HouseHunt: Finding Your Perfect Rental Home

1. Introduction

• **Project Title**: HouseHunt – Finding Your Perfect Rental Home

• **Team Members**:

- Posina Sai Chakra Amrutha Varshini (Team Leader)
- Nusum Venkata Surendra Reddy
- Savarapu Tejasri
- Yadla Devendra Manikanta

2. Project Overview

• **Purpose**

HouseHunt is a MERN stack-based house rental application aimed at simplifying the process of finding, listing, and booking rental properties. It supports three user roles: Renter, Owner, and Admin. Each user type is provided with a tailored experience for property management, discovery, and booking.

• **Features**

* User Registration & Login (Renter/Owner)
* Admin Verification of Owners
* Property Listings with Search Filters
* Property Inquiry and Booking
* Booking Status Management
* Owner Dashboard for Property CRUD
* Admin Dashboard for User/Platform Governance
* Responsive UI using Bootstrap, Material UI, and Ant Design

3. Architecture

• **Frontend**

The frontend is built using React.js and provides a dynamic, user-friendly interface for all types of users. It incorporates Axios for API integration and Material UI, Bootstrap, and Ant Design for responsive and attractive UI components.

The frontend allows:
* User Registration/Login
* Property Browsing and Filtering
* Booking Requests
* Booking Status Tracking
* Dashboard Views (Renter/Owner/Admin)

• **Backend**

The backend is powered by Node.js and Express.js and handles all server-side logic, routing, authentication, and CRUD operations. Responsibilities are:

* Handling API Routes
* User Authentication (JWT-based)
* Booking Logic
* Owner/Admin Role Management
* Middleware for protected routes

• **Database**

MongoDB is used for storing structured and unstructured data such as user profiles, property details, and booking information.

Collections include:
* users (Renter/Owner/Admin roles)
* properties (rental listings)
* bookings (tenant booking status)

4. Setup Instructions

• Prerequisites

Before running the application, ensure your system has the required tools such as Node.js, npm, MongoDB, and React.js. These form the core foundation of the MERN stack. Moment.js and Ant Design are also used to enhance features and the user interface.

Ensure the following are installed:

* Node.js & npm: https://nodejs.org/en/download/
* MongoDB: https://www.mongodb.com/try/download/community
* React.js
* Express.js (npm install express)
* Moment.js: https://momentjs.com/
* Ant Design: https://ant.design/docs/react/introduce

• Installation

The installation process begins by cloning the repository and navigating into the respective folders. Dependencies for both frontend and backend must be installed using npm.

```
# Clone the repository
git clone [your-repo-link]
cd house-rent

# Install frontend dependencies
cd frontend
npm install

# Install backend dependencies
cd ../backend
npm install
```

5. Folder Structure

- Client (Frontend)

```
frontend/
├── public/
├── src/
│   ├── components/
│   ├── pages/
│   ├── App.js
│   ├── index.js
│   └── styles/
```

- Server (Backend)

```
backend/
├── config/
│   └── config.js
├── middleware/
│   └── authMiddleware.js
├── models/
│   ├── User.js
│   ├── Property.js
│   └── Booking.js
├── routes/
│   ├── userRoutes.js
│   └── propertyRoutes.js
├── .env
├── index.js
```

6. Running the Application

- **Frontend**

```
cd frontend
npm start
```

- **Backend**

```
cd backend
npm start
```

> Access the app at: http://localhost:3000

7. API Documentation

The backend provides RESTful APIs for authentication, property management, and bookings, secured with role-based middleware.

Method	Endpoint	Description	Request Body / Params
POST	/api/auth/signup	User registration	{ name, email, password, role }
POST	/api/auth/login	User login	{ email, password }
GET	/api/properties	List all properties	-
POST	/api/properties	Add new property (Owner only)	property details
PUT	/api/properties/:id	Edit property	updated fields
DELETE	/api/properties/:id	Delete property	-
POST	/api/bookings	Create booking (Renter)	renterId, propertyId
GET	/api/bookings/:id	View booking status	-

8. Authentication

Authentication uses JWT for secure access and bcryptjs for password encryption. Tokens enable protected route access, and roles control permissions for renters, owners, and admins.

