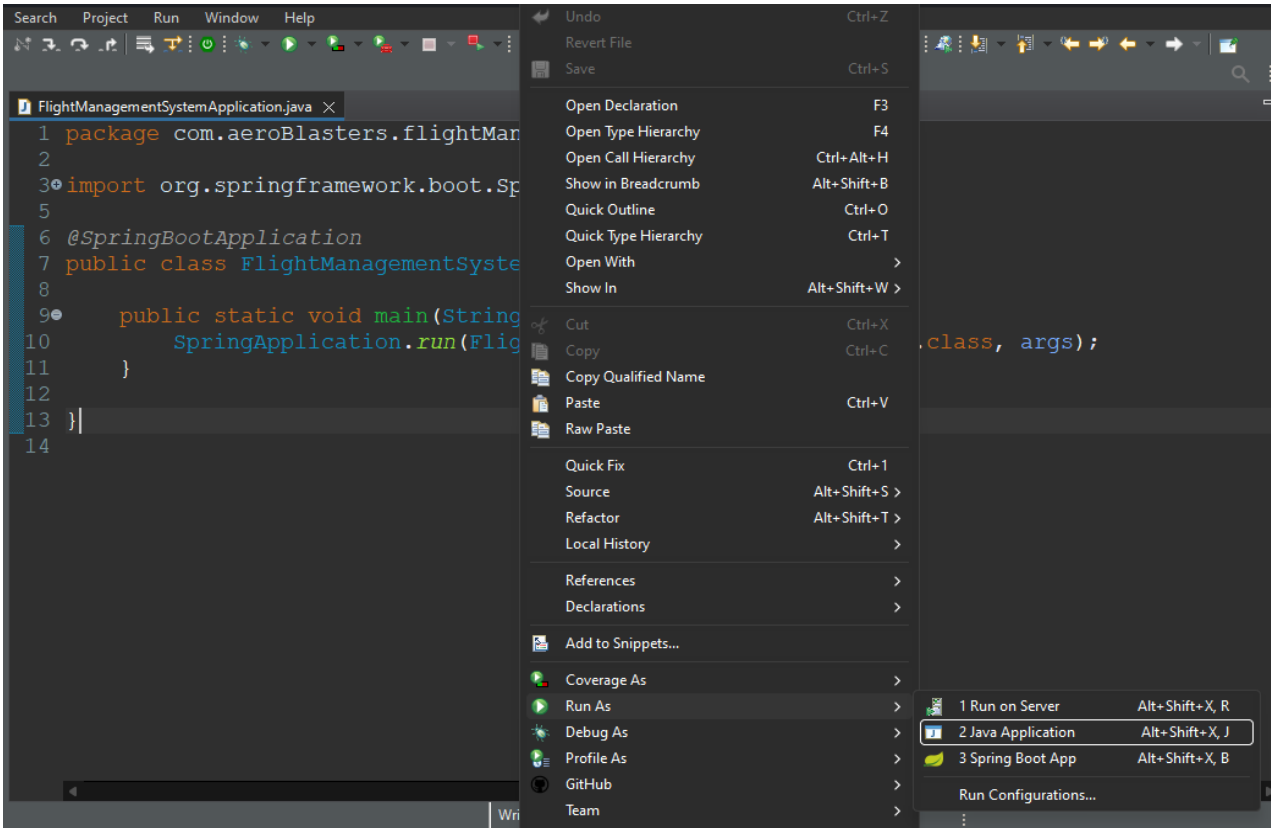
□ Debug Output □ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

Threads: 1 Color Output: Auto

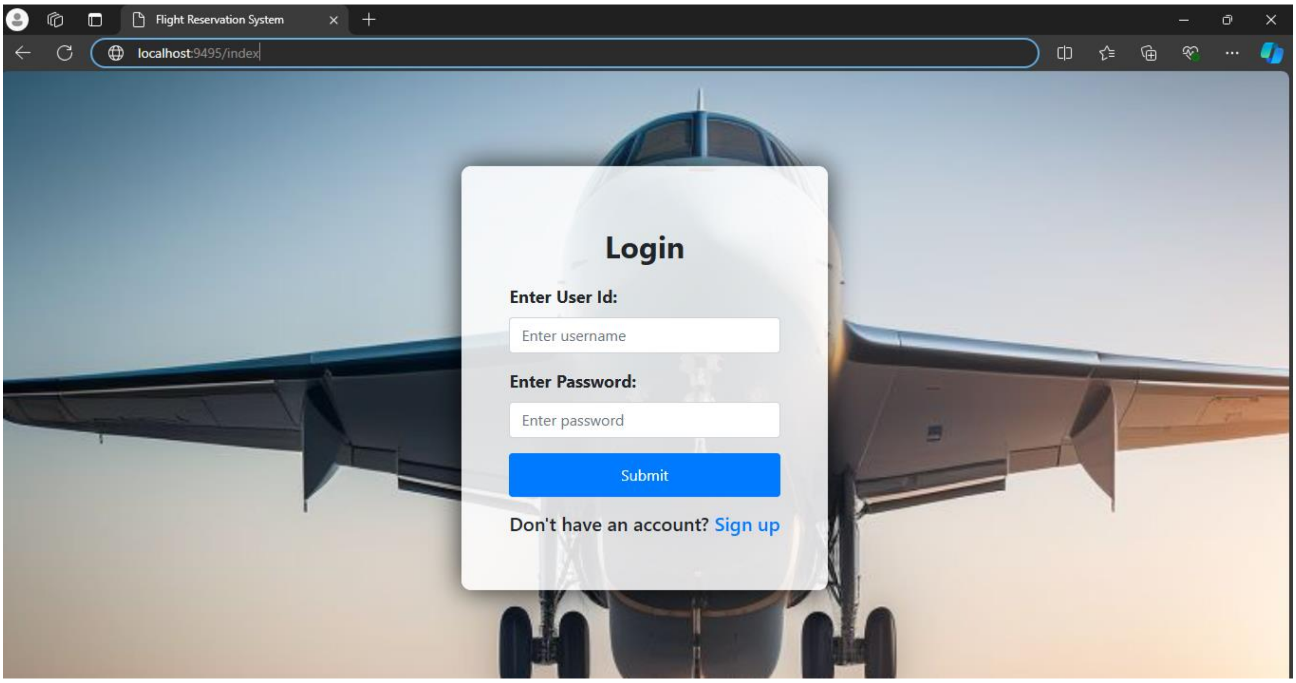
4.4 Run the Application: Start the Spring Boot application using your IDE. Here's how to do it in Eclipse:

1. Right-click on **`FlightManagementSystemApplication.java`** in the **`src/main/java/com/aeroBlasters/flightManagementSystem`** directory.
2. Navigate to **`Run As`** -> **`Spring Boot App`**.



```
package com.aeroBlasters.flightMan

import org.springframework.boot.Sp

@SpringBootApplication
public class FlightManagementSyste

    public static void main(String
        SpringApplication.run(Flig.class, args);
    }

}
```

4.5 Access the Application: Once the application is running, open your web browser and navigate to **http://localhost:9495/index**



Login

Enter User Id:

Enter username

Enter Password:

Enter password

Don't have an account? Sign up

4.6 Explore the Application

4.6.1 Customer Interface: Customers can view airports, routes, and flights, book tickets, and cancel tickets.

4.6.2 Admin Interface: Administrators have additional privileges to add new airports, routes, and flights. They can also view full details of tickets and passengers.

5. Key Functionalities of the Flight Management System:

The Flight Management System is designed to streamline the process of managing flights, routes, airports, and bookings. Here are the key functionalities:

5.1 User Management –

- **Customer Registration and Login:** Allows customers to create accounts, log in, and manage their profiles.
- **Admin Login:** Secure login for administrators to access the admin panel.

5.2 Airport Management –

- **View Airports:** Both customers and admins can view a list of all airports.
- **Add Airports:** Admins can add new airports, with airport code for the user.

5.3 Route Management –

- **View Routes:** Both customers and admins can view all available routes.
- **Add Routes:** Admins can manage route details, including adding new routes from the system.

5.4 Flight Management –

- **View Flights:** Both customers and admins can view all scheduled flights.
- **Add Flights:** Admins have the ability to add new flights.

5.5 Ticket Management –

- **Book Tickets:** Customers can book tickets for available flights.
- **View Bookings:** Customers can view their current booked tickets.
- **Cancel Tickets:** Customers can cancel their bookings if needed.
- **View Ticket Details:** Admins can view detailed information about each ticket, including passenger details.

5.6 Passenger Management –

- **View Passenger Details:** Admins can view detailed information about passengers for each flight.

