Estate Nexus: The AI Driven Property Marketplace

MAHEEN ANWAR
01-134221-036
SUFYAN HASSAN
01-134221-076

Supervisor: Dr. Hafiz Ishfaq Ahmad

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

October 2025

Certificate

We accept the work contained in the report titled “Estate Nexus: The AI Driven Property Marketplace”, written by MAHEEN ANWAR AND SUFYAN HASSAN as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor:

_____

Internal Examiner:

_____

External Examiner:

_____

Project Coordinator:

_____

Head of the Department:

_____

Abstract

The real estate market still has a big problem with fragmented communication, inefficient property matching, and transparency between buyers and sellers. Buyers cannot see properties that do not fit their needs because of generic listings. On the other hand, sellers have a hard time finding serious buyers. The existing platforms are scattered in more than one website which makes property information very difficult to find and verify. Fragmentation causes delay in negotiation, escalation of costs, and mistrust between the parties. Further, most existing platforms do not incorporate smart recommendations powered by artificial intelligence thus forcing users to manually go through irrelevant listings which is not only time-consuming but also inefficient. The existing systems do not have real-time communication tools such as direct chats therefore limiting support to the users. The lack of a unified system for processing property offers and user feedback adds more inefficiencies to the market, extends the negotiation time, and leads to lost opportunities. This platform solution has at its core AI and machine learning that will be continually trained to provide personalized property recommendations, improve matchmaking, and upgrade communications between parties involved. The centralization of this process with real-time assistance shall strive to cut delays by automating property matching as well as bringing more transparency into the deal hence creating a better real estate experience. This is Estate Nexus - a full-stack intelligent real estate platform meant for modernizing and streamlining property transactions through empowering not only the buyers but also the sellers by providing a centralized transparent data-driven marketplace. It uses a current tech stack that has the MERN (MongoDB, Express.js, React, Node.js) architecture for a responsive web application and smart features enabled by machine learning. Advanced search and filtering options, real-time chat to allow direct communication between buyers and sellers, as well as a strong review and rating system are part of features offered by Estate Nexus. The intelligence at its core comes from a content-based property recommendation engine suggesting relevant listings plus a machine learning model for property price prediction to help in valuation. In addition, sentiment analysis on user reviews is used to automatically determine satisfaction with a property. With the help of these technologies, Estate Nexus improves the efficiency and transparency of the process, as well as enriching the user experience with insightful and personalized information, making it a next generation solution for the property market.

Acknowledgments

In the name of Our Creator we start by thanking THE ONE who continuously bestows His Mercies, Blessings, Favors Welcomes us with open arms and who without Whom we are nothing. We are indebted to the grace and mercy of God Who sustained us throughout this endeavor and without Whose help it would not have been possible to bring this project to a fruition. We would like to convey our heartfelt gratitude to our loving parents. Their stable love, ceaseless prayers, and unwavering support is what we leaned on to propel us through the whole of our academic pursuit. Their patience, encouragement and confidence in us have continuallyinspired us and give us strength. It is with great credit and thankfulness that we recognize ourproject supervisor, Dr. Hafiz Ishfaq Ahmad. His unstinted support, extraordinary knowledge, and penetrating direction made a strong contribution to this work. His support and all the constructive comments we have had were invaluable when facing problems and critically reviewing the work, up to the final version. We are also thankful to our university and all the faculty members who have imparted their knowledge and supported us throughout our degree. Finally, we extend our thanks to our friends and colleagues for their camaraderie and encouragement, which contributed significantly to this journey.

MAHEEN ANWAR
SUFYAN HASSAN
Bahria University
E-8, Islamabad, Pakistan

October 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
SE	Software Engineering
SQL	Structured Query Language
UNESCO	United Nations Educational, Scientific and Cultural Organization
UNICODE	Unique, Universal, and Uniform Character enCoding
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Project Overview

Real estate is a pillar of the global economy, but what seemed a solid pillar turned out not to be so solid. Buyers and sellers alike are struggling. Buyers and sellers are alike in that they often struggle with issues such as information asymmetry, guessing the value of real estate, and poor communication mechanisms, which can result in drawn-out and fraught transaction processes [1]. Traditional real estate markets mainly act as listing services, providing very few tools to enable the users to access insight based on data and therefore leaving important decisions, such as price and trust evaluation, to rely on rigorous and often subjective manual inspection. New developments in artificial intelligence (AI) and machine learning (ML) are expected to transform this industry. It has been found that machine learning approaches especially Random Forest Regressors can predict real estate prices with high accuracy using different factors such as location, property information, and the market trends [2]. In a similar sentiment, user comments on properties (e.g. reviews) are analyzed for sentiment, which is emerging as the more effective measure of public perception and quality of the property compared to simple star-ratings, distilling a more nuanced sentiment from user experiences [3]. Furthermore, online real estate marketplaces that focus solely on the property purchase side have reported success and competitive advantage by designing features that reduce information asymmetry and bring users together [4]. The social media-fueled, greater communication efficiency-driven positive followed by negative economic effects on market activities and user utility in real estate trading activities are also confirmed [5]. Based on the traditional market concept, this papers proposes Estate Nexus, a full service AI-driven real estate marketplace designed for the 21st Century. Estate Nexus was developed using a powerful tech stack: MERN (MongoDB, Express.js, React.js, Node.js) framework along with cutting edge AI models,

real time messaging and much more. It features a Random Forest regression model, which is based on a massive number of property advertisements and predicts prices in an automated and data-driven manner [6]. This removes any uncertainty users have when making decisions about their money. In contrast to Traditional Platforms that Passively Showcase Reviews, Estate Nexus Dissects them with Logistic Regression. This is what turns the review sentiment into a proprietary "worthiness" score for each property, allowing it to provide buyers with a concise, trust-boosting metric [7]. Such policy helps to reduce information overload and biased reviewing. In order to improve user experience, the real-time chat system is embedded in the platform, which is built on Socket.io to support buyers and sellers to directly contact each other within the platform [8]. Combined with an interactive map powered by Leaflet.js, this means users will have a rich, focused experience.

With these three fundamental AI engines prediction pricing, sentiment-based recommendations, and smart property matchmaking integrated within a single, unified and secure web application, Estate Nexus is not just a listing service. It forms a multi-dimensional property trading ecosystem that improves transparency, efficiency and trust for all members of the market and is, indeed, a production-ready solution for the contemporary real estate oriented environment.

Table 1.1: Comparison of Existing Real Estate Applications and Implemented Solutions

Application Name	**Identified Weakness**	**Implemented Solution**
Zameen.com	1. Lacks a dedicated, direct chat system for quick communication between buyers and sellers. 2. Does not utilize sentiment analysis to interpret user feedback for refining recommendations.	1. Real-time messaging using WebSockets (Socket.IO) enabling instant communication between buyers and sellers. 2. NLP-based sentiment analysis module integrated to analyze reviews and improve trust signals.
OLX	1. A general classifieds platform, not dedicated to real estate — fragmented UX. 2. Lacks AI-driven personalized property recommendations. 3. No structured rating and review system.	1. Dedicated platform focused on property listing and discovery. 2. AI-powered recommendation for suggestions. 3. Trustworthy with a comprehensive rating and review system
Graana	1. Does not leverage sentiment analysis to refine listings based on user reviews. 2. Limited real-time communication features, slowing inquiries and negotiation.	1. Sentiment-enhanced insights from user comments to highlight reliable sellers. 2. Integrated real-time chat to ensure immediate engagement and reduce transaction delays.

1.2 Problem Description

The emergence of online property portals has revolutionised the real estate industry providing sellers and buyers a digital platform to list and hunt for houses. But those platforms are essentially staid digital classifieds, not employing technologies that might help to reduce some of the market's biggest inefficiencies. This effort revealed the following key gaps in the current solutions of property trading platforms that hinder transparency in decision-making and transaction efficiency.

- Information Asymmetry and Pricing Inaccuracy: A pivotal issue in real estate is the asymmetry of information between seller and buyer. Sellers often have a hard time figuring out how to price their homes competitively because they lack objective data to inform their calculations, and buyers don't have transparent tools to help them decide whether the asking price is aligned with market values. Pricing in classical platforms is manual and heuristics based, which ultimately could be a subjective and unstable point. In the absence of automated, data - driven valuation models, there are inefficiencies in the market, longer time on market, and mistrust between the parties.

- Trust, superficial review systems: While reviews are often integrated in property platforms, their usefulness is rather limited. The majority of systems aggregate simple star ratings without analyzing contrasting opinions within user feedback. This makes it more difficult for buyers to separate the truly credible properties from those with ratings inflated by a small number of reviews. Because they lack cognitive sentiment analysis, they cannot detect sentiments about, for example, the state of the property, the responsiveness of the landlord, or the safety of the neighborhood and therefore they provide shallow and untrustworthy trust metrics.

- Dispersed and Bottlenecked Communication Systems: Many markets employ a synchronous communication medium such as email or primitive contact forms that impede the pace of exchanges between buyers and sellers. This fragmentation often drives users to conduct off-platform communications with personal messaging apps, interrupting the workflow of a transaction and making oversight of property-related conversations more difficult. The absence of integrated real-time communication tools directly slows transaction progress and reduces user satisfaction.

- Sufficiency of Data for Personalisation and Data-Driven Insights: Buyers frequently face an overwhelming quantity of homes that are clearly neither remotely near their tastes, nor financial abilities. Basic search criteria (e.g., price range, number of bedrooms) are insufficient to enable effective personalization. The process of discovering houses is cumbersome and reactive since there are no AI-driven recommendation systems that take into account the user's interactions and the home's features.

- Usability and the “Tech-Savvy” Barrier: Sophisticated analytics tools, if they exist at all, are usually meant for professional traders or brokers. That’s a thorn for the casual buyer or seller, who may lack the expertise to understand market trends, or the patience to navigate facet-heavy interfaces. For this, we want to the platform to promote wide participation through strong design and clear interaction flows that can accommodate very different users."

These challenges clearly highlight the need for a smart, connected and user-friendly solution for the property industry. Estate Nexus actively addresses these limitations by integrating AI-driven valuations, sentiment- based trust scoring, and real-time communication in a single platform, resulting in a more transparent, efficient and reliable environment for property trading.

1.3 Project Objectives

Through this project, we want to establish a property trading platform ESTATE NEXUS which can simplify the property trading with AI technology. the app is designed to

- Utilize a Random Forest regressor to estimate the property value, reducing pricing uncertainty for buyers and sellers.
- Incorporate Sentiment Analysis model in order to analyze the text contents of property reviews and generate helpful worthiness scores to the quality of recommendations.
- Implement Socket.io for real-time direct communication between buyers and sellers, so no more delays and more fluid communication throughout the whole transaction.
- Build a full-fledged web-based solution leveraging MERN stack with an interactive map view and user authentication to provide an end-to-end property marketplace solution.

1.4 Project Scope

The outline of this project is the design of Estate Nexus, a full-stack web application with AI to transform how users search for and purchase homes. It will be a full-stack marketplace platform, connecting buyers and sellers, with smart feature to help buyers makes informed purchasing decision. The project will adopt a phased approach to build a platform where users can advertise properties, as well as search for and evaluate properties, and AI-based tools will offer automatic price appraisals and sentiment-driven investment advisories. A live chat will also be implemented for users to talk directly to each other for a better and more fluid experience. The basis of the project is to build and integrate a

machine learning price prediction model with Random Forest regression, based on a local property listings dataset. Moreover, the project will be developing a sentiment analysis engine based on Logistic Regression model that processes user reviews and computes property attractiveness score. The development specifically covers Meteor for the core app (MongoDB, Express, React, Node) with Leaflet.js for maps, Socket.io for real-time messaging, Cloudinary for image management. Please keep in mind that the project will not include the enabling of payment or legal contract execution on the platform, as these are regulated financial and legal services and this application is not an application for those services. Moreover, market estimates are provided by AI Price Prediction based on the information available and do not replace a formal, certified property appraisal. The system is also not intended to confirm the absolute legal availability or ownership of the properties listed, which the parties involved must do. With an emphasis on AI-driven insights, robust listing management and improved communication, this product aims to be a compelling middle layer between traditional property search and the raw power of data-tech.

1.5 Summary

In this paper, we propose an AI-based property trading platform, which can achieve the same efficiency as the classical market mechanism with automated price estimation, sentiment-aware recommendation, and human-like interaction. The system is based on a trained Random Forest regressor on the full property data, yielding precise price estimation. Furthermore, it applies a sentiment analysis model to the users' comments to determine property suitability scores for every comment of a user. By embedding these AI components into a full-stack MERN platform, this system is able to:

- Accurate estimation of the price based on active property features such as property distance to central business district (CBD), house specification and macro market data.
- Improved decisions through sentiment-based analysis of product reviews.
- Real-time chat, interactive map-based interface, and mobile- and web-compatible design for saving you property searching and communicating time. Merging machine learning, natural language processing and modern web best-practices, this project will provide a property dealing platform that is both scalable, power efficient, and user centered. Estate Nexus The Estate Nexus marketplace is pushing the boundaries of proptech by offering a practical demonstration of how AI can be harnessed in the real estate industry; enabling those who use it to have far greater market transparency, the ability to make better-informed decisions, and an overall smoother transactional process.

Chapter 2

Literature Review

Estate Nexus is to integrate AI price prediction, sentiment, AI recommendation and real time communication in one body to build a smart property market. By employing a Random Forest regression model to predict the value and Sentiment Analysis to evaluate the reviews, the service provides a level of data-driven intelligence that surpasses a mere listing.This review describes how our solution fits in the stateof the art in AI for real estate, exposes limitations of the current property platform and extends search and valuation, as well as transaction experience via our system.

2.1 Introduction to AI in Real Estate and Property Marketplaces

Applications of artificial intelligence (AI) have transformed conventional real estate industry, integrating data-centric methodologies into the process where physical inspection and subjective intuition were paramount in the past. Conventional real estate markets typically function like digital classifieds: information is static, and it does not address the underlying problems of asymmetric information or market pricing [9]. On the other hand, AI-enabled products are used to value homes, analyze market sentiment and make customized recommendations, enabling a more open and efficient market. It is well documented that ML is very good at price prediction in real estate. Studies demonstrate potential of ensemble techniques such as Random Forest Regressor to achieve high accuracy by capturing nonlinear relationship among significant attributes such as location, size and amenities [2][6]. In addition, the mining of unstructured data, i.e. user reviews, has become an important approach to evaluate the attractiveness of properties. Nursal et al. (2024) and Xu et al. (2025) applied sentiment analysis to real estate, showing that quantifying emotional tone in reviews provides deeper insights into property quality and neighborhood perception beyond numeric ratings [10][11].

Platforms in markets such as China have started to incorporate these technologies, employing AI to process online news sentiment and its spatial impacts on housing prices[12] . Yet, a lot of the available implementations are still fragmented, dealing with either valuation or sentiment analysis and very rarely combining the two into an integrated, end user marketplace. Moreover, issues related to the transparency of models, quality of data, and the "black box" nature of certain AI techniques are still open challenges to be investigated . Such limitations underscore the importance for powerful, explainable, and seamlessly integrated AI mechanisms that can directly enable the end user. Estate Nexus seeks to fulfill this demand by bringing together a number of AI features in a single interface.

2.2 Existing Property Platforms and Their Limitations

The present state of digital real estate services fall into numerous categories, each with its own benefits and drawbacks.

2.2.1 Traditional and Basic Online Marketplaces

Platforms like Zameen.com and Zillow have a wide and deep selection of listings, but they operate more like informational databases than smart platforms.

