Project Report

Project Title:

HouseHunt: Finding Your Perfect Rental Home

Team Members :
Kudupudi Divya Sri

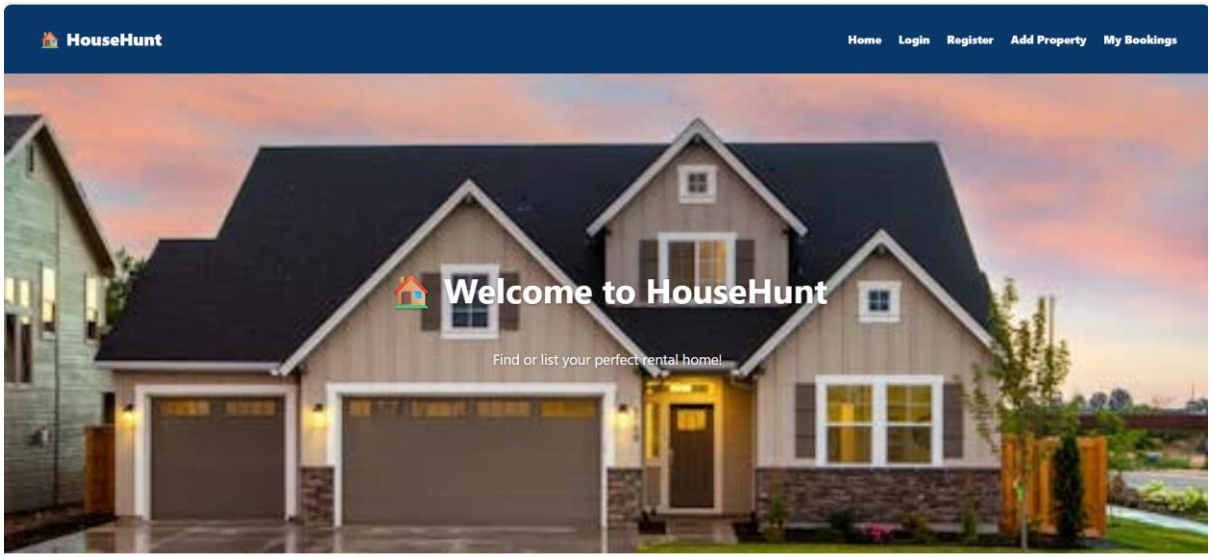
Kanala Geethapranathi
Kondala Soma Sekhar

