**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 26 -052025 |
| Team ID | LTVIP2025TMID46399 |
| Project Name | FlightFinder |
| Maximum Marks | 4 Marks |

Solution Architecture – FlightFinder

The solution architecture of FlightFinder is designed to deliver a secure, responsive, and scalable online flight booking registration and management system that effectively addresses the needs of travelers, airline representatives, and admins. It follows a modular client-server architecture built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and is capable of future enhancements such as real-time communication and AI-based routing.

**Key Architectural Components:**

***Frontend (Client Interface – React.js):***

* Built with React.js and styled using Bootstrap + Material UI
* Fully responsive UI supporting mobile, tablet, and desktop views
* Role-based components (User, Agent, Admin) rendered based on login session
* Axios used for asynchronous communication with backend APIs
* Animations via AOS (Animate On Scroll) for smoother traveler experience

***Backend (Server Logic – Node.js + Express):***

* RESTful APIs developed using Express.js
* Handles authentication, session management, flight booking routing, booking history, and admin logic
* Implements business logic for:
  + Flight Booking registration
  + Flight Booking status updates
  + Seat selection
  + Travel class filtering
* Middleware ensures secure routes and error handling

***Database (MongoDB + Mongoose):***

* NoSQL database to store flight bookings, travelers, airline representatives, booking booking messages, etc.
* Mongoose schemas for:
  + Users (name, phone, role)
  + Flight Bookings (category, description, status, timestamps)
  + Assigned flight bookings (airline representative mapping)
  + Chat/Feedback
* Indexed for fast querying and performance

***Authentication & Authorization:***

* JWT-based authentication ensures secure access
* Role-based authorization:
  + Users can submit/view flight bookings
  + Agents can view assigned tickets and respond
  + Admins have full control over data and routing
* Login sessions stored securely

***Real-Time & Notification Layer (Planned):***

* Integration with Socket.io for real-time updates & booking
* WebRTC-based support call option for flight booking escalation (future enhancement)
* Notifications via email/SMS on flight booking status changes

***Admin Dashboard and Analytics:***

* View all flight bookings with filters (pending/resolved)
* Assign airline representatives based on workload
* View platform metrics: booking confirmation time, flight booking counts, airline representative performance
* Export data for internal use/reporting

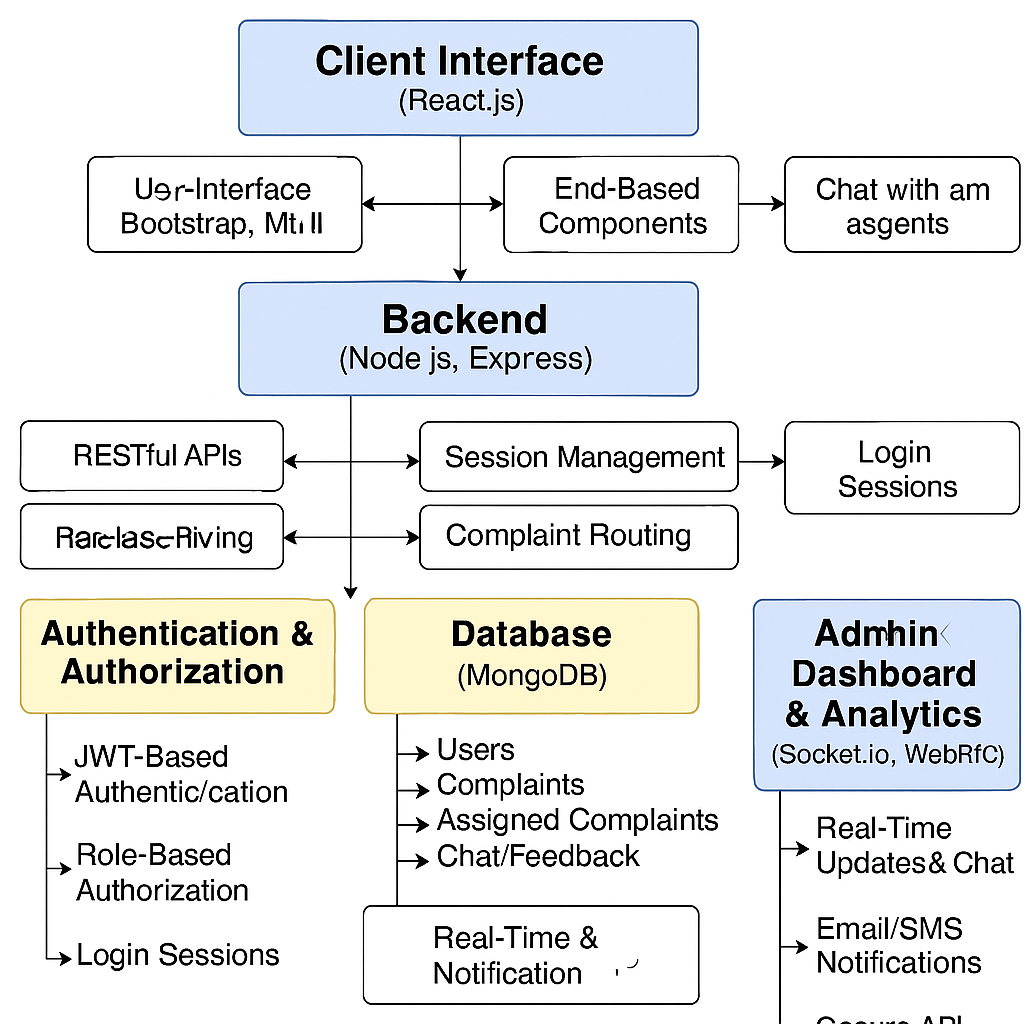
***Deployment & Scalability:***

* Hosted on Render or Railway for both frontend and backend
* .env used for environment config and API security
* Easily scalable architecture supports multiple departments or city-wide expansion
* API-driven design supports integration with external government systems or legacy tool

***Security Considerations:***

* Secure routing with encrypted tokens
* Input validation and rate limiting to prevent spam
* Role-based route protection

**Example - Solution Architecture Diagram:**



The solution architecture of FlightFinder is built using a modular, scalable, and role-based client-server model. It integrates a React.js frontend for a responsive and interactive traveler experience, and a Node.js + Express.js backend to manage core logic, authentication, and flight booking processing. MongoDB serves as the database to store traveler, flight booking, and resolution data in a flexible and efficient structure.

Authentication is handled through secure JWT tokens, enabling role-based access for travelers, airline representatives, and admins. The architecture also includes dashboards tailored to each role, allowing seamless flight booking submission, status tracking, airline representative assignment, and booking monitoring. Additionally, the system is designed to support real-time communication using technologies like Socket.io and WebRTC, enabling future features like live booking and video support. With this setup, FlightFinder delivers a secure, responsive, and future-ready platform for streamlined flight booking management across travel-related use cases