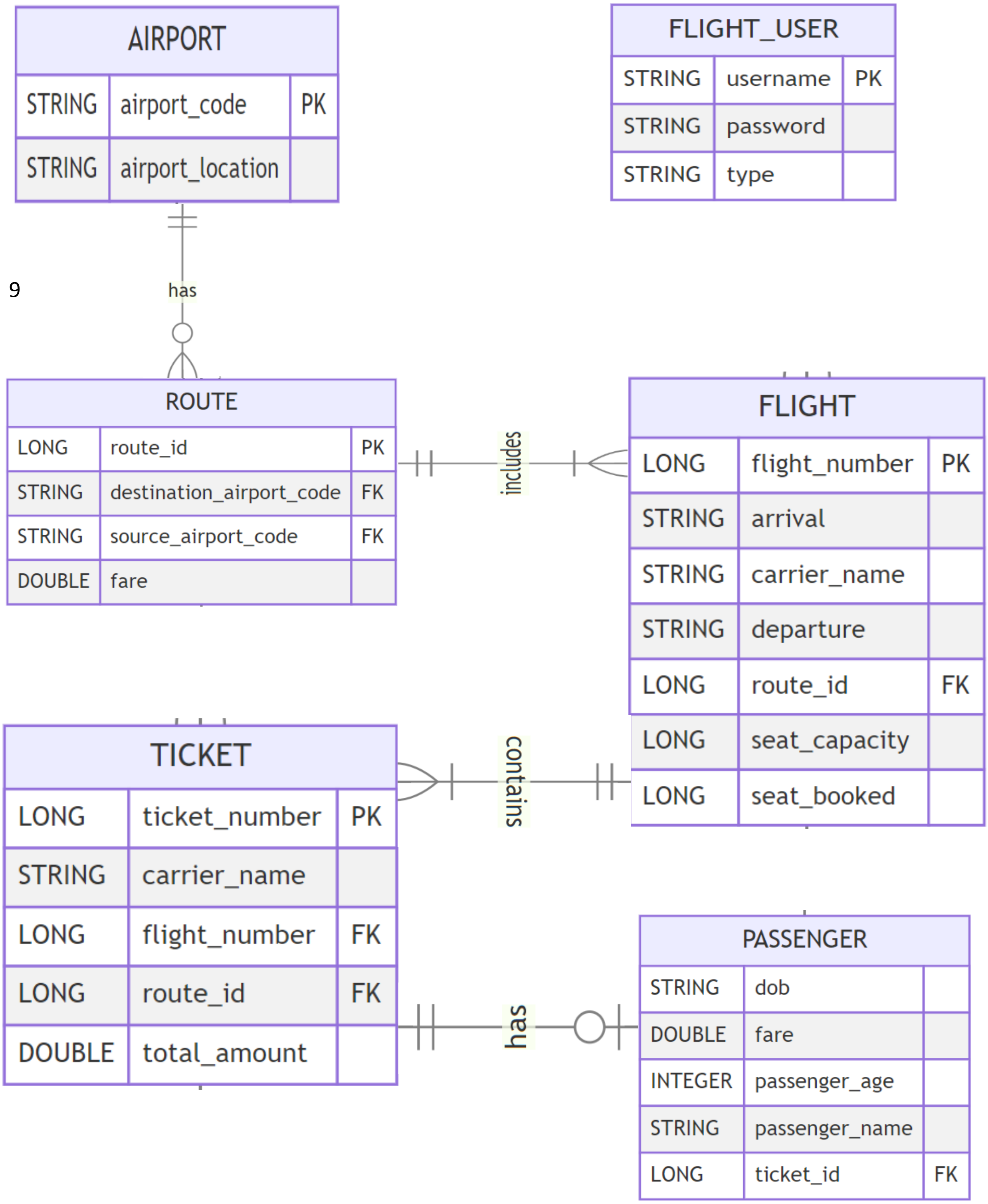
5.7 Reporting –

- **Generate Reports:** Admins can generate reports on airports, flights, routes, bookings, and passengers to gain insights and make informed decisions.

5.8 Security –

- **Authentication and Authorization:** Secure authentication for customers and admins, ensuring that only authorized users can access certain functionalities.
- **Data Encryption:** Sensitive data, such as passwords, is encrypted to ensure data security.

6. Entity Relationship Diagram –



AIRPORT		
STRING	airport_code	PK
STRING	airport_location	

FLIGHT_USER		
STRING	username	PK
STRING	password	
STRING	type	

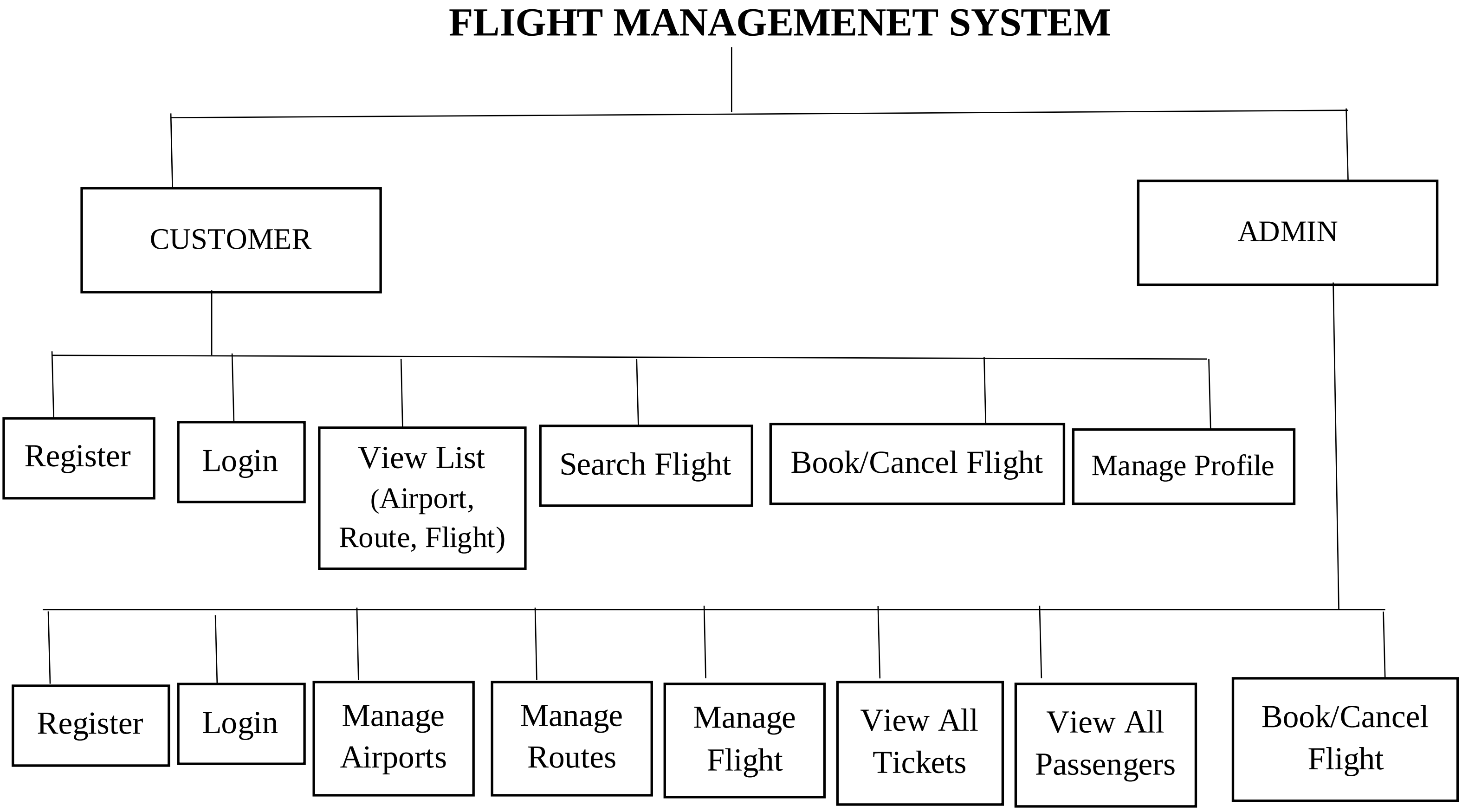
ROUTE		
LONG	route_id	PK
STRING	destination_airport_code	FK
STRING	source_airport_code	FK
DOUBLE	fare	

FLIGHT		
LONG	flight_number	PK
STRING	arrival	
STRING	carrier_name	
STRING	departure	
LONG	route_id	FK
LONG	seat_capacity	
LONG	seat_booked	

TICKET		
LONG	ticket_number	PK
STRING	carrier_name	
LONG	flight_number	FK
LONG	route_id	FK
DOUBLE	total_amount	

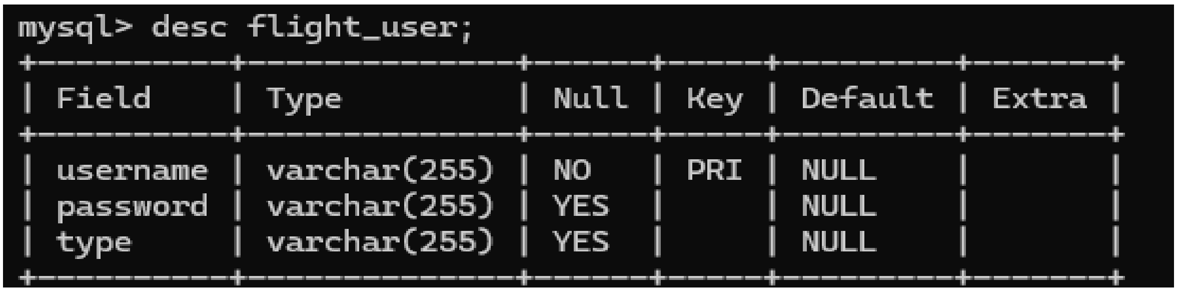
PASSENGER		
STRING	dob	
DOUBLE	fare	
INTEGER	passenger_age	
STRING	passenger_name	
LONG	ticket_id	FK

