

Project Design Phase

Solution Architecture

Date	15 February 2025
Team ID	
Project Name	Flight booking system
Maximum Marks	4 Marks

Flight Booking System

Solution architecture is a structured approach to design and implement a technology solution that addresses a defined business problem. For the Flight Booking System, the architecture is designed to ensure efficiency, scalability, and usability for users booking flights online.

Goals of the Solution Architecture

1. Find the Best Tech Solution to Solve the Problem

The solution uses a MERN-like stack (MongoDB, Express.js, React.js, Node.js) to deliver a seamless, full-stack application. This ensures both speed and flexibility while handling user bookings, searches, and payment flows.

2. Describe the Structure and Behavior

- **Frontend:** Built with React.js and styled using Tailwind CSS, ensuring a modern, responsive, and interactive UI.
- **Backend:** Developed using Node.js and Express.js to provide RESTful APIs for handling flight data, user authentication, and transactions.
- **Database:** MongoDB stores all flight, user, and booking data in a scalable NoSQL format.
- **Deployment:** Hosted using Vercel for the frontend and possibly Render/Heroku or similar for the backend (based on typical setups).

3. Define Features, Phases, and Requirements

Key Features:

- **User registration and login**
- **Real-time flight search**
- **Booking management**
- **Responsive UI for mobile and desktop**
- **Admin/CRUD capabilities for flight data**

Phases:

- **Phase 1: Basic UI, static data rendering**

- Phase 2: API integration and flight search
- Phase 3: Booking functionality and database connection
- Phase 4: Admin panel, validation, and deployment

Requirements:

- Secure user authentication
- Real-time data rendering
- Efficient backend communication
- Modern responsive design

4. Provide Clear Specifications

- Authentication: JSON Web Token (JWT) for secure sessions
- APIs: REST APIs for flight listings, bookings, and user operations
- UI Components: Reusable components built in React for modularity
- Security: Form validation, HTTPS, and secure storage practices
- Scalability: MongoDB's document model allows horizontal scaling
- Version Control: GitHub repository for collaborative development

Example - Solution Architecture Diagram:

