PROJECT FINAL REPORT

Project Title: HouseHunt: Finding Your Perfect Rental Home

Submitted by: Janani Appanabhotla
Lakshmi Gayathri Gunturu
Reshma Sri Macharla
Srija Gudivaka

Department: Computer Science and Engineering
College Name: DMSSVH College Of Engineering
Team Id: LTVIP2025TMID55971

Contents:

1. INTRODUCTION

In today's fast-paced and digitally driven world, the process of finding suitable rental property remains a significant challenge for many individuals and families. From the struggle of identifying legitimate listings to coordinating with multiple intermediaries, the traditional approach to house hunting is often time-consuming, inefficient, and emotionally draining.

To address these inefficiencies, "House Hunt – Finding Your Perfect Rental Home" emerges as a modern, intuitive solution designed to simplify the rental journey. This platform acts as a digital bridge between property owners and potential renters, offering a streamlined, user-friendly, and secure environment for all stakeholders involved. The goal is not to replace the values of traditional renting practices but to enhance and transform them through the integration of cutting-edge technology.

1.1 Project Overview

House Hunt is a comprehensive web-based application developed using the MERN stack—MongoDB, Express.js, React.js, and Node.js. It is designed to serve as a one-stop platform for rental property management and discovery, addressing the key pain points faced by both renters and property owners.

The application supports role-based access, allowing users to register and interact with the system as:

- Renters, who can browse and search for rental properties based on customizable filters such as location, budget, and amenities.
- Owners, who can list their properties, update availability, and manage booking requests.
- Administrators, who oversee platform activity, manage listings, and ensure compliance and security.

The system includes essential features such as secure user authentication, real-time property updates, responsive design for cross-device accessibility, and an admin dashboard to monitor user engagement and property data. By incorporating these functionalities, House Hunt provides seamless experience for users and fosters trust and transparency in the rental process.

Whether it is a student searching for affordable accommodation near a university, a working professional relocating to a new city, or a landlord seeking to advertise a vacant home—House Hunt ensures a smooth, efficient, and reliable rental experience for all.

1.2 Purpose

The primary purpose of House Hunt is to streamline the property rental process, making it more accessible, transparent, and efficient. Traditional house-hunting often involves middlemen, scattered listings, and lack of real-time updates. This platform seeks to eliminate those hurdles by:

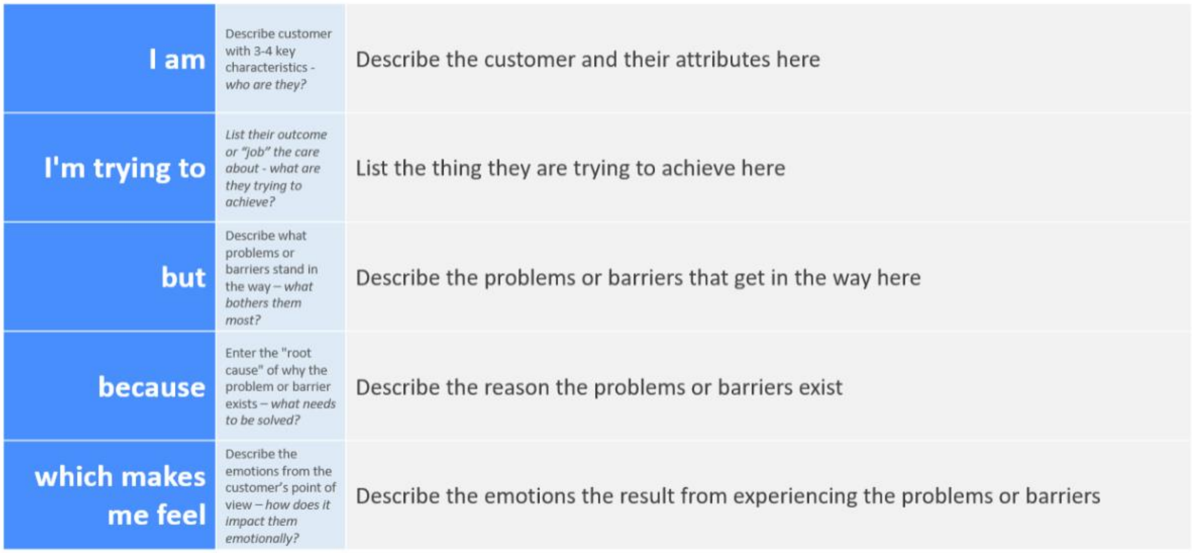
- Offering a centralized space for renters and owners to connect.
- Minimizing communication gaps and reducing dependency on brokers.
- Providing renters with filtered search options based on location, price, and amenities.
- Empowering owners to manage property listings and approve bookings directly.
- Giving admins tools to monitor user activity, approve listings, and ensure platform security.

Through this project, we aim to blend the reliability of traditional renting practices with the efficiency of modern web applications, creating a balanced and trustworthy rental ecosystem.

2. IDEATION PHASE

The ideation phase marks the foundation of the project's conceptual development. It involves identifying the real-world challenges in the current house rental process, understanding user needs, and exploring creative solutions that bridge the gap between problems and technology.

2.1 Problem Statement



I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

PS-1

I am a student or working professional relocating to a new city.

I'm trying to: find a clean, affordable rental property that suits my budget and preferences.

But: I keep finding outdated or fake listings across multiple websites, and there's no centralized or trusted platform.

Because: most rental platforms don't verify listings or provide real-time availability and direct communication with the property owner.

Which makes me feel frustrated, uncertain, and anxious about making the wrong decision or being scammed.

PS-2

I am: A property owner looking to rent out my house or apartment.

I'm trying to: list my property online and connect with genuine tenants quickly.

But: I find it difficult to manage multiple listings, track inquiries, and filter serious renters from casual ones.

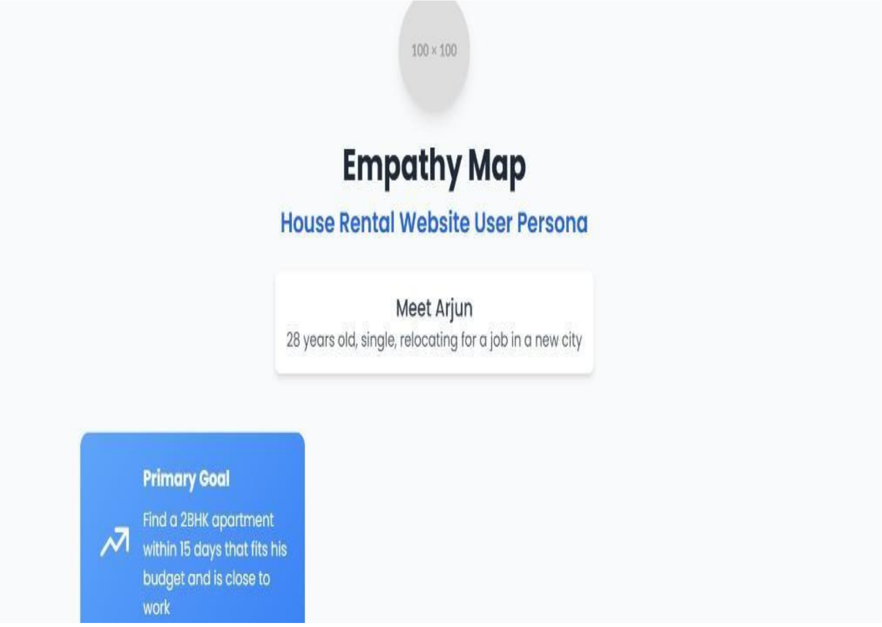
Because: there is no streamlined platform that offers admin approval, booking management, and renter verification.

Which makes me feel: overwhelmed and uncertain about whether my property will be rented safely and efficiently.

