[AIRLINE MANAGEMNET SYSTEM]

DOCUMENTATION

By

Sahil Kumar – 2312130

Ayesha Muzammil – 2312107

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to **Sir Abid Ali** for his continuous support, valuable guidance, and encouragement throughout the completion of this project. His dedication and expertise played a vital role in helping us understand the concepts and successfully complete this task within the given timeframe.

We are also thankful to our group members, **Sahil Kumar** and **Ayesha Muzammil**, for their collaboration, hard work, and team spirit throughout the project.

Lastly, we sincerely appreciate the support of our friends and everyone who contributed directly or indirectly, especially those who took the time to participate in our questionnaires. Their involvement was essential in bringing this project to life.

CONTENT

INTRODUCTION

The **Airline Reservation System** is a web-based data management platform that facilitates viewing, inserting, and analyzing airline-related records through an intuitive user interface. The project is designed using modern web technologies (HTML, CSS, and JavaScript) and is connected to a PostgreSQL database hosted on **Neon**. The development environment used is **GitHub Codespaces**, allowing for easy collaboration and code management.

Unlike traditional reservation systems that focus on booking and payments, this system emphasizes on **data handling and visualization**. It provides a structured and user-friendly way to manage data related to flights, passengers, airlines, airports, and more.

OBJECTIVES OF THE PROJECT

- To develop a centralized airline data management interface using modern web technologies.

- To provide a simplified UI that allows users to:
  - View all data entries from various airline-related tables.
  - Insert new data entries into specific tables.
  - Visualize the data using graphical charts.

- To integrate a **Neon PostgreSQL database** with a responsive **frontend built** using **HTML/CSS/JS**.

- To learn and demonstrate real-time interaction between a database and a dynamic web interface.

TOPIC OF OUR PROJECT

Project Title: Airline Management System

Type: Web-Based Data Management and Visualization System(ADIMIN_SIDE)

Hosted on: GitHub Codespaces

Database: Neon PostgreSQL

HARDWARE/SOFTWARE REQUIREMENTS

Hardware Requirements:

- A computer/laptop with:
  - Minimum 4GB RAM
  - Dual-core processor or higher
  - Internet connectivity

Software Requirements:

- **Frontend:**
- HTML5, Tailwind CSS, JavaScript
- Express.js, Node.js
- PostgreSQL(Neon)
- **Development & Hosting:**
- GitHub Codespaces
- Web browser (Chrome/Edge)

UML (Unified Modeling Language)

Use Case Description (Without Diagram):

Actor:

- **User** – The person interacting with the system (can be an admin or a general user)

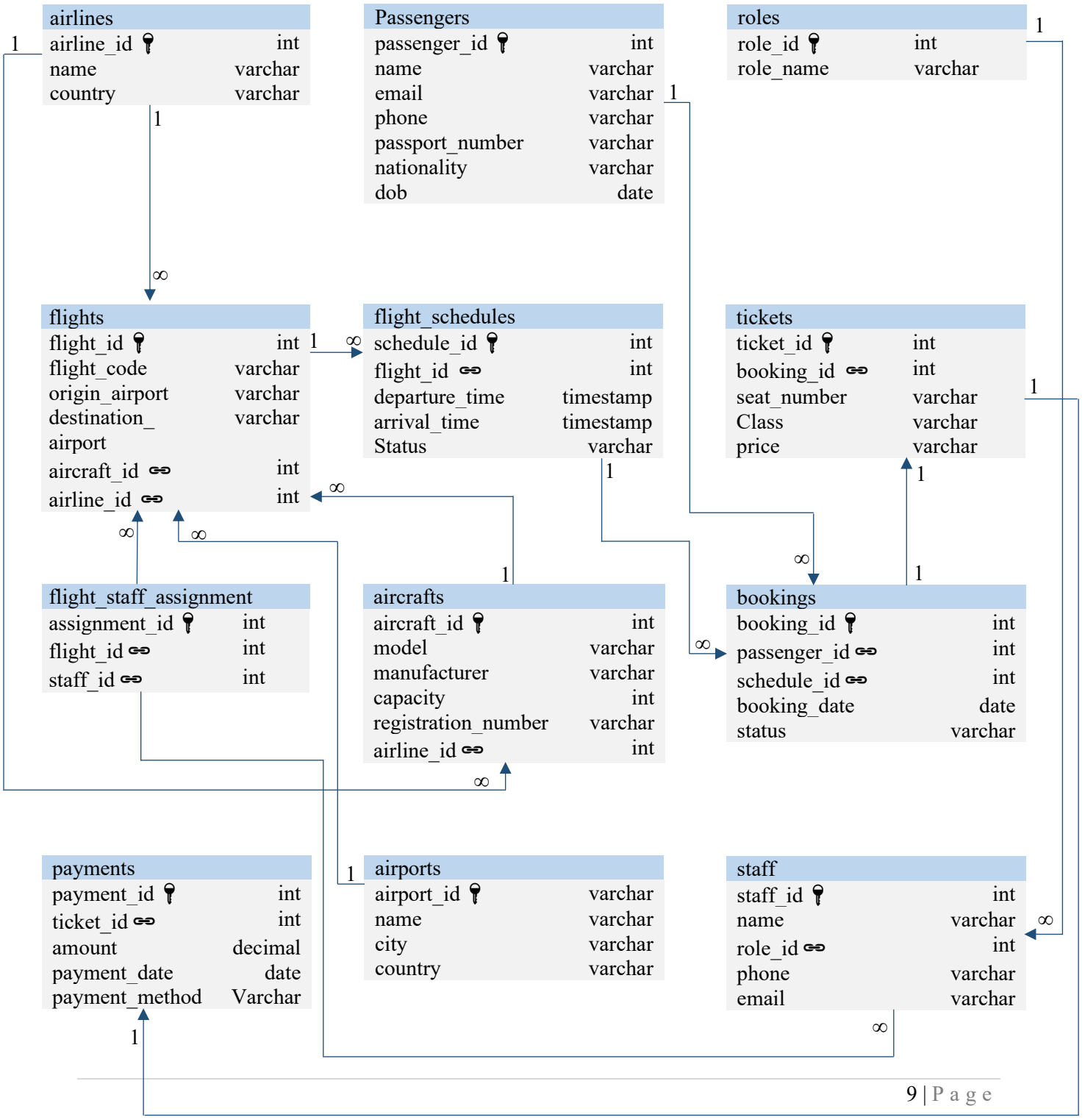
Use Cases:

1. **Select Table** – The user selects a table (such as Flights, Bookings, Passengers, etc.)
2. **View Data** – The system displays the data from the selected table
3. **Insert Data** – The user can insert new records into the selected table
4. **View Graphs** – (pie chart, bar) the system generates visual graphs based on the table's data.