* **Authentication:** Implemented using JWT tokens.
* **Password Encryption:** Done via bcryptjs.
* **Middleware:** authMiddleware.js checks for token validity and grants role-based access.
* **Authorization:**
  * Renter: Can browse and book
  * Owner: Can CRUD properties
  * Admin: Approves owners, monitors activity

9. User Interface

Built using:

* React

* Material UI

* Ant Design

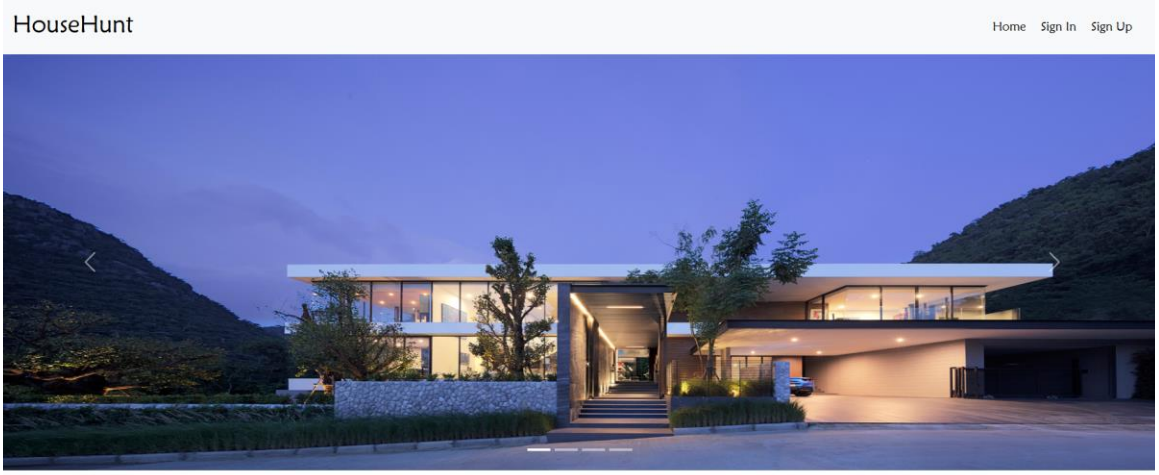
* Bootstrap

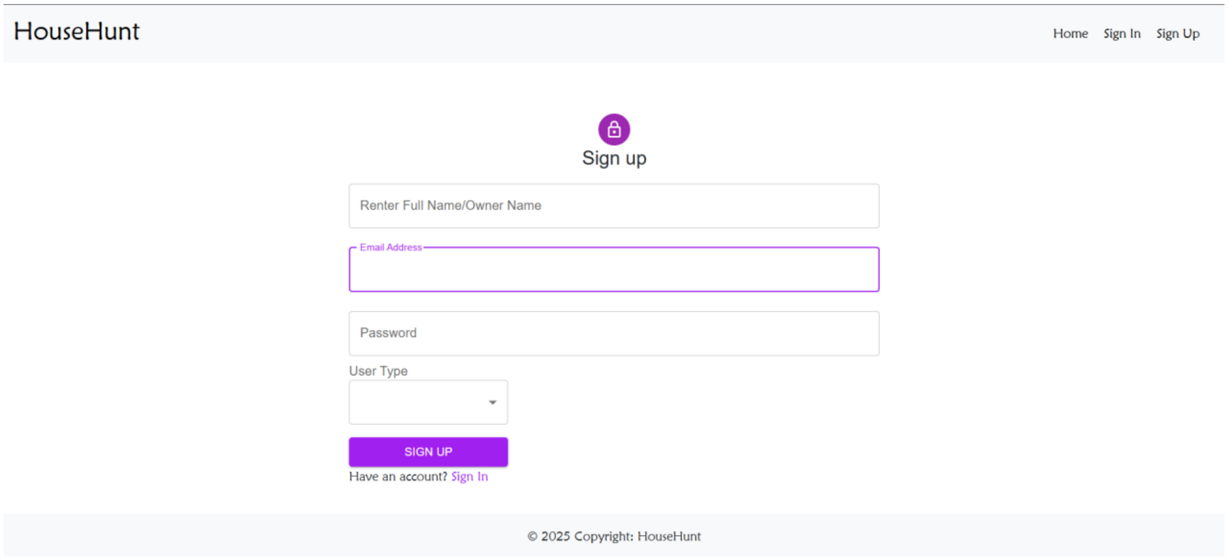
Key Screens:

* Login / Signup

* Renter Dashboard (Bookings, Property Search)

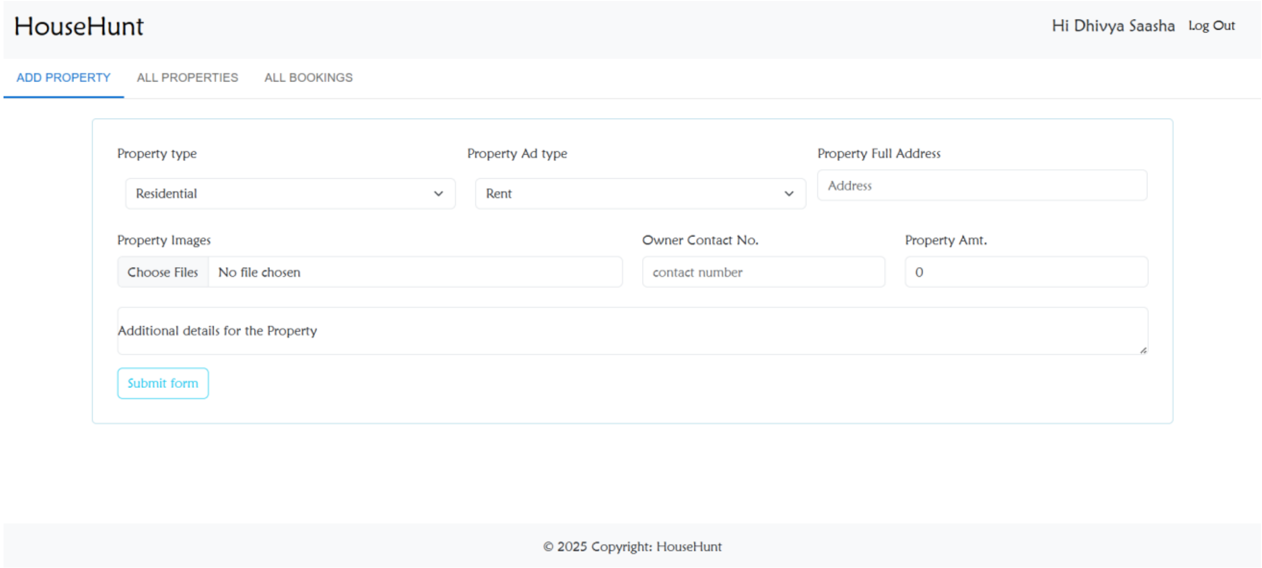
* Owner Dashboard (Add/Edit/Delete Properties)





