HOUSEHUNT: FINDING YOUR PERFECT RENTAL HOME

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The House Hunt Project is focused on finding a residential property that aligns with the buyer's financial, lifestyle, and long-term goals. It involves researching ideal locations, identifying essential features (such as size, layout, and safety), and evaluating properties within a defined budget. The process includes outlining needs versus wants, assessing neighborhood amenities like schools and commute times, and visiting shortlisted homes. The ultimate objective is to secure a property that is not only functional and affordable but also a good investment for the future.

The project follows a structured approach starting with requirement gathering, followed by property research, financial planning, and decision-making. Tools like online real estate platforms, mortgage calculators, and neighborhood databases are used to streamline the process. Success is measured by securing a property that satisfies at least 90% of the buyer's key criteria, is acquired within the timeline and budget, and ensures a smooth transition into the new home.

1.2 PURPOSE

The primary purpose of the HouseHunt project is to streamline and digitize the house rental process by offering a centralized, user-friendly platform for renters, property owners, and administrators. By leveraging modern web technologies (MERN stack), this application enhances the efficiency, transparency, and convenience of finding and managing rental homes.

FEATURES:

1. Simplify the Rental Search Process
    - Provide renters with a centralized database of verified rental listings.
    - Allow users to filter and search for properties based on specific needs (e.g., budget, location, amenities).
2. Enable Direct Communication Between Renters and Owners
    - Facilitate secure, in-app messaging for inquiries and lease negotiations.
    - Build trust and reduce the need for third-party agents.
3. Empower Property Owners with Management Tools
    - Allow property owners to easily add, update, and manage property listings.
    - Let owners handle booking requests and communicate directly with potential tenants.

4. Ensure Platform Integrity with Admin Oversight
    - Admin dashboard for approving owners, monitoring platform activity, and enforcing policies.
    - Maintain a secure and trustworthy platform for all users.
5. Provide End-to-End Rental Journey
    - From discovery and inquiry to booking confirmation and lease agreement—everything happens within the app.
    - Support seamless move-in by providing clear rental steps and status updates.
6. Support Scalability and Real-Time Data Handling
    - Use MongoDB and Express for efficient data handling and scalability.
    - Leverage responsive frontend (React + Bootstrap + Material UI) for optimal user experience.

7. Build a Transparent and Secure Rental Ecosystem
    - Keep users informed at every stage of the rental journey.
    - Provide secure login and authentication for all user roles (Renter, Owner, and Admin).

