

Haven AI

(Real Estate Support Triage Agent)

Manufacturing Operational Insights Report

Course Name: Agentic AI

Institution Name: Medicaps University – Datagami Skill Based Course

Sr no	Student Name	Enrolment Number
1	Rudra Sthapak	EN22CS301828
2	Rajvardhan Singh Rathore	EN22CS301786
3	Rupendra Singh Tomar	EN22CS301833
4	Ronak Bhawsar	EN22CS301824
5	Raj Kumawat	EN22CS301779
6	Priyanka Kumari	EN22CS301758

Group Name: 04D7

Project Number: AAI-27

Industry Mentor Name: Aashruti Shah

University Mentor Name: Prof. Ajeet Singh Rajput

Academic Year: 2025-2026

Table of Contents

Chapter 1	Problem Statement & Objectives	3-4
1.	Problem Statement	3
2.	Project Objectives	3
3.	Scope of the Project	4
Chapter 2	Proposed Solution	5-7
1.	Key Features	5
2.	Overall Architecture / Workflow	6
3.	Tools And Technologies Used	7
Chapter 3	Results and Outputs	8-12
1.	Screenshots / Outputs	8
2.	Reports / Dashboards / Models	10
3.	Key Outcomes	12
Chapter 4	Conclusion	13
Chapter 5	Future Scope & Enhancements	14

Problem Statement & Objectives

1. Problem Statement

The real estate industry generates a large volume of customer queries related to property search, pricing, availability, amenities, and booking procedures. Traditional real estate platforms rely heavily on manual filtering systems and human customer support agents to resolve these queries. This approach results in delayed responses, inefficient query handling, and increased operational costs.

Users often struggle to find relevant properties because keyword-based search systems fail to understand natural language queries such as "2BHK under 50 lakhs in Indore near Vijay Nagar." Conventional filtering systems require users to manually apply multiple filters, which affects user experience and engagement.

Additionally, real estate businesses require a secure authentication system to manage users, prevent misuse, and protect sensitive data. OTP-based verification systems must ensure expiration and security compliance.

There is a strong need for an intelligent system capable of:

- Understanding natural language queries
- Extracting meaningful entities such as price, location, property type
- Performing dynamic filtered database searches
- Providing instant contextual responses
- Ensuring secure user authentication

This project aims to address these challenges through an AI-powered real estate support and triage agent.

2. Project Objectives

The primary objectives of this project are:

1. To design a scalable backend architecture for a real estate support system.
2. To implement AI-driven intent recognition and entity extraction.
3. To design and structure NoSQL collections for optimized property search.
4. To integrate asynchronous database operations for improved performance.
5. To implement OTP-based authentication with expiration policies.
6. To ensure modular, maintainable, and extensible system design.
7. To provide a responsive and efficient user interaction workflow.
8. To follow industry-standard architectural principles for HLD and LLD.

3. Scope of the Project

The scope of the project includes the design and implementation of an AI-powered real estate query handling system with the following boundaries:

- Development of RESTful APIs using FastAPI.
- Implementation of secure login and OTP verification.
- Integration of an AI orchestration layer for processing user queries.
- MongoDB database integration for property and user storage.
- Implementation of filtered property search logic.
- Context-aware response generation.

The project does not currently include:

- Online payment gateway integration.
- Live property booking system.
- Advanced recommendation engine using machine learning.
- Full mobile application deployment.

The scope is limited to demonstrating a functional AI-enabled backend architecture for intelligent property search and support automation.

Proposed Solution

The proposed solution is a multi-layered intelligent support system built using modern backend technologies and AI orchestration frameworks.

The architecture consists of four primary layers:

1. User Interface Layer

The system provides a chat-based web interface through which users can interact with the system, enter queries, and receive property recommendations.

2. Backend Services Layer

This layer is responsible for:

- API request handling
- Authentication verification
- Business logic execution
- Data validation

FastAPI is used due to its high performance and asynchronous capabilities.

3. AI Orchestration Layer

This layer processes natural language input using LLM integration. It performs:

- Intent detection
- Entity extraction
- Context management
- Tool calling for database queries

LangChain is used to orchestrate LLM responses and integrate database search tools.

