**Project Design Phase-II**

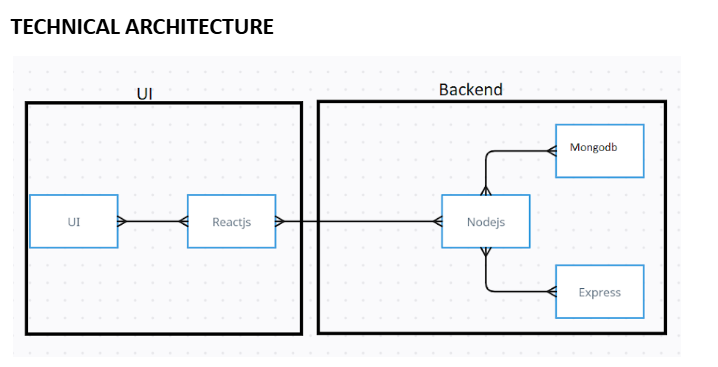
**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27/06/2025 |
| Team ID | LTVIP2025TMID55800 |
| Project Name | Househunt: finding your perfect rental home |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

The technical architecture of our House rent app follows a client-server model, where the frontend serves as the client and the backend acts as the server.



**Table-1: Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| **1** | **User Interface** | **Web interface for Renter, Owner, Admin** | **HTML, CSS, JavaScript, React.js, Antd** |
| **2** | **Application Logic-1** | **Handles client-side routing and UI states** | **React.js** |
| **3** | **Application Logic-2** | **Handles backend routing and REST APIs** | **Node.js, Express.js** |
| **4** | **Application Logic-3** | **Authentication, CRUD for bookings and properties** | **JWT, bcryptjs, MongoDB Mongoose** |
| **5** | **Database** | **Houses renters, owners, bookings, property details** | **MongoDB (NoSQL)** |
| **6** | **Cloud Database** | **Can be hosted on MongoDB Atlas** | **MongoDB Atlas (Optional)** |
| **7** | **File Storage** | **For image uploads of rental properties** | **Multer, Local Filesystem** |
| **8** | **External API-1** | **Maps, address geolocation** | **Google Maps API (optional integration)** |
| **9** | **External API-2** | **Email alerts / push notifications** | **Firebase Cloud Messaging (optional)** |
| **10** | **Infrastructure** | **Deployment and hosting** | **Localhost for dev, can be Docker/Cloud** |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| **1** | **Open-Source Frameworks** | **MERN stack, Axios, Antd, Bootstrap, Moment.js** | **React.js, Express.js, MongoDB, Node.js** |
| **2** | **Security Implementations** | **Password encryption, JWT for session control, Role-based access** | **bcryptjs, JWT, dotenv, CORS** |
| **3** | **Scalable Architecture** | **Tiered design with modularity to plug microservices in future** | **MERN, REST APIs, Docker (optional)** |
| **4** | **Availability** | **Can be hosted on cloud with load balancers, Node clustering** | **Express.js with PM2, MongoDB Replica Sets** |
| **5** | **Performance** | **Optimized API calls, database indexing, caching planned** | **Axios, Mongoose indexing, Lazy loading** |