**Full Stack Development with MERN**

**Project Documentation format**

**1. Introduction**

* **Project Title:** Fly High
* **Team Members:**

1. Parth Suri
2. Adi Seth
3. Aprajita Nandkeuliar
4. Sanjay K S

**2. Project Overview**

* **Purpose:** Fly High is designed to streamline the flight booking process, allowing users to easily search for flights, book seats, and manage their travel itineraries. The primary goal is to provide a user-friendly platform for both passengers and administrators to handle all aspects of flight management efficiently.
* **Features:**

 User registration and login

 Flight search and filter

 Seat selection and booking

 User profile management

 Flight and booking management for administrators

**3. Architecture**

* **Frontend:** The frontend is built using React, a popular JavaScript library for building user interfaces. It follows a component-based architecture, with each UI element encapsulated in its own reusable component. State management is handled using React's built-in hooks and Context API.
* **Backend:** The backend is developed using Node.js and Express.js. It follows a RESTful API design, providing endpoints for all necessary operations such as user authentication, flight search, booking management, and user profile updates. Middleware is used for error handling and authentication.
* **Database:** The database is MongoDB, a NoSQL database known for its flexibility and scalability. The schema includes collections for users, flights, bookings, and comments. MongoDB's aggregation framework is used for complex queries, and Mongoose is used as an ODM to facilitate schema definition and data validation.

**4. Setup Instructions**

* **Prerequisites:**

 Node.js (v14.x or later)

 MongoDB (v4.x or later)

 Git

* **Installation:**

 Clone the repository:

git clone https://github.com/your-username/travel-hub.git

 Navigate to the project directory:

cd travel-hub

 Install dependencies for both frontend and backend:

cd client

npm install

cd ../server

npm install

 Set up environment variables:

Create a .env file in the server directory with the following contents:

env

Copy code

PORT=5000

MONGO\_URI=mongodb://localhost:27017/travel\_hub

JWT\_SECRET=your\_jwt\_secret

**5. Folder Structure**

* **Client:**

client/

├── public/

├── src/

│ ├── components/

│ ├── contexts/

│ ├── pages/

│ ├── services/

│ ├── App.js

│ ├── index.js

│ └──

**Server:**

server/

├── config/

├── controllers/

├── middleware/

├── models/

├── routes/

├── utils/

├── server.js

└── ...

**6. Running the Application**

* Provide commands to start the frontend and backend servers locally.
  + **Frontend:** npm start
  + **Backend:** node index.js

**7. API Documentation**

**Endpoints:**

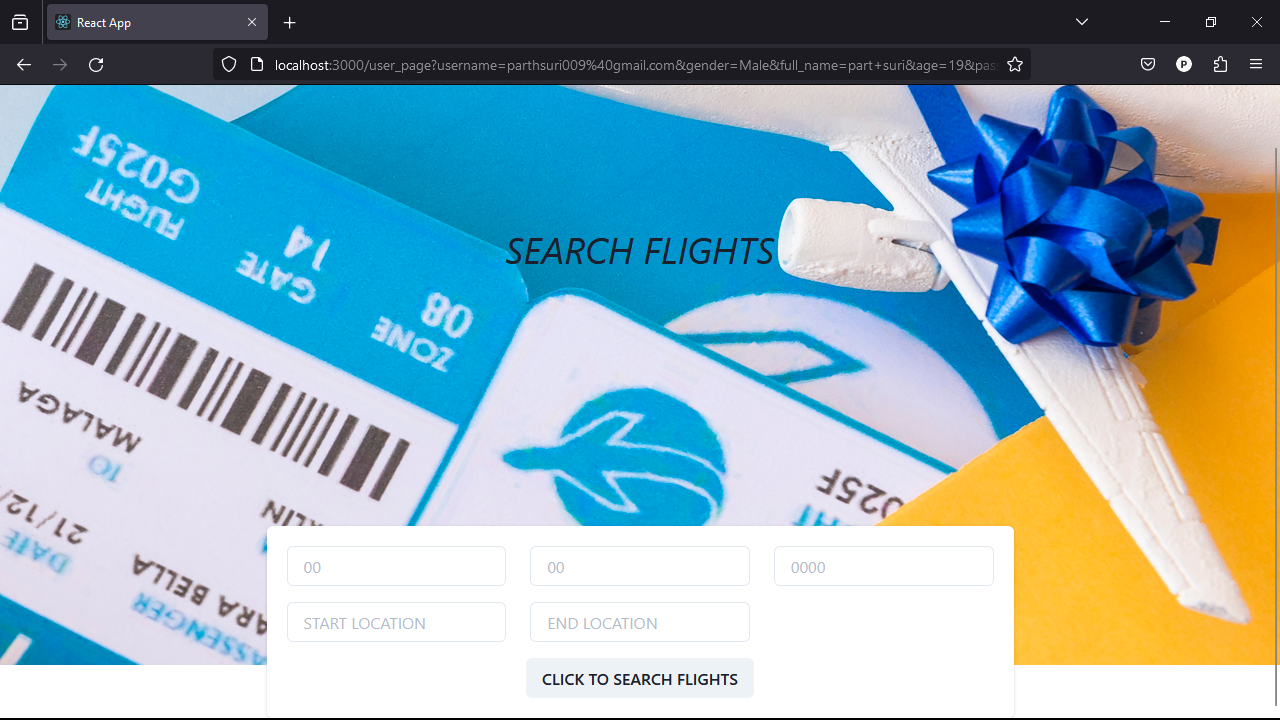
* **User Registration:** POST /api/users/register
  + **Request:** { name, email, password }
  + **Response:** { user, token }
* **User Login:** POST /api/users/login
  + **Request:** { email, password }
  + **Response:** { user, token }
* **Flight Search:** GET /api/flights
  + **Request:** ?origin&destination&date
  + **Response:** [ { flight details } ]
* **Book Flight:** POST /api/bookings
  + **Request:** { flightId, seatNumber, userId }
  + **Response:** { booking details }
* **Get User Profile:** GET /api/users/:id
  + **Request:** None
  + **Response:** { user details }

