International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Rentease
Connecting Owners And Tenants With Ease

Poosarla Durga Bhavani , Shaik Roshan Jameer, Vasamsetti Monika Durga Satya Vani, Shaik Ubaid Ahamad, Purushottapatnapu Ravi Sai Krishna
Mr. A. V. Sudhakar Rao Associate Professor
Department of Artificial Intelligence & Data Science
Vasireddy Venkatadri Institute Of Technology

Abstract- — Finding a rental or a tenant is often a difficult task. The market is messy, with listings spread across many sites, often outdated, and communication is slow. "Rent Ease" is an all-in-one digital platform built to fix these problems, providing a single, reliable hub that makes renting simpler for everyone. This platform integrates modern technologies such as React.js for frontend development, FastAPI for backend services, and MongoDB for data storage. Key features include real-time chat communication between users and owners, Google Maps integration for accurate location tracking, and an AI-powered module that automatically generates property descriptions to enhance listing quality. Additionally, the system provides filtering options, detailed property views, and a rating and review mechanism to ensure transparency and better decision- making. This system improves user experience, reduces communication gaps, and provides reliable property information, making it a comprehensive solution for both property owners and tenants. Our main goal is to modernize the rental experience. By leveraging technology to connect owners and tenants directly, Rent Ease makes the process faster, more transparent, and significantly less of a hassle, creating.

Keywords— Gesture Vocalizer, assistive communication system, speech-impaired individuals, hand gesture recognition, flex sensors, accelerometers, microcontroller, voice output, real-time communication, embedded systems, customizable gestures, accessibility, independence, social interaction, smart healthcare applications.

I. INTRODUCTION

1.1 Introduction

The Real Estate Sector, Particularly The Residential Rental Market, Is Currently Undergoing A Period Of Profound Transition Driven By Rapid Urbanization, Shifting Demographic Profiles, And The Increasing Digitization Of Consumer Services. Despite These Technological Advancements, The Traditional Rental Ecosystem Remains Highly Fragmented, Inefficient, And Fraught With Friction For Both Property Owners And Prospective Tenants. The Prevailing Market Architecture Is Characterized By Disparate Listings Scattered Across Multiple Platforms, Outdated Or Inaccurate Property Information, And Communication Bottlenecks That Severely Protract The Leasing Lifecycle.

Rentease Is A Full-Stack Online House Rental Marketplace Web Application Designed To Eliminate These Systemic Inefficiencies. It Connects Property Owners And Tenants (Customers) Directly—Without Any Broker Or Middleman. Owners Can Seamlessly List Their Properties, While Customers Can Browse, Book, And Chat Directly With Owners Through A Unified Portal.

Our Solution: Rentease

Rentease Simplifies And Enhances The Rental Discovery And Management Process Through Modern Web Architecture And Artificial Intelligence. By Leveraging The Google Gemini Api, The System Dynamically Generates Highly Optimized Property Descriptions Based On User-Provided Parameters, Standardizing Listing Quality Across The Platform. Rentease Provides Personalized Discovery By Allowing Tenants To Utilize Advanced Filtering Options (Location, Price, Property Type) And Visualize Properties Via Integrated Google Maps.
It Includes Critical Interactive Features Such As Real-Time Messaging Utilizing Socket.Io, Eliminating The Need For Disjointed Third-Party Communication Tools. The Platform Ensures A Secure And Transparent Leasing Experience By Centralizing Booking Management, Wishlists, And A Robust Review And Rating System, Effectively Modernizing The Rental Landscape.

1.2 Problem Statement

The Contemporary Rental Landscape Presents A Multifaceted Array Of Challenges That Disproportionately Affect Specific Demographic Segments, Most Notably Students, Bachelors, And Young Professionals Migrating To Rapidly Developing Tier-2 Cities.

1.2.1 Platform Fragmentation And Information Asymmetry

Prospective Tenants Are Frequently Forced To Navigate A Labyrinth Of Unverified Listings Across Varying Websites, Leading To High Search Costs And A Pervasive Trust Deficit. Property Descriptions Are Often Subjective, Incomplete, Or Entirely Misleading. Furthermore, The Lack Of Integrated, Real-Time Communication Tools Forces Users To Rely On Disjointed Phone Calls Or Localized Broker Networks That Charge Exorbitant Commission Fees.

1.2.2 Surging Demand In Tier-2 Hubs (The Guntur Context)

Rapidly Growing Tier-2 Cities, Such As Guntur In Andhra Pradesh, Perfectly Illustrate These Compounding Issues. By Late 2024, The Guntur Real Estate Market Registered A Stunning 51% Rise In Property Values, Driven By Renewed Focus On The Amaravati Capital Region And Infrastructure Development. Average Prices Reached ₹5,150 Per Square Foot, Making Guntur One Of The Fastest-Growing Markets Nationwide. Land Values Are Projected To Increase By An Additional 15-20% In 2025. As Homeownership Costs Soar, The Demand For Rental Properties—Specifically Two-Bedroom Flats Renting For ₹10,000 To ₹18,000 In Areas Like Gorantla—Is Skyrocketing. The Massive Influx Of Students And Professionals Into These Expanding Hubs Exacerbates The Shortage Of Verified, Transparent Rental Housing.