System Behavior:

- When the user selects a specific table, the system offers three main functionalities:
    - View existing data
    - Insert new data
    - View data in graphical form
- These use cases represent the core functionality of the airline reservation system, providing both data interaction and visualization

ERD


airlines
airline_id int
name varchar
country varchar
Passengers
passenger_id int
name varchar
email varchar
phone varchar
passport_number varchar
nationality varchar
dob date
roles
role_id int
role_name varchar
flights
flight_id int
flight_code varchar
origin_airport varchar
destination_ airport varchar
aircraft_id int
airline_id int
flight_schedules
schedule_id int
flight_id int
departure_time timestamp
arrival_time timestamp
Status varchar
tickets
ticket_id int
booking_id int
seat_number varchar
Class varchar
price varchar
flight_staff_assignment
assignment_id int
flight_id int
staff_id int
aircrafts
aircraft_id int
model varchar
manufacturer varchar
capacity int
registration_number varchar
airline_id int
bookings
booking_id int
passenger_id int
schedule_id int
booking_date date
status varchar
payments
payment_id int
ticket_id int
amount decimal
payment_date date
payment_method Varchar
airports
airport_id varchar
name varchar
city varchar
country varchar
staff
staff_id int
name varchar
role_id int
phone varchar
email varchar

WORK ANALYSIS

Technology Stack:

- **Frontend:** HTML5, Tailwind CSS, JavaScript
- **Backend:** Express.js, Node.js
- **Database:** PostgreSQL(Neon)
- **Environment:** GitHub Codespaces

Main Page Functionality:

- A **dashboard-style interface** shows all available tables (cards or list items).
- For each table, users can:
    - **View Data:** Shows all records in a table view.
    - **Insert Data:** Opens a form to insert a new record.
    - **View Graph:** Displays visual charts (e.g., bar or pie) representing the table's statistics.

Process Workflow:

1. User opens the main page.
2. All database tables are shown.
3. User clicks on a table.
4. Three options appear:
    - View Data (using SELECT query)
    - Insert Data (form submission using INSERT query)

- View Graph (uses JavaScript charts)

Challenges Faced:

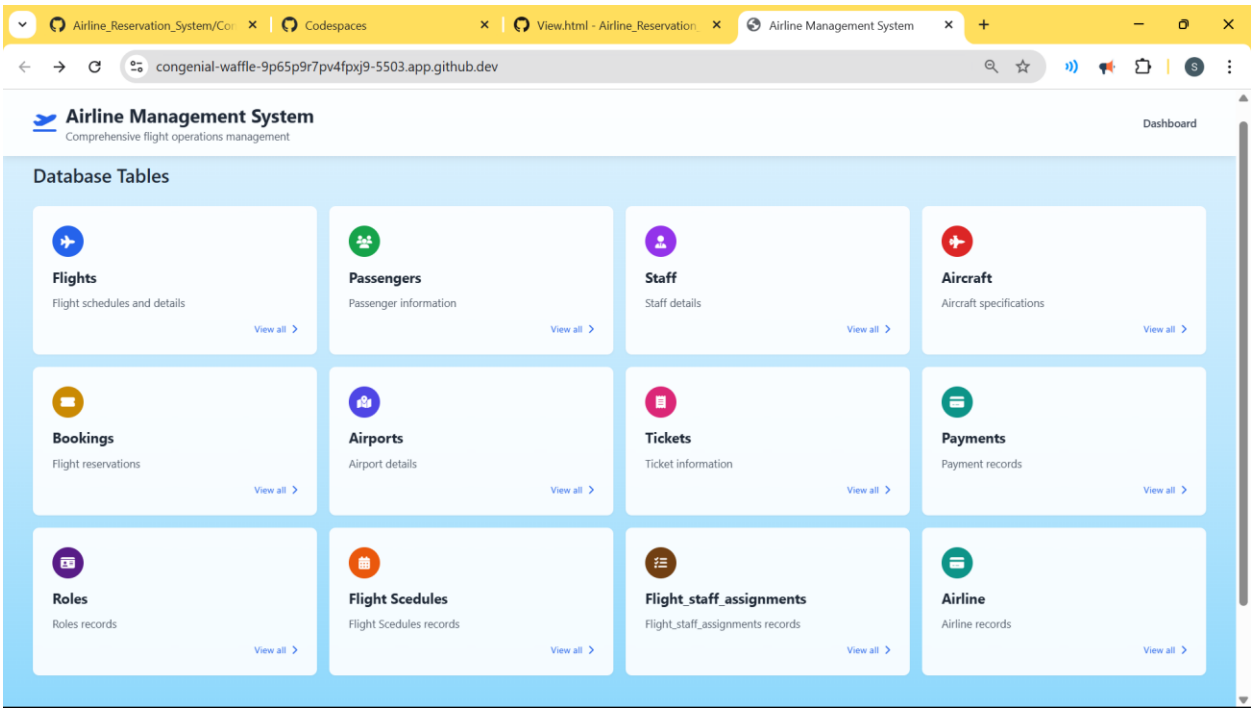
- Designing a reusable interface for multiple tables.
- Ensuring correct mapping between frontend and Neon-hosted database.
- Creating dynamic chart representations of each table's data.
- Managing real-time form input validation and data integrity.