2.2 Empathy Map Canvas

Empathy Map: An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges. **EXAMPLE**: **House Hunt: Finding Your Perfect Rental Home**



Empathy Map

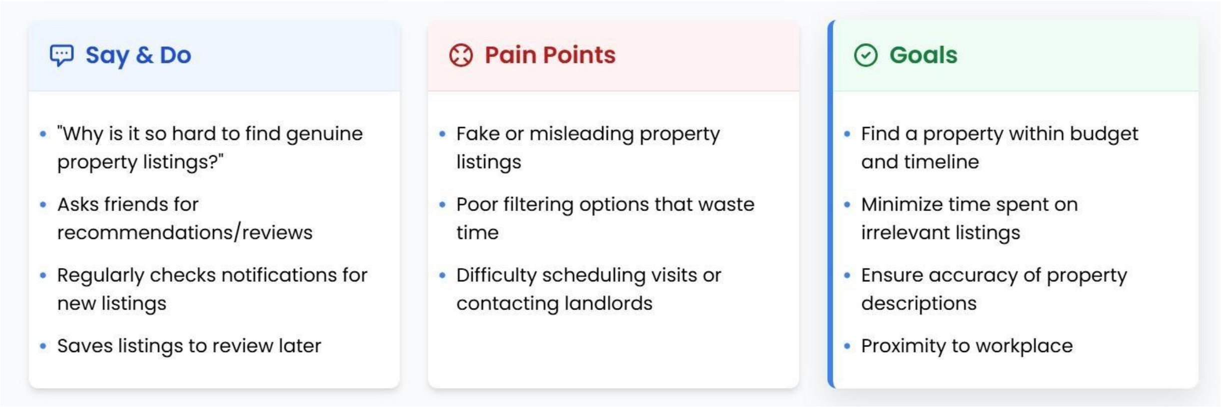
House Rental Website User Persona

Meet Arjun

28 years old, single, relocating for a job in a new city

Primary Goal

Find a 2BHK apartment within 15 days that fits his budget and is close to work



Say & Do

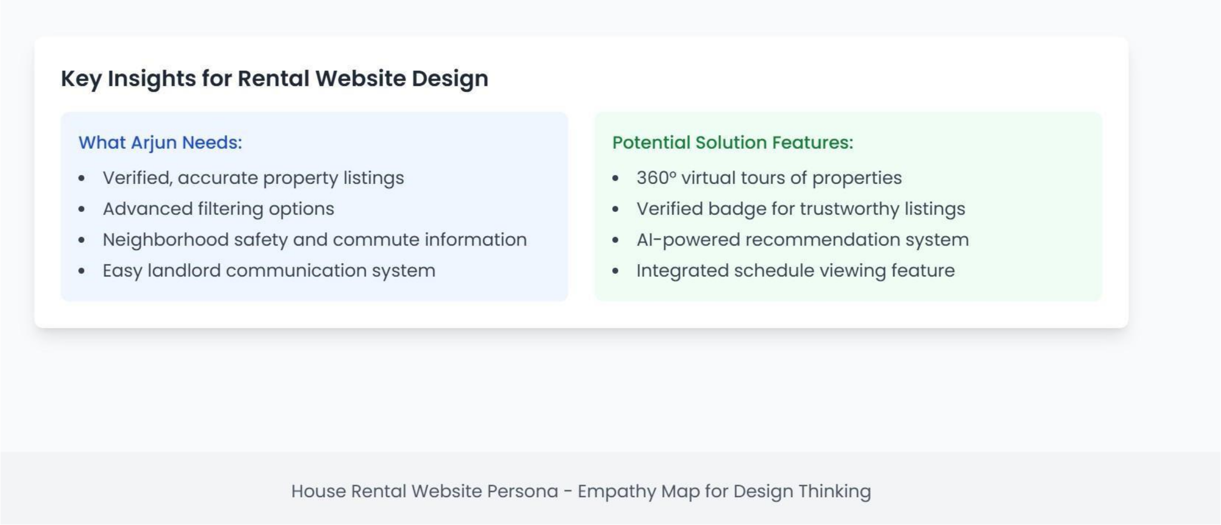
- "Why is it so hard to find genuine property listings?"
- Asks friends for recommendations/reviews
- Regularly checks notifications for new listings
- Saves listings to review later

Pain Points

- Fake or misleading property listings
- Poor filtering options that waste time
- Difficulty scheduling visits or contacting landlords

Goals

- Find a property within budget and timeline
- Minimize time spent on irrelevant listings
- Ensure accuracy of property descriptions
- Proximity to workplace



Key Insights for Rental Website Design

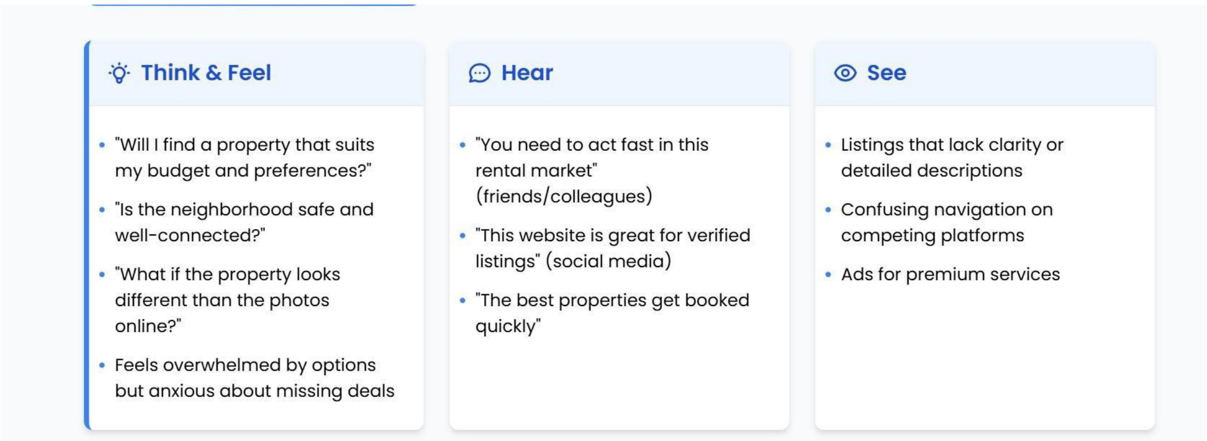
What Arjun Needs:

- Verified, accurate property listings
- Advanced filtering options
- Neighborhood safety and commute information
- Easy landlord communication system

Potential Solution Features:

- 360° virtual tours of properties
- Verified badge for trustworthy listings
- AI-powered recommendation system
- Integrated schedule viewing feature

House Rental Website Persona - Empathy Map for Design Thinking



Think & Feel

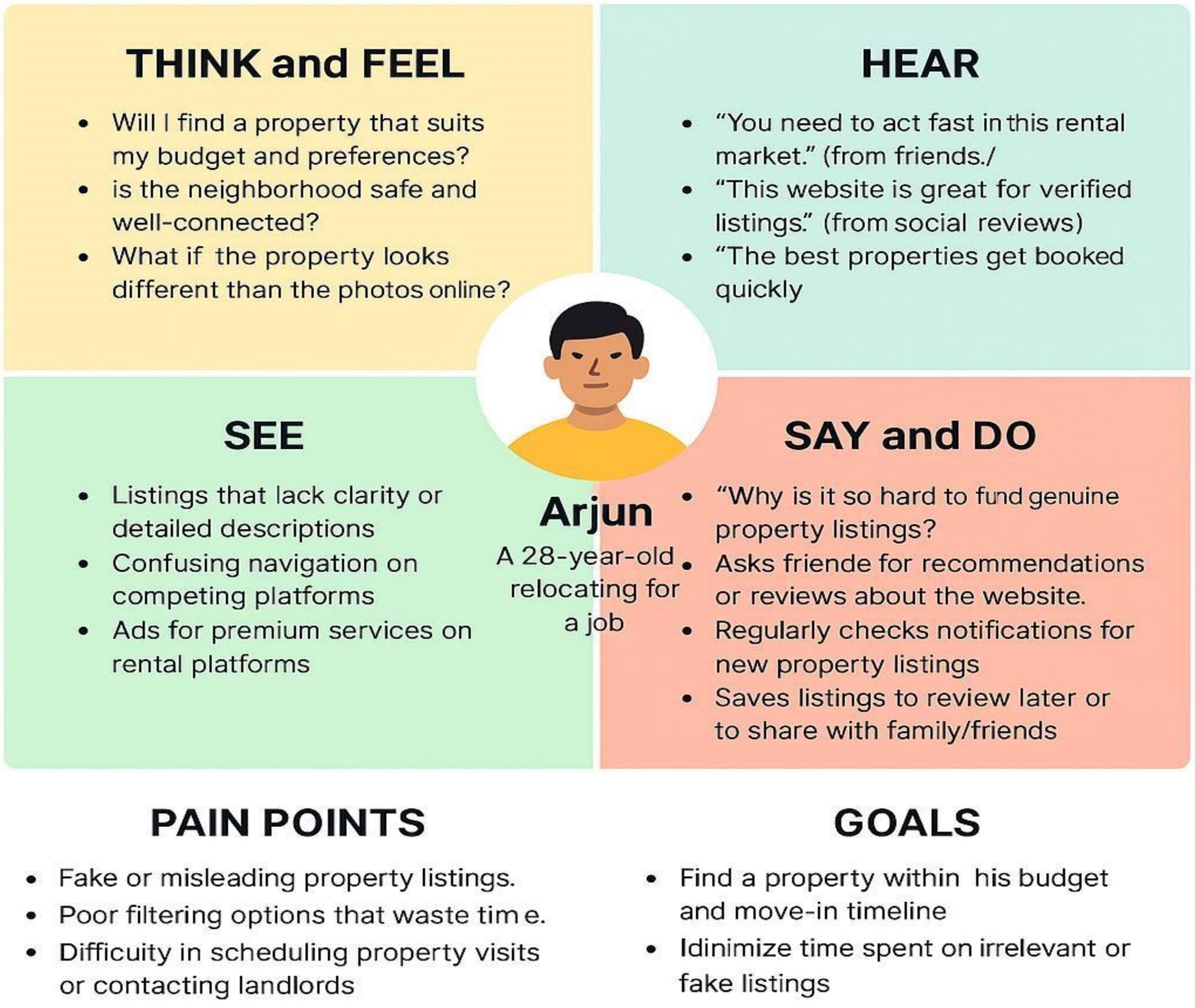
- "Will I find a property that suits my budget and preferences?"
- "Is the neighborhood safe and well-connected?"
- "What if the property looks different than the photos online?"
- Feels overwhelmed by options but anxious about missing deals

Hear

- "You need to act fast in this rental market" (friends/colleagues)
- "This website is great for verified listings" (social media)
- "The best properties get booked quickly"

See

- Listings that lack clarity or detailed descriptions
- Confusing navigation on competing platforms
- Ads for premium services



THINK and FEEL

- Will I find a property that suits my budget and preferences?
- is the neighborhood safe and well-connected?
- What if the property looks different than the photos online?

