**DOCUMENTATION REPORT**

**On**

# FLIGHT MANAGEMENT SYSTEM



Submitted to: Submitted By:

Mr. Suramya Biswas 1. Khushmn Sangha

2. Vanshika Rani

3. Soumik Saha

4. Vijay Yadav

**CONTENT**

1. **Introduction**

## Key Features

# 1.2 Technology Stack

1.3 Benefits

1.4 Customer Interface

1.5 Admin Interface

1. **Technology Used**
   1. Backend
   2. Frontend
   3. Database
   4. Build Tool
   5. Server
   6. Version Control
2. **Project Structure**
3. **Flight Management System - Setup Instructions**
   1. Clone the Repository
   2. Setup Database
      1. Create the Database
      2. Update Application Properties
   3. Build the Project
   4. Run the Application
   5. Access the Application
   6. Explore the Application

4.6.1 Customer Interface

4.6.2 Admin Interface

**5. Key Functionalities of the Flight Management System**

5.1 User Management

5.1.1 Customer Registration and Login

5.1.2 Admin Login

5.2 Airport Management

5.2.1 View Airport

5.2.2 Add Airport

5.3 Route Management

5.3.1 View Route

5.3.2 Add Route

5.4 Flight Management

5.4.1 View Flight

5.4.2 Add Flight

5.5 Ticket Management

5.5.1 Book Ticket

5.5.2 View Bookings

5.5.3 Cancel Tickets

5.5.4 View Ticket Details

5.6 Passenger Management

5.6.1 View Passenger Details

5.7 Reporting

5.7.1 Generate Reports

5.8 Security

5.8.1 Authentication and Authorization

5.8.2 Data Encryption

**6. Entity Relationship Diagram**

**7. System Workflow Diagram: Flight Management**

**8. Database Schema: Flight Management System**

**9. Flight Booking and Management System: Output Screens**

9.1 Login Page

9.2 Register New User

9.3 Index Page

9.3.1 Admin Index

9.3.2 Customer Index

9.4 Airport

9.4.1 Airport Addition

9.4.2 Airport Report

9.4.3 Airport Enquiry

9.5 Route

9.5.1 Route Addition

9.5.2 Route Report

9.6 Flight

9.6.1 Flight Addition

9.6.2 Flight Report

9.7 Flight Search Page

9.8 Ticket Report

9.8.1 Ticket Booking

9.8.2 Ticket Report

9.8.3 Ticket Enquiry

9.9 Passenger Report

**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:**

aeroBlasters/

├── src/main/java/com/aeroblasters/

│ ├── controller/

│ ├── model/

│ ├── repository/

│ ├── service/

│ ├── AeroblastersApplication.java

├── src/main/resources/

│ ├── static/

│ ├── templates/

│ ├── application.properties

├── src/main/webapp/WEB-INF/

│ ├── views/

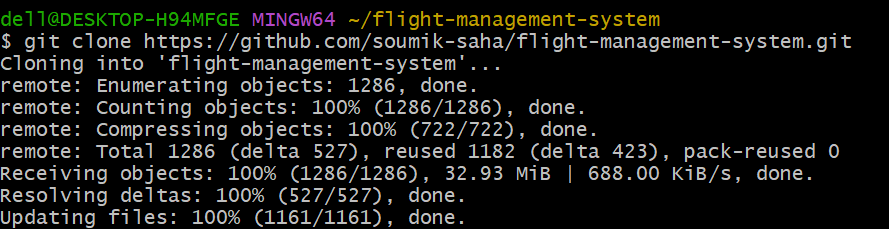
│ ├── web.xml

├── pom.xml

├── README.md

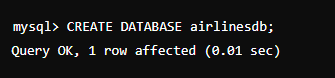
**4.** **Flight Management System - Setup Instructions:** Follow these steps to set up and run the Flight Management System project on local machine.

**4.1 Clone the Repository:** First, clone the repository to your local machine and navigate into the project directory.

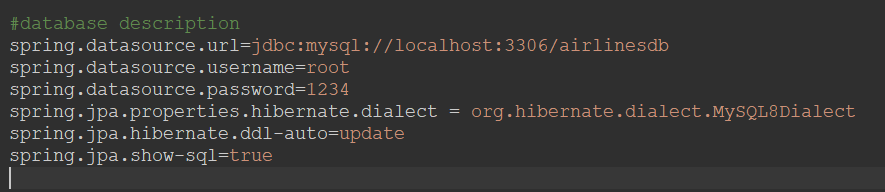


**4.2 Setup the Database:** Install MySQL and create a database named `**airlinesdb**`. Update the database configuration in `src/main/resources/application.properties` with MySQL credentials.

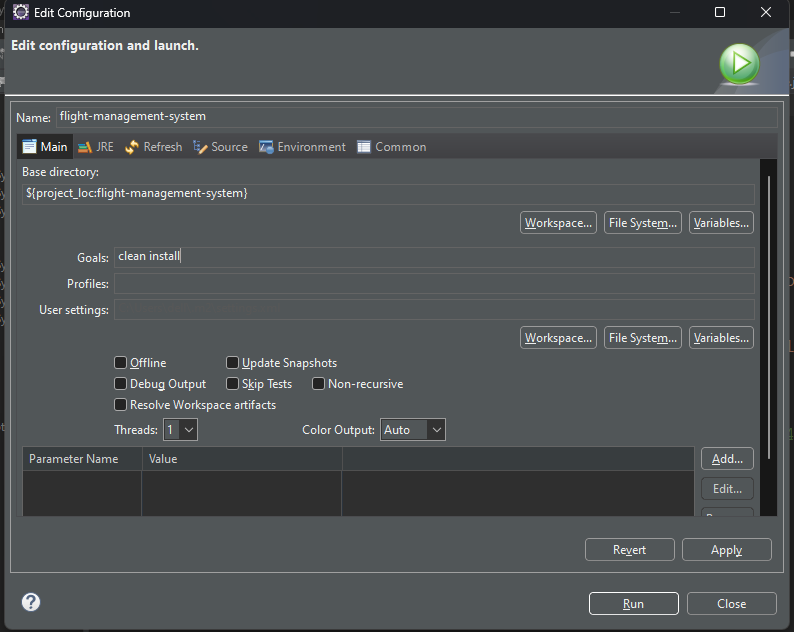
**4.2.1 Create the Database:**



**4.2.2 Update application.properties :**

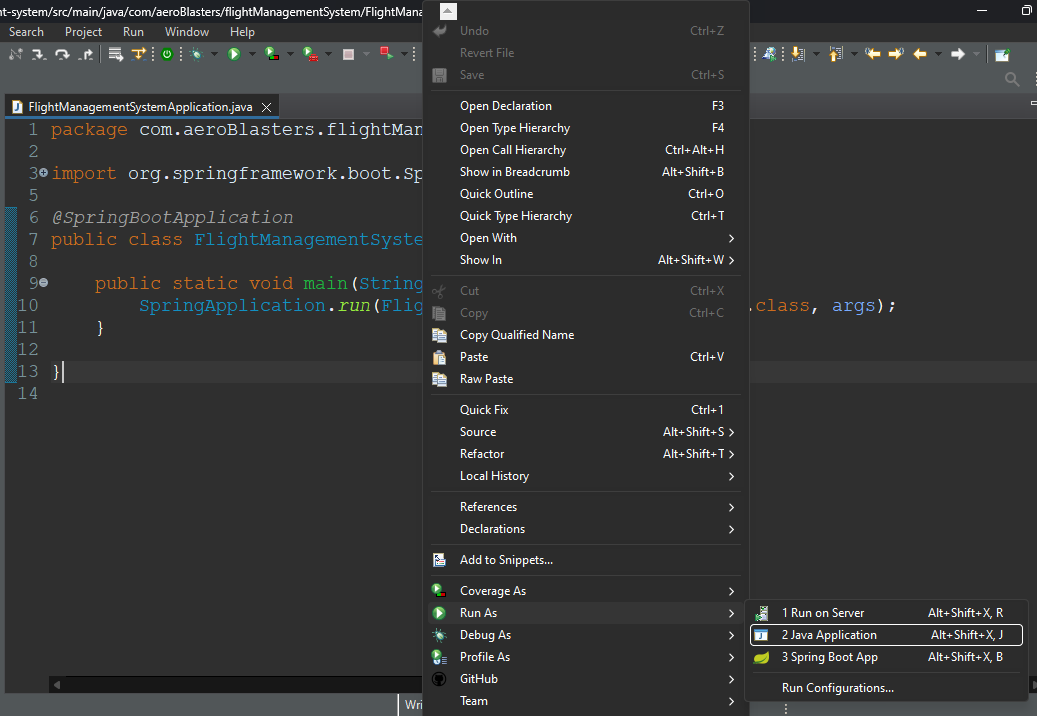


**4.3 Build the Project:** Use Maven to clean and build the project. This will compile the code, run tests, and package the application.