Achievements:

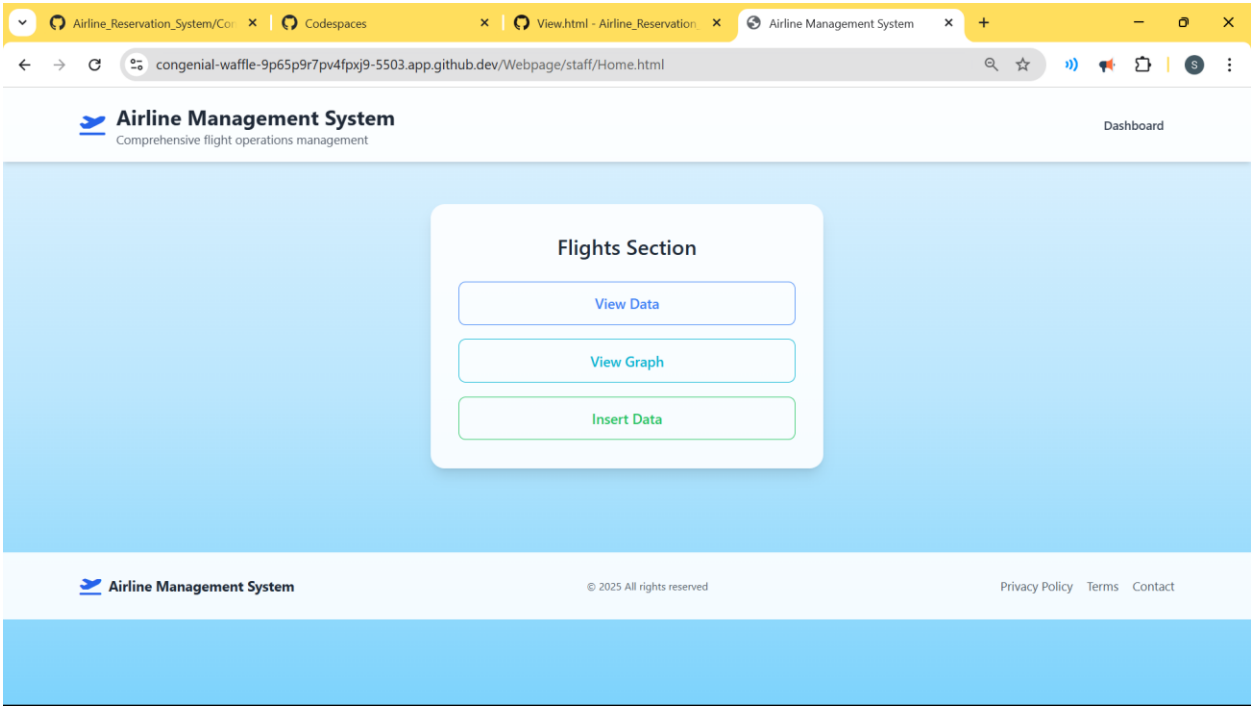
- A complete, responsive, and minimal web app that interacts with Neon DB.
- Integration of form-based data insertion and data visualization.
- A working CRUD-like model (Create and Read) for airline system data handling.