7. System Workflow Diagram: Flight Management



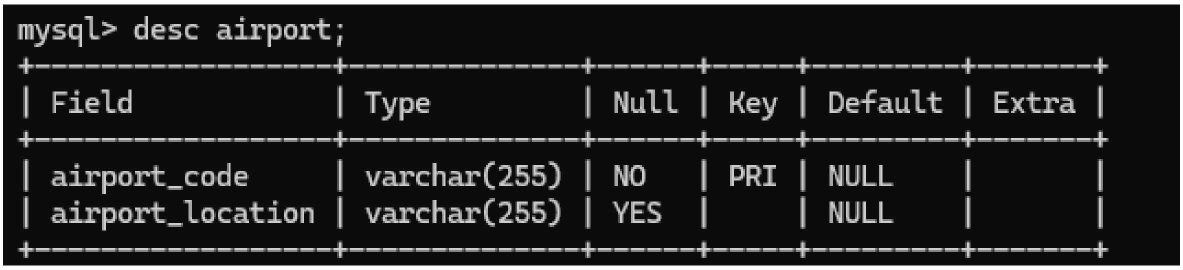
8. Database Schema: Flight Management System:

8.1 Flight_user



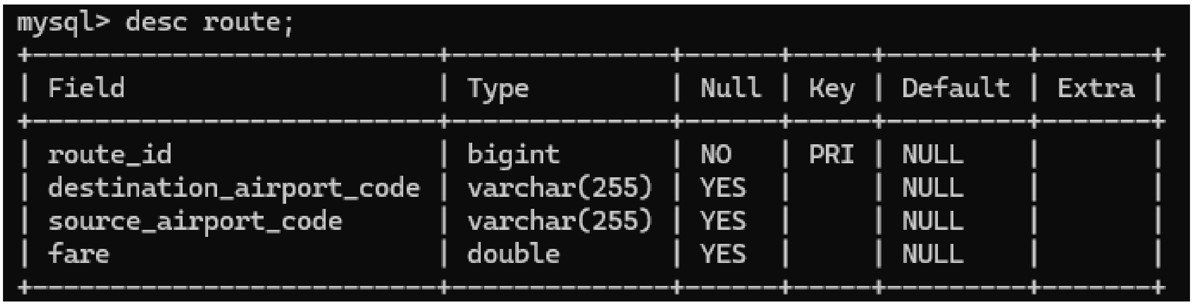
```
mysql> desc flight_user;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| username | varchar(255) | NO   | PRI | NULL    |       |
| password | varchar(255) | YES  |     | NULL    |       |
| type     | varchar(255) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

8.2 Airport



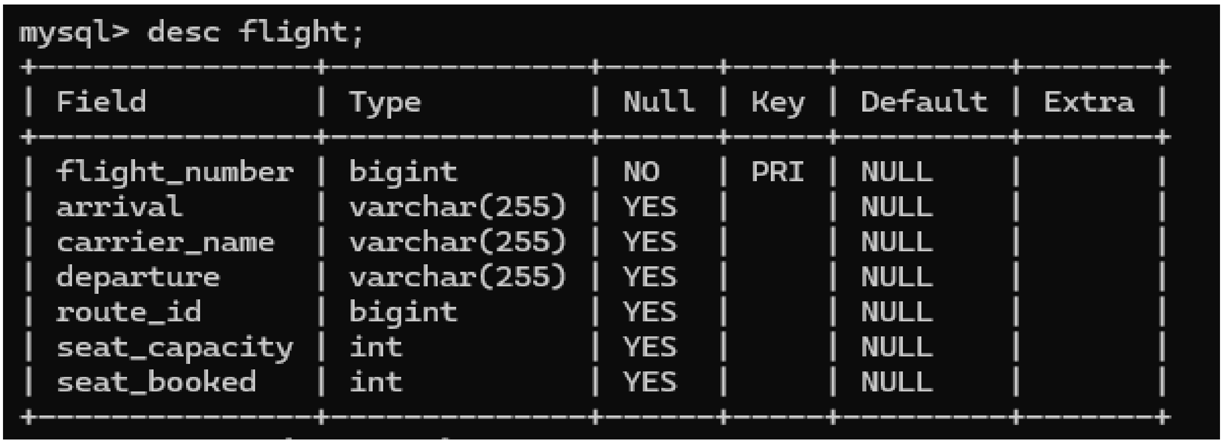
```
mysql> desc airport;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| airport_code     | varchar(255) | NO   | PRI | NULL    |       |
| airport_location | varchar(255) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
```

8.3 Route



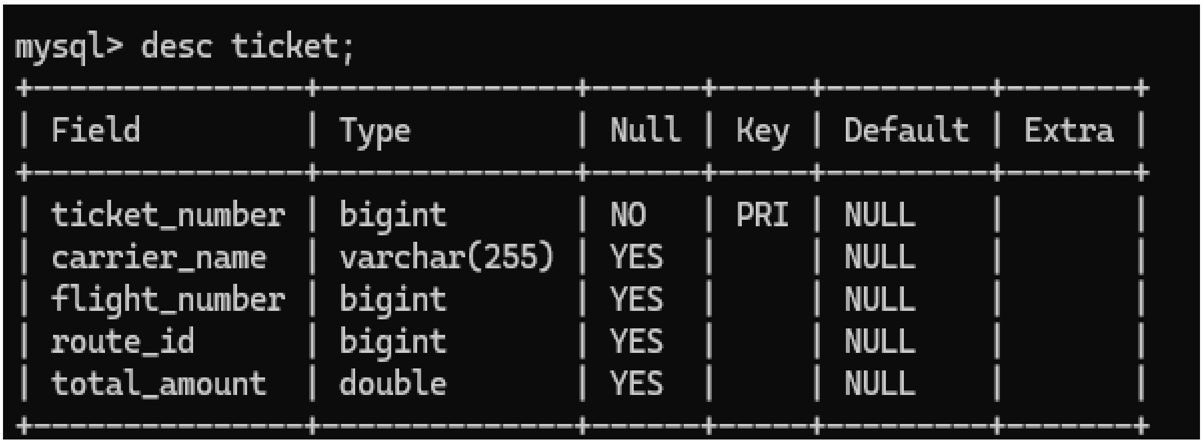
```
mysql> desc route;
+--------------------------+--------------+------+-----+---------+-------+
| Field                    | Type         | Null | Key | Default | Extra |
+--------------------------+--------------+------+-----+---------+-------+
| route_id                 | bigint       | NO   | PRI | NULL    |       |
| destination_airport_code | varchar(255) | YES  |     | NULL    |       |
| source_airport_code      | varchar(255) | YES  |     | NULL    |       |
| fare                     | double       | YES  |     | NULL    |       |
+--------------------------+--------------+------+-----+---------+-------+
```

8.4 Flight



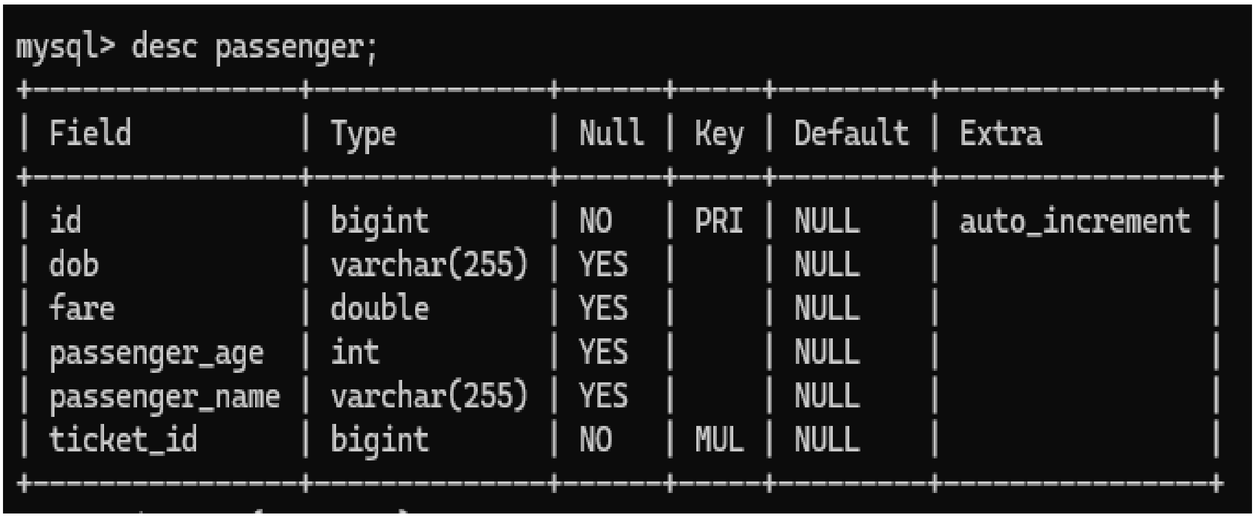
```
mysql> desc flight;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| flight_number | bigint       | NO   | PRI | NULL    |       |
| arrival       | varchar(255) | YES  |     | NULL    |       |
| carrier_name  | varchar(255) | YES  |     | NULL    |       |
| departure     | varchar(255) | YES  |     | NULL    |       |
| route_id      | bigint       | YES  |     | NULL    |       |
| seat_capacity | int          | YES  |     | NULL    |       |
| seat_booked   | int          | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

