**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 31 January 2025 |
| Team ID | LTVIP2025TMID55660 |
| Project Name | FlightFinder: Navigating Your Air Travel Options |
| Maximum Marks | 4 Marks |

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | - Registration through Form (name, email, password)- Registration as Flight Operator with Admin Approval |
| FR-2 | User Login | - Login using Email and Password- Operator approval check before login |
| FR-3 | Flight Search | - Search flights by From, To, and Date- Filter flights on frontend |
| FR-4 | Flight Booking | - Book a flight with passenger details- Reduce available seats after booking |
| FR-5 | My Bookings | - User can see their booking history- User can cancel bookings |
| FR-6 | Operator - Add Flight | - Operator can add new flights |

**Non-functional Requirements:**

Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Non-Functional Requirement** | **Description** |
| **NFR-1** | Usability | The system should provide a simple and intuitive user interface for users, operators, and admins for easy navigation and booking. |
| **NFR-2** | Security | User authentication is done via JWT. Passwords are hashed using bcrypt. Sensitive details (like JWT secret and database URL) are stored in environment variables (.env). |
| **NFR-3** | Reliability | The system should remain reliable and not crash during high user activity like flight searches and bookings. All CRUD operations (Register, Login, Book, Cancel) are thoroughly tested. |
| **NFR-4** | Performance | The website should load pages (Home, Flights, Bookings) in under 3 seconds even on moderate internet connection. Backend APIs should respond within 500ms for key routes. |
| **NFR-5** | Availability | The system should remain available 24x7 when deployed on Render. MongoDB Atlas ensures high data availability. |
| **NFR-6** | Scalability | The backend is built with Node.js and Express.js allowing easy horizontal scaling. The frontend React app can be scaled on platforms like Vercel/Netlify. MongoDB Atlas supports scalable storage for growing flight and booking data. |