Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	*31 January 3035*
Team ID	*LTVIP2025TMID53108*
Project Name	*FlightFinder*
Maximum Marks	*4 Marks*

Technical Architecture:

FlightFinder is a scalable, high-performance, and user-centric application designed to streamline the flight search experience using real-time APIs, intelligent indexing, and responsive interfaces across mobile and web platforms

FlightFinder – Technical Component Overview

S.No	Component	Description	Technology
1	User Interface	Web UI, Mobile App for flight search and alerts	HTML, CSS, React.js / Flutter / React Native
2	Application Logic-1	Core backend logic: search, indexing, alerts	Node.js / Python (FastAPI or Flask)
3	Database	Flight data, users, preferences, and alerts	MongoDB Atlas (NoSQL) with multikey and dynamic indexes
4	Cloud Database	Cloud-hosted DB for scaling and backups	IBM Cloudant or MongoDB Atlas Cluster

S.No	Component	Description	Technology
5	Infrastructure	Hosting backend, DB, and services on cloud or hybrid	Kubernetes, IBM Cloud Foundry, or AWS Elastic Beanstalk

FlightFinder – Table 2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frontend and backend frameworks used to build the app	React.js, Node.js, Express.js, MongoDB, Redis
2	Security Implementations	Secures data and controls access across services	HTTPS, JWT Auth, SHA-256 encryption, Role-based IAM, OWASP Top 10 mitigation
3	Scalable Architecture	Microservices for modular scaling; separate layers for UI, logic, data	Microservices, Docker, Kubernetes
4	Availability	Load-balanced and fault-tolerant deployments	HAProxy or NGINX Load Balancer, Multi-zone Cloud Deployments
5	Performance	Caching frequent queries, async processing, reduced latency, CDN support	Redis (cache), CDN (e.g. Cloudflare), Indexed MongoDB queries, Auto-scaling