- No AI-Powered Insights: These platforms have little in the way of automated valuation or data-backed suggestions, and price and appraise still depend heavily on manual work.
- Terminal Review Systems: Users typically are restricted to a binary star rating at best, without any more granular sentiment analysis, which may hide nuanced opinions about property condition or landlord reliability.
- Passive Search Experience: In the lack of intelligent, customized recommender systems, users are often overwhelmed by generic listings.

2.2.2 AI-Enhanced Analytics and Valuation Tools

A few standalone platforms and tools have incorporated AI for specific analytical functions, such as automated property valuations.

- Narrow Focus: Most of these products cater to investors or very technical users, and the analytics can be mind-boggling, plus they don't offer a full transaction platform, so they're not really usable for average buyers or sellers.
- Limited Line of Communication: Most of the analytical tools have no features of direct engagement between the buyers and sellers, which is a crucial part of the transaction.

2.2.3 New Integrated platforms

A other batch of marketplaces attempts to smartly integrate listings but are still limited in AI.

- Generic recommendations: Some recommendations approach may be based simple collaborative filtering approach without deeper semantic/sentiment analysis and provide less relevant recommendations.
- Scalability and Real-Time Performance Issues: Features such as live chat must be implemented at scale, and they come with certain technical challenges. Although there are libraries (ex: Socket.io) which makes easier the implementation of real-time communication, very few platforms can efficiently management a big volume of persistent connections and keep history of messages.

Chapter 3

Requirement Specifications

3.1 Existing System

The real estate market transactions are now monopolized in Pakistan by portals such as Zameen.com, OLX, Graana.com etc. These sites have transformed the real estate ads in traditional classified format into a digital format with a few basic search and filter options. Yet, they're really digital bulletin boards with significant limitations that dampen market fluidity and transparency. Sites like Zameen.com and Graana.com do aggregate a huge amount of listings, but none of them had an objective, data-driven means to value property. Consequently, there is significant information asymmetry and buyers and sellers make manual price comparisons as well as heuristic-driven tactics that often result in subjective, and in certain cases, inconsistent pricing which does not align with market's true value. Actually, the interaction is still disconnected as these sites usually don't promote in-site, real-time chatting. Users have to converse via the phone and Whatsapp, leading to delayed responses, disrupted negotiations, and a fragmented user experience [13]. Tribler is very bad in this respect and generally systems tend to be. Platforms such as OLX do not have any formal reviewing or rating system while the rest have simple star ratings that do not take into account the content of the text reviews. This stop users from having a more fine-grained understanding of property credibility or seller reliability as emotional sentiment and specific issues in reviews are not quantified [14].

Searching for properties is more of a passive activity that relies on manual user work. Existing websites do not have intelligent mechanism to actively recommend properties considering user behaviors, preferences, and property characteristics Users need to repeatedly use the same search filters, without any personalized, AI-based recommendations.[15].

These constraints subjective pricing, fragmented communication, shallow trust protocols and passive discovery combine to make for a market that is inefficient, tiresome, and

irritating to customers, and which fails to take advantage of contemporary AI and real-time web-technologies.

3.2 Proposed System

Our system, Estate Nexus, is a web-based full-stack application powered by AI which redefines the current real estate market by combining machine learning with web development to overcome the inefficiencies of present day real estate platforms.

Like an intelligent broker, Estate Nexus provides users with data-rich insights and effortless engagement. To fight against information asymmetry, a trained Random Forest regression model is employed by the system to produce an automated property price prediction. This unbiased valuation considers significant features such as location, property information and market data to enable you to make well-informed financial decisions whether you are a buyer or a seller [16]. Logistic Regression Regent Analysis in Arete Estate Nexus is built to your trust and authenticity. This NLP technique processes the input textual user reviews and returns a numeric "worthiness" score for each property. It goes beyond the star ratings by offering an aggregate, sentiment-weighted measure, that allows for more insight into the property and the sellers' reputation [10].

The platform answers the fractured conversation by providing a single, real-time chat system right in the app. Using Socket. A Redis adapter for socket.io is used to send messages between buyers and sellers in real time and persistently so that negotiations and questions stay in one trackable environment [8].

In order to provide an effective and dynamic interaction with users, an interactive map view with Leaflet.js is implemented to allow geographic browsing of the properties. The robust, scalable and responsive MERN stack (MongoDB, Express.js, React.js, Node.js) is used for the application.

The project has two main methodology tracks: the AI and Data Pipeline Track and the Application Development Track. Both tracks operate simultaneously and converge at defined integration points, ensuring efficient development and seamless system integration for the final deployment of Estate Nexus.

There are two primary areas of methodology in the project: AI and Data Pipeline Track and the Application Development Track. The two tracks run in parallel and interface at specific points, allowing for a smooth development process and eventual system integration for deployment of Estate Nexus.

The AI and Data Pipeline Track develops the intelligence of the system. It starts with data acquisition — by using web scraping on real estate sites to obtain property data. Then, the collected data are preprocessed and cleaned to treat missing values, noise, and inconsistent data. When the data is organized, feature engineering and vectorization are applied by using TF-IDF, which transforms text into numeric vectors with useful

information. These are features which are used to train and test machine learning models, such as Random Forest and Logistic Regression, on various tasks, including sentiment analysis and recommendation aid. Upon successful evaluation, the trained models are stored and exposed through APIs. The AI services are implemented in FastAPI with Python, and the predicting can be requested in the application layer at real time. This track terminates in an integration point that attaches the AI-services to the main system.

The Application development Track the main application built on the MERN stack, deals with the the most critical aspects of the solution the key functionality and user the platform. This path begins with designing the database schema based on user, property, and interaction data with MongoDB. Next, there is the backend development, which is where the CRUD operations are implemented along with the authentication methods to make sure that access is secured. Real-time communication is activated using Socket.io in order to implement a direct user chat and more. Frontend: The frontend application is built with React, with an emphasis on intuitive UI/UX and engaging features including property maps. After the build is done, the application is proceeded by integration and system testing to check on system behaviors. This track terminates at its final integration point after a successful system validation.

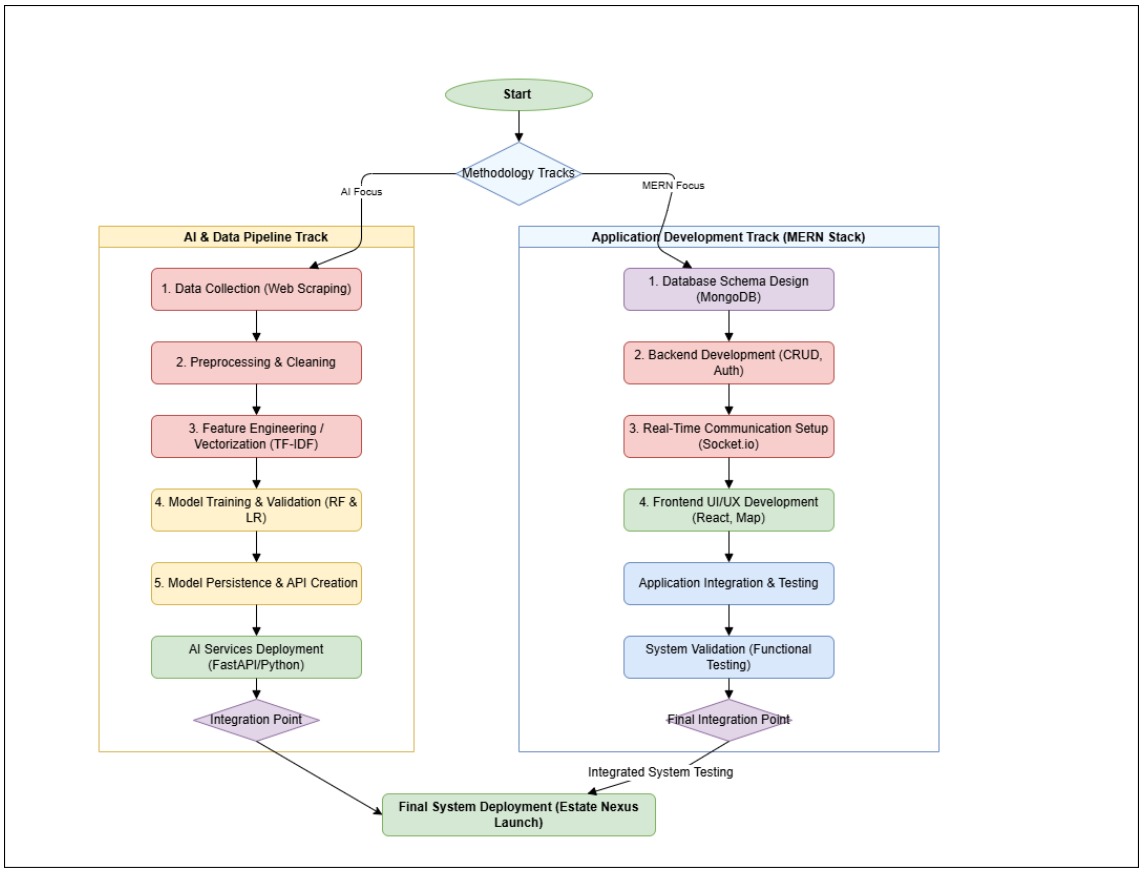

Figure 3.1: Methodology Diagram of Proposed System

3.2.1 Functional Requirements

- **AI-Powered Price Prediction:** The system should predict the price of the property with high precision by considering important features like location, bedrooms, bathrooms and area employing trained Random Forest regression model.
- **Sentiment-Based Recommendation:** A custom-Sentiment Analyzer is used to analyze the user review text and calculate and present a property "worthiness" score.
- **Real-Time Communication:** The system shall enable buyers and sellers to communicate in real time in a persistent manner directly within the platform, using a WebSocket-based connection.
- **Interactive Property Discovery:** The system will be capable of delivering an interactive map interface to enable visualization of a property location and to support searching and filtering of listings by various parameters.
- **User Authentication Authorization:** Users shall be able to register, login and keep their sessions safe with role-based differentiation for buyer and seller.

3.2.2 Non-Functional Requirements

- **Performance:** The price prediction model should produce an estimate in under 2 seconds, and live chat messages should be delivered with latency below 500ms.
- **Accuracy:** The AI price prediction model should have a good R^2 score on the test data, and sentiment analysis scores should truly represent the sentiment expressed in the text of the reviews.
- **Usability:** The user interface shall be easy to use and responsive, with a unified desktop, tablet and mobile experience for both technical and non-technical users.
- **Scalability:** The system architecture shall support up to 10,000 concurrent users and efficiently manage a growing database of over 100,000 property listings.
- **Security:** User passwords shall be encrypted using bcryptjs
- **Compatibility:** The web application shall be fully accessible and functional on all major modern web browsers (Chrome, Firefox, Safari, Edge).

3.2.3 Access Home Page

This use case describes When you first arrive at the Estate Nexus website, you'll find yourself on the homepage. Here, you can immediately see a variety of property listings that are open for anyone to browse. If you want to look around, you can start exploring

these properties straight away. Or, if you'd like to use more personal features, you'll find options to Log In or Sign Up for an account. Clicking on either of these will smoothly take you to the right page to continue.

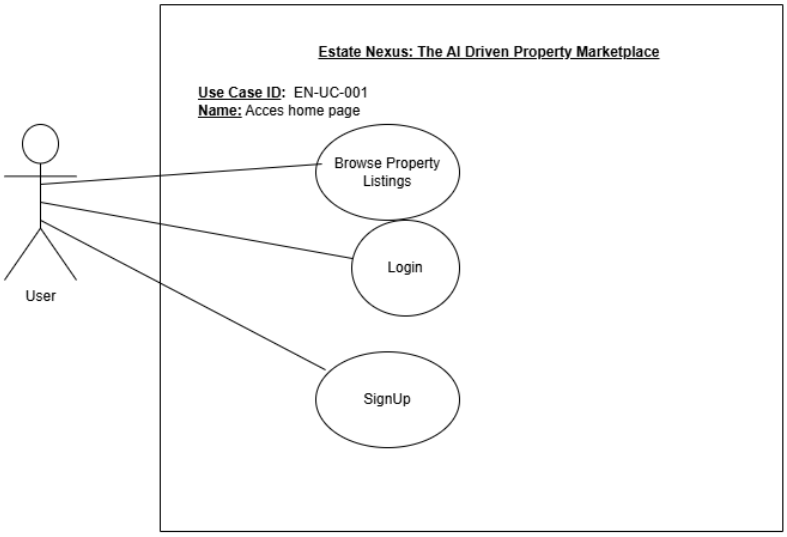

Figure 3.2: Use Case Diagram — Access Home Page

Table 3.1: Use Case Specification: Access Home Page

Use Case ID	EN-UC-001
Use Case Name	Access Home Page
Actors	User
Description	User opens the Estate Nexus homepage and uses navigation or authentication options.
Basic Flow	(a) User enters the platform URL. (b) Homepage loads. (c) User views public listings. (d) User selects Login or Sign Up. (e) System redirects accordingly.
Alternative Flow	System shows a maintenance page if the service is unavailable.
Preconditions	• Internet access. • Functional web browser.
Postconditions	User is directed to authentication (login/signup) or continues browsing public content.

3.2.4 New User Registration

This is the process for someone new, whether they want to buy or sell, to create an account. The person starts by entering their basic information and selecting their main role on the site. The system then checks this information. If the email address is already in use, or if something doesn't look right, it will ask the person to either log in (if they already have

an account) or fix the details they provided. If everything is correct, the system creates a secure account.

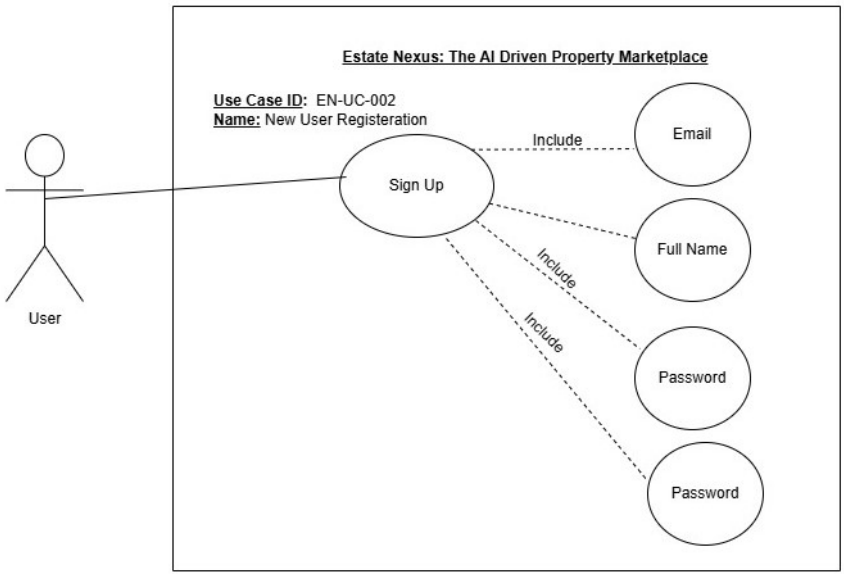

Figure 3.3: Use Case Diagram - New User Registration

Table 3.2: Use Case Specification: New User Registration

Use Case ID	EN-UC-002
Use Case Name	New User Registration
Actors	Prospective User
Description	Allows a new user to register an account on the Estate Nexus platform.
Preconditions	User is on the homepage with access to the Sign Up option.
Basic Flow	(a) User clicks “Sign Up”. (b) System displays registration form. (c) User enters username, email, password, and role. (d) System validates input. (e) System creates the account. (f) System sends verification email. (g) User verifies email. (h) System redirects to dashboard.
Alternative Flow	If the email already exists, system prompts user to log in.
Postconditions	A new user account is created and user is logged in.

3.2.5 User Login Authentication

This section covers the process for someone who already has an account to log back into the site. The user begins by entering their registered email address and password. The system then verifies this login information to confirm the person’s identity. If the email and password match the records, access is granted. The user is immediately taken to their

personal dashboard. If the details don't match for example, if the password is wrong or the email isn't recognized the system displays an error message.

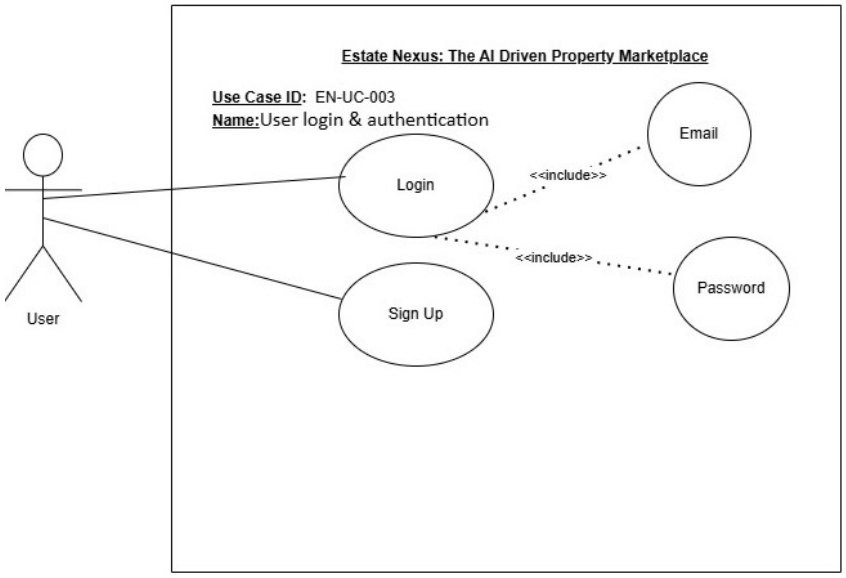

Figure 3.4: Use Case Diagram - User Login Authentication

Table 3.3: Use Case Specification: User Login Authentication

Use Case ID	EN-UC-003
Use Case Name	User Login Authentication
Actors	Registered User
Description	Allows an existing user to authenticate and access their account.
Preconditions	User already has a registered account.
Basic Flow	(a) User clicks "Login". (b) System displays login form. (c) User enters email and password. (d) System validates credentials. (e) System creates an authenticated session. (f) System redirects to dashboard.
Alternative Flow	If credentials are invalid, system shows an error message.
Postconditions	User is authenticated and can access personalized features.

3.2.6 Property Search and Discover

This shows how buyers can look for properties on the platform. In this process, a buyer can set search filters such as a preferred location, price range, or number of bedrooms to narrow down options. The system applies these criteria and displays a matching list of properties. The buyer can then click through the results and choose any listing to see its full details. This diagram describes the main flow of a discovery property. Additionally, it

covers what the system does if no properties satisfy the applied filters (to keep the buyer informed on that matter too).

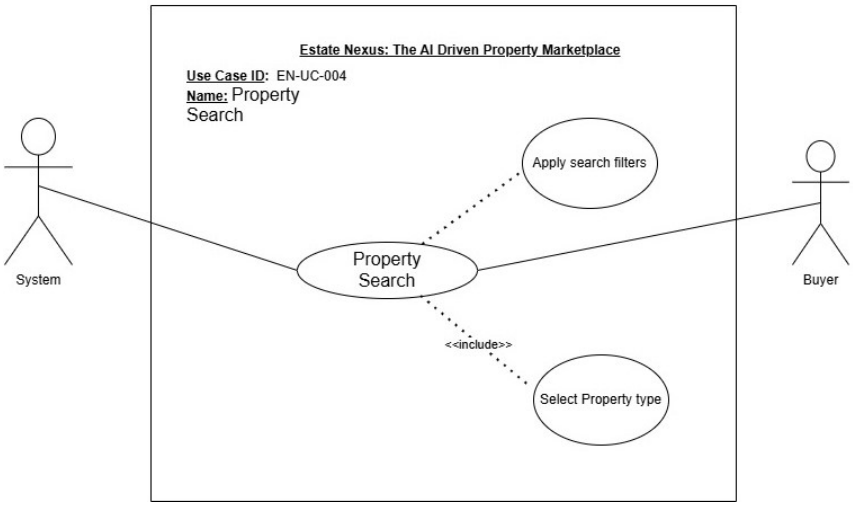

Figure 3.5: Use Case Diagram - Property Search and Discovery

Table 3.4: Use Case Specification: Property Search and Discovery

Use Case ID	EN-UC-004
Use Case Name	Property Search and Discovery
Actors	Buyer
Description	Allows users to search and explore properties using different criteria.
Preconditions	User is logged in and on the search page.
Basic Flow	(a) User selects filters: location, price range, bedrooms, type. (b) System queries the database using selected criteria. (c) System shows matching properties in list or grid view. (d) User selects a property to view details. (e) System displays detailed page with recommendations.
Alternative Flow	If no results match, system suggests widening search filters.
Postconditions	User views relevant listings and may proceed to inspect a property.

3.2.7 Real-Time Buyer-Seller Communication

This use case describes the instant messaging functionality that connects buyer and seller on the application. Buyers can also start a conversation from a property listing page. Once a message is delivered, stored and retrieved immediatelly, enabling a real-time conversation, perfect for discussing terms. Messages are sent and received in real time. If a message arrives while the recipient is offline, the message is kept securely. When that recipient logs back in, the message is delivered automatically, so you never lose a conversation.

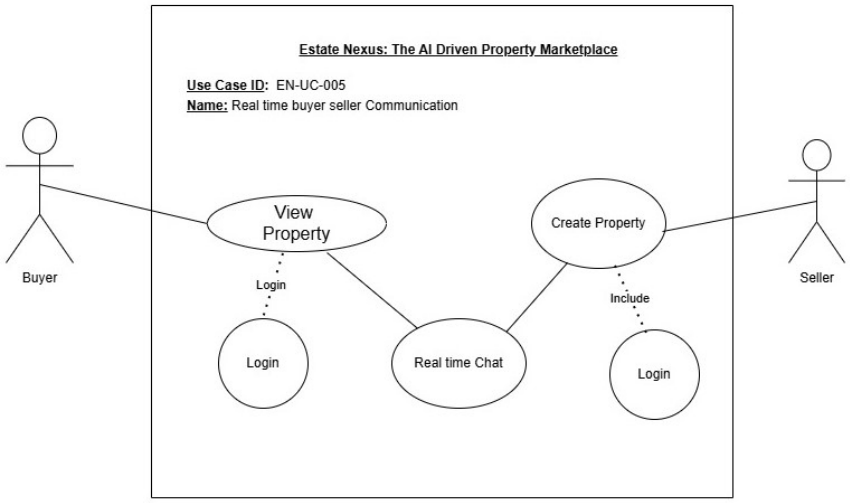

Figure 3.6: Use Case Diagram - Real-Time Buyer-Seller Communication

Table 3.5: Use Case Specification: Real-Time Buyer-Seller Communication

Use Case ID	EN-UC-005
Use Case Name	Real-Time Buyer-Seller Communication
Actors	Buyer, Seller
Description	Enables instant messaging between buyers and sellers for negotiations.
Preconditions	Both users are logged in; buyer is viewing the seller's property.
Basic Flow	(a) Buyer clicks "Message Seller". (b) System opens the chat interface. (c) Buyer sends a message. (d) System delivers it in real time via WebSocket. (e) Seller receives notification and replies. (f) Conversation continues in chat window.
Alternative Flow	If user is offline, messages are stored and delivered when they log in.
Postconditions	A direct communication channel is established for negotiation.

3.2.8 Review Submission and Analysis

This feature allows buyers to talk about a property they have visited. After looking at a listing or asking about it, the buyer has the option of leaving a written review with a star rating. The written feedback is firstly automatically processed by the system to determine its overall sentiment (e.g., positive or negative). This derived insight, along with the new rating, is then stored and added to the profile of the property. This gave prospective buyers a better idea of the listing.

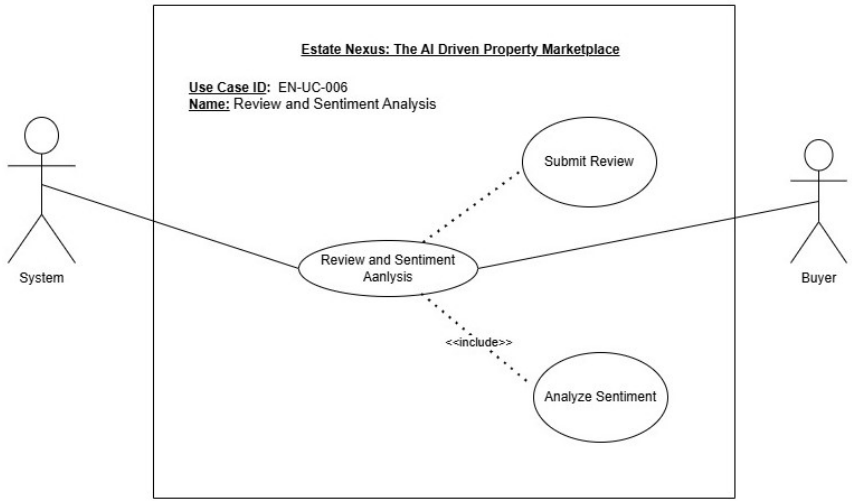

Figure 3.7: Use Case Diagram - Review Submission and Analysis

Table 3.6: Use Case Specification: Review Submission and Analysis

Use Case ID	EN-UC-006
Use Case Name	Review Submission and Analysis
Actors	Buyer
Description	Allows buyers to submit reviews and ratings for properties, which are analyzed for sentiment.
Preconditions	Buyer has viewed or completed a transaction for the property.
Basic Flow	(a) Buyer navigates to the property page and clicks “Add Review”. (b) Buyer selects a star rating and writes review text. (c) Buyer submits the review. (d) System processes the review using Sentiment Analyzer. (e) System extracts sentiment insights. (f) System updates the property rating and displays the analyzed review.
Alternative Flow	If the review contains inappropriate content, the system flags it for moderation.
Postconditions	Property rating is updated; sentiment insights are stored and visible to future viewers.

3.2.9 Property Listing Management

This use case tells how sellers can handle all their property listings from their account dashboard. This space lets them create brand new listings by adding descriptions, details, and photos. They can also go back to edit, update, or remove any of their existing listings at any time. Whenever changes are made, the system checks the information and saves it.

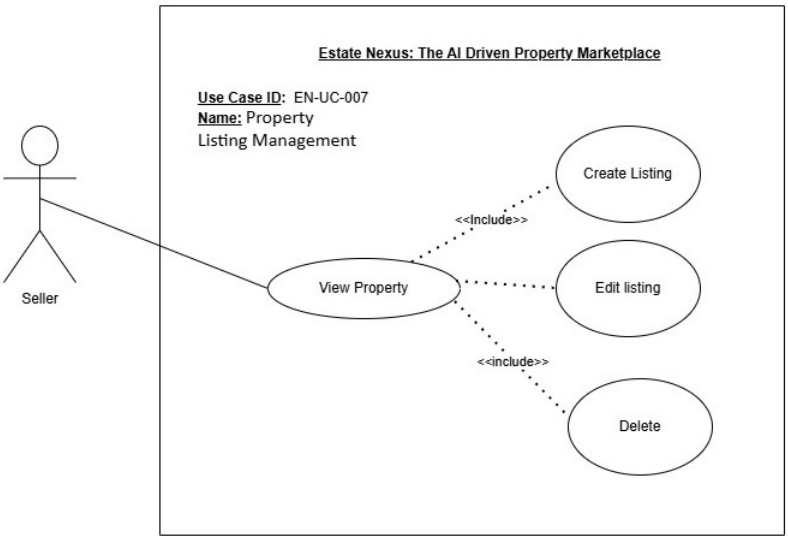

Figure 3.8: Use Case Diagram - Property Listing Management

Table 3.7: Use Case Specification: Property Listing Management

Use Case ID	EN-UC-007
Use Case Name	Property Listing Management
Actors	Seller
Description	Enables sellers to create, edit, and manage property listings on the platform.
Preconditions	Seller is logged in to their account.
Basic Flow	(a) Seller accesses the "My Listings" dashboard. (b) System displays all existing property listings. (c) Seller creates a new listing with detailed property information. (d) Seller edits existing listing details and images. (e) Seller updates pricing and availability information. (f) Seller delete listings when sold. (g) System validates and updates all changes.
Alternative Flow	Seller can save listings as drafts to complete at a later time.
Postconditions	All property listings are updated and managed successfully on the platform.

3.2.10 Property Recommendations (AI/ML-based)

This use case describes the AI-powered system that provides property suggestions to a Buyer based on what propert they are browsing. It learns from their browsing activity and preferences to recommend listings with similar details or locations. For someone new, it instead highlights popular properties to help them get started. These recommendations are shown on listing pages.

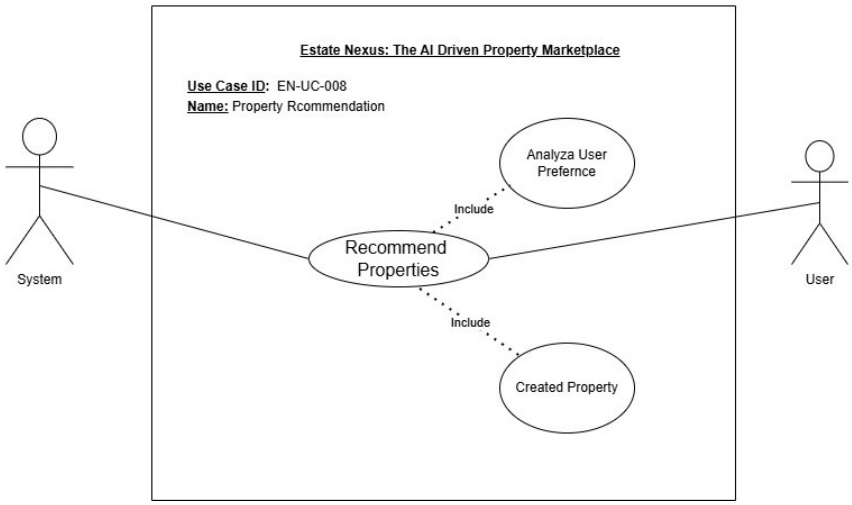

Figure 3.9: Use Case Diagram - Property Recommendations (AI/ML-based)

Table 3.8: Use Case Specification: Property Recommendations

Use Case ID	EN-UC-008
Use Case Name	Property Recommendations
Actors	System, Buyer
Description	Provides personalized property suggestions to buyers using AI-driven recommendation algorithms.
Preconditions	Buyer has interacted with properties on the platform.
Basic Flow	(a) System analyzes the buyer's property interactions and preferences. (b) System calculates similarity scores based on property features. (c) System applies recommendation algorithms. (d) System generates property suggestions. (e) Recommendations are displayed on the property pages. (f) System continuously updates recommendations based on new interactions.
Alternative Flow	For new users without interaction history, the system displays popular and trending properties.
Postconditions	Buyer receives relevant property suggestions according to their preferences.