2. IDEATION PHASE

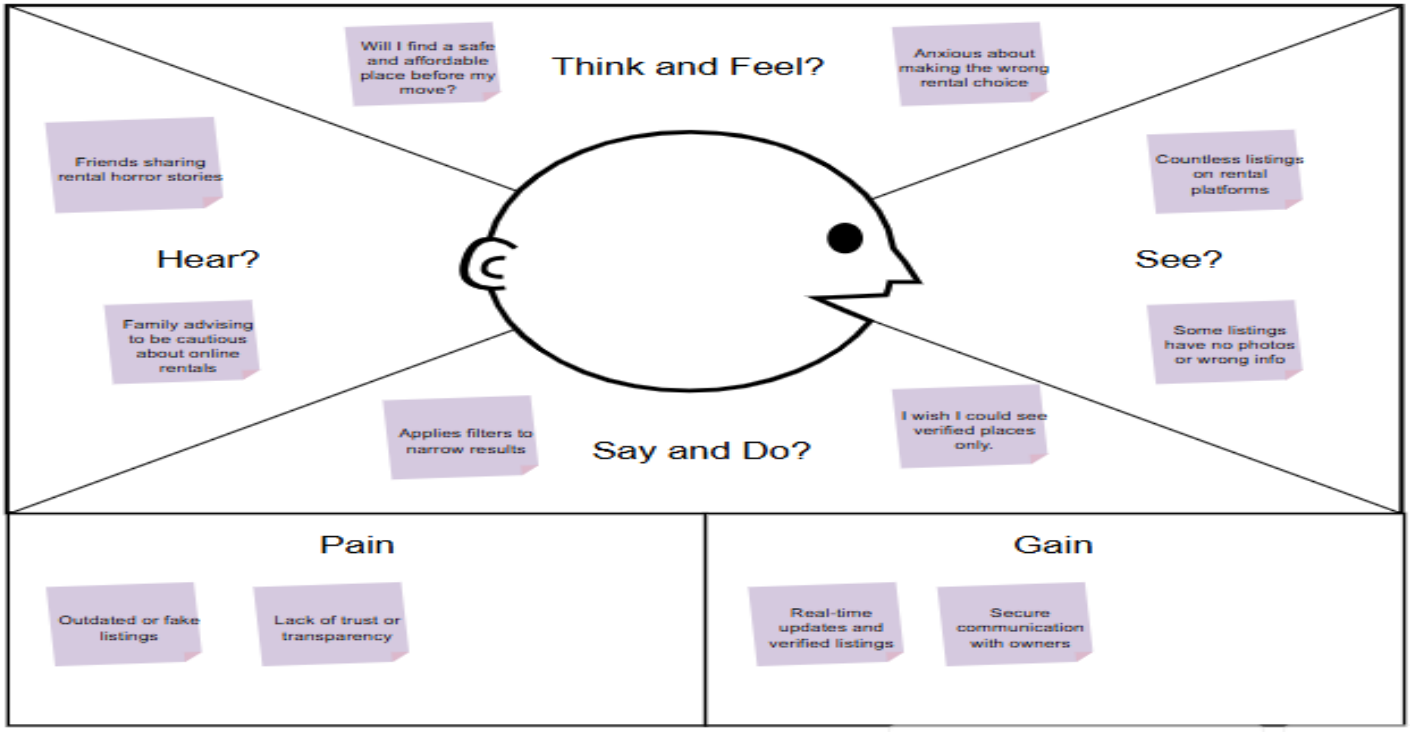
2.1 PROBLEM STATEMENT

Customer Problem Statements:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a working professional relocating to a new city	find a house that fits my budget and commute needs	I can't find accurate listings or verify details easily	most platforms are out-dated or lack transparency	frustrated and anxious about making the wrong choice
PS-2	a first-time home buyer	understand the best options available in my price range	I keep getting overwhelmed by choices and conflicting informat	there's no single trusted source that simplifies the decision-making process	confused and unsure about how to proceed

			ion		

2.2 EMPATHY MAP CANVAS



2.3 BRAIN STORMING

Step-1: Team Gathering, Collaboration and Selecting the Problem Statement

In the initial phase of the project, the team comes together to align on goals, roles, and expectations. This step involves building a collaborative environment where every team member actively contributes their ideas and insights. Effective communication and brainstorming sessions are conducted to explore various real-world problems that the team is passionate about solving.

The focus is on identifying a relevant, meaningful, and user-centered problem that can be addressed through technology. The team evaluates multiple ideas, discusses user pain points, and eventually selects one well-defined problem statement that will guide the rest of the project.

This foundational step ensures that all team members are on the same page and committed to solving a problem that truly matters to the target users.

Step-2: Brainstorm, Idea Listing and Grouping

After selecting the initial problem statement, the team enters the ideation phase, where creativity and open thinking are encouraged. This step involves brainstorming a wide range of possible solutions, features, or approaches that could address the selected problem from different angles.

Each team member shares ideas—no matter how big or small—while others listen and build upon them. These ideas are then documented in a shared space (whiteboard, sticky notes, or digital tools like Miro, Google Docs, etc.)

Step-3: Idea Prioritization

Once all ideas have been brainstormed and grouped, the next crucial step is prioritizing them based on impact, feasibility, and user value. Not every idea can be implemented immediately, so the team must decide which features or solutions offer the greatest value to the user while being realistic to develop within the project scope and timeline.

Common techniques used for idea prioritization include:

- MoSCoW Method (Must Have, Should Have, Could Have, Won't Have)
- Impact vs. Effort Matrix
- Voting or Dot Voting to reflect team consensus

This process helps:

- Focus on high-impact, achievable features
- Avoid overloading the app with unnecessary complexity
- Align the team around what matters most for the user experience

3. REQUIREMENT ANALYSIS

3.1 CUSTOMER JOURNEY MAP

	AWARENESS	REGISTRATION	BROWSING	INQUIRY	BOOKING	CONFIRMATION	MOVE-IN	POST MOVE SUPPORT
ACTIONS	Hears about HouseHunt from friends or online ads	Downloads app, creates account	Searches properties with filters	Contacts property owners via app messaging	Submits booking request and negotiates terms	Receives booking confirmation	Moves into the rental property	Uses app for maintenance requests or feedback
THOUGHTS	"Is this app reliable for finding rentals?"	"I hope registration is quick and easy."	"Will I find something that fits my budget and needs?"	Will they respond quickly and honestly?"	Is this process secure and transparent?"	"Great, I finally found a place!"	Hope everything goes smoothly now."	"Can I easily get help if needed?"
FEELINGS	Curious, hopeful	Slightly anxious	Optimistic but cautious	Nervous, hopeful	Relieved but cautious	Excited, reassured	Happy but slightly anxious	Confident if supported
PAIN POINTS	Uncertainty about app trustworthiness	Complicated signup or verification	Overwhelming number of listings, outdated info	Slow or no responses from owners	Unclear lease terms, payment security concerns	Delays in confirmation notifications	Unexpected issues at move-in or lack of support	Difficulties in communicating maintenance issues
OPPORTUNITIES	Showcase reviews, ratings, and verified listings	Simplify signup with social login options	Real-time listings, intuitive filters	Instant notifications, verified owner profiles	Digital lease agreements, secure payment gateways	Push notifications, status tracking	Move-in checklists, customer support features	In-app support chat, feedback systems

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 through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP

FR-3	Property Browsing & Filtering	View list of available rentals Apply filters (location, price, type) View property details (images, owner info, rent)
FR-4	Booking and Inquiry	Send inquiry to property owner Submit booking request - View booking status
FR-5	Admin Panel	Approve owner accounts Approve property listings Monitor user activities
FR-6	Property Management (Owner Side)	Add, edit, or delete property listings Update availability status

