HOUSEHUNT - YOUR PERFECT RENTAL HOME

Team Code: LTVIP2025TMID47660

Internship Project Report

INTRODUCTION

HouseHunt is a full-stack rental housing web application designed to streamline the process of searching and booking rental homes. Tenants can easily filter properties based on location, price, and amenities. Landlords can manage listings and bookings, while admins handle user approvals and platform governance.
Built using the MERN Stack (MongoDB, Express.js, React.js, Node.js), the platform offers secure authentication, responsive UI, and real-time data flow for an efficient user experience.

KEY FEATURES

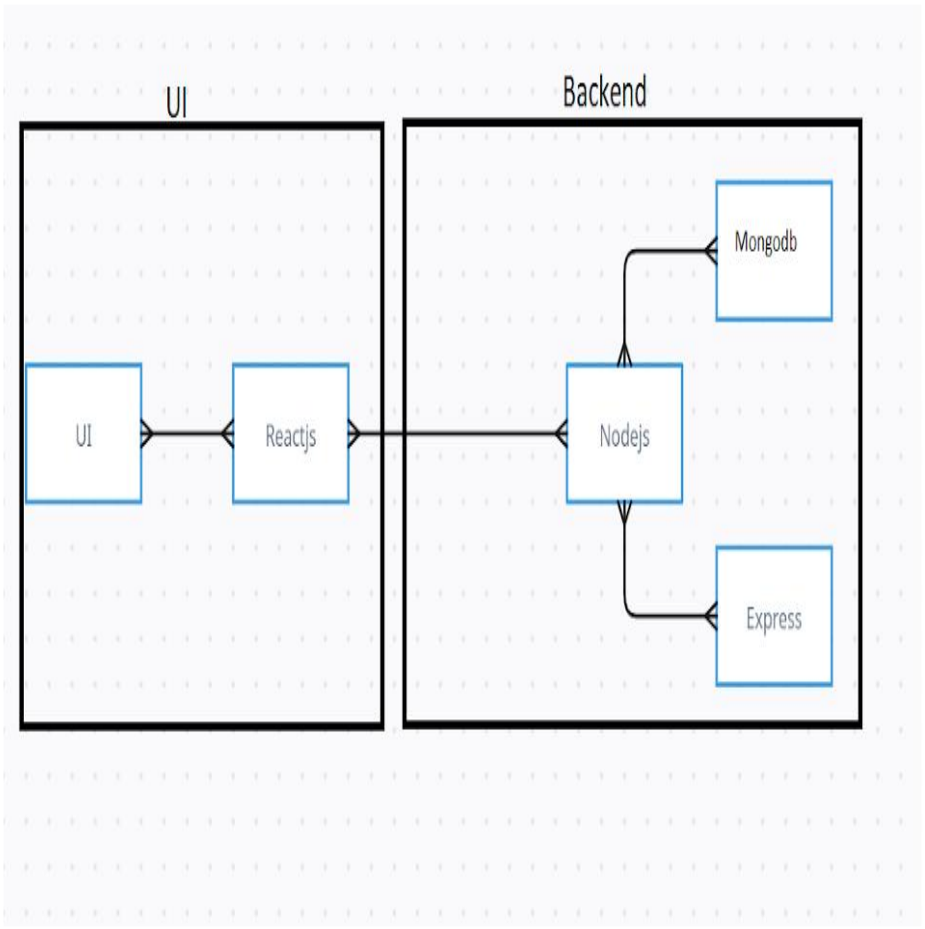
1. Landlord Dashboard
2. Booking System
3. Admin Panel

TECHNICAL ARCHITECTURE

The system is divided into UI and Backend:

PRE REQUISITES :
NODE.JS AND NPM:

● Node.js is a JavaScript runtime that allows you to run JavaScript code on the server-side. It provides a
scalable platform for network applications.
● npm (Node Package Manager) is required to install libraries and manage dependencies.
● Download Node.js: Node.js Download
● Installation instructions: Installation Guide
● Run npm init to set up the project and create a package.json file.



Frontend → ReactJS
Backend → Node.js, Express.js, MongoDB

This client-server model uses modern web technologies with RESTful API architecture and token-based authentication.

FRONTEND TECHNOLOGIES :

● **Bootstrap and Material UI:** Provide a responsive and modern UI that adapts to various devices,
ensuring a user-friendly experience.

● **Axios:** A promise-based HTTP client for making requests to the backend, ensuring smooth data
communication between the frontend and server.

BACKEND FRAMEWORK :

● **Express.js:** A lightweight Node.js framework used to handle server-side logic, API routing, and HTTP
request/response management, making the backend scalable and easy to maintain.

DATABASE AND AUTHENTICATION :

● **MongoDB**: A NoSQL database used for flexible and scalable storage of user data, doctor profiles, and
appointment records. It supports fast querying and large data volumes.
● **JWT (JSON Web Tokens):** Used for secure, stateless authentication, allowing users to remain logged
in without requiring session storage on the server.
● **Bcrypt**: A library for hashing passwords, ensuring that sensitive data is securely stored in the database.

