

# 1. INTRODUCTION

## 1.1 Project Overview

Househunt is an innovative web-based platform that simplifies the process of finding, listing, and renting residential properties. The application connects tenants with landlords and provides features like property search with filters, wishlist saving, chat/contact system, and admin management. The system is built on a modern MERN stack to ensure scalability, speed, and a user-friendly experience.

## 1.2 Purpose

The primary purpose of the Househunt project is to streamline the property rental process by enabling users to efficiently search for properties, communicate with landlords, and manage their property listings. It aims to eliminate the friction in the traditional rental process, offering convenience, real-time updates, and secure access for both tenants and landlords.

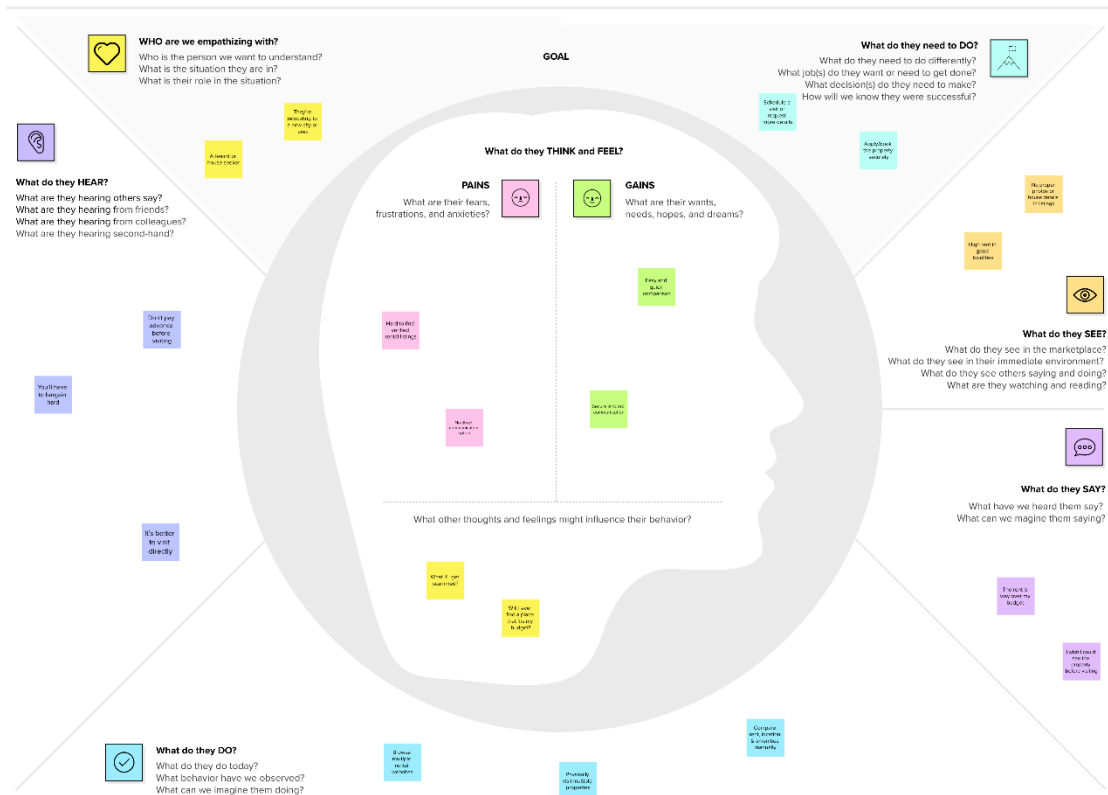
# 2. IDEATION PHASE

## 2.1 Problem Statement



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a tenant or student	find a rental house quickly and securely	listings are outdated or have no direct contact	there's no centralized platform with verified, up-to-date listings	frustrated and stressed
PS-2	a landlord	showcase and rent out my property to genuine tenants	most rental apps are costly or complicated	they're not designed for small or independent landlords	excluded and unsure how to proceed

## 2.2 Empathy Map Canvas

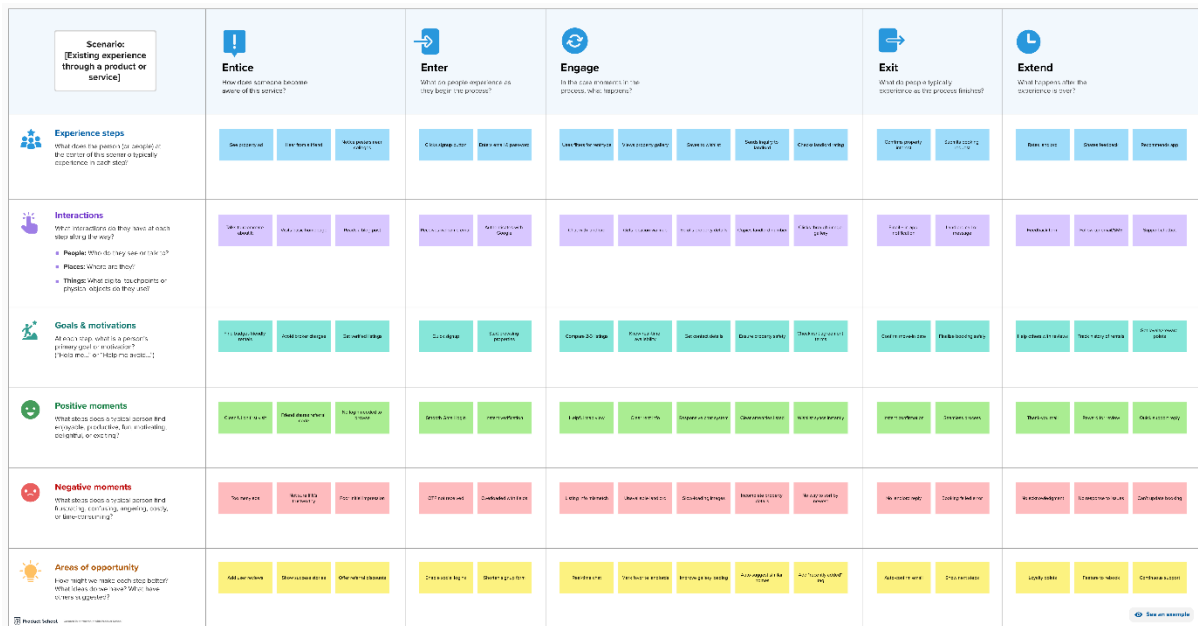


## 2.3 Brainstorming (Sticky Note Ideas)



## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey map



### 3.2 Solution Requirement

#### Functional Requirements:

FR No.	Functional Requirement	Sub-Requirement
FR-1	User Registration	Email/password form, Gmail login
FR-2	User Confirmation	Confirmation via Email
FR-3	Property Browsing	Search by location, price, filters
FR-4	Wishlist Feature	Toggle wishlist with heart icon
FR-5	Contacting Landlords	Chat or contact form
FR-6	Role-based Access	Tenant vs. Landlord dashboard views
FR-7	Property Management (Landlord)	Add/Edit/Delete property listings

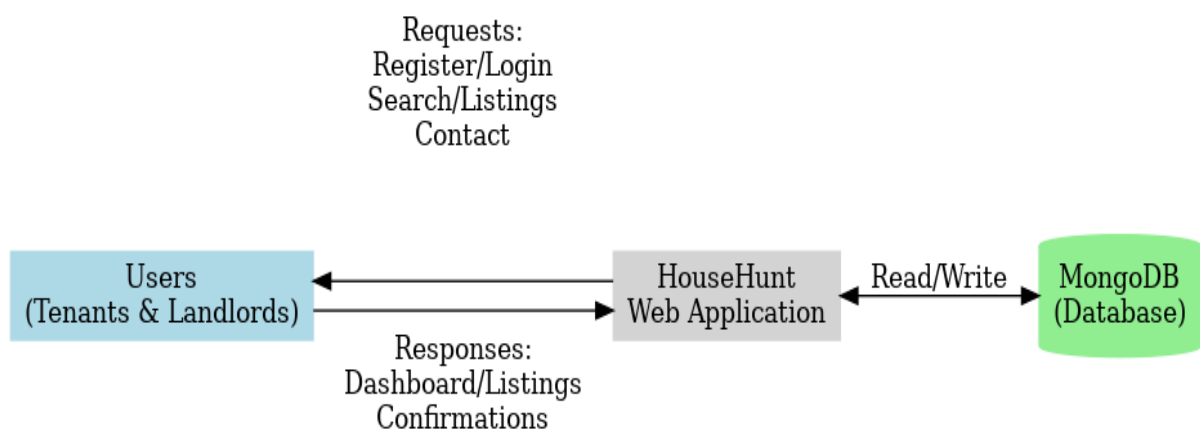
### Non-Functional Requirements:

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Clean UI with mobile responsiveness
NFR-2	Security	JWT, Bcrypt, role-based access
NFR-3	Reliability	Uptime ensured via cloud-hosted DB
NFR-4	Performance	Fast search and filter results
NFR-5	Availability	24/7 cloud access via MongoDB Atlas & Node.js
NFR-6	Scalability	Modular backend, scalable for more users