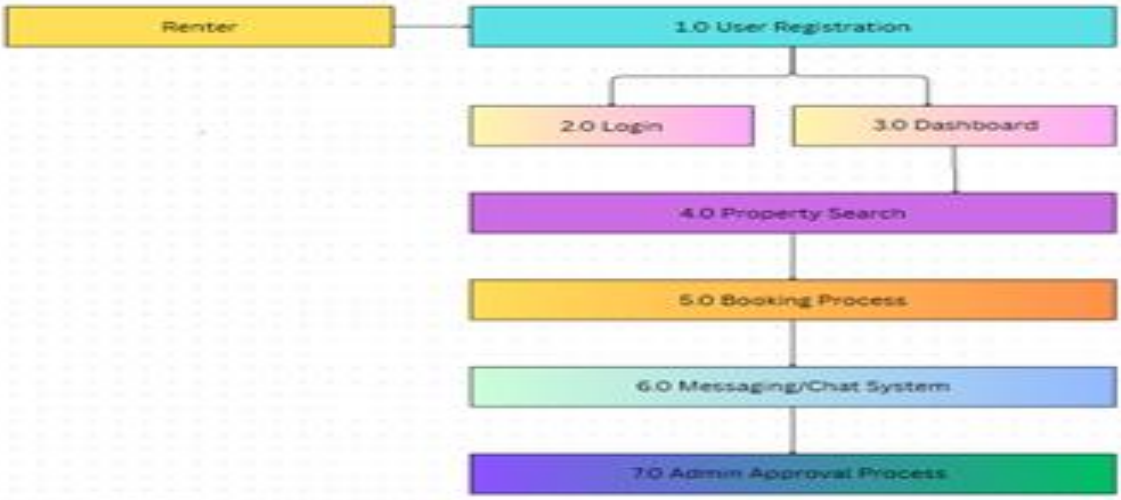
Non-functional Requirements:

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

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The app must offer a simple, intuitive interface for users of all experience levels
NFR-2	**Security**	All user data and transactions must be encrypted; authentication with OTP or email verification.
NFR-3	**Reliability**	The system must function consistently without frequent crashes or failures
NFR-4	**Performance**	App should load content (e.g., listings, messages) in under 2 seconds.
NFR-5	**Availability**	The platform must maintain at least 99.5% uptime.
NFR-6	**Scalability**	System should handle increasing users and listings without affecting performance.

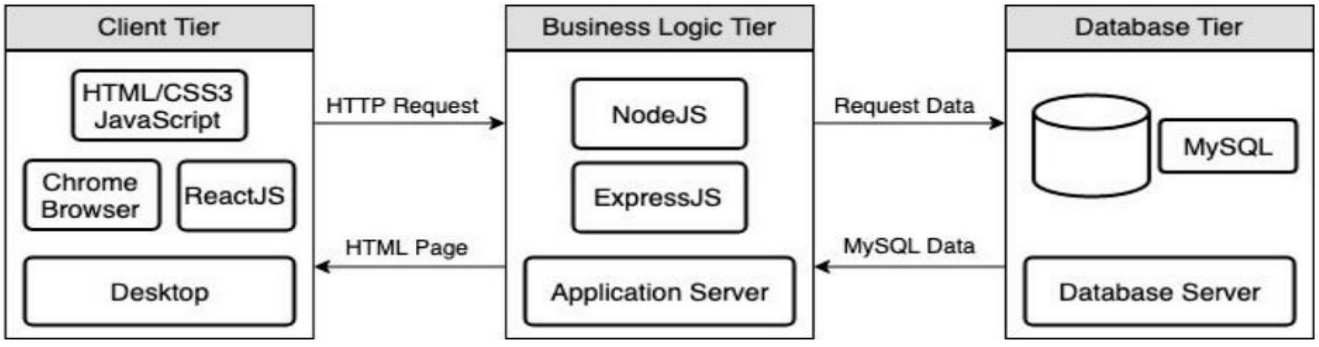
3.3 DATA FLOW DIAGRAM


Property Owner
Renter
Admin
House Hunt System


Renter
1.0 User Registration
2.0 Login
3.0 Dashboard
4.0 Property Search
5.0 Booking Process
6.0 Messaging/Chat System
7.0 Admin Approval Process

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile)	Registration	USN-1	As a user, I can register for the app using my email and password.	I can access my account/dashboard after registration.	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once registered.	I can receive a confirmation email and verify my account.	High	Sprint-1
		USN-3	As a user, I can register through Facebook.	I can sign up and access the dashboard using Facebook.	Low	Sprint-2
		USN-4	As a user, I can register using Gmail authentication.	I can log in using Gmail OAuth and access the dashboard.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the app with email and password.	I can access my dashboard upon successful login.	High	Sprint-1
	Dashboard	USN-6	As a user, I can view available property listings and filter them.	I can see listings and apply filters by price, location.	High	Sprint-2
Customer (Web)	Registration & Login	USN-7	As a web user, I can sign up and log in just like mobile users.	I can access all functionalities via the web browser.	Medium	Sprint-2
	Dashboard	USN-8	As a web user, I can search for properties and view details.	I can see images, prices, and location on web UI.	High	Sprint-2
Customer Care Exec	User Support	USN-9	As a support agent, I can view user issues and respond via dashboard.	I can manage tickets and send responses to users.	Medium	Sprint-3
	Reporting	USN-10	As a support agent, I can generate reports on user activities.	I can download issue logs in PDF or Excel.	Low	Sprint-3
Administrator	Admin Management	USN-11	As an admin, I can approve or reject property owner registrations.	I can view pending requests and take action.	High	Sprint-1
	Platform Governance	USN-12	As an admin, I can monitor listings and block suspicious users.	I can flag or remove content violating platform rules.	High	Sprint-2
	Analytics	USN-13	As an admin, I can view usage analytics across the app.	I can track total users, listings, and monthly activity.	Medium	Sprint-3