Have an account? Sign In

© 2025 Copyright: HouseHunt



Property type: Residential

Property Ad type: Rent

Property Full Address: Address

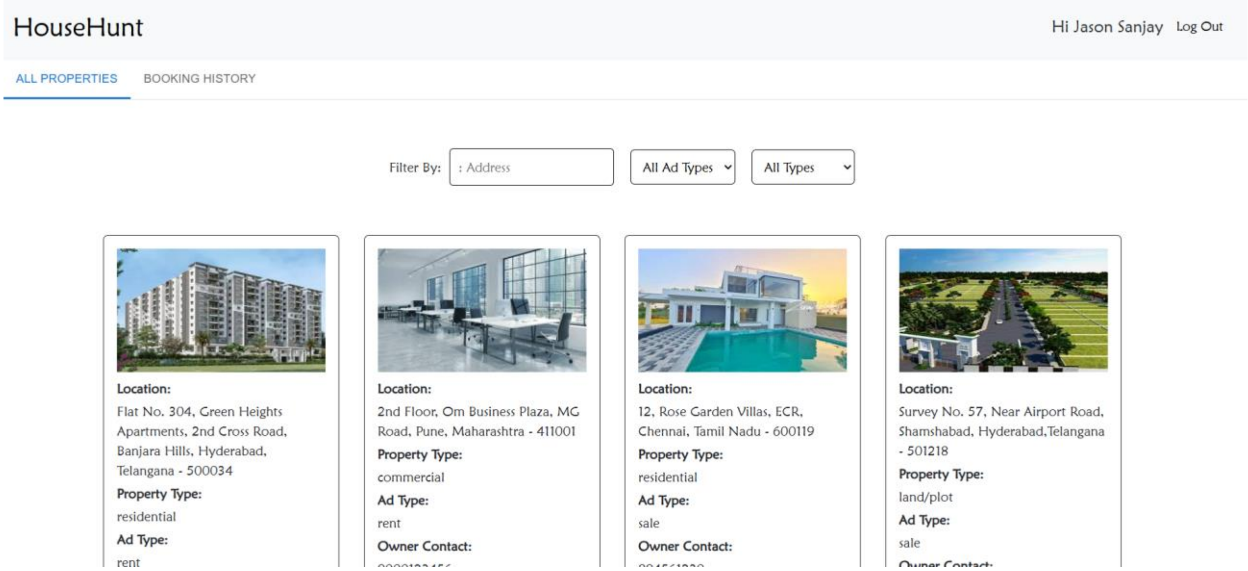
Property Images: No file chosen

Owner Contact No.: contact number

Property Amt.: 0

Additional details for the Property

© 2025 Copyright: HouseHunt



Filter By: : Address | All Ad Types | All Types

Location: Flat No. 304, Green Heights Apartments, 2nd Cross Road, Banjara Hills, Hyderabad, Telangana - 500034
Property Type: residential
Ad Type: rent

Location: 2nd Floor, Om Business Plaza, MG Road, Pune, Maharashtra - 411001
Property Type: commercial
Ad Type: rent
Owner Contact:

Location: 12, Rose Garden Villas, ECR, Chennai, Tamil Nadu - 600119
Property Type: residential
Ad Type: sale
Owner Contact:

Location: Survey No. 57, Near Airport Road, Shamshabad, Hyderabad,Telangana - 501218
Property Type: land/plot
Ad Type: sale
Owner Contact:

10. Testing

* Postman used for API testing

* Console Logs and Error Handling implemented for server

* Frontend tested via form validation and API integration

* Manual testing for booking flow and dashboard updates

11. Screenshots or Demo

Demo Link:
Click here:
https://github.com/AmruthaVarshinipsc/HouseHunt-App/tree/main/Video%20Demonstration

12. Known Issues

* No password reset functionality

* No OTP/email verification for sign-up

* Admin approval process lacks real-time notification

* Booking history not filterable by date

13. Future Enhancements

* Implement chat between Renter and Owner

* Add Google Maps API for property location

* Add payment integration for booking confirmation

* Enable email notifications

* Add a "Featured Properties" section

Conclusion:

HouseHunt successfully delivers a streamlined and user-friendly platform for managing rental properties using the MERN stack. By integrating secure authentication, role-based access, and responsive UI design, the application ensures a smooth experience for renters, owners, and admins. With scalable architecture and room for future enhancements, HouseHunt is well-positioned as a practical solution for modern house rental needs.

College: Aditya College of Engineering and Technology