SCREENSHOTS OF OUTPUT

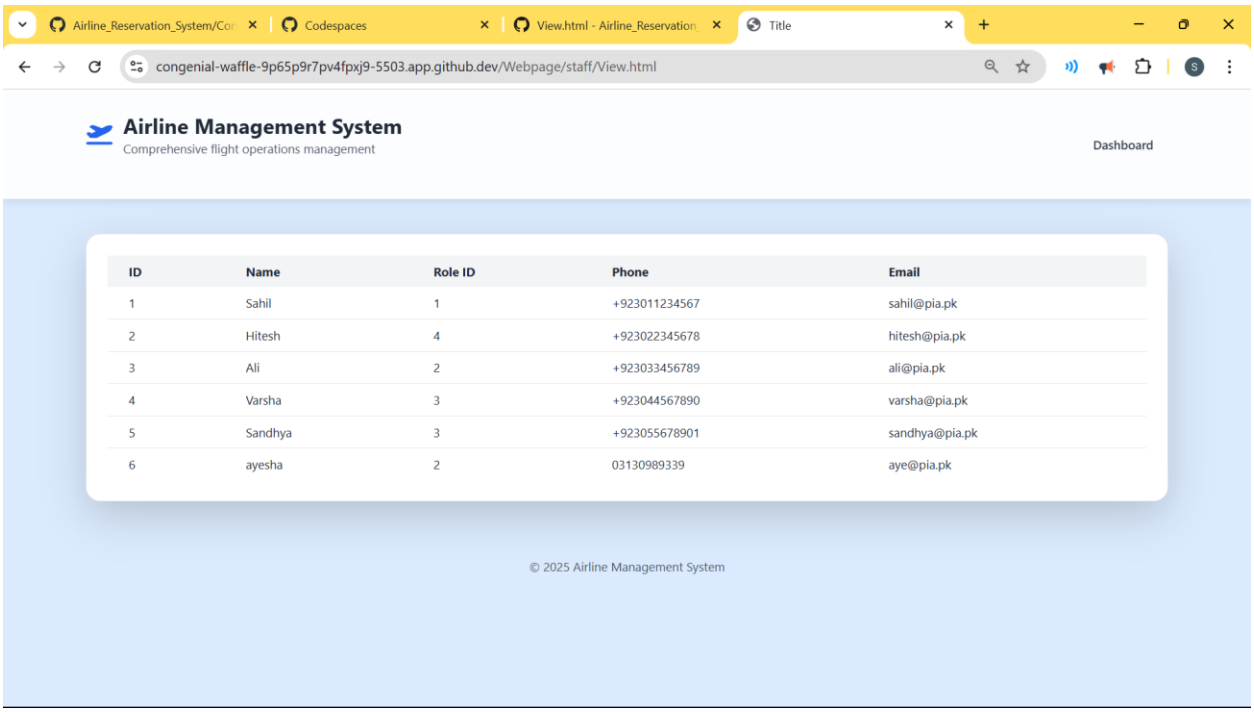
Main Page



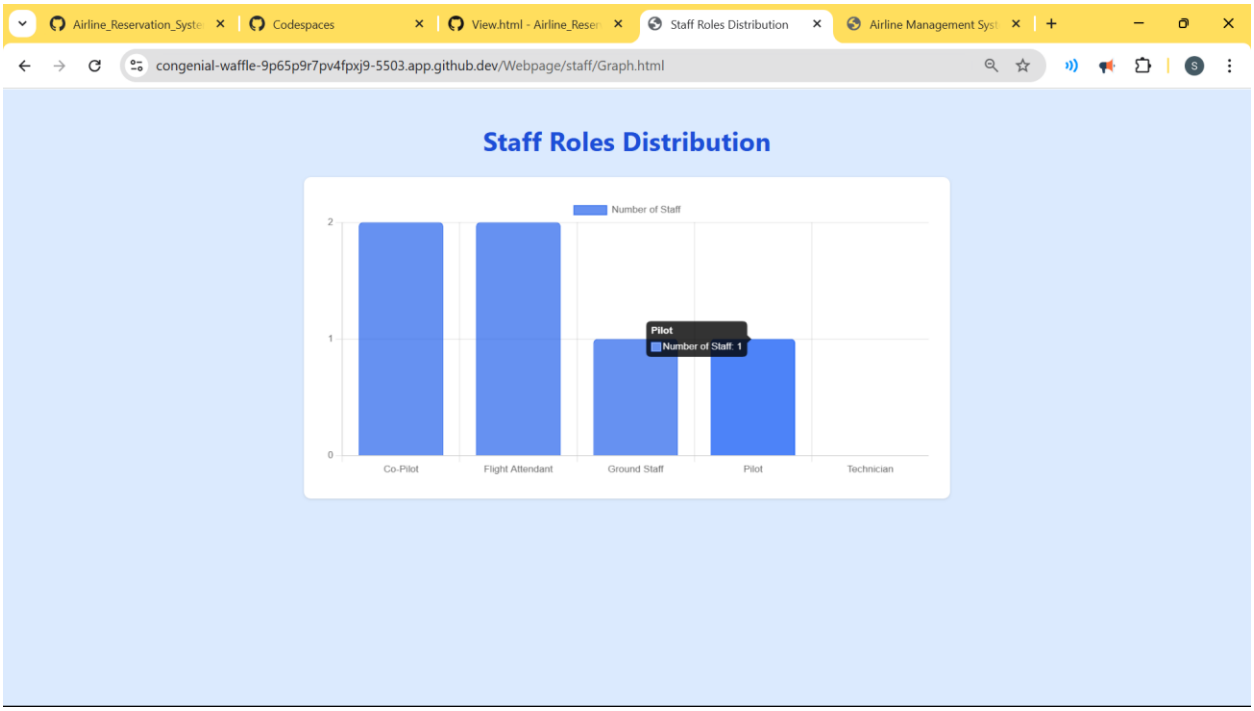
Staff



View Data

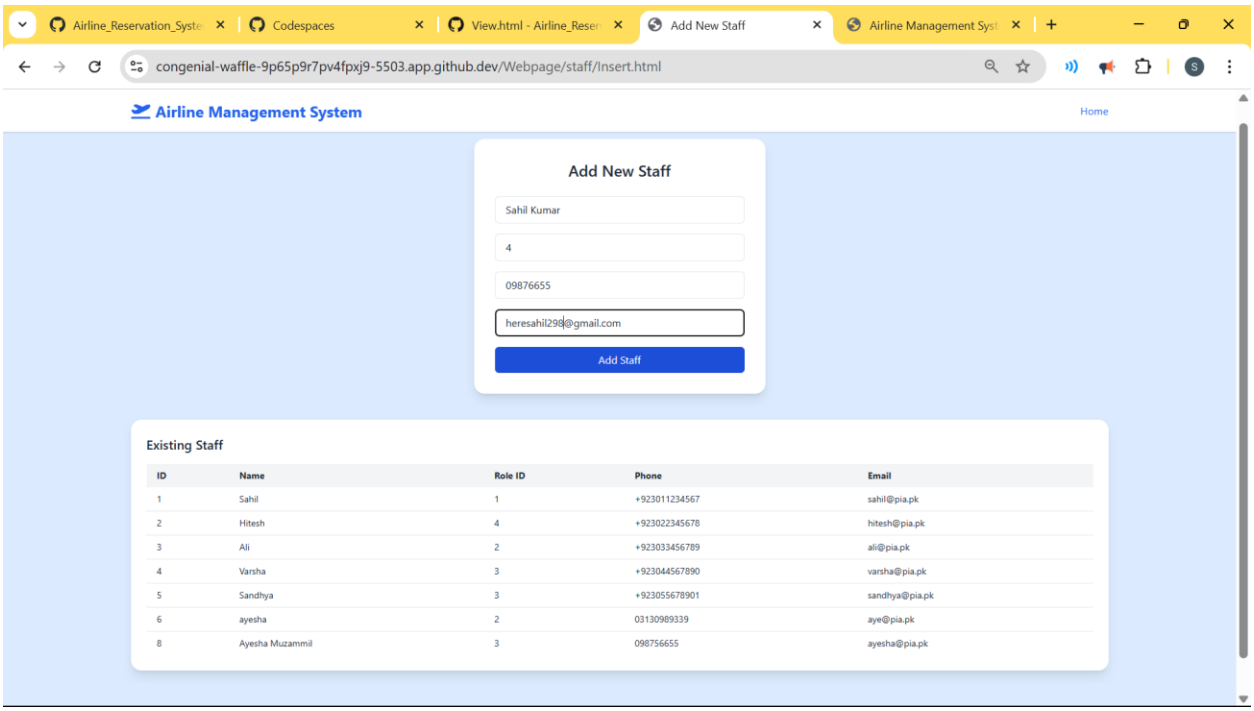


View Graph

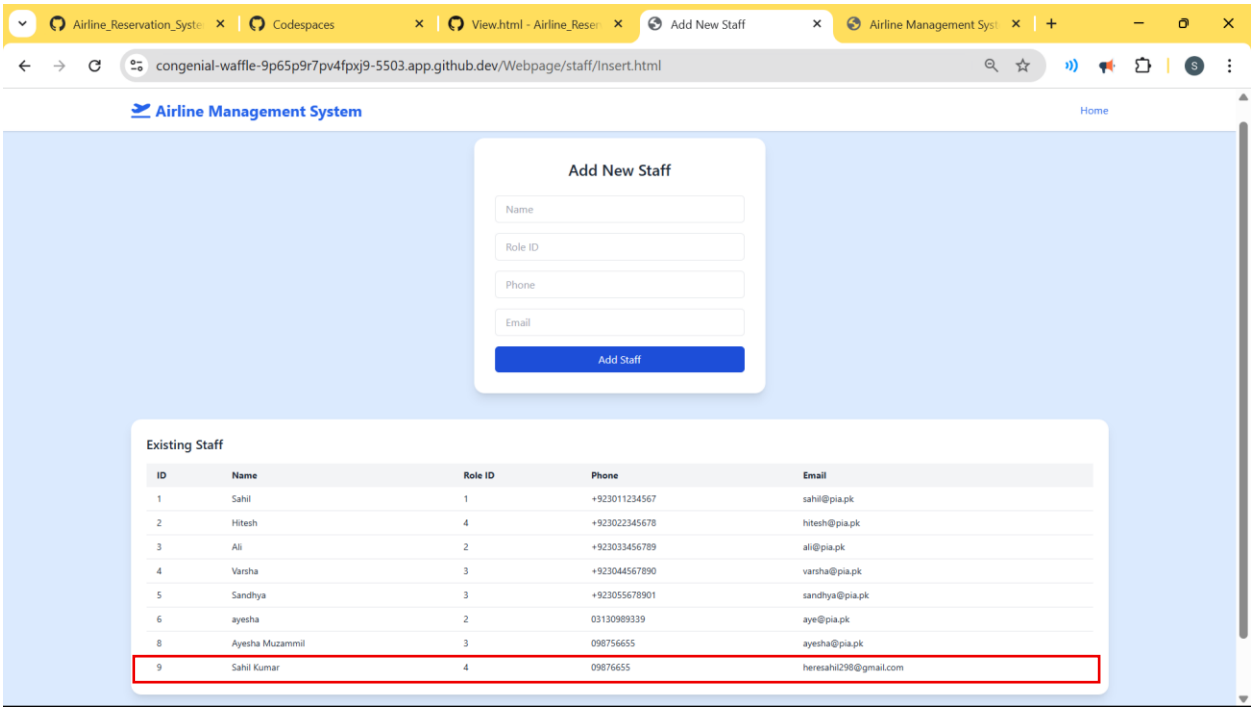


Insert Data