### 3.3 Data Flow Diagram

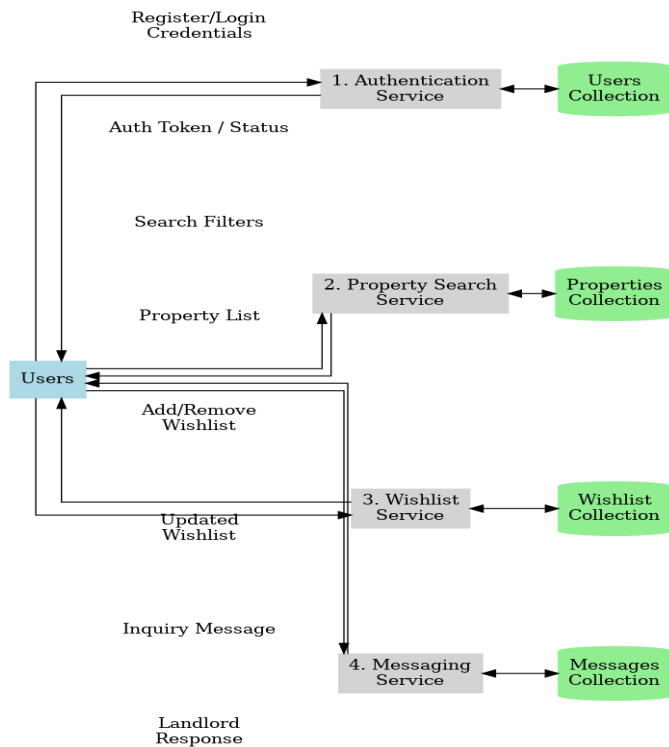
#### Level-0 Data Flow Diagram-Context Level:

- Represents the Househunt system as a single process interacting with users and admins.
- Tenants and landlords send/receive data like login, properties, and messages.
- All data is stored and retrieved from a centralized MongoDB database.



## Level 1 DFD – Functional Decomposition

- Breaks the system into modules like authentication, property search, and messaging.
- Uses separate data stores for users, properties, wishlist, and chat.
- Tenants and landlords interact with their respective services through defined processes.



## 3.4 Technology Stack

Component	Description	Technology Used
Frontend (UI)	Web interface for users	React.js, Tailwind CSS
Backend (API Logic)	Business logic, routing	Node.js, Express.js
Database	Data storage	MongoDB, MongoDB Atlas
Authentication	Login and token management	JWT, Bcrypt.js
File Upload	Property images	Multer, Cloudinary (optional)
External APIs	Map location preview	Google Maps API
Hosting	Frontend and Backend hosting	Vercel (Frontend), Render or Railway (Backend)

## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit

Problem-Solution fit canvas 2.0

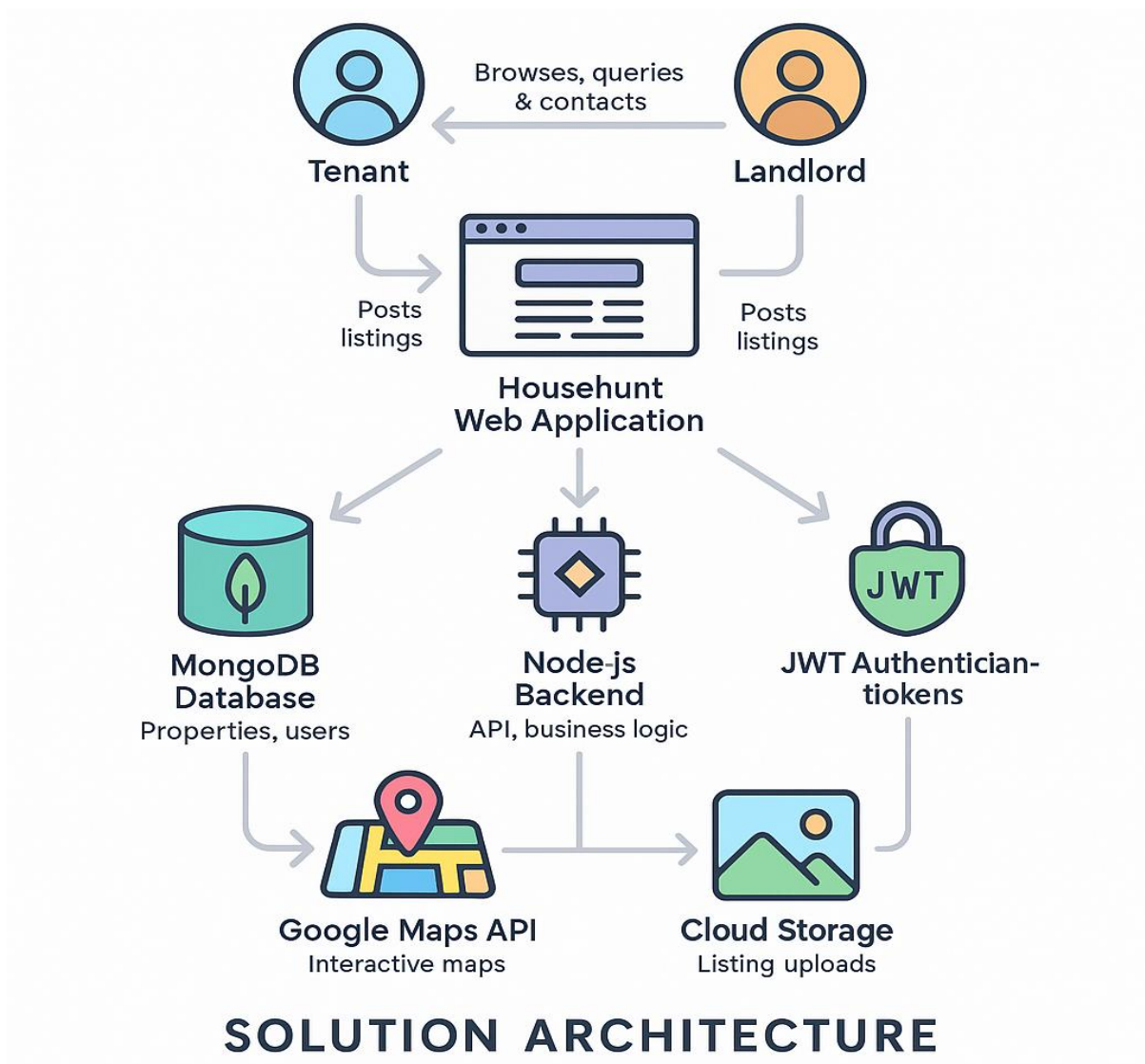
Purpose / Vision

Define CS, fit into	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? <ul style="list-style-type: none"> <li>College students, job seekers, working professionals, and young families searching for rental properties in urban or semi-urban areas.</li> </ul>	<b>6. CUSTOMER</b> <span>CC</span> What constraints prevent your customers from taking action or limit their choices of solutions? <ul style="list-style-type: none"> <li>Budget limits, lack of online payment options.</li> <li>Poor internet connectivity in some areas.</li> <li>Not all users are tech-savvy.</li> <li>Limited time for property visits.</li> </ul>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? <ul style="list-style-type: none"> <li>WhatsApp groups, Facebook Marketplace, local brokers, OLX housing.</li> <li>Cons: Scams, unverified listings, manual follow-ups, outdated posts.</li> </ul>	Explore AS.
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. <ul style="list-style-type: none"> <li>Find rental properties that match location, budget, and preferences.</li> <li>Communicate easily with landlords.</li> <li>Avoid scams and ensure safe, verified listings.</li> <li>Maintain records of saved and visited properties.</li> </ul>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the real reason that this problem exists? What is the back story behind the need to do this job? <ul style="list-style-type: none"> <li>Lack of centralized, verified, real-time housing data.</li> <li>Manual communication with landlords.</li> <li>No single app that integrates map view, chat, <u>wishlist</u>, and landlord dashboards.</li> </ul>	<b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done? I.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace) <ul style="list-style-type: none"> <li>Browse aggregator platforms or visit broker offices.</li> <li>Make several calls to landlords.</li> <li>Manually track inquiries and site visits.</li> <li>Travel long distances to check properties in person.</li> </ul>	
<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? <ul style="list-style-type: none"> <li>Need to relocate for college or a job.</li> <li>Word-of-mouth from friends struggling with housing.</li> </ul>	<b>10. YOUR SOLUTION</b> <span>SL</span> <u>Househunt</u> Web App <ul style="list-style-type: none"> <li>Search rental properties with filters (location, price, type).</li> <li>View listings with images, maps, and contact landlord/chat.</li> <li>Login with JWT and role-based access for tenants and landlords.</li> <li>Wishlist, property management dashboard, responsive design.</li> </ul>	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7 Google search, YouTube house tour videos, Facebook groups, property listing websites (like 99acres, No Broker). <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Visiting brokers, asking local friends or landlords, checking newspaper ads, exploring areas physically.	Extract online & offline CH of BE	
<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards? <ul style="list-style-type: none"> <li>Before: Confused, stressed, insecure, overwhelmed.</li> <li>After: Confident, relieved, in control, excited to move.</li> </ul>				