3.4 TECHNOLOGY STACK



S.No	Component	Description	Technology
1.	User Interface	Web interface for users (Renters, Owners, Admin)	HTML, CSS, JavaScript, React.js, Bootstrap, Material UI
2.	Application Logic-1	Backend logic for registration, booking, messaging	Node.js, Express.js
3.	Application Logic-2	Image upload, status management	Multer, Custom Middleware
4.	Application Logic-3	Email/OTP notifications	Nodemailer, Twilio (for OTP)
5.	Database	Stores user profiles, listings, messages, bookings	MongoDB (NoSQL)
6.	Cloud Database	Cloud-hosted version of MongoDB	MongoDB atlas
7.	File Storage	Storage for property images	Cloudinary, Amazon S3, or Local Filesystem
8.	External API-1	Google Maps integration for location-based search	Google Maps API
9.	External API-2	Email/phone verification	Twilio API, SendGrid API
10.	Machine Learning Model	(Future Scope) Detect fake listings, auto-tag properties	Custom ML Model (Python, TensorFlow or FastAPI backend)
11.	Infrastructure (Server / Cloud)	Hosting of frontend/backend, deployment & scaling	Render, Vercel, or AWS EC2, Docker, Kubernetes (if scaled).

4. PROJECT DESIGN

4.1 PROBLEM SOLUTION FIT

HouseHunt aims to address the pressing issues faced by individuals looking to rent homes, particularly in urban areas. Renters—such as working professionals, students, and small families—often face difficulties navigating the fragmented rental property market. Current options like OLX, Facebook Marketplace, or traditional agents are either unverified, outdated, or lack transparency. These challenges lead to wasted time, increased effort, and a lack of trust in the rental process.

The core problem is the absence of a centralized, trustworthy, and efficient platform that allows renters to discover, evaluate, and book rental properties with ease and confidence. Users struggle with fake listings, delayed communication, and an overall lack of reliability in property information.

HouseHunt offers a streamlined solution by creating a mobile and web-based rental application where users can:

- View verified property listings with up-to-date photos, pricing, and availability.
- Use smart search filters based on location, budget, amenities, and more.
- Communicate directly with property owners via in-app messaging.
- Book properties securely with status tracking and confirmation notifications.
- Rely on admin-approved listings, ensuring that only genuine owners and properties are on the platform.

By solving frequent annoyances like scams, incomplete information, and unresponsive landlords, HouseHunt improves the rental experience and builds trust with its users. The app

fits naturally into the digital behavior of its target users—who are already mobile-savvy and value speed, security, and simplicity in their online transactions.

The solution not only addresses the current pain points but also increases user satisfaction, retention, and overall efficiency in the home rental process. It positions itself as a one-stop, credible solution in an otherwise chaotic market.

Purpose:

The purpose of defining the Problem–Solution Fit for HouseHunt is to ensure that the product is solving a real, validated problem faced by renters. By understanding the user's pain points and current behavior, we can:

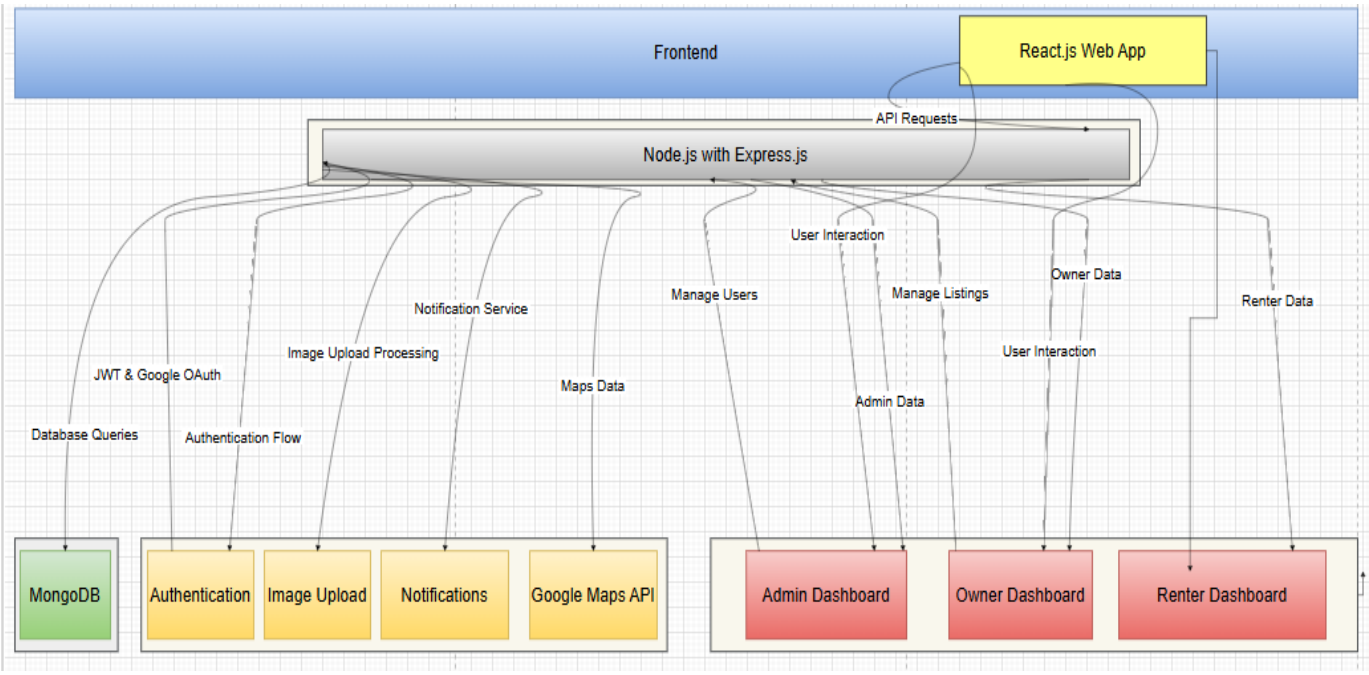
- Solve complex rental issues in a way that aligns with how customers actually search and book homes.
- Accelerate user adoption by integrating features that match user expectations and daily behavior.
- Sharpen communication strategies with focused messaging that addresses real frustrations like fake listings or slow responses.
- Increase engagement and trust by solving high-frequency, high-impact problems like availability mismatch, scam risk, or disorganized listings.
- Build a usable and desirable solution that fits into the user’s lifestyle and delivers value from the first interaction.

4.2 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Urban renters face difficulties finding reliable, verified rental homes due to scattered information, scams, and lack of direct, fast communication with owners.
2.	Idea / Solution description	HouseHunt is a web and mobile-based rental platform that connects renters with verified property listings. It enables smart search filters, real-time availability, secure in-app messaging, and booking confirmations—all managed within a streamlined, user-friendly interface.
3.	Novelty / Uniqueness	Unlike generic listing platforms, HouseHunt focuses on verified property data, admin-approved owners, and direct messaging. Its real-time status updates and

		secure booking system eliminate uncertainty and fraud, offering a complete rental journey in one app.
4.	Social Impact / Customer Satisfaction	HouseHunt reduces housing search stress and improves safety and transparency in the rental process. It empowers students, families, and professionals to find homes faster, safer, and without intermediaries. This boosts customer trust and satisfaction.
5.	Business Model (Revenue Model)	Freemium model with paid listing options for property owners and agents. Revenue is generated via: – Premium listing charges – Subscription plans for agencies – Commission on successful bookings – In-app ads (optional)
6.	Scalability of the Solution	HouseHunt is built using scalable MERN architecture. The platform can expand to support more cities, multiple property types (PGs, hostels, co-living), and integrate AI-driven property recommendations or chatbot assistance for user support.

4.3 SOLUTION ARCHITECTURE



The architecture follows a client-server model enabling efficient data flow and secure interactions between users and the system.

1. **Client Layer (Frontend):**
   Users access the HouseHunt application through a web or mobile interface built with React.js and styled using Bootstrap and Material-UI. The frontend handles user input, search filters, property browsing, and messaging features.

2. **API Gateway & Backend Layer:**
   Requests from clients are routed to the backend API server implemented using Express.js on Node.js. This layer manages business logic, user authentication via JWT and OAuth, and validation of user requests.

3. **Database Layer:**
   MongoDB stores all application data including user profiles, property listings, booking details, and transaction records. It supports flexible, scalable document storage.

4. **Authentication Service:**
   Secure user login and registration are managed with JSON Web Tokens (JWT) and OAuth providers (Google, Facebook). Email and OTP verification are handled via integrated email/SMS APIs (e.g., SendGrid, Twilio).

5. **File Storage Service:**
   Property images and documents are uploaded and stored in cloud storage services such as AWS S3 or alternatives, with secure access controlled by the backend.

6. **Notification Service:**
   Email and SMS notifications for confirmations, alerts, and status updates are sent using third-party APIs integrated into the backend.
7. **Admin Panel:**
   A separate dashboard for administrators to monitor, approve, and manage property owners, renters, and listings.

5. PROJECT PLANNING & SCHEDULING

5.1 PROJECT PLANNING

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High
Sprint-1	Registration	USN-3	As a user, I can register through Gmail.	2	Low
Sprint-1	Login	USN-4	As a user, I can log in with email and password	2	Medium
Sprint-2	Registration	USN-5	As a user, I can register through Facebook.	1	High
Sprint-2	Dashboard	USN-3	As a user, I can view my profile and saved listings after login.	2	Medium
Sprint-2	Property Search	USN-6	As a user, I can filter listings by price, location, and amenities.	4	High
Sprint-3	Booking Process	USN-7	As a user, I can book a property and get notified once approved.	3	High
Sprint-3	Messaging	USN-8	As a user, I can message the property owner before booking.	3	Medium
Sprint-4	Admin Approval	USN-10	As admin, I can approve/reject new owner	2	High

Project Tracker, Velocity & Burn down Chart:

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	2 Days	13 June 2025	15 June 2025	20	26 June 2025
Sprint-2	20	2 Days	16 June 2025	18 June 2025	20	-
Sprint-3	20	2 Days	19 June 2025	21 June 2025	20	-
Sprint-4	20	2 Days	22 June 2025	24 June 2025	20	-