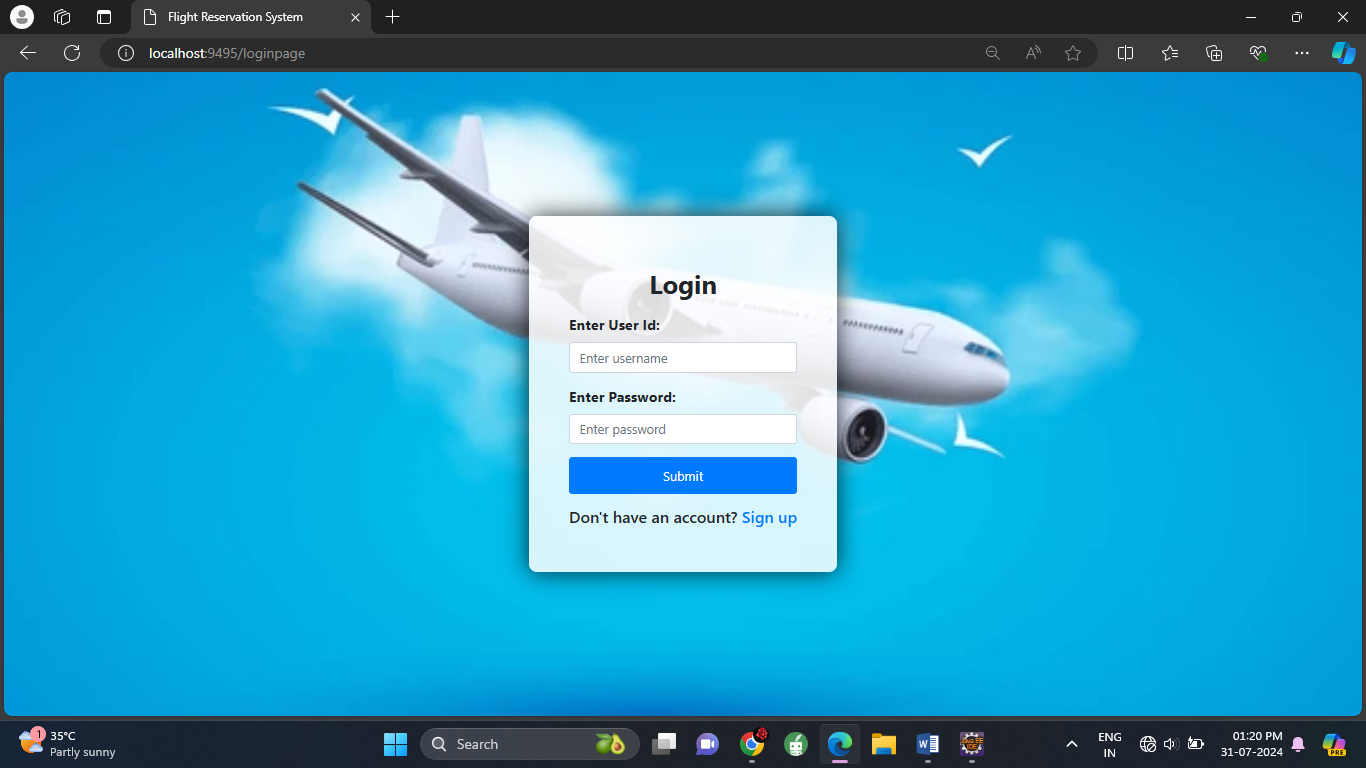


**4.4 Run the Application:** Start the Spring Boot application using your IDE. Here’s how to do it in Eclipse:

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



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



**4.6 Explore the Application**

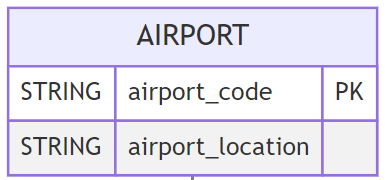
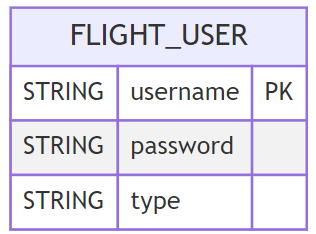
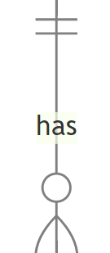
**4.6.1** **Customer Interface:** Customers can view airports, routes, and flights, book tickets, and cancel tickets.

**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:

* 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.
  1. **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.
  1. **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.
  1. **Flight Management –**
* **View Flights:** Both customers and admins can view all scheduled flights.
* **Add Flights:** Admins have the ability to add new flights.
  1. **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.
  1. **Passenger Management –**
* **View Passenger Details:** Admins can view detailed information about passengers for each flight.
  1. **Reporting –**
* **Generate Reports:** Admins can generate reports on airports, flights, routes, bookings, and passengers to gain insights and make informed decisions.
  1. **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.

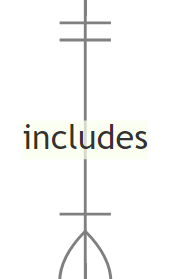
**6. Entity Relationship Diagram –**



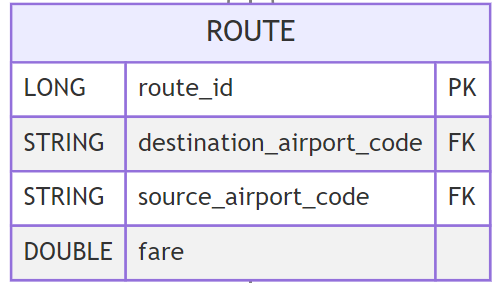


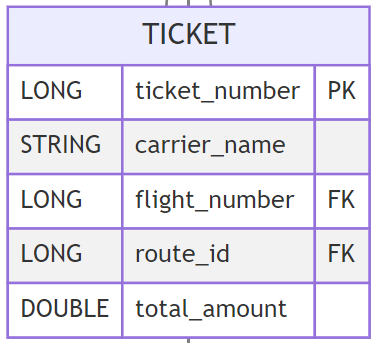
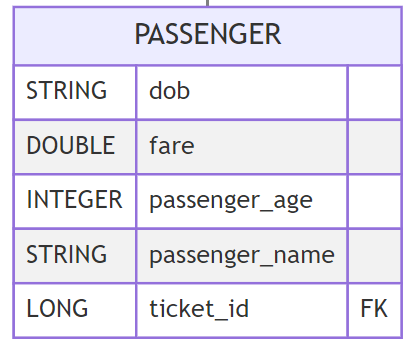
9



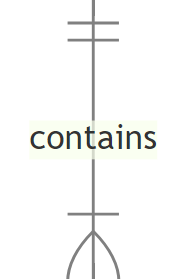














**7.** **System Workflow Diagram: Flight Management**

**FLIGHT MANAGEMENET SYSTEM**

Book/Cancel Flight

View All Passengers

View All Tickets

Manage Flight

Manage Routes

Manage Airports

Login

View List (Airport, Route, Flight)

Search Flight

Book/Cancel Flight

Manage Profile

Login

Register

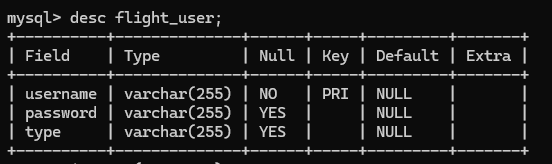
ADMIN

CUSTOMER

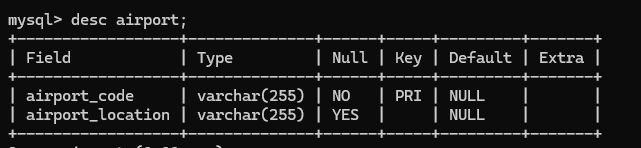
Register

**8.** **Database Schema: Flight Management System:**

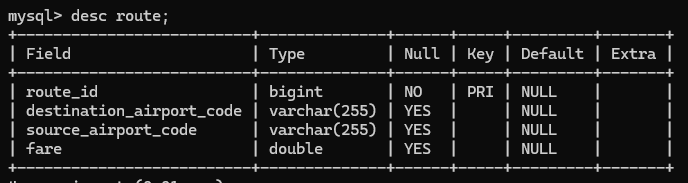
* 1. Flight\_user

****

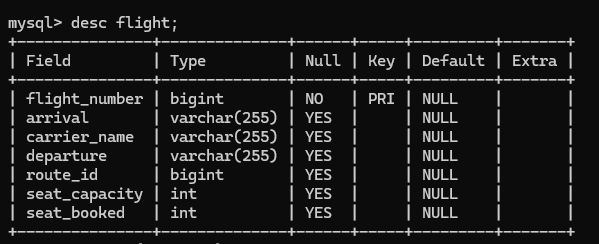
* 1. Airport



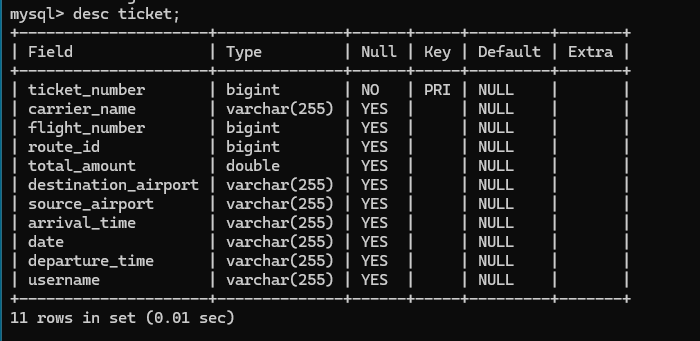
* 1. Route



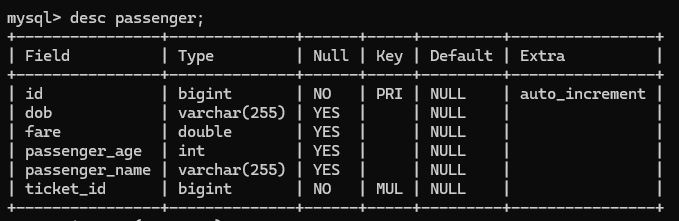
* 1. Flight



* 1. Ticket

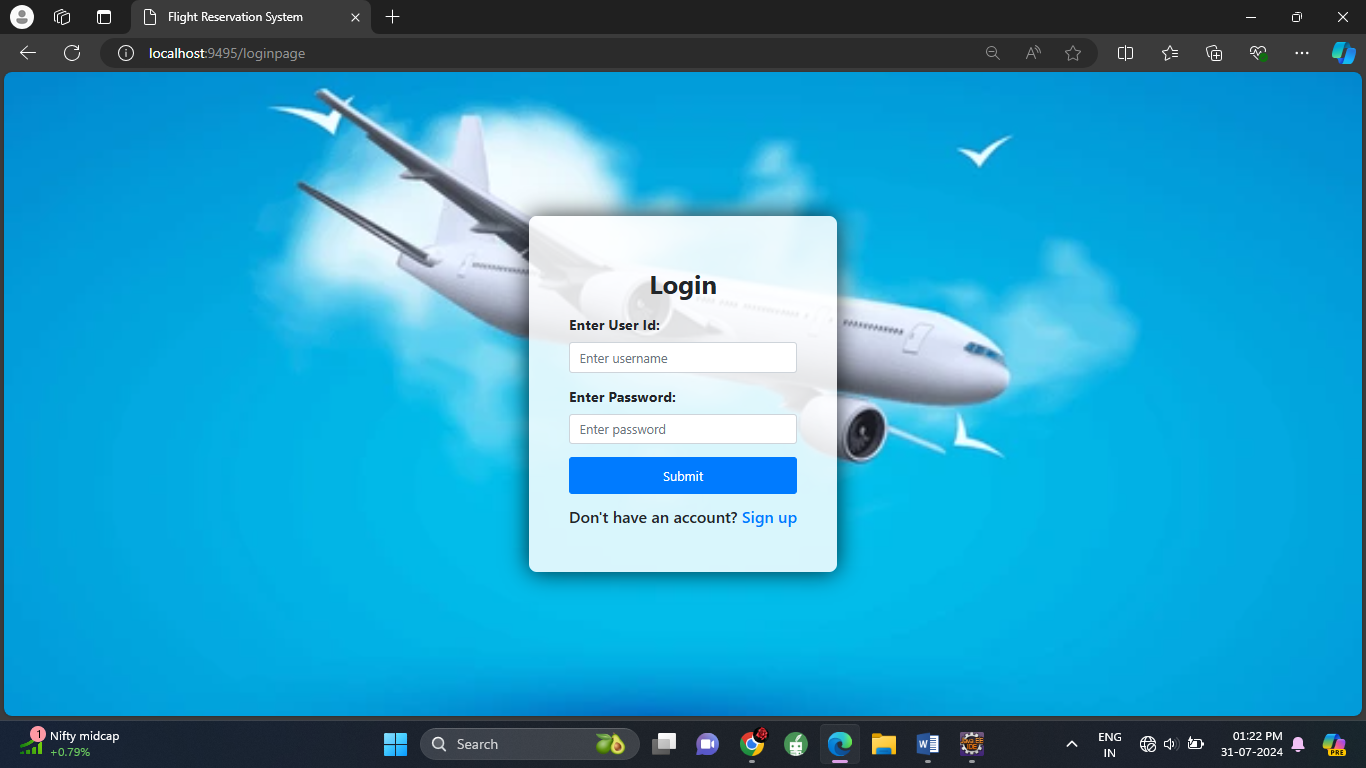


* 1. Passenger

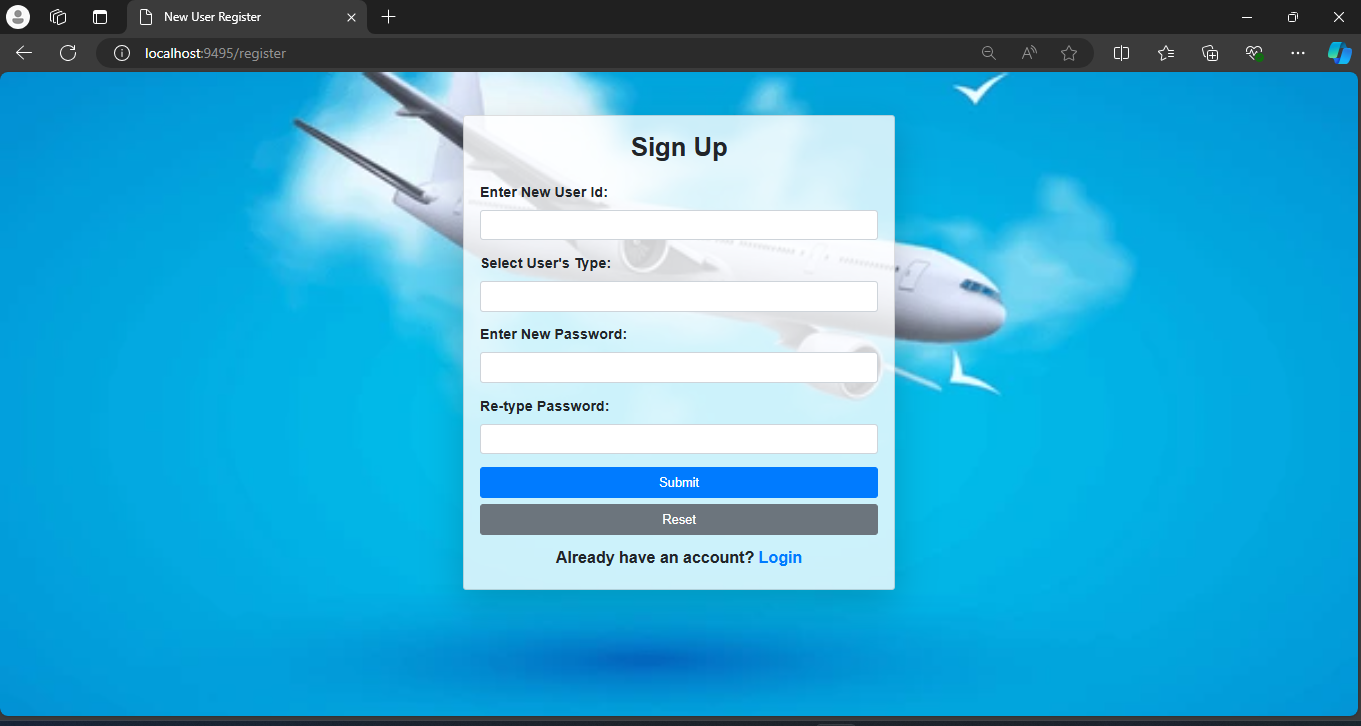


**9. Flight Booking and Management System: Output Screens:**

9.1 Login Page

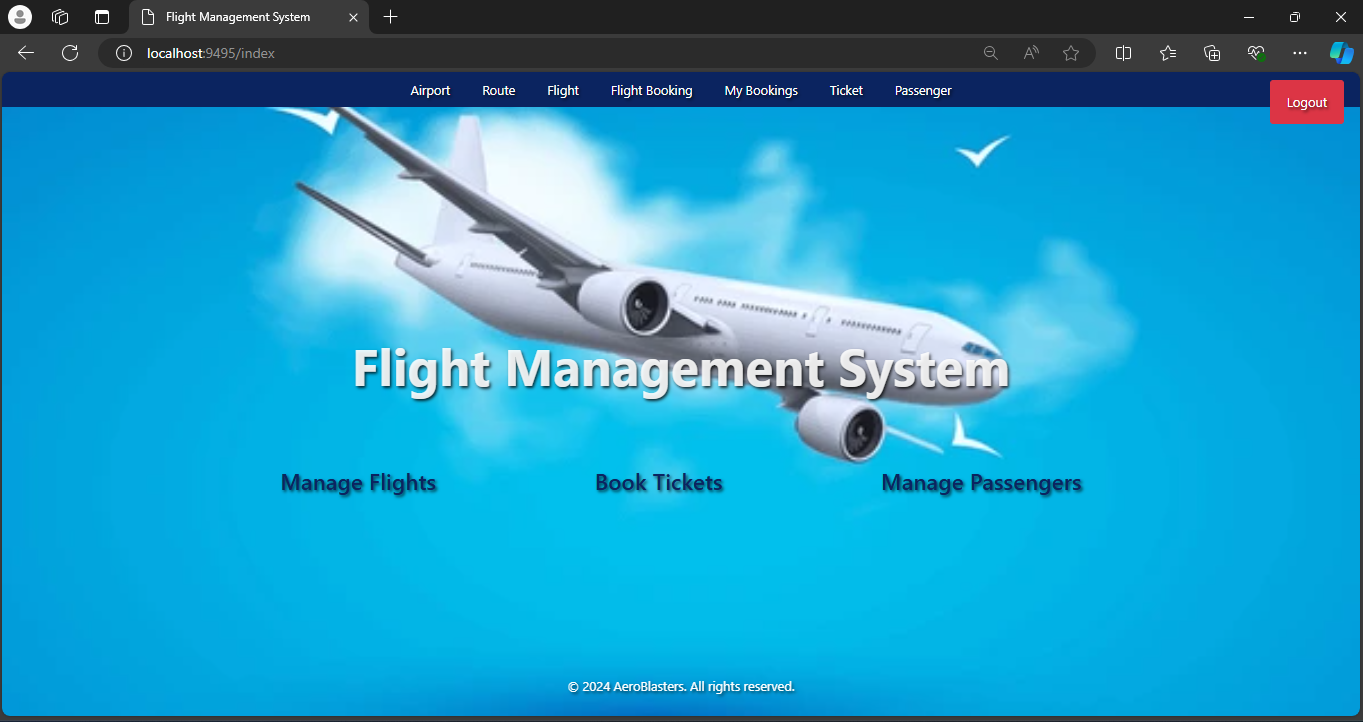
****

9.2 Register New User

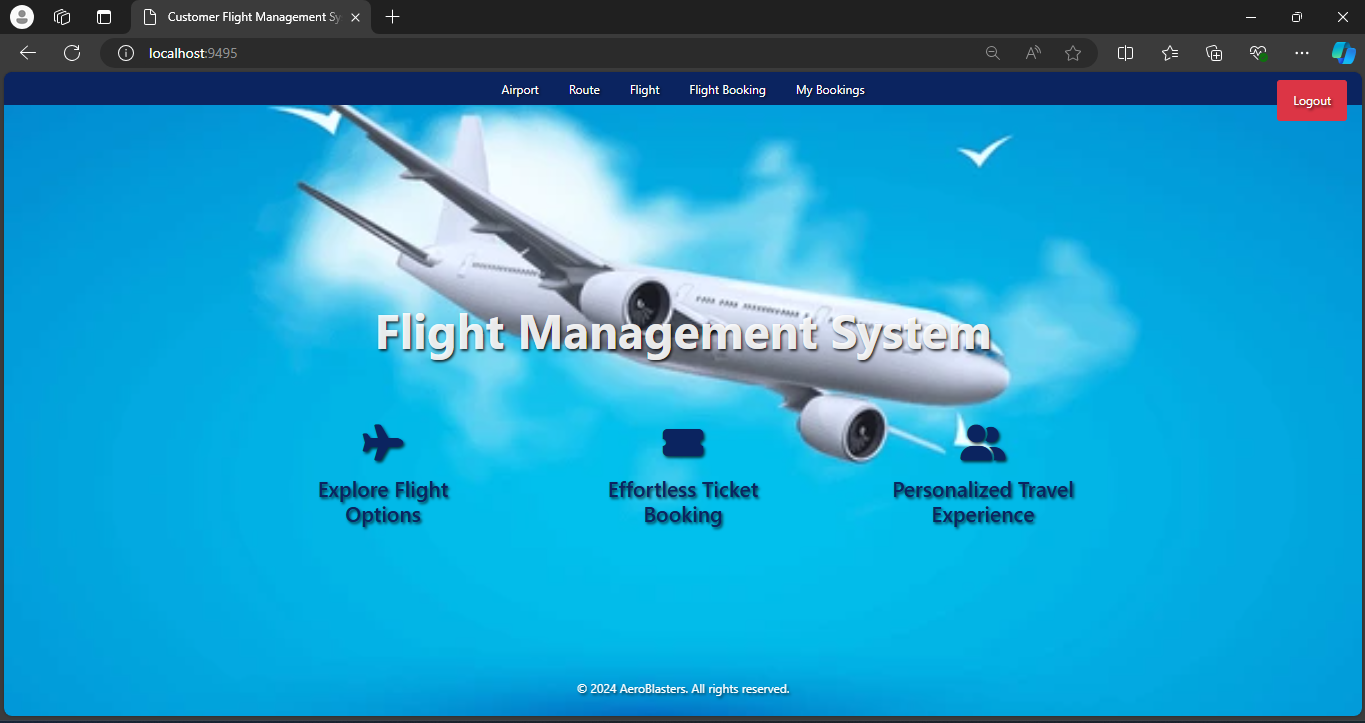


9.3 Index Page

9.3.1 Admin Index

