

FINAL REPORT

1. Introduction

1.1 Project Overview

The HouseHunt project is a full-stack web application developed using the MERN stack that simplifies the process of renting and managing properties. It provides a platform where property owners can list properties with images and details, while renters can browse, view, and book properties online. The system includes secure authentication, role-based dashboards, and real-time booking management. The application focuses on usability, scalability, and efficient data handling to streamline the property rental workflow.

1.2 Purpose

The purpose of the HouseHunt project is to design and develop a reliable and user-friendly rental platform that simplifies the process of finding, listing, and booking rental homes by connecting renters, property owners, and administrators through a secure and efficient digital system.

The Goals of the Project are as follows:

- To simplify the rental property search process for renters
- To reduce the time and effort required to find suitable rental homes
- To provide a secure and transparent platform for renters and property owners
- To enable property owners to efficiently manage listings and bookings
- To ensure role-based access and proper platform governance through admin control
- To improve user experience with an intuitive and responsive interface
- To maintain reliable and scalable data management using modern web technologies.

2. Ideation Phase

2.1 Problem Statement

Problem Statement 1

I am a renter looking for a suitable rental home who is trying to find suitable properties quickly, but current property search tools are slow and return many irrelevant results, because filters and ranking options are limited, which makes me feel frustrated and overwhelmed.

Problem Statement 2

I am a renter trying to compare rental properties easily, but the lack of a clear comparison feature forces me to manually switch between listings, because the platform does not support side-by-side comparison, which makes me feel stressed and indecisive.

Customer Problem Statements – HouseHunt

Problem Statement 1

I am... a renter looking for a suitable rental home
 who is trying to... find suitable properties quickly,
 but... current property search tools are slow and return many irrelevant results,
 because... filters and ranking options are limited,
 which makes me feel... frustrated and overwhelmed.

Problem Statement 2

I am... a renter
 who is trying to... compare rental properties easily,
 but... the lack of a clear comparison feature forces me to manually switch between listings,
 because... the platform does not support side-by-side comparison,
 which makes me feel... stressed and indecisive.

Customer Problem Statements – HouseHunt

Problem Statement 1

I am...	I'm trying to...	But...	Because...	Which makes me feel
a renter	find suitable rental homes quickly	existing tools show too many irrelevant properties	filters and sorting options are limited	Frustrated

Problem Statement 2

I am...	I'm trying to...	But...	Because...	Which makes me feel
a renter	compare rental properties easily	no side-by-side comparison feature	I have to switch between listings manually	Indecisive

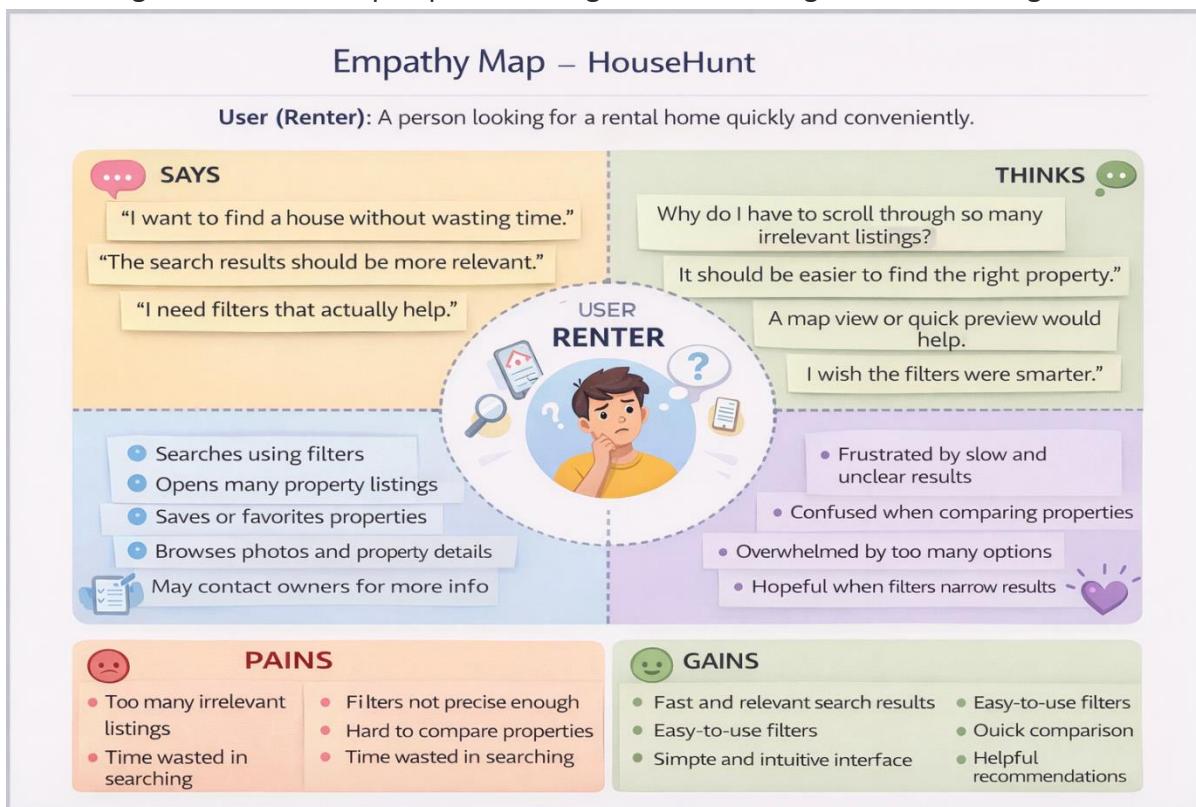
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a renter	find a suitable rental home quickly	it takes too much time to search through many listings	the search filters are limited and results are not well organized	frustrated and tired

PS-2	a renter	compare different rental properties easily	I have to open and check each property one by one	there is no proper comparison feature available	confused and stressed
------	----------	--	---	---	-----------------------

2.2 Empathy Map Canvas

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.

