HouseHunt: Finding Your Perfect Rental Home

# Full Stack MERN Application Documentation

## 1. Introduction

Project Title: HouseHunt: Finding Your Perfect Rental Home

Team Members:

-Vidya Prasunna(Team leader) - Full Stack Developer (Frontend, Backend, Database Design, Deployment)

-Botla Veerendra - Full Stack Developer (Frontend, Backend, Database Design, Deployment)  
- Bonam Chandra Durga Gowri Shankar – Full Stack Developer (Frontend, Backend, Database Design, Deployment)

## 2. Project Overview

Purpose:  
HouseHunt is a web-based house rental application designed to simplify and streamline the process of renting homes. It connects renters with property owners and offers a transparent, secure, and user-friendly platform.

Features:  
- User Authentication (Renter, Owner, Admin)  
- Property Listings with Images and Descriptions  
- Search & Filter (by location, budget, type, bedrooms)  
- Owner Property Management (Add/Edit/Delete)

## 3. Architecture

Frontend (React):  
Built using ReactJS with functional components and React Router for navigation.  
Axios used for HTTP requests to backend APIs.  
UI Libraries: Bootstrap, Material UI.

Backend (Node.js + Express.js):  
Express.js handles RESTful APIs for all operations (auth, properties, bookings, etc.).  
Role-based routes and middleware for JWT auth.

Database (MongoDB):  
Collections include users, properties, bookings, and messages.

## 4. Setup Instructions

Prerequisites:  
- Node.js (v18+)  
- MongoDB  
- npm or yarn

Installation Steps:  
1. git clone https://github.com/Shankars57/SB-House   
2. cd househunt  
  
Frontend Setup:  
cd client  
npm install  
  
Backend Setup:  
cd server  
npm install  
  
Environment Variables (.env in server):  
MONGO\_DB = 'mongodb://127.0.0.1:27017/'

JWT\_KEY = '12345'  
PORT=8001

## 5. Folder Structure

Client Folder:  
client/  
├── public/  
├── src/  
│ ├── components/  
│ ├── pages/  
│ ├── context/  
│ ├── App.js  
│ └── index.js

Server Folder:  
server/  
├── controllers/  
├── models/  
├── routes/  
├── middleware/  
├── config/  
├── server.js

## 6. Running the Application

Frontend:  
cd client  
npm start

Backend:  
cd server  
npm start

## 7. API Documentation

Sample Endpoints:  
- /api/auth/register [POST]: Register new user  
- /api/auth/login [POST]: Login user  
- /api/properties/ [GET]: Get all properties  
- /api/properties/:id [GET]: Get single property  
- /api/bookings/ [POST]: Create booking  
- /api/bookings/user [GET]: Renter bookings  
- /api/bookings/owner [GET]: Owner bookings

## 8. Authentication

JWT-based authentication.  
Role-Based Access Control via middleware.  
Tokens stored in localStorage and attached in headers.

