**Full Stack Development with MERN**

**Project Documentation**

**Online Flight Booking App** **(DFK)**

**DHAANISH AHMED COLLEGE OF ENGINEERING**

**Department of Computer Science And Engineering**

**Team Members:**

* + - 1. **F.Fazil Ahmed Frontend Developer**
      2. **R.Dinesh Kumar Backend Developer**
      3. **P.Hema Malini Manual/Automated Tester**
      4. **P.K.Jothika UI/UX Developer**

**Submitted to Naan Mudhalavan Team**

**ANNA UNIVERTSITY.**

**UNDER THE GUIDANCE OF**

**PROF S.Saranya**

**1.INTRODUCTION**

**PROJECT TITLE : Online Flight Booking App using MERN STACK**

* **This report provides a comprehensive overview of an online flight booking application developed using the MERN stack. The MERN stack consists of MongoDB, Express.js, React.js, and Node.js, which together provide a powerful and flexible foundation for building modern web applications. This flight booking app allows users to search for flights, register, log in, and book tickets online.**

**2.PROJECT OVERVIEW**

**PURPOSE:**

**The purpose of this flight booking app is to provide users with a convenient and efficient platform for searching, booking, and managing flight tickets online. It aims to simplify the travel planning process by allowing users to easily find available flights, register or log in to their accounts, and complete bookings in a seamless, secure environment. By leveraging the MERN stack, the app ensures a fast, scalable, and responsive experience across devices.**

**The key objectives of DFK are:**

** Flight Search: Allow users to search for flights based on destination, date, and passengers.**

** User Registration & Login: Enable account creation and secure login for personalized access.**

** Flight Booking: Let users select flights and book tickets easily.**

** Secure Payment: Integrate a safe payment gateway for online transactions**

**FEATURES**

**1. User Registration and Login: Users can register by creating an account, log in, and access their booking history.**

**2. Flight Search: Allows users to search for flights by entering departure and destination cities and selecting travel dates.**

**3. Booking System: Users can book flights after selecting the desired journey options.**

**4. Return Journey Option: Option to book round trips for added convenience.**

**5. Responsive Design: User interface is optimized for various devices, providing a seamless experience across desktops, tablets, and smartphones**

**3. ARCHITECTURE**

**Frontend:**

* The frontend is built using React.js, featuring a component-based architecture that promotes reusability and maintainability. It utilizes React Router for navigation and Redux for state management.

**Backend:**

* The backend is developed using Node.js and Express.js, providing a RESTful API that handles client requests, business logic, and data processing.

**Database:**

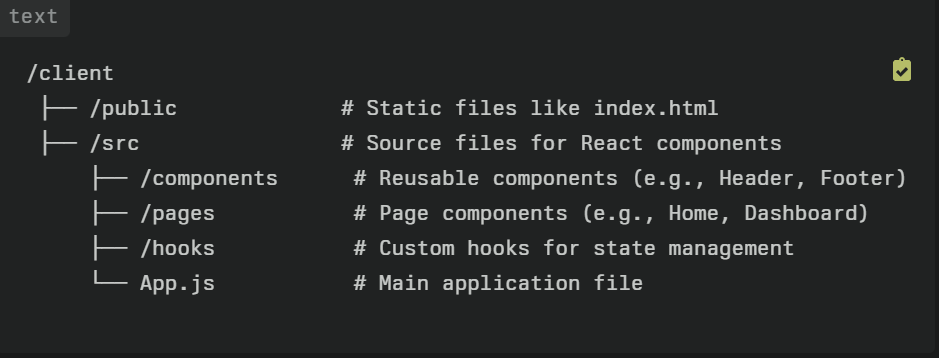
* MongoDB is used as the database to store user profiles, task details, and collaborative project information. Mongoose is employed for object modeling and schema validation.