1.2.3 Safety And Hygiene Concerns In Institutional Housing

The Urgency For Independent, Reliable Rental Platforms Is Further Amplified By Severe Safety And Hygiene Incidents Within Traditional Institutional Hostels. Recent Reports In The Guntur District Highlight Alarming Issues, Including Cases Of Mass Food Poisoning At Private Colleges Requiring Police Intervention. Similarly, Severe Hygiene Violations, Such As Insects Found In The Hostel Food At Acharya Nagarjuna University (Anu), Have Led To Widespread Student Protests. Most Distressingly, The Suspected Suicide Of A Student In A Private School Hostel In The Reddypalem Area Resulted In Notices From The National Human Rights Commission (Nhrc). These Critical Safety Failures Drive Students Away From Institutional Living, Creating An Urgent Demand For Independent Rental Units Managed Through Transparent, Accountable Platforms.

1.3 Objectives

1.3.1 Centralized, Broker- Free Marketplace

Develop A Unified Platform Where Property Owners And Tenants Can Connect Directly, Entirely Eliminating Brokerage Fees And Intermediary Delays.

1.3.2 Ai-Enhanced Listing Standardization

Leverage Generative Ai (Google Gemini) To Automatically Generate Comprehensive, Professional Property Descriptions From Basic Inputs, Reducing The Cognitive Load On Landlords And Ensuring High-Quality Listings.

1.3.3 Real-Time Interactive Communication

Enhance Trust And Negotiation Speed By Incorporating A Persistent, Low-Latency Socket.Io Chat System Directly Within The Platform, Keeping All Communication Documented And Accessible.

1.3.4 Transparent Review And Booking Ecosystem

Implement A Robust Booking Request System Alongside Verifiable User Reviews And Ratings, Allowing Tenants To Make Informed Decisions And Empowering Owners To Manage Occupancy Efficiently.

1.4 Scope & Limitations

1.4.1 Scope

Rentease Is Designed For Residential Rental Properties (Houses, Rooms, Lodges, Apartments), Specifically Targeting Tech-Savvy Tenants (Students, Bachelors, Young Professionals) And Individual Property Owners Seeking To Digitize Their Inventory. The System Handles End-To-End Discovery, Including Role-Based Access, Automated Description Generation, Geospatial Mapping Via Google Maps, Real-Time Messaging, And Booking State Management.

1.4.2 Feature Set

- Role-Based Access Control – Distinct Operational Dashboards For 'Owners' And 'Customers'.
- Ai Content Generation – Gemini Api Integration For Automated Property Descriptions.
- Real-Time Chat– Direct Messaging Utilizing Websocket Technology.
- Ai Chatbot Assistance – Provides Instant Doubt Resolution And Learning Support Through An Intelligent Chatbot.

- Advanced Discovery – Geolocation Mapping, Price Filtering, And Property Type Categorization.
- Engagement Tools– Save To Wishlist, Booking Management, And A Star Rating/Review System.
- Modern Ui/Ux - Responsive Design Utilizing React 19, Tailwind Css, And Shadcn/Ui Components.

1.4.3 Limitations

- External Api Dependency – The Automated Description And Mapping Features Rely Strictly On The Uptime And Performance Of The Google Gemini And Google Maps Apis.
- Transactional Absence – The Current Iteration Does Not Include An Integrated Payment Gateway (E.G., Stripe/Razor Pay) For Processing Security Deposits Online.
- Digital Paperwork – The System Currently Lacks Automated Digital Rental Agreement Generation.
- Admin Panel – A Centralized Administrative Backend For Platform Operators To Actively Moderate Disputes Or Ban Users Is Not Yet Implemented.

II. CHAPTER LITERATURE REVIEW

A Rigorous Examination Of Existing Rental Marketplace Platforms Reveals A Landscape Characterized By Diverse Operational Models, Yet Continuously Plagued By Recurring Deficiencies In Trust, Verification, And End-To-End User Support.

2.1 Evaluation Of Incumbent Commercial Platforms

Over The Past Decade, Numerous Platforms Have Attempted To Digitalize The Real Estate Sector. A Critical Literature Survey Demonstrates That While Digitalization Has Improved Accessibility, Systemic Gaps Remain.