HEAR

- "You need to act fast in this rental market." (from friends./
- "This website is great for verified listings." (from social reviews)
- "The best properties get booked quickly

Arjun

A 28-year-old relocating for a job

SEE

- Listings that lack clarity or detailed descriptions
- Confusing navigation on competing platforms
- Ads for premium services on rental platforms

SAY and DO

- "Why is it so hard to fund genuine property listings?
- Asks friende for recommendations or reviews about the website.
- Regularly checks notifications for new property listings
- Saves listings to review later or to share with family/friends

PAIN POINTS

- Fake or misleading property listings.
- Poor filtering options that waste tim e.
- Difficuity in scheduling property visits or contacting landlords

GOALS

- Find a property within his budget and move-in timeline
- Idinimize time spent on irrelevant or fake listings

Reference: https://www.mural.co/templates/empathy-map-canvas

2.3 Brainstorming

Step 1: Team Gathering, Collaboration, and Select the Problem Statement

Team Members:

- Janani
- Gayathri
- Reshma
- Srija

Selected Problem Statement:

Renters face difficulty finding suitable, trustworthy rental properties with real-time availability, while property owners struggle to reach genuine tenants through existing platforms.

Step 2: Brainstorm, Idea Listing, and Grouping

Group	Ideas Generated
Property Listing Features	- Advanced filters (location, price, size) - Image galleries and 360° virtual tours - Owner profile visibility
Communication & Booking	- Direct booking requests - Admin-approved communication - Real-time booking status ("Pending", "Approved")
Admin Features	- Owner verification and approval - Dashboard for managing properties and bookings - Flag/report system for fraudulent listings
User Experience	- Push/email alerts for new listings or price drops - Saved searches - Easy sign-up/login with email, Gmail, or Facebook
Business Model	- Premium listing option for property owners - In-app ads - Value-added services (e.g., document verification or legal support)

Step 3: Idea Prioritization

Idea	Value to Customer	Feasibility (Team Capacity)	Priority
Direct property booking and real-time status	High	High	High
Admin approval system for owners	High	Medium	High
Advanced property filters with virtual tours	High	Medium	Medium
Owner dashboard for property management	Medium	Medium	Medium
Push/email alerts for renters	Medium	Medium	Low
Premium listing and ads revenue model	High	Low	Low

3. REQUIREMENT ANALYSIS

Requirement analysis is the phase where ideas begin to solidify into actionable functionalities. It bridges the gap between what users expect and what the application must deliver. In this stage, we carefully gather, analyze, and structure both functional and non-functional requirements to guide the app's development. The goal is to ensure House Hunt not only addresses the technical problem of connecting renters and owners—but also ensures a seamless, secure, and pleasant user experience.

3.1 Customer Journey Map

Stage	User Goal	User Action	System Response	Pain Points	Opportunities
Awareness	Find a platform to rent a house	Searches online for rental apps	House Hunt appears in search results	Overwhelmed by too many options	SEO & marketing to improve discoverability
Consideration	Explore the platform	Visits website, browses features	Shows homepage with search bar, "Browse Properties" button	Confusion in navigation	Ensure clean, intuitive UI
Onboarding	Register and create an account	Clicks "Sign Up", fills in details	Confirms registration, logs in	Signup failures, complex forms	Simple forms, validation messages
Search	Looking for available houses based on filters	Apply filters: location, budget, amenities	Shows relevant property listings	Not enough matching listings	Use tags, categories, and advanced filters
Engagement	Book a house	Views listing → clicks "Book Now"	Sends request to owner, shows confirmation	Uncertainty about owner response	Show expected time for confirmation, real-time updates
Post-Booking	Waiting for owner's approval	Checks booking status	Updates automatically when approved	Long wait times	Add in-app notifications, optional chat with owner
Closure	Move in / Exit platform	Confirms booking success and logs out	Thank you, message, optional feedback request	No follow-up	Encourage rating the property or referring to others

3.2 Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration via Form Registration via Gmail (optional)
FR-2	User Login & Authentication	Login using email & password Role-based login (Admin, Owner, Renter)
FR-3	Property Management	Owner can Add / Edit / Delete property View property list
FR-4	Property Search	Renter can search by filters (location, price, amenities) Save searches
FR-5	Booking Process	Renter submits booking request Owner updates booking status
FR-6	Admin Management	Admin approves users as Owners Admin monitors property listings
FR-7	Notifications	Email alerts on booking & approval Price drop alerts
FR-8	View Booking History	Renter can view previous/current bookings Status shown as Pending / Approved / Rejected
FR-9	Media Upload	Owner uploads property images Supports multiple images
FR-10	Dashboard	Individual dashboards for Renter, Owner, and Admin

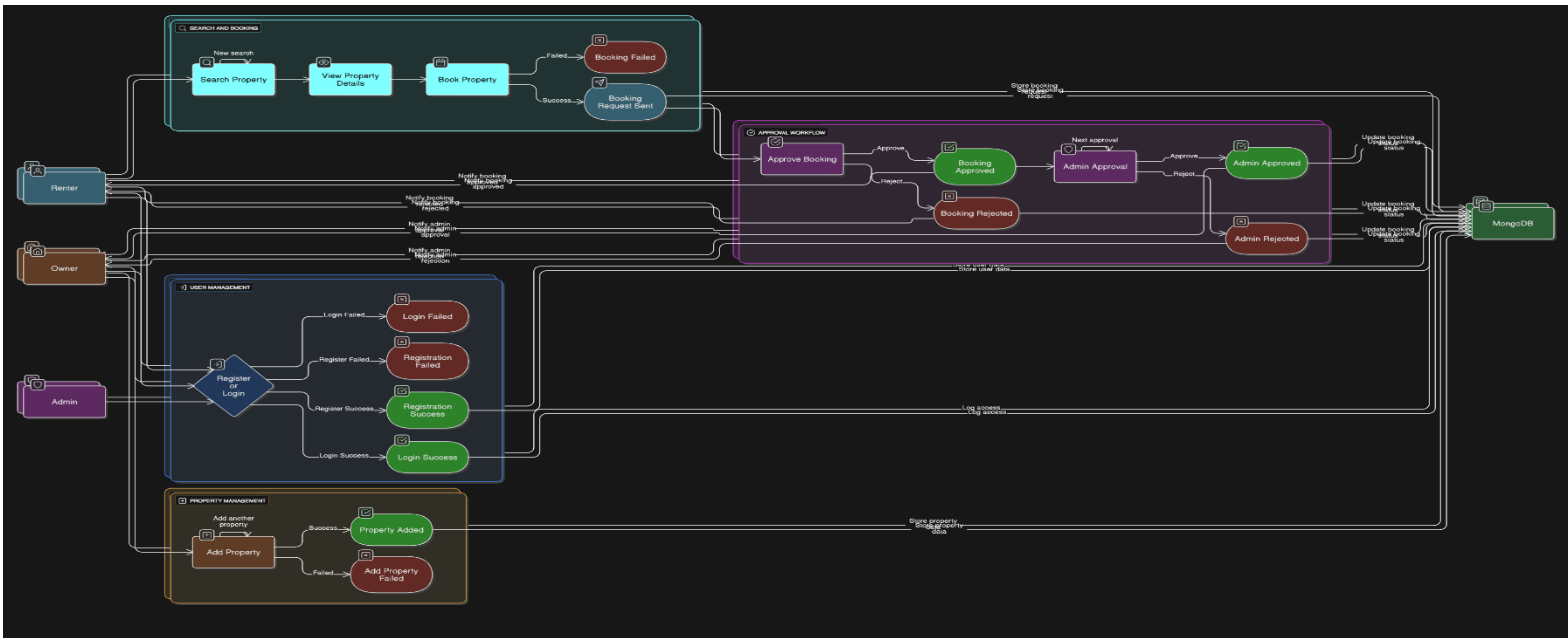
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Intuitive UI using React, Material UI, and Bootstrap for a smooth user journey
NFR-2	Security	Secure authentication with JWT, encrypted passwords using bcryptjs
NFR-3	Reliability	Consistent MongoDB storage, error-handling with Express.js
NFR-4	Performance	Fast load times, Axios for efficient API requests, optimized routing
NFR-5	Availability	24/7 access on browser; can be hosted on cloud platforms like Vercel/Heroku
NFR-6	Scalability	Built on scalable MERN architecture, modular codebase for future upgrades