8.5 Ticket



```
mysql> desc ticket;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| ticket_number | bigint       | NO   | PRI | NULL    |       |
| carrier_name  | varchar(255) | YES  |     | NULL    |       |
| flight_number | bigint       | YES  |     | NULL    |       |
| route_id      | bigint       | YES  |     | NULL    |       |
| total_amount  | double       | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

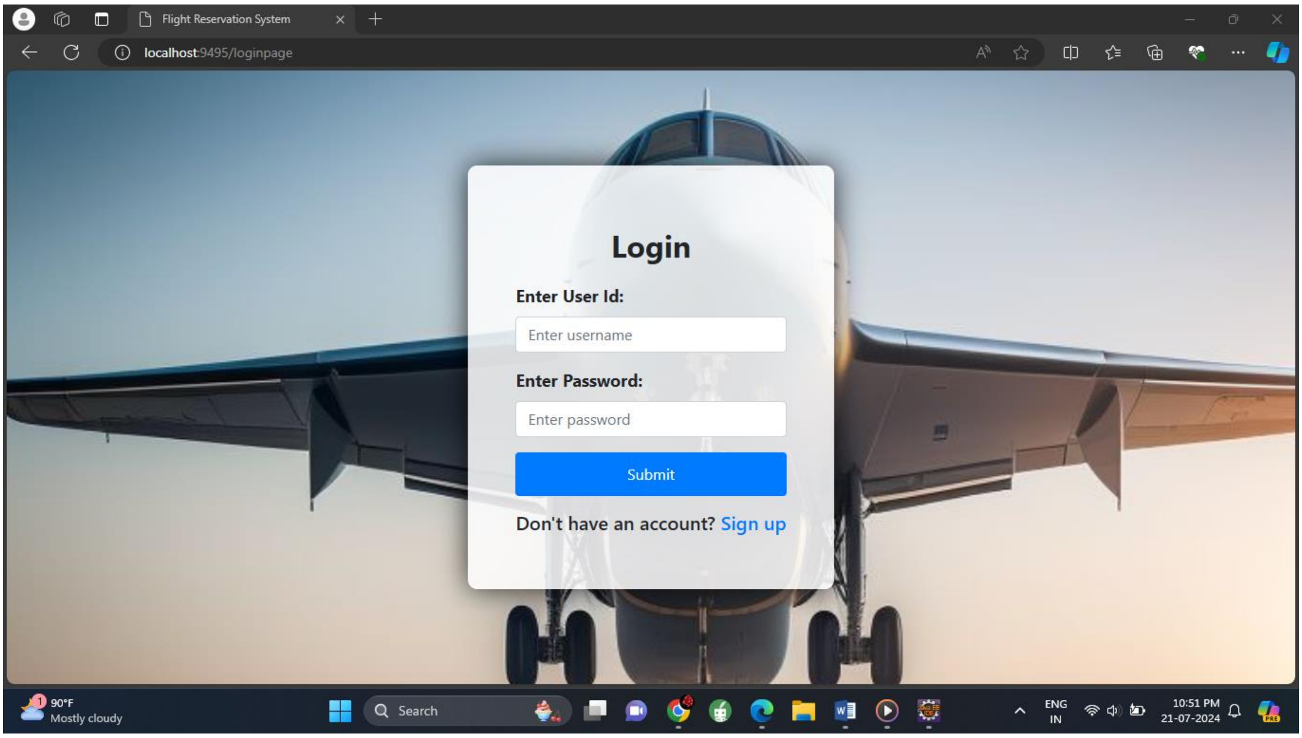
8.6 Passenger