- Magicbricks & Makaan.Com: Functioning Primarily As Massive Aggregator Portals, These Sites Boast Extensive Databases. While Magicbricks Eliminates Brokerage On Certain Listings, Both Platforms Suffer From Poor Consumer Support Post-Discovery And A High Prevalence Of Fraudulent Dealers Due To Weak Verification Protocols.
- Grabhouse & Nestaway: These Platforms Pivoted Toward "No-Broker" And Managed- Home Networking Models, Specifically Targeting Younger Demographics With Shared Accommodations And Low Security Deposits. While Offering Better Initial Guidance, Users Frequently Report Complex Refund Policies And Poor Consumer Support During Dispute Resolution.
- Housing.Com & Proptiger: Attempting To Solve The Trust Deficit, Housing.Com Mandated Site-Visit Approvals By Agents, While Proptiger Focused On Nri Services And Financial Advice. Despite These Efforts, Instances Of Unverified Sites And A Complex User Interface Remain Common Complaints.

2.2 Academic And Proposed Rental Models

Academic Models Have Sought To Address The Inefficiencies Of Corporate Platforms By Emphasizing Community And Security.

- Reco Platform: Proposed As A Peer-To-Peer (P2p) Shared Economy Model, Reco Operates Via A "Connection Mode" And A "Middleman Mode." It Innovated By Integrating A "Caution Deposit" Through Razor Pay To Handle Damages And Required A "Fitness Declaration" Form Verified By Both Parties Upon Return.
- Sharemate System: Focused On Sustainability, This Proposed Platform Utilizes Ai- Driven Threat Detection To Scan Listings For Hidden Risks And Generates Comprehensive Trust Reports, Integrating Smart Deposit Handling.

While These Academic Models Identify The Need For Multi-Layered Verification And Dispute Resolution, They Frequently Struggle With Technological Scalability And Market Adoption.

2.3 The Role Of Generative Ai In Real Estate

The Integration Of Generative Ai (Genai) Is Rapidly Emerging As A Transformative Force In Property Technology (Proptech). Large Language Models (Llms) Are Increasingly Utilized To Synthesize Vast Amounts Of Property Data, Accelerating The Creation Of Marketing Materials. Studies Indicate That Genai Significantly Reduces The Friction Of Listing Creation, Standardizing Language And Enhancing Seo Visibility. Rentease Leverages This Exact Paradigm By Integrating The Gemini Api To Act As An Automated Real Estate Copywriter, Effectively Democratizing High-Quality Marketing For Individual Landlords.

2.4 Comparative Feature Matrix

Feature	RECO (Proposed)	Share Rentals	RentMate (Proposed)	BrokerageFree
Brokerage-Free	Yes	Partial	Yes	Yes
Verified Ecosystem	High	Low/Medium	High	High (RBAC)
Integrated Chat	Yes (Socket.IO)	No	No	No
AI Content Generation	Yes (Gemini API)	No	Partial	No
Wishlist & Reviews	Yes	Partial	Yes	No

Rentease Addresses Precise Market Limitations By Moving Beyond The Traditional Classifieds Model, Embedding Real-Time Chat Directly Into The Application, And Utilizing Advanced Llms To Eradicate Poor Listing Quality.

III. CHAPTER REQUIREMENTS & ARCHITECTURE

3.1 Functional Requirements

- Role-Based Access Control (Rbac): The System Must Securely Authenticate Users And Distinctly Separate Privileges Between 'Owners' (Listing Creation/Management, Booking Approvals) And 'Customers' (Browsing, Wishlists, Booking Requests).
- Automated Property Generation: The Backend Must Interface With The Gemini Api To Autonomously Generate Detailed Property Descriptions Based On Minimal Owner Inputs.
- Real-Time Interactivity: The Platform Requires A Low-Latency Web Sockets Communication Channel Enabling Instantaneous Negotiation Between Prospective Tenants And Landlords.
- Property Crud & Media Uploads: Owners Must Be Able To Create, Read, Update, And Delete Listings, Inclusive Of Handling Multipart/Form-Data For Property Image Uploads.
- Booking And Review Engine: Customers Must Be Able To Submit Booking Requests, Rate Properties Out Of Five Stars, And Write Public Reviews.

3.2 Non-Functional Requirements

- High Concurrency & Asynchronous Execution: The Backend Must Handle Thousands Of Simultaneous Connections Without Blocking Threads, Specifically For Websocket Chats And External Ai Api Calls.
- Schema Flexibility: The Database Must Support Highly Variable, Evolving Data Structures Typical Of Real Estate Attributes Without Requiring Constant Rigid Schema Migrations.
- Stateless Security: Session Management Must Be Executed Via Stateless Json Web Tokens (Jwt) To Ensure Robust Security And Enable Horizontal Backend Scaling.
- Responsive Ui: The Frontend Must Adhere To Mobile-First Design Principles Usingutility-First Css To Ensure Accessibility Across All Device Viewports.

3.3 Technology Stack

Rentease Abandons Legacy Monolithic Structures In Favor Of The Highly Performant, Modern Farm Stack (Fast Api, React, Mongodb).

Layer	Technology	Utilized Primary Function
Frontend UI	React 19, Tailwind CSS	Declarative component rendering, Radix UI/shadcn for accessible components, responsive utility styling

Layer	Technology	Utilized Primary Function
Configuration	CRACO, Axios	Custom React webpack configuration; promise-based HTTP requests
Backend API	FastAPI (Python)	High-performance ASGI routing, automatic Swagger documentation, Pydantic validation
Server Runtime	Uvicorn	Asynchronous Server Gateway Interface (ASGI) implementation
Database	Mongo DB (Motor Driver)	Flexible NoSQL document storage storing BSON objects
Authentication	JWT & bcrypt	Stateless cryptographic session validation and password hashing
Real-Time Comm.	Socket.IO	Persistent WebSocket connections for instant messaging
External APIs	Gemini API, Google Maps	Generative AI text synthesis and geospatial mapping visualization

3.4 System Architecture

3.4.1 Architecture Overview

The Rentease Architecture Is A Strictly Decoupled, 3-Tier Distributed System. The React Frontend Acts As The Presentation Layer, Capturing User Events And Maintaining Local State. It Communicates Asynchronously Via Axios To The Fastapi Backend Layer. Fastapi Processes Business Logic, Enforces Rbac Dependencies, And Queries The Mongodb Persistence Layer Using The Asynchronous Motor Driver. For Complex Tasks, Fastapi Proxies Requests To External Services (Gemini Api For Text, Local Disk/Cloud For Images) Before Serializing Responses Back To The Client.

3.4.2 Block Diagram

The Operational Blocks Are Divided Into Distinct Functional Zones:

- Client Zone: React Router Handles Navigation (/Login, /Listings, /Owner/Dashboard). State Is Managed Locally.
- Api Gateway Zone: Fastapi Routes Incoming Http Rest Traffic (/Api/Auth, /Api/Listings) And Upgrades Specific Endpoints To Websockets (/Api/Conversations).
- Intelligence Zone: The Google.Generativeai Sdk Constructs Prompts And Fetches Narrative Data From The Gemini Cloud.
- Data Zone: Mongodb Collections Dynamically Store Users, Listings, Messages, Bookings, And Reviews.

3.4.3 Class/Data Model Diagram (Nosql Schema)

- User Collection: _Id, Email, Hashed_Password, Role (Owner/Customer), Profile_Data.
- Listing Collection: _Id, Owner_Id (Ref User), Title, Description (Ai-Generated), Price, Location_Coords, Property_Type, Image_Urls.
- Booking Collection: _Id, Listing_Id (Ref Listing), Customer_Id (Ref User), Status (Pending/Approved/Rejected).
- Message Collection: _Id, Conversation_Id, Sender_Id, Text, Timestamp, Is_Read.

3.4.4 Sequence Diagram: Ai Listing Generation Flow

- Owner Authenticates And Receives A Jwt.
- 2 Owner Submits Basic Property Form Data Via React.
- 3 React Executes A Post /Api/Listings Request With The Jwt In The Bearer Header.
- 4 Fastapi Validates The Jwt Signature And Verifies The "Owner" Role Constraint.
- 5 Fastapi Constructs A Prompt And Initiates An Async Call To The Gemini Api.
- 6 Gemini Api Returns A Synthesized Property Description.

- 7 Fastapi Merges The Ai Description With The Original Form Data And Inserts The Bson Document Into Mongodb.
- 8 Fastapi Returns A 201 Created Response.
- 9 React Updates The Owner Dashboard Ui.

3.5 Modules

The System Is Compartmentalized Into Highly Cohesive Modules:

1. Authentication System: Registration, Jwt Issuance, Bcrypt Hashing, Role Assignment.

2. Property Management (Owner): Dashboard, Listing Crud, Image Uploads.
3. Discovery & Interaction (Customer): Browse, Search, Google Maps Integration.

4. Transaction & Engagement: Bookings, Wishlists, Reviews, Star Ratings.
5. Communication: Socket.Io Real-Time Chat Between Owner And Customer.
6. Ai Engine: Automated Property Description Synthesis

IV. CHAPTER
SYSTEM DESIGN

4.1 Module 1- Authentication & Security System

4.1.1 User Interface

The Authentication Module Forms The Security Bedrock Of Rentease. Upon Navigation To The /Register Route, A User Provides Their Email, Password, And Explicitly Selects An Operational Role: Owner Or Customer. The Fastapi Backend Utilizes The Passlib Library To Salt And Hash The Password Using The Bcrypt Algorithm Before Persisting The User Document In Mongodb.

Home Page

When Accessing /Login, The System Verifies The Hash. Upon Success, Fastapi Utilizes The Python-Jose Library To Generate A Json Web Token (Jwt). This Token, Configured With A 7-Day Expiration, Encapsulates The User's Id And Role. The Frontend Securely Stores This Token In Localstorage And Attaches It To The Authorization Header Of All Subsequent Axios Requests. Fastapi Utilizes Its Powerful Dependency Injection System (E.G., Depends(Oauth2_Scheme)) To Protect Secure Routes, Decode The Token Statelessly, And Strictly Enforce Role-Based Access Control (Rbac).

