**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 31 January 3035 |
| --- | --- |
| Team ID | SWTID1743511769 |
| Project Name | TravelSphere(Social media for travel enthusiast) |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

You can include a 3-tier architecture diagram:

* **Client Tier:** Web/Mobile App
* **Application Tier:** Backend Services
* **Data Tier:** Cloud Databases + Local Storage

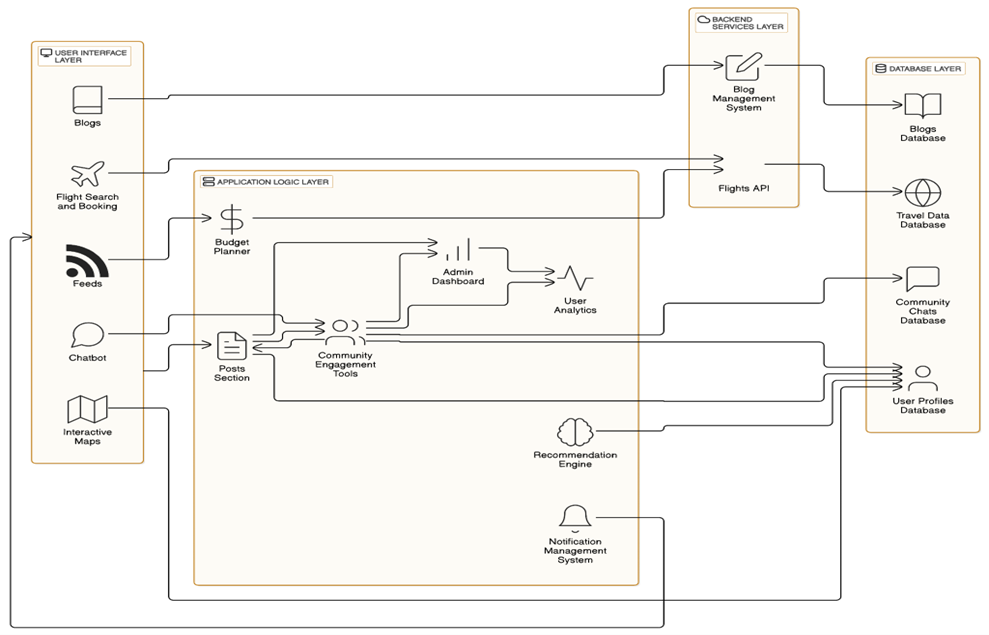
****

Fig: Architecture of TravelSphere Application

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | How user interacts with the app: web and mobile UIs | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | User registration, login, search & filter travel packages | JSpring Boot (Java) / Node.js |
|  | Application Logic-2 | Travel suggestions using ML | Python-based Recommendation Engine |
|  | Application Logic-3 | Chatbot for travel inquiries | Dialogflow |
|  | Database | Stores user info, travel packages, bookings | MySQL for structured data |
|  | Cloud Database | Cloud-hosted version of DB for scalability | Firebase Realtime DB / MongoDB Atlas |
|  | File Storage | Storage of images, documents, itineraries | Firebase Storage / AWS S3 / Local FS |
|  | External API-1 | Weather details for destinations | OpenWeatherMap API |
|  | External API-2 | Travel visa info, geolocation validation | REST Countries API / Google Maps API |
|  | Machine Learning Model | Recommend personalized travel packages | Scikit-learn / TensorFlow-based ML model |
|  | Infrastructure (Server / Cloud) | Deployed on cloud platform | Cloud Foundry / Kubernetes / AWS / Render. |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frontend and backend libraries and packages | React.js, Node.js, Flask, Spring Boot |
|  | Security Implementations | Securing user data, login, and transactions | SHA-256, JWT Tokens, HTTPS, OAuth 2.0, IAM |
|  | Scalable Architecture | Supports growing users, services split by domain (microservices) | Technology used |
|  | Availability | Maintains uptime with load balancing and cloud redundancy | AWS Load Balancer / Cloud Foundry Routing |
|  | Performance | High-performance with caching and CDN for images & static content | TRedis (for caching), Cloudflare CDN |