```
mysql> desc passenger;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id             | bigint       | NO   | PRI | NULL    | auto_increment |
| dob            | varchar(255) | YES  |     | NULL    |                |
| fare           | double       | YES  |     | NULL    |                |
| passenger_age  | int          | YES  |     | NULL    |                |
| passenger_name | varchar(255) | YES  |     | NULL    |                |
| ticket_id      | bigint       | NO   | MUL | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
```

9. Flight Booking and Management System: Output Screens:

9.1 Login Page



Login

Enter User Id:

Enter username

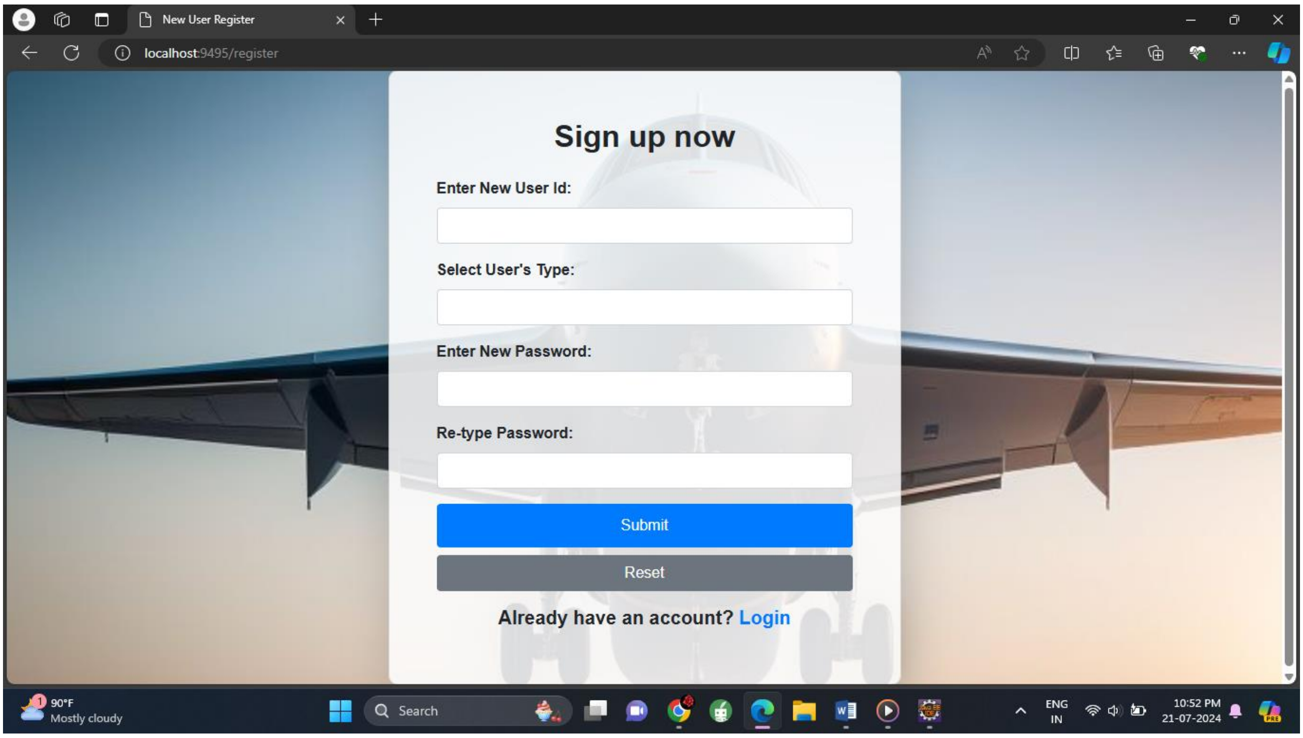
Enter Password:

Enter password

Submit

Don't have an account? Sign up

9.2 Register New User



Sign up now

Enter New User Id:

Select User's Type:

Enter New Password:

Re-type Password:

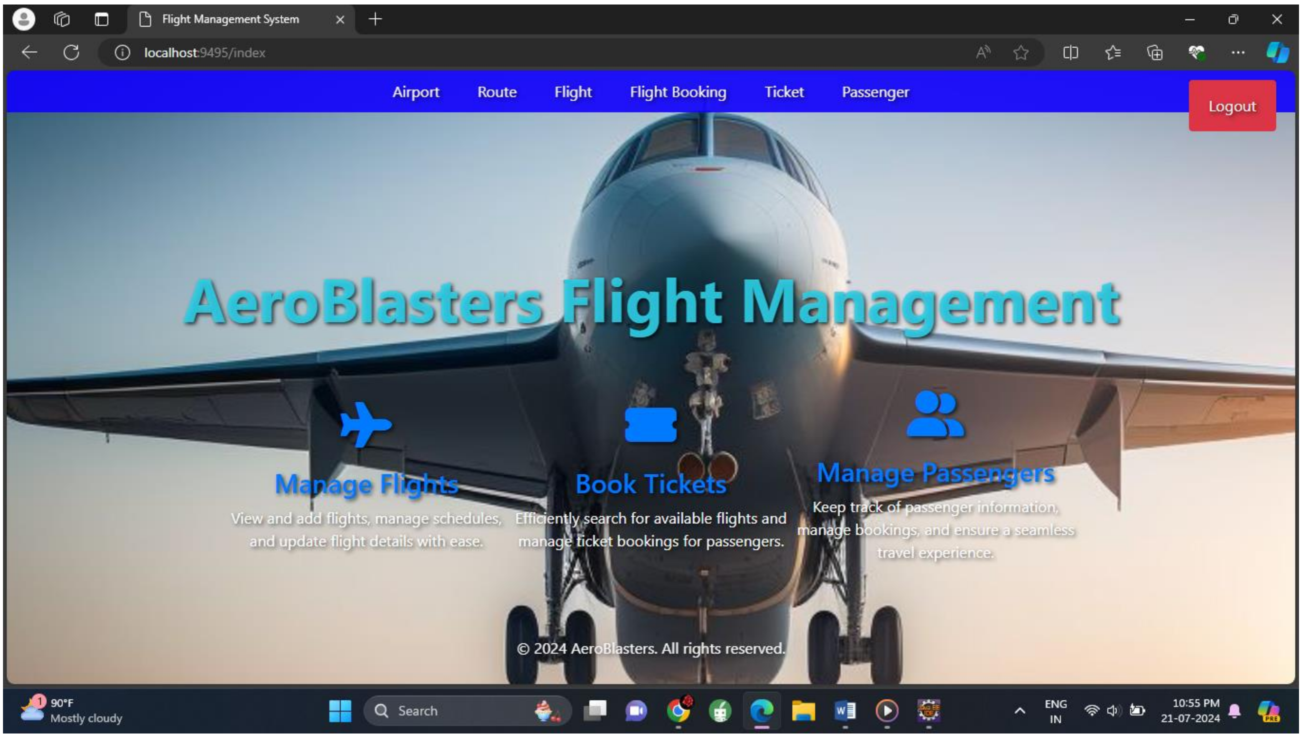
Submit

Reset

Already have an account? Login

9.3 Index Page

9.3.1 Admin Index



AeroBlasters Flight Management

Manage Flights