**4. SETUP INSTRUCTIONS**

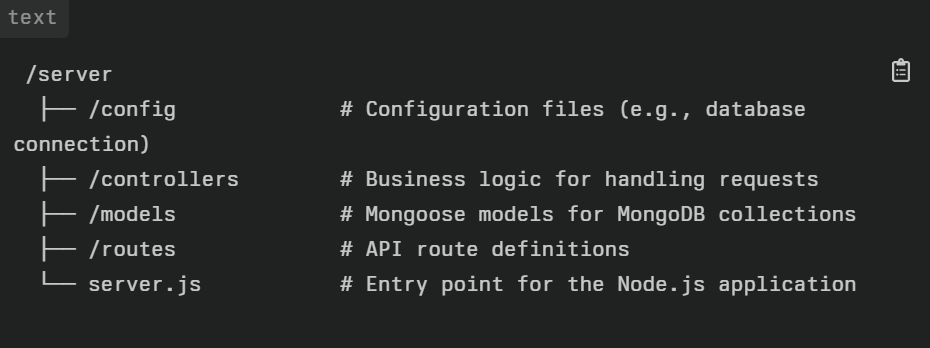
* **Prerequisites:**
* Node.js
* MongoDB
* **Clone the repository:**
* git clone https://github.com/Fazil0070/Flight\_booking\_app.git
* **Navigate to the client directory:**
* cd Flight\_booking\_app-main /client
* **Install frontend dependencies:**
* npm install
* **Set up environment variables (create a .env file):**
  + - MONGODB\_URI=your\_mongodb\_uri
    - JWT\_SECRET=your\_jwt\_secret
* **Navigate to the server directory:**
* cd ../server
* **Install backend dependencies:**
* npm install

**5. FOLDER STRUCTURE**

* **Client Directory Structure**:



* **Server Directory Structure:**



**6. RUNNING THE APPLICATION**

* Provide commands to start the frontend and backend servers locally.
  + **Frontend:** **npm start** in the client directory.
  + **Backend:** **npm start** in the server directory.

**7. API DOCUMENTATION**

* **GET /api/tasks**: Retrieve all tasks for the authenticated user.
* **POST /api/users/register**: Register a new user.
* **POST /api/users/login**: Authenticate a user.
* **POST /api/tasks**: Create a new task.
* **PUT /api/tasks/:id**: Update an existing task.
* **DELETE /api/tasks/:id**: Delete a task.

**8. AUTHENTICATION**

* Authentication is handled using JWT (JSON Web Tokens) for secure user login and session management, where tokens are stored in HTTP-only cookies or local storage. Authorization is managed by verifying user roles and permissions based on the JWT claims to control access to protected routes and resources.