3.3 Data Flow Diagram (DFD)

Data Flow Diagrams:
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Renter	Registration	USN-1	As a renter, I can register with email and password	I can see a success message and access my dashboard	High	Sprint-1
Renter	Booking	USN-2	As a renter, I can book a property after viewing its details	I can see the booking status and get confirmation	High	Sprint-2
Renter	Booking Status	USN-3	As a renter, I can see my booking status under the Booking section	I see status as Pending/Approved	High	Sprint-2
Owner	Add Property	USN-4	As an owner, I can add properties after admin approval	My properties are listed on the platform	High	Sprint-2

Owner	Booking Management	USN-5	As an owner, I can update the status of bookings	Status is updated and visible to the renter	High	Sprint-2
Admin	User Approval	USN-6	As an admin, I can approve owner accounts	Approved users can start adding properties	High	Sprint-1
Admin	Platform Monitoring	USN-7	As an admin, I can monitor users and properties	I see all user activity in the dashboard	High	Sprint-2

3.4 Technology Stack

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

The technical architecture of our House rent app follows a client-server model, where the front end serves as the client and the backend acts as the server.

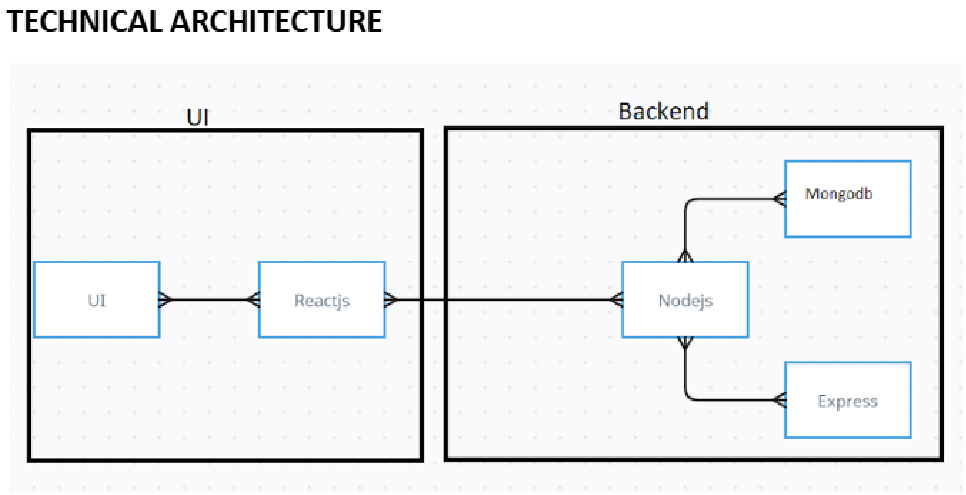


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web interface for Renter, Owner, Admin	HTML, CSS, JavaScript, React.js, Antd
2	Application Logic-1	Handles client-side routing and UI states	React.js
3	Application Logic-2	Handles backend routing and REST APIs	Node.js, Express.js
4	Application Logic-3	Authentication, CRUD for bookings and properties	JWT, bcryptjs, MongoDB Mongoose
5	Database	Houses renters, owners, bookings, property details	MongoDB (NoSQL)
6	Cloud Database	Can be hosted on MongoDB Atlas	MongoDB Atlas (Optional)
7	File Storage	For image uploads of rental properties	Multer, Local Filesystem
8	External API-1	Maps, address geolocation	Google Maps API (optional integration)
9	External API-2	Email alerts / push notifications	Firebase Cloud Messaging (optional)
10	Infrastructure	Deployment and hosting	Localhost for dev, can be Docker/Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	MERN stack, Axios, Antd, Bootstrap, Moment.js	React.js, Express.js, MongoDB, Node.js
2	Security Implementations	Password encryption, JWT for session control, Role-based access	bcryptjs, JWT, dotenv, CORS
3	Scalable Architecture	Tiered design with modularity to plug microservices in future	MERN, REST APIs, Docker (optional)
4	Availability	Can be hosted on cloud with load balancers, Node clustering	Express.js with PM2, MongoDB Replica Sets
5	Performance	Optimized API calls, database indexing, caching planned	Axios, Mongoose indexing, Lazy loading

4. PROJECT DESIGN

For House Hunt, the design phase aims to create a robust architecture that supports user needs while ensuring scalability, maintainability, and security.

This section elaborates on the alignment between the problem and the proposed solution, supported by architectural decisions that bring the platform to life.

4.1 Problem-Solution Fit

Problem Statement:
Renters often face challenges finding suitable rental homes that meet their needs in terms of budget, location, amenities, and reliability. The current market lacks centralized platforms offering verified property listings, easy booking workflows, and real-time status updates. On the other hand, property owners struggle to showcase their listings effectively and connect with genuine tenants.

Proposed Solution:

House Hunt is a full-stack MERN web platform that streamlines the rental property process for both renters and owners. It provides:

- A powerful search tool with filters for price, amenities, location, etc.
- Interactive listings with images, floor plans, and virtual tours.
- A booking system that shows real-time status updates.
- Admin approval flow for owners and renters to ensure platform trust.
- Dashboards for renters and property owners to manage bookings and listings.
- Alerts for new properties and price drops.

Fit with Customer Behavior:

- **Renters** today prefer online platforms to browse and book properties; HouseHunt matches this behavior by offering a user-friendly interface accessible 24/7.
- **Owners** want visibility and ease of managing listings; HouseHunt offers them tools to do so, including adding/updating properties and tracking bookings.
- **Trust and transparency** are addressed by admin moderation and clear property data, which aligns with modern expectations in online real estate services.

Why This Works:

- Solves a **frequent and frustrating problem** faced by young professionals, students, and relocating families.
- Taps into existing online behavior: people already use web platforms to search homes — HouseHunt enhances this with verified listings and instant booking.
- Provides a **one-stop, real-time rental ecosystem** that builds trust, simplifies tasks, and supports all user roles — renter, owner, and admin.

4.2 Proposed Solution

Proposed Solution : The proposed solution is a full-stack web application using the MERN stack (MongoDB, Express.js, React.js, Node.js) to deliver an interactive, fast, and scalable rental platform.

S.No.	Parameter	Description
1	**Problem Statement**	The current process of finding rental properties is often time-consuming, unorganized, and lacks transparency. Renters struggle to find suitable homes, while property owners face difficulties reaching the right tenants.
2	**Idea / Solution Description**	*HouseHunt* is a full-stack MERN (MongoDB, Express.js, React.js, Node.js) web application that bridges the gap between renters and property owners. It offers advanced property search, virtual tours, booking management, and real-time alerts. The platform also enables admins to approve users and manage the ecosystem for a smooth experience.
3	**Novelty / Uniqueness**	HouseHunt combines real-time booking status updates, admin approval workflows, and support for both renters and property owners in a single platform. It also offers features like 360° property tours, price trends, and owner profiles — all within a clean and responsive user interface built using React, Bootstrap, and Ant Design.
4	**Social Impact / Customer Satisfaction**	HouseHunt reduces stress and time for individuals seeking rental homes, especially for students, professionals, and migrating families. It also empowers property owners to showcase their listings easily. The transparency and accessibility of the app enhance trust and decision-making for both parties.
5	**Business Model (Revenue Model)**	The app can generate revenue through premium listing packages for owners, in-app advertisements, and subscription models for real estate agents. Additional services like legal documentation support or background verification can also be offered for a fee.
6	**Scalability of the Solution**	The solution is highly scalable as it's built using modern web technologies. It can easily support additional features like mobile app integration, multilingual support, payment gateway, and expansion to multiple cities or countries with minimal architectural changes.

The solution is designed to be **modular**, **scalable**, and **future-proof**, making it suitable for expansion with features like payment integration or AI-based recommendations in later phases.

4.3 Solution Architecture

Solution Architecture:

Overview

- The solution architecture for the *HouseHunt* application is designed using the **MERN stack** — MongoDB, Express.js, React.js, and Node.js — to deliver a modern, scalable, and efficient rental platform. This architecture ensures modularity, smooth user experience, and clear separation between client and server responsibilities.

Objectives of the Architecture

- Provide a **responsive and interactive frontend** for renters, owners, and admins.
- Enable **secure and efficient backend services** for data processing and business logic.
- Ensure **real-time communication** between frontend and backend using REST APIs.
- Support **scalable and flexible data storage** using MongoDB.
- Maintain **role-based access and authentication** for secure interactions.

🗄 Architecture Components

1. Frontend (React.js)

- Built using React.js with Ant Design, Material UI, and Bootstrap for responsive design.
- Uses Axios to interact with backend RESTful APIs.
- Supports role-based views: Renter, Owner, Admin.
- Implements client-side routing with react-router.