4. Database Layer

MongoDB is used as the primary NoSQL database for storing:

- User information
- Property listings
- OTP verification records

This layered approach ensures scalability, separation of concerns, and maintainability.

1. Key Features

- AI-powered natural language understanding
- Real-time property search
- Secure OTP-based authentication
- JWT-based session management
- Asynchronous database access
- Modular API structure
- MongoDB NoSQL storage
- TTL indexing for OTP expiration
- Entity-based filtered search
- Scalable backend design

2. Overall Architecture / Workflow

The system workflow operates in the following sequence:

1. The user sends a query via the chat interface.
2. The backend API receives and validates the request.
3. If authentication is required, OTP verification is performed.
4. The query is forwarded to the AI orchestration module.
5. The AI module extracts entities such as:
 - Location
 - Budget
 - Bedrooms
 - Property Type
6. A filtered database query is generated dynamically.
7. MongoDB returns matching property documents.
8. The response is formatted and returned to the user interface.

The architecture follows a stateless RESTful approach, ensuring scalability and distributed deployment readiness.

3. Tools & Technologies Used

Backend Framework

- Python
- FastAPI

Database

- MongoDB (NoSQL)
- Motor (Async driver)

AI Integration

- LangChain
- LLM API (for intent & entity processing)

Authentication

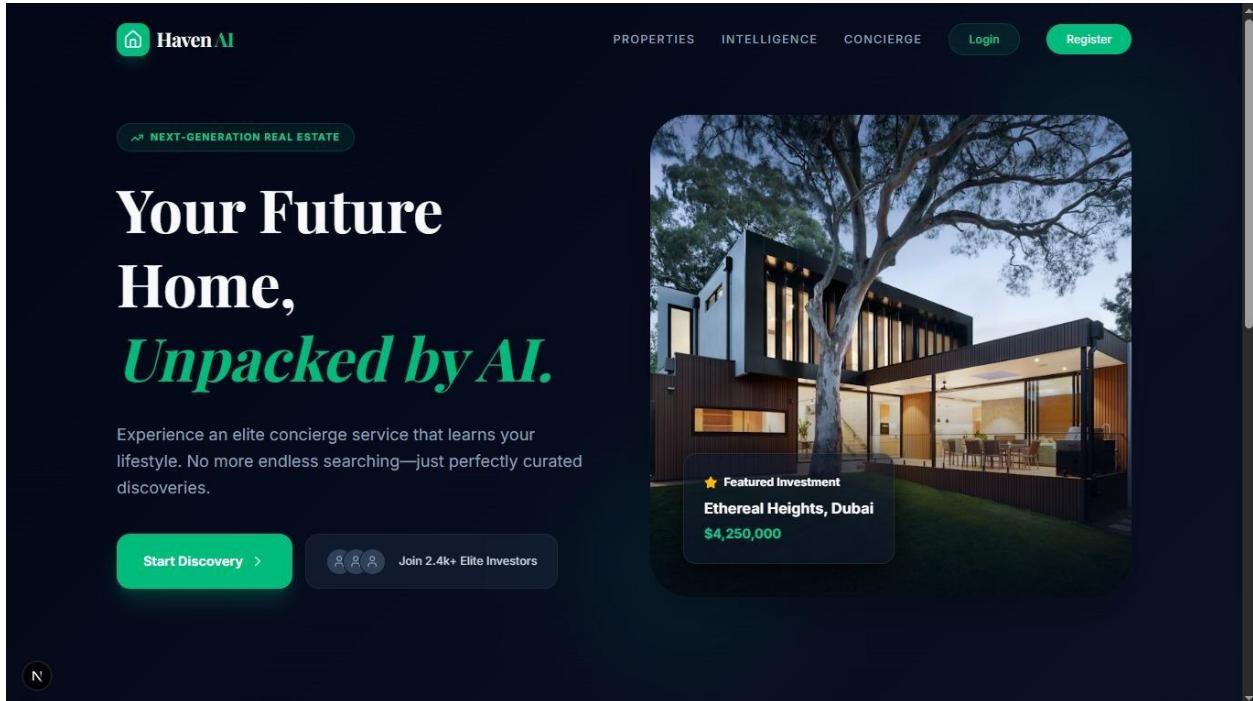
- OTP Verification
- JWT Token Handling

Development & Version Control

- GitHub
- Postman (API testing)

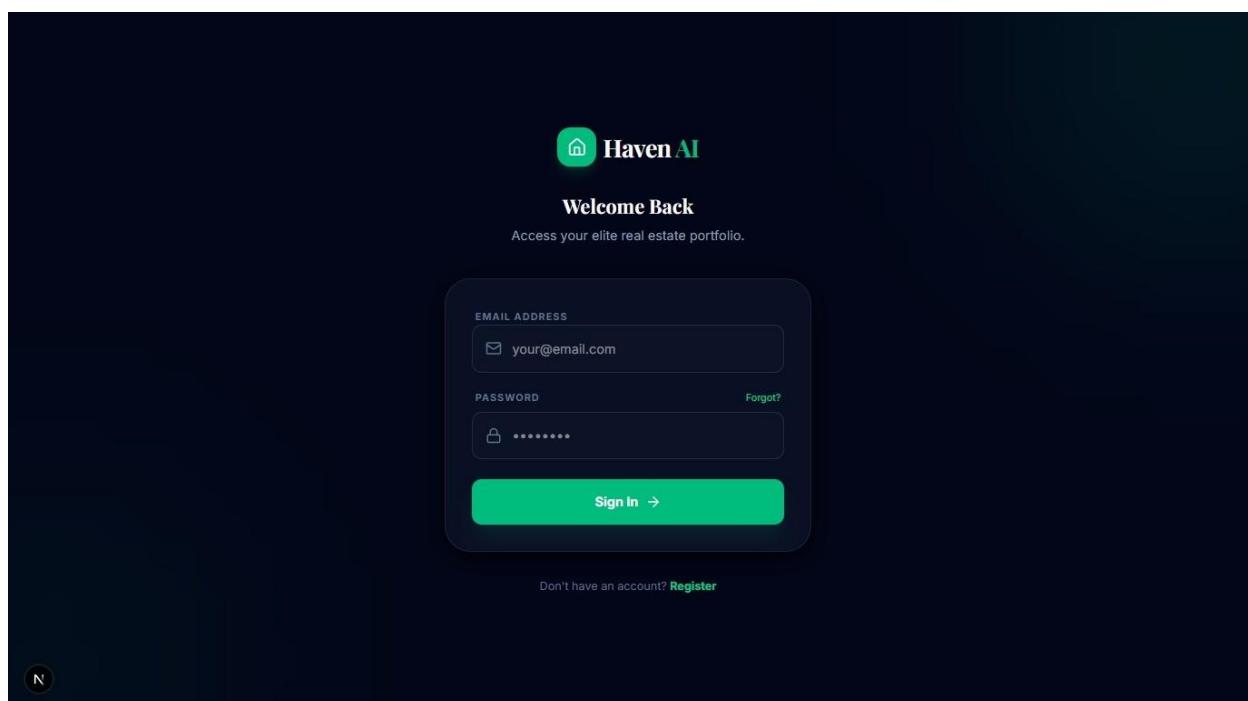
Results And Outputs

1. Screenshots / Outputs



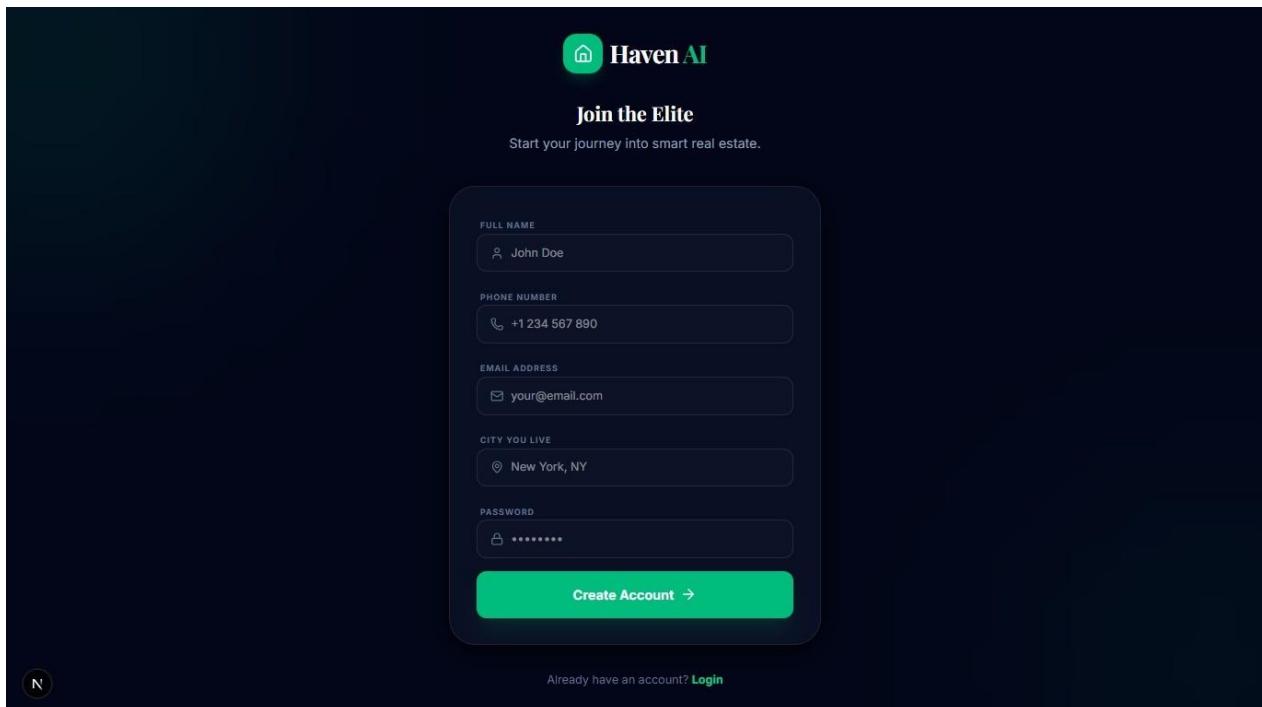
The screenshot shows the homepage of Haven AI. At the top, there's a navigation bar with the logo 'Haven AI' and links for 'PROPERTIES', 'INTELLIGENCE', 'CONCIERGE', 'Login', and 'Register'. Below the header, a large banner features the text 'Your Future Home, Unpacked by AI.' in white and teal. A sub-header 'NEXT-GENERATION REAL ESTATE' is visible above the main text. To the right of the text is a large image of a modern, two-story house with large glass windows and doors, set against a backdrop of trees. A callout box on the right side of the image reads: 'Featured Investment Ethereal Heights, Dubai \$4,250,000'. At the bottom left, there's a green button labeled 'Start Discovery >' and a dark button labeled 'Join 2.4k+ Elite Investors'.

Home Page

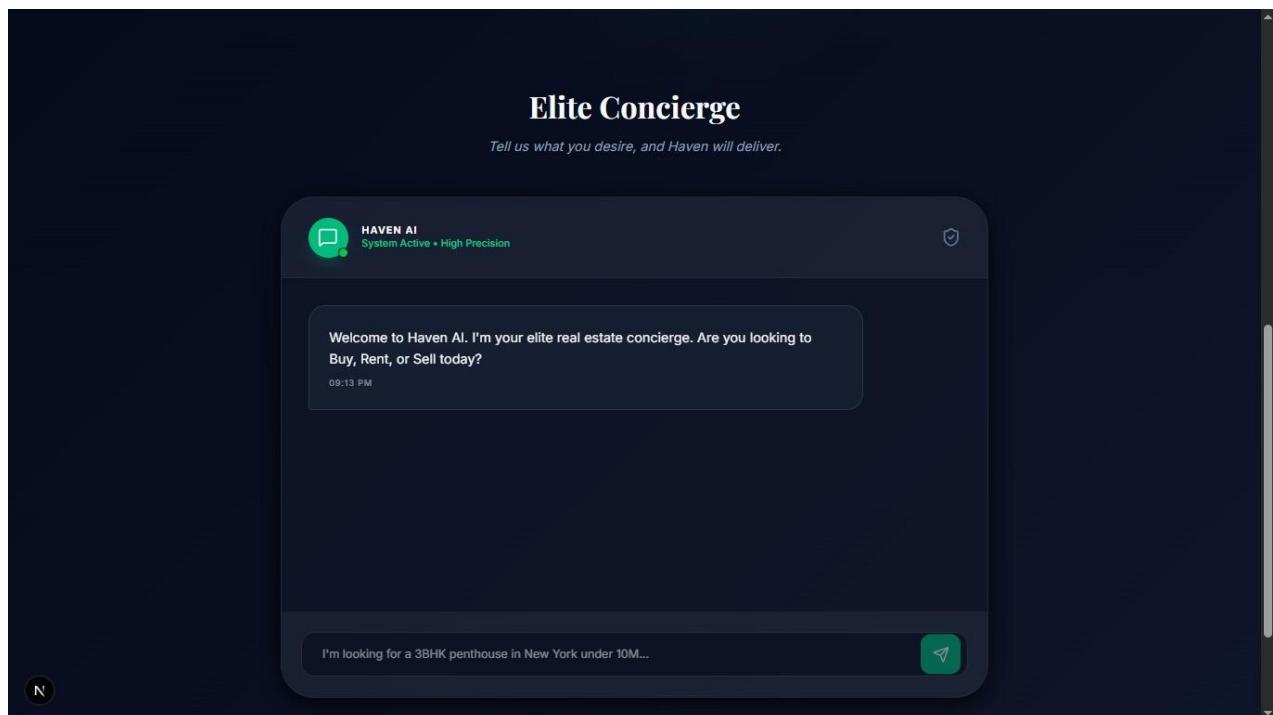


The screenshot shows the login page of Haven AI. It features a dark background with the 'Haven AI' logo at the top. Below it, the text 'Welcome Back' is displayed, followed by the sub-instruction 'Access your elite real estate portfolio.'. A central form contains fields for 'EMAIL ADDRESS' (with placeholder 'your@email.com') and 'PASSWORD' (with placeholder '*****'). To the right of the password field is a 'Forgot?' link. At the bottom of the form is a green 'Sign In →' button. Below the form, a small note says 'Don't have an account? [Register](#)'.