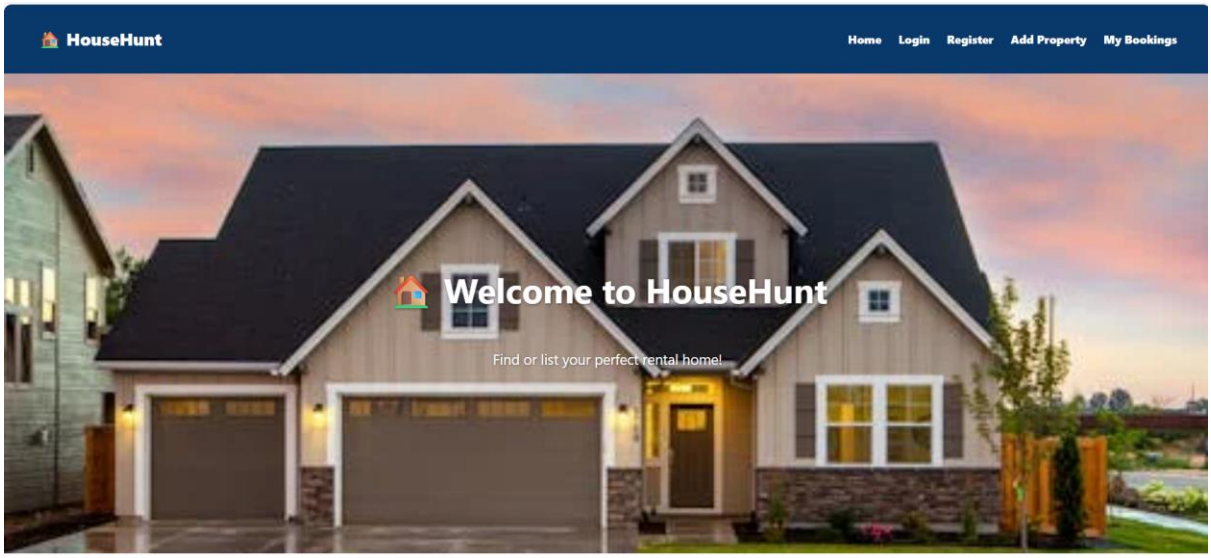
Jyothirganesh Kanuboyina

---

1. Introduction

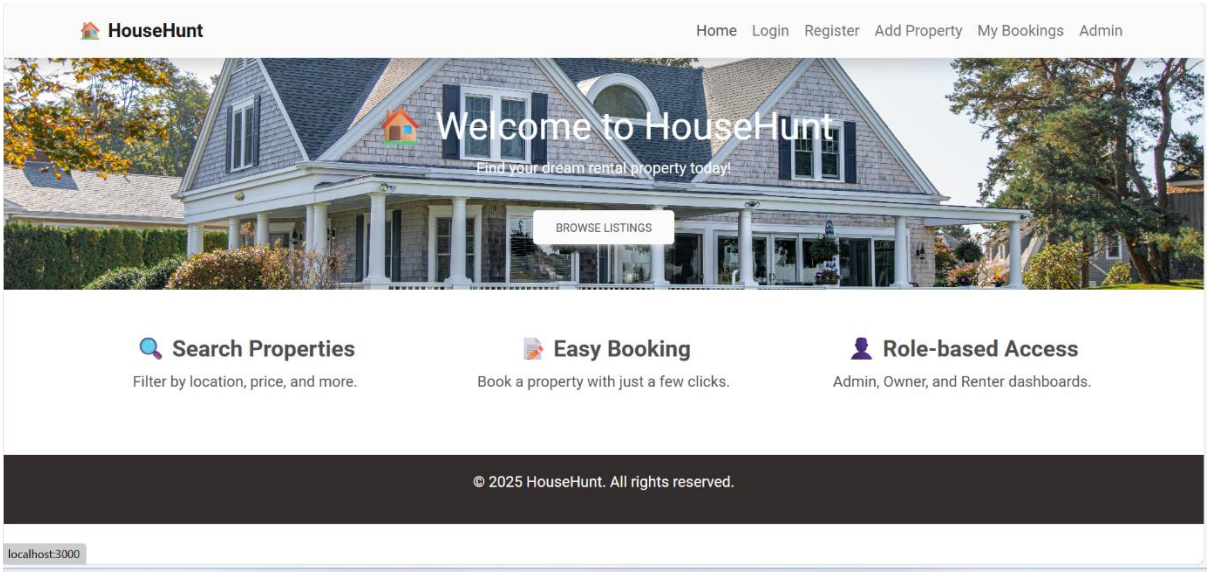
HouseHunt is a MERN stack-based rental application designed to simplify the process of finding, booking, and managing rental homes and apartments. It serves as a digital platform for tenants, landlords, and administrators to streamline the rental process through a user-friendly interface and secure backend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