**8. Authentication**

Authentication is handled using JWT (JSON Web Tokens). Upon successful login, a token is issued and stored on the client-side (usually in localStorage or cookies). Each subsequent request to protected routes includes this token in the Authorization header. The backend verifies the token to authenticate and authorize the user

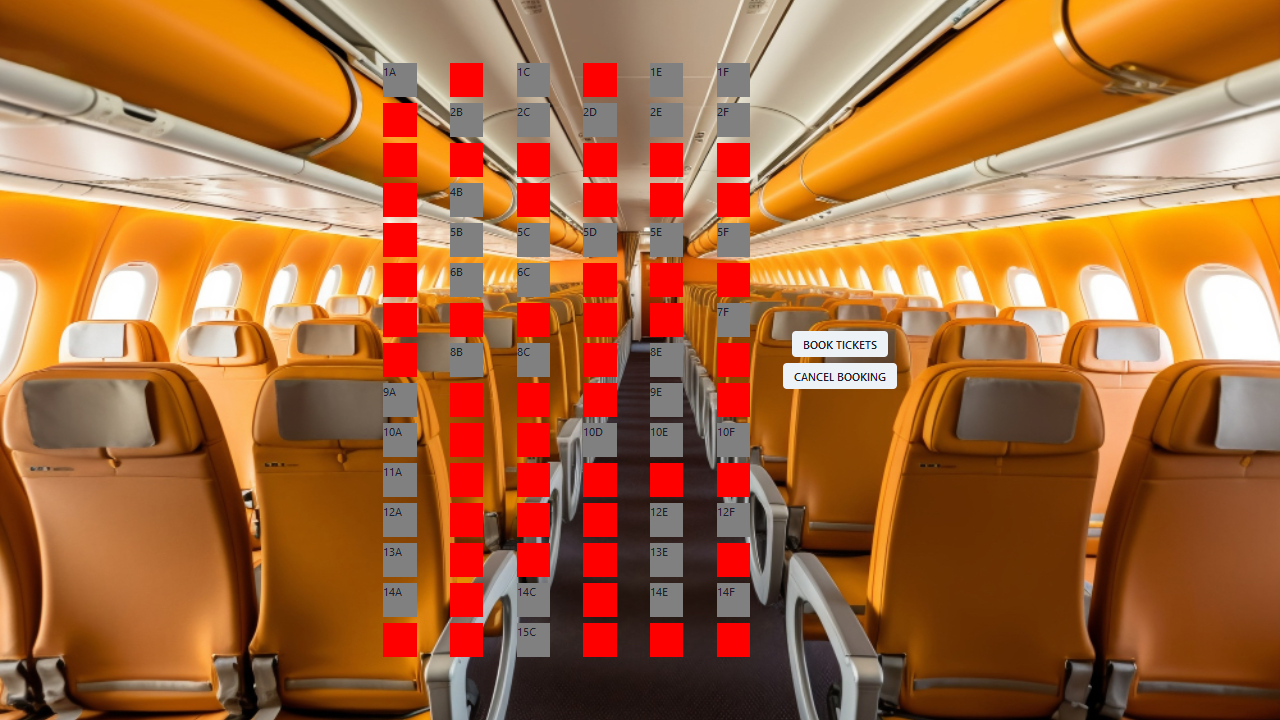
.