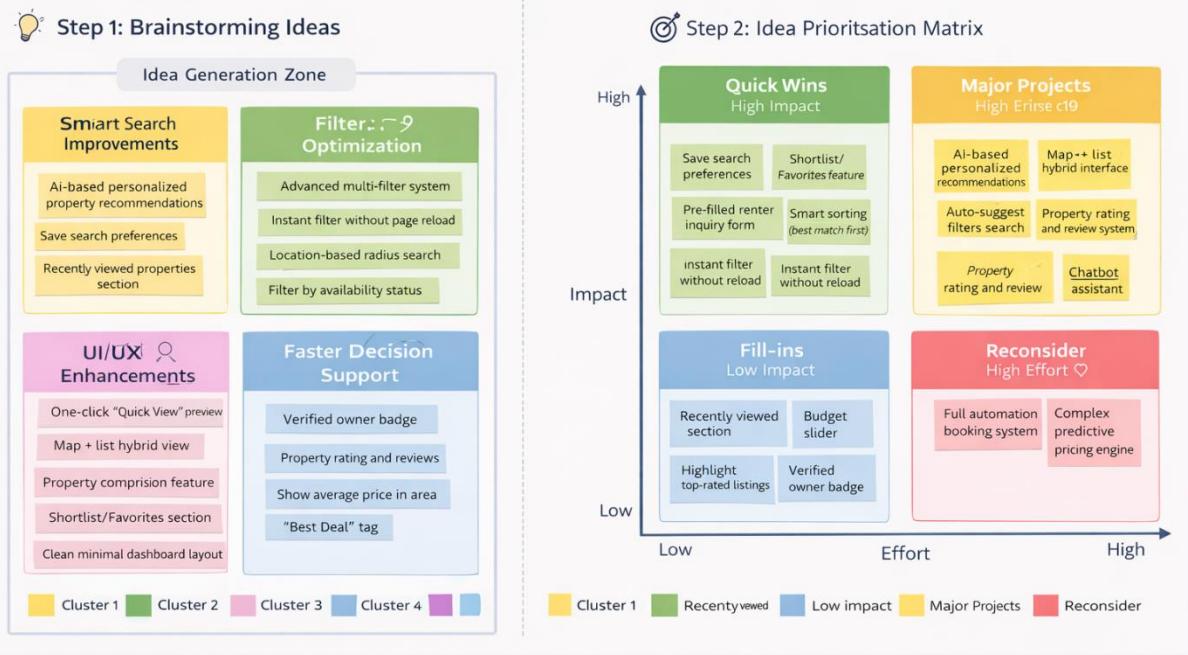
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



2.3 Brainstorming

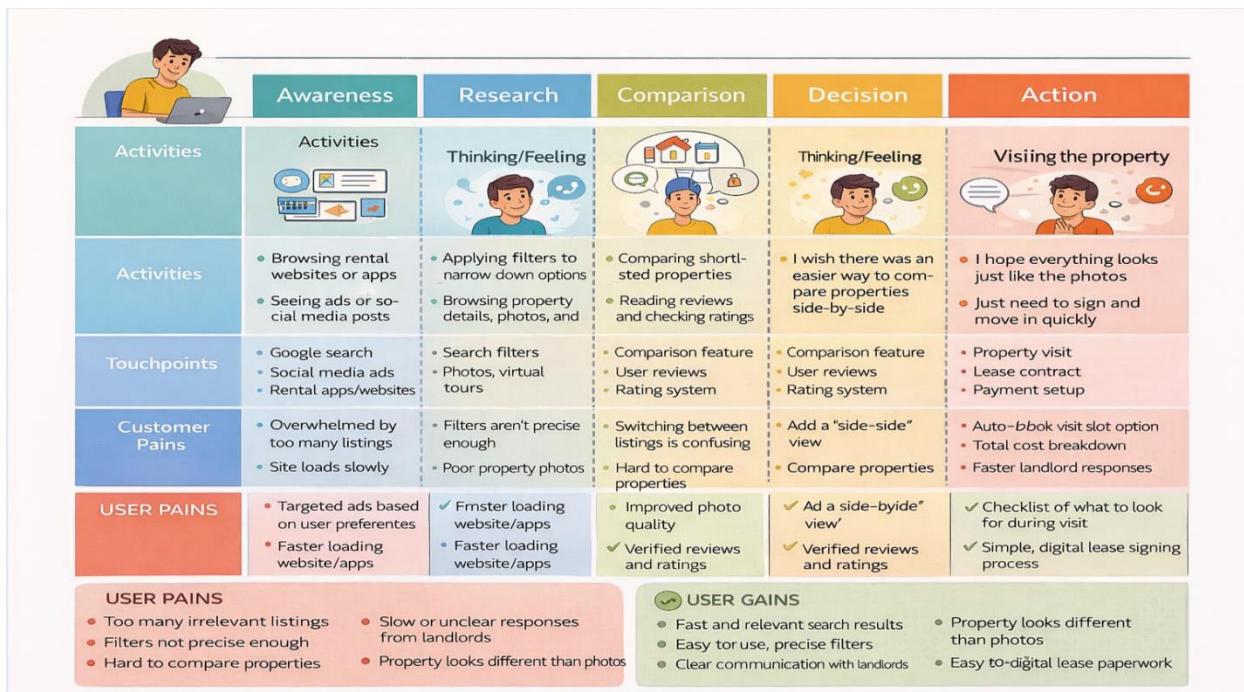
Brainstorm & Idea Prioritization – HouseHunt

How might we reduce property search time for renters?



3. Requirement Analysis

3.1 Customer Journey Map



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 & Authentication	System allows users (renters and owners) to register, login, and securely authenticate into the app.
FR-2	Property Listing Management	Owners can add, edit, delete, and update property listings with details, images, and availability status.
FR-3	Property Search & Filtering	Renters can search available properties using filters such as location, price, type, and amenities.
FR-4	Property Details & Viewing	Renters can view full details of a property (images, price, description, owner contact) on a dedicated page.
FR-5	Booking Request & Management	Renters can send booking requests; owners can approve or reject bookings, and the system tracks booking status.
FR-6	Admin User & Platform Control	Admin can approve owner accounts, manage inappropriate listings, view users, and monitor platform activity.