View and add flights, manage schedules, and update flight details with ease.

Book Tickets

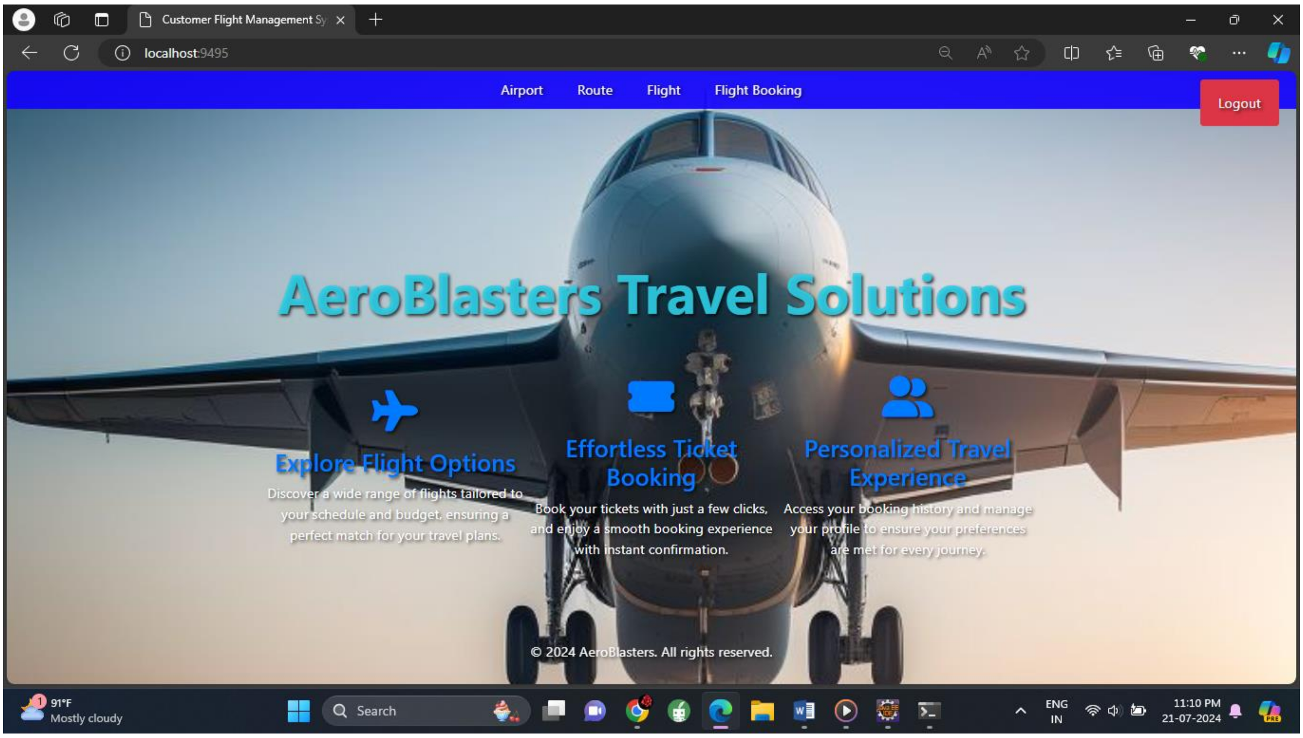
Efficiently search for available flights and manage ticket bookings for passengers.

Manage Passengers

Keep track of passenger information, manage bookings, and ensure a seamless travel experience.

© 2024 AeroBlasters. All rights reserved.

9.3.2 Customer Index



AeroBlasters Travel Solutions

Explore Flight Options

Discover a wide range of flights tailored to your schedule and budget, ensuring a perfect match for your travel plans.

Effortless Ticket Booking

Book your tickets with just a few clicks, and enjoy a smooth booking experience with instant confirmation.

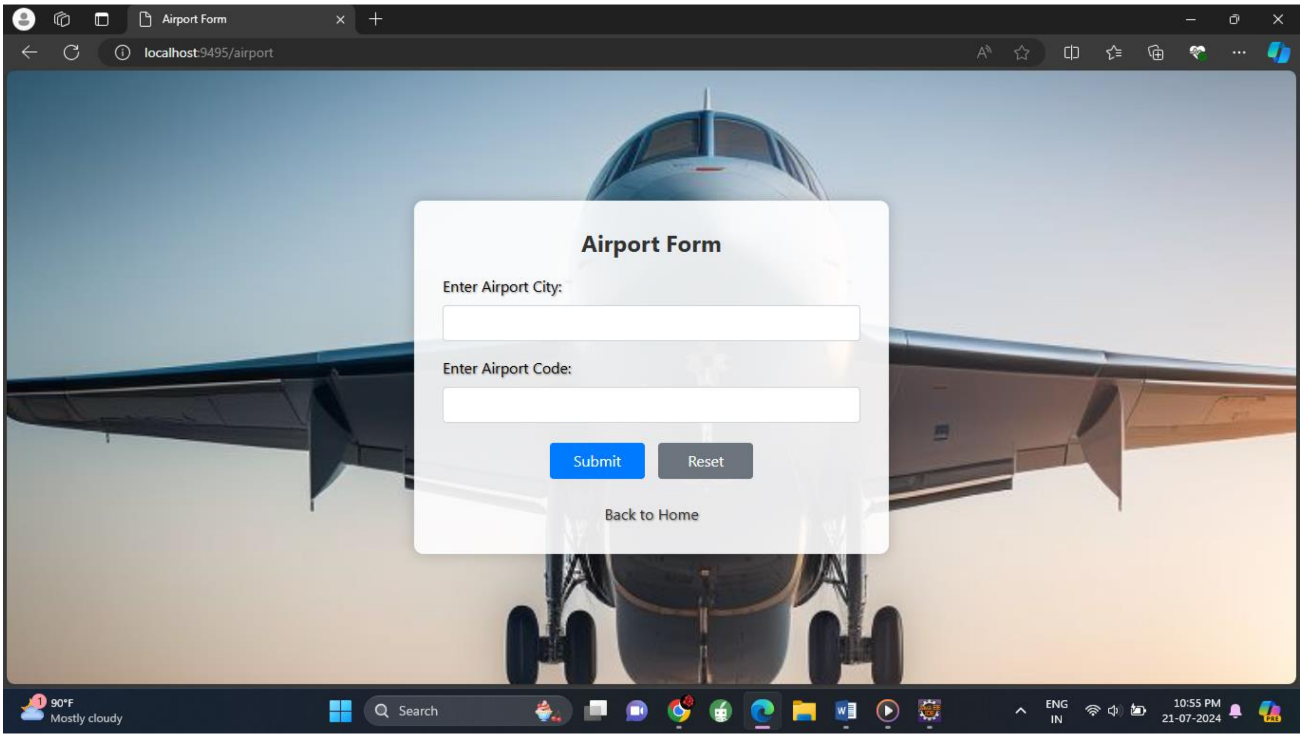
Personalized Travel Experience

Access your booking history and manage your profile to ensure your preferences are met for every journey.

© 2024 AeroBlasters. All rights reserved.

9.4 Airport

9.4.1 Airport Addition



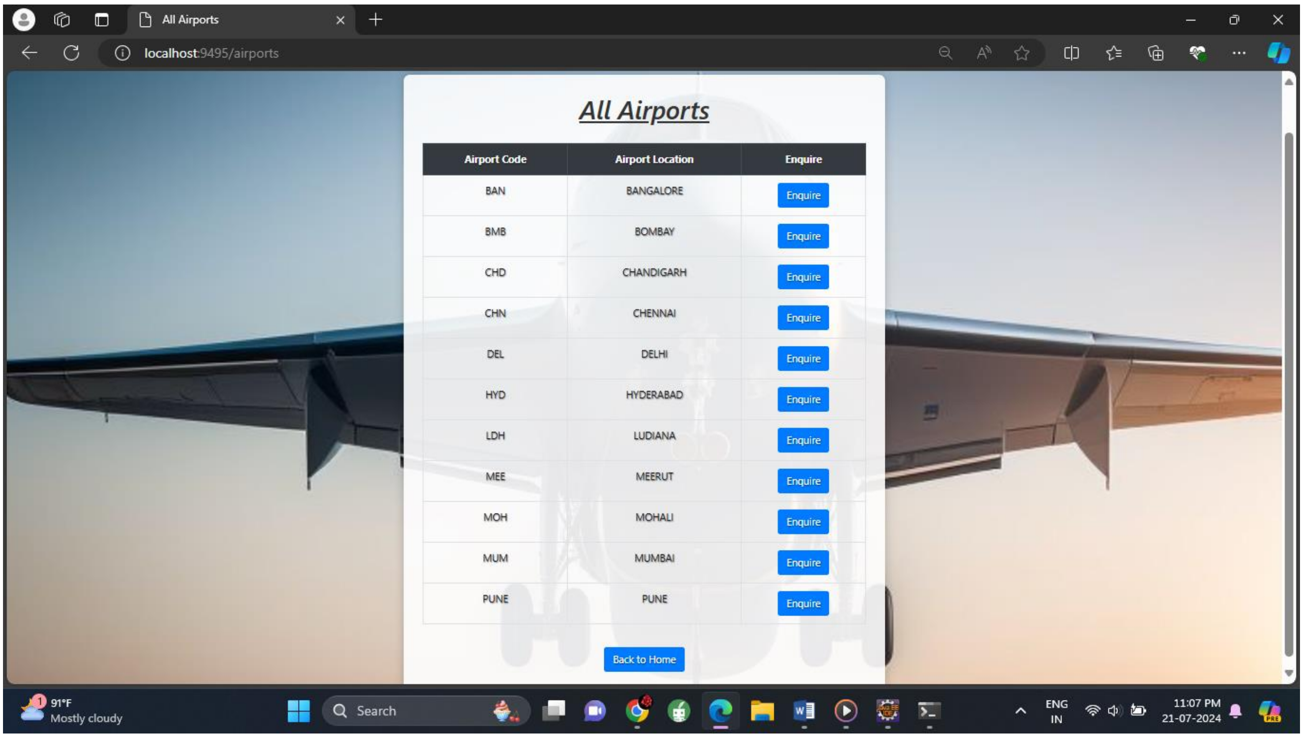
Airport Form

Enter Airport City:

Enter Airport Code:

Back to Home

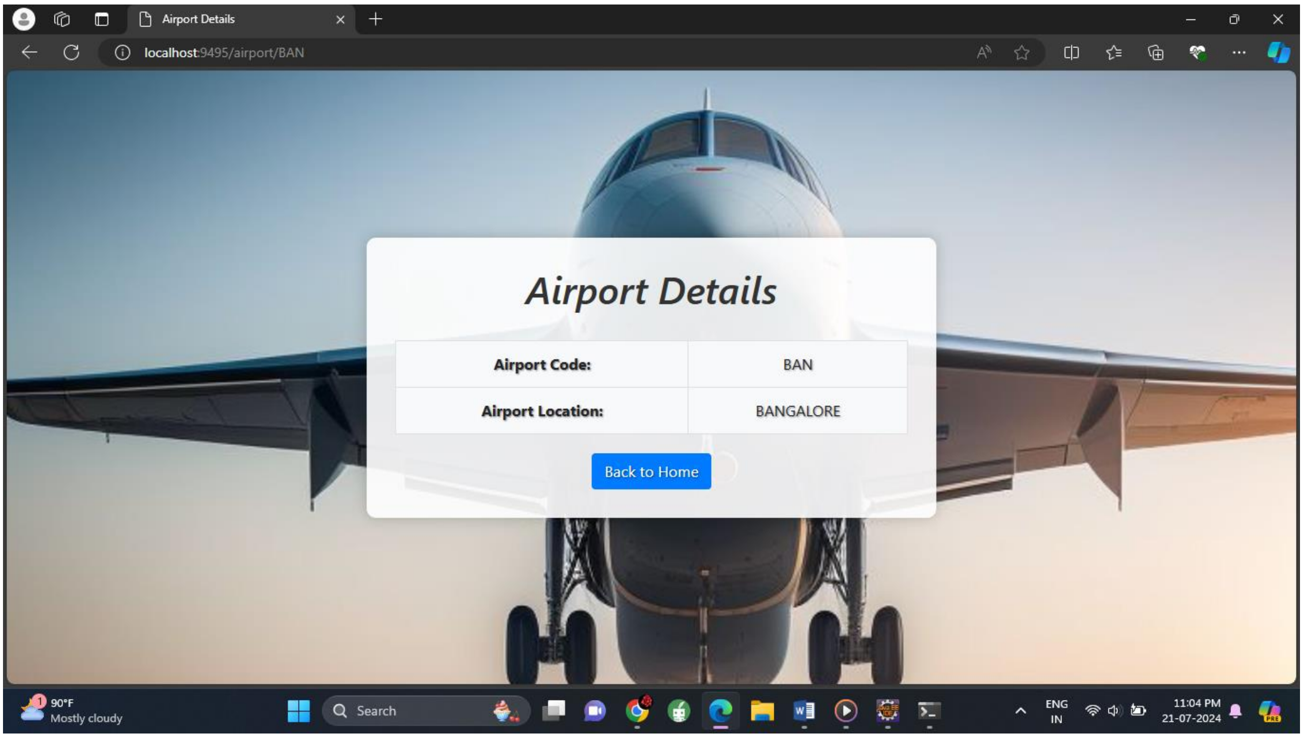
9.4.2 Airport Report