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 PERFORMANCE TESTING

Project Overview:

Project Name: HouseHunt Rental Application
Project Description: A MERN based platform for renters to search, filter and book rental properties for owners to manage listings.
Project Version: 1.0.0

Testing Scope:

List of features to be tested

- User Registration (Email, Gmail, Facebook)
- Email Confirmation and OTP verification
- Login (with valid/invalid credentials)

- Property Search and Filter
- Booking Process
- Messaging between Renter and Owner
- Admin Approvals

List of User Stories or Requirements to be Tested

- USN-1 to USN-6 (Renter flow)
- USN-11 to USN-13 (Admin flow)
- USN-7 to USN-8 (Web user)

Test Cases:

Test Case Id	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	Email Registration	1. Go to register page 2. Enter email/password 3. Submit	User receives confirmation email	Email sent successfully	Pass
TC-002	Login with valid credentials	1. Go to login 2. Enter valid email/password 3. Click Login	User is redirected to dashboard	Redirected successfully	Pass
TC-003	Property search with filters	1. Login 2. Apply location/price filters 3. Click search	Filtered listings are shown	Listings match filters	Pass
TC-004	Booking a property	1. Select a listing 2. Click “Book” 3. Confirm details	Booking request is sent to owner	Owner received booking	Pass
TC-005	Chat feature between users	1. Click on message icon 2. Type message 3. Send	Message appears in real-time	Real-time message received	Pass

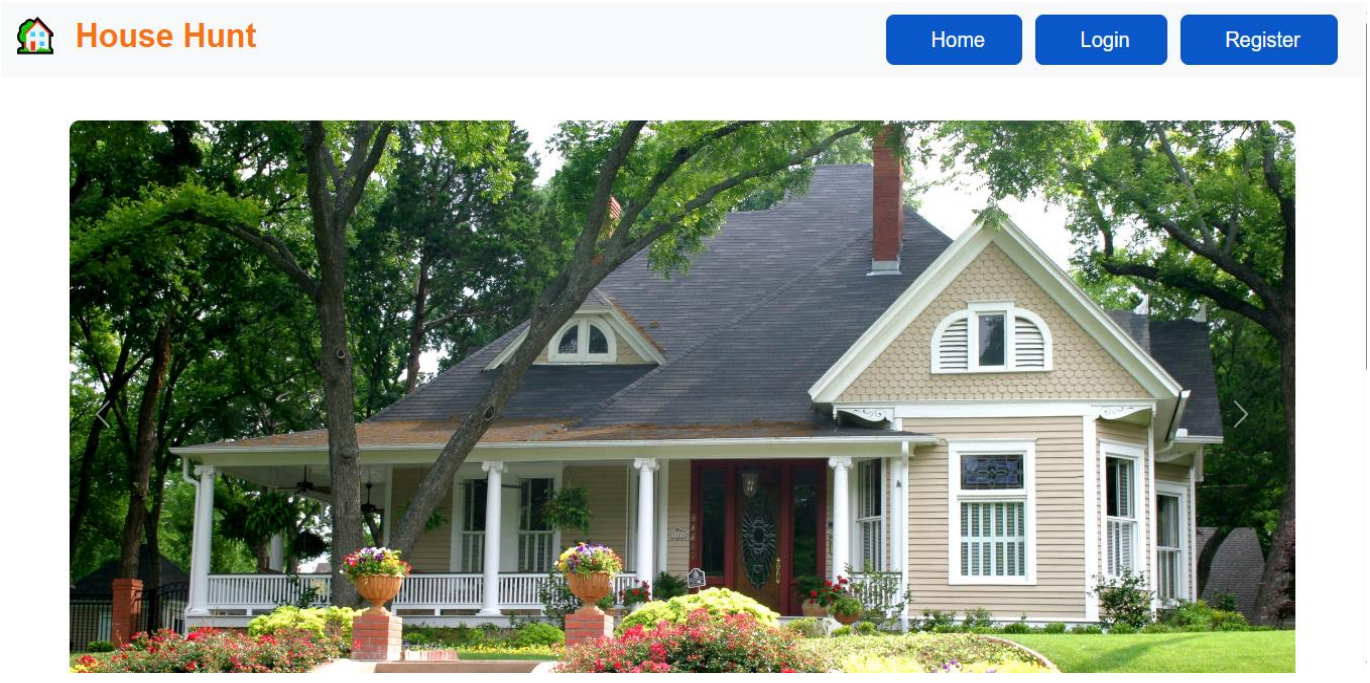
Bug Tracking:

Bug ID	Bug Description	Steps to reproduce	Severity	Status	Additional feedback
BG-001	OTP not received during registration	1. Register with email 2. Wait for OTP email	High	Open	Check mail server configuration
BG-002	Search not filtering by location	1. Apply location filter 2. Listings still show all data	Medium	In Progress	Possibly filter logic error on backend