Non-functional Requirements:

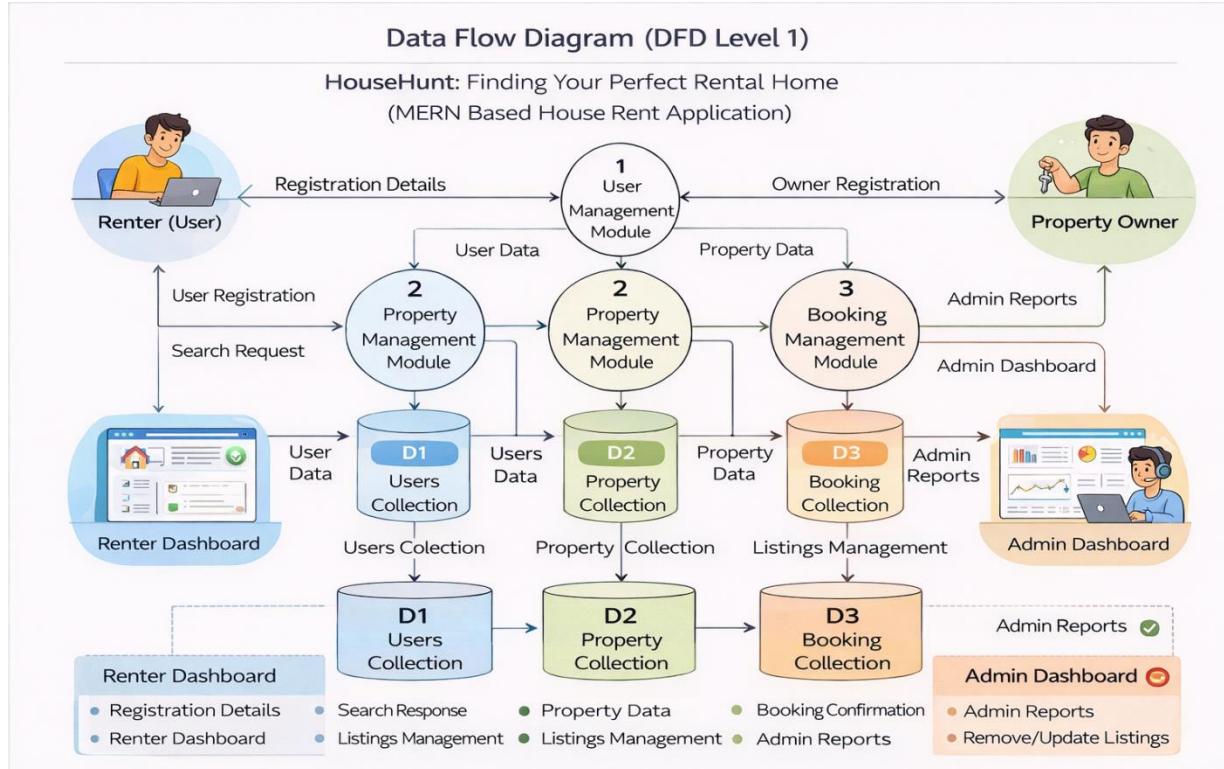
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Performance	The system should load property listings within 2 seconds for quick user interaction.
NFR-2	Scalability	The platform must support increasing numbers of users, properties, and booking requests without performance degradation.
NFR-3	Security	User credentials, personal data, and session tokens must be protected using secure practices (e.g., hashing, HTTPS).
NFR-4	Usability	The interface should be intuitive, responsive, and accessible on both web and mobile devices.
NFR-5	Reliability	The system should be available 99% of the time and handle data consistently without loss.

NFR-6	Maintainability	The system should allow easy updates and feature additions with well-structured code and modular design.
-------	-----------------	--

3.3 Data Flow Diagram

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.



3.4 Technology Stack

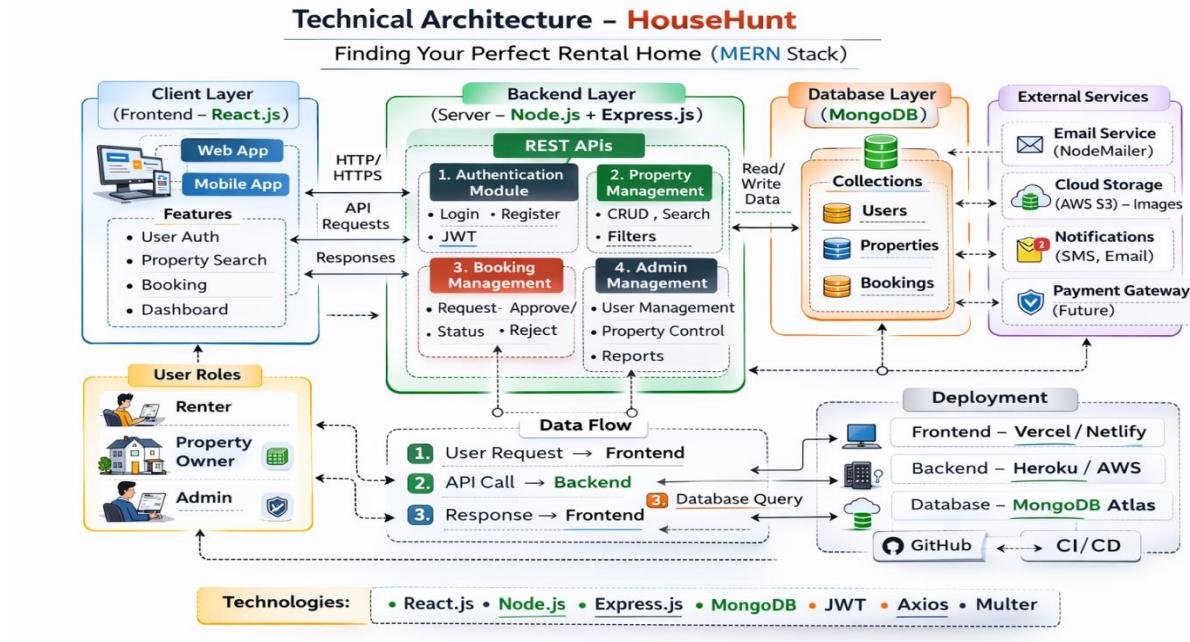


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Frontend framework to build responsive user views, dashboards, pages	React.js
2.	API Development	Backend framework to create RESTful APIs and route handling	Express.js
3.	Server Platform	Server runtime that executes backend code	Node.js
4.	Database	NoSQL database to store users, properties, and bookings	MongoDB
5.	Authentication	Token-based security for login sessions and protected routes	JSON Web Tokens (JWT)
6.	UI Libraries	UI component libraries for clean layout and interactive design	Bootstrap, Material UI, AntD
7.	HTTP Client	For frontend to backend API communication	Axios