ADMIN PANEL & GOVERNANCE :

● **Admin Interface:** Provides functionality for platform admins to approve doctor registrations, manage
platform settings, and oversee day-to-day operations.
● **Role-based Access Control (RBAC):** Ensures different users (patients, doctors, admins) have
appropriate access levels to the system's features and data, maintaining privacy and security.

SCALABILITY AND PERFORMANCE :

● **MongoDB:** Scales horizontally, supporting increased data storage and high user traffic as the platform
grows.
Load Balancing: Ensures traffic is evenly distributed across servers to optimise performance, especially during
high traffic periods.
● **Caching:** Reduces database load by storing frequently requested data temporarily, speeding up response
times and improving user experience.**TIME MANAGEMENT AND SCHEDULING**
● **Moment.js:** Utilised for handling date and time operations, ensuring precise appointment scheduling,
time zone handling, and formatting.

SECURITY FEATURES :

● **HTTPS**: The platform uses SSL/TLS encryption to secure data transmission between the client and
server.
● **Data Encryption:** Sensitive user information, such as medical records, is encrypted both in transit and
at rest, ensuring privacy and compliance with data protection regulations.
●

NOTIFICATIONS AND REMINDERS :

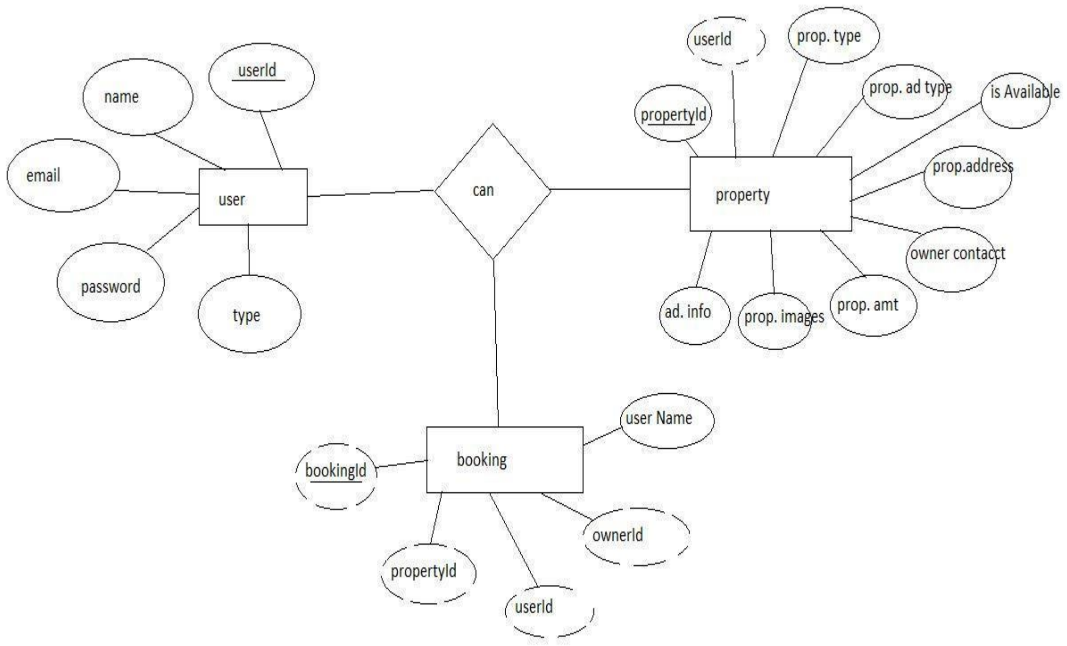
● **Email/SMS Integration:** Notifications for appointment confirmations, reminders, cancellations, and
updates are sent to users via email or SMS, ensuring timely communication

ENTITY RELATIONSHIP DIAGRAM (ERD)

ER diagram includes Users, Properties, Bookings, and Admin collections with respective relationships.

SCENARIO-BASED CASE STUDY

1. Registration: Alice signs up as a renter with email/password.
2. Owner Sign-up: Owner can sign up and post properties.
3. Browsing Listings: Alice filters
properties



.

4Sending Request: Alice submits a booking request for a flat.
5. Owner Review: Owner receives and approves the request.
6. Admin Role: Admin verifies landlord identities.
7. Platform Governance: Admin handles complaints and ensures policy compliance.
8. Lease Finalization: Alice and Bob finalize lease through in-app chat.

PRE REQUISITES :

NODE.JS AND NPM:

● Node.js is a JavaScript runtime that allows you to run JavaScript code on the server-side. It provides a
scalable platform for network applications.
● npm (Node Package Manager) is required to install libraries and manage dependencies.
● Download Node.js: Node.js Download
● Installation instructions: Installation Guide
● Run npm init to set up the project and create a package.json file.