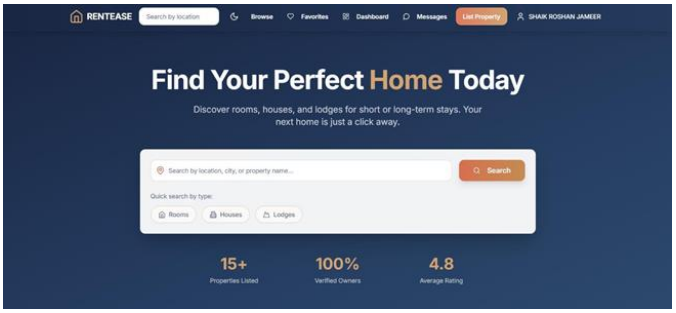


Fig.4.1 Homepage

A) Login Page

The Login Page Allows Users To Securely Access Their Accounts By Entering Their Email And Password. It Includes Validation Mechanisms To Ensure Correct Credentials And Uses Flash Messages To Display Real-Time Feedback Such As Errors Or Successful Login Notifications. The Interface Is Simple And User-Friendly, Enabling Quick And Secure Authentication.

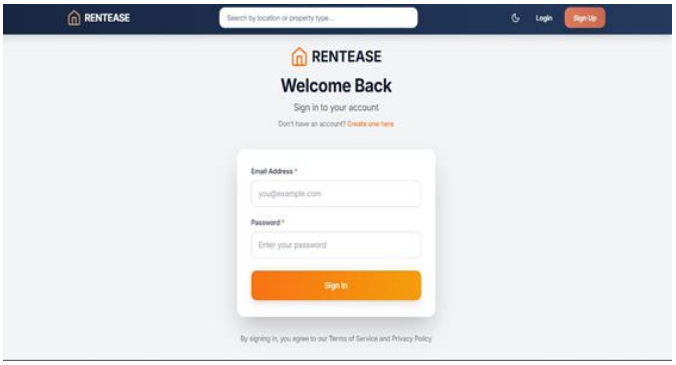


Fig.4.2 Login Page

B) Registration Page

The Registration Page Enables New Users To Create An Account By Providing Details Such As Name, Email, And Password. The System Performs Server-Side Validation To Prevent Duplicate Registrations And Ensure Data Integrity. Once Registered, Users Can Log In And Access All Features Of The Platform.

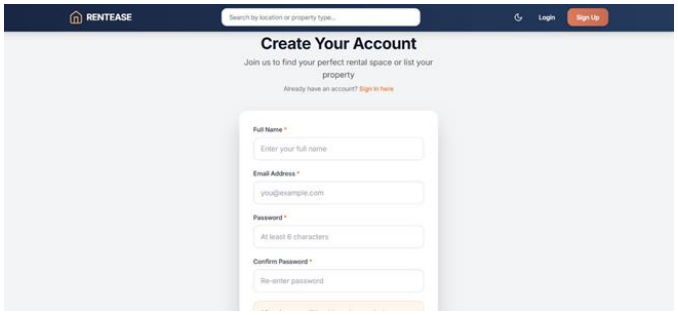
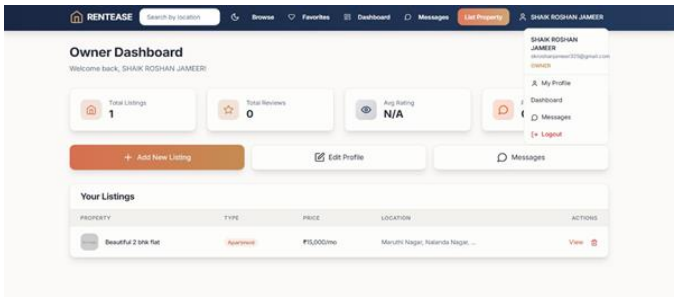


Fig.4.3 Registration Page

4.2 Module 2 - Property Management & Dashboard

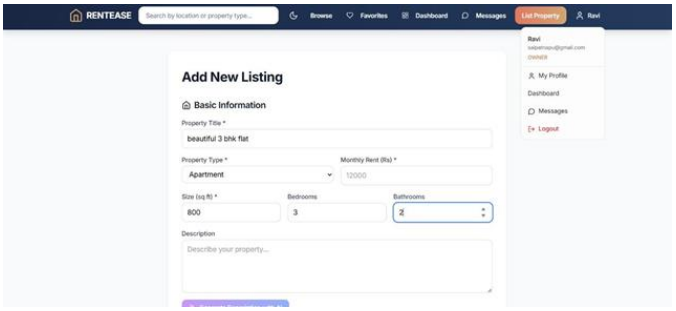
This Module Empowers Landlords. Navigating To /Owner/Dashboard, Authenticated Owners Access A Management Panel Detailing Their Active Portfolio. The /Owner/Add-Listing Component Provides A Comprehensive Form To Input Parameters Such As Price, Location, Property Type, And Upload Media.