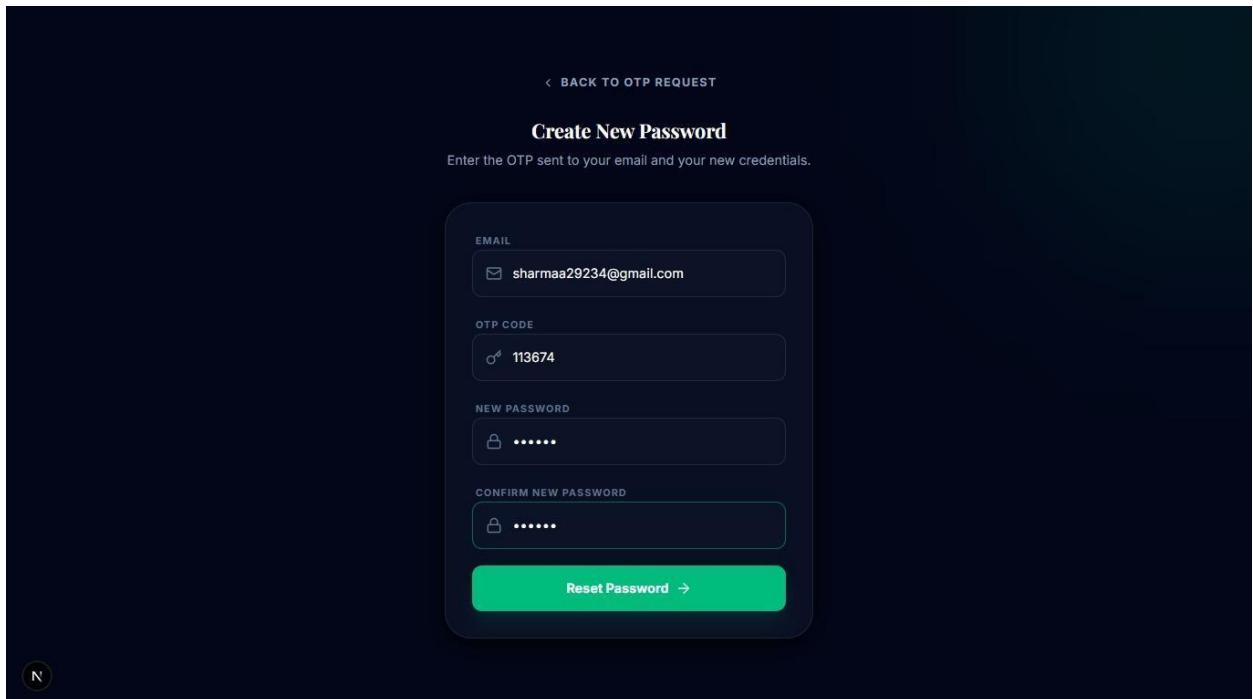
Login Page



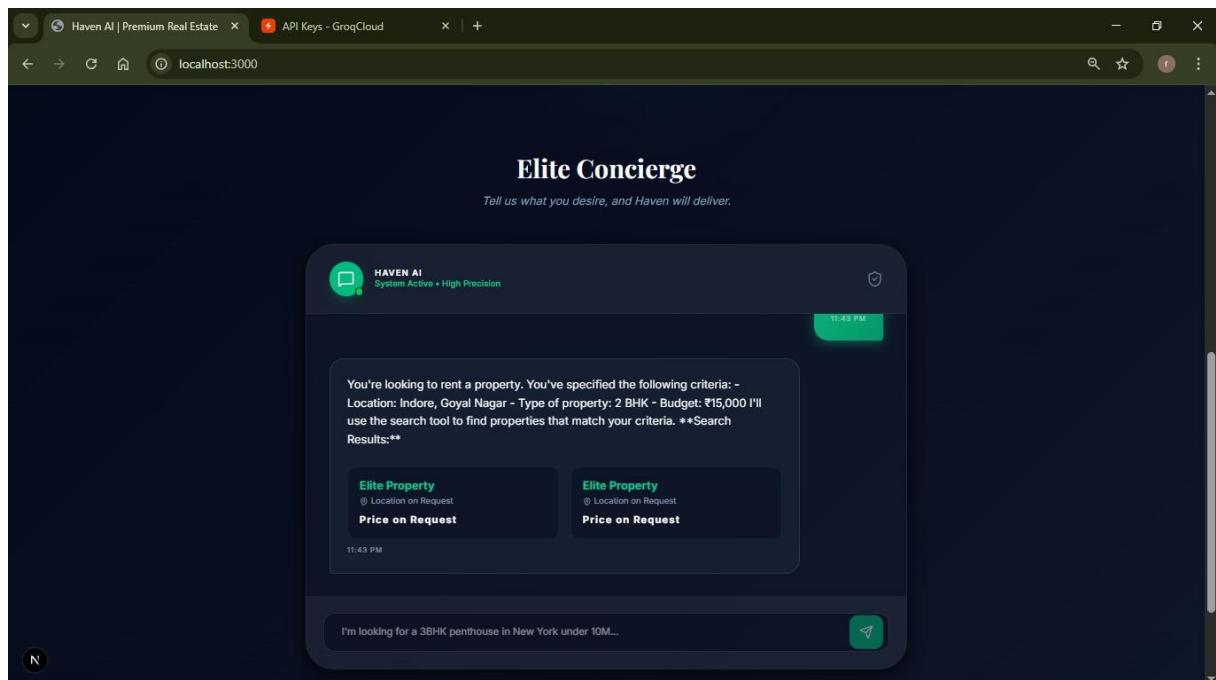
Sign up Page



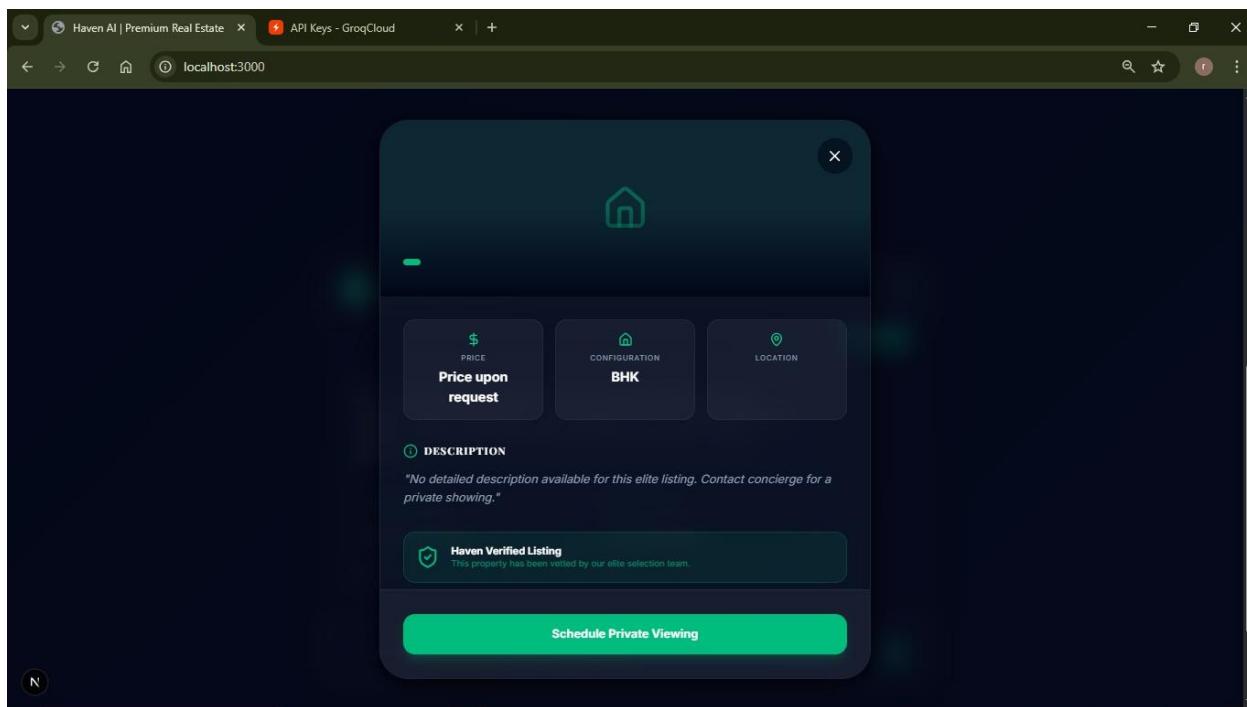
Chat Bot Agent



Change Password



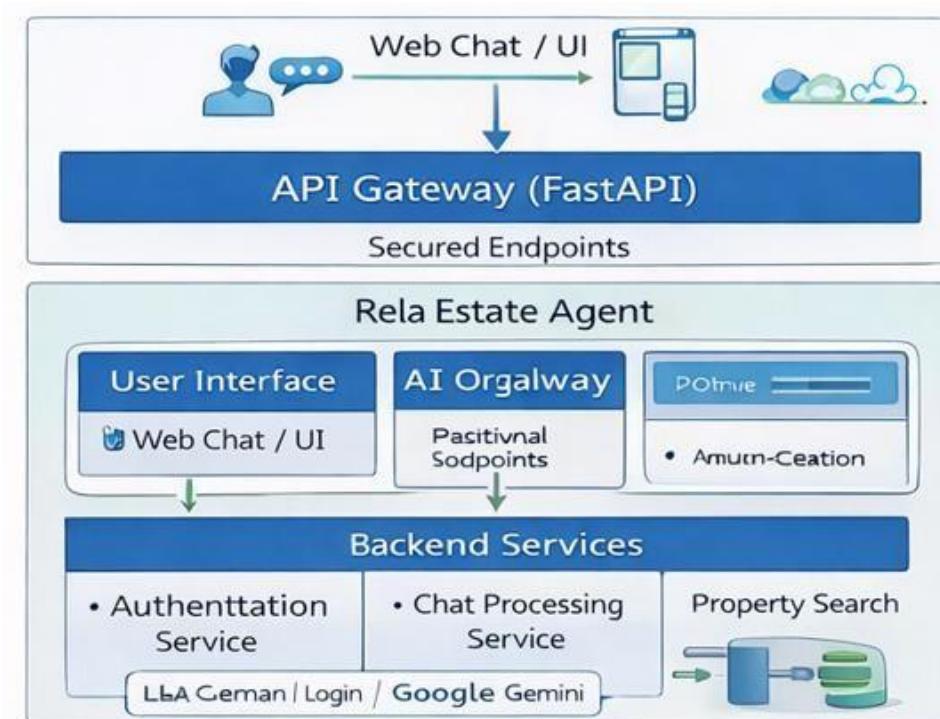
Output



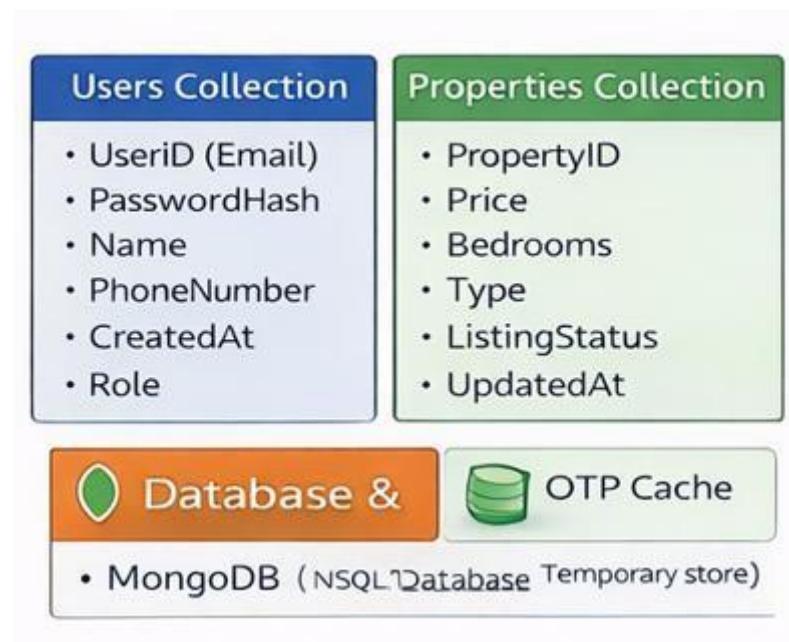
Property Card

2. Reports / Dashboards / Models

1. HLD Architecture Diagram



2. Data Model Diagram



3. Key Outcomes

- Successfully integrated AI with database search operations.
- Achieved real-time entity extraction from user queries.
- Implemented secure OTP expiration using TTL indexing.
- Designed scalable backend architecture.
- Improved search accuracy through filtered database queries.

Conclusion

The Real Estate Support & Intelligent Triage Agent System demonstrates the effective integration of artificial intelligence with backend web technologies and NoSQL database systems. The system successfully processes natural language queries, extracts relevant entities, and dynamically performs filtered property searches.

Through this project, practical experience was gained in backend API development, AI orchestration, database schema design, and system architecture planning. The project highlights how AI-driven systems can significantly improve user experience and operational efficiency in the real estate domain.

The modular and scalable architecture ensures that the system can be extended for production-level deployment with minimal modifications.

Future Scope & Enhancements

The system can be enhanced further by incorporating:

1. Cloud Deployment (AWS / Azure / GCP)
2. Docker-based containerization
3. Redis caching for faster query responses
4. Advanced recommendation engine using ML models
5. Real-time analytics dashboard
6. Role-based admin control panel
7. Multi-language AI support
8. Voice-enabled search integration
9. Microservices architecture implementation
10. Automated property insights using data analytics

These enhancements would make the system production-ready and suitable for enterprise-level real estate platforms.