EXPRESS.JS:

● Express.js is a web application framework for Node.js that helps you build APIs and web applications
with features like routing and middleware.
● Install Express.js to manage backend routing and API endpoints.
● Install Express:
● Run npm install express

MONGODB:

● MongoDB is a NoSQL database that stores data in a JSON-like format, making it suitable for storing
data like user profiles, doctor details, and appointments.
● Set up a MongoDB database for your application to store data.
● Download MongoDB: MongoDB Download
● Installation instructions: MongoDB Installation Guide

MOMENT.JS:

•
● Moment.js is a JavaScript package for handling date and time operations, allowing easy manipulation
and formatting.
● Install Moment.js for managing date-related tasks, such as appointment scheduling.● Moment.js Website: Moment.js Documentation

REACT.JS:

● React.js is a popular JavaScript library for building interactive and reusable user interfaces. It enables
the development of dynamic web applications.
● Install React.js to build the frontend for your application.
● React.js Documentation: Create a New React App

ANTD (ANT DESIGN):

● Ant Design is a UI library for React.js, providing a set of reusable components to create user-friendly
and visually appealing interfaces.
● Install Ant Design for UI components such as forms, tables, and modals.
● Ant Design Documentation: Ant Design React

HTML, CSS, AND JAVASCRIPT:

● Basic knowledge of HTML, CSS, and JavaScript is essential to structure, style, and add interactivity to
the user interface.

DATABASE CONNECTIVITY (MONGOOSE):

● Use Mongoose, an Object-Document Mapping (ODM) library, to connect your Node.js backend to
MongoDB for managing CRUD operations.
● Learn Database Connectivity: Node.js + Mongoose + MongoDB

FRONT-END FRAMEWORKS AND LIBRARIES:

● React.js will handle the client-side interface for managing doctor bookings, viewing appointment
statuses, and providing an admin dashboard.
● You may use Material UI and Bootstrap to enhance the look and feel of the application.

SETUP AND INSTALLATION INSTRUCTIONS :

CLONE THE PROJECT REPOSITORY:

● Download the project files from GitHub or clone the repository using Git.

INSTALL DEPENDENCIES:

● Navigate to the frontend and backend directories and install all required dependencies for both parts of
the application.
● Frontend:
● Navigate to the frontend directory and run npm install.
● Backend:
● Navigate to the backend directory and run npm install.

START THE DEVELOPMENT SERVER:

● After installing the dependencies, start the development server for both frontend and backend.
● Frontend will run on http://localhost:3000.
● Backend will run on http://localhost:8001 or the specified port.**ACCESS THE APPLICATION:**

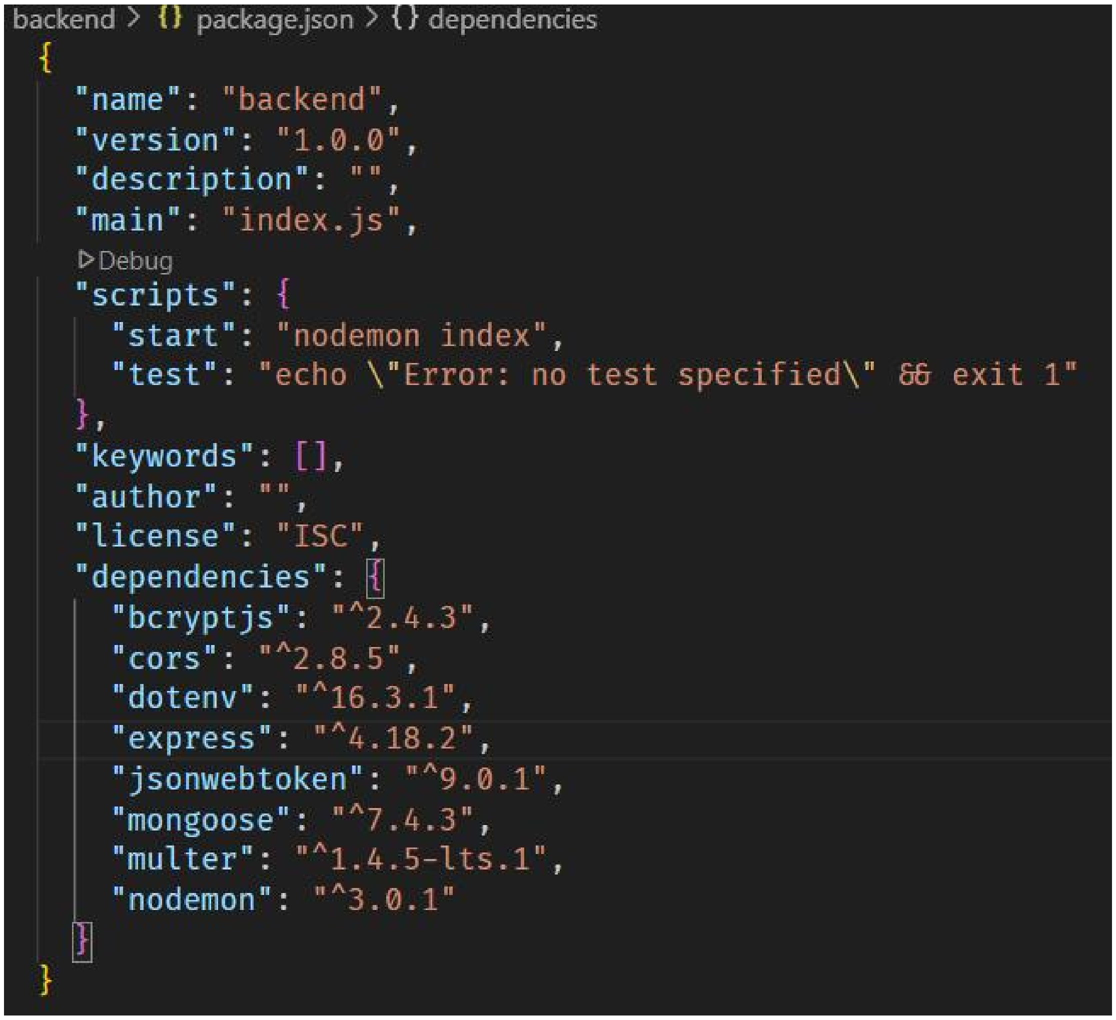
● After running the servers, access the Doctor Appointment Webpage in your browser at http://localhost:3000 for the frontend interface and http://localhost:8001 for backend API services.

