Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	23rd May 2025
Team ID	LTVIP2025TMID53841
Project Name	FlightFinder
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 Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Flight Browsing & Search	Browse Flights by Date & Route Search by Airline or Flight No. Filter by Price, Time, or Duration
FR-4	Booking & Checkout	Select Flight & Seat Add Passenger Details Make Payment (e.g., Razorpay/Stripe) Receive Booking Confirmation Email
FR-5	Wishlist (Optional/Future Scope)	Add Flight to Wishlist View Wishlist Remove Flight from Wishlist

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The platform will offer a responsive and intuitive interface using React-Bootstrap for both desktop and mobile devices to maximize ease of use and user satisfaction.
NFR-2	**Security**	Authentication will be managed using JWT tokens; passwords are secured with bcrypt. HTTPS will be enforced, and API endpoints protected. File uploads (e.g., ID proofs) will be secured with Multer.
NFR-3	**Reliability**	The system will provide stable performance with error handling, request retries, and MongoDB replica sets/backups to ensure data availability and durability.
NFR-4	**Performance**	The application will support at least 500 concurrent users with response times under 2 seconds, leveraging backend optimizations, caching (Redis), and potential CDN for static assets.
NFR-5	**Availability**	99.9% uptime is ensured through proper server monitoring, health checks, and deployment across multiple availability zones.
NFR-6	**Scalability**	The backend is built on a scalable Node.js + Express architecture, with MongoDB designed for horizontal scaling and service modularization.