**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 27 june 2025 |
| Team ID | LTVIP2025TMID53139 |
| Project Name | House hunt: Finding your perfect rental home |
| Maximum marks | 4 marks |

**🏗️ Solution Architecture**

**📌 Components:**

1. **Frontend (React.js)**
   * Handles user interface (login, listing, search, filtering)
   * Communicates with backend via REST APIs
2. **Backend (Node.js + Express)**
   * Handles API routing, business logic
   * Manages authentication, authorization, validation
3. **Database (MongoDB Atlas)**
   * Stores user data, property listings, chat messages
4. **Authentication Service (JWT + Bcrypt)**
   * JWT for session tokens
   * Bcrypt for secure password hashing
5. **Hosting**
   * Frontend: Vercel
   * Backend: Render or Railway
   * DB: MongoDB Atlas (Cloud)

**🖼️ System Architecture Diagram (Text Version)**

less

CopyEdit

[User Interface - React.js]

| REST API Calls

↓

[Node.js + Express Backend]

| |

[MongoDB Atlas] [Auth Service - JWT/Bcrypt]

|

[Cloud Deployment - Render]

**Architecture Diagram (Conceptual):**

+---------------------++---------------------+ +---------------------+ +---------------------+

| | | | | |

| User Devices | | CDN | | Admin Dashboard |

| (Web/Mobile Clients)|------->| (Static Content) | | |

| | | | | |

+----------+----------+ +----------+----------+ +----------+----------+

| | |

| HTTP(S) | HTTP(S) | HTTP(S)

| | |

+----------V----------+ +----------V----------+ +----------V----------+

| | | | | |

| Load Balancer |------->| API Gateway |------->| Authentication/ |

| (e.g., ALB, NGINX) | | (e.g., AWS API GW, | | Authorization |

| | | Kong, Apigee) | | (e.g., Cognito, |

+----------+----------+ +----------+----------+ | Auth0, OAuth2) |

| | +----------+----------+

| | |

| | REST/GraphQL |

+----------V------------------------------V--------------------------------V----------+

| | |

| Microservices Layer (Backend) |

| | |

+-----------------------------------------+-------------------------------------------+

| Property Service | User Service | Search/Indexing Service | Notification Service |

| (Listings, Details) | (Profiles, Saved) | (Elasticsearch, Solr) | (Email, SMS, Push) |

+-----------------------+------------------+----------------------------+--------------------------+

| | | | |

| Messaging/Queueing | Image/File | Payments Service | Analytics/Logging |

| (e.g., Kafka, SQS) | Storage | (if applicable) | (e.g., ELK Stack) |

| | (e.g., S3) | | |

+-----------------------+------------------+----------------------------+--------------------------+

| | |

| RDBMS/NoSQL (Primary Data Stores) | Search Indexing DB |

| | |

+----------V-----------------------------------------V--------------------------------V----------+

| |

| Database Layer |

| |

+-----------------------------------------------------------------------------------------------+

| Property DB (PostgreSQL, MySQL) | User DB (PostgreSQL, MongoDB) | Search Index (Elasticsearch) |

| | | Geolocation DB (PostGIS) |

+-----------------------------------+-------------------------------+------------------------------+

|

| Caching Layer (e.g., Redis, Memcached)

|

+----------V----------+

| |

| Caching Service |

| |