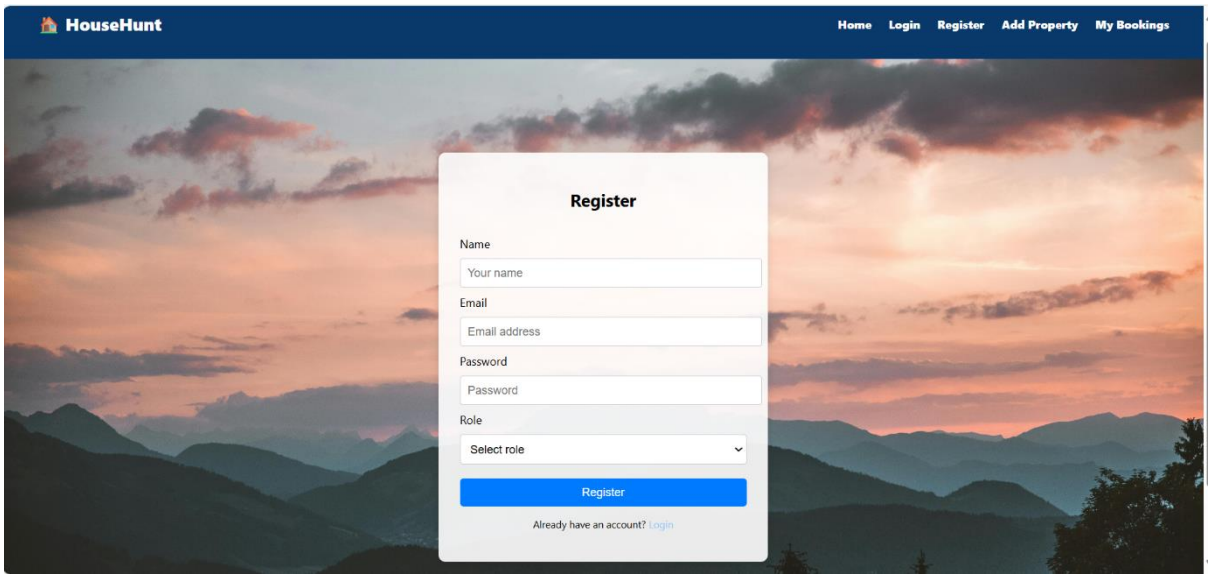
- **2. Project Overview**
- **Purpose:**
  HouseHunt aims to simplify the process of finding and renting homes by providing a platform where renters can search, filter, and book properties seamlessly while property owners can manage listings efficiently.
- **Features:**
- User Registration & Login (Renter, Owner, Admin)
- Property Listings with images and descriptions
- Search with filters (location, price, type, amenities)
- Contacting landlords via form submission
- Admin approval for owner registration
- Owner dashboard for property management
- Booking management and notifications
- Secure lease negotiation and confirmation