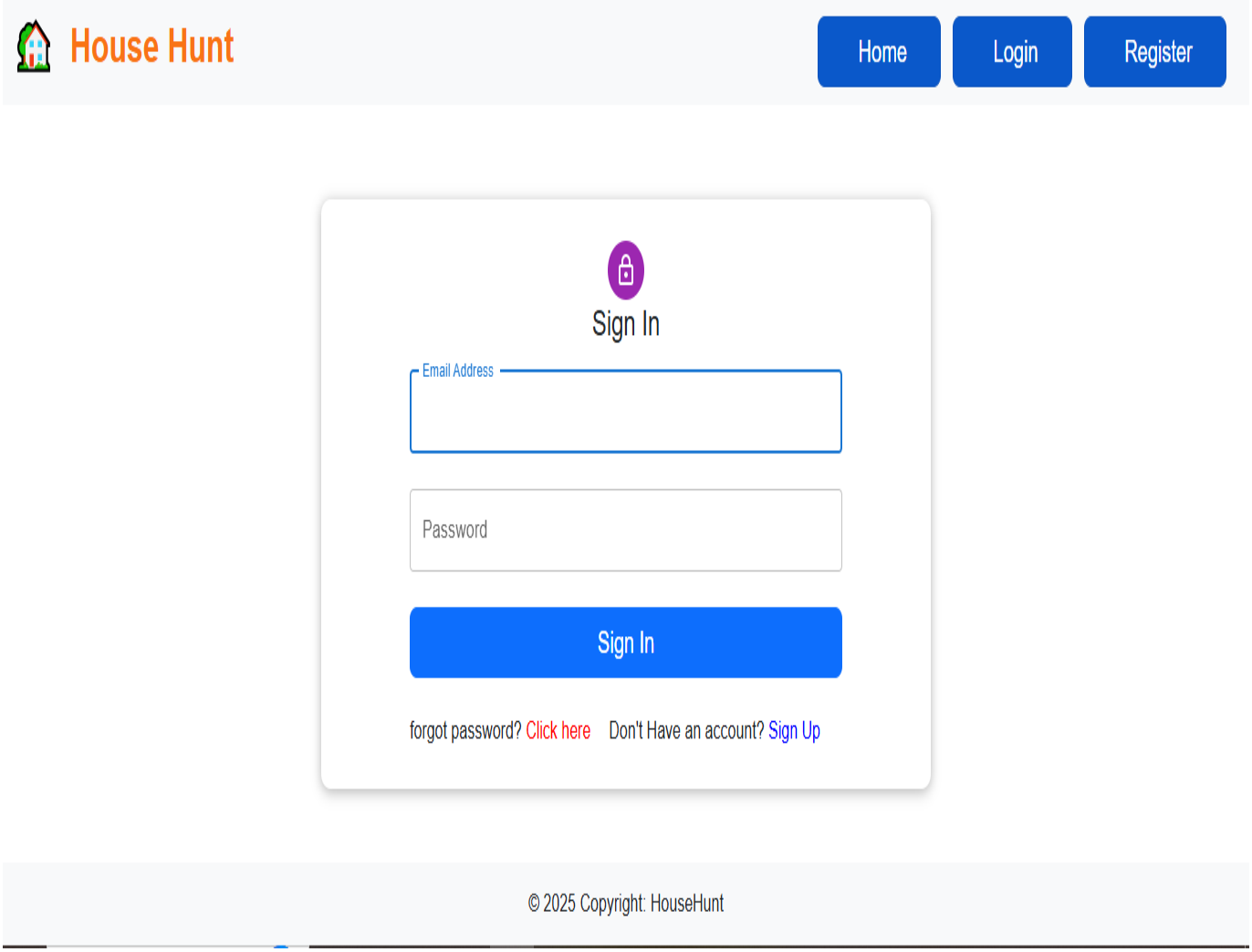
7. RESULTS

7.1 OUTPUT SCREENSHOTS

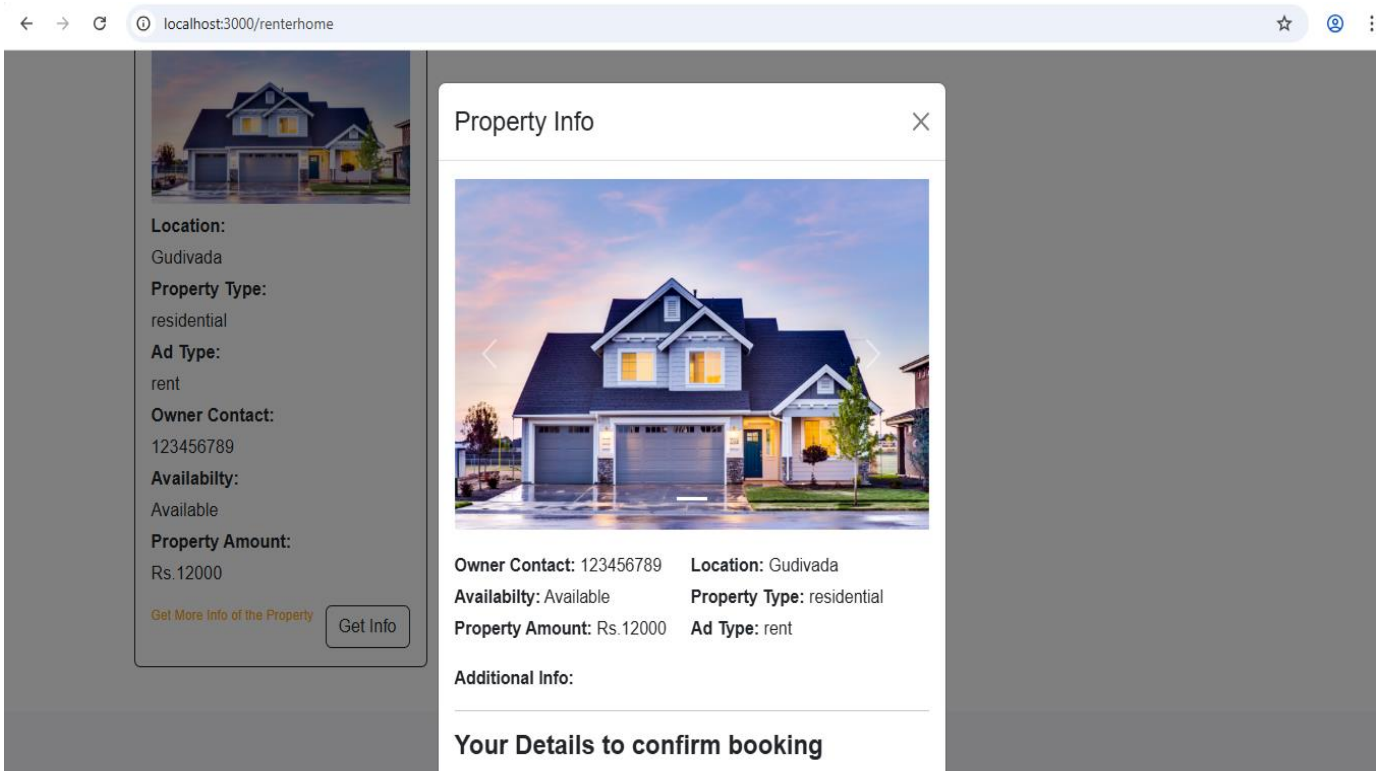
Home Page:



Login Page:



Renter Properties Page:



8. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. **Convenience and Time Saving**

- Users can browse multiple listings from one place without needing to physically visit every home.
- Filters and search tools help quickly narrow down options based on budget, location, size, etc.

2. Market Insights and Data Analysis

- Integrated data analytics can provide insights into market trends, pricing history, and neighborhood statistics.
- Helps users make informed decisions.

3. Automation and Alerts

- Automated alerts for new listings, price changes, or matching properties improve engagement.
- Saves users from manually checking updates.

4. Improved Communication

- Direct messaging between buyers and agents through the platform increases transparency and reduces delays.

5. Wider Reach for Sellers

- Sellers and agents get access to a larger audience, increasing the chances of a sale.

6. Visual and Virtual Tools

- Photos, videos, and virtual tours enhance the online house hunting experience.

DISADVANTAGES:

1. Data Accuracy Issues

- Listings may be outdated, duplicated, or inaccurate (e.g., incorrect prices or availability).
- Misleading photos can cause confusion or disappointment.

2. Privacy and Security Risks

- User data and communication could be at risk if the platform lacks robust security.

3. Tech Dependency

- Users without internet access or tech-savvy skills may struggle to use the system.
- Platform downtime can hinder user experience.

4. Overwhelming Choices

- Too many options may lead to decision fatigue or paralysis.
- Users may miss out on good options due to filter misuse.

5. Lack of Human Interaction

- Some buyers prefer in-person help and may feel disconnected from agents or sellers.

6. Cost of Development and Maintenance

- Building a reliable, scalable, and secure platform can be expensive and time-consuming.
- Continuous maintenance, data updates, and customer support are required.