Before Inserting

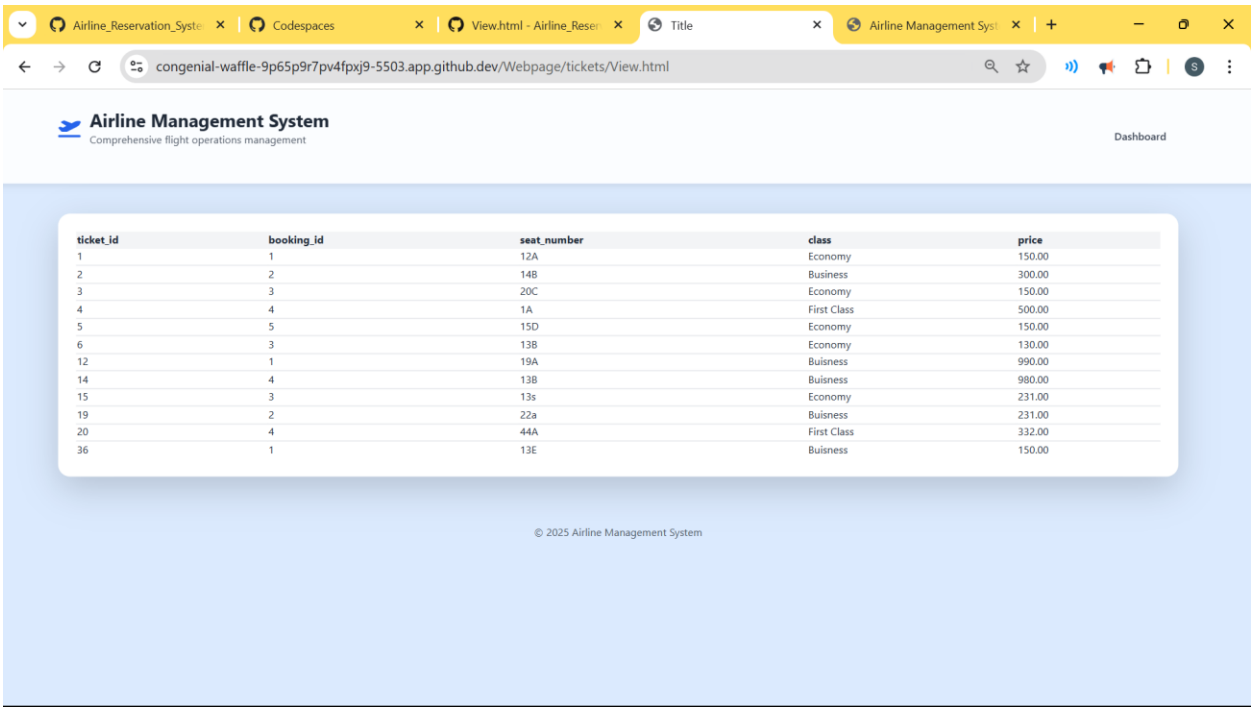


After Inserting



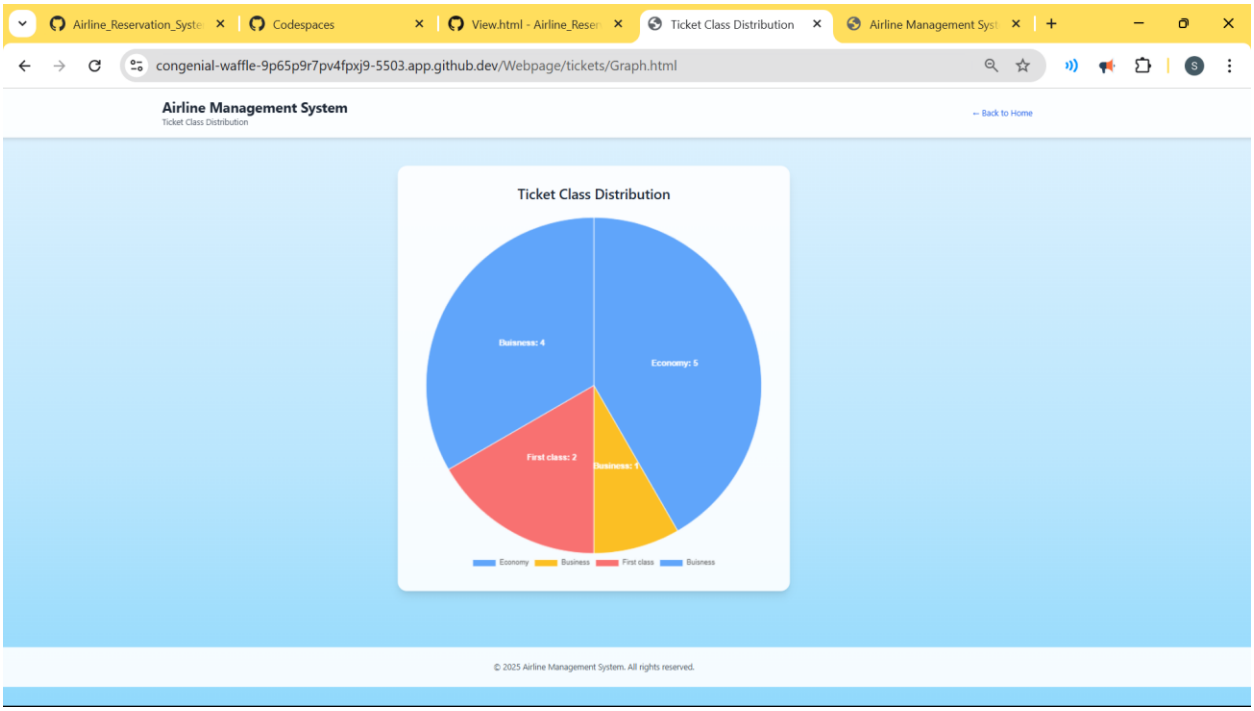
Tickets

View Data



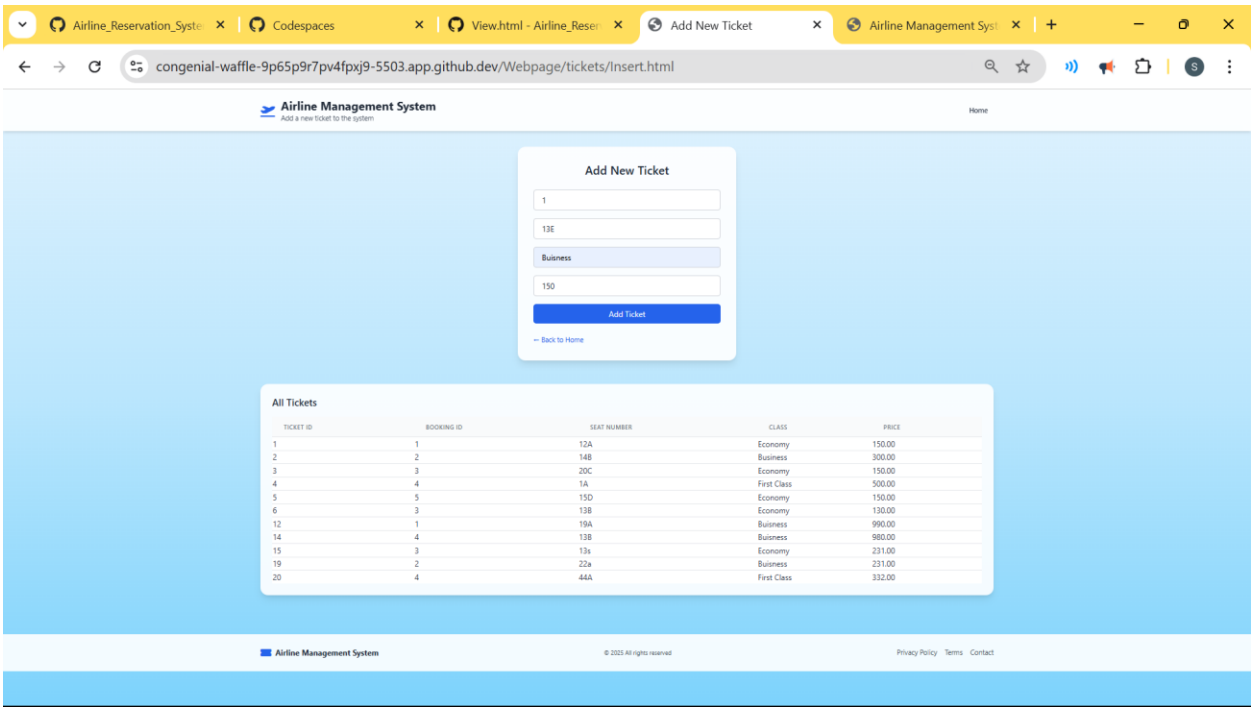
ticket_id	booking_id	seat_number	class	price
1	1	12A	Economy	150.00
2	2	14B	Business	300.00
3	3	20C	Economy	150.00
4	4	1A	First Class	500.00
5	5	15D	Economy	150.00
6	3	13B	Economy	130.00
12	1	19A	Buisness	990.00
14	4	13B	Buisness	980.00
15	3	13s	Economy	231.00
19	2	22a	Buisness	231.00
20	4	44A	First Class	332.00
36	1	13E	Buisness	150.00