PROJECT FOLDER STRUCTURE

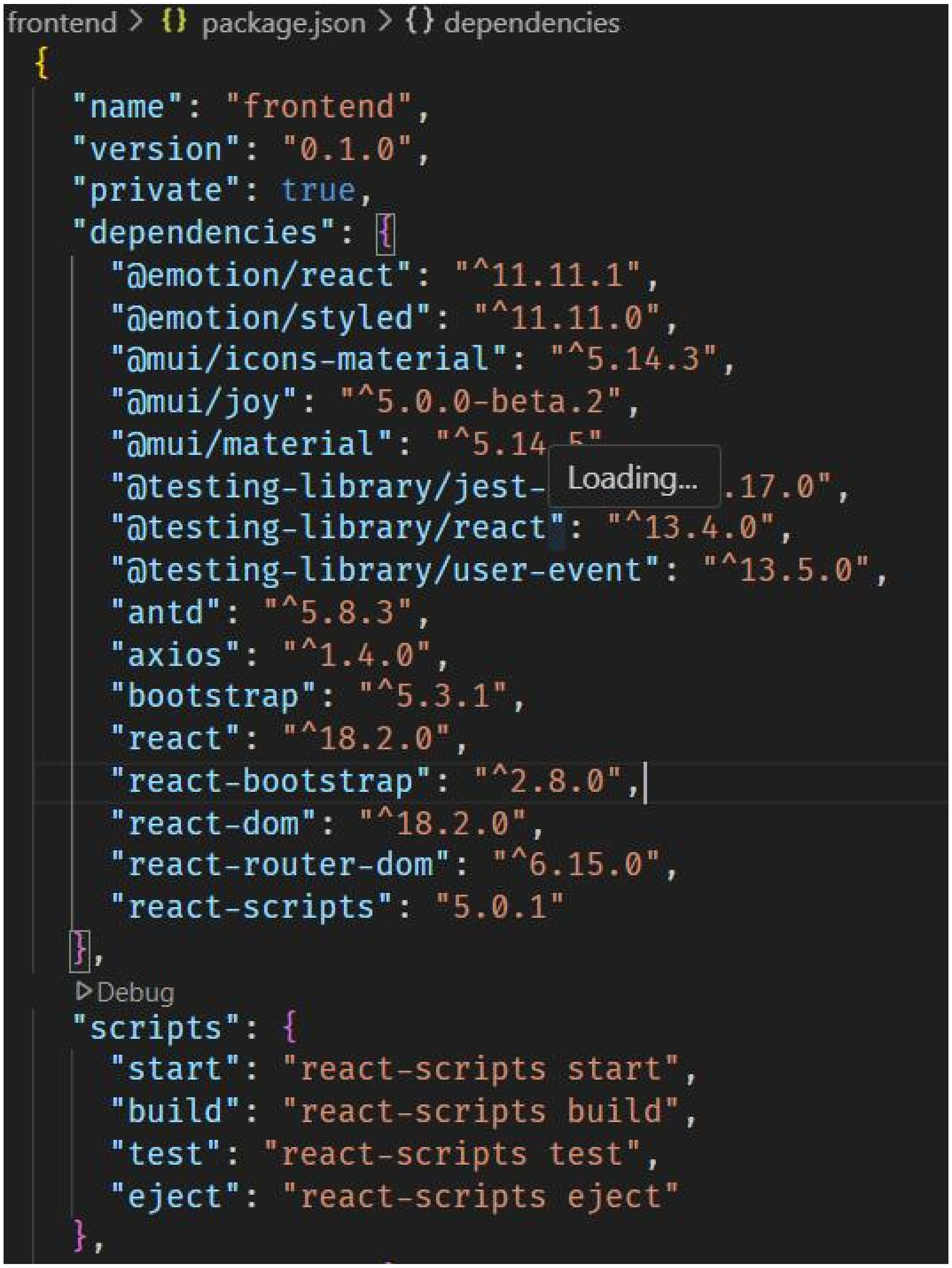
Backend:
```
├── config/connect.js
├── controllers/
```

