FULL STACK DEVELOPMENT WITH MERN

FLIGHT BOOKING APPLICATION

PROJECT REPORT

Team Members

1. **Diya Arshiya – Roll Number: 2021115033**
2. **Preetham Vijayandhran – Roll Number: 2021115077**
3. **Samiukktha Dharmalingam – Roll Number: 2021115089**
4. **Sathyadharini Srinivasan – Roll Number: 2021115097**

FlightBookingApp

An efficient and interactive web application designed to simplify flight booking operations, tailored for end-users to search and book flights and for administrators to manage flight schedules and bookings.

Purpose

To build a platform for convenient flight management, allowing users to search flights and make reservations while administrators efficiently manage data and streamline operations.

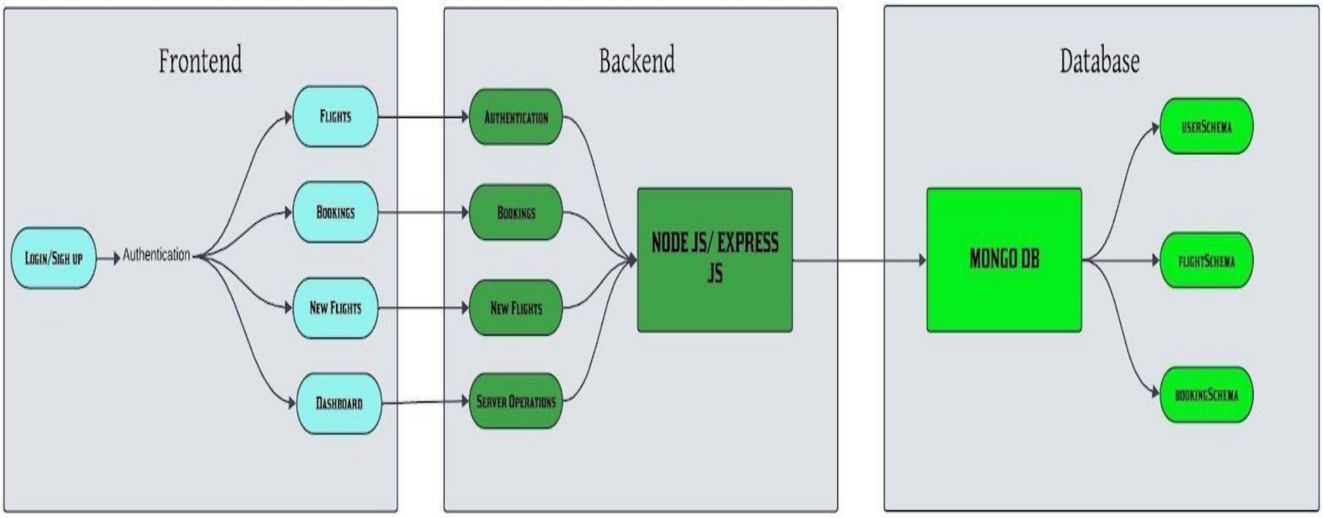
Vision

To deliver a secure, scalable, and user-centric application for smooth flight booking and management processes.

Objectives

- Facilitate real-time flight searching and booking functionalities for users.
- Provide a robust admin panel to manage flight details, view bookings, and access customer information.
- Build a responsive interface ensuring seamless performance across devices.
- Incorporate secure authentication for user and admin roles.

Architecture



Admin Features

1. **Dashboard**: Centralized management for viewing bookings and analytics.
2. **Flight Management**: Add, update, or delete flight details dynamically.
3. **User Management**: Access and manage user data securely.

System Architecture

The application adheres to a **three-tier architecture**:

1. **Frontend**: Built with React.js, delivering a dynamic, responsive user experience.
2. **Backend**: Developed using Node.js with Express.js, handling business logic and data processing.
3. **Database**: MongoDB stores and manages flight schedules, user information, and booking data.

Workflow

1. **User Journey**
   - Enter details → Search flights → Book flight → Confirm booking → View details
2. **Admin Journey**
   - Login → View dashboard → Manage flights and bookings

Technologies and Tools

Frontend

- **Framework**: React.js for reusable components.
- **Styling**: Dynamic CSS styling for adaptive design.

Backend

- **Framework**: Node.js with Express.js
- **Features**:
  - REST API for frontend-backend communication.
  - Secure and scalable middleware handling.

Database

- **Database Type**: MongoDB (NoSQL).
- **Hosting**: Cloud database hosted on MongoDB Atlas.

Key Implementation

JWT Authentication

The system uses JSON Web Tokens (JWT) for user authentication. Both users and admins have distinct roles with role-based access control to ensure secure operations.

Code Snippet

User Login Endpoint

javascript

Copy code

```
app.post('/login', async (req, res) => {
    const { email, password } = req.body;
    const user = await User.findOne({ email });
    if (user && bcrypt.compareSync(password, user.password)) {
        const token = jwt.sign({ id: user._id }, process.env.JWT_SECRET, { expiresIn: '1d' });
        res.status(200).json({ token, user });
    } else {
        res.status(401).json({ error: 'Invalid credentials' });
    }
});
```

Future Enhancements

- **Multi-language Support:** Cater to a global audience with language options.
- **Improved Flight Filtering:** Provide advanced filtering options for precise search.
- **Wishlist and Price Alerts:** Allow users to save preferences and receive price notifications.
- **User Reviews and Ratings:** Enable user feedback to improve decision-making.
- **Offline Mode:** Allow viewing past searches and bookings without internet.
- **Customer Support Chatbot:** Integrate AI for 24/7 support and assistance.