3.2.11 Property Price Prediction

This feature provides an estimated market value for a property, offering guidance to both buyers and sellers. A user enters the key details of a property like location, size, number of bedrooms. The algorithm of the system then processes the information with market data to calculate a price prediction. When there is not enough local sales history to confidently make an estimate, the system will communicate this limitation to the user so the results are delivered with a full sense of transparency.

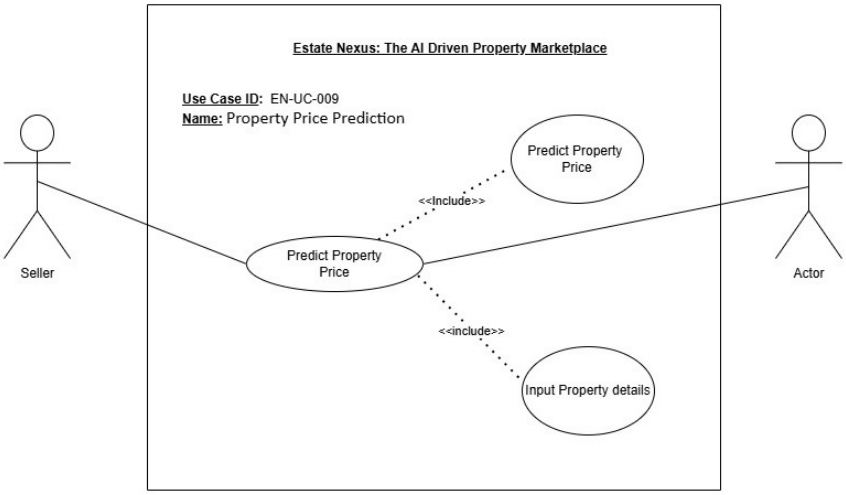

Figure 3.10: Use Case Diagram - Property Price Prediction

Table 3.9: Use Case Specification: Property Price Prediction

Use Case ID	EN-UC-009
Use Case Name	Property Price Prediction
Actors	Buyer, Seller
Description	Provides an AI-powered market valuation for a property to help users make informed pricing decisions.
Preconditions	User is logged into their account. Sufficient market data is available in the system.
Basic Flow	(a) User (Seller or Buyer) navigates to the price prediction feature. (b) User enters or selects key property attributes (e.g., location, size, bedrooms, amenities). (c) System processes the input using the trained machine learning model. (d) System generates and displays a predicted price range with key influencing factors. (e) User views the valuation report.
Alternative Flow	If the system lacks sufficient data for a reliable prediction, it displays a message indicating the limitation.
Postconditions	The user receives a data-driven price estimate, which is logged for model improvement.

Chapter 4

Design

In this chapter, we detail the design of the Estate Nexus platform at the system, software, data flow and component levels. The objective is to have a well defined roadmap of implementation by showing the modules exchanges to realize the platform functionalities.

4.1 System Architecture

EstateNexus is a web-based real estate application that aims to bring the property search and management process into the 21st century. It includes smart recommendations, property listing capabilities, real-time user interaction, and price estimation powered by artificial intelligence in one system. The design is based on client-server architecture to achieve the modularity, scalability and maintainability.

Above all, the front end is based on React, where we have created a dynamic and intuitive user interface where you can browse properties, create listings, talk to other users, and get recommendations. The front-end makes RESTful HTTP requests to the FastAPI backend to send and receive data in JSON format.

Backend can be considered as brain of the platform. It provides authentication, handles property information, manages chat the conversations, contains business logic, and integrates with external AI/ML services. A PostgreSQL database is used to store users, properties, chats, and recommendation information persistently. Furthermore, external APIs are utilized for AI-driven components including sentiment analyses and price prediction.

A real-time communication layer enables buyers and sellers to have seamless interaction within the platform through instant messaging, and equally facilitates transactions.

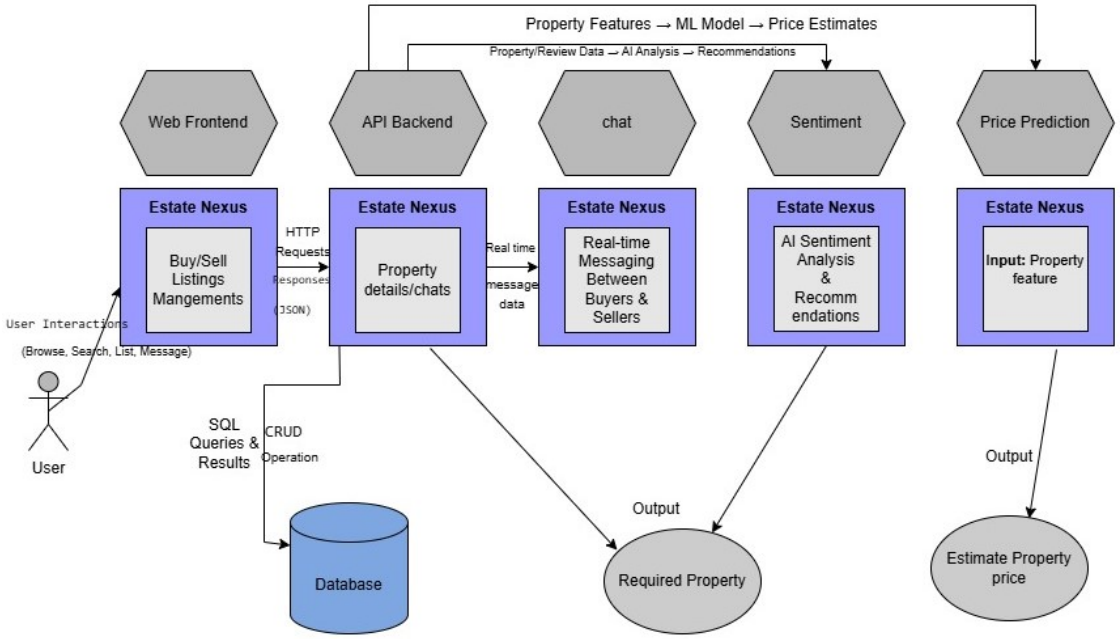

Figure 4.1: System Architecture

Figure 4.1 depicts the architecture of the system and the communication interface between React frontend, the FastAPI backend layer, the database layer, and the integrated AI/ML service.

4.2 Software Architecture

The Estate Nexus platform architecture is developed to deliver a modular, scalable and efficient system development environment. Based on a client–server architecture, the front-end and back-end layers have well-defined responsibilities and can be developed, tested and deployed in isolation. This division of responsibilities makes debugging easier, improves maintainability and renders the system more flexible for future enhancements.

The system has been designed with React as front-end which facilitates an interactive and responsive interface for the users to browse properties, communicate and manage users. The backend is written in Node.js and Express which controls the business logic, authentication and secure api communication. MongoDB: The primary database incorporated into the architecture is MongoDB, which accommodates user data, property listings, reviews, and transaction records in a manner that is both flexible and scalable. An AI services layer is added separately to support intelligent services with Python and FastAPI for sentiment analysis, recommendations, and predictive insights. Real-time functionalities, i.e., direct messaging, are enabled via Socket.io to provide low-latency services for users. The design allows each component to scale independently depending on the workload.

Such an architectural style also enhances system dependability yet ensuring maintainability and future extensibility.

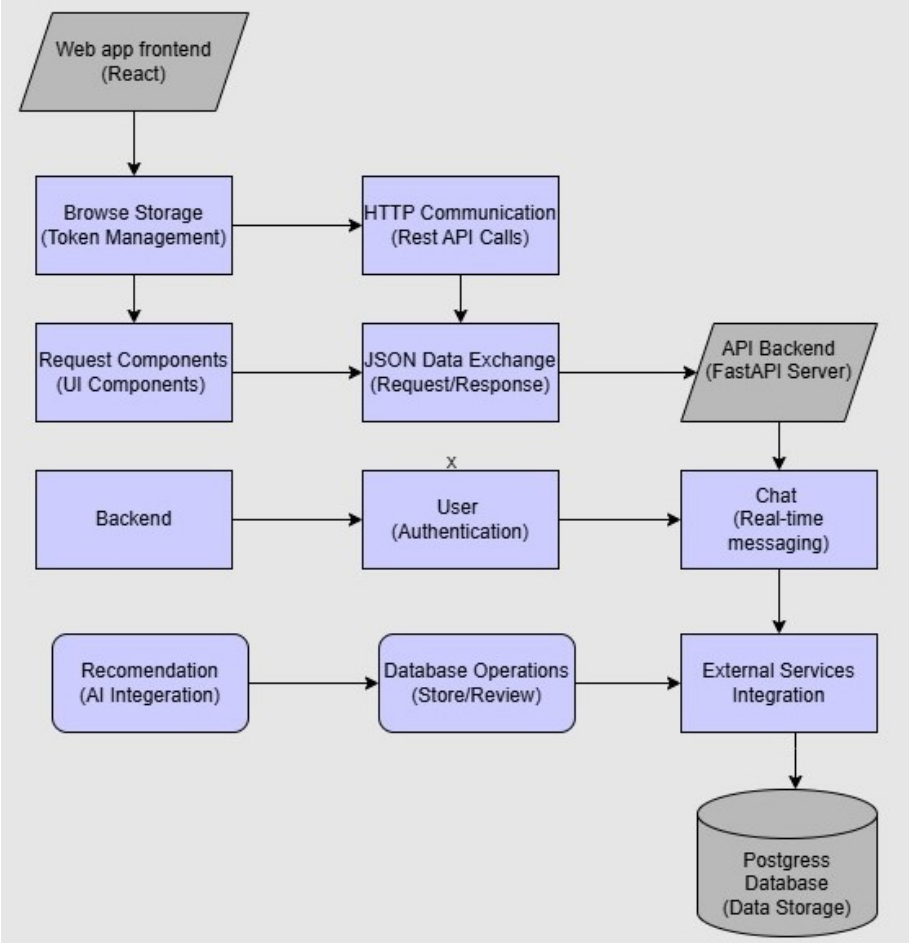

Figure 4.2: Software Architecture Diagram

Figure 4.2 also shows that the architecture of the Estate Nexus platform is a React FastAPI client server Application Stack. Handling the user interface with the use of React, the front-end application manages the user side interaction like viewing properties, creating listings, real-time chatting and viewing customizable suggestions.

The front end and the FastAPI backend communicate via RESTful HTTP requests and respond with JSON. As the central access for all business logic, the FastAPI server routes requests to backend modules (e.g., Property App, User App, Chat App, Recommendation App) based on the type of request received and all business logic resides here. These modules cover essential services such as JWT Based Authentication, Real Estate Management Services, Chat and AI Based Recommendation Services.

All data is backed by a PostgreSQL database, providing durable storage for user profiles, properties, chat messages, recommendations, and system logs. Additional external services

(AI/ML models for price prediction Cloudinary for image storage) are also consumed by the backend.

4.3 Design Constraints

The Estate Nexus platform was the subject of the series of technical and usability challenges, which led to the the most important design decisions. Among the main limitations are:

- **Limited computational resources:** It's a juggling act to have your backend server, database, and frontend all running efficiently and smoothly, particularly when being accessed by many users at once.
- **Real-Time Performance Constraints:** Property search, real-time chat and other services need to be very responsive to ensure a fluid user experience.
- **Scalability constraints :** The system must be designed to be scalable to support increasing number of property listings, user accounts, and active sessions without adversely affecting performance.
- **Integration Constraints:** The platform relies on third party services such as map and geolocation APIs, that may impose rate limits or other restrictions, and that the system should be able to handle gracefully.
- **User Acceptance / Usability Constraints:** The UI must be simple to use even for first-time inexperienced users, but at the same time must expose all the advanced features required by the agents and sellers, which poses a very difficult usability challenge.

4.4 Design Methodology

The development approach of Estate Nexus was based on the agile methodology and focused on iterative design and build the base modules for each iteration, property transactions, real-time communication, AI-based capabilities, etc. This iterative process ensured compatibility between all parts of the system. The process began under Develop Concepts– accumulating requirements by defining use cases for property searching/selling, chat channels, and ML models for price predictions and recommendations. Develop User Interface., at this stage, we created interactive prototypes for property search filters, listing management dashboards, and real-time chat facilities. Finally, Testing and Evaluating confirmed system integration via accuracy tests of sentiment analysis, performance validations of the recommendation engine, and execution tests of real-time chat. After each iteration, user feedback and system performance data were incorporated into further refinement of the concept, leading to improvements in all modules right to project completion.

4.5 High Level Design

This section presents descriptions of the Estate Nexus system at different levels of architectural detail:

1. **Conceptual/Logical View:** Exposes a functional grouping of User Management, Property Management, Search and Discovery, Communication, and AI/ML Services.
2. **Process View:** Models execution: sessions, property searches, real-time chat, workflows for AI processing.
3. **Physical View:** Illustrates deployment in cloud infrastructure showing the web server, application server, database server, and AI/ML service tiers.
4. **Module View:** Differentiates between client React components, server FastAPI routers, and AI/ML-specific modules.
5. **Security View:** Deals with authentication, authorization, data encryption, and API security layers.

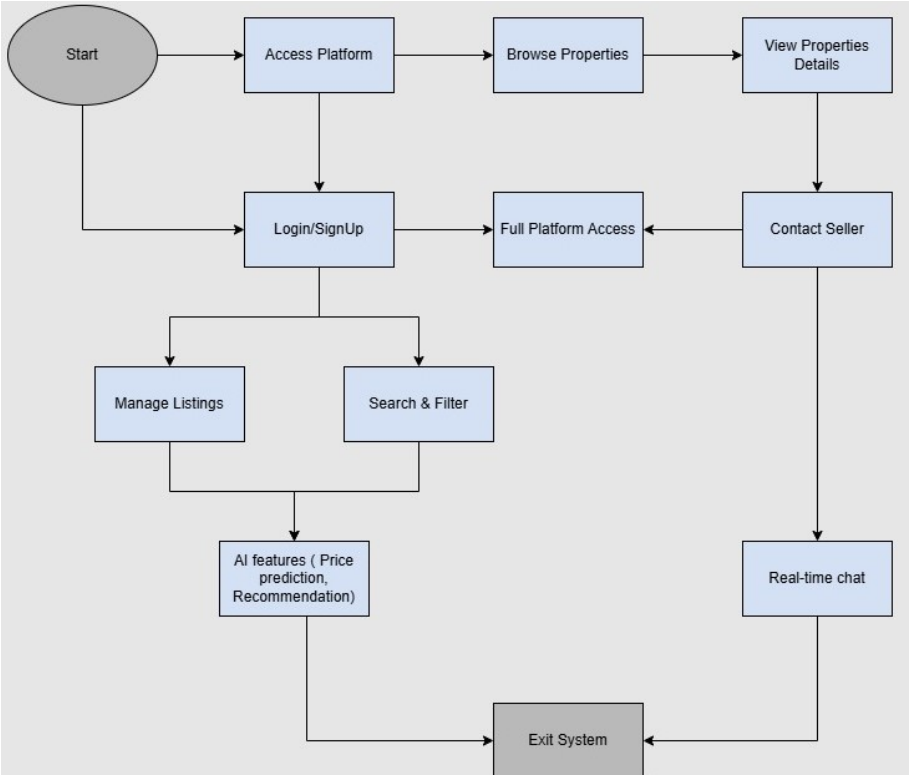

Figure 4.3: System High Level Diagram

Figure 4.3 is a simplified view of the Estate Nexus platform that shows the flow of user interaction, from the discovery of properties to the completion of transactions.

