**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 16 July 2025 |
| Team ID | LTVIP2025TMID59690 |
| Project Name | HouseHunt – Rental Property Recommender |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**🔧 Tech Stack Overview**

| **Layer** | **Technology Used** | **Purpose** |
| --- | --- | --- |
| **Frontend** | React.js, HTML5, CSS3, TailwindCSS/Bootstrap | Build interactive user interfaces |
| **Backend** | Node.js, Express.js | Handle API requests, business logic |
| **Database** | MongoDB, Mongoose ORM | Store structured and unstructured data |
| **Authentication** | JWT (JSON Web Token), bcrypt | Secure login, token-based authorization |
| **File Uploads** | Multer | Upload lease agreements, property images |
| **API Testing** | Postman | Test REST API endpoints |
| **Deployment** | Vercel (frontend), Render/Heroku (backend), MongoDB Atlas | Host the full stack app |
| **Version Control** | Git & GitHub | Code management and collaboration |

**🧱 MERN Architecture (High-Level)**

**Frontend (React.js)**

**↓**

**API Calls (Axios/Fetch)**

**↓**

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

**↓**

**Database (MongoDB via Mongoose)**

**🧩 Component-wise Breakdown**

**🔹 Frontend (React.js)**

* **Built as a Single Page Application (SPA)**
* **Components for:**
  + **Home, Login/Register, Dashboard (Renter/Owner/Admin)**
  + **Property List, Property Details**
  + **Inquiry Form, Booking Form**
* **Form validation using React Hook Form / Formik**
* **Routing handled by React Router**

**🔹 Backend (Node.js + Express.js)**

* **RESTful APIs with modular routes and controllers**
* **Middleware:**
  + **authMiddleware.js for role-based access control**
  + **errorHandler.js for error responses**
* **Endpoints:**
  + **/api/auth – User login, registration**
  + **/api/properties – CRUD operations for properties**
  + **/api/bookings – Booking and approval flow**
  + **/api/admin – Admin user and property controls**

**🔹 Database (MongoDB + Mongoose)**

* **Collections:**
  + **users – Info about Renters, Owners, Admins**
  + **properties – Rental listings with location, price, etc.**
  + **bookings – Booking history and status**
  + **leases – Lease documents, references to property/user**
* **Mongoose schemas for validation and relationships**

**🔹 Authentication & Security**

* **JWT tokens stored in localStorage**
* **Passwords hashed using bcrypt**
* **Middleware to protect private routes**
* **CORS enabled for frontend-backend communication**

**⚙️ Deployment Strategy**

| **Component** | **Platform** | **Purpose** |
| --- | --- | --- |
| **Frontend** | **Vercel** | **Hosting React app** |
| **Backend** | **Render/Heroku** | **Running Express server** |
| **Database** | **MongoDB Atlas** | **Cloud MongoDB database** |

**🛡️ Additional Tools (Optional but Recommended)**

| **Tool/Library** | **Purpose** |
| --- | --- |
| **Redux / Context API** | **State management (for bookings, auth)** |
| **Framer Motion** | **UI animations** |
| **Cloudinary / Firebase** | **Image/file hosting** |
| **Swagger** | **API documentation** |
| **Jest / Mocha** | **Unit testing** |