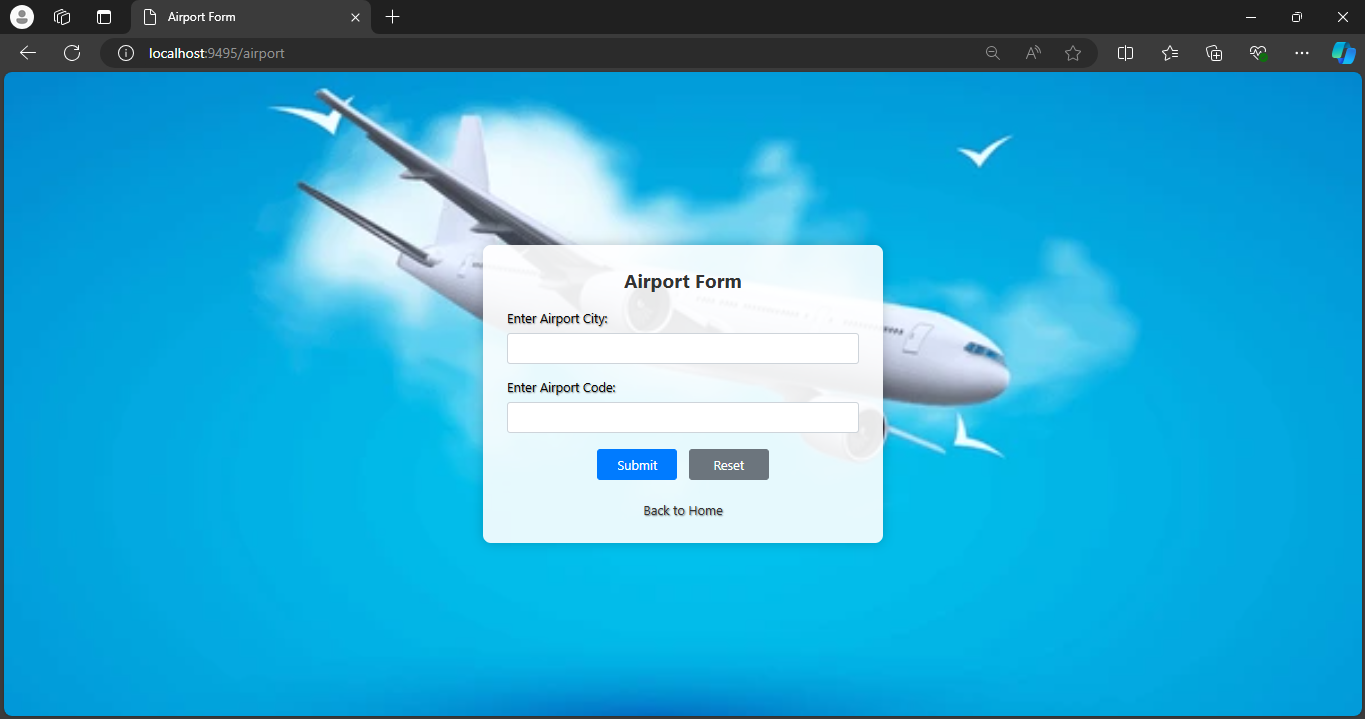


9.3.2 Customer Index

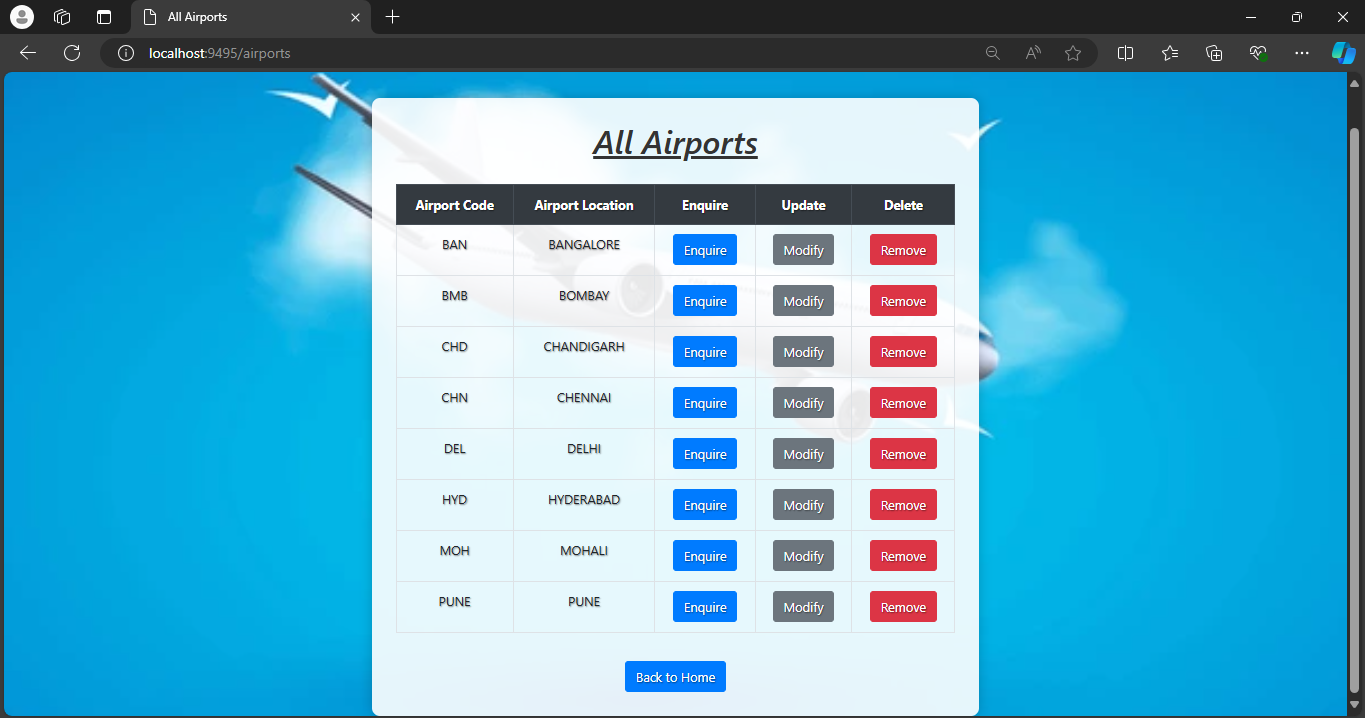


9.4 Airport

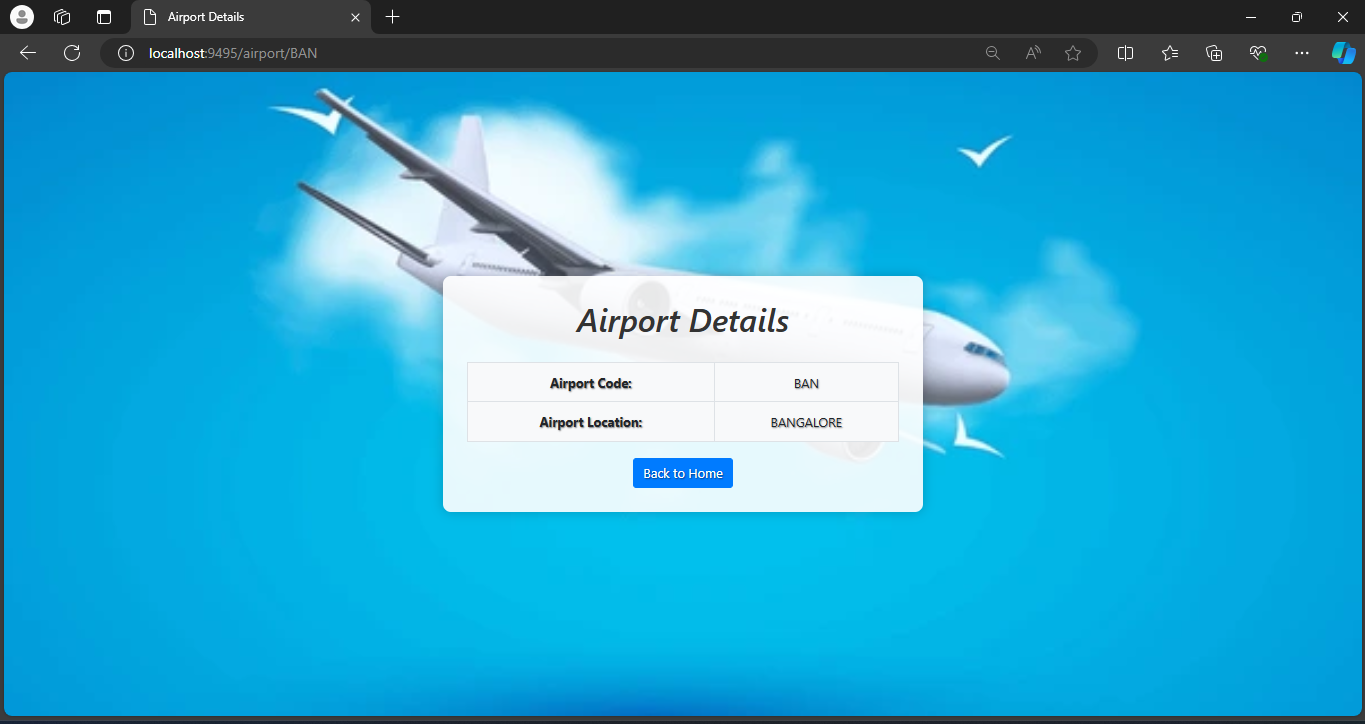
9.4.1 Airport Addition



9.4.2 Airport Report

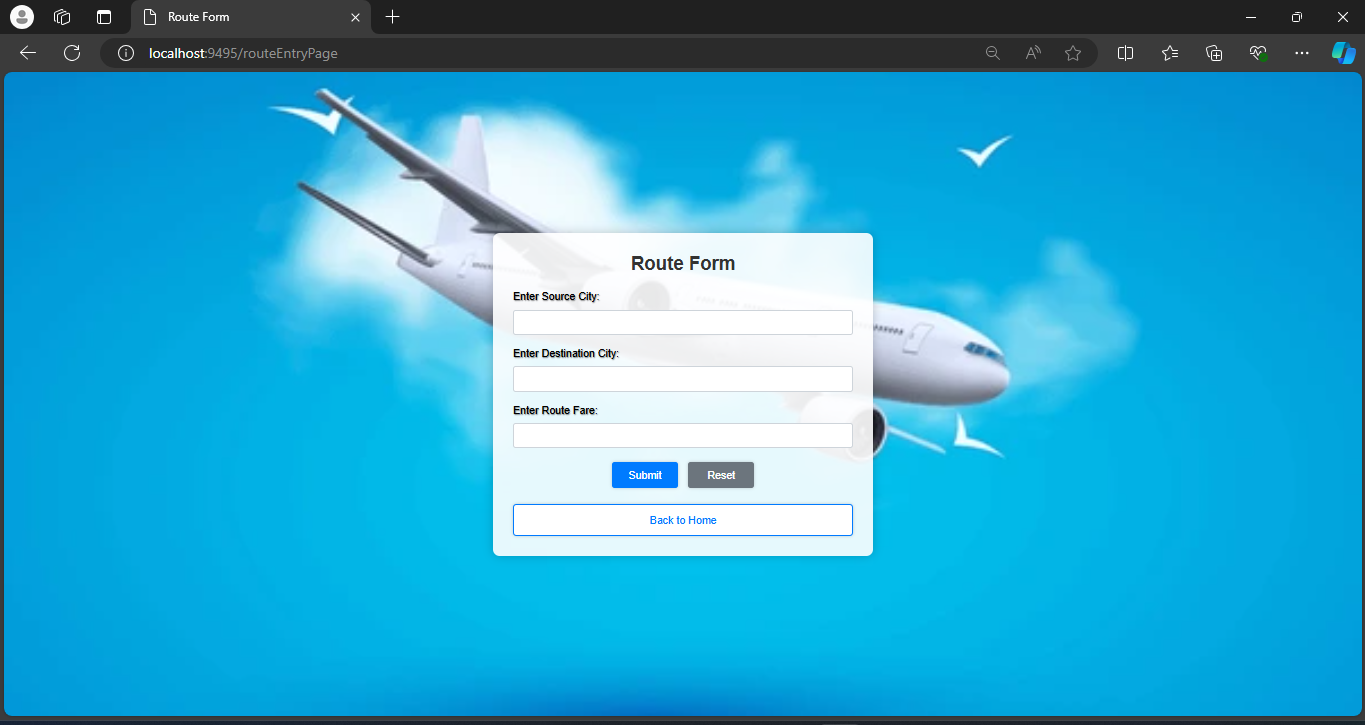


9.4.3 Airport Enquiry