├── middlewares/
├── routes/
├── schemas/
├── uploads/
├── .env, index.js

Frontend:
├── public/index.html
├── src/
  ├── images/
  ├── modules/admin/
  ├── modules/user/
  ├── modules/common/



```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷Debug
  "scripts": {
    "start": "nodemon index",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "jsonwebtoken": "^9.0.1",
    "mongoose": "^7.4.3",
    "multer": "^1.4.5-lts.1",
    "nodemon": "^3.0.1"
  }
}
```



```
frontend > {} package.json > {} dependencies
{
  "name": "frontend",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.11.1",
    "@emotion/styled": "^11.11.0",
    "@mui/icons-material": "^5.14.3",
    "@mui/joy": "^5.0.0-beta.2",
    "@mui/material": "^5.14 5"
    "@testing-library/jest- Loading... .17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^5.8.3",
    "axios": "^1.4.0",
    "bootstrap": "^5.3.1",
    "react": "^18.2.0",
    "react-bootstrap": "^2.8.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.15.0",
    "react-scripts": "5.0.1"
  },
  ▷ Debug
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
```

DEPENDENCIES

Backend:
- express
- mongoose
- jsonwebtoken

- bcryptjs
- multer
- cors
- dotenv
- nodemon

Frontend:
- react
- axios
- react-router-dom
- antd
- material-ui
- bootstrap
- react-bootstrap

SETUP & INSTALLATION

1. Git init and Clone:

```
git clone <repo_url>
```

Frontend:

```
cd frontend
npm install
npm start
```

Backend:

```
cd backend
npm install
npm start
```

MongoDB Setup:

Use MongoDB Atlas or local MongoDB instance
Configure .env with MONGO_URI and JWT_SECRET

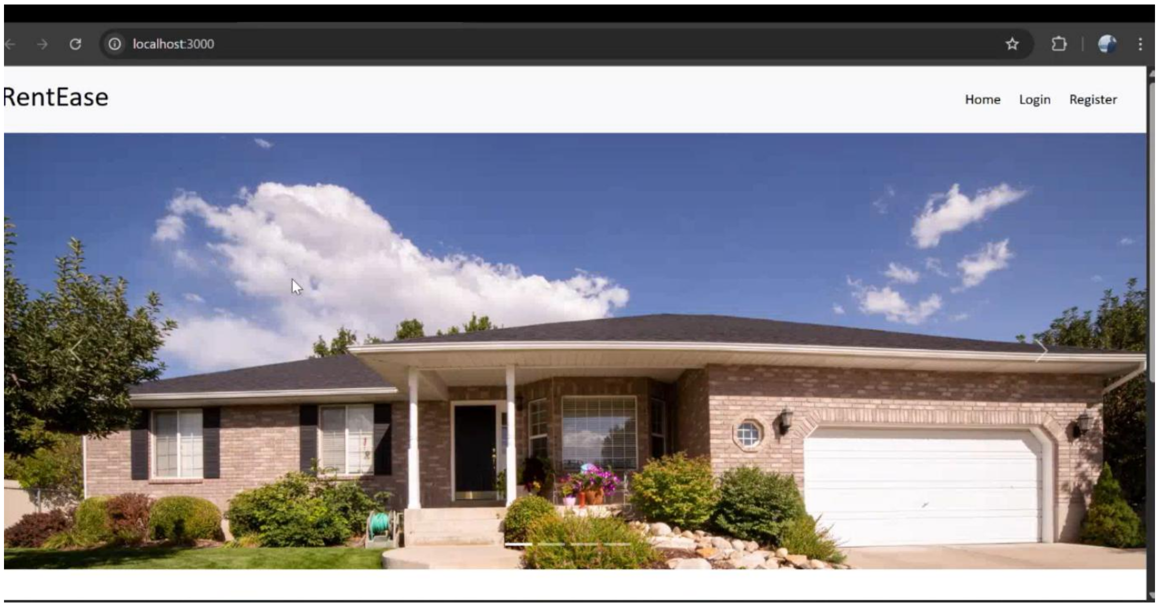
FEATURES

- Sign Up Page: Renter or Owner registration.
- Property Listings: Filterable list of available rentals.
- Landlord Dashboard: Manage properties and view requests.
- Booking Requests: View and approve/cancel booking requests.