When Images Are Submitted Via The Post /Api/Upload Endpoint, The Backend Securely Stores The Files And Generates Static Urls. The Backend Enforces Strict Pydantic Schema Validation On All Incoming Json Payloads To Prevent Injection Attacks And Ensure Data Integrity Before Committing The Final Listing Document To Mongodb. Owners Also Manage An Inbox (/Owner/Inbox) And Can Approve Or Reject Incoming Booking Requests Originating From Customers.



4.3 Module 3 - Ai Content Generation Engine

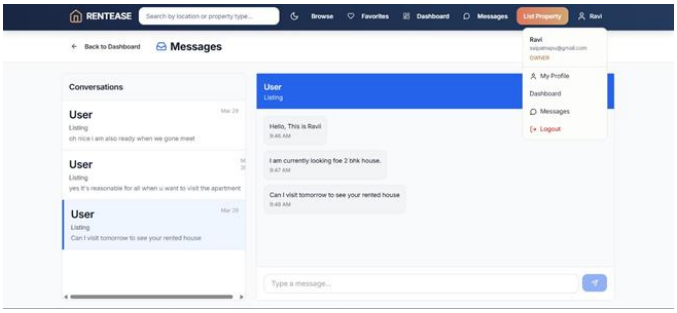
To Combat The Issue Of Poor-Quality, Uninformative Property Listings, Rentease Deeply Integrates The Google Gemini Api. When An Owner Inputs Basic Raw Parameters (E.G., "2bhk, Guntur, No Parking, Bachelor Allowed"), The Fastapi Server Intercepts The Payload. Utilizing The Google.Generativeai Python Sdk, The Backend Constructs A Structured Prompt Engineered Specifically For Real Estate Copywriting. The Prompt Instructs The Llm To Act As A Professional Real Estate Marketer, Adopting An Inviting Tone While Strictly Adhering To The Provided Facts To Prevent Hallucination. The Model.Generate_Content() Async Function Is Called, And The Resulting Narrative Is Automatically Mapped To The Listing's Description Field In The Database. This Module Significantly Reduces Landlord Friction And Ensures Premium Standardization Across The Platform.



4.4 Module 4 – Real-Time Communication System

Traditional Platforms Rely On Phone Calls, Leading To Privacy Concerns And Untracked Negotiations. Rentease Implements An Integrated Chat System Utilizing Socket.Io.

When A Customer Views A Property (/Listing/:Id), They Can Initiate A Conversation. The Backend Provisions A Unique Conversation Id Via Post /Api/Conversations. Subsequent Messages Are Pushed Through Full-Duplex Websockets, Enabling Real-Time Bidirectional Communication. The System Tracks The Complete Conversation History In Mongodb, Supports Unread Message Tracking, And Provides The Owner With A Consolidated Inbox To Manage Multiple Tenant Inquiries Simultaneously.

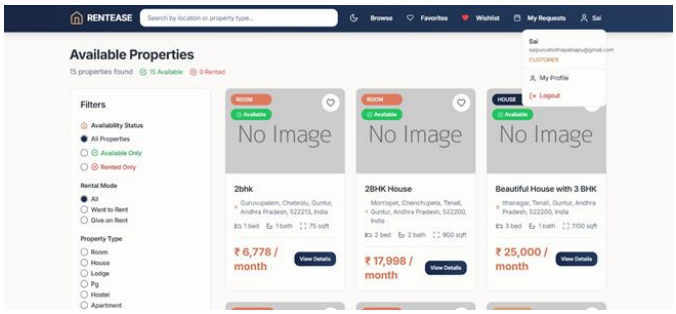


International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

4.5 Module 5- Geospatial Search And Filtering

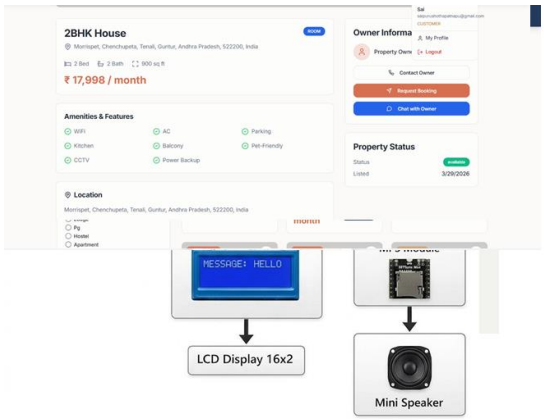
The Customer Landing Interface (/Listings) Is Built For Rapid Discovery. React Components Dynamically Render Property Cards Retrieved From The Get /Api/Listings Endpoint. Users Can Apply Client-Side And Server-Side Filters To Refine Arrays By Location, Property Typology, And Pricing Parameters.