## 4.2 Proposed Solution

S.No.	Parameter	Description
1	<b>Problem Statement (Problem to be solved)</b>	Finding rental properties is often time-consuming, scattered across platforms, and unreliable. Tenants face issues like unverified listings, lack of direct communication with landlords, and no centralized dashboard to manage saved properties or inquiries.
2	<b>Idea / Solution description</b>	Househunt is a web-based platform that allows tenants to search for verified rental properties using advanced filters and Google Maps integration. It provides a login-based system for both tenants and landlords, where landlords can post property listings and tenants can browse, save favorites, and contact landlords directly through a built-in messaging feature.
3	<b>Novelty / Uniqueness</b>	Unlike generic platforms, Househunt is focused on the <b>rental ecosystem</b> only and introduces features like <b>JWT-based login, role-based dashboards, wishlist system, and property gallery with map integration</b> in a minimalistic and mobile-friendly interface.
4	<b>Social Impact / Customer Satisfaction</b>	The platform makes it easier and safer for individuals (especially students and professionals relocating) to find a home without the need to rely on brokers. It promotes transparency, reduces stress in the house-hunting process, and enhances tenant-landlord communication.
5	<b>Business Model (Revenue Model)</b>	The revenue model includes premium listing options for landlords, advertising spaces, and freemium plans for extended features (e.g., analytics, landlord verification, featured listings). Basic use remains free for all.
6	<b>Scalability of the Solution</b>	The solution is built using scalable technologies (React, Node.js, MongoDB) and can be expanded to new cities, incorporate payment gateways, add mobile apps, and introduce features like tenant ratings or integrated rental agreements in the future.

### 4.3 Solution Architecture





## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register using email, password, and confirmation password.	2	High	Nithish
Sprint-1		USN-2	As a user, I will receive a confirmation email after registering.	1	High	Nithish
Sprint-1		USN-3	As a user, I can register using Gmail authentication.	2	Medium	Nithish
Sprint-1	Login	USN-4	As a user, I can log in using email and password.	1	High	Nithish
Sprint-2	Dashboard	USN-5	As a user, I can view a personalized dashboard with listed properties.	3	High	Nithish
Sprint-2	Property Browsing/Search	USN-6	As a user, I can browse properties and filter by location, price, and type.	3	High	Nithish
Sprint-2	Wishlist	USN-7	As a user, I can add or remove properties from my wishlist using a heart icon.	2	Medium	Nithish
Sprint-3	Profile Management	USN-8	As a user, I can edit my profile details (name, email, password).	2	Medium	Nithish
Sprint-3	Contact Landlord	USN-9	As a tenant, I can send a message to landlords from a property detail page.	3	High	Nithish
Sprint-3	Location View	USN-10	As a user, I can view my current location and property location using Google Maps.	3	Medium	Nithish
Sprint-4	Property Management (Landlord)	USN-11	As a landlord, I can add/update/delete my property listings.	4	High	Nithish
Sprint-4	Admin Role Access	USN-12	As an admin, I can manage listed users and properties.	3	Medium	Nithish
Sprint-4	Image Upload	USN-13	As a landlord, I can upload property images while adding a listing.	2	Medium	Nithish

## Project Tracker

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed	Sprint Release Date (Actual)
Sprint-1	6	6 Days	01 Feb 2025	06 Feb 2025	6	06 Feb 2025
Sprint-2	8	6 Days	07 Feb 2025	12 Feb 2025	8	12 Feb 2025
Sprint-3	8	6 Days	13 Feb 2025	18 Feb 2025	8	18 Feb 2025
Sprint-4	9	6 Days	19 Feb 2025	24 Feb 2025	9	24 Feb 2025