View Graph

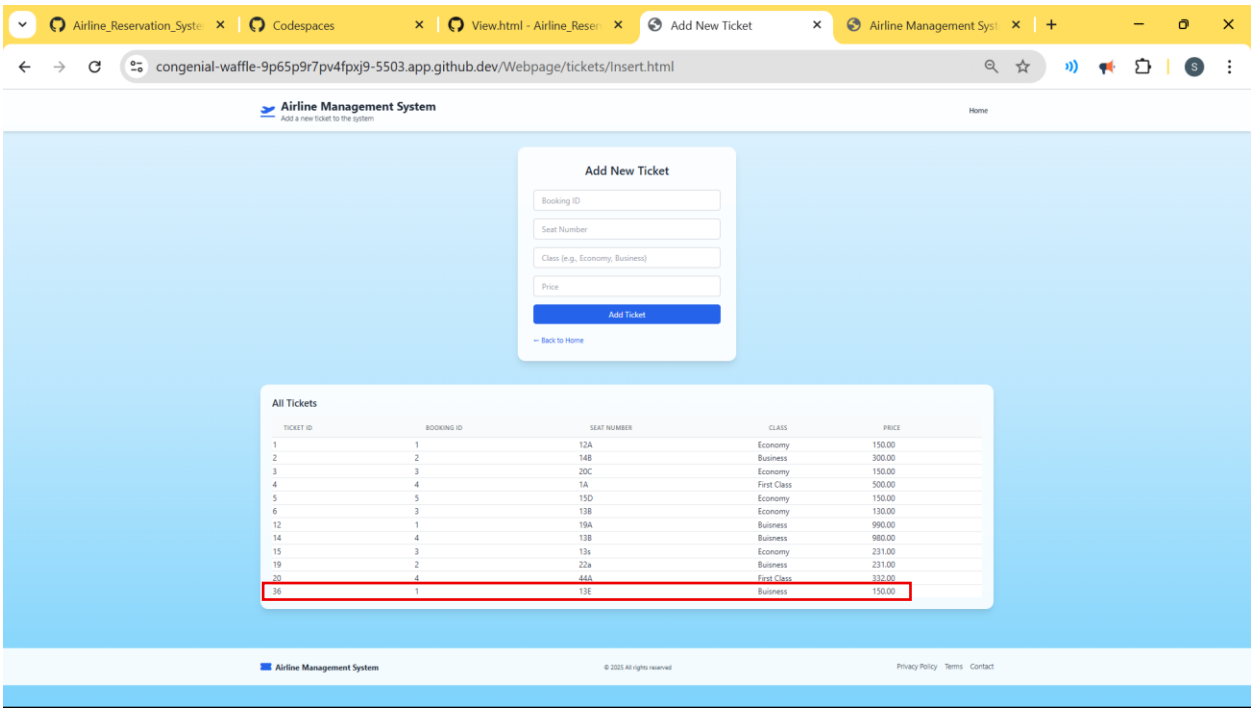


Insert Data

Before Inserting

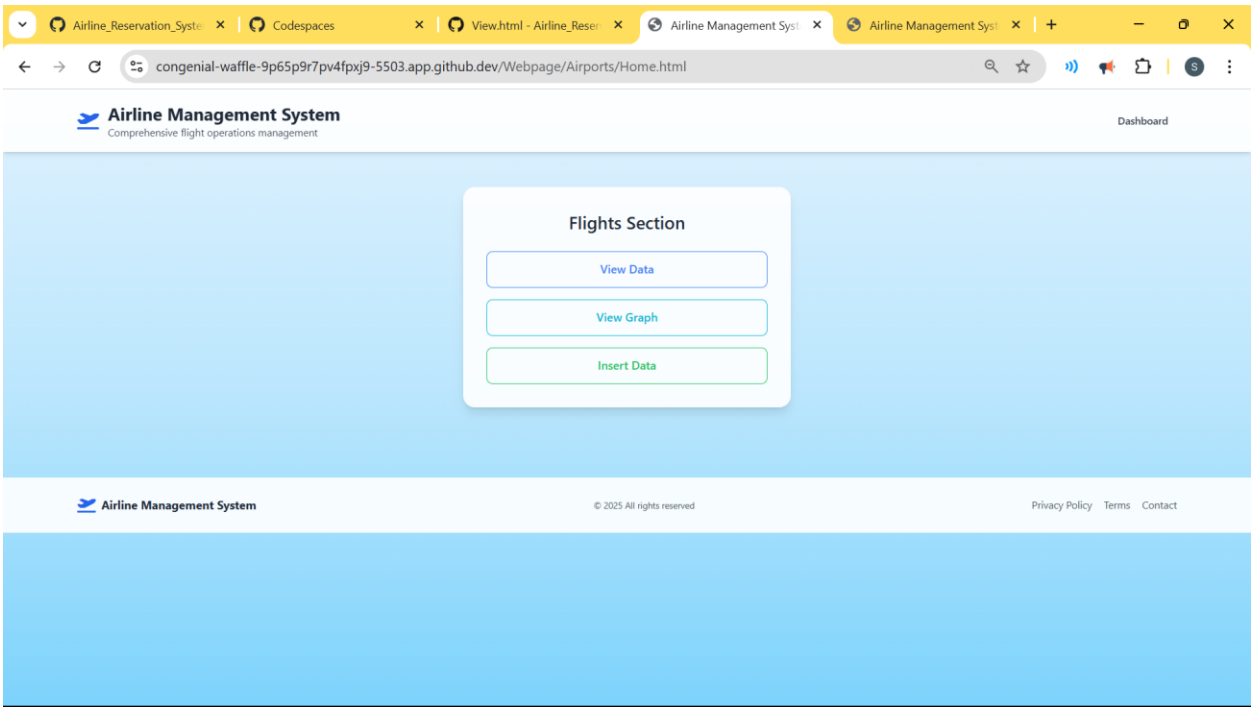


After Inserting