8.	File Upload	Handles property image uploads and storage	Multer
9.	Middleware	Enables secure cross-origin requests and request parsing	CORS, Body-Parser
10	Security	Hashes passwords to secure user credentials	bcryptjs

Table-2: Application Characteristics:

S.No	Characteristics	Description & Technology
1.	Performance	The system responds quickly with minimal delay in property search (< 2 seconds) and navigation throughout the app
2.	Scalability	Handles increasing numbers of users and properties without performance impact due to MERN stack and modular design
3.	Security	Implements secure login, data encryption, hashed passwords, role-based access control (admin, owner, renter)
4.	Usability	The interface is intuitive, responsive, and easy to use on both desktop and mobile devices
5.	Availability	The application is designed to be highly available, reliable, and accessible 24/7
6.	Maintainability	Clean modular code with proper directory structure for easier updates, debugging, and feature enhancements
7.	Compatibility	Works across modern web browsers (Chrome, Firefox, Edge, Safari) and devices
8.	Reliability	Ensures consistent data updates and safe storage, with proper error handling and fail-safes

4. Project Design

4.1 Problem Solution Fit

The Problem

Renters face significant challenges while using existing rental property platforms. The search process is slow and overwhelming due to a lack of relevant filtering, limited comparison features, and poorly organized results. This causes renters to spend excessive time scrolling through irrelevant listings, leading to frustration and indecisiveness.

Additionally, current platforms often lack intuitive booking workflows and reliable status updates, further complicating the user experience.

The Solution

HouseHunt: Finding Your Perfect Rental Home is a user-centric rental platform designed to simplify and accelerate the property search experience. It provides advanced filtering options, interactive user interfaces, quick property previews, easy booking workflows, and real-time status updates. The application connects renters with property owners while offering admin oversight and governance. Its intuitive dashboard and structured presentation of property details make the process efficient and enjoyable.

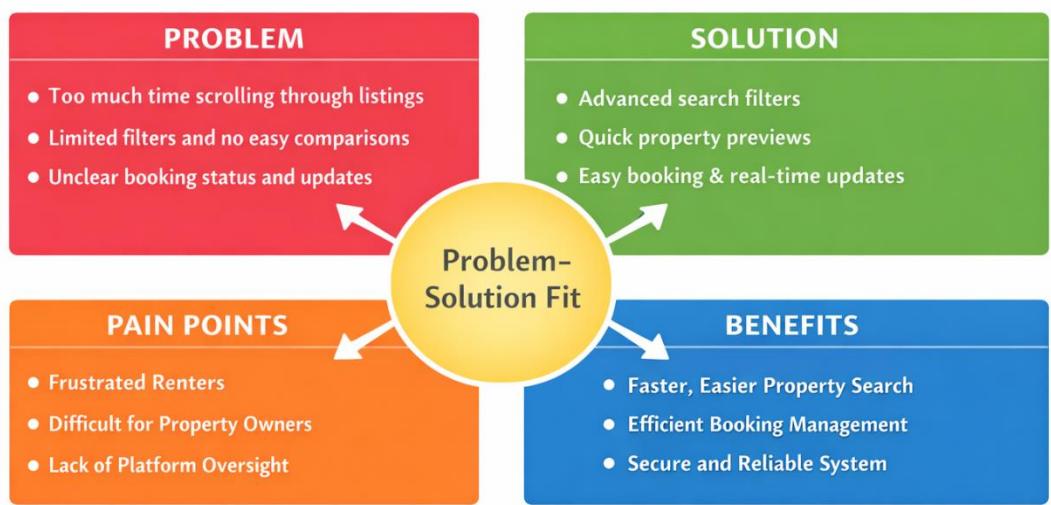
Why it Fits

HouseHunt aligns directly with the needs and pain points of renters, property owners, and administrators. By focusing on relevance, speed, and usability, the platform addresses key frustrations experienced by users of traditional rental apps. Renters benefit from organized workflows and personalized search tools, owners gain practical management features, and administrators can enforce platform standards. The solution fits because it reduces time spent searching, enhances decision-making, and ensures a secure and scalable rental environment.

Purpose

The purpose of HouseHunt is to provide a comprehensive and efficient rental property platform that connects renters with property owners using modern technologies. It aims to enhance the real-estate search experience by delivering reliable results, seamless booking mechanisms, and clear status tracking. The system also supports platform governance with admin oversight for secure and dependable usage.