It starts when users open the website on web browser. Browse properties or login/signup for full access. Verified users can search properties, get full details, contact sellers by chat and get AI based recommendations. Sellers post and manage listings and communicate with buyers and the reverse happens for buyers. Real-time communication is enabled and the property discovery and valuation are intelligent-driven.

4.6 Low-Level Design

In this section, we present the low-level design of the Estate Nexus platform, which outlines the module internals and their interactions. The components are fully decomposed into basic units to facilitate the development and to make the responsibility and the interfaces clear. The backend is Node.js with Express and the frontend is Angular with Material Design.

4.6.1 Property Management & Recommendation Module

The Property Management and Recommendation Module manages property listings and makes customized suggestions to consumers. Property information is taken in via user contributions and then normalized and processed to extract information features. The Recommendation Engine performs Collaborative filtering between users and properties such as user behavior, property similarities. Recommendations are scored by relevance and delivered via the front end.

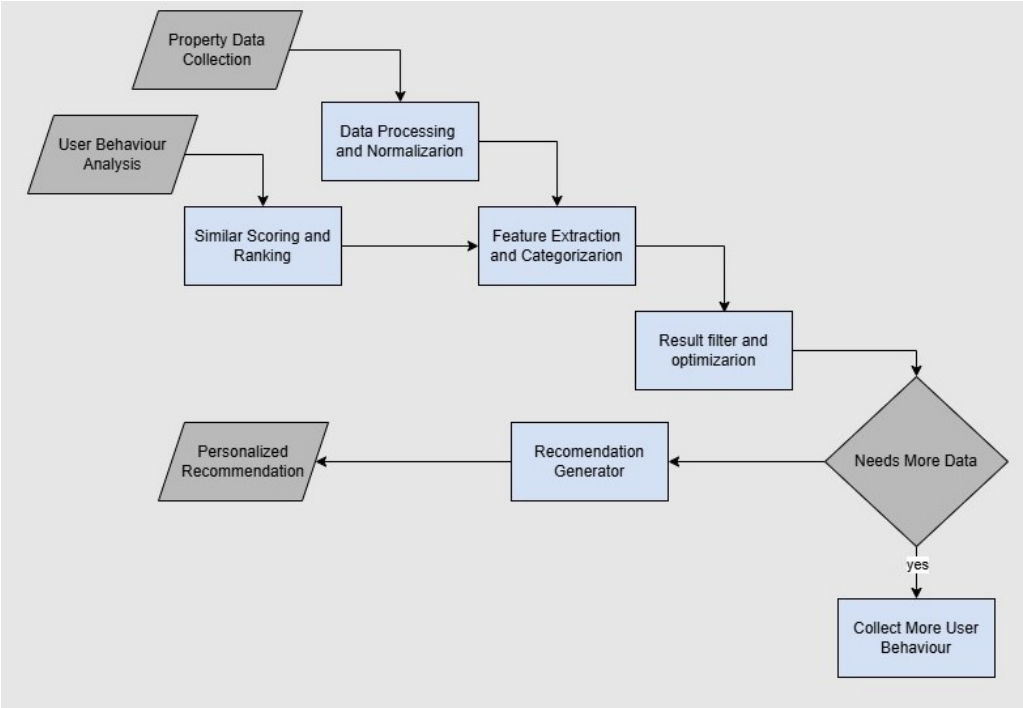

Figure 4.4: Property Management & Recommendation Module

4.6.2 Real-time Chat Module

Real-time Chat Module allows instant communication between buyers and sellers. Chat sessions are created when a buyer contacts a seller. The Message Handling Processor validates incoming messages and stores them. The Real-time Delivery subsystem uses WebSocket connections to deliver messages in real time, and the Chat History component takes care of message persistence and retrieval.

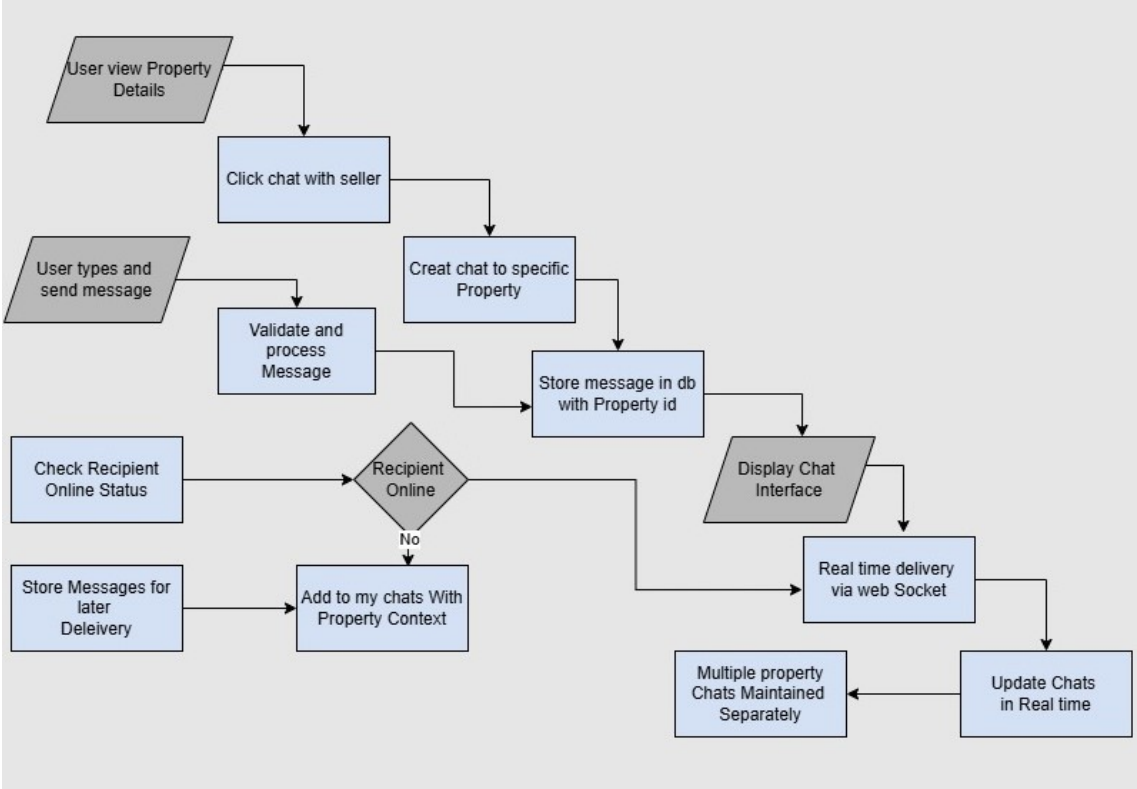

Figure 4.5: Real-time Chat Module

4.6.3 Price Prediction Module

The Price Prediction Module leverages machine learning to predict property prices. It gathers the past data, does the feature engineering, and trains the regression models. For a prediction, the module takes property features, runs the trained model and gives an estimate of the price with confidence intervals when prediction request is received.

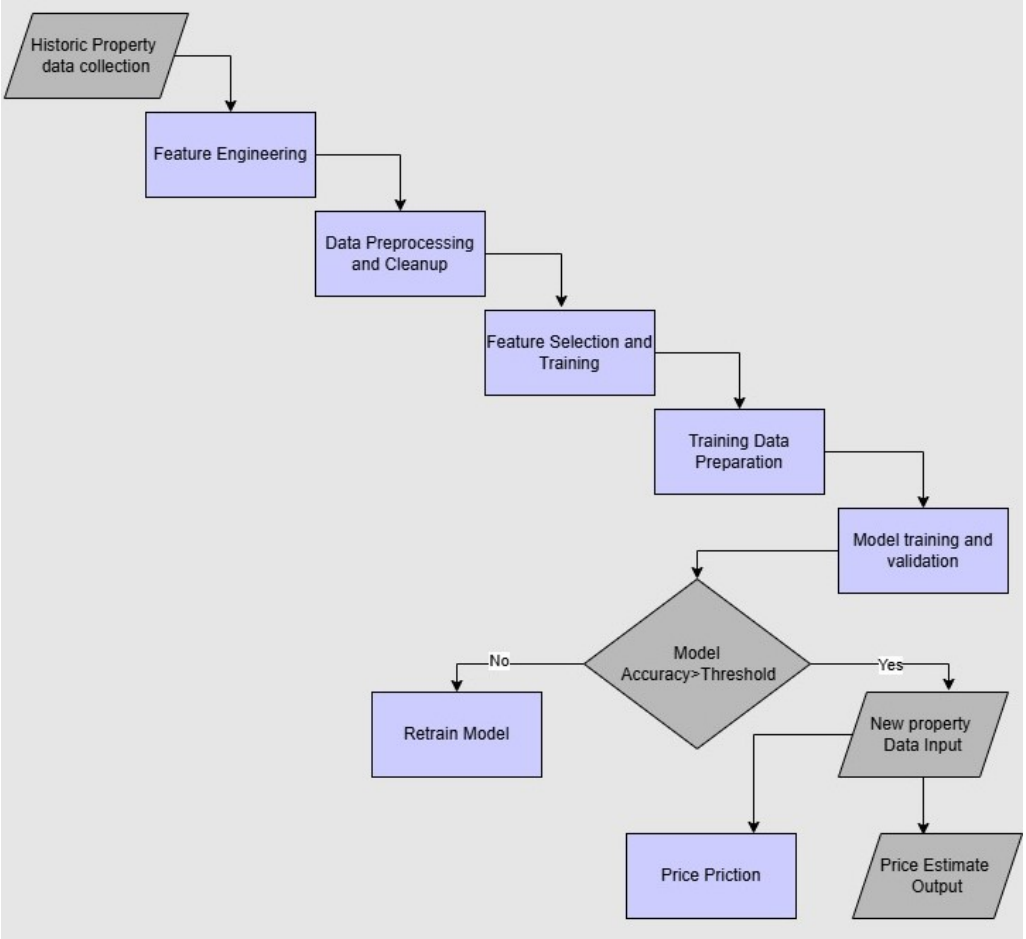

Figure 4.6: Price Prediction Module

4.6.4 User Authentication Module

The User Authentication Module handles register, login, and session management. It offers password hashing, JWT token generation, session validation, and role-based access control to secure platform resources and user data. All modules (referred also as components in this section) are implemented as individual packages with strictly defined interfaces, based on the object oriented principles and design patterns suitable for the Estate Nexus technology stack.

The flow starts after a user clicks login or register option and input the credential. The input format is checked first, to make sure the inputed data confirms to " format requirements". The system also checks if the request is login or register according to the user's option.

For registration, the system creates a strong password and checks the credentials given for validation before continuing. Once the credentials are confirmed valid, the user information is saved and process advances. At login, it verifies that the user exists in the

database. If no email is found, an error message saying “user not found” is sent. If the user is found, the entered password is checked to the stored credentials.

After a successful validation of the credentials, the system issues a JSON Web Token (JWT) that will be used to authenticate the user session. The session is then stored securely and a success response is sent back to user. At any point, when the credentials are invalid, an appropriate error message is shown. The result of this authentication flow is that it allows the system to have secure access control, avoid unauthorized access, and have a dependable way to manage sessions.

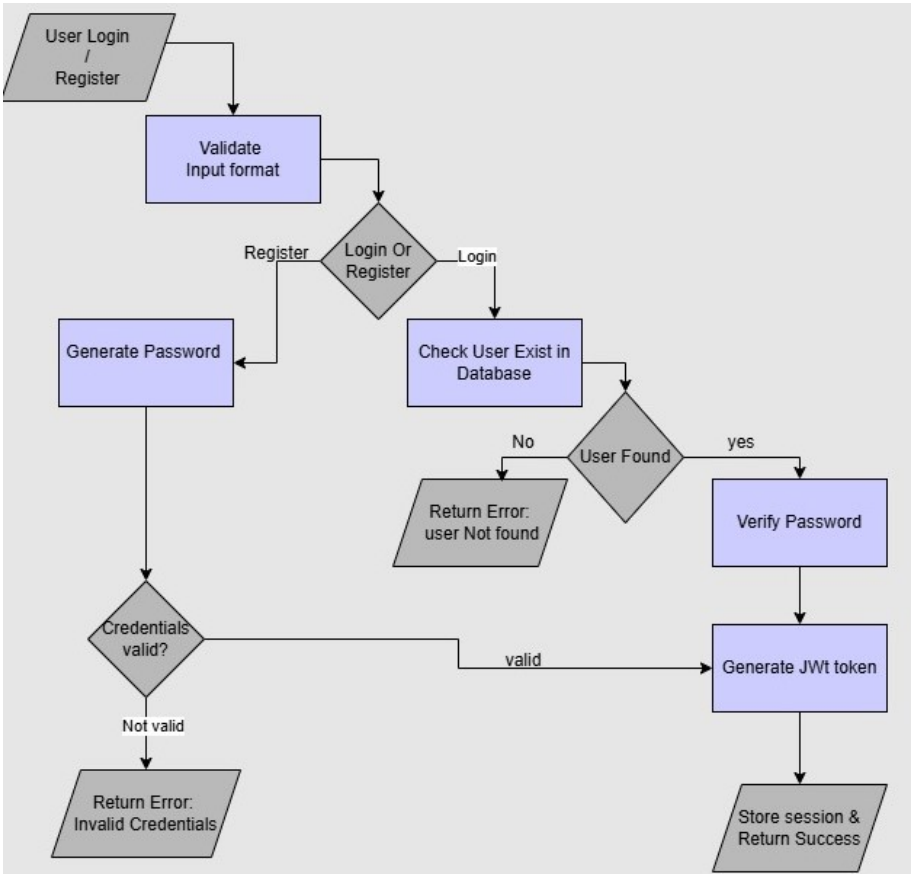

Figure 4.7: User Authentication Module

4.7 Database Design

The Estate Nexus platform organizesthe data efficiently by storing it as .csv files in a PostgreSQL relational database. The main tables and their roles are as follows:

- **Users Table:** Contains user information and credentials for authentication such as username, email, hashed password and profile data.

- **Properties Table:** Contains details of the property listing including title, description, price, location, number of bedrooms and bathrooms, size and type of property.
- **Chats Table:** Tracks chats between buyers and sellers with log of chat and references to related properties.
- **Messages Table:** Includes live messages between users from the senders identity with timestamps and the readers identity.
- **Reviews Table:** Contains user reviews and ratings for properties such as review text and timestamp..

Proper primary keys, foreign keys, data types, validation contraints and other similar mechanisms have been taken care of in each of the tables included in the Estate Nexus platform to guarantee data, relational and platform integrity and reliability.

4.8 GUI Design

The Estate Nexus platform is equipped with a responsive and intuitive graphical user interface implemented in React. The system lets users search for properties and filter the results, list a property, chat in real time, see suggestions, and get information from AI in one fell swoop. The principles of design focus on learnability, consistency, accessibility, and effective navigation to provide a better user experience.