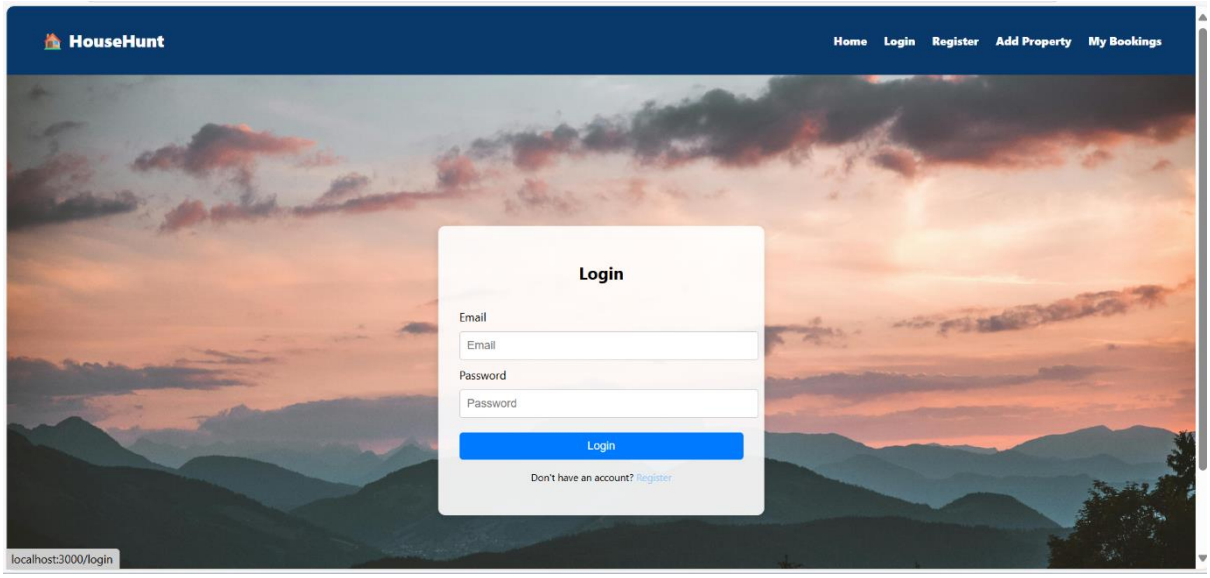
- **3. Architecture**



- **Frontend (React.js):**
- Built using React with routing and state management
- Axios used to make HTTP requests to backend APIs
- UI styled with Bootstrap, Material UI, and Ant Design
- **Backend (Node.js + Express.js):**
- RESTful APIs for user authentication, property management, and booking
- Middleware for error handling and authentication
- **Database (MongoDB):**
- Collections: Users, Properties, Bookings
- Mongoose used for schema design and CRUD operations
REQUIREMENT ANALYSIS
- **3.1 Customer Journey Map**
- User signs up → Browses listings → Applies filters → Views details → Sends inquiry → Books apartment → Owner confirms → Admin moderates → Deal finalized
- **3.2 Solution Requirement**
- Secure user authentication
- Efficient CRUD operations

- Seamless UX/UI
- Admin control panel
- Real-time updates and notifications
- **3.3 Data Flow Diagram**
- Frontend ↔ Express Server ↔ MongoDB
- Users interact with React components
- Backend processes requests and responses
- MongoDB stores persistent data
- **3.4 Technology Stack**
- Frontend: React.js, Bootstrap, Material UI, Ant Design
- Backend: Node.js, Express.js
- Database: MongoDB + Mongoose
- Other Tools: Moment.js, Axios
-





---

4. PROJECT DESIGN

4.1 Problem Solution Fit

- **Problem: Manual property search is time-consuming and scattered.**
- **Solution: A centralized web platform for easy rental property browsing and management.**

4.2 Proposed Solution

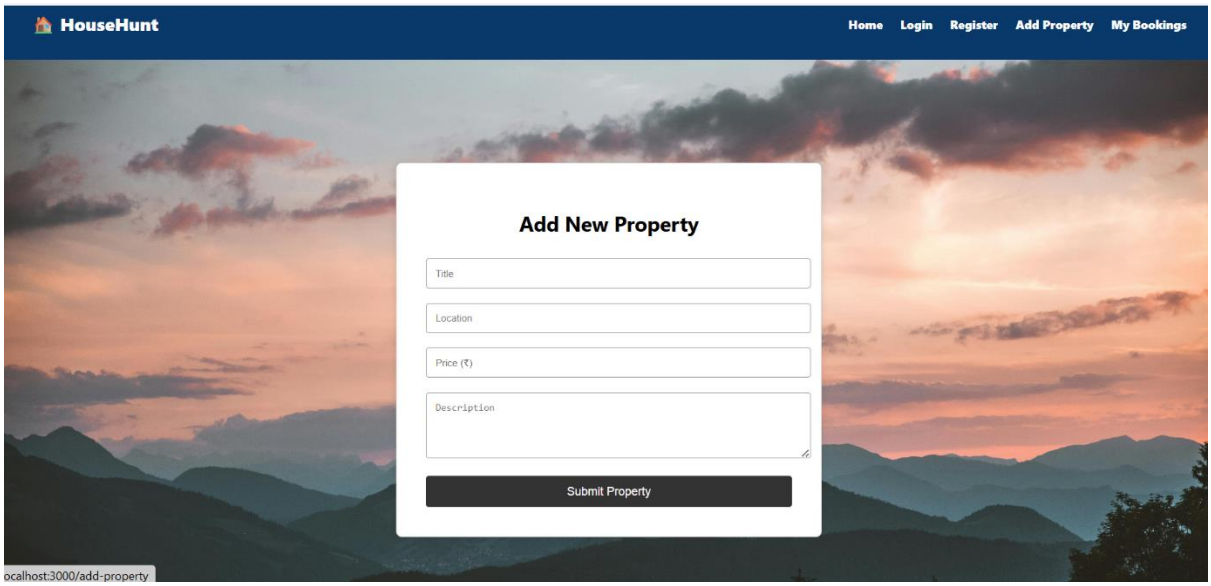
- **Web-based platform supporting renters, owners, and admin workflows.**
- **End-to-end booking and management system.**

