**Project Final Report**

**Project Title:**

**HouseHunt: Finding Your Perfect Rental Home**

**1. Introduction**

**HouseHunt** is a MERN stack-based rental application designed to simplify the process of finding, booking, and managing rental homes and apartments. It serves as a digital platform for tenants, landlords, and administrators to streamline the rental process through a user-friendly interface and secure backend infrastructure.

**2. Objective**

The objective of HouseHunt is to:

* Facilitate property discovery and rental bookings.
* Provide a seamless experience for renters and owners.
* Enable administrators to govern platform activity and user authenticity.
* Digitize lease negotiations, property management, and booking confirmation.

**3. Key Features**

* **User Registration & Login:** Secure account creation for renters and property owners.
* **Property Listings:** Detailed descriptions, photos, rent amount, and availability.
* **Advanced Search Filters:** Location, price range, type, bedrooms, and amenities.
* **Property Inquiry & Booking:** Instant inquiry and booking request from renter to owner.
* **Messaging & Notifications:** In-app communication for lease discussions.
* **Owner Dashboard:** Add, edit, and manage property listings.
* **Admin Dashboard:** Monitor users, approve owners, enforce platform rules.
* **Lease Agreement Finalization:** Transparent negotiation and payment.

**4. Scenario-Based Case Study**

**Scenario: Renting an Apartment**

**User Journey:**

1. **Alice**, a renter, registers on the app.
2. She browses listings, applies filters, and selects a suitable apartment.
3. She inquires by submitting a form to the property owner.
4. **Bob**, the owner, receives the request, reviews, and confirms booking.
5. The admin verifies Bob’s owner credentials and activates his listing permissions.
6. Lease terms are finalized via the messaging system.
7. Alice receives confirmation and successfully moves into her new apartment.

**5. Technical Architecture**

HouseHunt follows a client-server architecture comprising:

**Frontend (Client):**

* **React.js**: User interface development.
* **Axios**: Handles HTTP requests to the backend.
* **Bootstrap & Material UI**: Responsive and consistent UI components.
* **Ant Design**: Enhanced UI elements and design systems.

**Backend (Server):**

* **Node.js + Express.js**: RESTful API, routing, middleware.
* **MongoDB**: Document-oriented NoSQL database for users, properties, and bookings.
* **Mongoose**: ODM for MongoDB schema modeling.
* **Moment.js**: Date formatting and manipulation.

**6. ER Diagram Structure**

**Collections:**

**Users**

* \_id
* name
* email
* password
* type (Renter / Owner / Admin)

**Property**

* \_id
* userID (owner’s reference)
* type (Flat / Room / House)
* adType (For Rent / For Lease)
* isAvailable
* address
* contact
* amount
* images
* additionalInfo

**Booking**

* \_id
* propertyId
* userId (renter)
* ownerId
* username

**7. Pre-requisites**

To develop and run HouseHunt, the following tools and libraries are essential:

| **Tool** | **Purpose** |
| --- | --- |
| **Node.js & npm** | JavaScript runtime and package manager |
| **Express.js** | Backend framework for routing and APIs |
| **MongoDB** | NoSQL database |
| **Mongoose** | MongoDB ODM for schema modeling |
| **React.js** | Frontend UI framework |
| **Axios** | HTTP client for REST calls |
| **Bootstrap & Material UI** | Styling UI components |
| **Ant Design** | Advanced UI components |
| **Moment.js** | Date/time formatting |
| **HTML, CSS, JavaScript** | Core web technologies |

**8. Installation & Setup**

**Step 1: Clone Repository**

bash

Copy code

git clone <your-repo-url>

**Step 2: Install Dependencies**

bash

Copy code

cd house-rent

cd frontend

npm install

cd ../backend

npm install

**Step 3: Start Development Servers**

bash

Copy code

# Frontend (React)

cd frontend

npm start

# Access: http://localhost:3000

# Backend (Express)

cd ../backend

npm start

# Runs on default backend port (e.g., 5000)

**9. Admin Roles & Responsibilities**

* Verify owner accounts.
* Monitor property listings for compliance.
* Manage reports and disputes.
* Maintain user data integrity and safety protocols.

**10. Conclusion**

**HouseHunt** effectively bridges the gap between property seekers and owners, enabling a digital-first approach to renting homes. Its modular design, strong backend, and rich UI experience make it scalable and reliable for real-world use cases. The platform not only simplifies the rental journey but also ensures transparency, safety, and trust among its users.