9. CONCLUSION

The House Hunt Project provides an efficient and user-friendly platform that simplifies the process of searching for residential properties. By integrating key features such as customizable search filters, real-time listings, location-based insights, and user-agent communication tools, it enhances the overall home-buying experience.

While there are challenges such as data accuracy, technical limitations, and security concerns, these can be addressed with proper design, regular updates, and user education. Ultimately, the project demonstrates how technology can bridge the gap between buyers and sellers, making house hunting more accessible, informed, and convenient.

10. FUTURE SCOPE

1. AI-Powered Recommendations

- Integrate machine learning to suggest properties based on user behavior, preferences, and past searches.

2. Augmented and Virtual Reality (AR/VR)

- Enable virtual home tours using VR, allowing users to explore properties remotely in immersive 3D environments.

3. Integration with Financial Services

- Offer tools like mortgage calculators, loan pre-approvals, and partnerships with banks for a seamless home-buying process.

4. Smart Notifications and Chatbots

- Implement AI chatbots for instant customer support and smart notifications for price drops or new listings.

5. Blockchain for Secure Transactions

- Use blockchain to ensure secure, transparent, and tamper-proof property transactions and documentation.

6. Sustainability and Green Property Filters

- Add filters and badges for eco-friendly homes with solar panels, energy ratings, or green certifications.

7. Localized Market Insights

- Provide hyper-local data such as school ratings, crime rates, traffic conditions, and future development plans.

8. Mobile App Development

- Expand accessibility through dedicated mobile apps for Android and iOS with offline features and GPS support.

11. APPENDIX

Github Link: https://github.com/harinisayani/HouseHunt-FindingYour-Perfect-Rental-Home.git
Video Link:
https://drive.google.com/file/d/1VSjSST_P9MDcIjPt25emWQB7ApwzmoBR/view?usp=sharing