NAAN MUDHALVAN PROJECT REPORT

FLIGHT BOOKING APP USING MERN STACK

Team Lead

- V.HARISH KUMAR

Team Member

- A. NAVEEN
- V.BALAJI
- R. KARTHICK

Project Introduction:

A seamless flight booking app that connects travelers to their dream destinations with ease. Built on the robust MERN stack, Sky Link offers an intuitive interface to search flights, view seat availability, and manage bookings all in one place. With secure payment options and real-time booking status, Sky Link is designed for a smooth, worry-free travel experience. Whether you're planning a weekend getaway or a business trip, Sky Link gets you there – swiftly and effortlessly!

Tools And Technology used:

Flight Booking App with MERN Stack

Built with the powerful MERN stack (MongoDB, Express, React, Node.js), this Flight Booking App offers a seamless booking experience with real-time flight data, user-friendly interfaces, and secure payment integration. MongoDB ensures fast data retrieval for flights, users, and

bookings, while Express and Node.js provide a robust backend for handling requests and managing data flow. React powers the responsive, interactive frontend, giving users a smooth experience across devices. With its scalable architecture and efficient API management, this app is ready to support millions of users and global operations effortlessly.

Developing a Flight Booking App:

Developing a Flight Booking App using the MERN stack involves a structured, agile approach to ensure seamless functionality and user experience. Here's a streamlined process outline.

- ✓ **Planning & Requirement Analysis**: Define core features like flight search, booking, payment integration, user profiles, and admin controls. Gather insights on the user journey and map out all required functionalities.

- ✓ **Design**: Craft an intuitive, visually engaging UI with wireframes and a sleek user interface. Design the ER diagram to structure the data flow, ensuring efficient relationships between users, flights, bookings, and payments.

- ✓ **Frontend Development**: Use **React** to create responsive pages for search, booking, and user profiles. Implement seamless navigation, booking workflows, and payment interactions, ensuring high responsiveness and UX standards.

- ✓ **Backend Development**: Build a robust **Node.js** and **Express** backend to handle APIs for user authentication, flight management, and booking logic. Ensure secure handling of data and support scalability.

- ✓ **Database Design & Integration**: Use **MongoDB** to store and manage data, with collections for users, flights, bookings, payments, and more. Optimize for high performance, supporting large volumes of data without lag.

- ✓ **Testing & Debugging**: Perform comprehensive testing to ensure all modules work smoothly. Use unit and integration tests to verify accuracy and catch potential bugs.

- ✓ **Deployment & Maintenance**: Deploy on cloud services like AWS or Heroku, configuring CI/CD pipelines for regular updates. Monitor performance, gather user feedback, and roll out enhancements to improve the app continuously.

Folder Structure:

- ➢ Client Folder: The client side manages the React application, including UI components, pages, and services for API requests. This structure promotes reusability and scalability, ensuring a smooth user experience.

- ➢ Server Folder: The server side is a Node.js/Express backend, organizing configurations, controllers, and routes for clean and efficient data handling. Using this layered architecture

enables separation of concerns, allowing easy updates and maintenance.

Pre Requisites:

```
/flight-booking-app
├── /client            # Frontend - React App
│   ├── /public        # Static assets, index.html
│   ├── /src
│   │   ├── /assets    # Images, icons, fonts
│   │   ├── /components  # Reusable components  - NavBar, SearchBar
│   │   ├── /pages     # Main pages - Home, Booking, Profile
│   │   ├── /services  # API calls to server
│   │   ├── /redux     # Redux store, actions, reducers
│   │   ├── App.js     # Main app component
│   │   └── index.js   # Entry point
│   └── package.json   # Client dependencies
│
├── /server            # Backend - Node/Express
│   ├── /config        # Config files - DB connection, environment variables)
│   ├── /controllers   # Route handlers - BookingController
│   ├── /models        # Mongoose schemas - User, Flight, Booking
│   ├── /routes        # API routes  -/api/users, /api/flights
│   ├── /middleware    # Auth and validation middleware
```

```
│   ├── /utils          # Utility functions, helpers
│   ├── server.js       # Server entry point
│   └── package.json    # Server dependencies
│
├── .env                # Environment variables
└── README.md           # Project documentation
```

Frontend (React.js):

- **HTML, CSS, and JavaScript**: Understanding of the fundamentals of web development.
- **React.js**: Familiarity with React concepts like
- **Components** (functional and class-based)
- **State and Props**
- **Hooks** (use State, Use Effect, etc.)
- **React Router** for navigation
- **Context API** or **Redux** for state management
- **Component lifecycle** and **Event handling**

Backend (Node.js & Express.js):

- **Node.js**: Knowledge of JavaScript on the server side.
- **Express.js**: Familiarity with this framework to handle HTTP requests and manage middleware.
- **RESTful API principles**: Understanding of how to design and structure REST APIs (GET, POST, PUT, DELETE).
- **Authentication**: Familiarity with user authentication using **JWT (JSON Web Tokens)**, **sessions**, or **OAuth** for secure logins.
- **Middleware**: Understanding of how to write middleware functions (e.g., for authentication, logging, error handling).