Problem-Solution Fit

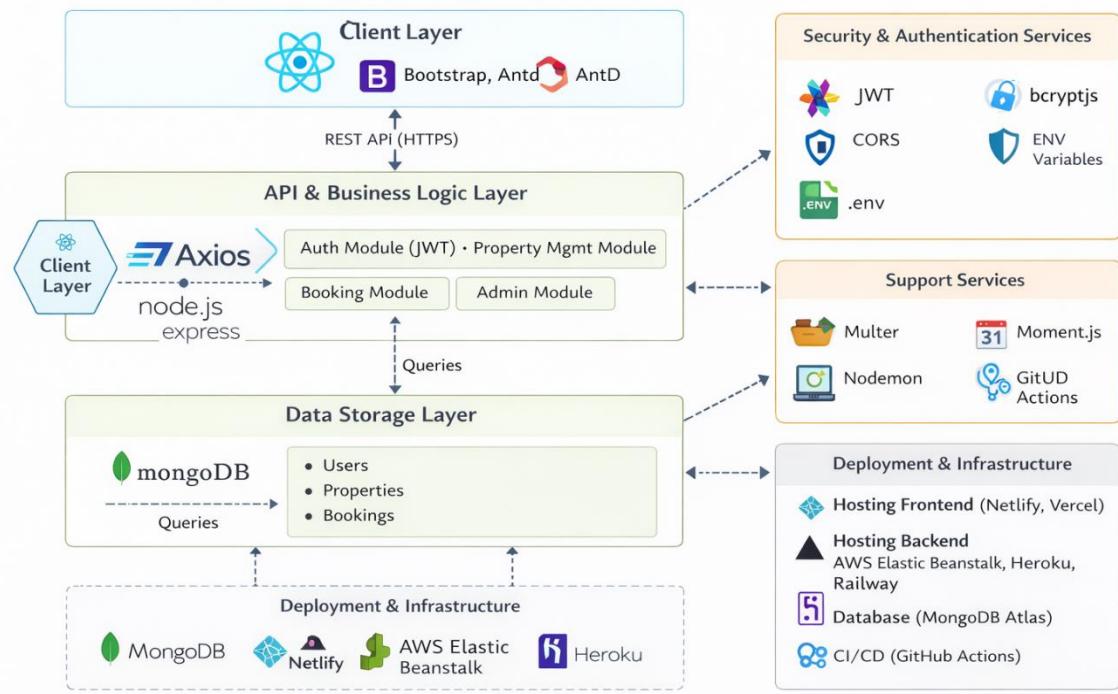


4.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Finding reliable and affordable rental houses is time-consuming due to scattered listings and lack of transparency.
2.	Idea / Solution description	A MERN-based house rental platform that connects renters, owners, and admins in one secure and organized system.
3.	Novelty / Uniqueness	Provides role-based access (Admin, Owner, Renter) with verified listings and real-time booking management.
4.	Social Impact / Customer Satisfaction	Ensures safe, transparent, and convenient house renting experience for both tenants and property owners.
5.	Business Model (Revenue Model)	Generates revenue through property listing fees, featured ads, and commission on successful bookings.
6.	Scalability of the Solution	Built on scalable MERN architecture, allowing expansion across cities with growing user and property data.

4.3 Solution Architecture

Solution Architecture for HouseHunt



5. Project Planning and Scheduling

5.1 Project Planning

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	US-1	Renter Registration	5	High	Bandaru Raatna Sai
Sprint-1	Login	US-2	Renter Login	3	High	Bandi Satwika
Sprint-1	Property Search	US-3	Search properties with filters	8	High	Kanthete Naga Durga Yaswanth
Sprint-1	View Property Details	US-4	View property details page	5	High	Karanki Naga Mounika
Sprint-1	Owner Registration	US-7	Owner Registration	5	High	Kanthete Naga Durga Yaswanth
Sprint-1	Add Property	US-8	Add new property listing	8	High	Bandi Satwika

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Booking Property	US-5	Book property	8	High	Bandaru Raatna Sai
Sprint-2	View Booking History	US-6	View booking history	5	Medium	Karanki Naga Mounika
Sprint-2	Update Property	US-9	Update property details	5	Medium	Kanthete Naga Durga Yaswanth
Sprint-2	Delete Property	US-10	Delete property listing	3	Medium	Bandi Satwika
Sprint-2	View Bookings	US-11	Owner view booking requests	5	High	Kanthete Naga Durga Yaswanth
Sprint-2	Manage Users	US-12	Admin manage users	5	High	Karanki Naga Mounika
Sprint-2	Manage Properties	US-13	Admin manage properties	5	High	Bandaru Raatna Sai
Sprint-3	View Reports & Dashboard	US-14	Admin view reports and analytics dashboard	5	Medium	Bandi Satwika

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	15 Feb 2026	20 Feb 2026	20	21 Feb 2026
Sprint-2	20	6 Days	15 Feb 2026	20 Feb 2026	20	21 Feb 2026
Sprint-3	20	6 Days	15 Feb 2026	20 Feb 2026	20	21 Feb 2026

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.



6. Functional and Performance Testing

6.1 Performance Testing

Project Overview:

Project Name: HouseHunt: Finding Your Perfect Rental Home

Project Description: HouseHunt is a MERN-stack based web application that enables renters to search and book rental properties, property owners to manage listings and bookings, and administrators to oversee users and platform activities through role-based access control.

Project Version: Version 1.0 (MVP Release)

Testing Period: 20 February 2026 to 23 February 2026

Testing Scope:

List of Features and Functionalities to be Tested

- User registration and login (Renter & Owner)
- Role-based access control (Renter, Owner, Admin)
- Property search and advanced filtering

- View property details with images and contact info
- Property addition, update, and deletion (Owner)
- Booking request and status tracking
- Booking approval/rejection (Owner)
- Admin user and property management
- Dashboard and report display
- Responsive UI on desktop and mobile

List of User Stories or Requirements to be Tested:

- US-1: Renter Registration
- US-2: Renter Login
- US-3: Property Search
- US-4: View Property Details
- US-5: Booking Property
- US-6: View Booking History
- US-7: Owner Registration
- US-8: Add Property
- US-9: Update Property
- US-10: Delete Property
- US-11: View Bookings
- US-12: Admin Manage Users
- US-13: Admin Manage Properties
- US-14: Admin Dashboard & Reports

Testing Environment:

URL/Location: <http://localhost:5173> (Development Environment)

Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-01	Renter Registration	Enter valid name, unique email, password → Click Register	Account created successfully and redirected to login	Account created and redirected to login	Pass
TC-02	Renter Login (Valid)	Enter registered email & correct password → Click Login	User logged in and dashboard displayed	Dashboard displayed successfully	Pass

TC-03	Renter Login (Invalid)	Enter wrong password → Click Login	Error message shown for invalid credentials	Error message displayed	Pass
TC-04	Property Search with Filters	Select location & price range → Click Search	Filtered properties displayed correctly	Relevant properties shown	Pass
TC-05	Book Property	Click Book on available property → Confirm booking	Booking stored and status set to "Pending"	Booking saved with pending status	Pass
TC-06	Owner Add Property	Enter property details → Upload image → Click Save	Property saved and visible in listings	Property successfully added	Pass