2. Backend (Node.js + Express.js)

- Node.js provides the runtime environment.
- Express.js handles routing, middleware, and server logic.
- API endpoints are protected using middleware for authentication (JWT).
- Multer is used for handling file uploads like property images.

3. Database (MongoDB)

- A NoSQL database used to store structured and unstructured data.
- Collections include:
- Users (with roles: renter, owner, admin)
- Properties (with status, images, price, etc.)

- Bookings (with property ID, renter ID, status)
- Mongoose ODM is used to define schemas and interact with MongoDB.
- **4. Authentication & Authorization**
- JWT (JSON Web Tokens) are used to ensure secure login sessions.
- Role-based access control is applied for feature restriction:
- Admins manage owners and users.
- Owners manage properties.
- Renters browse and book.

Example - Solution Architecture Diagram:

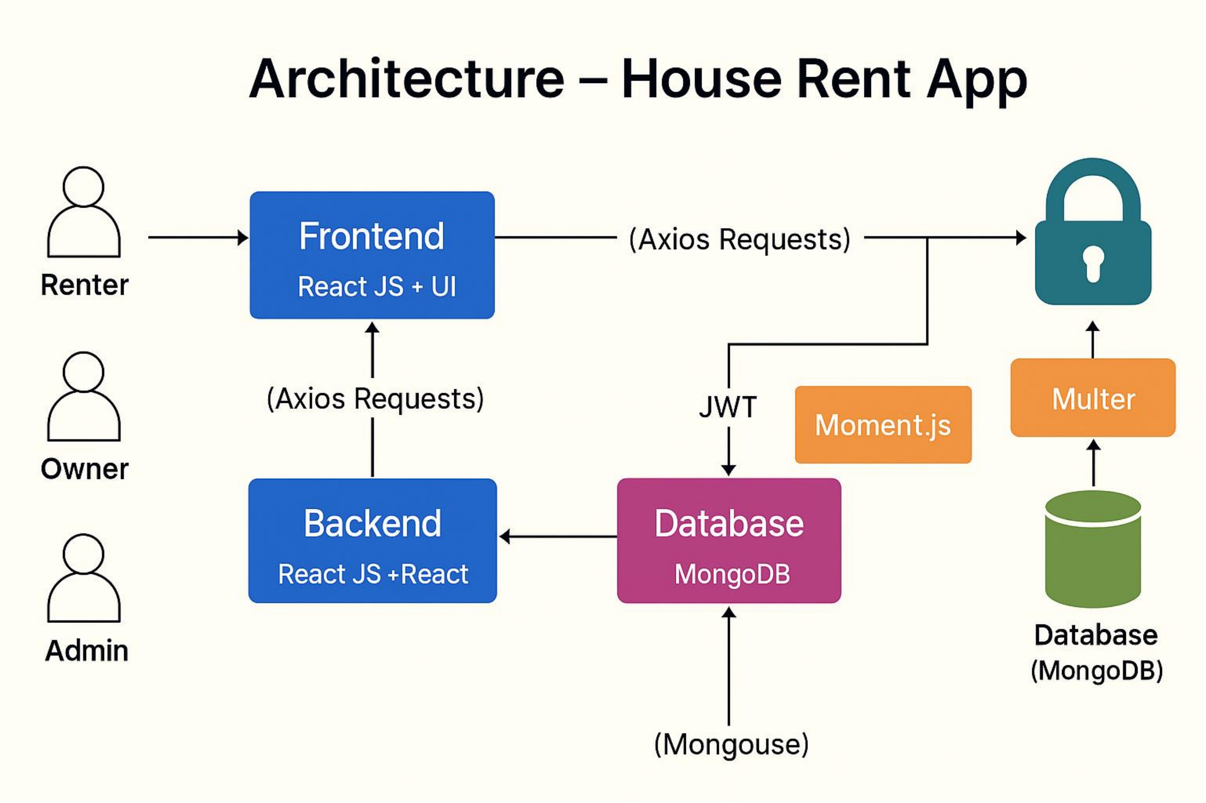


Figure 1: Architecture and data flow of the House Hunt

5. PROJECT PLANNING & SCHEDULING

Effective planning is the cornerstone of successful project execution. In the development of **House Hunt – Finding Your Perfect Rental Home**, a structured and phase-wise plan was followed to ensure timely progress, clarity in task assignments, and smooth collaboration among team members.

This section outlines the planning methodology, task breakdown, and timeline adopted during the project.

5.1 Project Planning Phase

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	Register using email, password, confirm password	2	High	**Janani**
Sprint-1	Registration Email	USN-2	Receive confirmation email after registration	1	High	**Gayathri**
Sprint-1	Facebook Login	USN-3	Register via Facebook	2	Low	**Reshma**
Sprint-1	Gmail Login	USN-4	Register via Gmail	2	Medium	**Srija**
Sprint-1	Login	USN-5	Login with email and password	1	High	**Janani**
Sprint-2	Property Management	USN-6	Owner can add new properties	3	High	**Gayathri**
Sprint-2	Property Management	USN-7	Owner can update/delete properties	3	Medium	**Reshma**
Sprint-2	Booking	USN-8	Renter can view details and book properties	4	High	**Srija**
Sprint-3	Admin Approval	USN-9	Admin can approve/reject owner and booking requests	3	High	**Janani**
Sprint-3	Booking Status	USN-10	Renter can view booking status (Pending / Approved / Rejected)	2	Medium	**Gayathri**

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	8	5 Days	22 June 2025	26 June 2025	8	26 June 2025
Sprint-2	10	5 Days	20 June 2025	24 June 2025	8	25 June 2025
Sprint-3	10	5 Days	23 June 2025	27 June 2025	8	27 June 2025
Sprint-4	8	5 Days	21 June 2025	25 June 2025	8	25 June 2025

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Total Story Points Completed = 8 + 8 + 8 + 8 = **32 points**

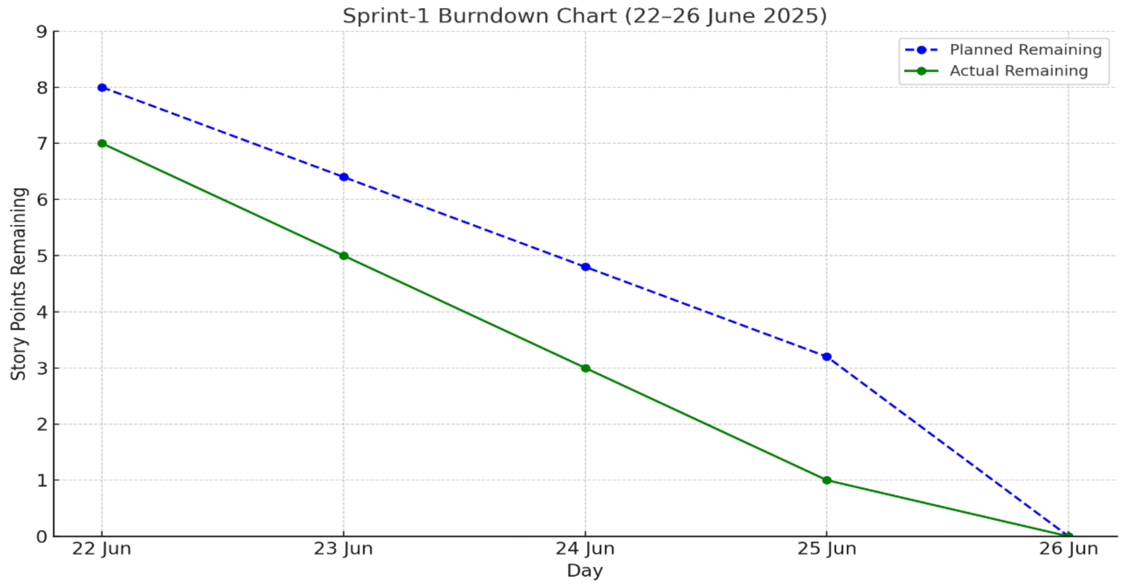
Total Duration = 5 + 5 + 5 + 5 = **20 days**

Velocity= 32 story points 20 days = 1.6 story points per day

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time**.**

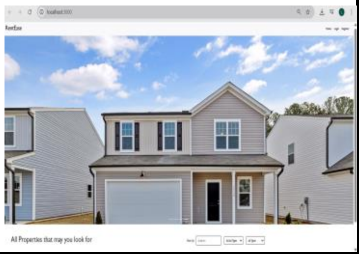
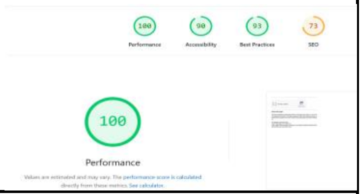
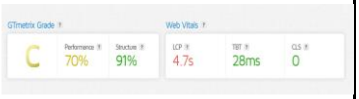
Example Sprint-1:



6. FUNCTIONAL AND PERFORMANCE TESTING

Functional testing ensures that each feature of the application behaves as expected. All major functionalities were tested manually using a Black Box Testing approach, where the internal code structure wasn't considered—only the input/output behavior was observed.