4.3 Solution Architecture

- **Client-Server architecture**
- **RESTful API interaction**

---

5. Technical Architecture



HouseHunt follows a client-server architecture comprising:

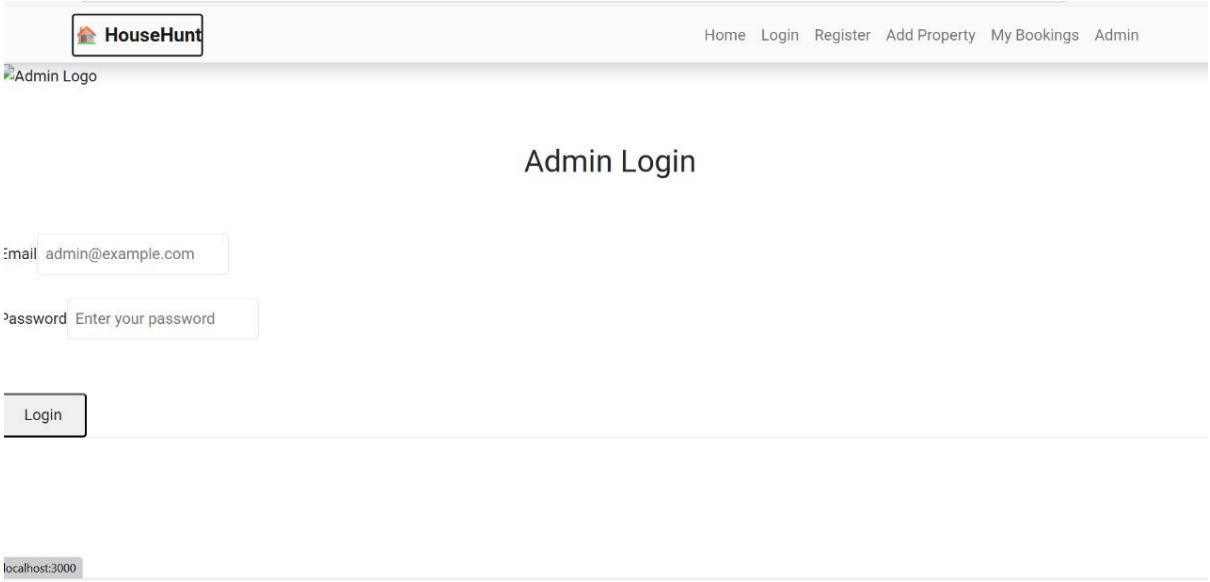
Frontend (Client):

- **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.



---

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

CopyEdit

git clone <your-repo-url>

Step 2: Install Dependencies

bash

CopyEdit

cd house-rent

cd frontend

npm install

cd ../backend

npm install

Step 3: Start Development Servers

bash

CopyEdit

Frontend (React)

cd frontend

npm start

Access: http://localhost:3000

Backend (Express)

cd ../backend

npm start

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

10.ADVANTAGES & DISADVANTAGES

Advantages:

- Intuitive and clean UI
- Role-based access and workflows
- Real-time property updates
- Centralized management and governance

Disadvantages:

- No offline mode
- Limited to web (no native mobile app yet)

Relies on internet connectivity

11. 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.

12. FUTURE SCOPE

- Add payment integration
- Develop a mobile application
- Enable geolocation-based searches
- Integrate AI recommendations based on user behavior
- **13. APPENDIX**

  Source Code:
  GitHub & Project Demo Link :
  https://github.com/GeethaPranathi/HouseHunt-Finding-Your-Perfect-Rental-Home.git