Bug Tracking:

Bug ID	Bug Description	Steps to reproduce	Severity	Status	Additional feedback
BUG-01	Minor UI misalignment on property cards in mobile view	Open app on mobile screen → Navigate to property listing page	Low	Open	Adjust CSS margins and card responsiveness for smaller screens
BUG-02	Search filter does not reset after page refresh	Apply filters → Refresh browser page	Low	Open	Reset filter state on page reload to improve user experience
BUG-03	Confirmation message disappears too quickly after booking	Book a property → Observe confirmation message duration	Low	In Progress	Increase message display time to 3–4 seconds for better visibility

Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Open	2	0	0	0	2

In Progress	1	0	0	0	1
Resolved	0	0	0	0	0
Closed	0	0	0	0	0
Total	3	0	0	0	3

Test Case Analysis:

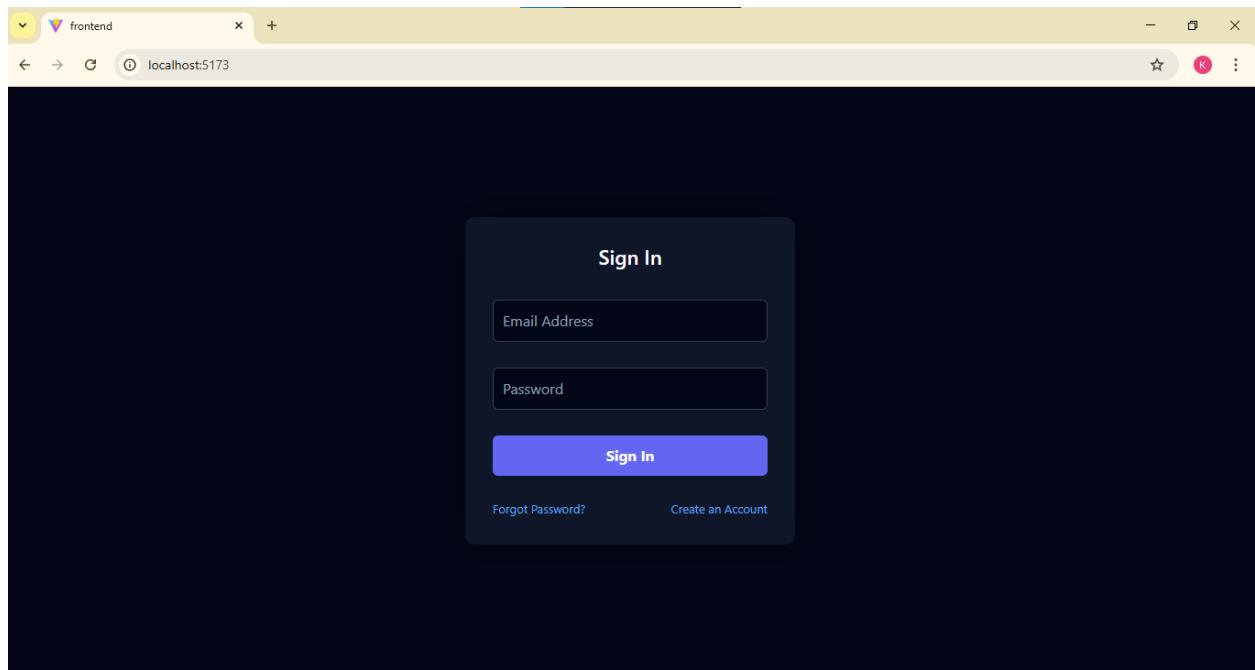
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Registration & Login	3	0	0	3
Property Search & View	1	0	0	1
Booking Module	1	0	0	1
Property Management (Owner)	1	0	0	1
Total	6	0	0	6