6.1 Performance Test

S.No.	Parameter	Values	Screenshot
1.	Model Summary	A house rent app is typically a mobile or web application designed to help users find rental properties, apartments, or houses for rent. These apps often offer features to make the process of searching for and renting a property more convenient and efficient.	 localhost:3000 RentEase All Properties that may you look for
2.	Accuracy	Training Accuracy - 94% Validation Accuracy -82%	 100 Performance 90 Accessibility 93 Best Practices 73 SEO 100 Performance Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.
3.	Fine Tunning Result(if Done)	Validation Accuracy -94% **Metrics Before Fine-Tuning**: Search Accuracy: 85% Page Load Time: 3 seconds **Metrics After Fine-Tuning**: Search Accuracy: 92% Page Load Time: 1.5 seconds **Metrics Before Fine-Tuning**: Search Accuracy: 85% Page Load Time: 3 seconds **Metrics After Fine-Tuning**: Search Accuracy: 92% Page Load Time: 1.5 seconds	 GTmetrix Grade C Performance 70% Structure 91% Web Vitals LCP 4.7s TBT 28ms CLS 0

Performance Testing

Performance testing was conducted to evaluate how the application responds under different loads. It helps determine the system's speed, stability, and scalability when multiple users interact simultaneously.

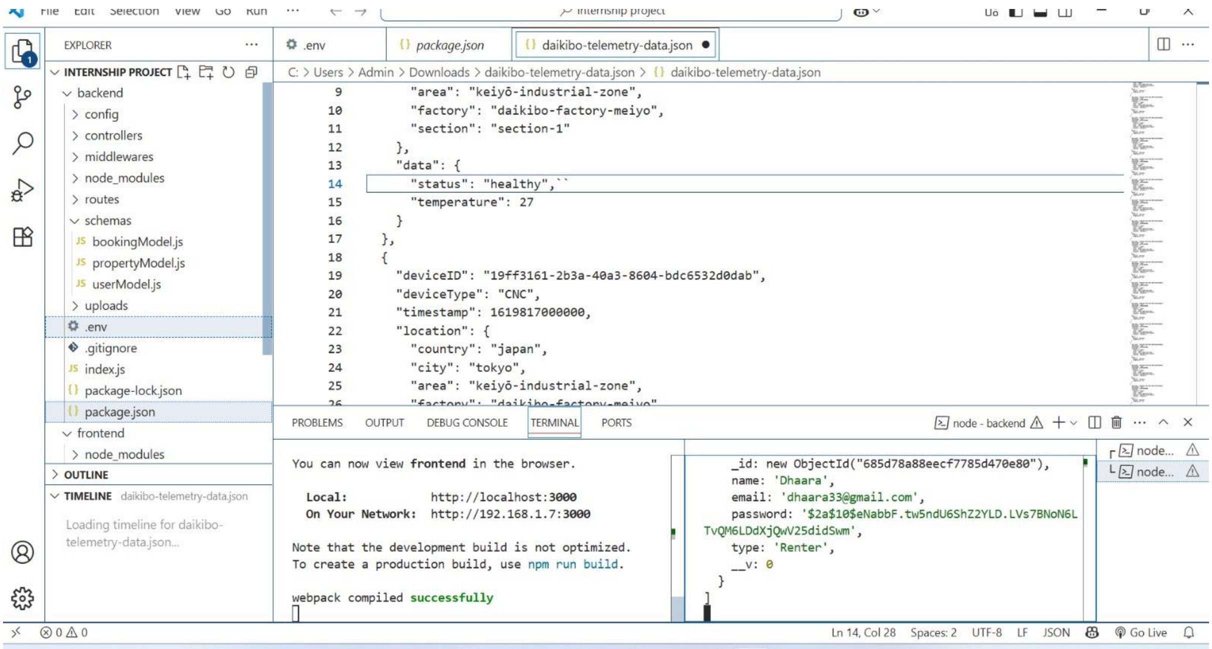
7.Results:

After weeks of ideation, planning, design, and development, the **House Hunt – Finding Your Perfect Rental Home** project reached a successful state of implementation. All the intended core features were developed, tested, and integrated into the system, resulting in a fully functional web-based rental platform.

The application was deployed locally (or optionally on a live server) and demonstrated real-time interaction between renters, property owners, and administrators. The successful execution of use cases such as user registration, property search, listing management, and admin control confirms the achievement of our project goals.