All Airports

Airport Code	Airport Location	Enquire
BAN	BANGALORE	Enquire
BMB	BOMBAY	Enquire
CHD	CHANDIGARH	Enquire
CHN	CHENNAI	Enquire
DEL	DELHI	Enquire
HYD	HYDERABAD	Enquire
LDH	LUDIANA	Enquire
MEE	MEERUT	Enquire
MOH	MOHALI	Enquire
MUM	MUMBAI	Enquire
PUNE	PUNE	Enquire

9.4.3 Airport Enquiry

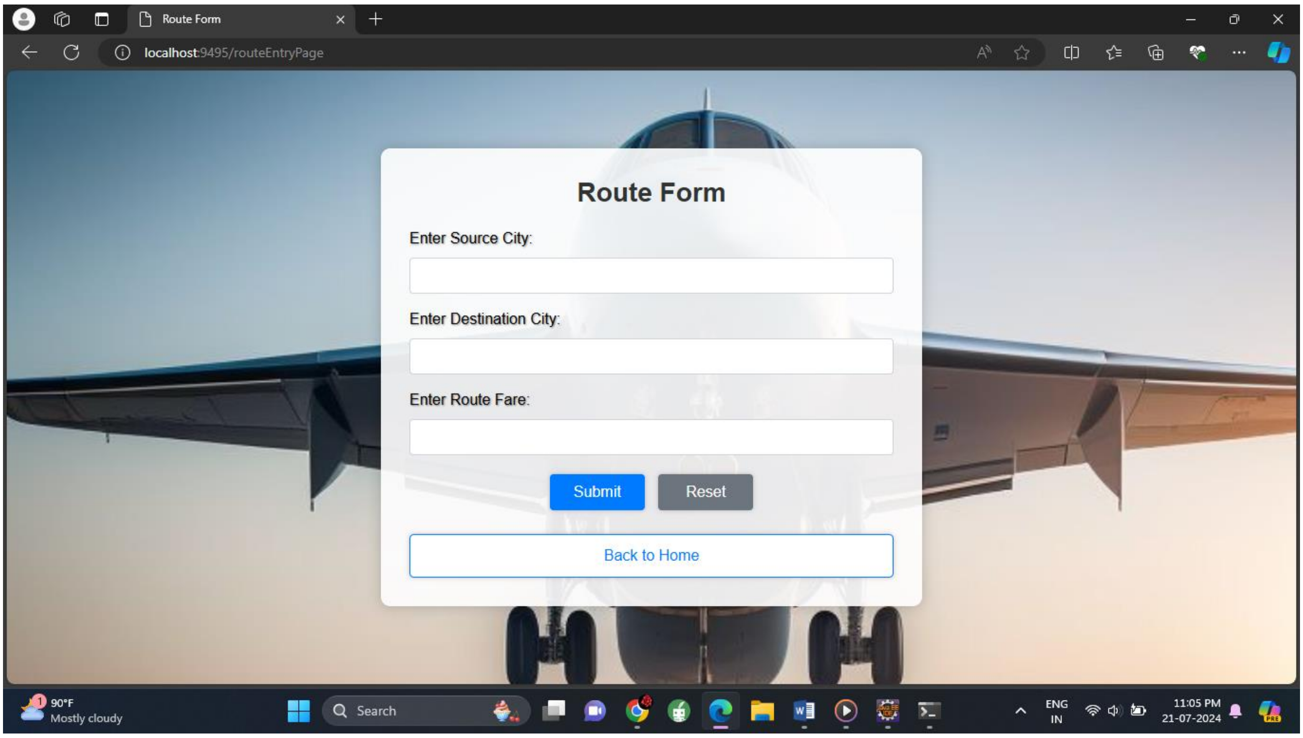


Airport Details

Airport Code:	BAN
Airport Location:	BANGALORE

9.5 Route

9.5.1 Route Addition



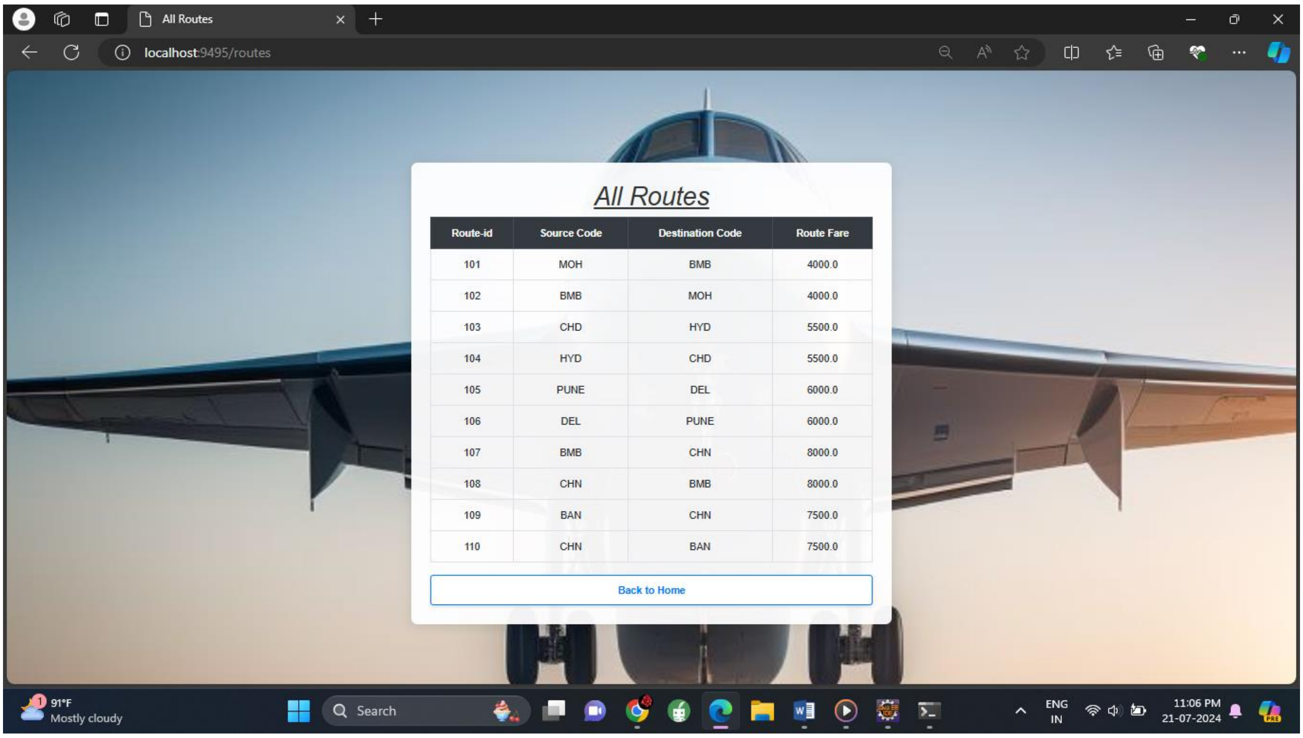
Route Form

Enter Source City:

Enter Destination City:

Enter Route Fare:

9.5.2 Route Report



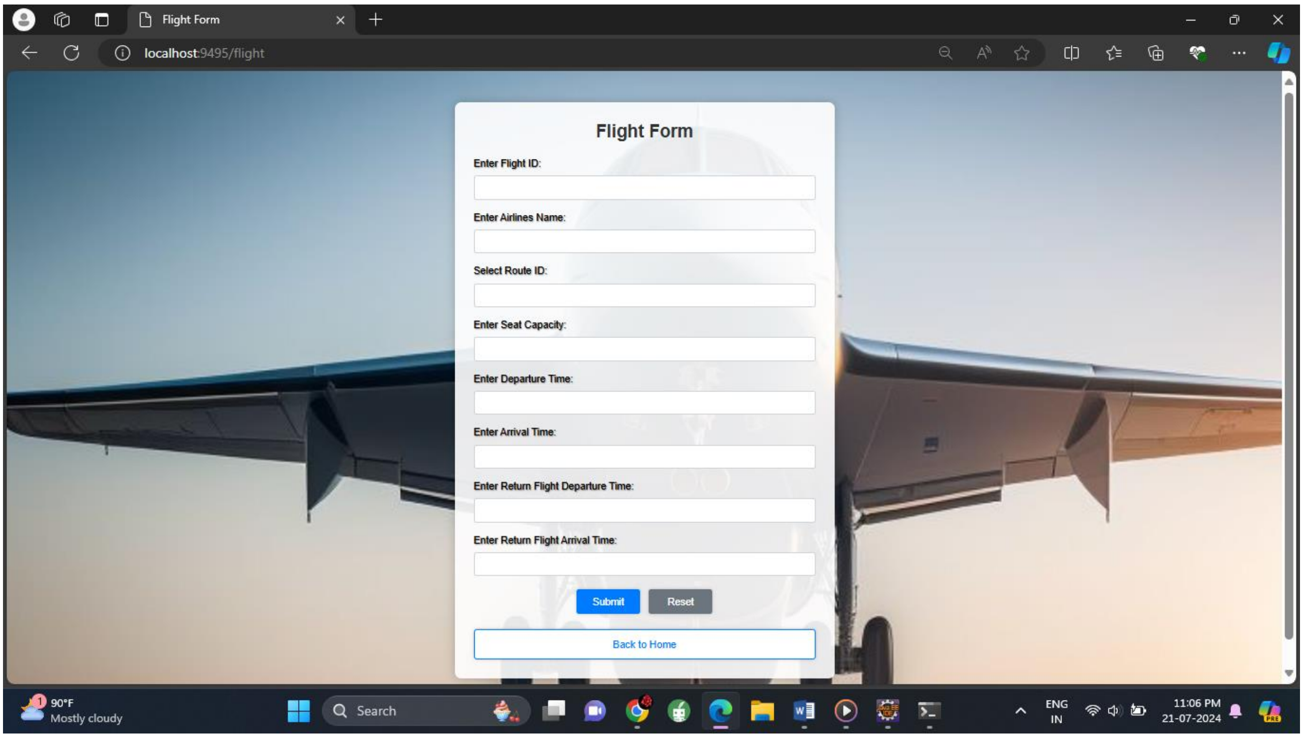
All Routes

Route-id	Source Code	Destination Code	Route Fare
101	MOH	BMB	4000.0
102	BMB	MOH	4000.0
103	CHD	HYD	5500.0
104	HYD	CHD	5500.0
105	PUNE	DEL	6000.0
106	DEL	PUNE	6000.0
107	BMB	CHN	8000.0
108	CHN	BMB	8000.0
109	BAN	CHN	7500.0
110	CHN	BAN	7500.0

Back to Home

9.6 Flight

9.6.1 Flight Addition



Flight Form

Enter Flight ID:

Enter Airlines Name:

Select Route ID:

Enter Seat Capacity:

Enter Departure Time:

Enter Arrival Time:

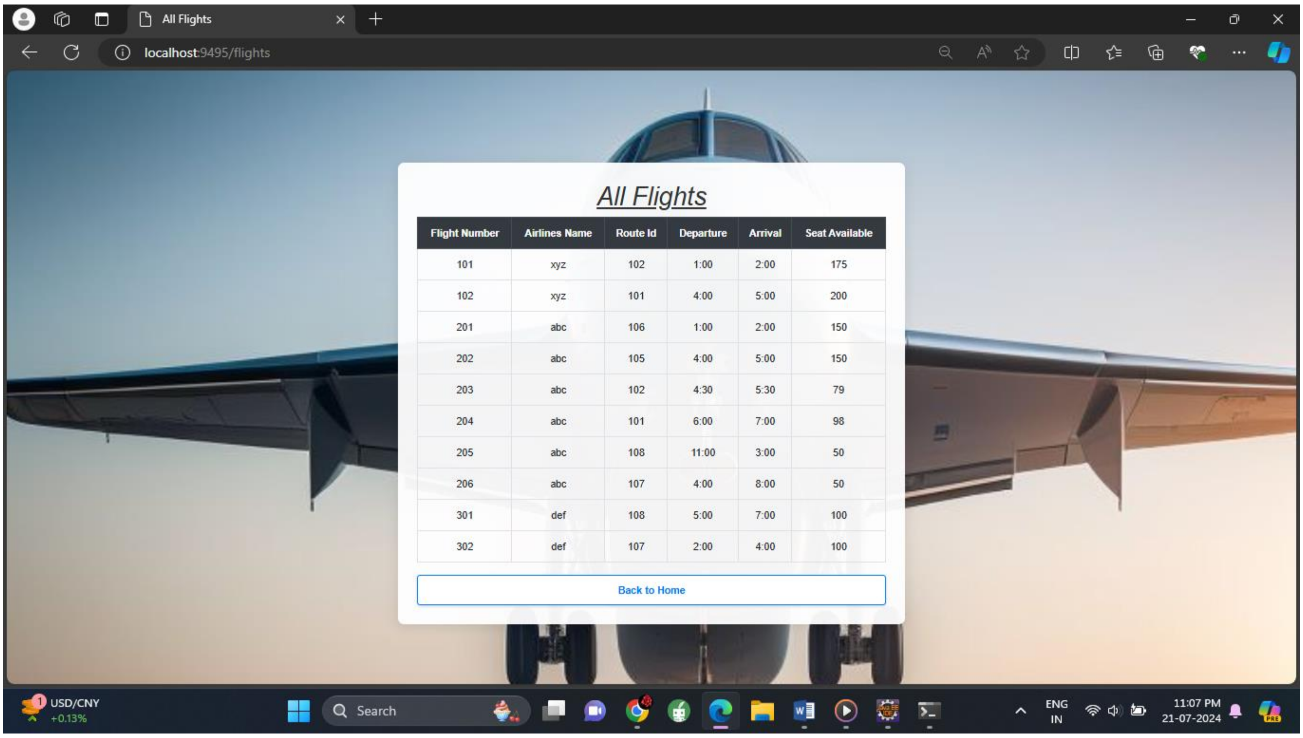