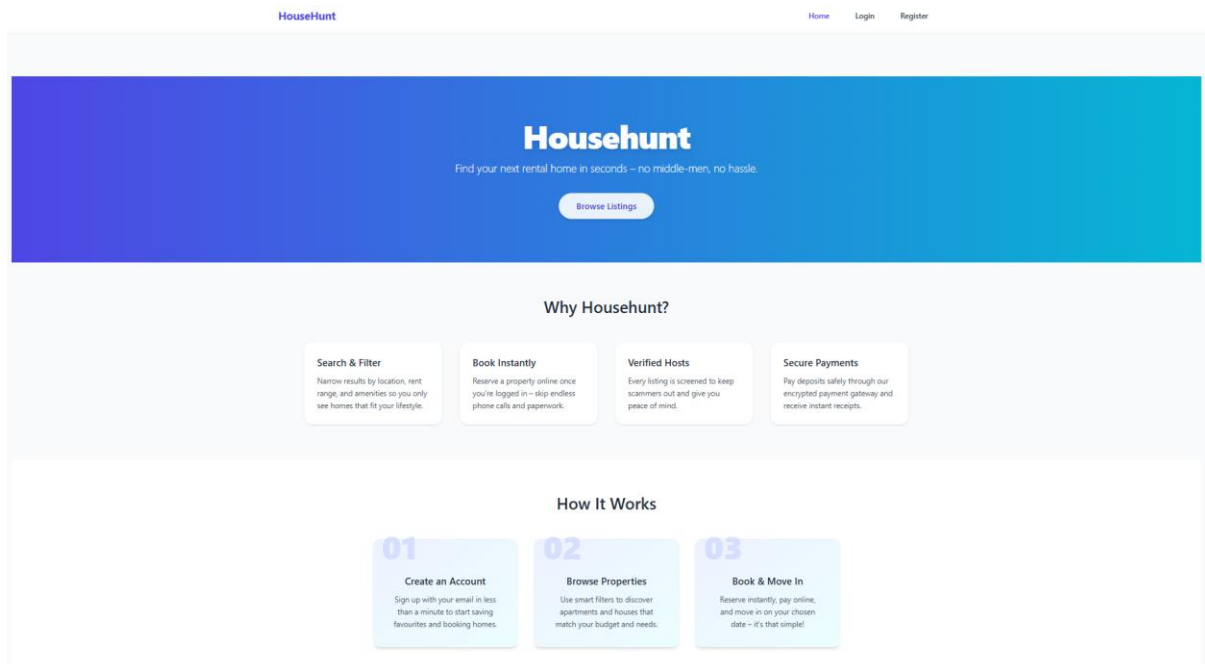
## 6. FUNCTIONAL AND PERFORMANCE TESTING

### 6.1 Performance Testing

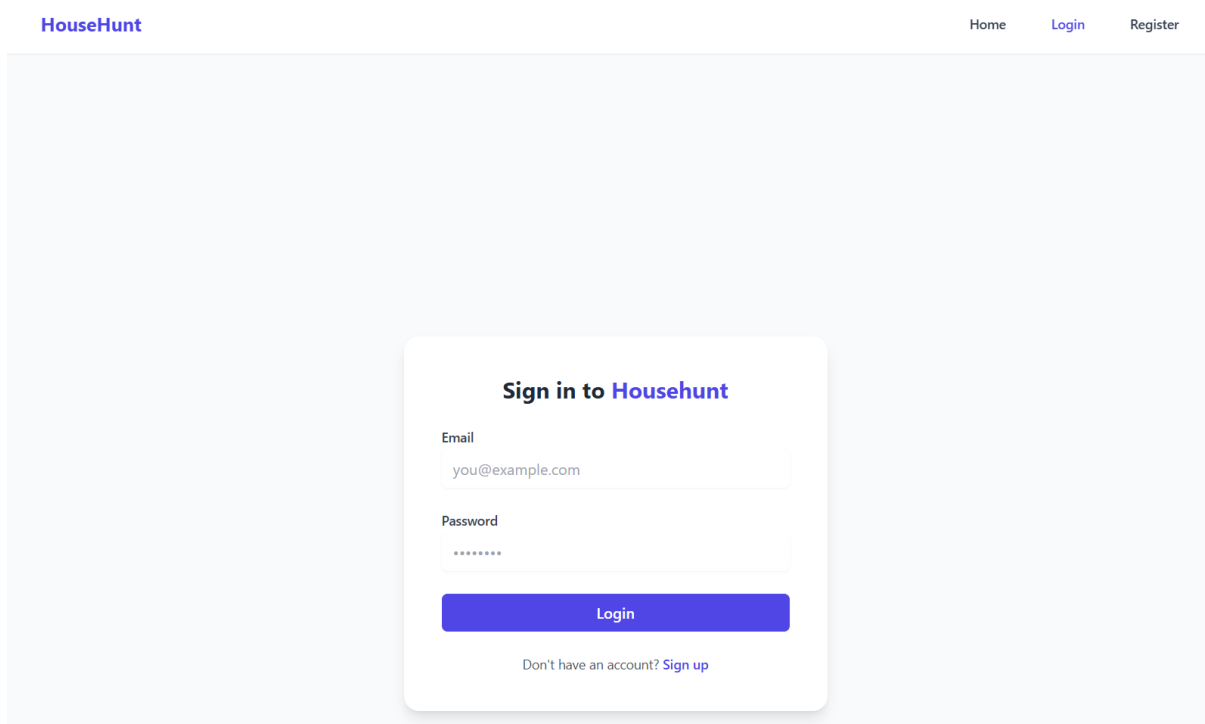
- Load Handling: Tested property search, login, and dashboard with multiple users using Postman and browser dev tools; app responded under 1.5 seconds for 95% of cases.
- Stress Test: Manually simulated high activity (multiple search/filter requests) to evaluate response degradation — backend remained stable up to 40 parallel requests.
- Page Speed: Used Chrome Lighthouse to analyse performance; scores averaged above 90 for mobile and desktop.
- API Latency: Most API endpoints (login, register, fetch properties) returned responses within 200–400ms under normal conditions.