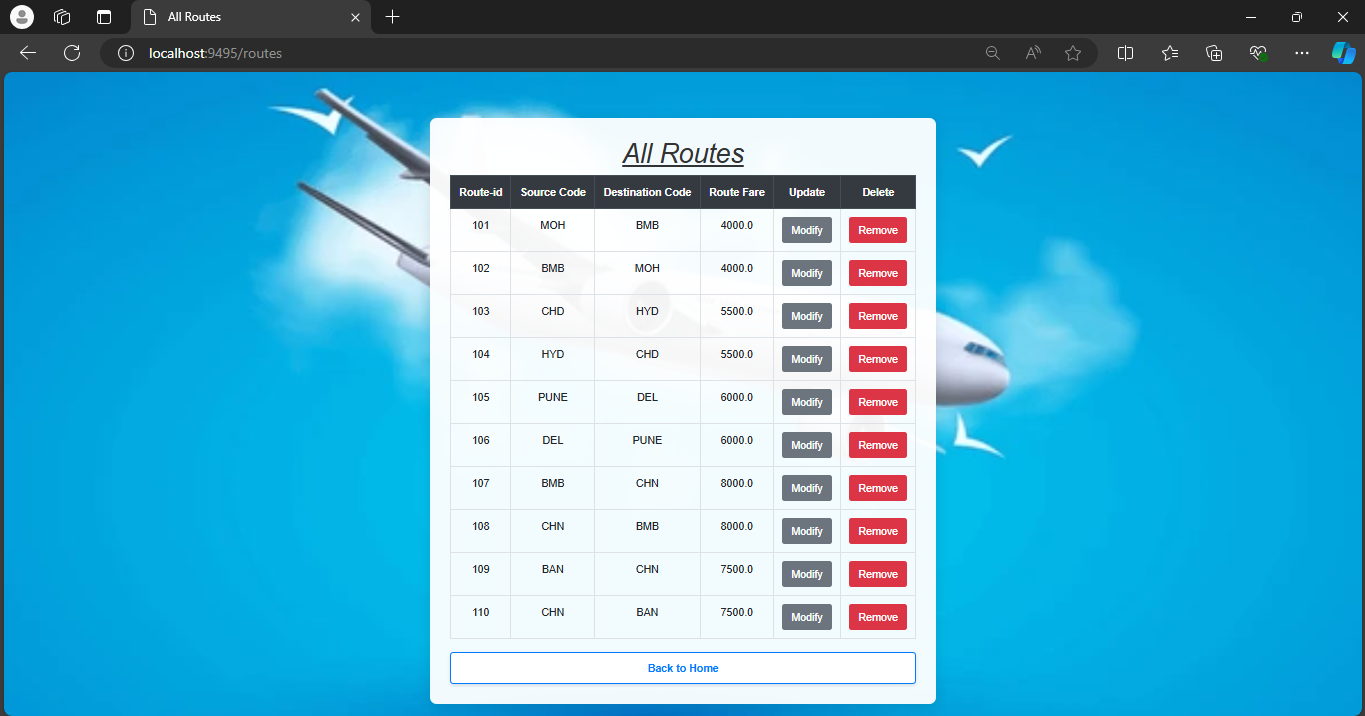


9.5 Route

9.5.1 Route Addition

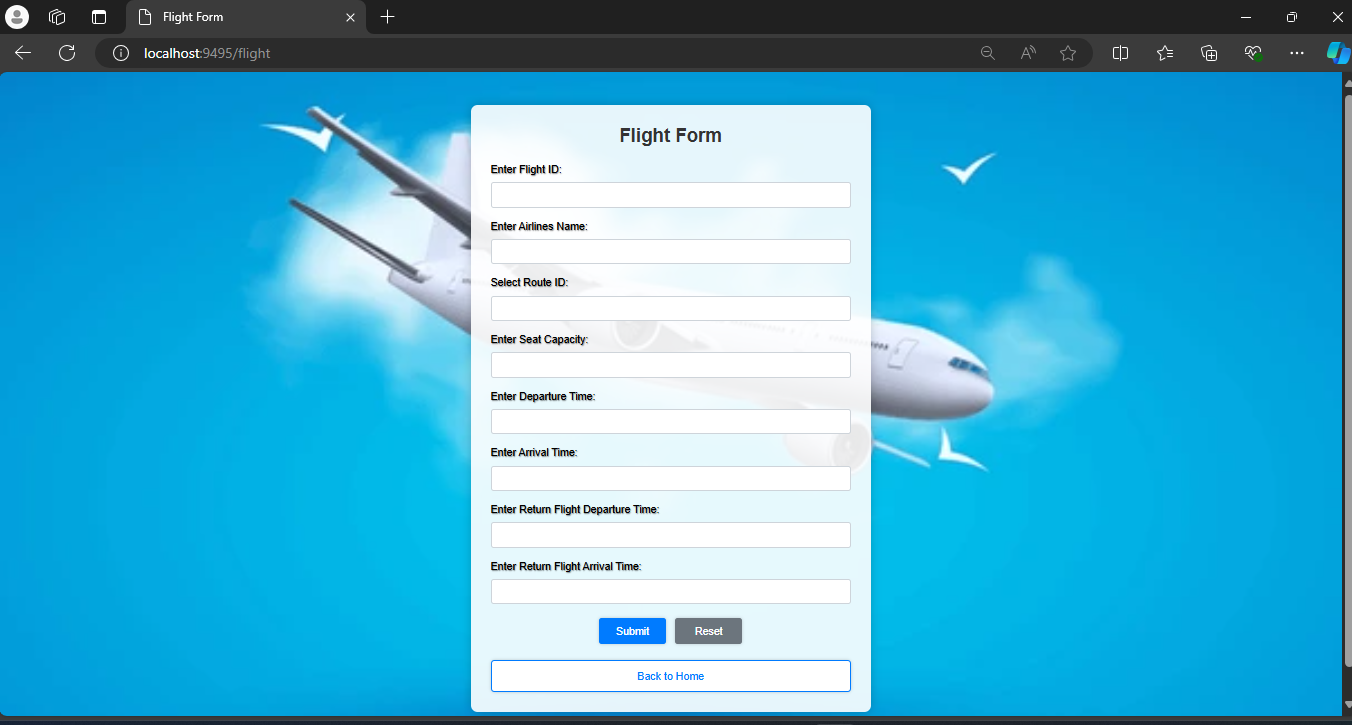


9.5.2 Route Report

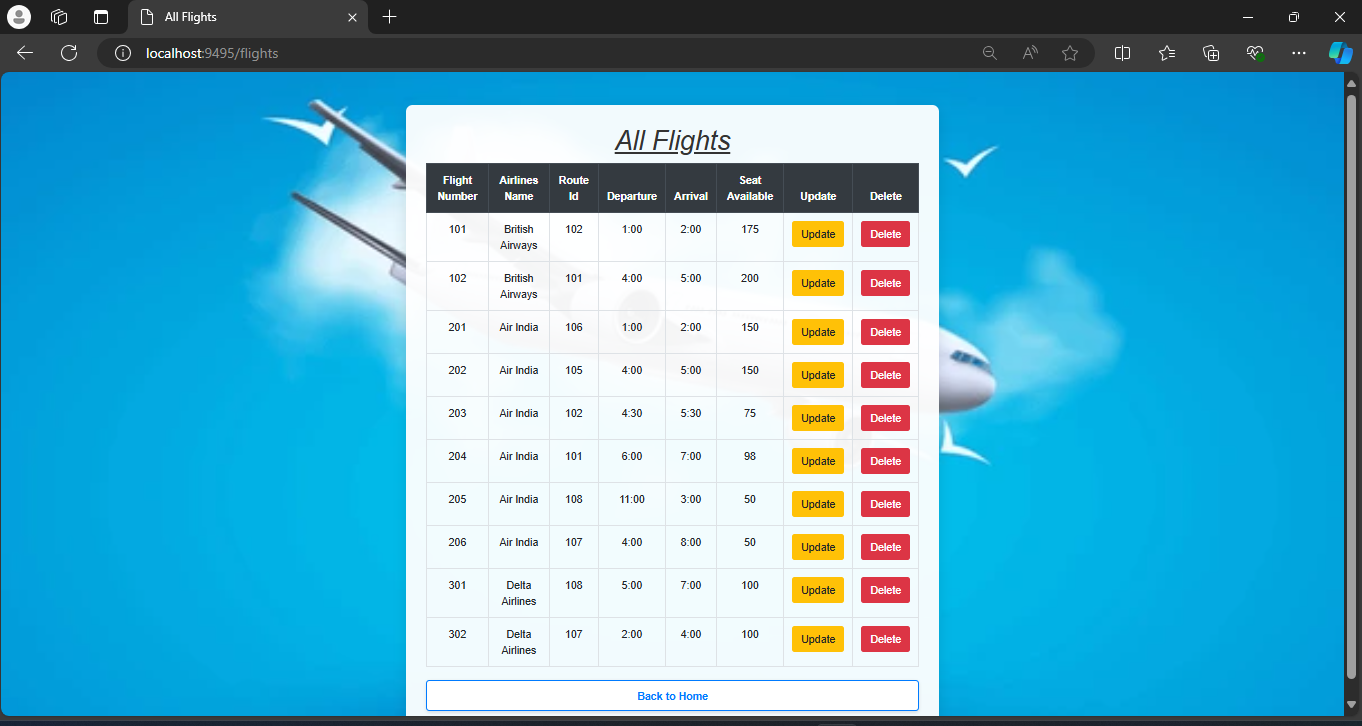


9.6 Flight

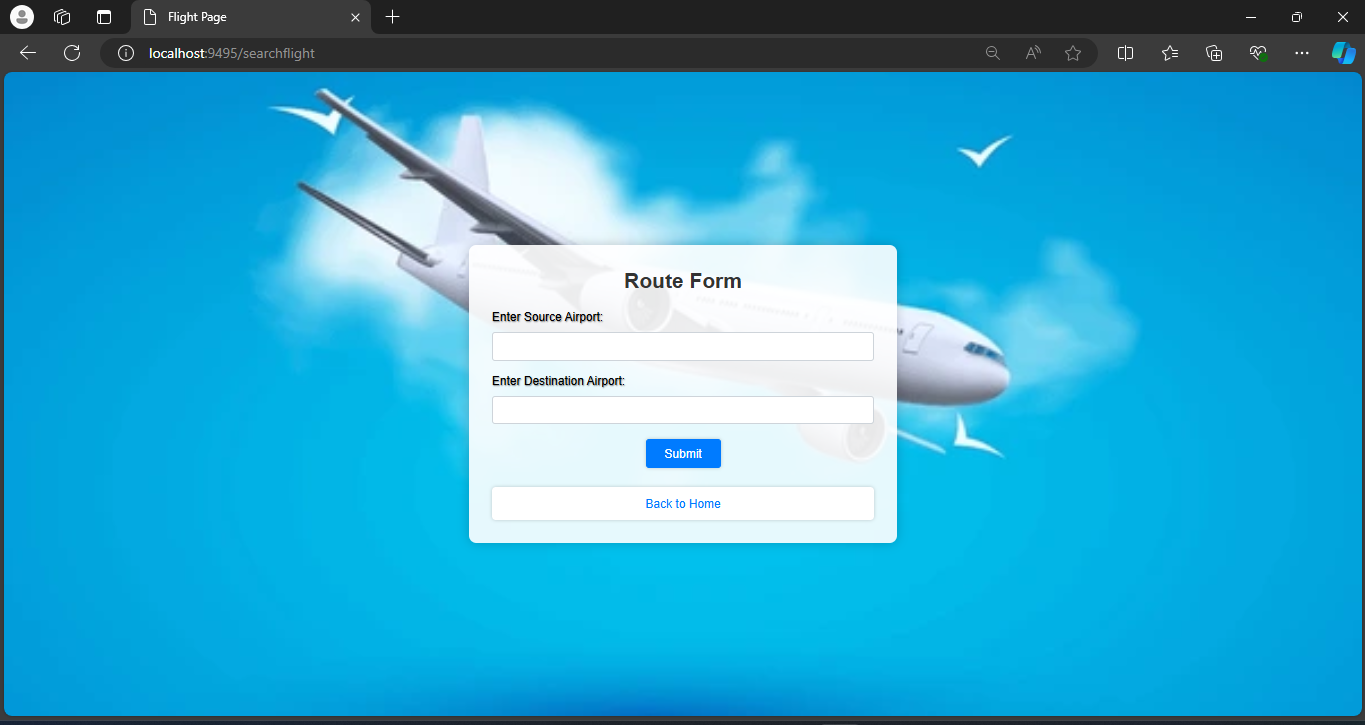
9.6.1 Flight Addition