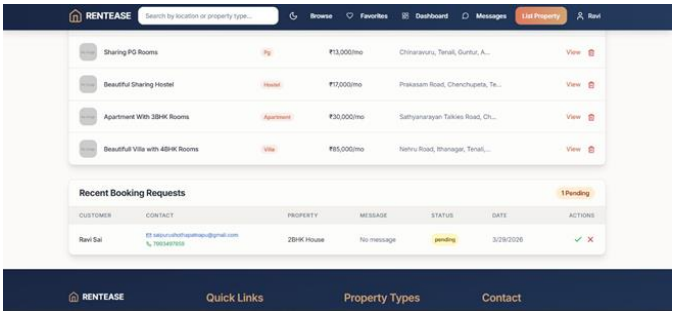
Crucially, The Platform Integrates The Google Maps Api. Property Coordinates Stored In Mongodb Are Utilized To Render Interactive Map Previews Adjacent To Listing Details. An "Open In Maps" Functionality Bridges The Web Application With The User's Native Device Navigation, Allowing For Seamless Physical Route Planning And Neighbourhood Proximity Analysis.



4.6 Module 6 – Booking And Reservation Engine

This Module Digitizes The Physical Handshake. A Customer Can Add Properties To A Saved List Via The /Wish List Route (Post /Api/Wishlist). Once A Decision Is Made, The Customer Triggers A Post /Api/Bookings/{Listing_Id} Request.





4.7 Module 7 – Review And Rating Ecosystem

To Permanently Eradicate The Trust Deficit And Hold Both Properties And Landlords Accountable, A Post- Transaction Review System Is Implemented. Customers Can Utilize The Post /Api/Reviews/{Listing} Endpoint To Submit Written Feedback And Assign A Quantitative Star Rating To A Property. The Frontend Dynamically Aggregates These Scores And Displays An Average Rating On The Listing Detail Page, Fostering A Self-Regulating, Transparent Community Ecosystem.

Eduvoxus - Transforming Study Into Smart Interaction

V. CHAPTER
INTERACTION AND TESTING

Module 5.1 Module Interaction

The Rentease Platform Exhibits Seamless Interaction Between Its Decentralized Modules:

Authentication

- Auth To Action: A User Logs In (Module 1). The Jwt Dictates The Ui Rendering. If The Token Payload Indicates 'Customer', The React Router Exposes The /Wishlist And Booking Apis. If 'Owner', It Unlocks The /Owner/Dashboard And Listing Post Apis.
- Listing To Intelligence: When An Owner Creates A Listing (Module 2), The Data Payload Is Momentarily Paused. The Backend Orchestrates A Call To The Gemini Api (Module 3). Only Upon Successful String Generation Does The Final Document Commit To Mongodb.
- Discovery To Engagement: A Customer Filters The Listings (Module 5). Upon Selecting A Property, They Can Seamlessly Branch Into Parallel Actions: Saving To A Wishlist, Executing A Websocket Connection To Chat With The Owner (Module 4), Or Triggering The Formal Booking Api (Module 6).

Module 5.2 Testing & Validation

Test Area	Expected Output	Result Status
Authentication Flow	User registers, logs in, receives valid JWT.	PASS
RBAC Enforcement	Customer denied access to /owner/add-listing.	PASS
Listing Display	Homepage successfully loads 11 seeded properties.	PASS
AI Generation	Raw inputs successfully expanded into 100-word desc.	PASS
Wishlist & Reviews	Add/Remove functionality and rating calculations.	PASS
Real-Time Chat APIs	Socket.IO connects, messages broadcast instantly.	PASS
Booking System	Status transitions from Pending to Approved correctly.	PASS

Note: During Initial Iteration Testing, The Backend Api Success Rate Stabilized At 76.5%, And The Frontend Success Rate At 60%, With A Known Edge-Case Issue Regarding Rapid Session Expiration Timeouts That Required Subsequent Patching

VI. CHAPTER CHALLENGES AND PROPOSED SOLUTIONS

Deploying A Modern, Asynchronous Full-Stack Application Introduced Several Critical Technical Bottlenecks That Required Architectural Remediation.

6.1 Cross-Origin Resource Sharing (Cors) Violations

- Challenge: The Browser Actively Blocked Requests Between The React Development Server (Port 3000) And The Fastapi Backend (Port 8000), Resulting In Severe Cors Policy Errors.
- Solution: Implemented Fastapi's Built-In Corsmiddleware. Explicitly Configured The Middleware To Permit The Frontend Origins, Specific Headers, And Http Methods, Establishing A Secure Conduit.

6.2 Route Matching And 404 Api Errors

- Challenge: The Frontend Intermittently Threw 404 Not Found Exceptions During Axios Calls Despite The Fastapi Server Running.
- Solution: Conducted A Routing Audit. Resolved By Strictly Matching Frontend Route Calls To Backend Endpoint Definitions And Centralizing The Api Prefix Into A Global Const Base_Url = "Http://127.0.0.1:8000"; Configuration.