Enter Return Flight Departure Time:

Enter Return Flight Arrival Time:

Submit Reset

Back to Home

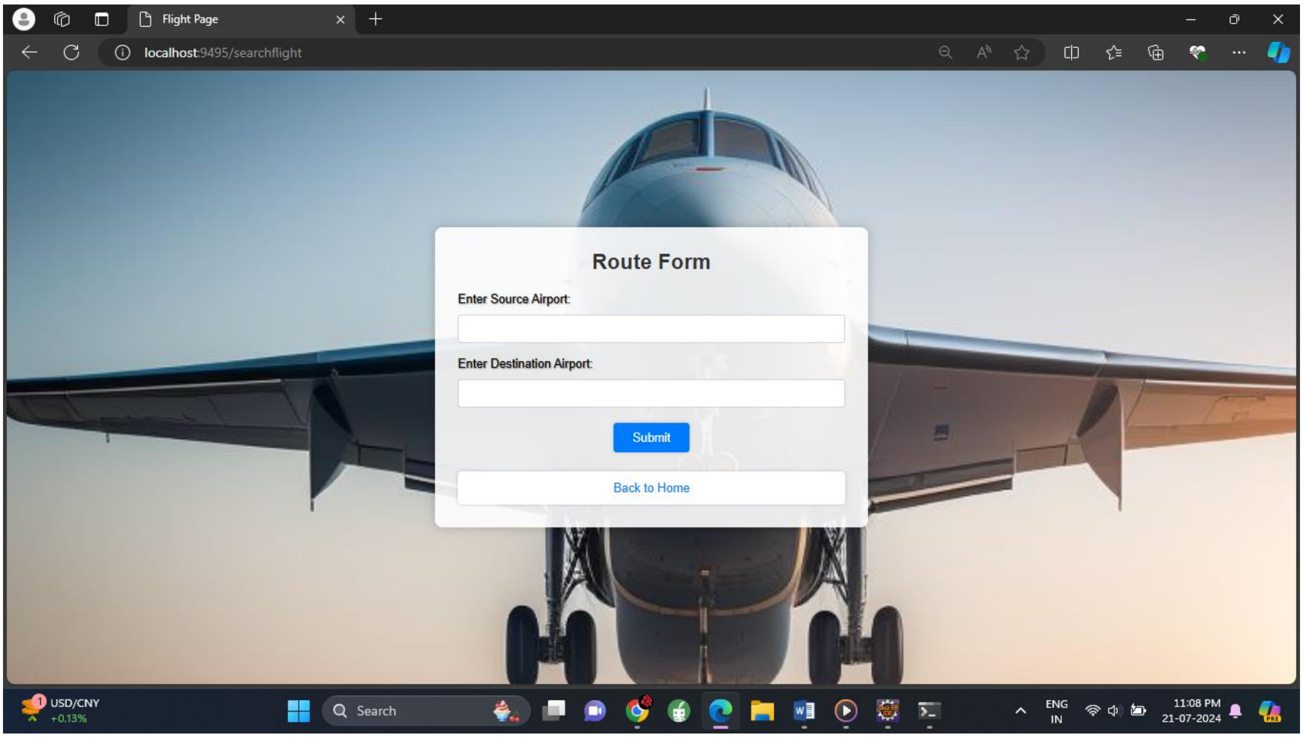
9.6.2 Flight Report



All Flights

Flight Number	Airlines Name	Route Id	Departure	Arrival	Seat Available
101	xyz	102	1:00	2:00	175
102	xyz	101	4:00	5:00	200
201	abc	106	1:00	2:00	150
202	abc	105	4:00	5:00	150
203	abc	102	4:30	5:30	79
204	abc	101	6:00	7:00	98
205	abc	108	11:00	3:00	50
206	abc	107	4:00	8:00	50
301	def	108	5:00	7:00	100
302	def	107	2:00	4:00	100

9.7 Flight Search Page

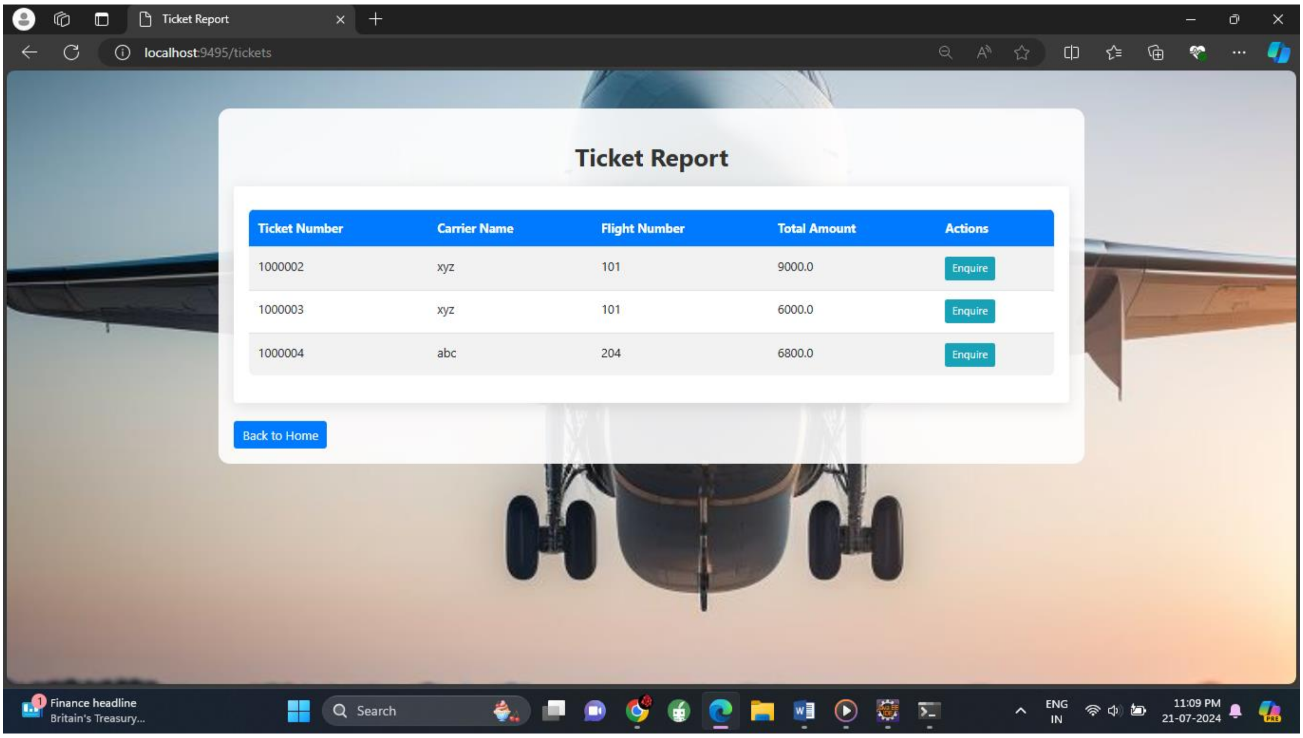


Route Form

Enter Source Airport:

Enter Destination Airport:

9.8 Ticket Report

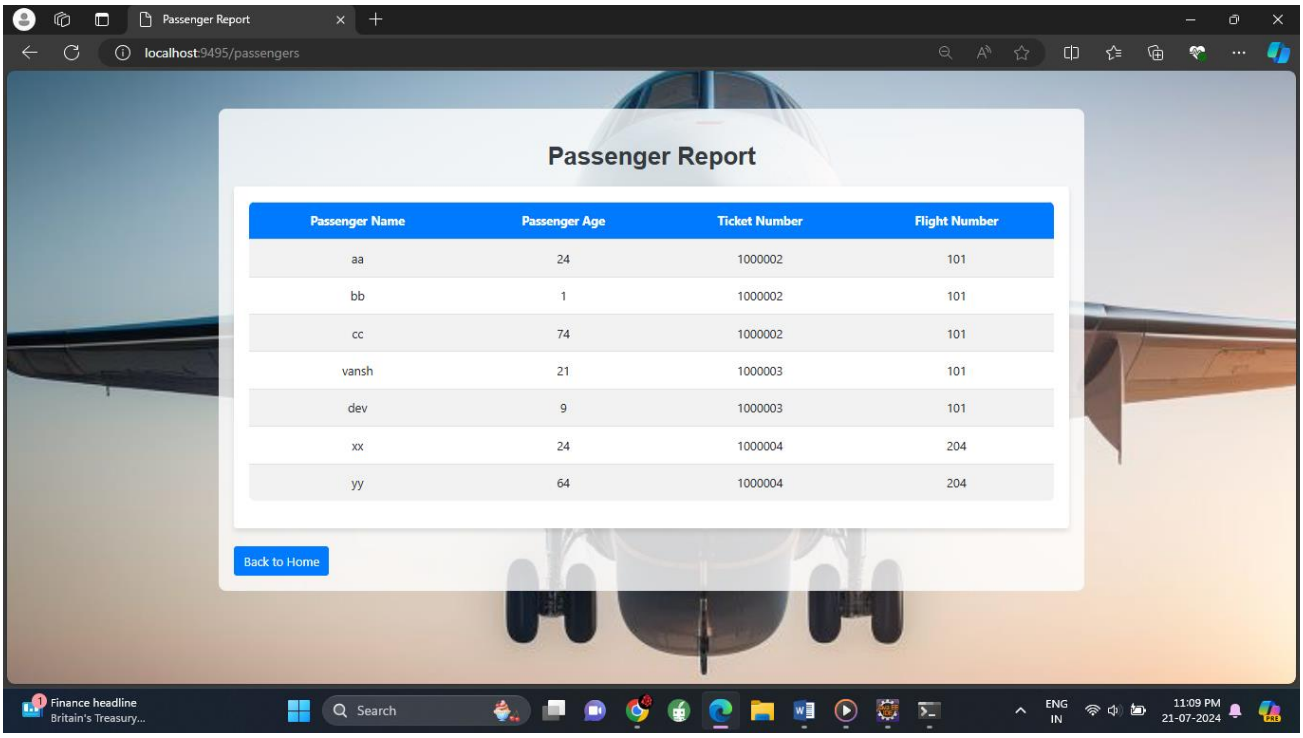


Ticket Report

Ticket Number	Carrier Name	Flight Number	Total Amount	Actions
1000002	xyz	101	9000.0	Enquire
1000003	xyz	101	6000.0	Enquire
1000004	abc	204	6800.0	Enquire

Back to Home

9.9 Passenger Report



Passenger Report

Passenger Name	Passenger Age	Ticket Number	Flight Number
aa	24	1000002	101
bb	1	1000002	101
cc	74	1000002	101
vansh	21	1000003	101
dev	9	1000003	101
xx	24	1000004	204
yy	64	1000004	204

Back to Home