4.8.1 User Signup

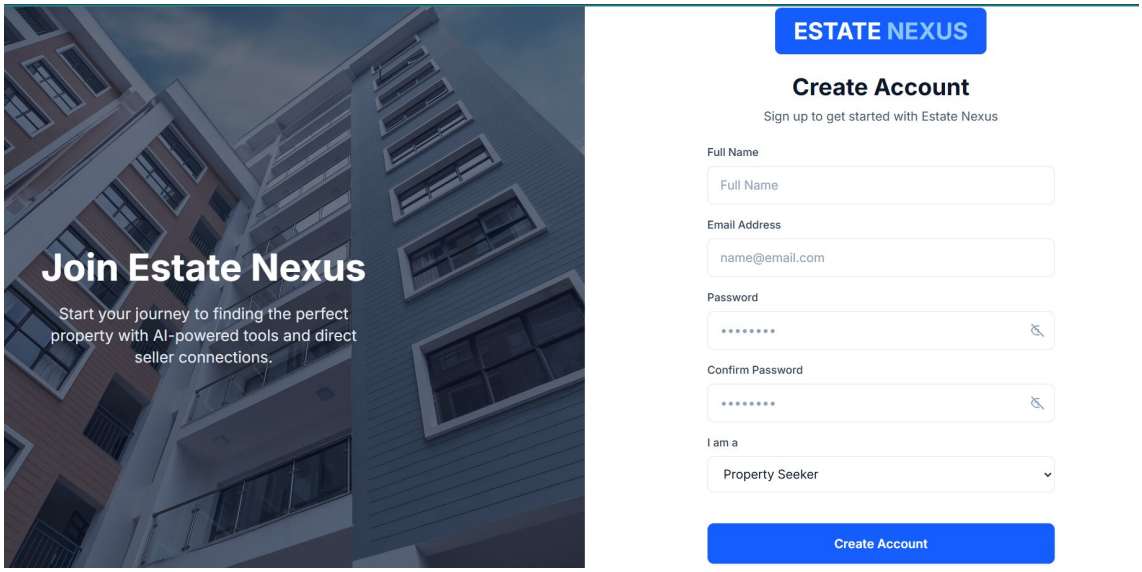

Figure 4.8: User Signup Interface

4.8.2 User Login

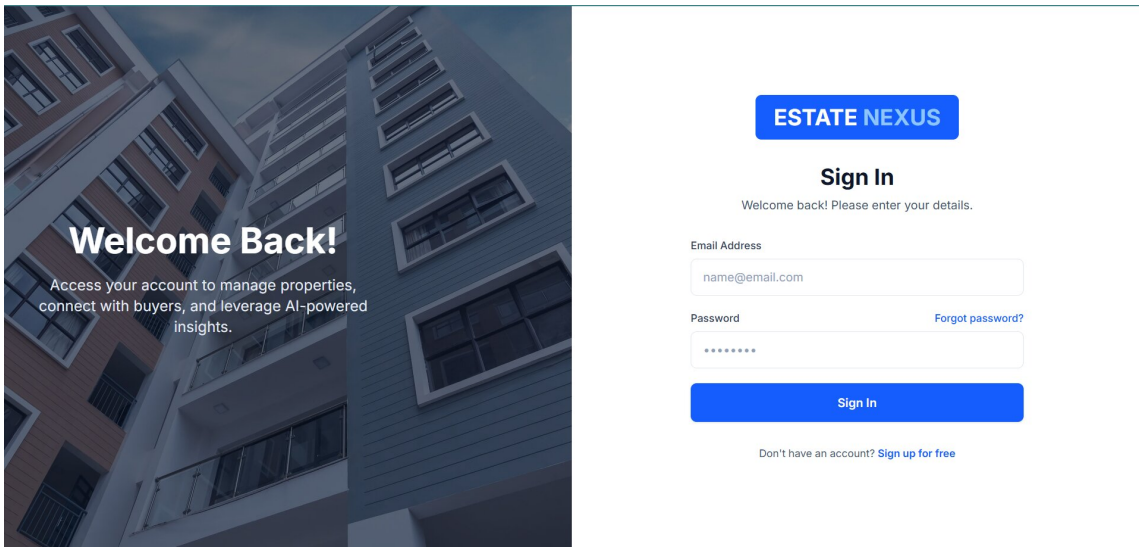

Figure 4.9: User Login Page

4.8.3 Property Listing Page

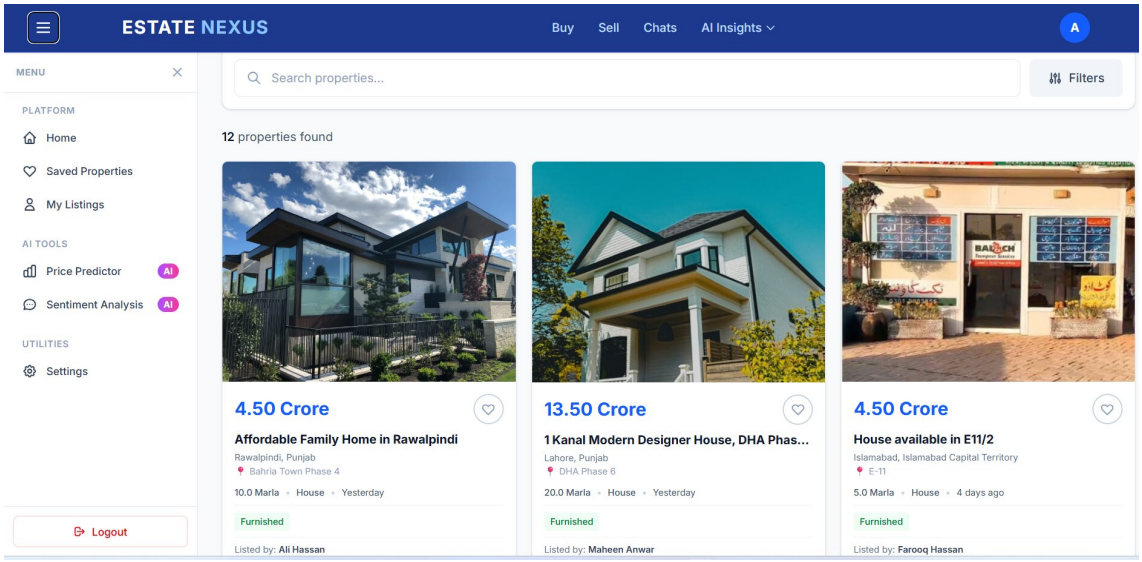

Figure 4.10: Property Listing Page UI

4.8.4 Seller Dashboard

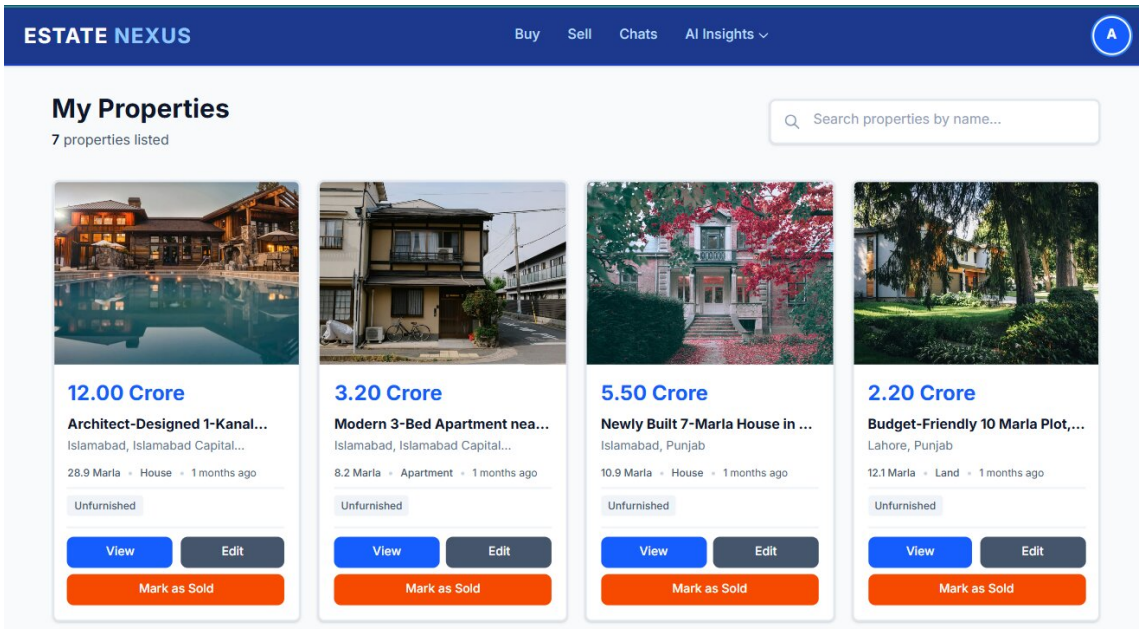

Figure 4.11: Seller Dashboard Showing All Listed Properties

4.8.5 Real-Time Communication (Chat System)

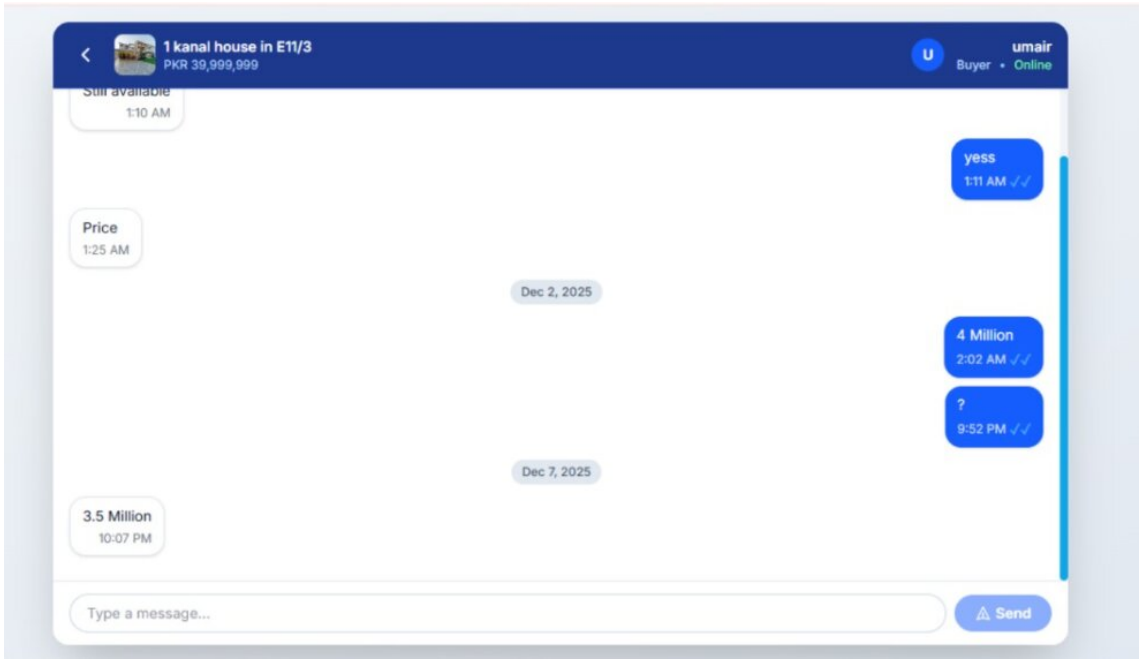

Figure 4.12: Real-Time Buyer–Seller Chat Interface

4.8.6 Displayed Property View

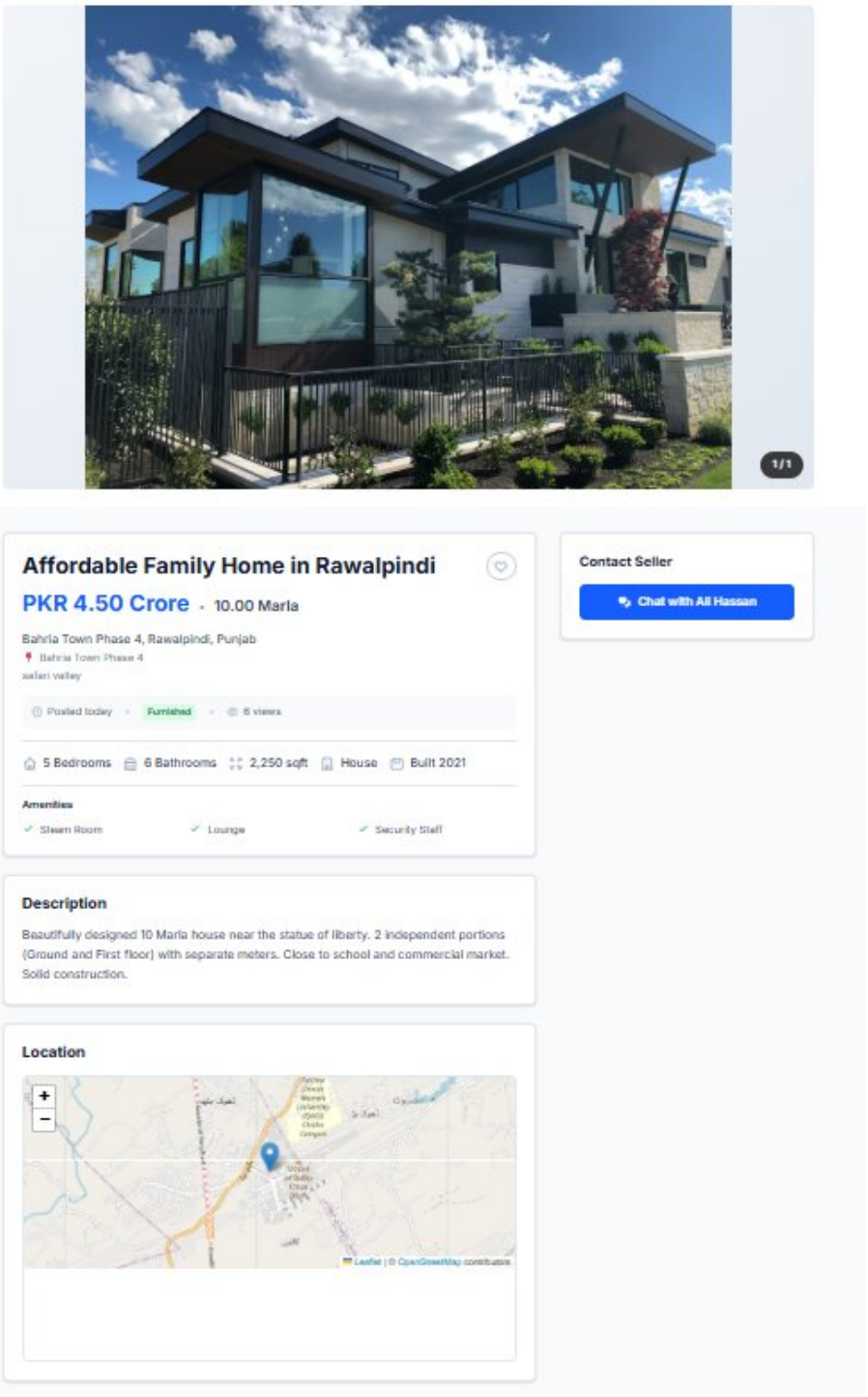

Figure 4.13: Detailed Property Display Page

4.8.7 Add New Property

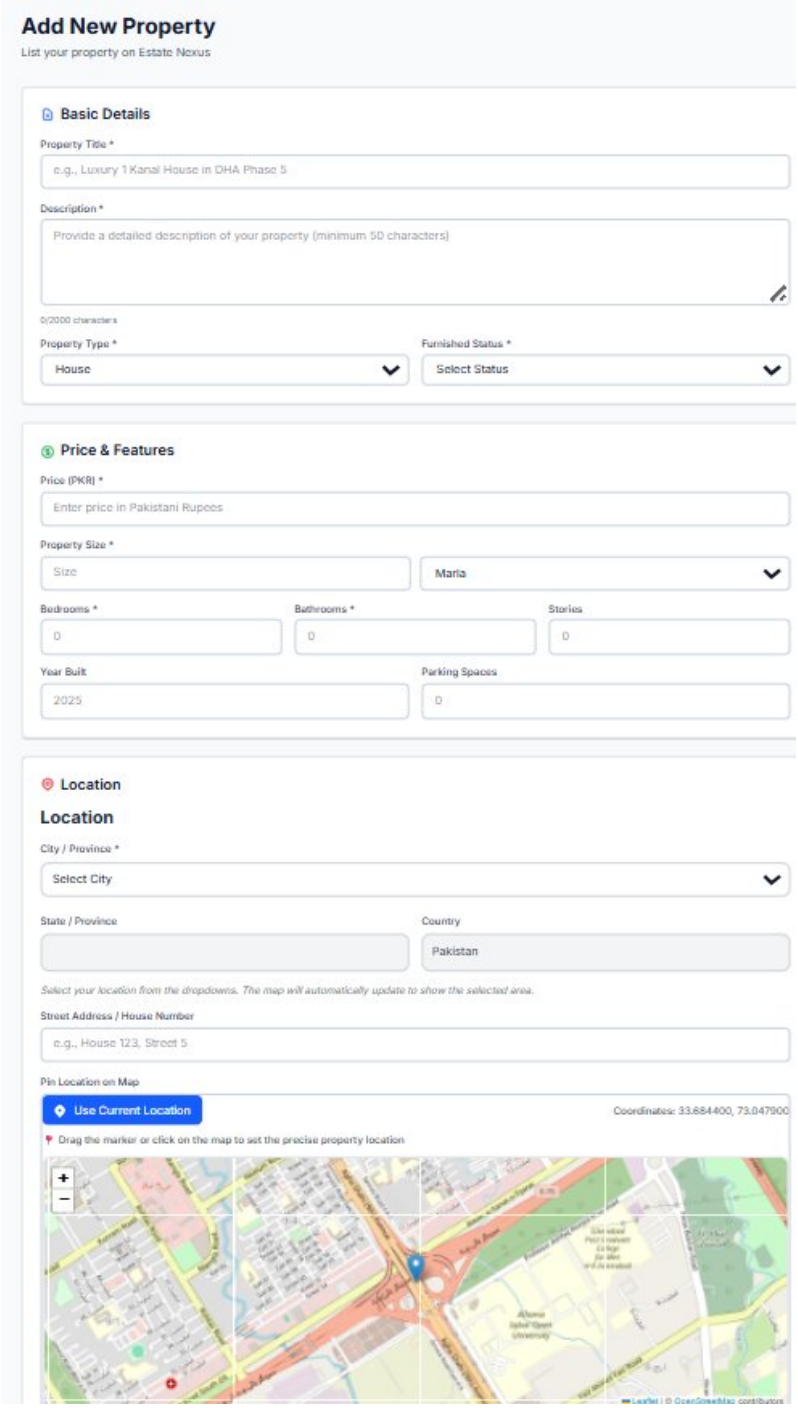

Figure 4.14: Add Property Page UI

4.8.8 Price Predictor

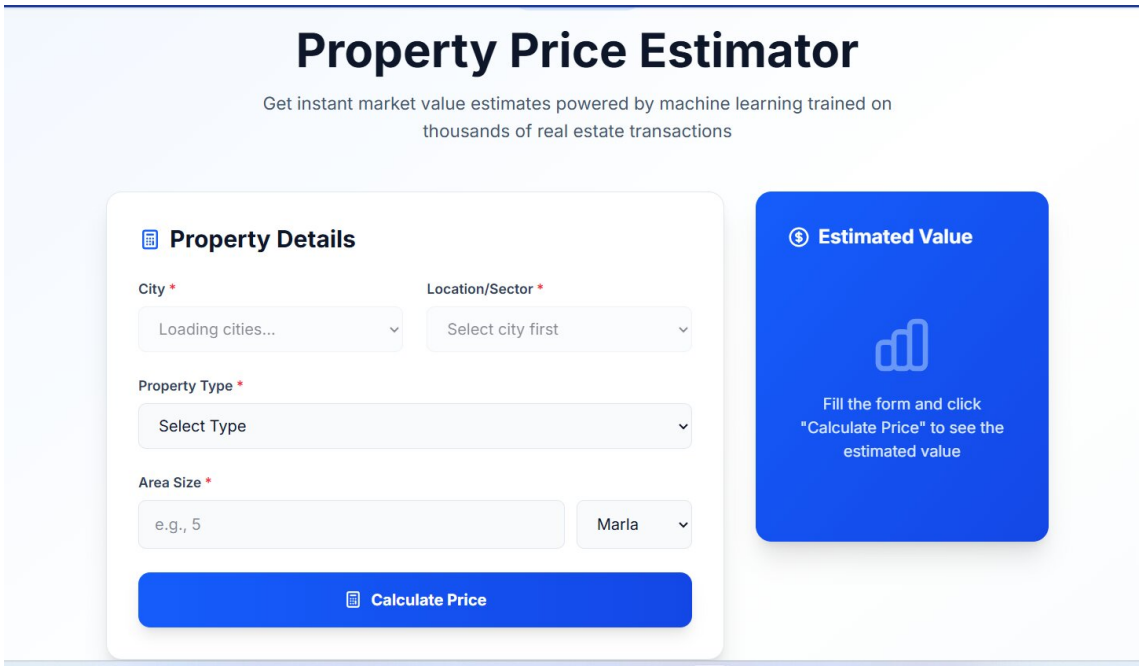

Figure 4.15: Price Prediction Form

4.8.9 Reviews Section

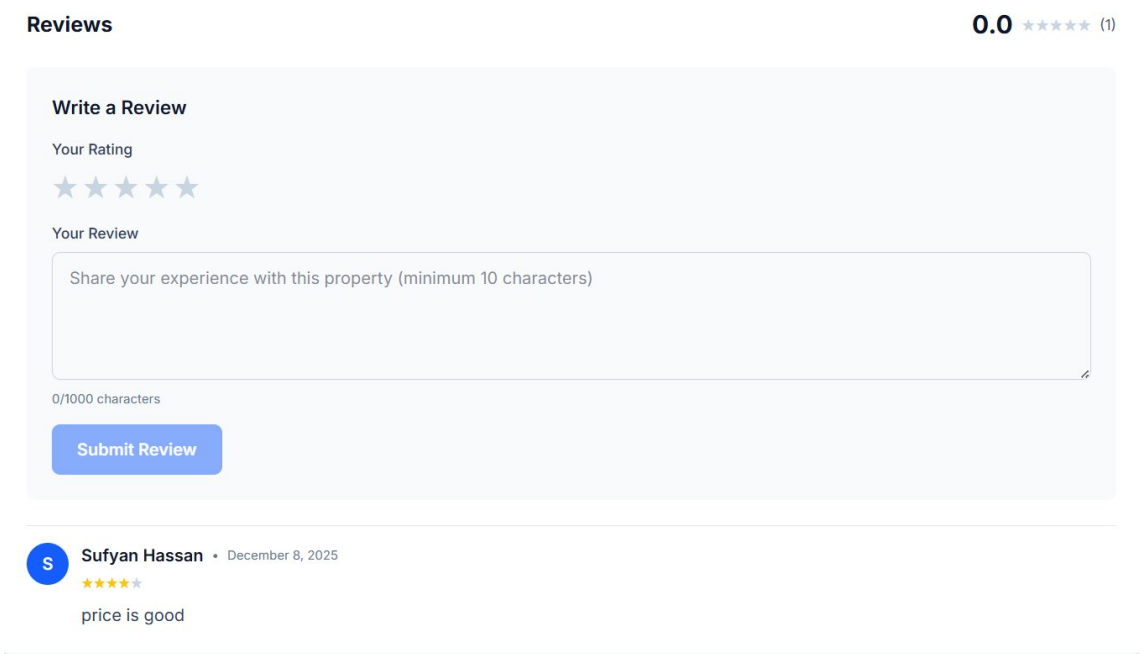

Figure 4.16: User Reviews and Ratings Section

4.8.10 Recommended Properties

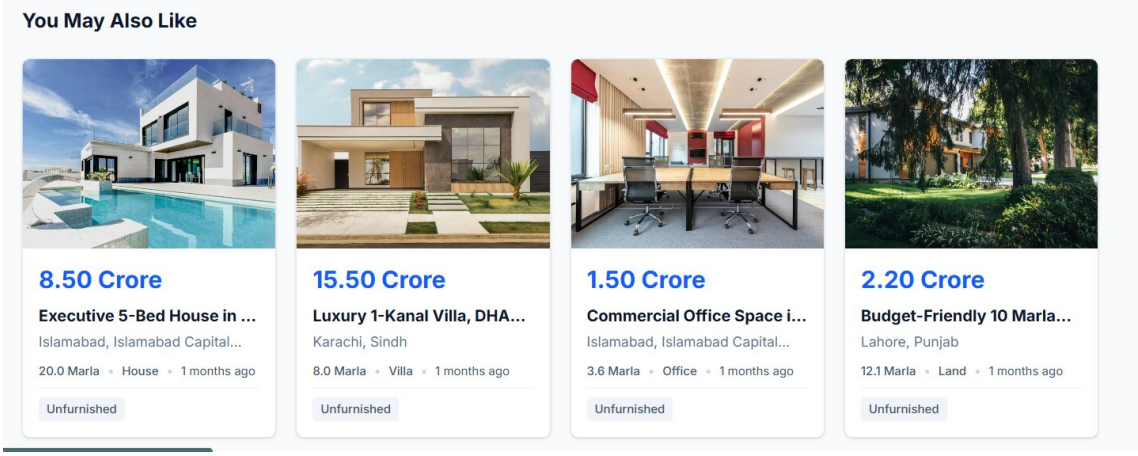

Figure 4.17: AI-Based Recommended Properties

4.9 External Interfaces

The Estate Nexus platform integrates with external systems using standardized electronic interfaces to extend functionality and enhance user experience:

- **AI/ML Services:** Connected via REST APIs for property price prediction and sentiment analysis of reviews.
- **Real-time Chat:** Implemented using WebSocket protocols to enable instant messaging between buyers and sellers.
- **Map and Location Services:** Provided by the Leaflet mapping library with OpenStreetMap data for property geolocation and area visualization.
- **Image Management:** Cloudinary API handles all property image storage, optimization, and delivery.

All external interfaces use proper authentication and secure communication protocols to ensure data protection, reliability, and system integrity.

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. This chapter should have the following sections:

4.10 System Architecture

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the

system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable.

4.11 Design Constraints

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made during the developing the system design.

4.12 Design Methodology

Summarize the approach that will be used to create and evolve the designs for this system. Cover any processes, conventions, policies, techniques or other issues which will guide design work. This is for deciding whether you will use structured, object-oriented or other specific methodologies. Most people will use some object-oriented technique with UML.

4.13 High Level Design

This section describes in further detail elements discussed in the Architecture. High-level designs are most effective if they attempt to model groups of system elements from a number of different views. Typical viewpoints are:

1. Conceptual or Logical: This view shows the logical functional elements of the system. Each component represents a similar grouping of functionality. For UML, this would be a component diagram or a package diagram.

2. Process: this view is the runtime view of the system. The components are threads or processes or distributed applications. In UML, this would be a process interaction diagram.

3. Physical: this view is for distributed systems. The components are physical processors that have parts of the system running on them. For UML, this would be a deployment diagram.

4. Module: this view is for project management and code organization. The components are typically files or directories. This picture shows how the directory structure of the build and development environment will be designed.

5. Security: this view typically focuses on the components that cooperate to provide security features of the system. It is often a subset of the Conceptual view.

4.14 Low Level Design

This section provides low-level design descriptions that directly support construction of modules. Normally this section would be split into separate documents for different areas of the design. For each component we now need to break it down into its fundamental units or modules. For an OO implementation in Java, our components would become packages. Then the low level design will take each package and break it down into its classes. For smaller systems, you may have a single UML class diagram that each module description refers to.

4.15 Database Design

The section should reveal the final design of all database management system (DBMS) files and the non-DBMS files associated with the system under development. Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (create, read, update, delete capability), data stores, outputs, aliases, and description.

