**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 15 February 2025 |
| Team ID | LTVIP2025TMID55800 |
| Project Name | House hunt: Finding your perfect rental home |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

**Overview**

* The solution architecture for the *HouseHunt* application is designed using the **MERN stack** — MongoDB, Express.js, React.js, and Node.js — to deliver a modern, scalable, and efficient rental platform. This architecture ensures modularity, smooth user experience, and clear separation between client and server responsibilities.

**Objectives of the Architecture**

* Provide a **responsive and interactive frontend** for renters, owners, and admins.
* Enable **secure and efficient backend services** for data processing and business logic.
* Ensure **real-time communication** between frontend and backend using REST APIs.
* Support **scalable and flexible data storage** using MongoDB.
* Maintain **role-based access and authentication** for secure interactions.

**🧱 Architecture Components**

**1. Frontend (React.js)**

* Built using React.js with Ant Design, Material UI, and Bootstrap for responsive design.
* Uses Axios to interact with backend RESTful APIs.
* Supports role-based views: Renter, Owner, Admin.
* Implements client-side routing with react-router.

**2. Backend (Node.js + Express.js)**

* Node.js provides the runtime environment.
* Express.js handles routing, middleware, and server logic.
* API endpoints are protected using middleware for authentication (JWT).
* Multer is used for handling file uploads like property images.

**3. Database (MongoDB)**

* A NoSQL database used to store structured and unstructured data.
* Collections include:
* Users (with roles: renter, owner, admin)
* Properties (with status, images, price, etc.)
* Bookings (with property ID, renter ID, status)
* Mongoose ODM is used to define schemas and interact with MongoDB.
* **4. Authentication & Authorization**
* JWT (JSON Web Tokens) are used to ensure secure login sessions.
* Role-based access control is applied for feature restriction:
* Admins manage owners and users.
* Owners manage properties.
* Renters browse and book.

**Example - Solution Architecture Diagram:**

**A diagram of a house rent application

AI-generated content may be incorrect.**

*Figure 1: Architecture and data flow of the House Hunt*