

Full Stack Development with MERN: Project Documentation

1. Introduction

Project Title: HouseHunt: Finding Your Perfect Rental Home

Team ID: LTVIP2026TMID53603

Team Members:

1. Pillapalem Tejaswi
2. Shruthi Vendoti
3. Vendoti Keerthi

Project Overview

Purpose:

HouseHunt is a web-based rental management platform developed using the MERN stack (MongoDB, Express.js, React.js, Node.js). It aims to streamline the process of finding, listing, and booking rental properties for tenants, owners, and admins.

Key Features:

- Secure login/signup for Renter, Owner, and Admin roles
- Property listing with photos, amenities, and pricing
- Search filters for location, budget, property type, and amenities
- Owner property management (add, edit, remove listings)
- Inquiry and booking system with admin and owner approvals
- Admin controls for user verification and platform governance

2. Architecture

Frontend:

- Built using React.js

- Styled with Tailwind CSS and Bootstrap
- Utilizes Axios for API communication

Backend:

- Node.js + Express.js based RESTful API
- JWT authentication
- Modular routing and middleware-based access control

Database:

- MongoDB with Mongoose ODM
- Collections: Users, Properties, Bookings, Admin Approvals

3. Setup Instructions

Prerequisites:

- Node.js ($\geq 14.x$)
- MongoDB Community Edition

Installation:

1. Clone the repo
2. Run “npm install” in both “/client” and “/server”
3. Set environment variables (`.env` file) for MongoDB URI and JWT secret

4. Folder Structure

➤ **Client (React Frontend):**

/client
 /src
 /components
 /pages
 /assets
 /services

App.js

➤ **Server (Node Backend):**

/server
 /controllers
 /models
 /routes
 server.js

5. Running the Application

Frontend:

- ✓ `cd client`
- ✓ `npm start`

Backend:

- ✓ `cd server`
- ✓ `npm start`

6. API Documentation

Sample Endpoints:

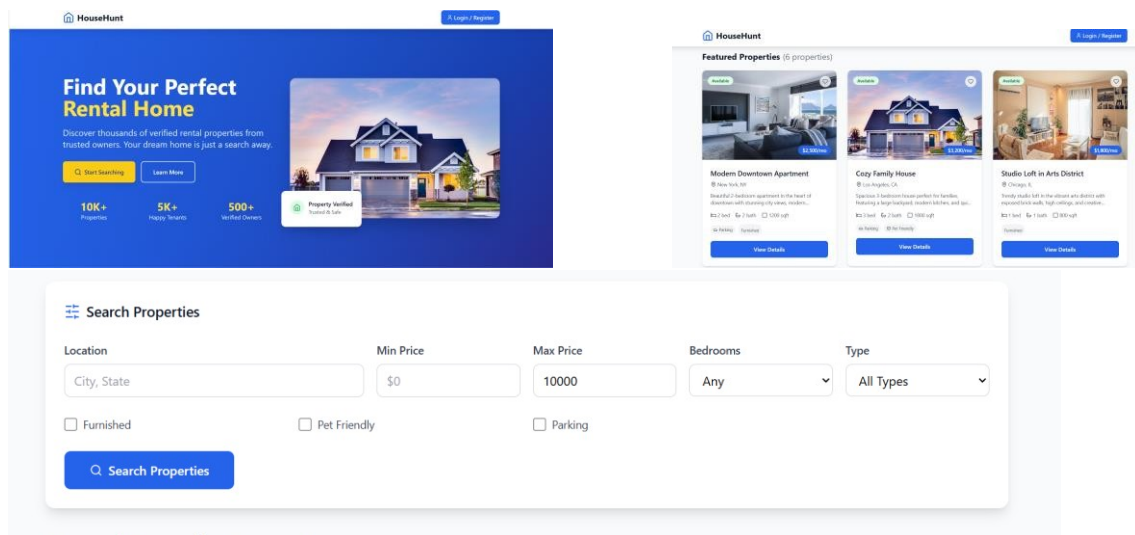
- `POST `/api/auth/register`` - User registration
- `POST `/api/auth/login`` - User login
- `GET `/api/properties`` - Fetch all properties
- `POST `/api/properties`` - Add new property (Owner only)
- `PUT `/api/bookings/:id`` - Booking update

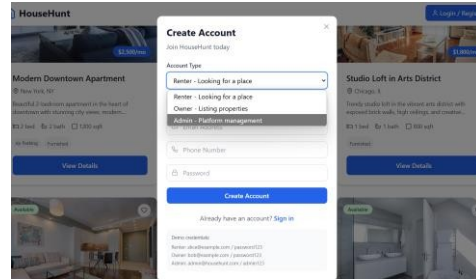
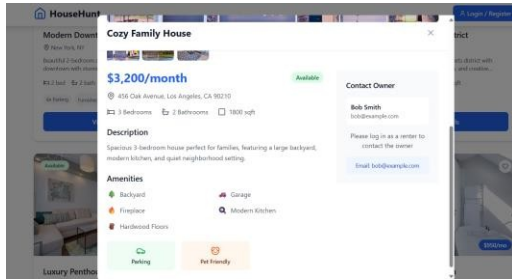
7. Authentication

- JWT-based token system
- Role-based access (Renter, Owner, Admin)
- Tokens stored in `localStorage`

8. User Interface

Screenshots:





9. Testing

Manual testing with demo credentials:

- Renter: <https://lustrous-rugelach-aa4d9b.netlify.app/>
- Owner: <https://lustrous-rugelach-aa4d9b.netlify.app/>
- Admin: <https://lustrous-rugelach-aa4d9b.netlify.app/>

10. Screenshots or Demo

Live: <https://lustrous-rugelach-aa4d9b.netlify.app/>

Screenshots:

- Homepage, Filters, Listings, Booking Modals, Contact Forms, etc.

11. Known Issues

- No integrated payment gateway
- Booking conflicts not handled via calendar
- Needs optimization for mobile responsiveness

12. Future Enhancements

- Payment integration (Stripe/PayPal)
 - Calendar-based booking
- Chat system between renter and owner
- Progressive Web App (PWA) version