**Project Requirements Analysis**

**Project Design Phase-II: Technology Stack (Architecture & Stack)**

**Date:**

25 June 2025

**Team ID:** LTVIP2025TMID53374

**Project Name:**

**HouseHunt: Finding Your Perfect Rental Home**

**Maximum Marks:**

4 Marks

**Technical Architecture (Overview)**

**Architecture Type**: 3-Tier (Frontend – Backend – Database)  
**Deployment**: MERN stack hosted on **Render / Vercel (Frontend)** and **Railway / MongoDB Atlas (Backend + DB)**

**High-Level Components:**

* **Client UI (React)** – interacts with backend via REST API
* **Node.js Express Server** – handles API logic, auth, booking
* **MongoDB** – stores users, listings, and booking data
* **Admin Tools** – verify owners, moderate listings
* **APIs** – integrate user authentication, location (if used), or maps

**Table-1: Components & Technologies**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | User Interface | Web app used by renters, owners, and admin | HTML, CSS, JavaScript, React.js, Bootstrap, Material UI |
| 2 | Application Logic-1 | User authentication, registration, login | Node.js, Express.js |
| 3 | Application Logic-2 | Booking flow and status tracking | Node.js, Express.js |
| 4 | Application Logic-3 | Admin verification and approval module | Node.js, Express.js |
| 5 | Database | Stores users, listings, and bookings | MongoDB (NoSQL) |
| 6 | Cloud Database | Hosted database with cloud access | MongoDB Atlas |
| 7 | File Storage | Stores property images | Cloudinary / Local Filesystem |
| 8 | External API-1 | Geolocation or map integration (optional) | Google Maps API |
| 9 | External API-2 | (Optional) Email service for confirmation emails | Nodemailer / SendGrid |
| 10 | Machine Learning Model | Not used in this version (can be added later for price prediction) | – |
| 11 | Infrastructure | Hosting and deployment on cloud | Vercel (Frontend), Railway (Backend), MongoDB Atlas (DB) |

**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | React.js, Node.js, Express.js, MongoDB | JavaScript, MongoDB |
| 2 | Security Implementations | Password hashing, role-based auth, JWT, HTTPS | bcrypt, JWT, CORS, Helmet.js, HTTPS |
| 3 | Scalable Architecture | 3-tier separation of concerns, component-based UI, NoSQL DB scalability | React, Express.js, MongoDB Atlas |
| 4 | Availability | Cloud-hosted backend & database, frontend on Vercel, 99.9% uptime hosting | Render / Vercel / Railway / MongoDB Atlas |
| 5 | Performance | Uses REST APIs, database indexing, compression, optimized rendering | Axios, MongoDB Indexes, CDN (Vercel), Lazy Loading |

**Architectural Diagram Description (Visual Summary)**

less

CopyEdit

[ User Browser (React UI) ]

|

↓

[ Frontend Hosting (Vercel) ]

|

↓

[ Express.js Backend API (Railway) ]

|

↓

[ MongoDB Atlas ]

|

┌─────────────┐

│ Collections │ → Users, Properties, Bookings

└─────────────┘

**Future Improvements**

* Add **Machine Learning Model** for dynamic rent prediction
* Add **SMS/email OTP integration**
* Support for **map-based property search**
* Use **Kubernetes** for microservices if scaled to city/state level