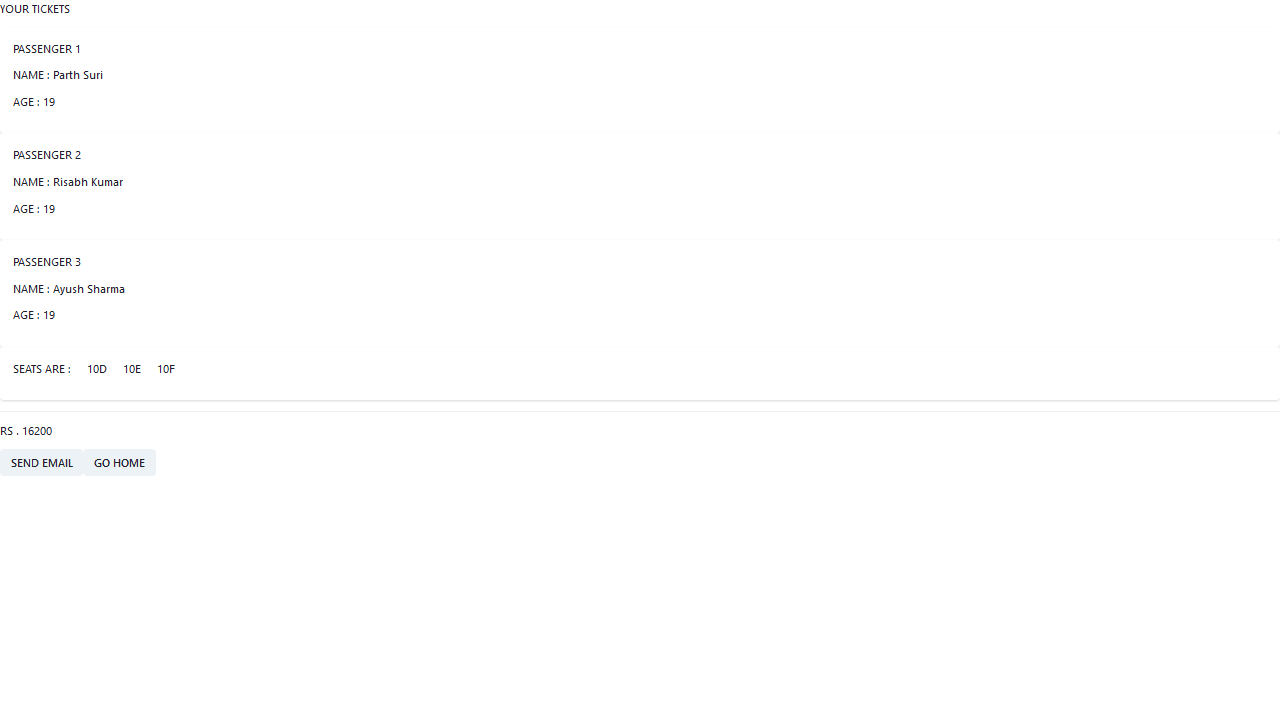
**9. User Interface**



A cartoon of a person and person in an airplane

Description automatically generated





**10. Testing**

**Testing Strategy:**

* Unit tests for individual components and functions using Jest.
* Integration tests for API endpoints using Supertest.
* End-to-end tests for user workflows using Cypress.

**11. Screenshots or Demo**

Link to Demo:

https://drive.google.com/drive/folders/1W-FZzkzFelQcWP44A-vLyU6RVw60DuWY

**12. Known Issues**

 **OTP login not there:** This is currently under investigation.

 **Promocodes are not available to the user:** This requires a page refresh to display the latest data.

**13. Future Enhancements**

* Implement real-time notifications for booking updates.
* Add support for multiple languages.
* Integrate payment gateway for online payments.
* Develop a mobile application for iOS and Android platforms.