**9. USER INTERFACE**

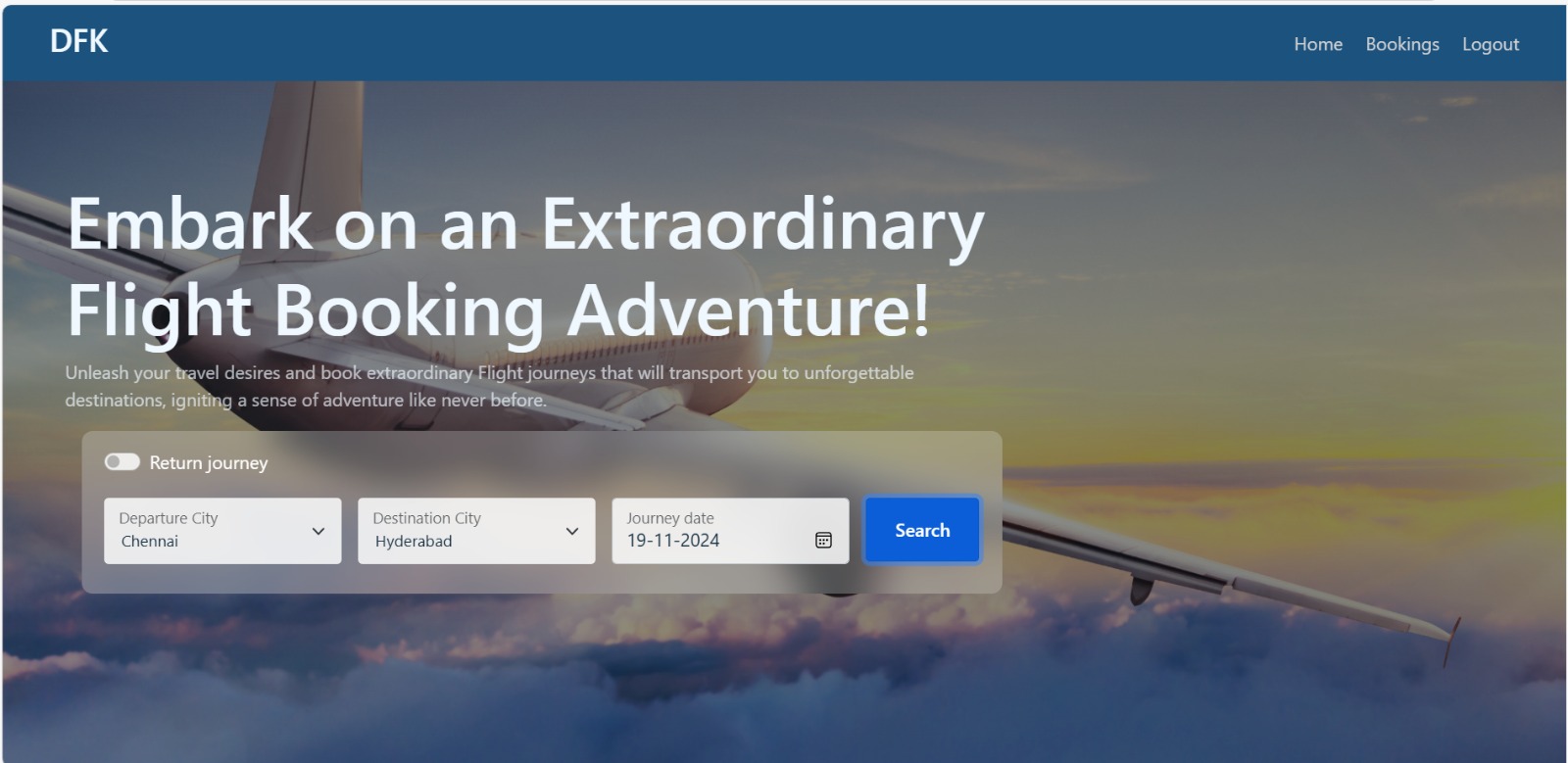
* Provide screenshots or GIFs showcasing different UI features.
* Responsive Design
* Intuitive Navigation
* Product Display
* Product Filtering and Sorting
* Product Detail Page
* Shopping Cart Interface
* Checkout Process