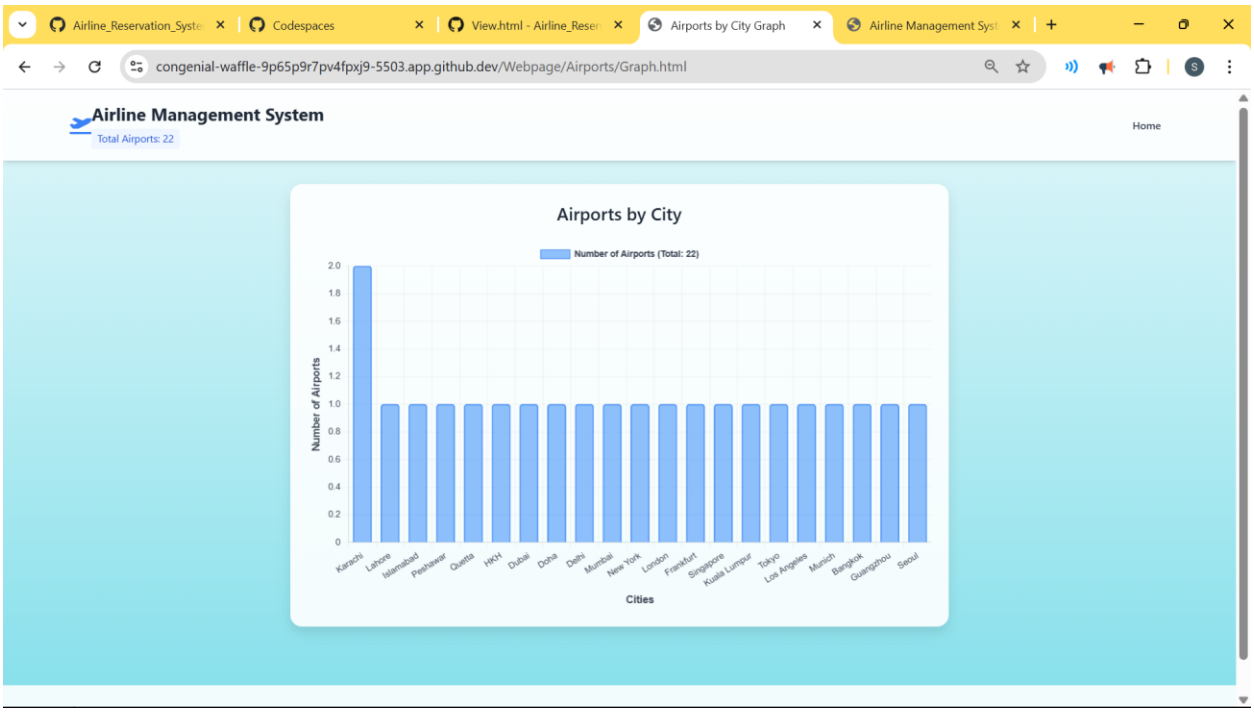


Airports

View Data

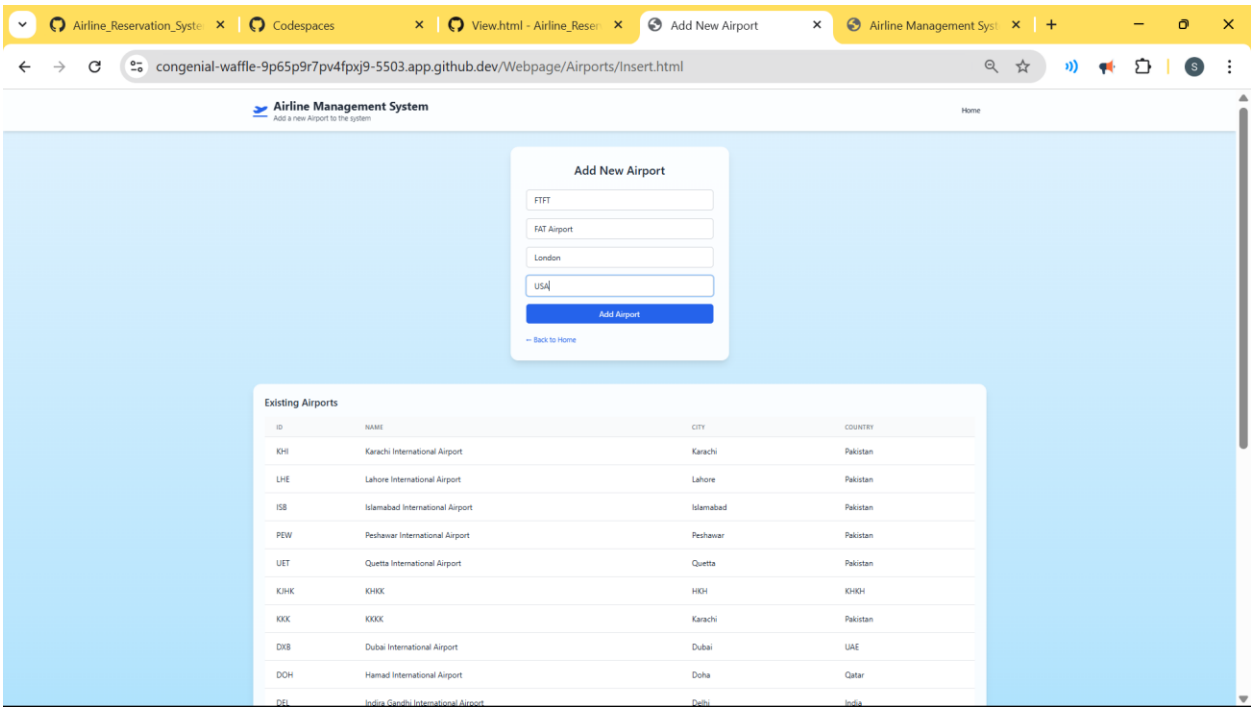


View Graph



Insert Data

Before Inserting



After Inserting