**PROF. V. B. SHAH INSTITUTE OF MANAGEMENT,**

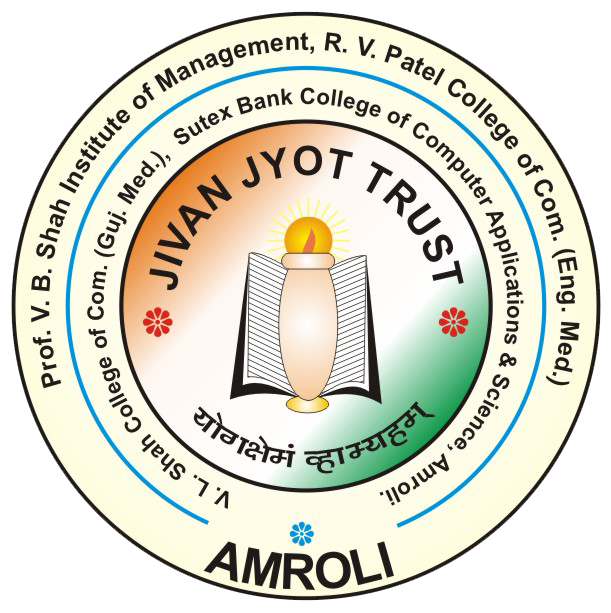
**R. V. PATEL COLLEGE OF COMMERCE (ENG. MED.),**

**V. L. SHAH COLLEGE OF COMMERCE (GUJ. MED.) &**

**SUTEX BANK COLLEGE OF COMPUTER APPLICATION & SCIENCE**

(Accredited ‘B’ (CGPA 2.55) by NAAC Dec.-2009)

VEER NARMAD SOUTH GUJARAT UNIVERSITY, SURAT



MINOR-PROJECT REPORT

BACHELOR OF COMPUTER APPLICATION

(5TH SEMESTER) YEAR 2025-26

**AIRSTRO**

**GUIDED BY:**

Ms. Sanskruti Ramani

**SUBMITTED BY:**

647 - NISHIL CHAVDA

658 - DEEP SORATHIYA

678 - RAHUL GADHIYA

**INDEX**

|  |  |  |
| --- | --- | --- |
| **NO** | **CONTENT** | **PAGE NO** |
| **1** | **INTRODUCTION** | **1** |
| **2** | **PROJECT OVERVIEW** | **4** |
| **3** | **PROJECT PROFILE** | **3** |
| **4** | **PROJECT TARGET USER** | **4** |
| **5** | **PROJECT OVERVIEW / FEATURES** | **5** |
| **6** | **ER DIAGRAM** | **6** |
| **7** | **PROJECT INSTRUCTIONS** | **10** |
| **8** | **SCREESHOTS** | **11** |
| **9** | **DATA TABLES** | **17** |
| **10** | **SUMMARY** | **19** |

**PROJECT OVERVIEW**

* **Flight Search Functionality:**

Allows users to search for available flights based on their travel preferences.

* **Flight Booking Capability:**

Enables users to book flights directly through the platform.

* **User Registration and Authentication:**

Provides a system for users to register and log in, ensuring secure access to their booking information.

* **Admin Panel Access:**

Includes an administrative interface for managing flight schedules, user accounts, and bookings.

* **Responsive Web Interface:**

Features a user-friendly and responsive design, ensuring accessibility across various devices.

* **Backend Integration:**

Utilizes a server-side component to handle data processing, user management, and flight information retrieval.

|  |  |
| --- | --- |
| **PROJECT PROFILE** | |
| **TITLE** | **AIRSTRO** |
| **FRONTEND** | **React.JS, Tailwind CSS** |
| **BACKEND** | **Node.JS, Express.JS** |
| **DATABASE** | **MongoDB** |
| **BROWSER** | **CHROME, FIREFOX, SAFARI** |
| **GUIDE BY** | **Ms. Sanskruti Ramani** |
| **SUBMITTED TO** | **Ms. Sanskruti Ramani** |