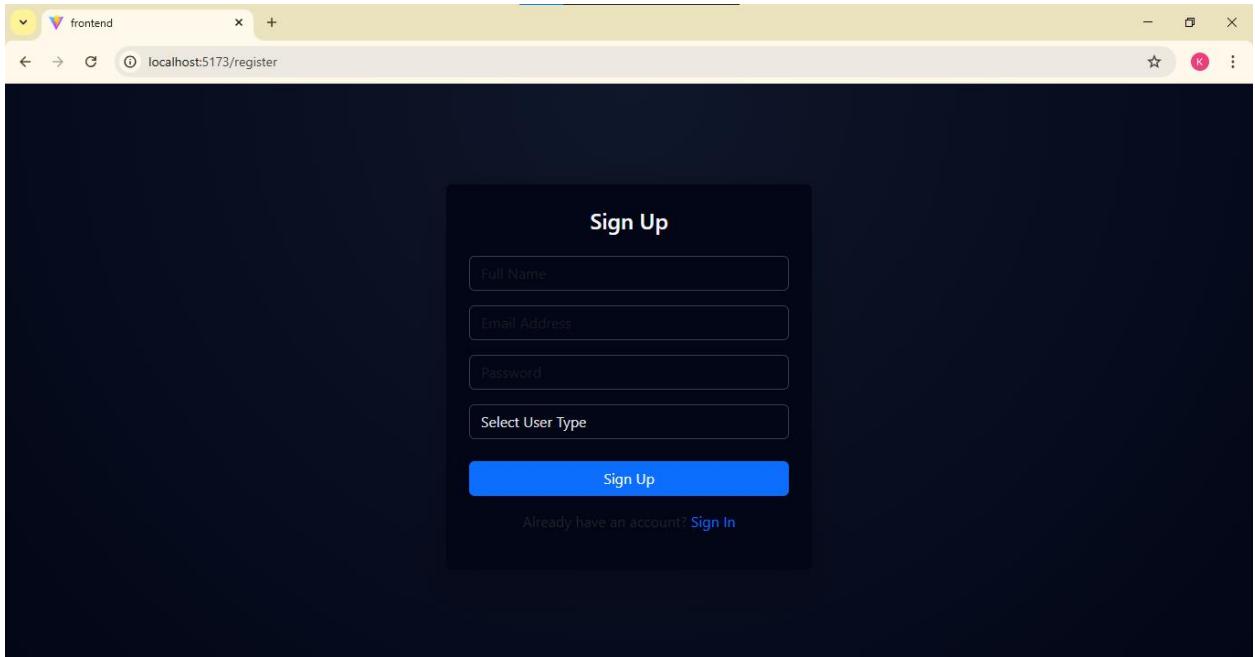
7. Results

7.1 Output Screenshots

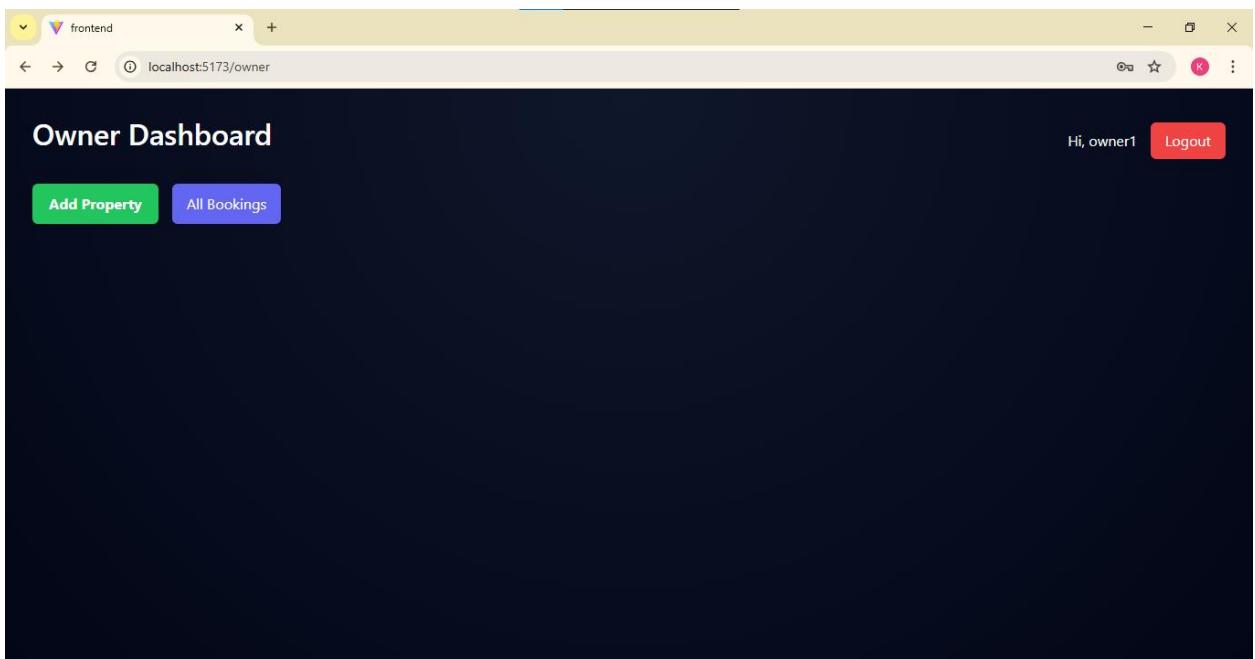
Combined Login Page for Renter and Owner



SignUp Page



Owner Dashboard



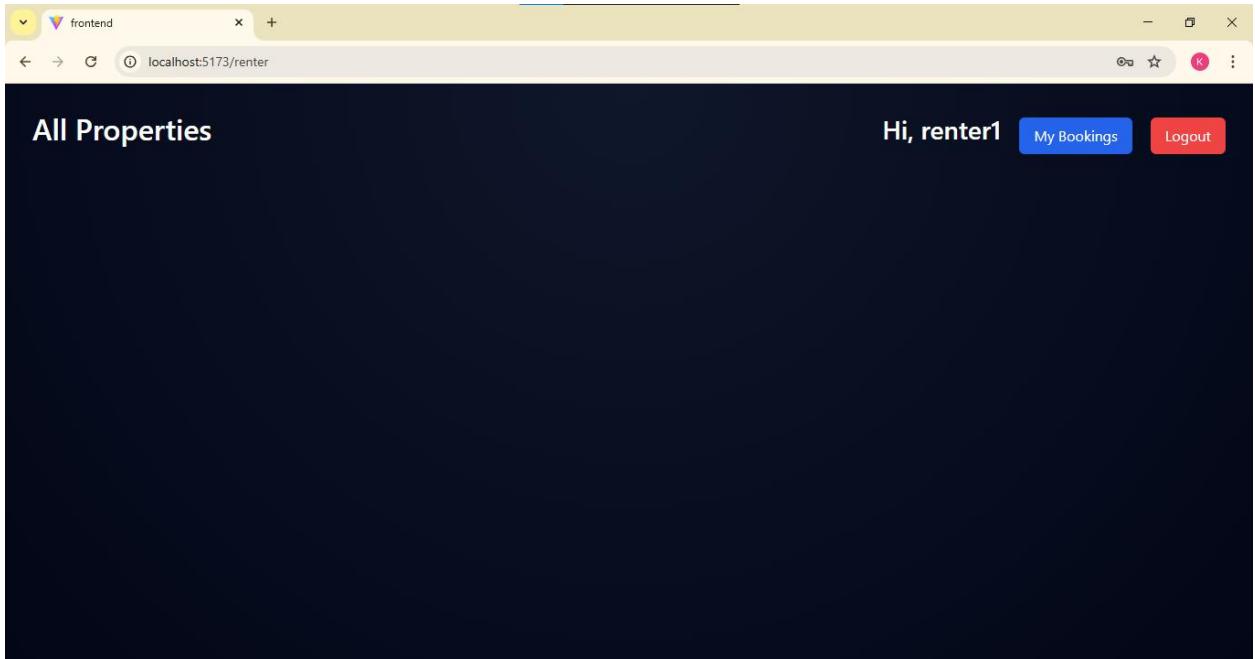
Owner Dashboard-Add property

A screenshot of a web browser window titled "RentEase" showing the "Add New Property" form. The URL in the address bar is "localhost:5173/add-property". The form consists of several input fields: "Select Property Type", "Select Ad Type", "Full Address", "Owner Contact", "Amount", and "Additional Information". Below these fields is a file upload section with "Choose Files" and "No file chosen". At the bottom is a large blue "Submit Property" button.