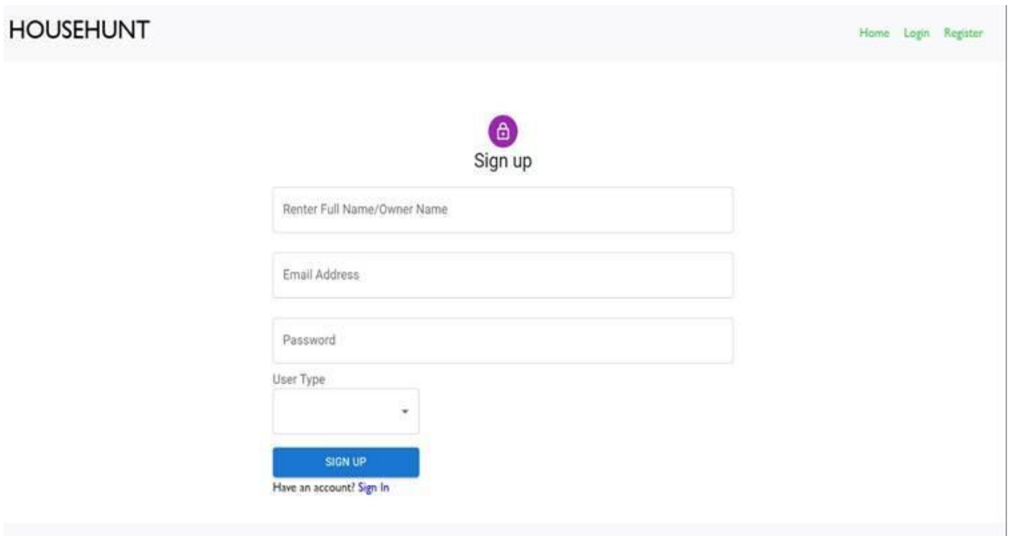
USER INTERFACE ELEMENTS:

Testing the UI includes verifying the look and feel of each page—landing, login, registration, and dashboards

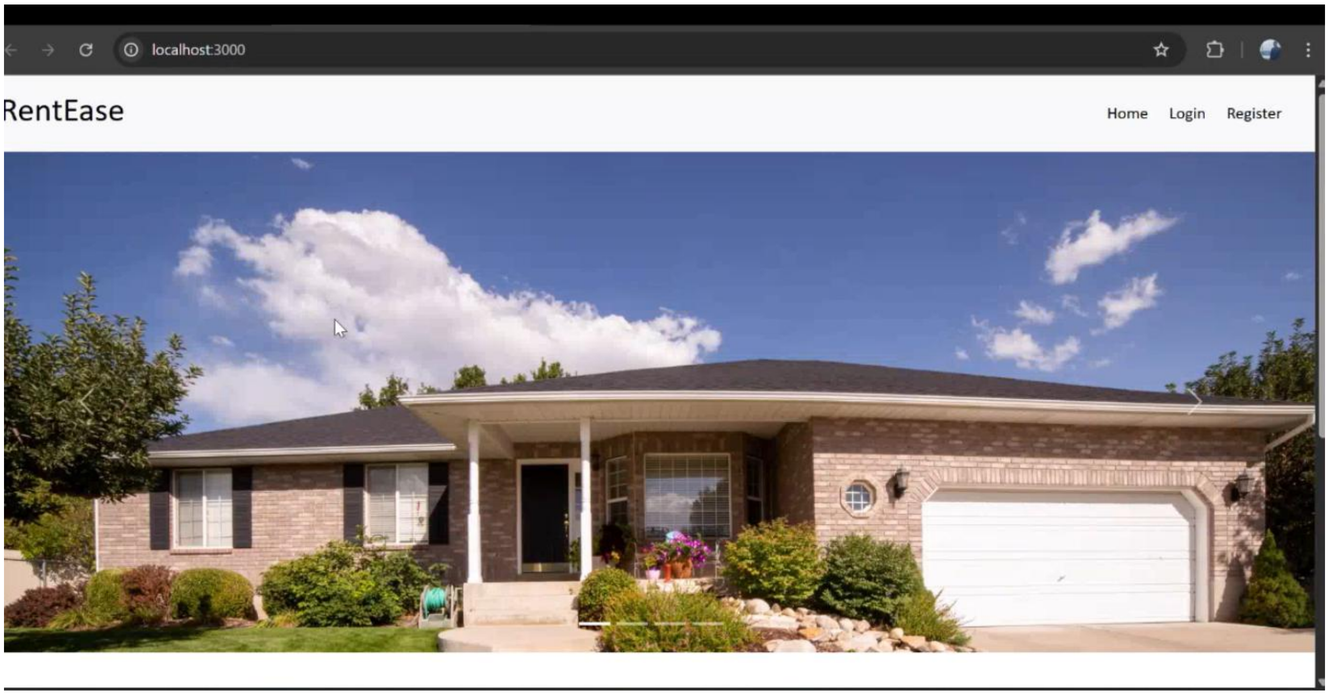
LANDING PAGE :