**PROJECT TARGET USER**

**PROJECT OVERVIEW / FEATURES**

****

**Admin Management**

**Content/Functionality Display**

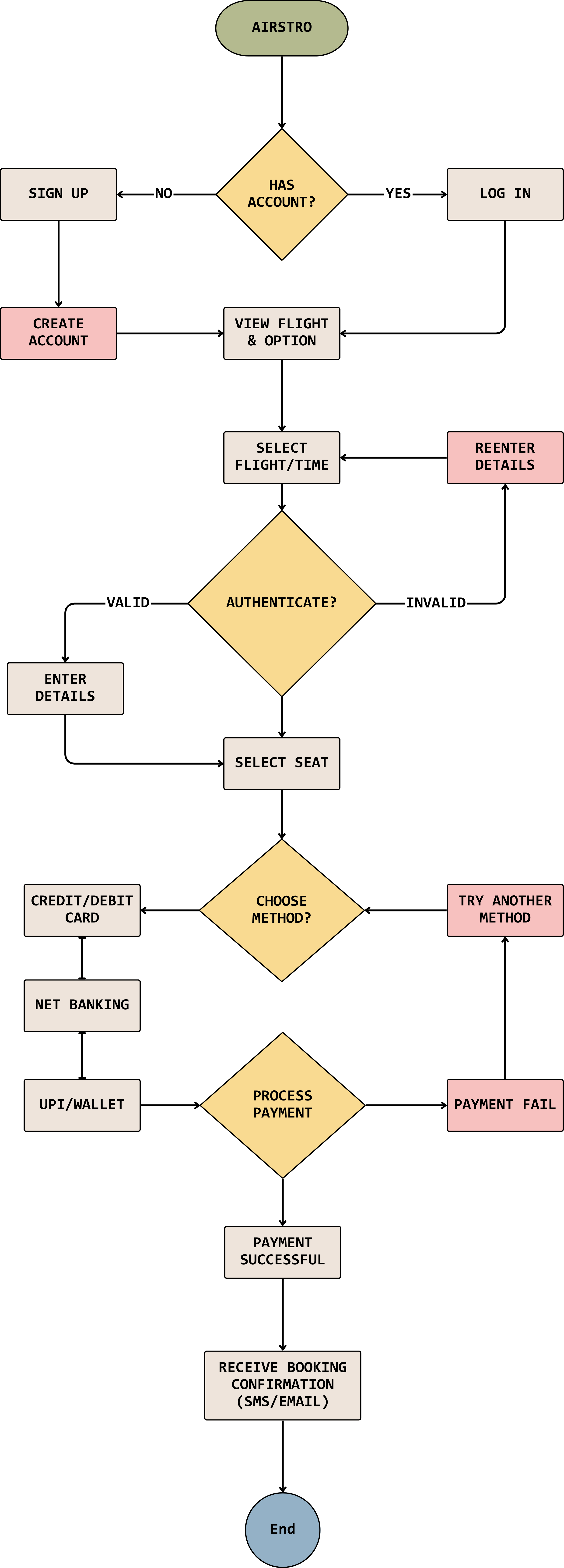
**Data Interaction**

**User Access Client Interface**

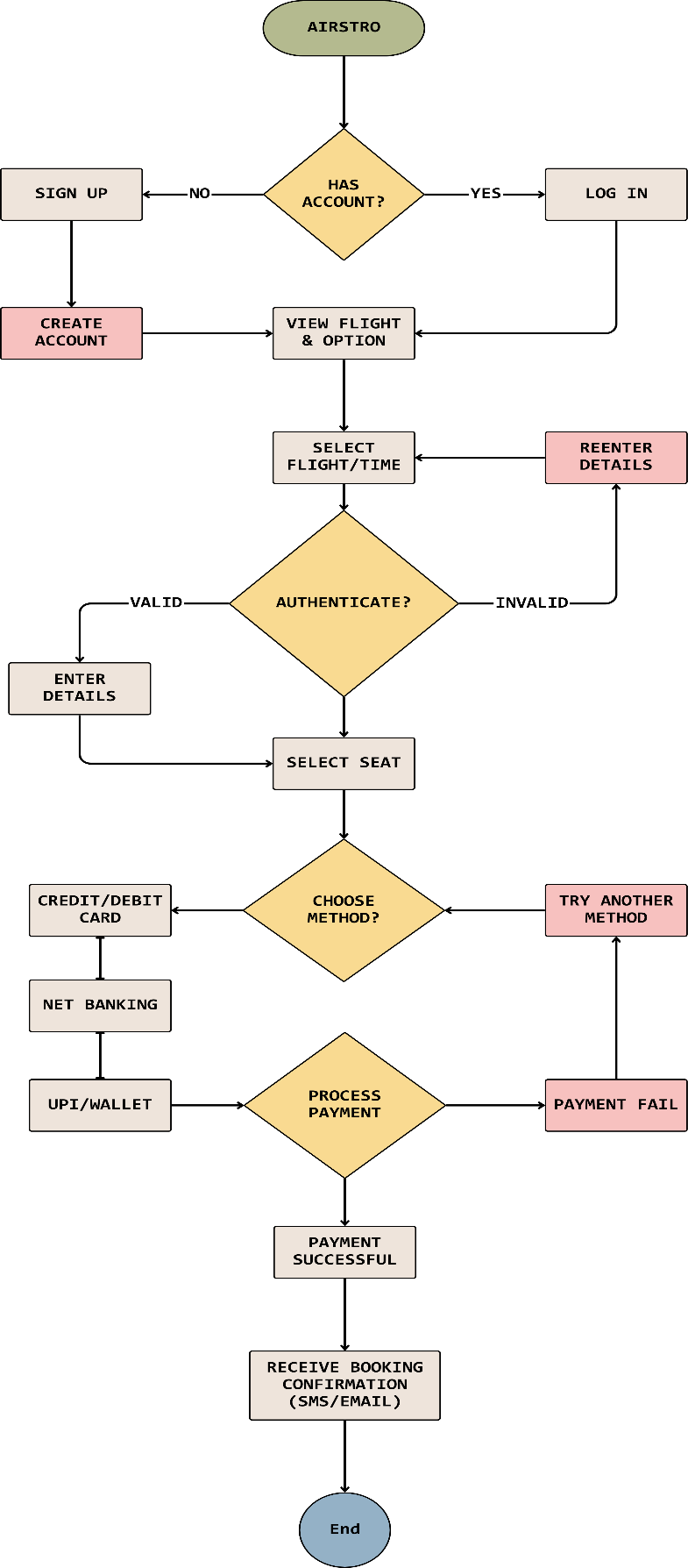
