**Project Design Phase-II**

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

|  |  |
| --- | --- |
| Date | 31 January 2025 |
| Team ID | LTVIP2025TMID53108 |
| Project Name | Flight Finder |
| Maximum Marks | 4 Marks |

**Flight Finder – Functional Requirements**

| **No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| --- | --- | --- |
| FR-1 | User Registration | Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | Flight Search | Search by source, destination, date |
|  |  | Sort results by relevance, price, travel time |
| FR-4 | Alerts & Recommendations | Set price drop alerts for specific routes |
|  |  | Receive flight recommendations based on user history |
| FR-5 | Booking Redirection | Click on selected flight to visit external booking site |
|  |  | Track redirection success or failure |
| FR-6 | User Dashboard | View saved searches, alerts, recent activity |
|  |  | Monitor Users and Operators |
| FR-7 | Admin Functions | View analytics on searches and performance |
|  |  | Modify indexing and search configurations |
| FR-8 | Customer Support | Admin/Executive can respond and close tickets |

**Flight Finder – Non-Functional Requirements**

| **NFR No.** | **Non-Functional Requirement** | **Description** |
| --- | --- | --- |
| NFR-1 | Usability | The interface should be intuitive and responsive across devices (web & mobile), ensuring a seamless user experience for travelers of all skill levels. |
| NFR-2 | Security | Secure authentication (e.g. JWT, OAuth), encrypted data transmission (TLS), and adherence to OWASP standards to protect user data and prevent attacks. |
| NFR-3 | Reliability | The system should deliver consistent results and handle API failures or retries gracefully with fallback caching and error handling mechanisms. |
| NFR-4 | Performance | The platform should support high-speed queries |
| NFR-5 | Availability | Ensure 99.9% uptime via multi-zone cloud deployment, load balancers. |
| NFR-6 | Scalability | Modular microservices and dynamic indexing architecture should support increasing traffic and growing datasets without service degradation. |