9.6.2 Flight Report

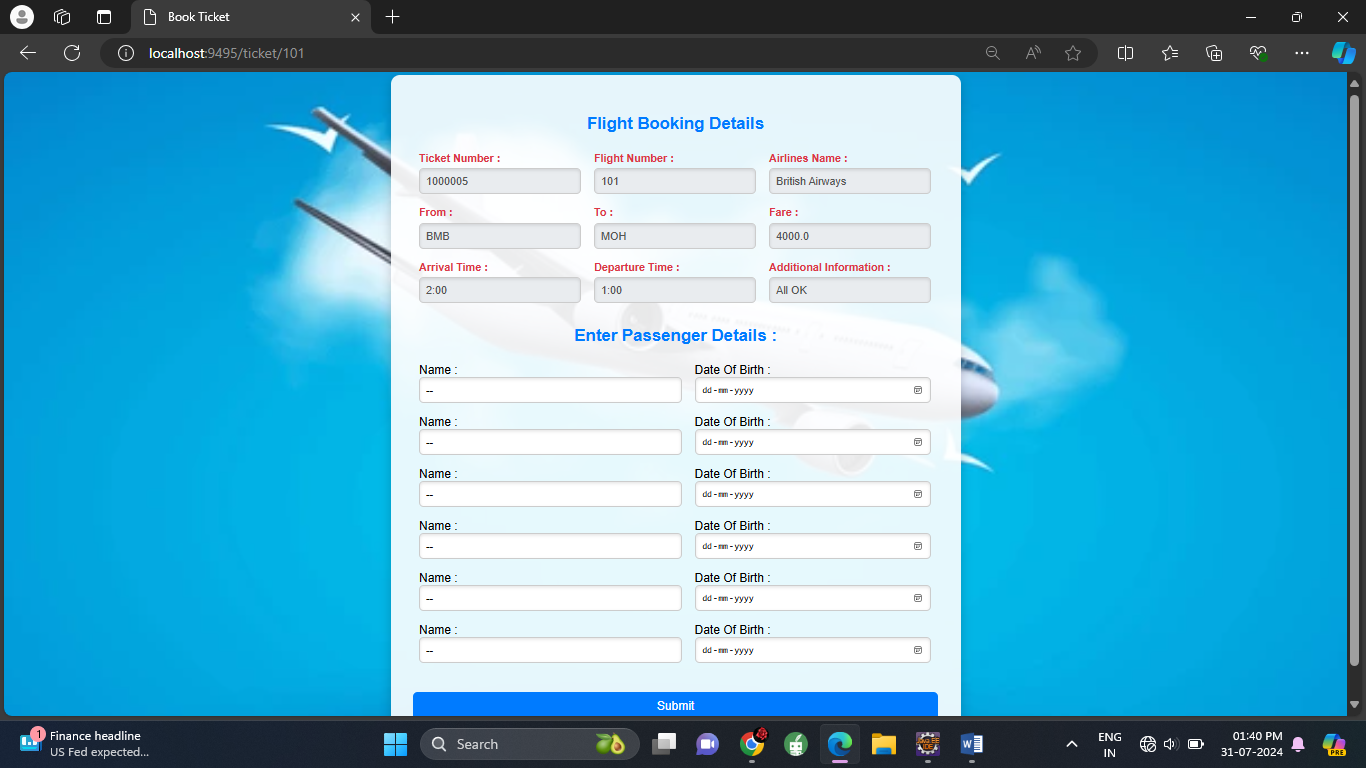


9.7 Flight Search Page

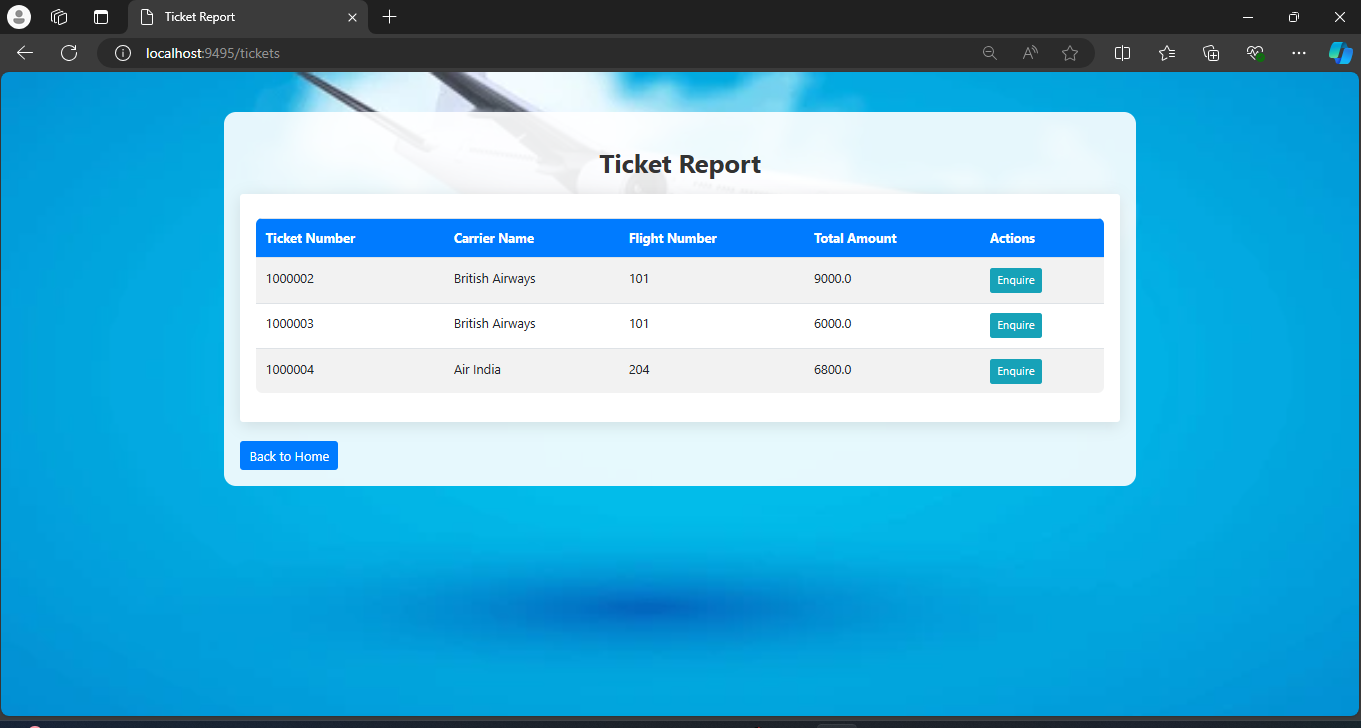


9.8 Ticket

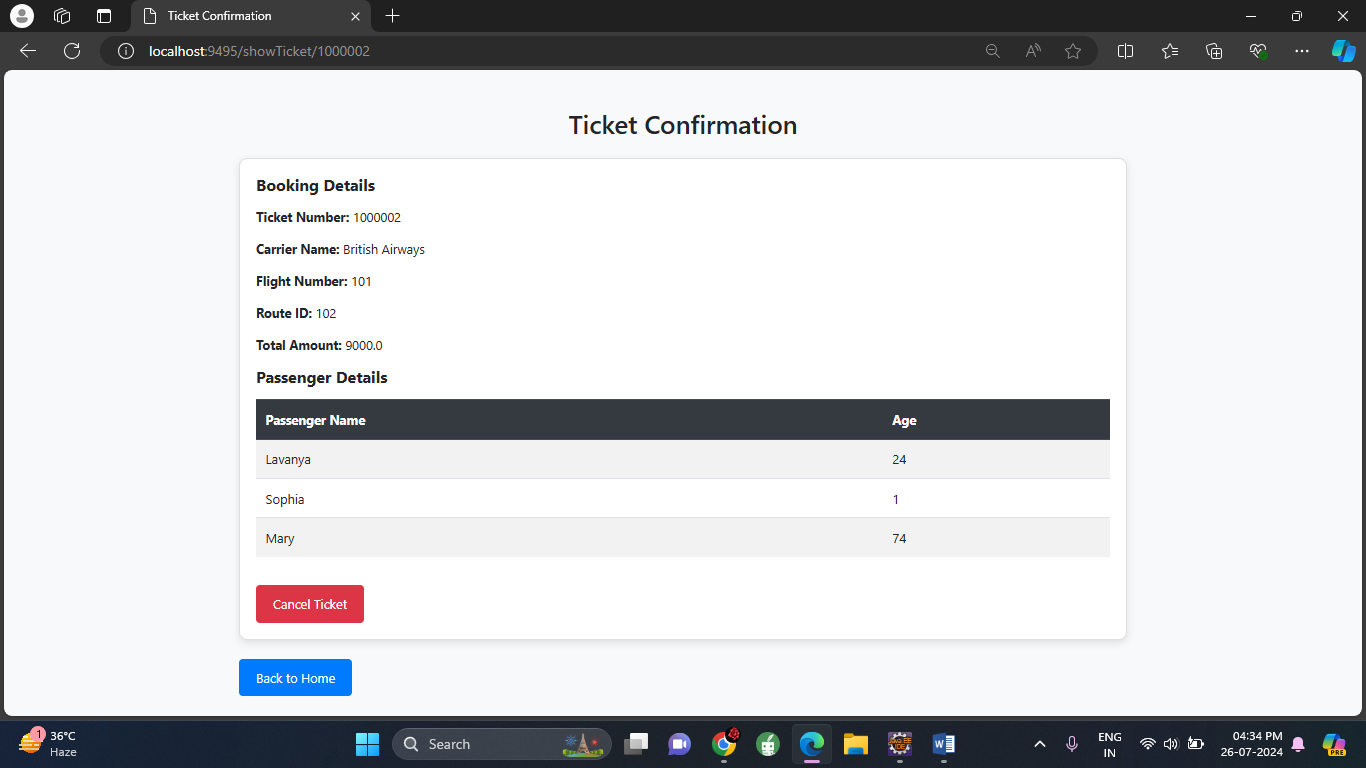
9.8.1 Ticket Booking



9.8.2 Ticket Report



9.8.3 Ticket Enquiry



9.9 Passenger Report