**10. TESTING:**

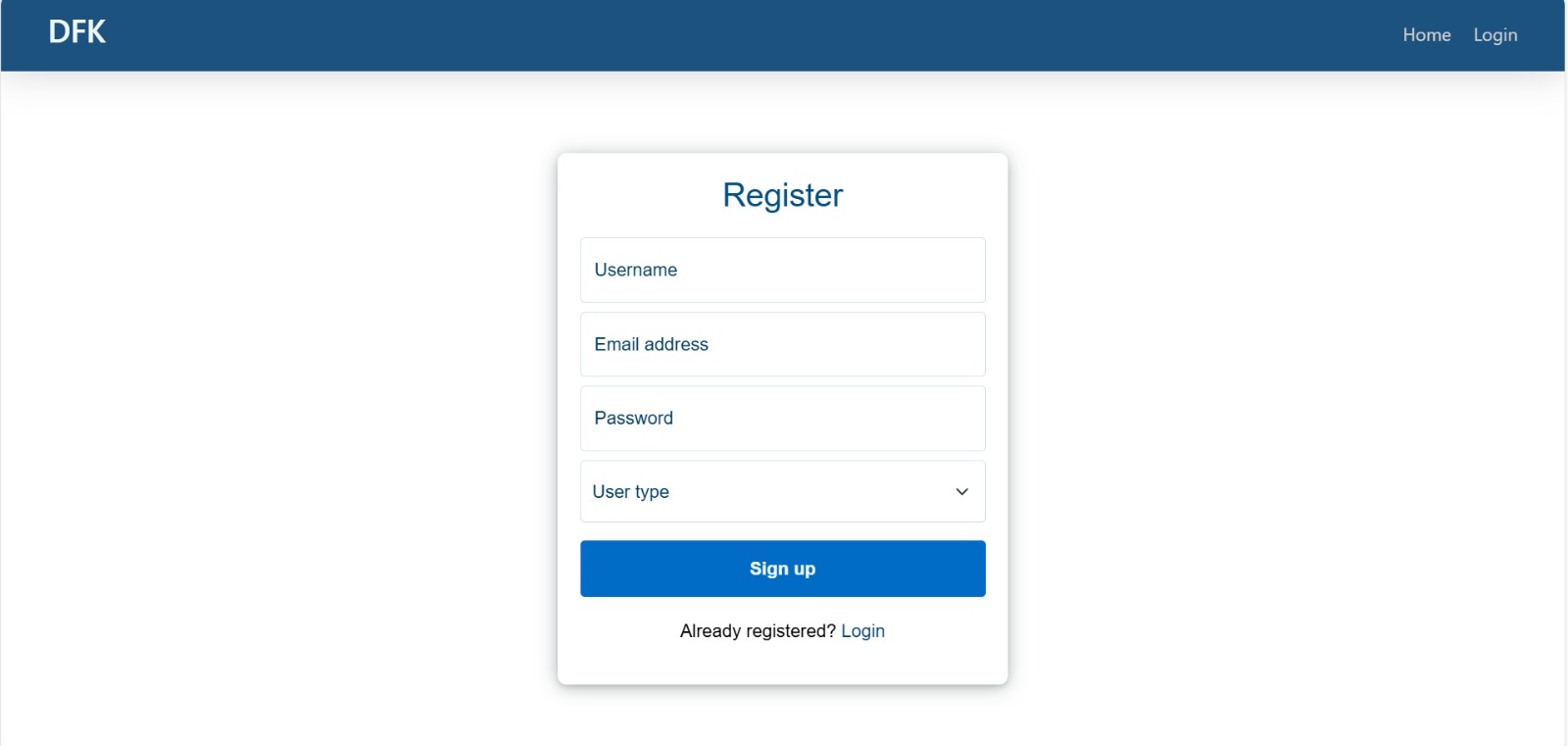
* **Exploratory Testing**
  + Conduct unscripted testing sessions to discover unexpected issues and assess user experience.
* **User Acceptance Testing (UAT)**
  + Engage end-users to validate that the application meets their expectations and requirements before launch**.**
* **Cross-Browser Compatibility Testing**
  + Verify that the site functions correctly across different web browsers (e.g., Chrome, Firefox, Safari) to ensure a consistent user experience**.**
* **Device and Responsive Testing**
  + Test the application on various devices (desktops, tablets, smartphones) to ensure it is fully responsive and functional**.**
* **Performance Testing**
  + Assess the application’s performance under various conditions, including load times and responsiveness during peak usage scenarios**.**