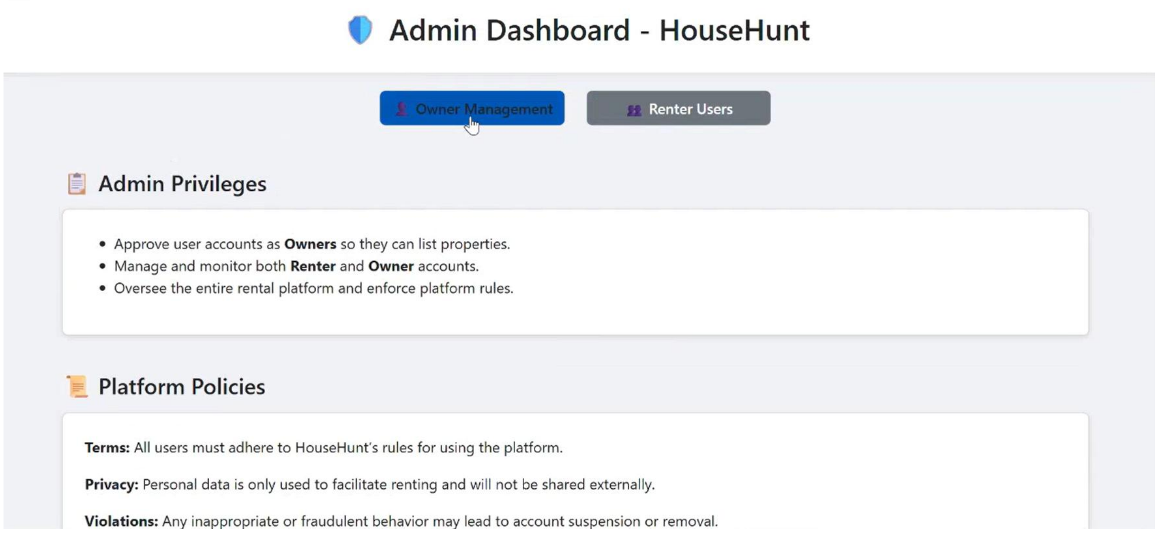
LOG IN-PAGE :



REGISTRATION PAGE :



ADMIN PAGE :



Admin Dashboard - HouseHunt

Admin Privileges

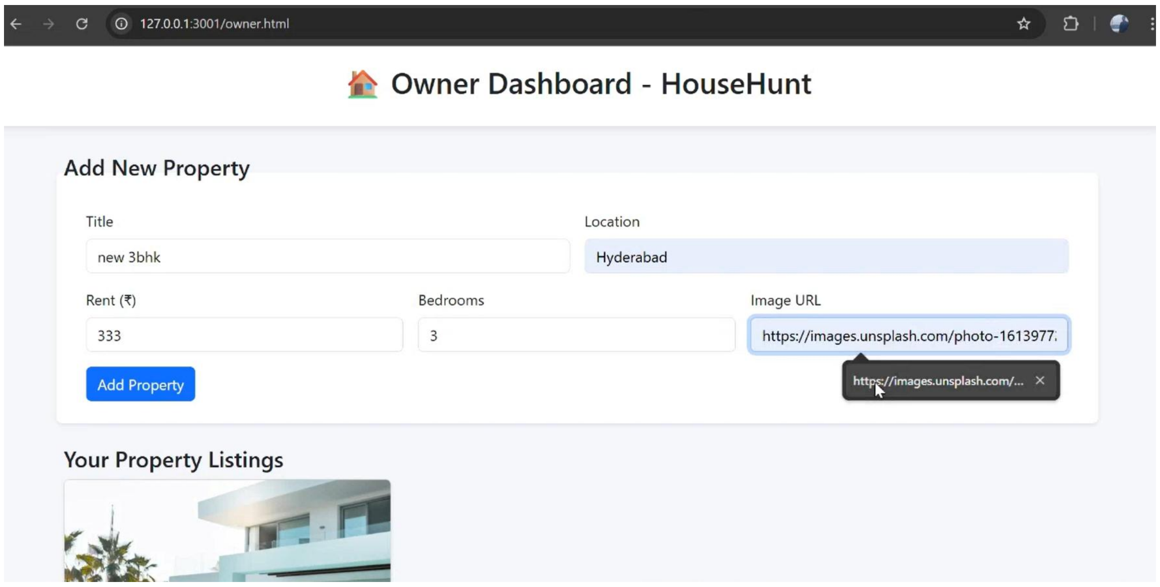
- Approve user accounts as **Owners** so they can list properties.
- Manage and monitor both **Renter** and **Owner** accounts.
- Oversee the entire rental platform and enforce platform rules.

Platform Policies

Terms: All users must adhere to HouseHunt's rules for using the platform.

Privacy: Personal data is only used to facilitate renting and will not be shared externally.

Violations: Any inappropriate or fraudulent behavior may lead to account suspension or removal.