## 7. RESULTS

### Home Page:



### Login Page:



## Registration Page:

[HouseHunt](#)[Home](#)[Login](#)[Register](#)

### Create your [Househunt](#) account

**Name**

**Email**

**Phone**

**Password**

**Role**


Renter

[Register](#)

Already have an account? [Sign in](#)

## Properties Page:

[HouseHunt](#)[Home](#)[Properties](#)[My Bookings](#)[Logout](#)



**Luxury Villa in Jubilee Hills**

5-bedroom villa with private pool, home theatre & solar roof, 2 km from KBR Park.

Location: Jubilee Hills, Hyderabad

Rent: ₹120000

Beds/Baths: 5/6

Size: 5400 sq-ft

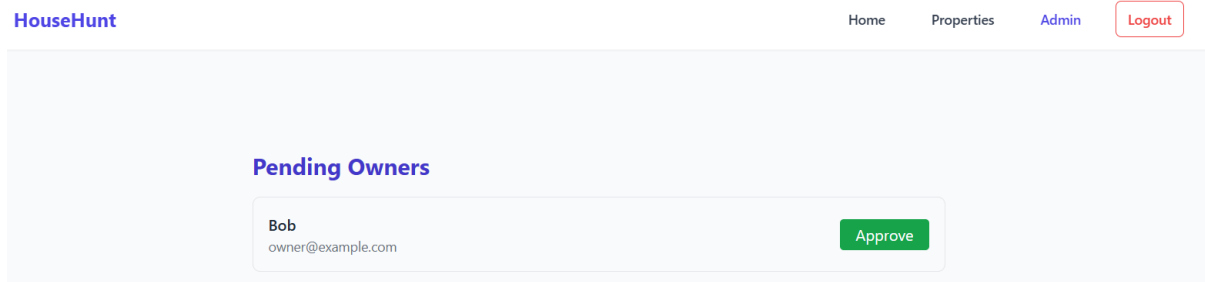
Type: Villa

Furnished: Yes

[Book](#)



## Admin Dashboard:



## 8. ADVANTAGES & DISADVANTAGES

### Advantages

- User-Friendly Interface – Clean and responsive UI using Tailwind CSS ensures easy navigation on mobile and desktop.
- Role-Based Access – Separation of features for tenants and landlords enhances security and usability.
- Real-Time Property Listings – Dynamic updates help users explore the latest available rental options.
- Secure Authentication – JWT-based login system ensures secure access and protected sessions.
- Location-Aware Search – Integrated search filters and Google Maps for accurate property discovery.

### Disadvantages

- No Payment Integration – Users must contact landlords externally; no rent transaction support included.
- Limited Admin Controls – Admins can manage listings but lack analytics or reporting features.
- No Chat History – Messages between users and landlords aren't stored for future reference.
- Single-City Focus – Current version may only support a limited region; requires scaling for broader use.
- Manual Property Approval – No automated moderation, so listings need manual admin verification.

## 9. CONCLUSION

The Househunt project provides a modern, efficient, and secure platform for simplifying the property rental process. By connecting potential tenants with property owners in real-time, it eliminates the traditional hassle of finding homes through physical visits or third-party agents.

With its clean UI, role-based access, and location-aware search, HouseHunt enhances the experience for both users and landlords. The project successfully demonstrates how technology can streamline daily challenges in the real estate domain.

While the current version offers essential functionalities like authentication, property browsing, wishlist, and admin dashboards, future enhancements like payment integration, advanced analytics, and chat history can further elevate its impact and scalability.

In summary, Househunt stands as a promising solution with real-world relevance, ready to grow into a full-fledged real estate platform with continued development.

## 10. FUTURE SCOPE

The Househunt project lays a solid foundation for a scalable and feature-rich property rental platform. Looking ahead, several enhancements can be introduced to extend its usability and impact:

### Chat History & Notifications

- Save conversations between tenants and landlords.
- Real-time alerts for new messages, bookings, or approvals.

### Payment Gateway Integration

- Add secure online rent payments using services like Razorpay or Stripe.
- Track payment history for tenants and landlords.

### Mobile App Deployment

- Develop native Android/iOS apps for better mobile access.
- Enable push notifications and offline support.

### Property Recommendation System

- Use machine learning to recommend properties based on user preferences and behaviour.