## 9. User Interface

Highlights:  
- Filters and search  
- Booking forms

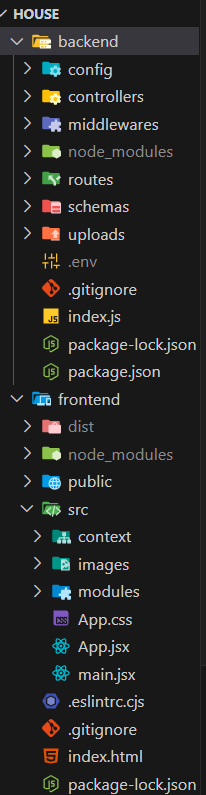
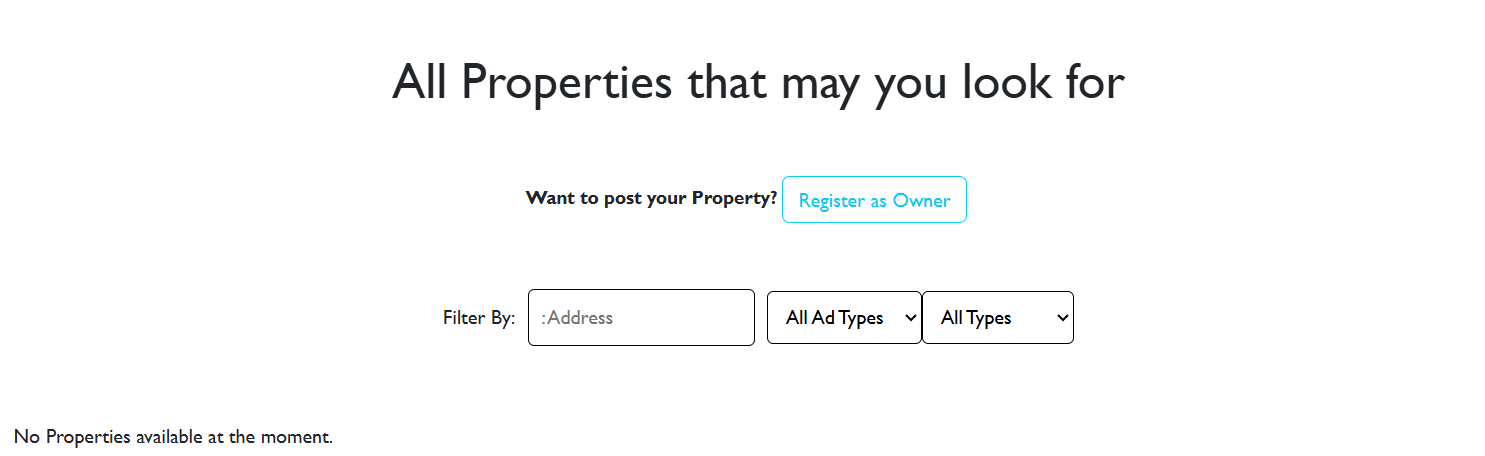
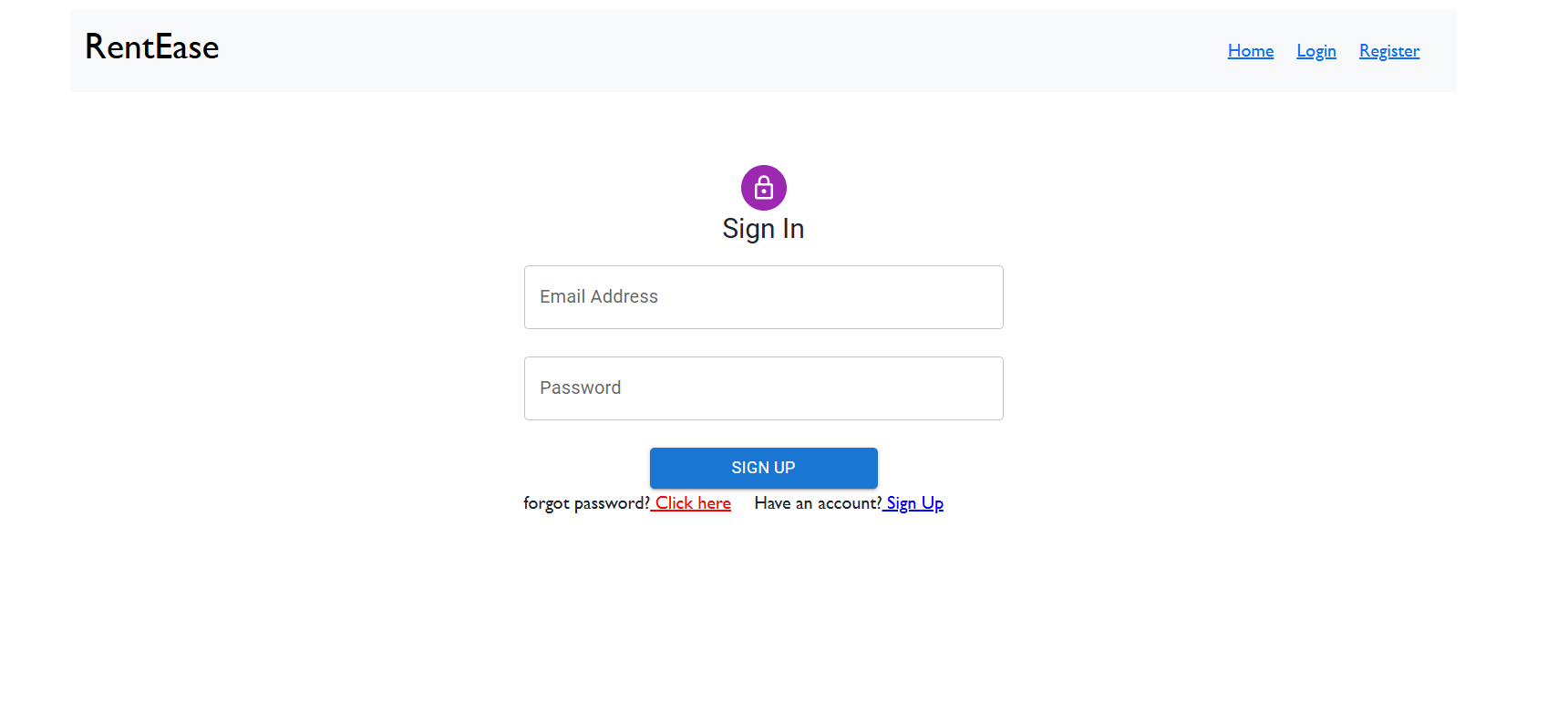
## 10. Testing

Tools:  
- Postman for API  
- Manual flow testing

## 11. Screenshots or Demo

Github: https://github.com/Shankars57/SB-House

Live Demo: <https://house-rental-project-tqa3.vercel.app/>



## 12. Known Issues

- No real-time chat yet  
- No multi-image upload support  
- Mobile view needs polishing

## 13. Future Enhancements

- Real-time chat (Socket.IO)  
- Push/email notifications  
- Payment integration  
- Lease generator  
- Admin analytics dashboard