Project Structure:

Application Flow

User:

- User Visits the Homepage
- User Searches for Flights
- User Selects a Flight
- User Logs In or Signs Up
- User Makes a Booking
- User Pays for the Booking

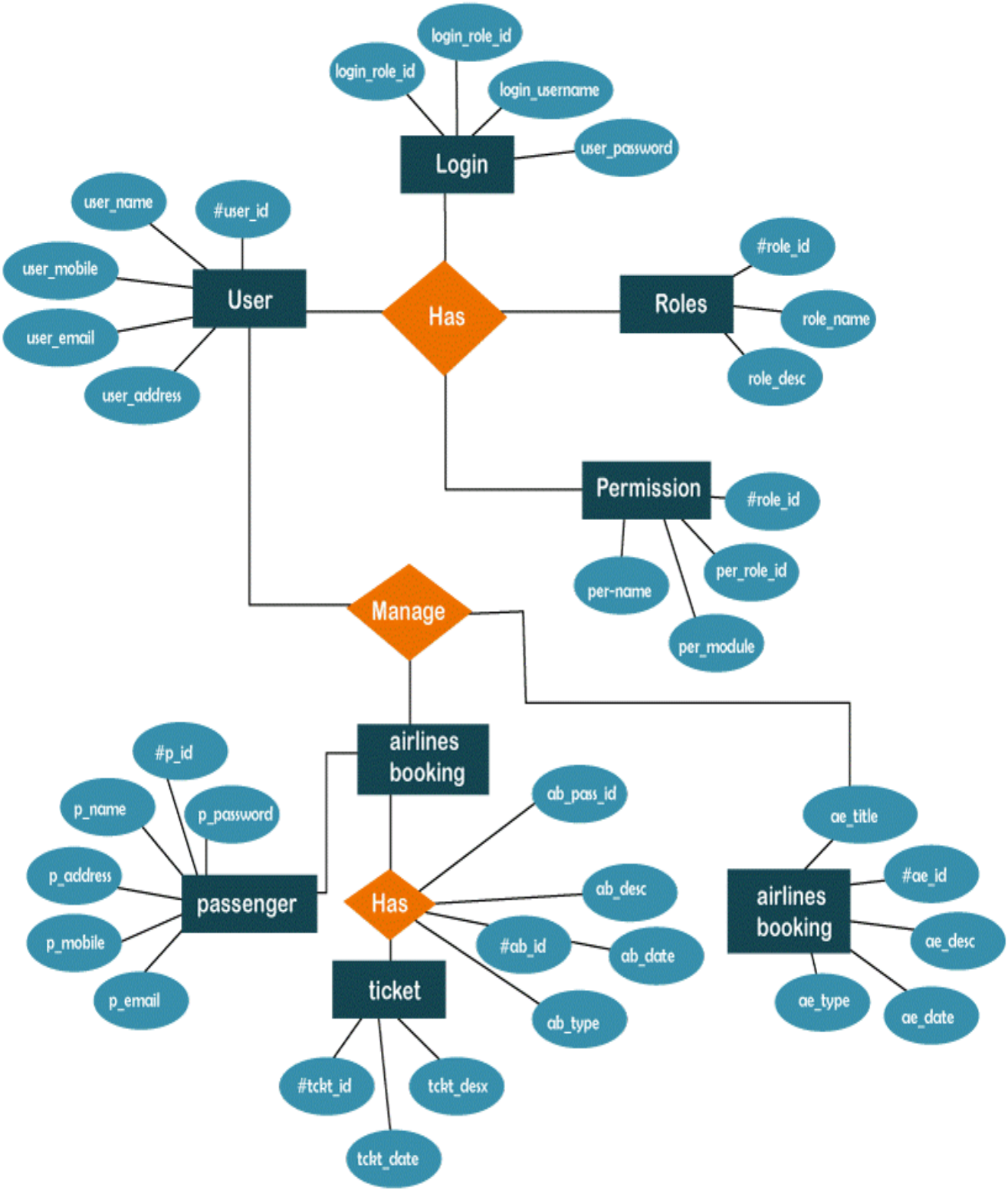
Admin Panel:

- Manages all bookings
- Add new flight and services
- Monitor user activities

Key Features:

- **Airline Booking**: Associated with a **Flight**, has one **Payment**.
- **Flight**: Departs from one **Airport**, operated by one **Airline**.
- **Ticket**: Assigned to a **Passenger** and a **Seat**.
- **Payment**: Linked to a **Booking**.
- **Seat**: Assigned to a **Ticket** and a **Flight**.
- **Passenger**: Can book multiple **Bookings**.

ER Diagram :


login_role_id
login_role_id
login_username
user_password
Login
user_name
#user_id
user_mobile
User
user_email
user_address
Has
#role_id
Roles
role_name
role_desc
Permission
#role_id
per-name
per_role_id
per_module
Manage
airlines booking
#p_id
p_name
p_password
p_address
passenger
p_mobile
p_email
Has
ab_pass_id
ab_desc
#ab_id
ab_date
ab_type
ticket
#tckt_id
tckt_desx
tckt_date
ae_title
airlines booking
#ae_id
ae_desc
ae_type
ae_date

Conclusion for Flight Booking App:

The Flight Booking App built with the MERN stack is a robust, user-centric platform that seamlessly connects travelers to flights, offering a secure, streamlined booking experience. By leveraging MongoDB, Express, React, and Node.js, the app ensures high performance, scalability, and a responsive interface across devices. This project not only simplifies the booking process but also demonstrates the power of modern web development, delivering an impactful, real-time solution for the travel industry. With continuous updates and user-focused enhancements, this app has the potential to redefine convenience in flight booking.

Team Contribution for Project:

Team Lead- V. HARISH KUMAR Guided the team throughout the project lifecycle, from planning and design to development and deployment. It also provided critical input in decision-making, problem-solving, and code review to maintain high-quality standards.

Team Members:

V. BALAJI gave full contribution in frontend Development , he plays a critical role on creating an intuitive, visually appealing interface.

A. NAVEEN made an invaluable contribution to both the frontend backend development of the Flight Booking App, ensuring stylings, robust data handling and seamless integration of core features.

R. KARTHICK played a crucial role in the ER design and error handling for the Flight Booking App, ensuring a well-structured database and smooth, reliable user interactions