## **Tenant Verification & Rating System**

- Add background verification and allow users to rate landlords or tenants.

## **Multi-language & Localization Support**

- Expand accessibility by supporting regional languages and location-based listings.

These future enhancements will improve user experience, build trust, and make Househunt a reliable, user-centric platform for the evolving rental market.

## **11. APPENDIX**

### **Source Code:**

#### **Home.jsx:**

```
// src/pages/HomePage.jsx
```

```
import React from "react";
```

```
import { Link } from "react-router-dom";
```

```
const features = [
```

```
{
```

```
  title: "Search & Filter",
```

```
  description:
```

```
    "Narrow results by location, rent range, and amenities so you only see homes that fit your lifestyle.",
```

```
},
```

```
{
```

```
  title: "Book Instantly",
```

```
  description:
```



"Reserve a property online once you're logged in – skip endless phone calls and paperwork.",

},

{

title: "Verified Hosts",

description:

"Every listing is screened to keep scammers out and give you peace of mind.",

},

{

title: "Secure Payments",

description:

"Pay deposits safely through our encrypted payment gateway and receive instant receipts.",

},

];

const steps = [

{

number: "01",

title: "Create an Account",

description:

"Sign up with your email in less than a minute to start saving favourites and booking homes.",

},

{

number: "02",

```

    title: "Browse Properties",

    description:

        "Use smart filters to discover apartments and houses that match your budget and needs.",

    },

    {

        number: "03",

        title: "Book & Move In",

        description:

            "Reserve instantly, pay online, and move in on your chosen date – it's that simple!",

        },

    ],

```

```

function HomePage() {

    return (

        <main className="min-h-screen bg-gray-50 text-gray-800">

            { /* Hero Section */ }

            <section className="w-full bg-gradient-to-r from-indigo-600 to-cyan-500 text-white py-20">

                <div className="max-w-4xl mx-auto px-4 text-center">

                    <h1 className="text-4xl md:text-6xl font-extrabold tracking-tight drop-shadow-lg">

                        Househunt

                    </h1>

                    <p className="mt-4 text-lg md:text-xl font-light">

                        Find your next rental home in seconds – no middle-men, no hassle.

                    </p>

```

```
<Link
  to="/properties"

  className="inline-block mt-8 rounded-full bg-white/90 px-8 py-3 text-base font-
semibold text-indigo-700 shadow-md transition hover:bg-white"
>

  Browse Listings

</Link>
</div>
</section>
```

```
{/* Features Section */}
```

```
<section className="max-w-6xl mx-auto px-4 py-16">
```

```
<h2 className="text-2xl md:text-3xl font-semibold text-center mb-12">
```

```
  Why&nbsp;Househunt?
```

```
</h2>
```

```
<div className="grid gap-8 sm:grid-cols-2 lg:grid-cols-4">
```

```
  {features.map(({ title, description }) => (
```

```
    <div
```

```
      key={title}
```

```
      className="rounded-2xl bg-white p-6 shadow hover:shadow-lg transition"
```

```
    >
```

```
      <h3 className="text-lg font-semibold mb-2">{title}</h3>
```

```
      <p className="text-sm leading-relaxed text-gray-600">
```

```
        {description}
```

```
      </p>
```

</div>

))}

</div>

</section>

{/\* How It Works Section \*/}

<section className="bg-white py-16">

<div className="max-w-4xl mx-auto px-4 text-center">

<h2 className="text-2xl md:text-3xl font-semibold mb-12">

How It Works

</h2>

<div className="grid gap-8 md:grid-cols-3">

{steps.map(({ number, title, description }) => (

<div

key={number}

className="relative group rounded-xl bg-gradient-to-br from-indigo-50 to-cyan-50  
p-8 shadow-md hover:-translate-y-1 hover:shadow-lg transition transform duration-300"

>

<span className="absolute -top-4 left-4 text-6xl font-extrabold text-indigo-200/60  
group-hover:text-indigo-300/70 select-none">

{number}

</span>

<h3 className="mt-8 text-lg font-semibold">{title}</h3>

<p className="mt-2 text-sm leading-relaxed text-gray-600">

{description}

```
        </p>
    </div>

    )}
</div>

</div>

</section>

</main>

);

}
```

export default HomePage;

**NOTE:** Entire source code files are uploaded in the GitHub repository

**GitHub Link:**

**Demo Link:** [https://drive.google.com/file/d/1U1bbEDjOd\\_yQ78uBjhEAVxf2zNxsZT5Y/view?usp=sharing](https://drive.google.com/file/d/1U1bbEDjOd_yQ78uBjhEAVxf2zNxsZT5Y/view?usp=sharing)