

HOUSEHUNT

Team ID	LTVIP2025TMID50829
Team Size	4
Team Leader	C Sai Sumanjali
Team member	Chanagani Neelima
Team member	Choppa Pavani
Team member	Dimmagudi Yasmin
Project Name	HOUSEHUNT

HouseHunt – Brainstorm & Idea Prioritization

HouseHunt is a platform designed to streamline the rental property search and booking process. It bridges the gap between renters and verified property owners, while ensuring a secure, real-time, and admin-monitored environment.

The brainstorming phase revolved around solving pain points in current rental platforms such as unverified listings, unclear communication, and lack of centralized booking status management.

Step 1: Team Collaboration and Problem Identification

- The team analyzed current rental platforms and identified major issues such as:
- Fake or outdated listings
- Lack of booking confirmation or tracking
- No role-based verification (anyone can post)
- Absence of admin moderation or property validation

Step 2: Idea Listing and Categorization

- Initial ideas discussed included:
- Role-based registration: Renter, Owner, Admin
- Secure login using JWT authentication
- Admin verification system for owners
- Booking system with status: Pending → Approved
- Property filters: location, type, rent, rooms

- In-app messaging system (optional future scope)
- Dashboard views based on user roles

Step 3: Idea Prioritization

Ideas were prioritized based on:

- Feasibility – Can it be developed within the timeframe?
- Impact – Will it solve the user pain points?
- User-Centricity – Will it enhance the user experience?

High-Priority Features:

- Secure Login & Role-Based Access (JWT)
- Property Listing & Filtering System
- Booking Request Flow with Status Updates
- Admin Panel for Owner Approval
- Separate Dashboards for Renter/Owner/Admin

Medium-Priority Features (Future Enhancements):

- Messaging system between renter and owner
- Email/SMS notifications for booking confirmation
- Payment integration
- Image upload for property listings with gallery view

HouseHunt -Empathy Map – HouseHunt Users

What the user thinks and feels:

- “I want to find a home that matches my needs quickly and safely.”

What the user sees:

- A user-friendly interface and real-time availability of properties.

What the user says and does:

- “I found a good place. The owner responded instantly!”

What the user hears:

- Positive feedback from others using the app for rentals.

Pain Points:

- Fake property listings, poor owner communication, lack of confirmation.

Gains:

- Instant access to real listings, trusted owners, booking status, secure platform.

HOUSEHUNT - Problem Statement

Customer Problem:

Renters face challenges in identifying verified rental listings, communicating with owners, and tracking booking confirmations. Simultaneously, owners need a trusted medium to list and manage properties.

HouseHunt Solves This By:

- Secure login with role-based access (Admin/Owner/Renter)
- Real-time listing, filtering, and inquiry features
- Owner-side property CRUD features
- Booking status management and transparency
- Admin-side approval and monitoring systems

HouseHunt – Project Overview

Project Name : HouseHunt

HouseHunt is a MERN-stack-based platform designed to simplify the property rental process by connecting renters with verified property owners. The system supports user registration, property browsing, inquiry/booking handling, owner property management, and admin governance to ensure platform safety and transparency.

Project Description:

HouseHunt enables:

- Renters to browse, filter, and inquire/book homes.
- Owners to register, manage listings, and confirm bookings.
- Admins to verify users, manage listings, and govern activities.

Testing Scope

TC ID	Test Case	Steps	Expected Result	Status
TC-001	Renter Registration	Fill form, submit	Account created, dashboard loaded	Pass
TC-002	Owner Approval	Admin approves owner	Owner gets access to dashboard	Pass
TC-003	Property Booking	Renter sends request	Owner sees booking with status “pending”	Pass
TC-004	Property CRUD	Owner adds/edits/deletes property	Properties updated live	Pass
TC-005	Admin Dashboard	Lists pending owners	Approve/Reject works	Pass

HouseHunt – Project Design Phase

Problem–Solution Fit Overview

The goal of the design phase was to validate whether HouseHunt meaningfully addresses the challenges faced by renters, property owners, and platform administrators in the property rental ecosystem.

Purpose :

- Simplify and streamline the rental property discovery and booking process.
- Offer a centralized and verified platform to reduce confusion and fraud.
- Ensure transparency and trust between renters and owners.
- Enable admin governance for safer property listing and rental activity.