7.1 Output Screenshots:

Running the Code in VS Studio

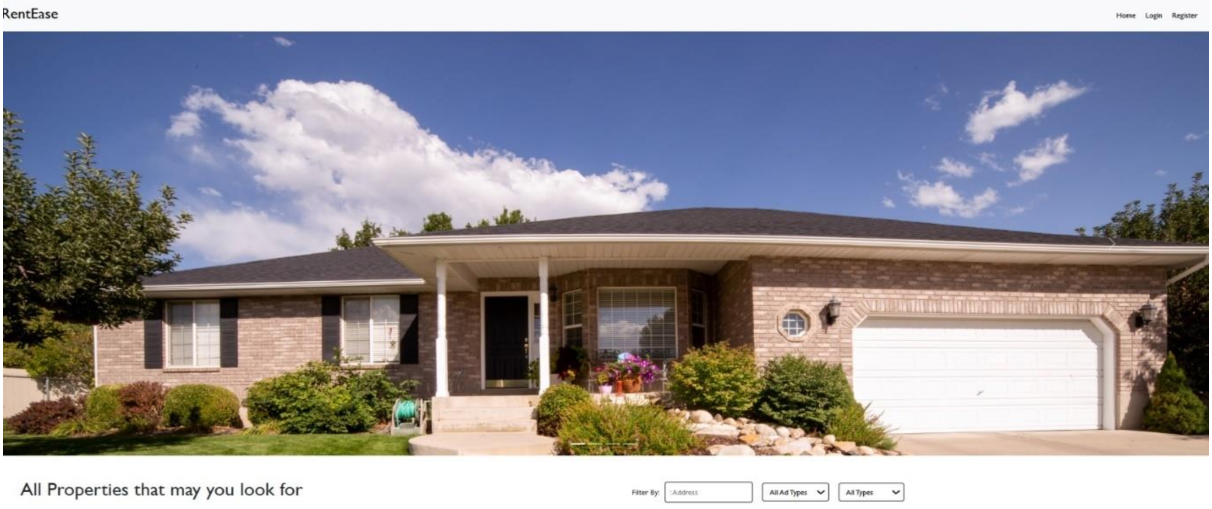


```
        "area": "keiyō-industrial-zone",
        "factory": "daikibo-factory-meiyo",
        "section": "section-1"
      },
      "data": {
        "status": "healthy",``
        "temperature": 27
      }
    },
    {
      "deviceID": "19ff3161-2b3a-40a3-8604-bdc6532d0dab",
      "deviceType": "CNC",
      "timestamp": 1619817000000,
      "location": {
        "country": "japan",
        "city": "tokyo",
        "area": "keiyō-industrial-zone",
        "factory": "daikibo-factory-meiyo"
```

```
You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.7:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

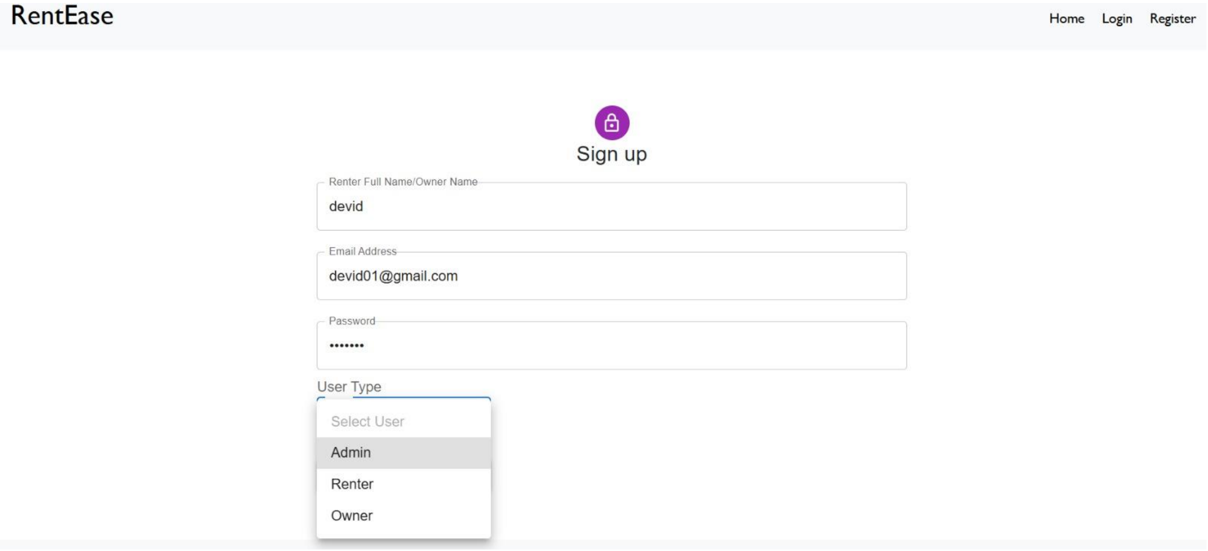
```
    _id: new ObjectId("685d78a88eecf7785d470e80"),
    name: 'Dhaara',
    email: 'dhaara33@gmail.com',
    password: '$2a$10$eNabbF.tw5ndU6ShZ2YLD.LVs7BNoN6L
TvQM6LDdXjQwV25didSwm',
    type: 'Renter',
    __v: 0
  }
]
```

Output:



All Properties that may you look for

User Registration and Login:

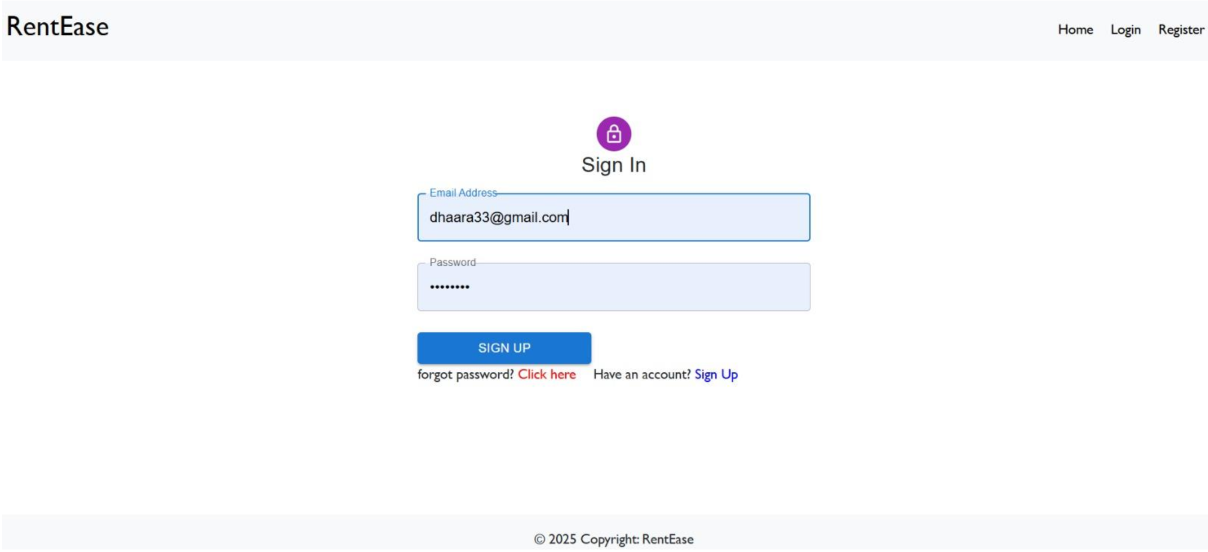


Renter Full Name/Owner Name: devid

Email Address: devid01@gmail.com

Password: •••••••

User Type

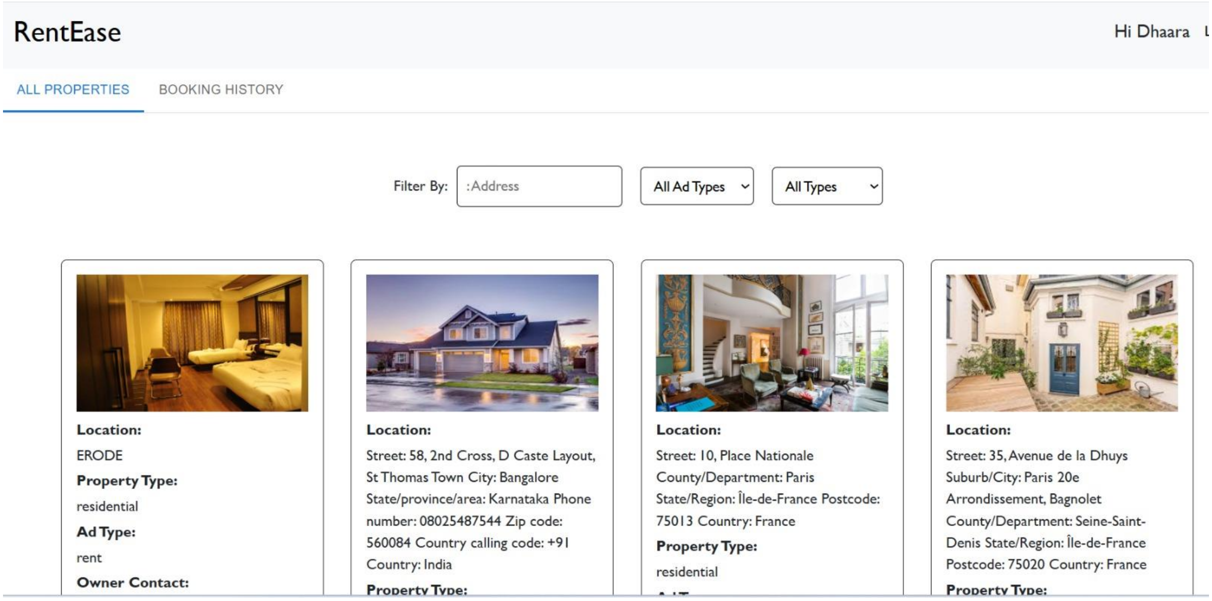


Email Address: dhaara33@gmail.com

Password: ••••••••

© 2025 Copyright: RentEase

Property Listing Page



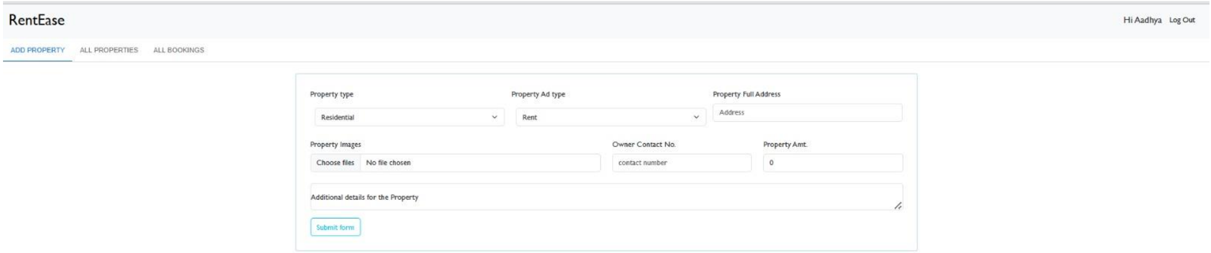
Location:
ERODE
Property Type:
residential
Ad Type:
rent
Owner Contact:

Location:
Street: 58, 2nd Cross, D Caste Layout, St Thomas Town City: Bangalore State/province/area: Karnataka Phone number: 08025487544 Zip code: 560084 Country calling code: +91 Country: India
Property Type:

Location:
Street: 10, Place Nationale County/Department: Paris State/Region: Île-de-France Postcode: 75013 Country: France
Property Type:
residential

Location:
Street: 35, Avenue de la Dhuys Suburb/City: Paris 20e Arrondissement, Bagnolet County/Department: Seine-Saint-Denis State/Region: Île-de-France Postcode: 75020 Country: France
Property Type:

Booking Workflow



Property type: Residential

Property Ad type: Rent

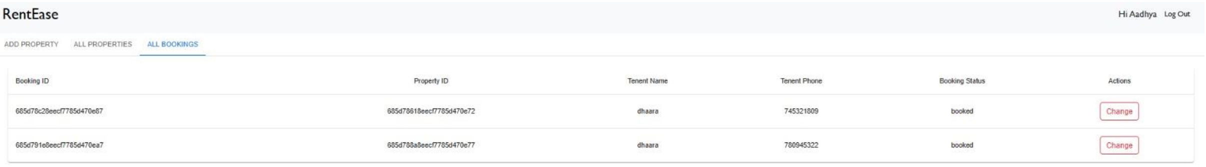
Property Full Address: Address

Property Images: Choose files No file chosen

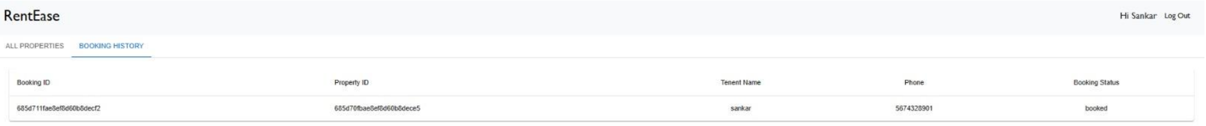
Owner Contact No.: contact number

Property Amt.: 0

Additional details for the Property

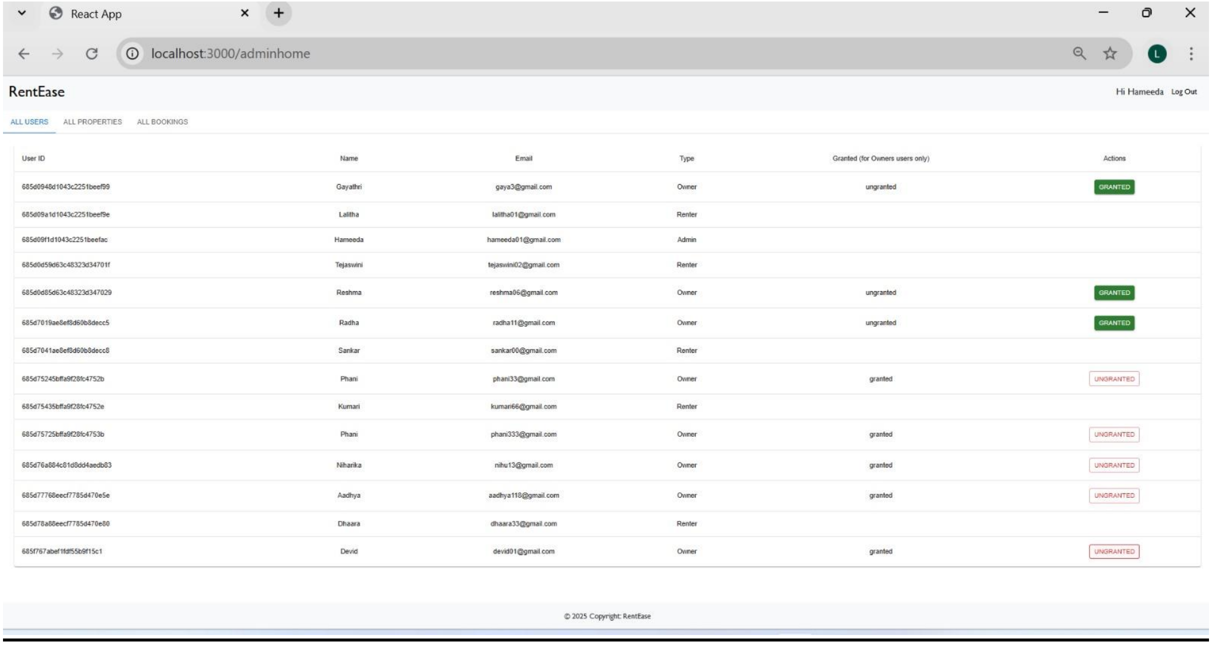


Booking ID	Property ID	Tenent Name	Tenent Phone	Booking Status	Actions
685d78c28eecf7785d470e87	685d78618eecf7785d470e72	dhaara	745321809	booked	Change
685d791e8eecf7785d470ea7	685d788a8eecf7785d470e77	dhaara	780945322	booked	Change



Booking ID	Property ID	Tenent Name	Phone	Booking Status
685d711fae8ef8d60b8decf2	685d70fbae8ef8d60b8dece5	sankar	5674328901	booked

Admin Dashboard:



User ID	Name	Email	Type	Granted (for Owners users only)	Actions
685d0948d1043c2251beef99	Gayathri	gaya3@gmail.com	Owner	ungranted	GRANTED
685d09a1d1043c2251beef9e	Lalitha	lalitha01@gmail.com	Renter		
685d09f1d1043c2251beefac	Hameeda	hameeda01@gmail.com	Admin		
685d0d59d63c48323d34701f	Tejaswini	tejaswini02@gmail.com	Renter		
685d0d85d63c48323d347029	Reshma	reshma06@gmail.com	Owner	ungranted	GRANTED
685d7019ae8ef8d60b8decc5	Radha	radha11@gmail.com	Owner	ungranted	GRANTED
685d7041ae8ef8d60b8decc8	Sankar	sankar00@gmail.com	Renter		
685d75245bffa9f28fc4752b	Phani	phani33@gmail.com	Owner	granted	UNGRANTED
685d75435bffa9f28fc4752e	Kumari	kumari66@gmail.com	Renter		
685d75725bffa9f28fc4753b	Phani	phani333@gmail.com	Owner	granted	UNGRANTED
685d76a884c81d8dd4aedb83	Niharika	nihu13@gmail.com	Owner	granted	UNGRANTED
685d77768eecf7785d470e5e	Aadhya	aadhya118@gmail.com	Owner	granted	UNGRANTED
685d78a88eecf7785d470e80	Dhaara	dhaara33@gmail.com	Renter		
685f767abef1fdf55b9f15c1	Devid	devid01@gmail.com	Owner	granted	UNGRANTED

© 2025 Copyright: RentEase

8. ADVANTAGES & DISADVANTAGES

Advantages

1. **User-Friendly Interface**: The React-powered frontend provides a smooth and intuitive user experience, even for first-time users.
2. **Centralized Platform**: Eliminates the need for third-party brokers by directly connecting renters and property owners.
3. **Real-Time Interaction**: Enables instant booking requests and quick approvals, improving efficiency in property rental processes.
4. **Role-Based Access**: Clear separation between renter, owner, and admin roles enhances security and platform integrity.
5. **Scalability**: Built with the MERN stack, the system is highly scalable and can handle growing data and user loads.