**11. SCREENSHOTS OR DEMO**

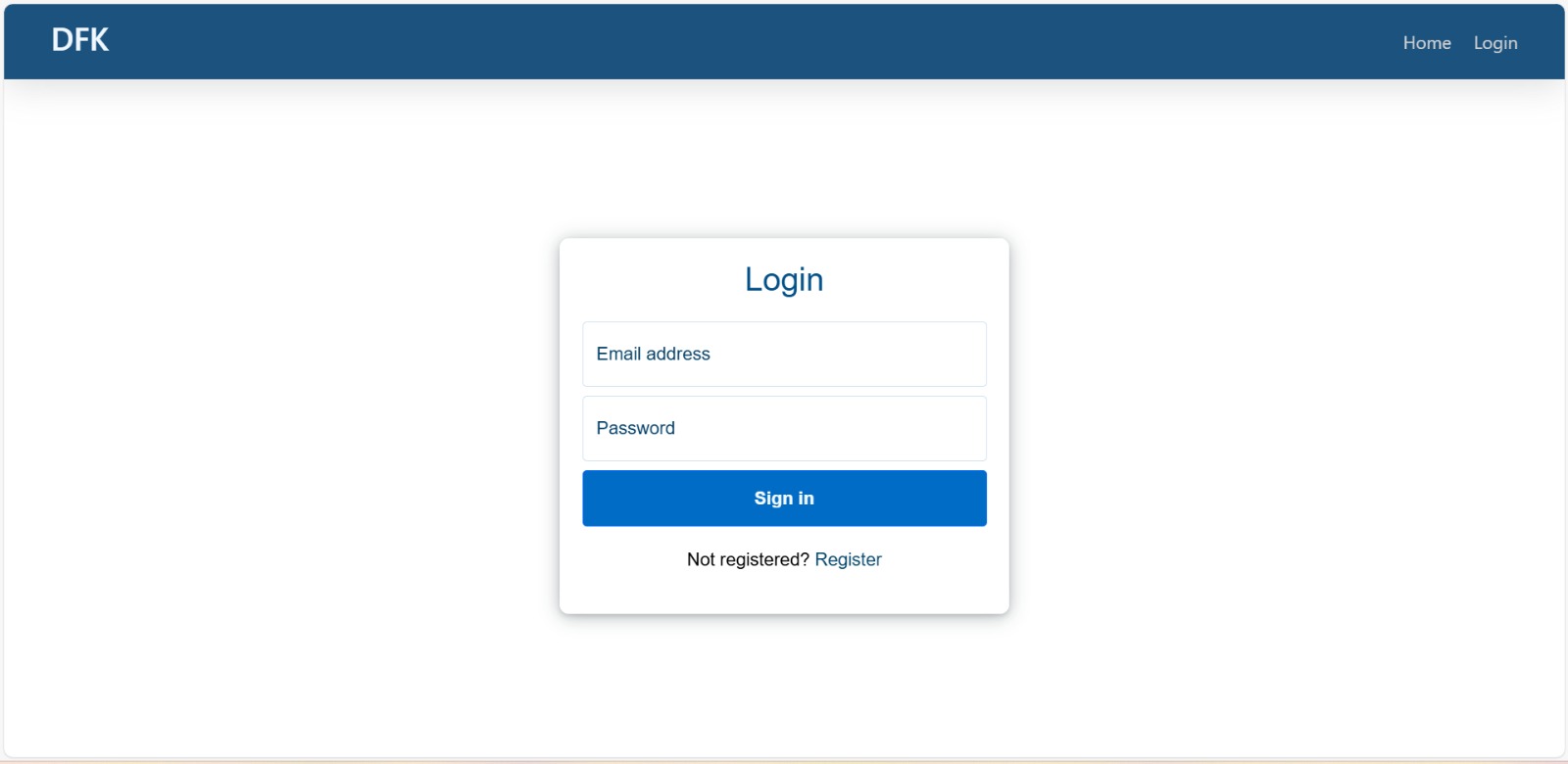
**Home Page:**

****

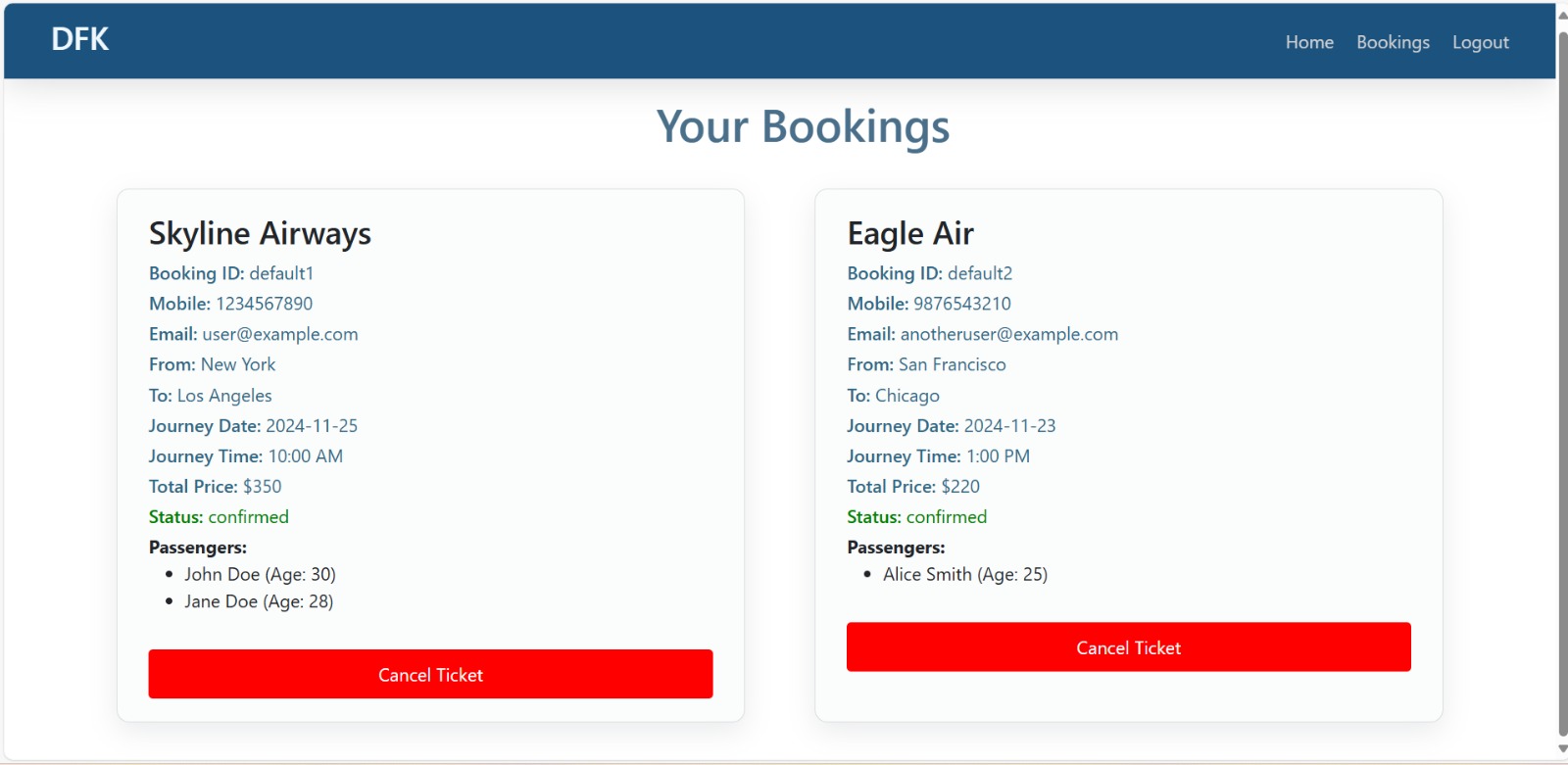
**Sign up Page:**

****

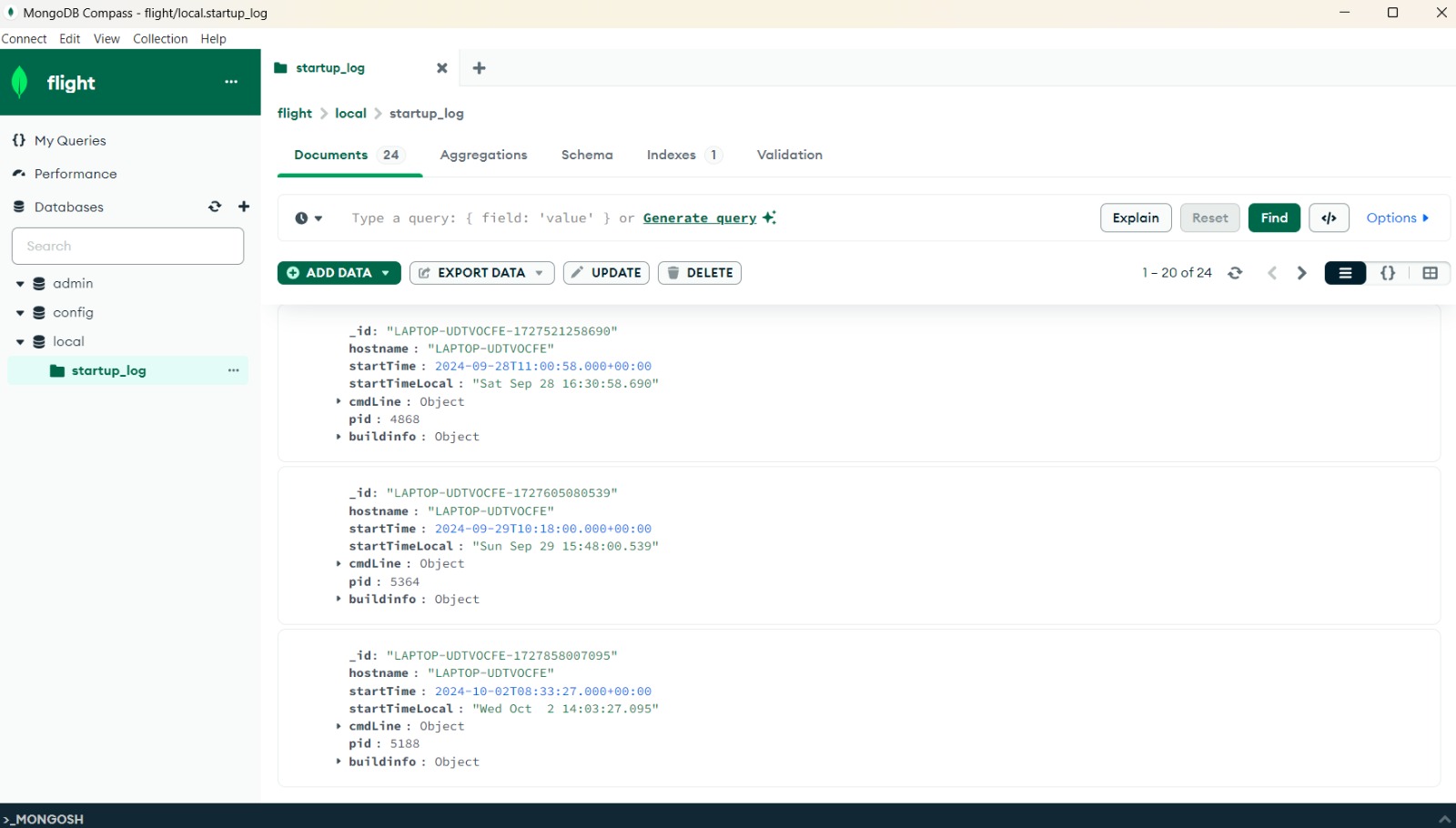
**Sign in:**

****

**Booking Page:**

****

**Database:**

****

**12. KNOWN ISSUES**

* In the context of software development, particularly for applications built using the MERN stack, the inability to explore the application through automated testing can present significant challenge for us**.**

**13. FUTURE ENHANCEMENTS**

“ To enhance the flight booking app, features like flight comparison and advanced filters can be added to help users find the best options. Real-time notifications for flight status and booking updates will keep users informed. Integrating a loyalty program would reward frequent travelers, while multi-language and currency support would make the app more accessible globally. Additionally, customer reviews and ratings would help users make informed decisions about airlines and flights.”

**GITHUB REFERENCE LINK :**

https://github.com/Dineshkumar12300/NM-Flight\_Booking\_App/upload