127.0.0.1:3001/owner.html

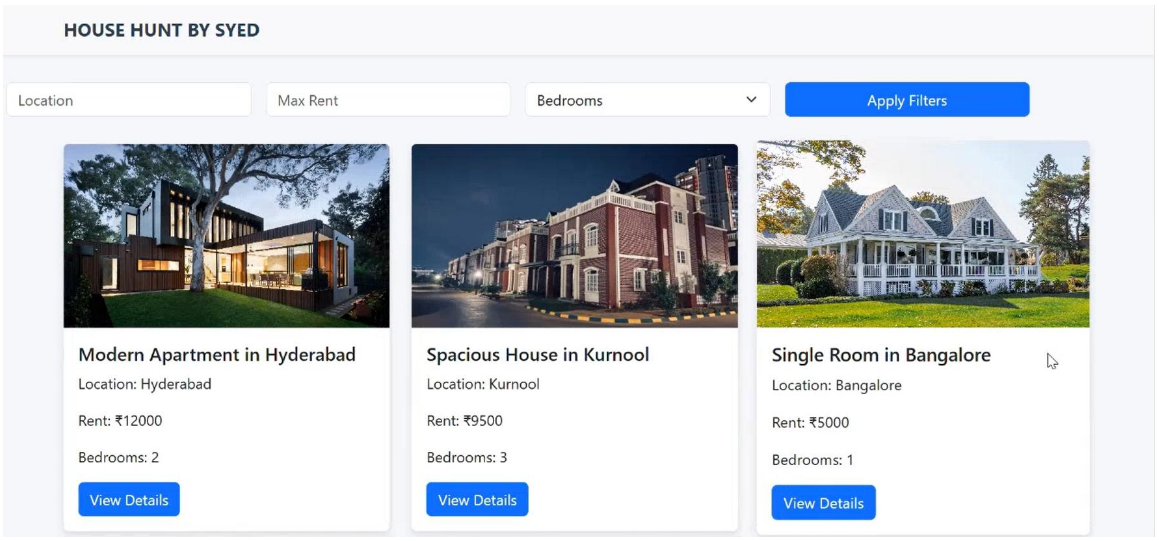
🏠 Owner Dashboard - HouseHunt

Add New Property

Title	new 3bhk
Location	Hyderabad
Rent (₹)	333
Bedrooms	3
Image URL	https://images.unsplash.com/photo-1613977

Your Property Listings

RENTER PAGE :

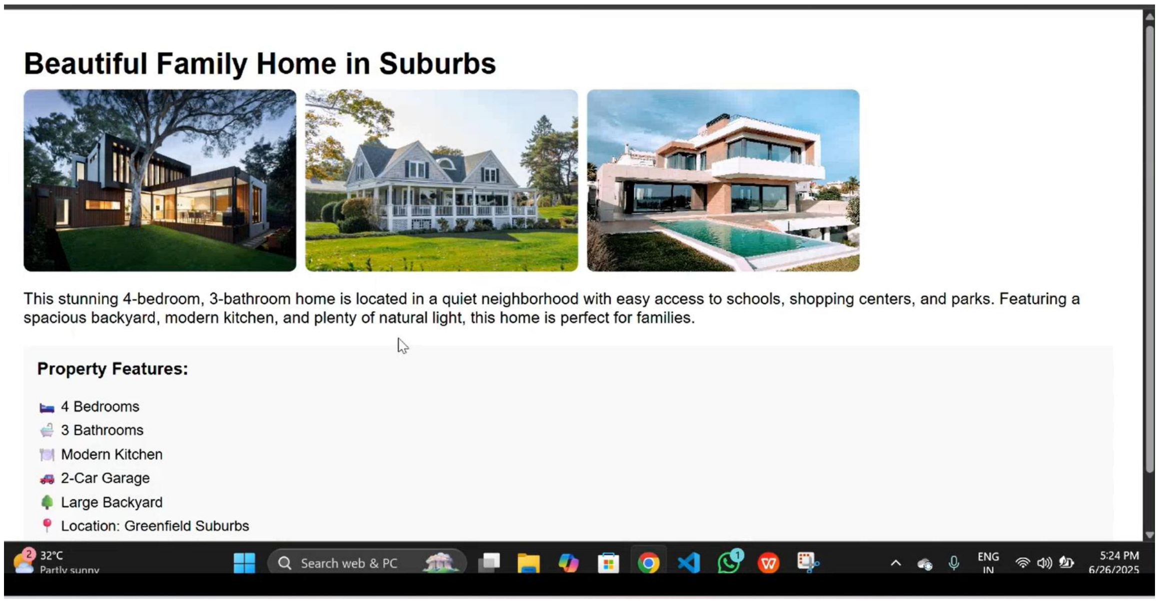

HOUSE HUNT BY SYED

Modern Apartment in Hyderabad
Location: Hyderabad
Rent: ₹12000
Bedrooms: 2

Spacious House in Kurnool
Location: Kurnool
Rent: ₹9500
Bedrooms: 3

Single Room in Bangalore
Location: Bangalore
Rent: ₹5000
Bedrooms: 1

RENT DETAILS :



Beautiful Family Home in Suburbs

This stunning 4-bedroom, 3-bathroom home is located in a quiet neighborhood with easy access to schools, shopping centers, and parks. Featuring a spacious backyard, modern kitchen, and plenty of natural light, this home is perfect for families.

Property Features:

- 🛏 4 Bedrooms
- 🛁 3 Bathrooms
- 🍽 Modern Kitchen
- 🚗 2-Car Garage
- 🌳 Large Backyard
- 📍 Location: Greenfield Suburbs

CONCLUSION

HouseHunt offers a complete rental solution using the MERN stack. It streamlines the tenant-landlord connection and provides secure, scalable, and real-time interaction between users. This project fulfills the goals of an efficient and modern rental home management system.