**Admin**

**Management**

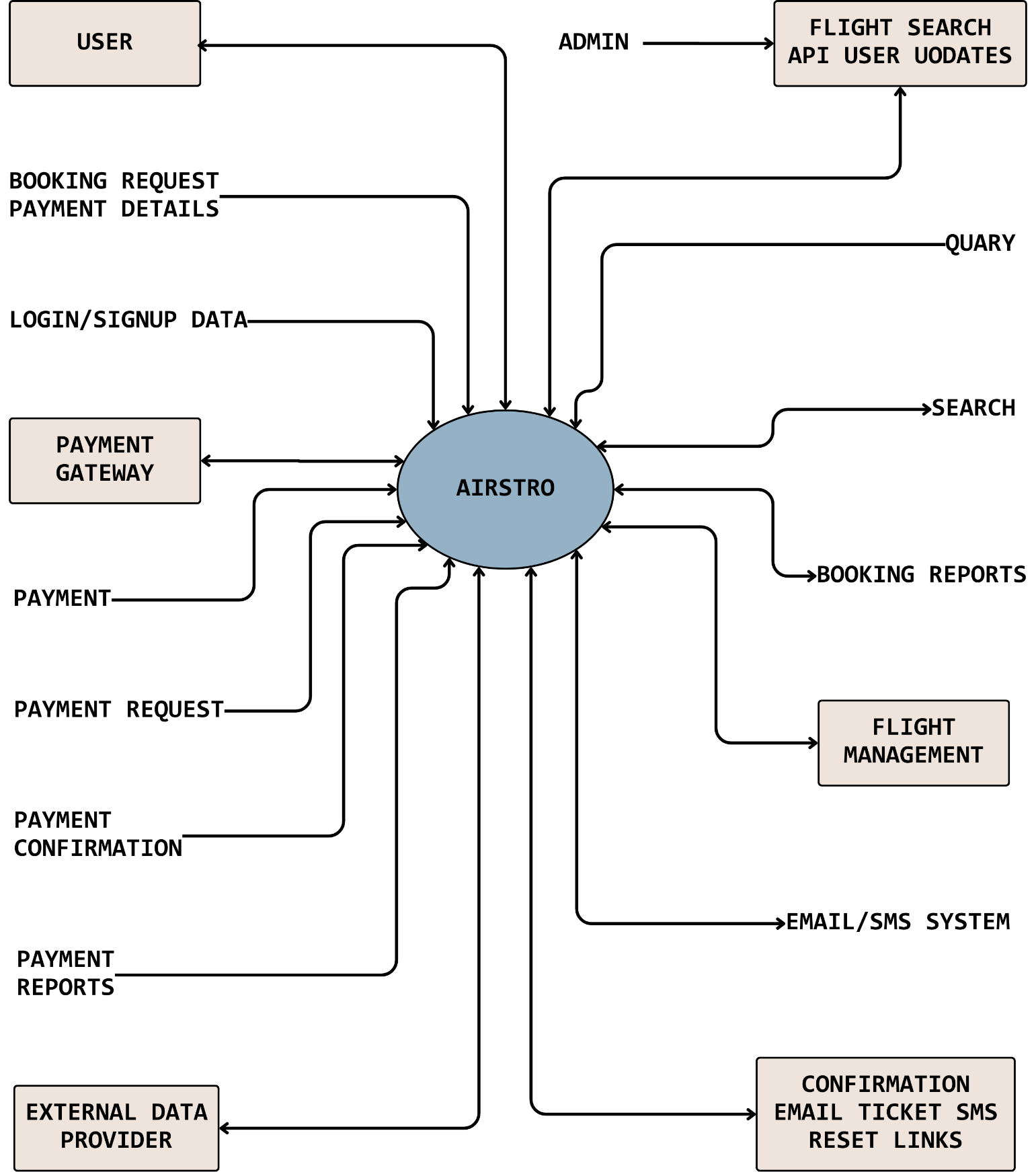


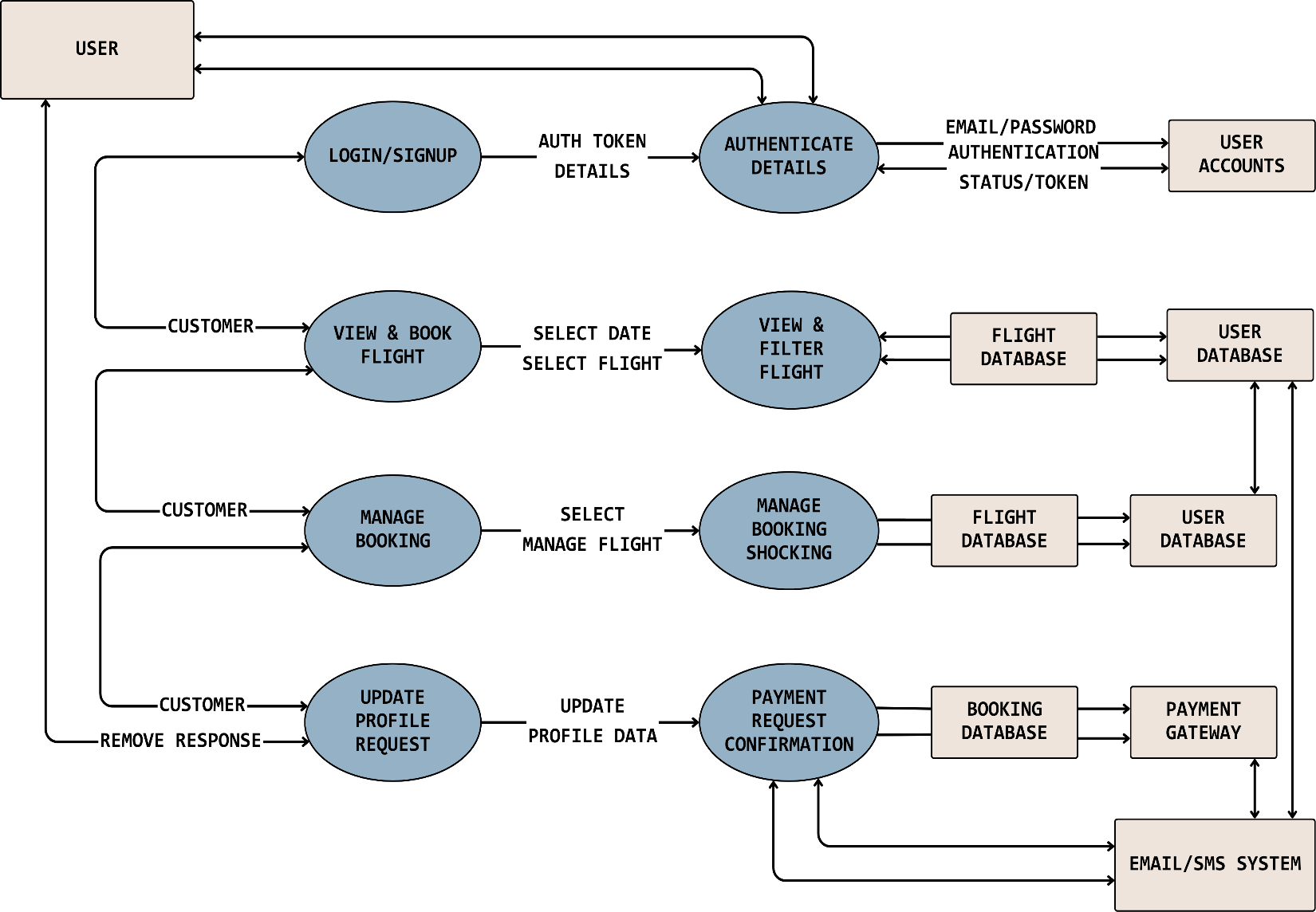
**SIGN UP/LOG IN**

**USER PROCESS FLOW DIAGRAM**



**0 LEVEL DIAGRAM**

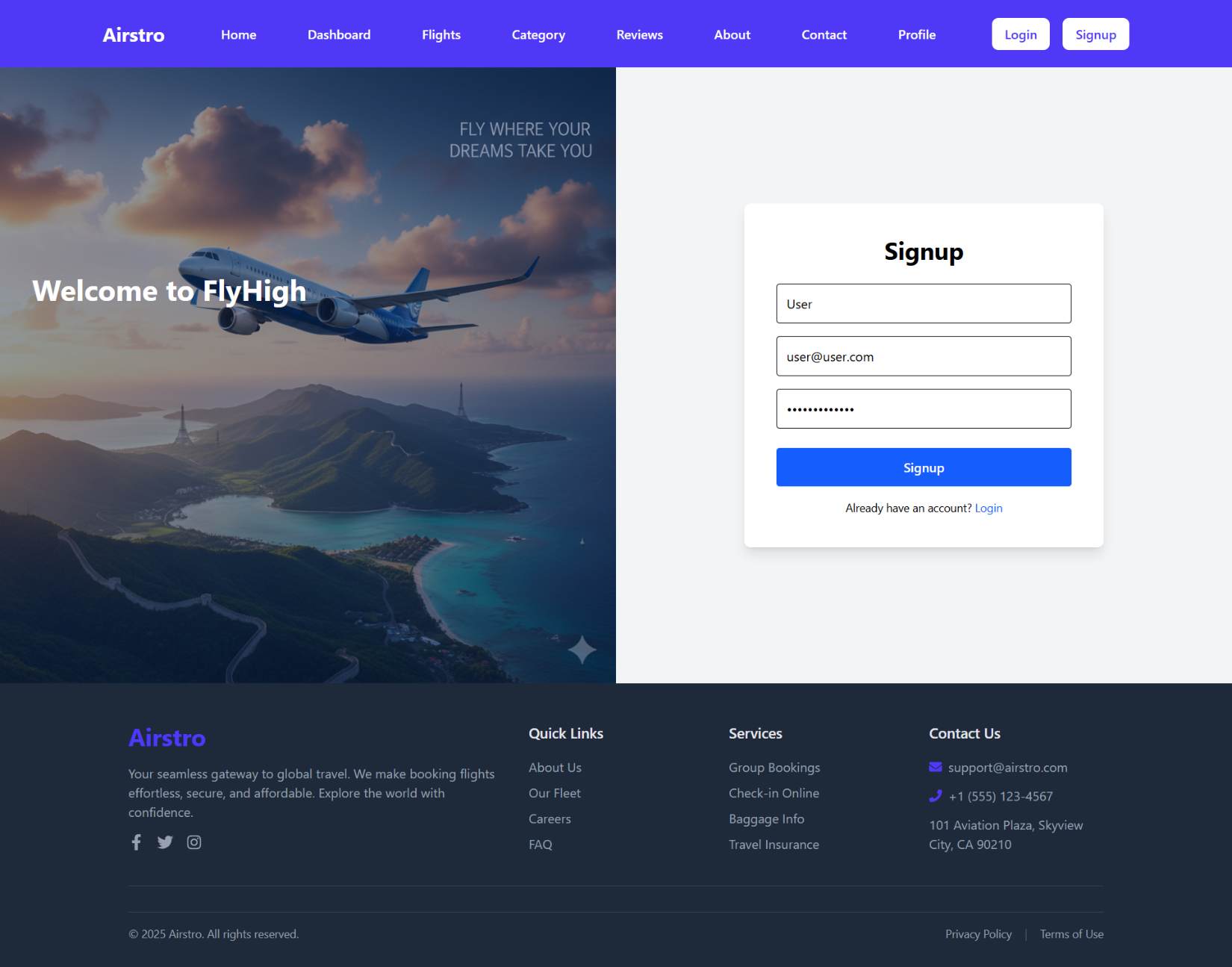


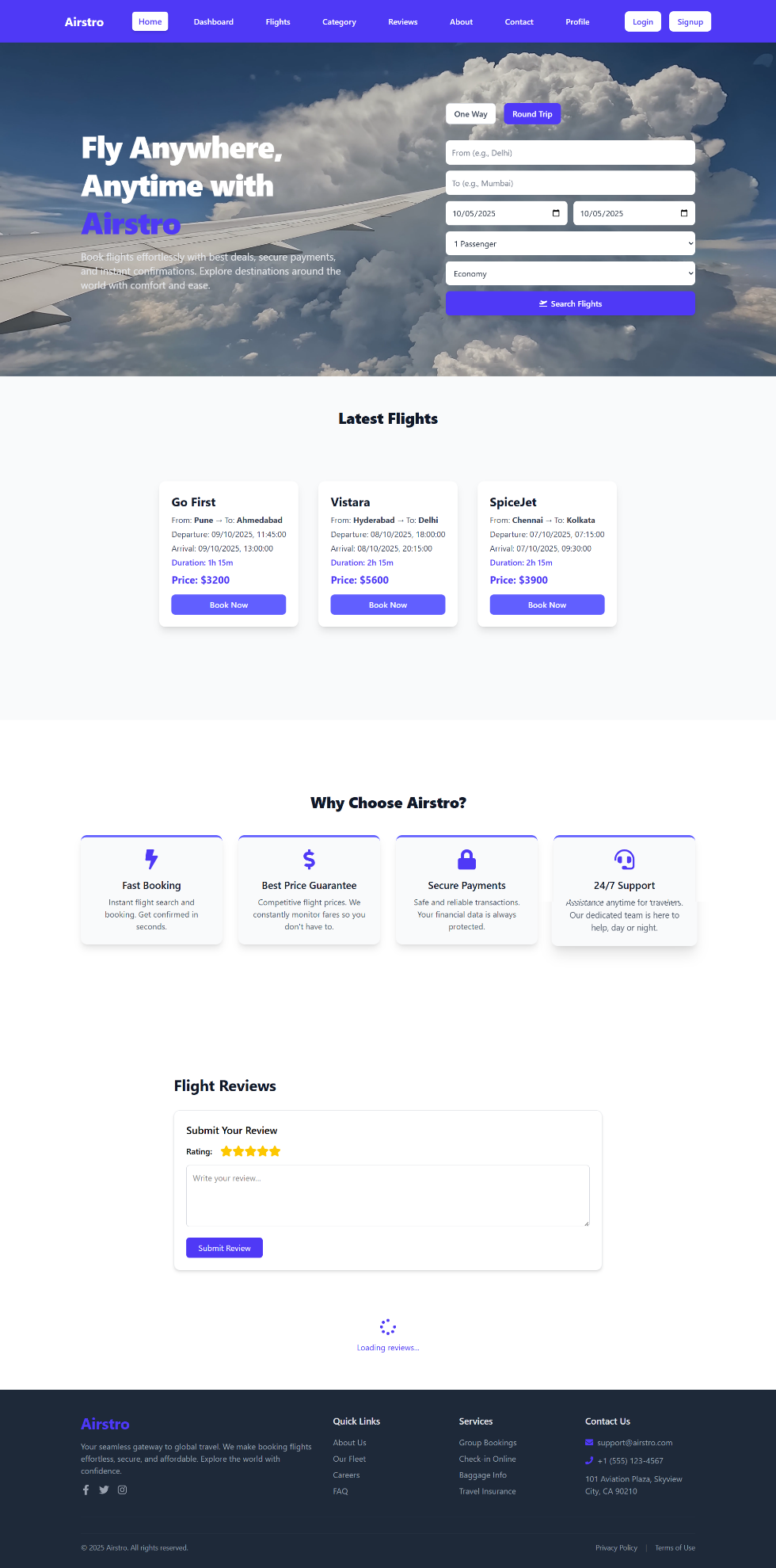


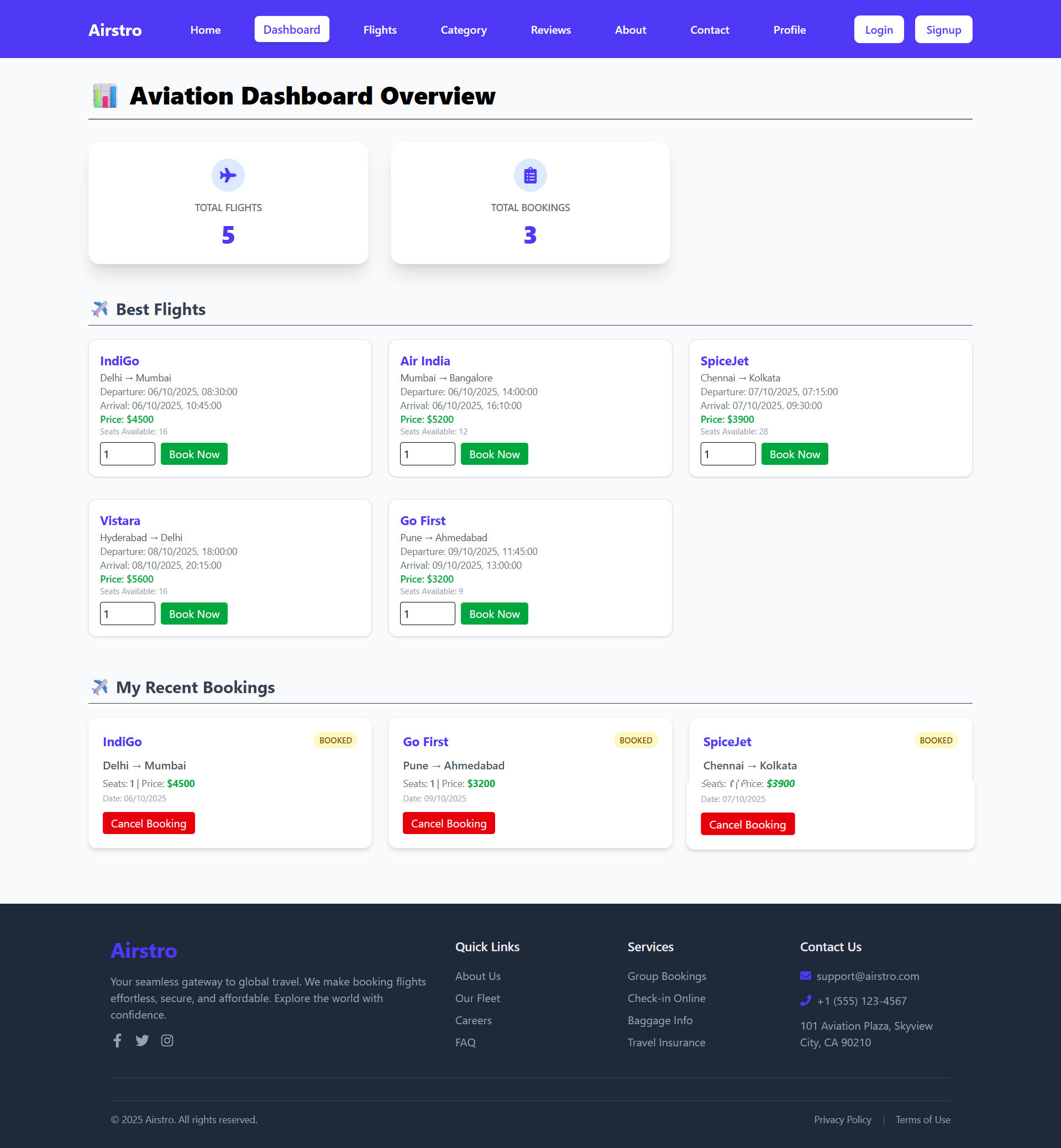