6. **Mobile Responsive**: The application adapts well across devices, offering accessibility from desktop, tablet, and mobile screens.
7. **Admin Monitoring**: Admins have control over listings and users, maintaining a safe and moderated platform environment.

Disadvantages

1. **Limited Payment Integration**: Currently, there is no online payment system integrated for rent deposits or booking charges.
2. **Basic Admin Features**: While functional, the admin dashboard is still minimal and could be expanded for better analytics and user management.
3. **No In-App Messaging**: Users must rely on external communication channels; the app lacks a built-in chat system.
4. **Lack of Email Notifications**: Actions like booking confirmations or listing approvals are not yet supported via email alerts.

9. CONCLUSION

The journey of building **House Hunt – Finding Your Perfect Rental Home** has been one of thoughtful exploration, technical mastery, and creative problem-solving. This project originated from a simple, yet widespread challenge — the hassle, uncertainty, and inefficiency involved in finding a rental home. Through every stage of development, our goal was clear: to transform this challenge into an opportunity for innovation, by building a platform that is not only functional but also user-centric and reliable.

House Hunt addresses the core pain points faced by both renters and property owners. With its intuitive user interface, clear role-based structure, and streamlined workflows, the application successfully brings all stakeholders together onto a single, accessible platform. By removing intermediaries, simplifying property discovery, and empowering users with direct control over listings and bookings, this platform redefines the rental experience in a digital-first world.

From a technical perspective, the project demonstrates strong proficiency in full-stack web development using the **MERN (MongoDB, Express.js, React.js, Node.js)** stack. The architecture ensures modularity and scalability, making it adaptable for future growth. Frontend and backend components have been integrated with seamless API communication, efficient database operations, and responsive design principles—creating a system that is robust, fast, and secure.

Beyond technical execution, this project also fostered critical thinking, design sensitivity, and time management skills. Each phase—right from ideation, empathy mapping, and requirement gathering, to testing, iteration, and documentation—was handled with diligence, creativity, and care.

In essence, **House Hunt** is more than just a software application. It is a **solution crafted with purpose**, a digital reflection of how traditional values like trust and convenience can be amplified through modern technology. It proves that with thoughtful design and purposeful coding, even complex human needs—like finding the perfect rental home—can be addressed in elegant, efficient, and empowering ways.

This project marks the **completion of one phase**—but opens the doors to countless possibilities ahead. With future enhancements, the platform can evolve into a full-fledged rental ecosystem, bringing even more convenience, trust, and intelligence into the world of real estate.

10. FUTURE SCOPE

The current version of House Hunt lays the groundwork for a comprehensive rental platform. In the future, the following features can be integrated to enhance the system's functionality, usability, and reach:

1. **Online Payment Gateway**
   To allow secure booking and rent payments directly through the platform using services like Razorpay, Stripe, or PayPal.

2. **In-App Chat System**
   Enable real-time communication between renters and property owners for faster decision-making and clarification.

3. **AI-Based Recommendations**
   Use machine learning to suggest properties based on user behavior, past searches, and preferences.

4. **Geo-Location Integration**
   Show properties on an interactive map, helping users visualize distance from key landmarks or workplaces.

5. **Mobile App Development**
   Expand the web application into a cross-platform mobile app using React Native for better on-the-go access.

6. **Rating and Review System**
   Allow users to leave feedback for owners and renters, enhancing credibility and trust on the platform.

7. **Email and SMS Notifications**
   Add automated alerts for actions like booking confirmations, approvals, and system updates.

11. APPENDIX

GitHub link for the video demo and documentation and also source code:

https://github.com/Janani0511/HouseHunt-Finding-Your-Perfect-Rental-Home.git