4.16 GUI Design

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels.

4.17 External Interfaces

External systems are any systems that are not within the scope of the system under development. In this section, describe the electronic interface(s) between this system and each of the other systems and/or subsystem(s), emphasizing the point of view of the system being developed.

Chapter 5

System Implementation

The design and development of the Estate Nexus real estate platform incorporated cutting edge tools and methodologies to provide a seamless, responsive and intelligent user experience. The goal of the implementation phase was to transform the system design and architecture into an executable, web-based application. Focus was on modularity, maintainability, scalability and user friendliness.

A full-stack methodology was followed, leveraging React in the frontend and Node.js/Express in the backend, as well as various other technologies in relation to managing the database, real-time communication, and AI/ML attributes. Table 5.1 presents the main tools and techniques in the implementation.

Table 5.1: Implementation tools and methods for the web application

Component	Tools and methods
Frontend development	React.js; component-based UI implemented with reusable React components.
Backend development	Node.js with Express.js; MongoDB (access via Mongoose); RESTful API design for server–client communication.
Real-time communication	Socket.IO for bidirectional live chat and presence features.
AI & ML features	Custom recommendation algorithms and sentiment-analysis integration for review processing.

5.1 Back-end Development with Node.js and Express.js

The Estate Nexus backend is written in Node.js with the Express.js framework. Express is an asynchronous middleware-based environment that is ideal for building scalable RESTful APIs or for managing real-time services.

Responsibilities on the server side are user authentication, management of the property listings, the search and filter functionalities, active chats and interfacing with persistence storage. The endpoints are RESTful and respond with JSON payload to make the client interactions lightweight and predictable. Middleware is used for parsing requests, validating, authenticating (JWT), and handling files. Bcrypt hashing is used to secure passwords and signed JWTs are required for protected endpoints, enabling stateless session management that makes horizontal scaling easier.

5.2 Database Management with MongoDB

MongoDB was used as the primary datastore and accessed from the backend using Mongoose ODM. Its document-oriented structure can effectively represent the heterogeneous and extensible property listings. Significant collections include the following:

- **users**: user accounts information along with hashed credentials and user roles (buyer, seller);
- **properties**: listings metadata like coordinates, price, amenities, picture references and status flags;
- **conversations** and messages: message metadata, chat history of buying and selling communication;
- **reviews**: ratings and text reviews related to either properties or users.

This design lends itself well to fast iteration, denormalized read patterns for UI rendering, and straightforward sharding/replication for scale.

5.3 The Front-end with React

The front end is built with React and is fully responsive and component based. Modules-Based UI Construction The UI consists of reusable modules (property cards, search/filter controls, detailed views, dashboards) that help ensure consistent styling and predictable behavior. The application offers:

- a searchable property grid with multiple filter controls (location, price, number of bedrooms, type);
- homes include home pages with photos, features, captured sentiment measures and ways to get in touch;
- seller dashboards – users can home sellers to manage listings and inquiries.

We also use React hooks and the Context API for state management (global state), we use Axios/fetch for the async backend interaction which includes quite robust error handling and optimistic UI updates when it makes sense.

5.4 Challenges Associated with Property Data and Integration

Challenges were as follows:

- **Inconsistent data:** a variety of formats (measurement units for space, address standards, lists of services) necessitated creation of harmonization streams and validation at ingestion.
- **Real-time chat latency:** we developed connection pooling mechanisms and efficient message persistence to sustain low-latency socket.io connections for high traffic, and rate-limited from server side.
- **Access control:** Role-based authorization and scope-restricted APIs allow for buyers, sellers, and others to see just the right amount of data.