**USER DIAGRAM**

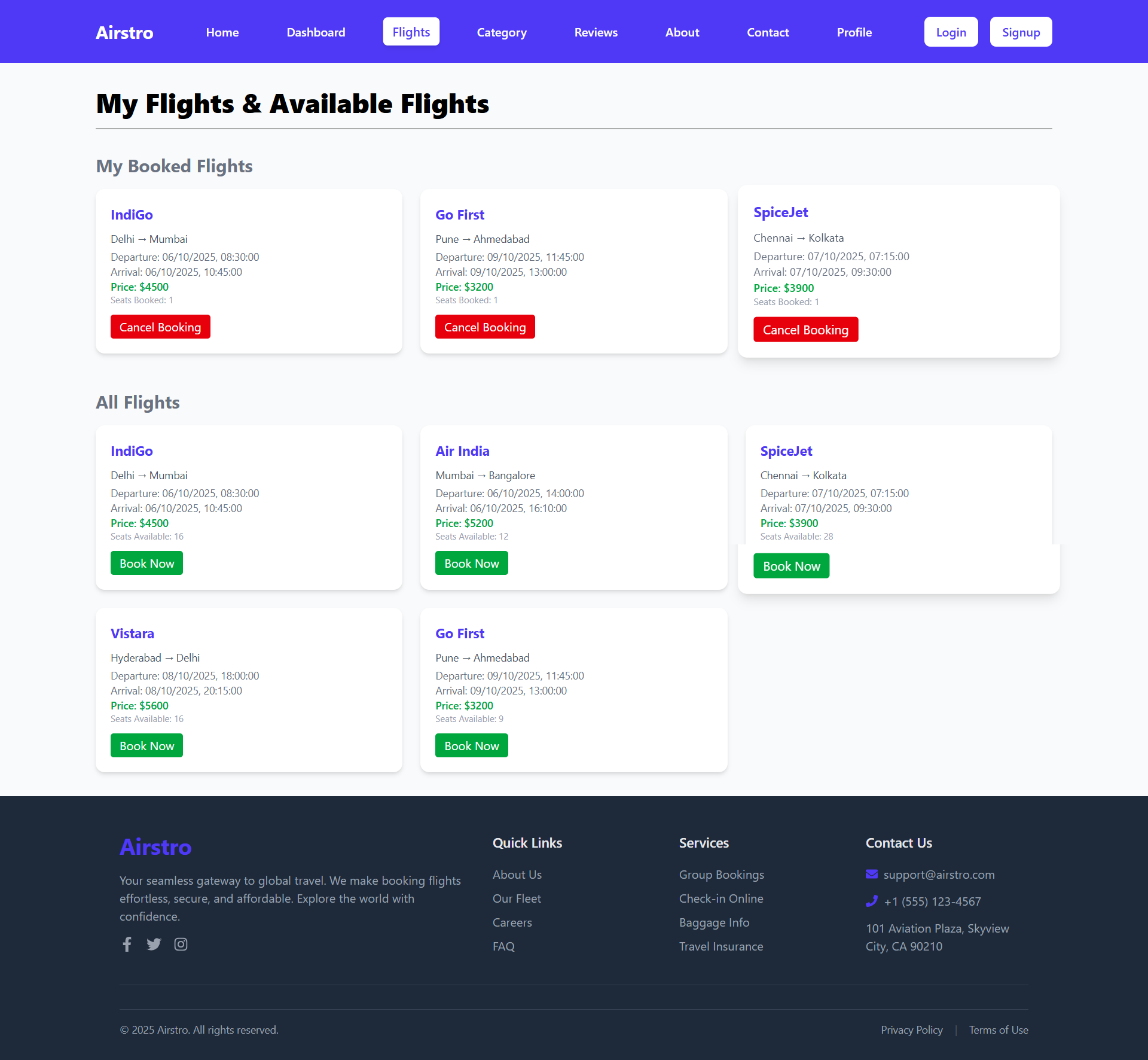
**PROJECT INSTRUCTIONS**

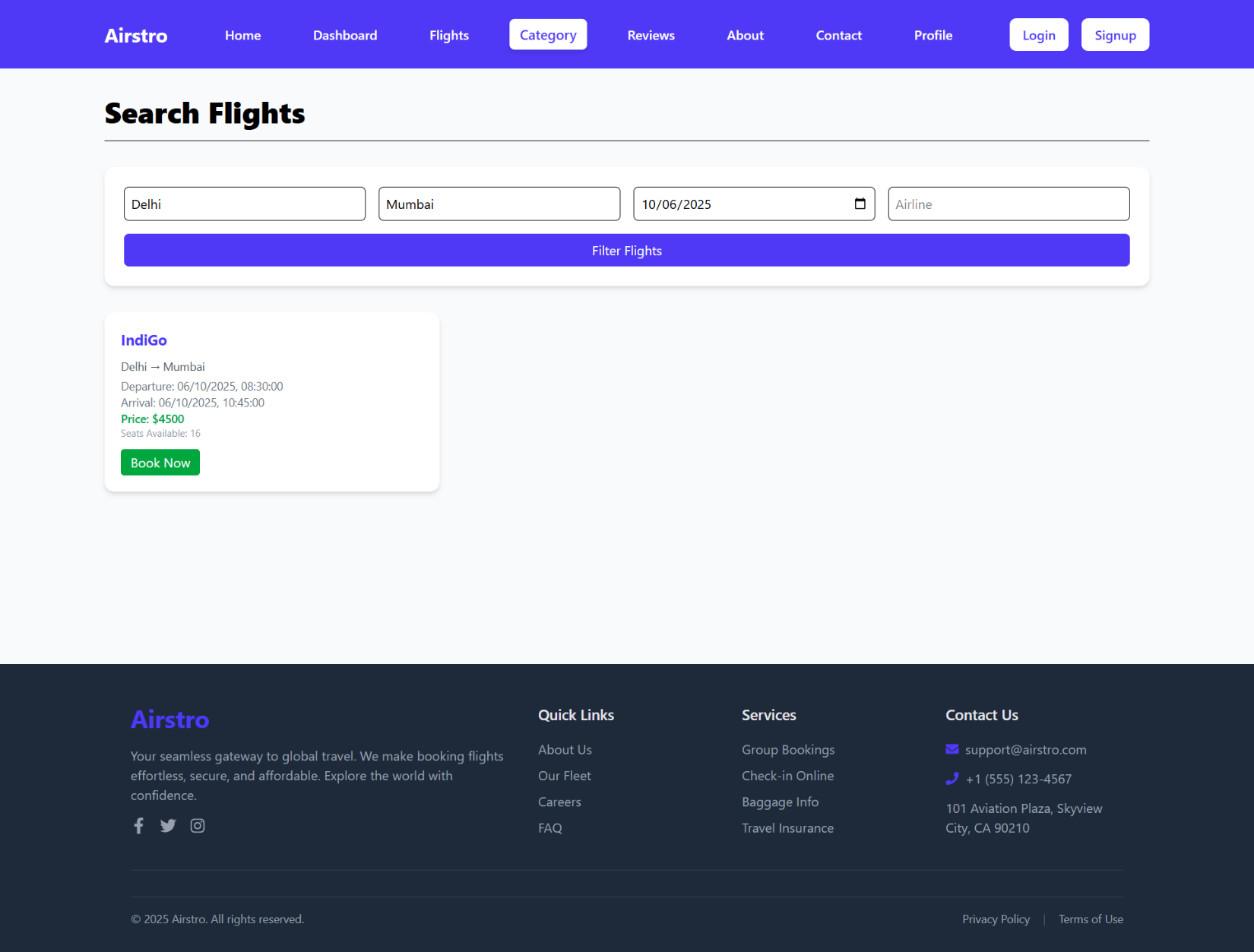
* + **Database & Cache Configuration**
  + Update MongoDB and Redis connection details in the project configuration using environment variables.
  + MONGO\_URI=mongodb+srv://<username>:<password>@cluster0.mongodb.net/yourDB  
    REDIS\_HOST=127.0.0.1  
    REDIS\_PORT=6379
  + **Email Service Setup**
  + Use an official email service (SendGrid, Gmail SMTP, or Mailtrap for testing) for OTPs and notifications. Configure credentials in environment variables:
  + EMAIL\_USER=your-email@example.com  
    EMAIL\_PASS=your-email-password
  + **Environment & Running the Project**
* Store sensitive credentials (MongoDB URI, Redis, and SMTP credentials) in a .env file in the server folder.
* Make sure .env is added to .gitignore to avoid committing sensitive data.