Identified Problems in Traditional Rental Systems

Problem	Description
Complex & unverified listings	Many listings are outdated, fake, or lack details.
No role-based control	No differentiation between normal users and verified property owners.
Lack of booking status clarity	Renters don’t get updates on the booking request status.
No admin monitoring	

HouseHunt Solutions :

Feature	How it Solves the Problem
Secure, role-based login system	Ensures only verified users access relevant features.
Owner CRUD operations for properties	Owners can manage their listings in real-time.

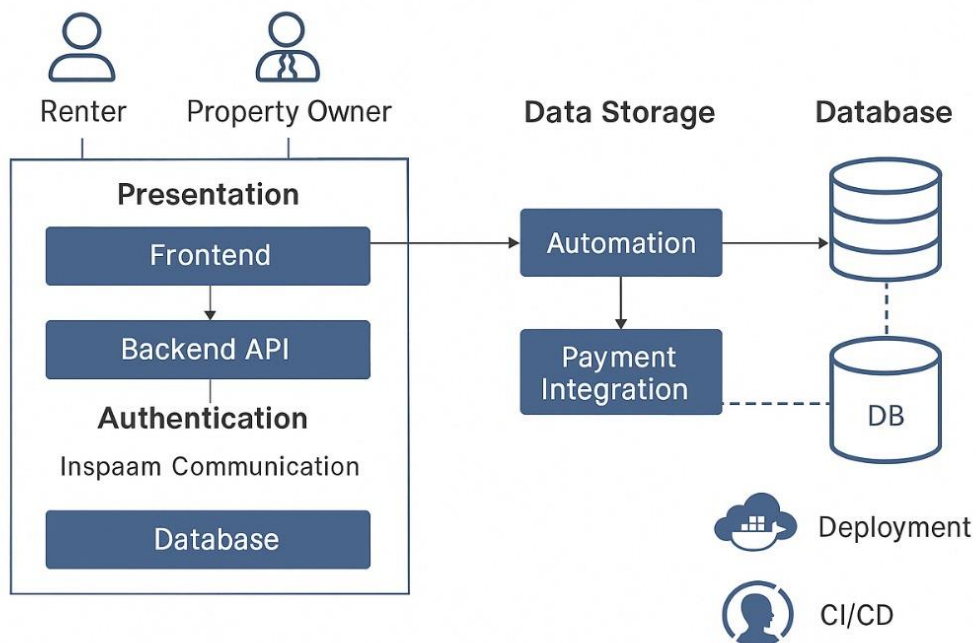
Booking system with status tracking	Renters can see booking status: pending, confirmed, rejected.
Admin approval for owners	Reduces risk by ensuring only real owners list properties
User-centric dashboards	Personalized views for renter, owner, and admin roles.

Design Objectives

- Ensure simplicity and clarity in UI for all user roles.
- Guarantee security in data handling (JWT, bcrypt).
- Make the system responsive and mobile-friendly.
- Implement scalable MongoDB schemas and REST APIs.
- Keep the platform modular for future expansion (e.g., payments, messaging, notifications).

Solution Architecture

Househunt: Seamless Appointment Bookg Properties



Project Planning Phase

FUNCTIONAL REQUIREMENTS

FR No	Functional Requirement	Sub-requirement
FR-1	User Authentication	Sign up, Login, JWT Token, Role-based Access
FR-2	Property Management	Add, View, Edit, Delete Property
FR-3	Booking System	Renter inquiry, owner confirmation, status updates
FR-4	Admin Panel	Approve owners, verify properties, manage reports
FR-5	Notifications	Booking status update (future scope: email/SMS)

User Stories

Renter

- As a renter, I can browse all properties with filters.
- As a renter, I can send booking requests.
- As a renter, I can track booking status.

Owner

- As an owner, I can register and wait for admin approval.
- As an owner, I can manage my properties.
- As an owner, I can approve/decline bookings.

Admin

- As an admin, I can approve legitimate owners.
- As an admin, I can manage platform users and ensure compliance.

Non-Functional Requirements

NFR No	Requirement	Description
NFR-1	Usability	Simple and mobile-responsive UI
NFR-2	Security	JWT auth, password hashing with bcrypt
NFR-3	Reliability	CRUD & booking APIs covered and tested
NFR-4	Performance	Page loads < 2s, real-time API updates

Technology Stack :

Component	Technology Used	Component
UI	HTML, CSS, React.js	UI
Backend Logic	Node.js, Express.js	Backend Logic
Database	MongoDB	Database
Authentication	JWT (JSON Web Tokens)	Authentication
File Handling	Multer (for property images)	File Handling
Styling	Bootstrap, Material UI (optional)	Styling

References :

[React Documentation](#)

[Nodejs](#)

[json Web server Refernce](#)