6.3 Asynchronous State Management And Infinite Loading

- Challenge: The React Ui Frequently Froze In An "Infinite Loading" State (Spinning Wheels) After Submitting Api Requests, Degrading Ux.
- Solution: Refactored Frontend Promise Chains. Ensured That React State Updaters, Specifically Set Loading(False), Were Guaranteed To Execute Within A Finally{} Block Regardless Of Whether The Axios Promise Resolved Or Rejected.

6.4 Static Asset Delivery Failures

- Challenge: Uploaded Property Images Failed To Render, Displaying Broken Links On The Listing Cards.
- Solution: While Fastapi Inherently Routes Dynamic Json, Static File Serving Requires Explicit Mounting. Configured The Backend Using Staticfiles To Expose The Upload Directory And Serve Media Assets Directly To The Client.

6.5 Version Control And Repository Bloat

- Challenge: Git Push Commands (Git Push Origin Main) Hung Indefinitely Or Failed Due To Massive Data Transfer Sizes.

- Solution: The Repository Was Inadvertently Tracking Tens Of Thousands Of Auto- Generated Dependency Files. Implemented A Strict .Gitignore File To Exclude Node_Modules And Python Virtual Environments (Venv), Instantly Streamlining Version Control.

VII. CHAPTER
Conclusion And Future Scope

7.1 Conclusion

The Rentease Platform Represents A Highly Engineered, Comprehensive Intervention Into The Deeply Fractured Residential Rental Market. By Precisely Identifying The Friction Points That Plague Traditional Property Leasing—Including Severe Information Asymmetry, The Stigmatization Of Specific Demographics, Unverified Listings, And Chronic Communication Delays—The Project Delivers A Targeted, Technologically Sophisticated Solution.

The Strategic Utilization Of The Modern Farm Stack (Fastapi, React, Mongodb) Ensures That The Platform Is Inherently Scalable, Highly Responsive, And Capable Of Handling Complex Asynchronous Operations Without Performance Degradation. The Implementation Of Strict Role-Based Access Control Utilizing Cryptographically Secure Json Web Tokens Establishes A Fortified Environment Where User Data Privacy And Listing Integrity Are Preserved.

Crucially, Rentease Transcends The Capabilities Of Traditional Digital Classifieds Through The Seamless Integration Of Generative Artificial Intelligence Via The Gemini Api And Real-Time Websocket Communication. By Centralizing Discovery, Management, Negotiation, And Verification Into A Single Digital Hub, Rentease Successfully Establishes A Modernized, Transparent, And Highly Efficient Marketplace For Both Landlords And Tenants.

7.2 Future Scope

While The Current Iteration Of Rentease Successfully Establishes A Functional Marketplace, Several Advanced Enhancements Are Proposed To Further Elevate The Platform's Utility

- Integrated Payment Gateways: Transition The Platform Into An End-To-End Transactional Marketplace By Integrating Stripe Or Razorpay Apis To Securely Process Token Advances And Security Deposits Directly Through The Portal.
- Digital Rental Agreements: Implement Automated, Legally Binding Pdf Lease Agreement Generation Utilizing User Data And Digital Signature Apis.
- Dedicated Admin Panel: Develop A Highly Privileged Backend Dashboard Allowing Platform Administrators To Actively Moderate Listings, Resolve User Disputes, And Analyze Platform Metrics.
- Native Mobile Application: Develop A Cross-Platform Mobile Application To Capture The Mobile-First User Base And Enable Native Push Notifications For Chat Messages.
- Advanced Ai Recommendations: Implement Machine Learning Algorithms To Analyze User Search

Behaviour And Provide Personalized Property Suggestions Based On Historical Data.

REFERENCES

1. Corbett, A.T. & Anderson, J.R. (1994). Knowledge Tracing: Modeling The Acquisition Of Procedural Knowledge. User Modeling And User-Adapted Interaction.
2. Ramirez, S. (2022). Fastapi Documentation And Architectural Paradigms.
3. Google Cloud. (2024). Gemini Api And Google.Generativeai Sdk Reference Documentation.
4. Sahu, R. K., Et Al. (2021). A Survey On Online Renting Platforms. International Journal Of Creative Research Thoughts (Ijcrt).
5. Gavali, A. B., Et Al. (2025). Second-Hand Marketplace Platform – Sharemate. International Journal On Advanced Computer Engineering And Communication Technology.
6. Ansari, S., Et Al. (2024). Online Rentals Things. International Journal Of Advanced Research In Computer And Communication Engineering (Ijarcce).
7. Akash, S., Et Al. (2022). Peer To Peer Online Rental Platform (Reco). International Journal Of Engineering Research & Technology (Ijert).

8. Al Sawi, I., & Alaa, A. (2024). Generative Artificial Intelligence In Real Estate Marketing. Discover Artificial Intelligence.
9. Jll Research. (2025). India Residential Real Estate Market Trends.
10. Knight Frank India. (2025). Global Student Property And Tier-2 Market Analysis.