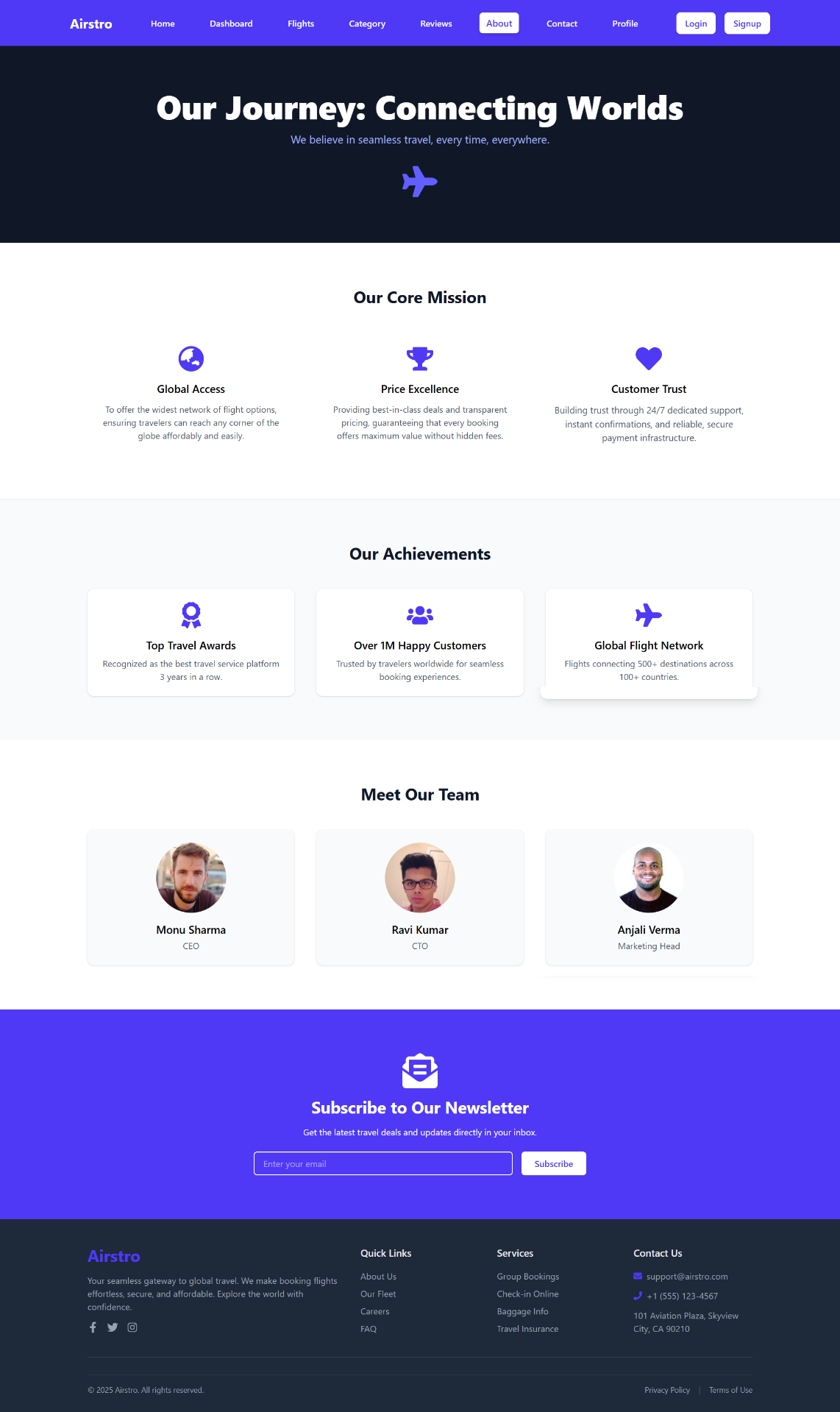






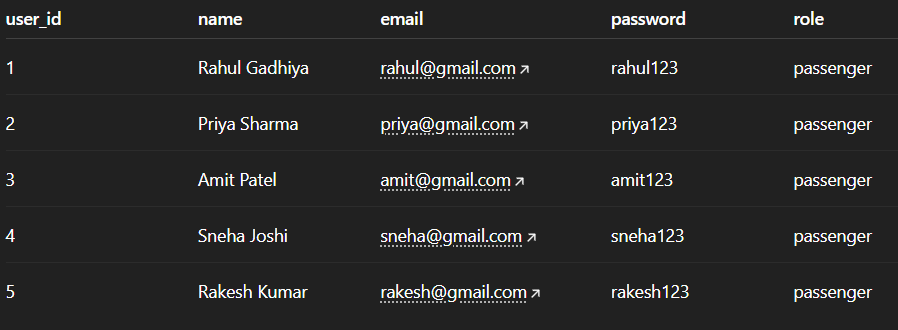




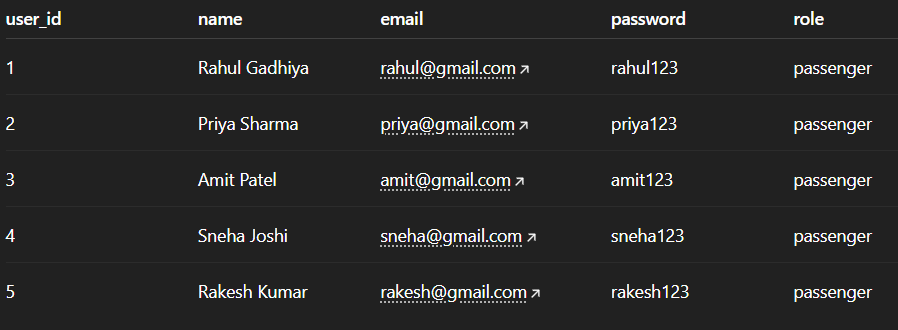


## **PROJECT DATA OBJECT**

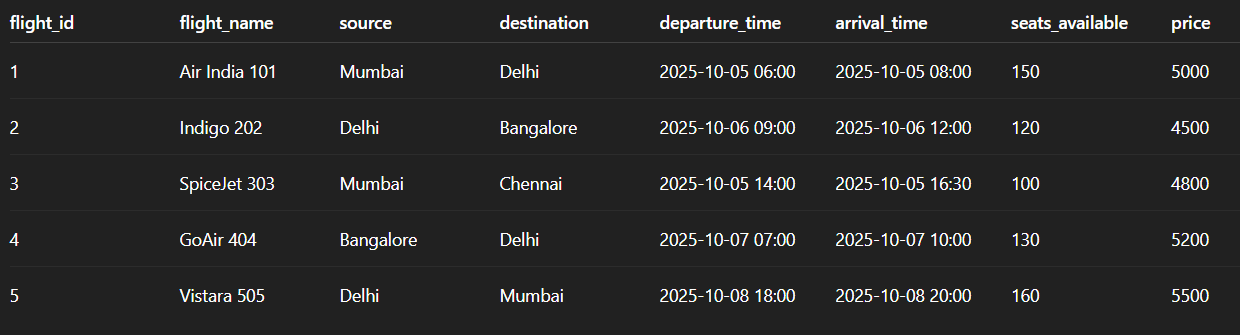
**User** :



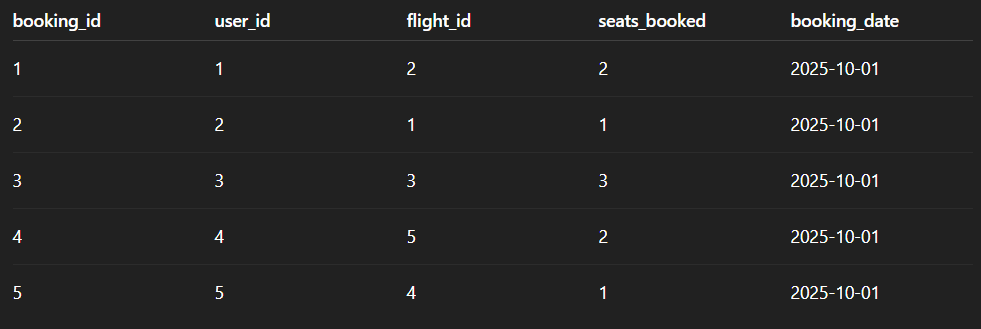
**Admin :**

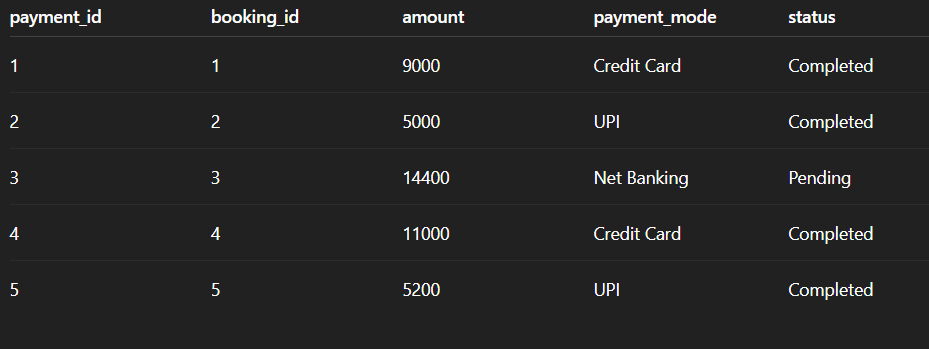


**Flights:**



**Booking:**



**Payment:** 

## **PROJECT SUMMARY/REFERENCES**

**Summary:**

* Airstro is a full-stack car rental web application built with Node.js, Express.js, MongoDB, and JavaScript.
* Users can register, log in, browse cars, and book vehicles for specific time slots.
* Admins can manage cars, update details, and track bookings in real-time.
* The system uses CRUD operations, authentication, and RESTful APIs for seamless functionality.
* It demonstrates a complete, secure, and user-friendly rental management system.

**🔗 References:**

Here are some useful resources that helped in building this project:

* **Node.js Official Documentation :** <https://nodejs.org/en/docs/>
* **Express.js Documentation :** <https://expressjs.com/>
* **MongoDB Documentation :** <https://www.mongodb.com/docs>
* **Mongoose ODM Tutorial :** <https://mongoosejs.com/docs/>
* **JavaScript Basics (MDN) :** <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
* **RESTful API Design (GeeksforGeeks) :** <https://www.geeksforgeeks.org/rest-api-design-best-practices/>
* **CRUD Operations in Node.js & MongoDB (GeeksforGeeks) :** <https://www.geeksforgeeks.org/crud-operations-using-node-js-and-mongodb/>