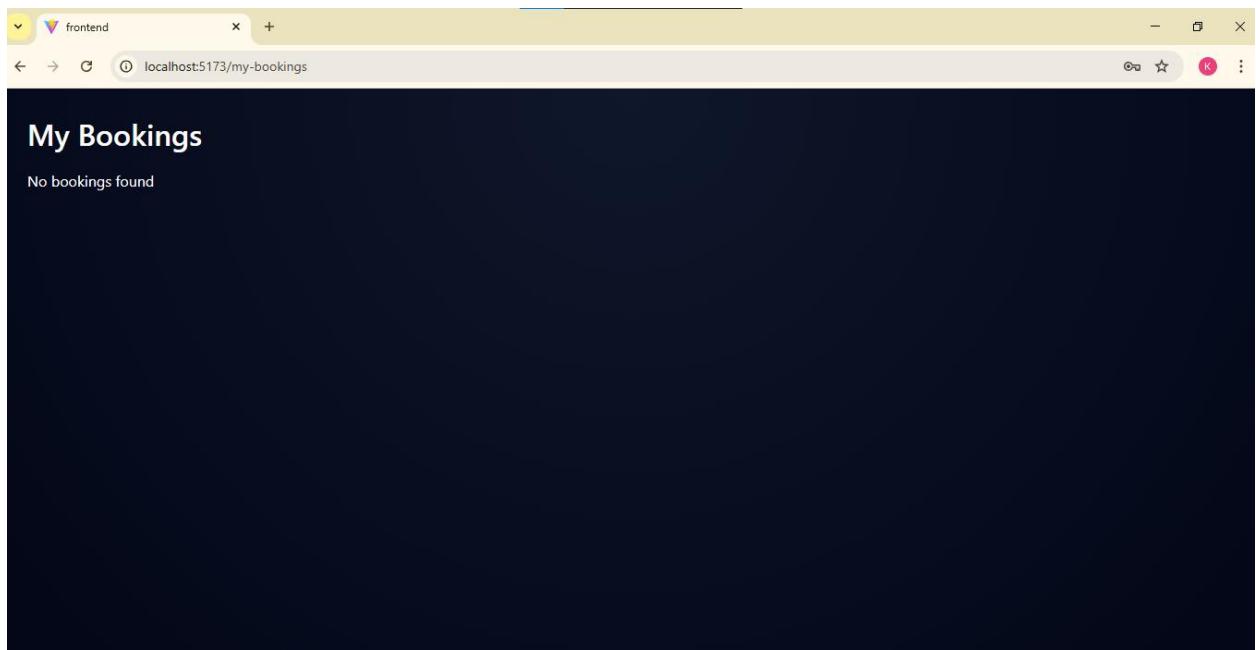
Owner Dashboard - All Bookings Page

A screenshot of a web browser window titled "RentEase" showing the "Booking Requests" page. The URL in the address bar is "localhost:5173/owner/bookings". The page displays the message "No bookings yet".

Renter Dashboard



Renter Dashboard – All bookings Page



8. Advantages & Disadvantages

Advantages

- Provides a centralized platform for property owners and renters to manage listings and bookings efficiently.
- Role-based authentication ensures secure access for owners and renters.
- Image upload functionality enhances property visibility and user decision-making.
- Real-time booking system improves communication between renters and owners.

- Built using the MERN stack, making the application scalable and easy to maintain.

Disadvantages

- Requires a stable internet connection to access all features.
- Initial setup and configuration can be complex for beginners.
- Limited advanced filtering and search options in the current version.
- No integrated payment gateway for rent or booking transactions.
- Performance may degrade with very large datasets without further optimization.

9. Conclusion

The HouseHunt (RentEase) project successfully demonstrates the development of a full-stack web application using the MERN stack to simplify property listing and rental management. The system enables property owners to list properties with images and manage booking requests, while renters can easily browse, view, and book available properties.

Through role-based authentication, secure data handling, and a responsive user interface, the application ensures a smooth and reliable user experience. Despite a few limitations, the project lays a strong foundation for future enhancements and showcases practical implementation of real-world full-stack development concepts.

10. Future Scope

The HouseHunt application can be further improved by implementing the following enhancements in future versions:

- 1. Cloud-Based Image Storage**
 - Integrate cloud storage services such as AWS S3 or Cloudinary to securely store and manage property images.
 - This will improve scalability and prevent data loss during server restarts or deployments.
- 2. Advanced Search and Filters**
 - Add filters based on price range, location, property type, and ad type.
 - Implement sorting options such as lowest price, highest price, and newest listings.
- 3. Email and Notification System**
 - Send email or in-app notifications for booking requests, approvals, and rejections.
 - Notify owners and renters about important updates in real time.
- 4. Online Payment Integration**
 - Integrate secure payment gateways (e.g., Razorpay, Stripe) for booking confirmation or advance payments.
 - This will enhance trust and automate the rental process.
- 5. Improved Security Measures**
 - Implement refresh tokens and role-based access control (RBAC).
 - Add rate limiting and request validation to prevent unauthorized access.
- 6. Automated Testing**

- Introduce automated unit and integration testing using tools like Jest and Cypress.
- This will ensure better reliability and easier maintenance of the application.

7. Enhanced User Experience

- Improve UI/UX with animations, responsive layouts, and accessibility features.
- Add dark/light mode support.

8. Admin Dashboard

- Create an admin panel to manage users, properties, and bookings.
- Enable monitoring and moderation features.

9. Mobile Application

- Develop a mobile version of the application using React Native or Flutter.
- This will allow users to access the platform on the go.

11. Appendix

Github Link

<https://github.com/Yaswanth181004/HouseHunt-Finding-Your-Perfect-Rental-Home-LTVIP2026TMIDS38973>

Project Demo Link

<https://drive.google.com/file/d/1P7VxEgTiWRb071xeYuZdxU8ANqgbtMc/view?usp=sharing>