5.5 Review System and Sentiment Analysis

Estate Nexus features a review and rating system with AI-enhanced sentiment analysis to derive actionable insights from community feedback.

5.5.1 Review System Implementation

Users post reviews with a numerical rating (from 1 to 5) and they may optionally add textual comments. Each review has references to the author and the property which is the review's target. Server side validation ensures users cannot review the same property twice and it does perform certain content moderation on the reviews.

5.5.2 Sentiment Analysis Integration

We utilize an ABSA (Aspect-Based Sentiment Analysis) methodology. ABSA identifies aspects (e.g., location", price", "amenities") within a review and classifies the sentiment associated with each aspect, providing more detailed information than a single overall polarity.A pre-trained model (deployed as backend service) processes every review that is sent:

- Polarity: Positive / Negative;

5.5.3 Evaluation

The sentiment component was validated against a representative sample of real-world property reviews. Evaluation criteria included:

- **Correctness:** agreement of automated labels with human judgments;
- **Clarity:** whether aggregate sentiment metrics are interpretable to end users;
- **Completeness:** robustness to process all incoming review submissions without loss.

5.6 Sequence Diagrams

This section shows Sequence Flow Diagrams of features included in Estate Nexus.

5.6.1 Sign Up / Login

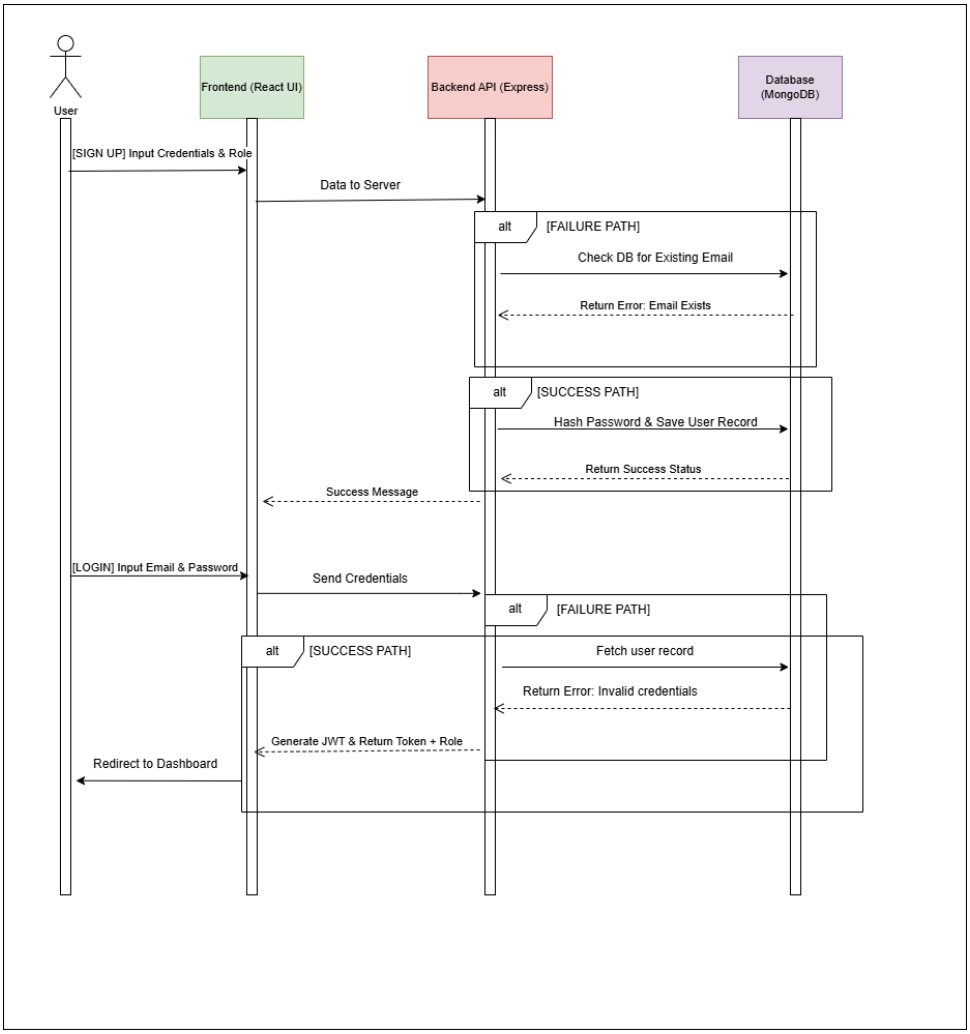

Figure 5.1: Sequence Diagram — Login and Sign Up

This sequence diagram shows the flow of how user can Sign up and login into the website and how it handles credentials and verifies their accuracy. It further determines the appropriate success or error response to send back to the client.

When the credentials are input by the user when signing up data is sent to the server then the email is matched in Database if there is same email already existing, if yes then sign up is failed and user is asked to enter new email. When the email is not matched then the user is successfully signed up.

For Login, the credentials entered by user is checked with the already signed up user. If it is found in database then the user is logged in, if not then an error message is displayed that user email or password is incorrect.

5.6.2 Seller's Property Management

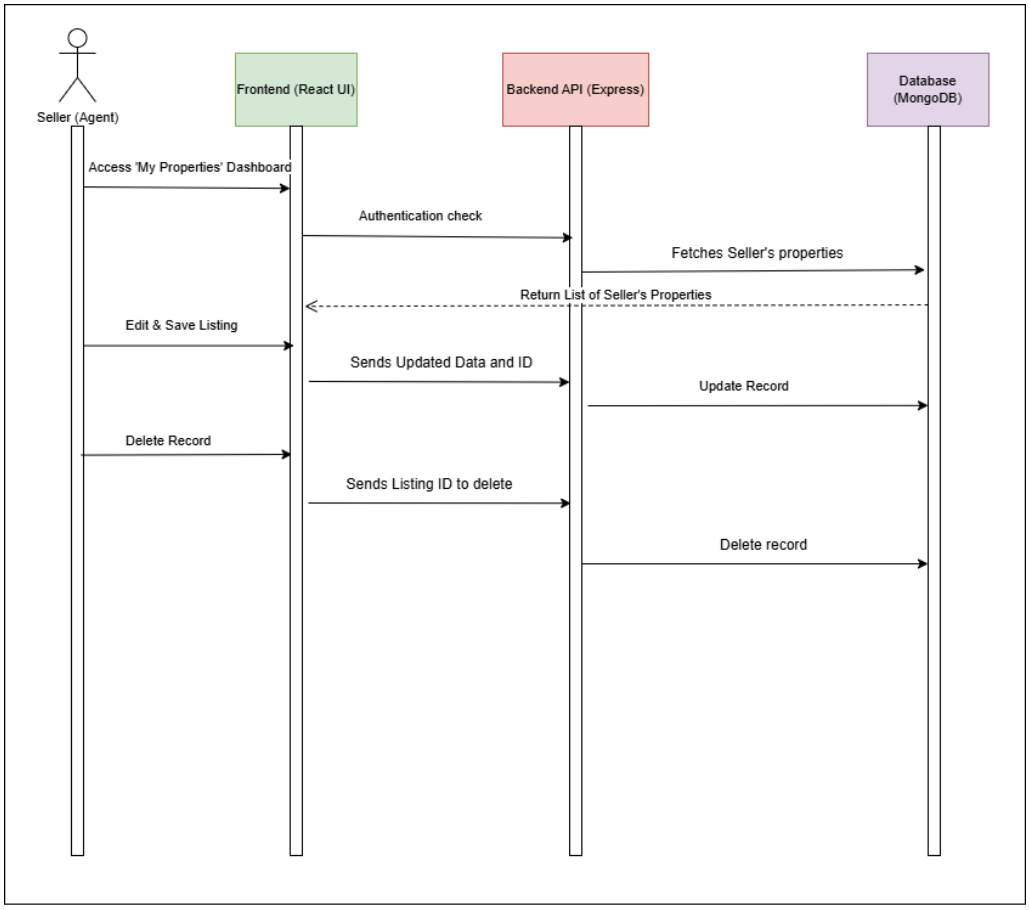

Figure 5.2: Sequence Diagram — Seller's Property CURD flow (Edit, Delete)

This diagram illustrates how the seller can Manage their properties. How their id will be verified and the properties that belong to them will be fetched .On their request they can than edit or delete their property.

After successfully adding the property by seller they can request to View, Edit or Delete their property. When the request is entered by seller its credentials are checked and after sellers authentication that yes the property matches the seller ID their request is accepted and they can accordingly edit or Delete their property.

5.6.3 Real-time Communication

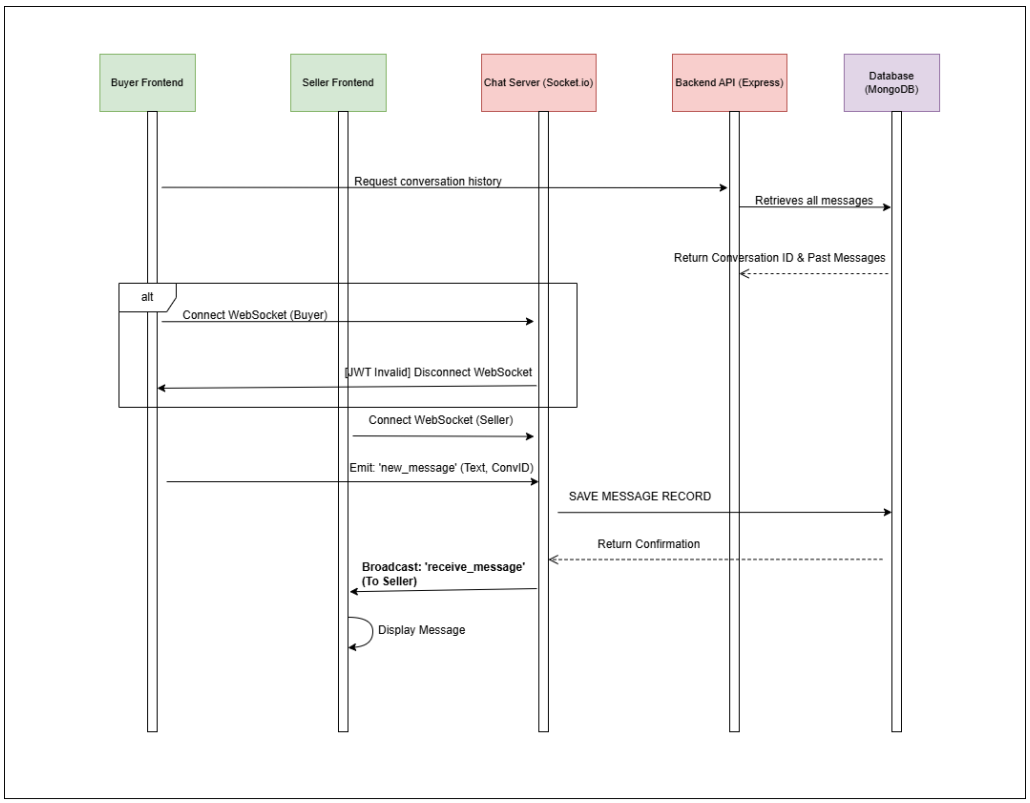

Figure 5.3: Sequence Diagram — Real-time Communicaion between Buyer and Seller

This Sequence Diagram shows the connection made with Socket.io server after requested by the user to start conversation with seller to send message. The user can select chat with seller button and request to chat with seller. After this request, from the backend history messages are fetched and request to message is sent to Database. Then the conversation ID is sent to Backend and also if there are any past messages between the buyer and seller already. The connection with Web Socket is built to enable buyer to real-time chat with seller and Web socket connection of seller is also made to make the communication between the two parties possible.

After the connection is built, the buyer can easily sent a message and then received by the seller. The message is then displayed on the chat interface.

The sellers online status is also checked. Buyer can see if their chat is read by the buyer and if the buyer is available at the moment or not.

5.6.4 Review and Sentiment Analysis

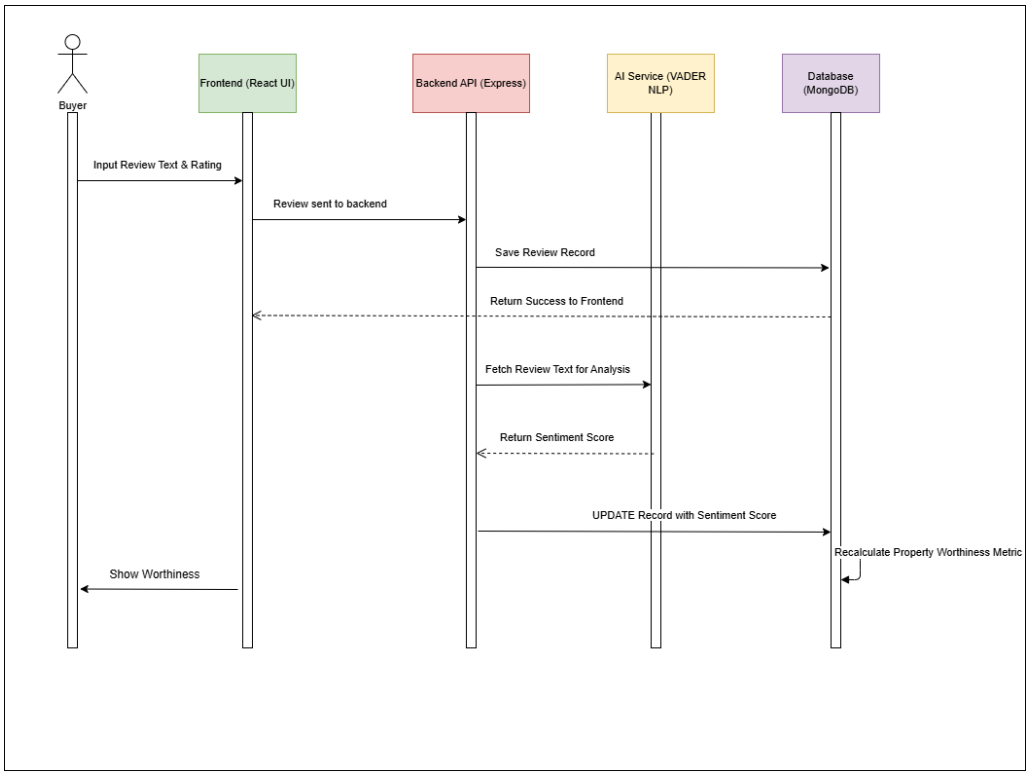

Figure 5.4: Sequence Diagram — Review Submission and Sentiment Scoring

This Sequence Diagram shows the flow of how a review is submitted, sent to backend and than sent it to the Sentiment Analyzer to process and tell if the review is negative or positive and that the property is worth buying or not.

The flow starts as a Buyer enters a review text with a rating on the Frontend (React UI). This data is then immediately sent to Backend API (Express) which persists the raw data by creating a review document in the MongoDB Database. When the record is saved, the database reports success back to the frontend, probably enabling the UI to refreshing or clearing the form of input.

At the same time or immediately after, the Backend interacts with an AI Service. It submits the review text to the service for linguistic processing, and the AI computes a Sentiment Score. Subsequently, the Backend updates the existing database record with this new score. The last step takes place at the database layer where a trigger or an internal process recalculates the Property Worthiness Metric using the updated data. Finally, this new worthiness information is sent back through the frontend and shown to the Buyer.

5.7 Class Diagram

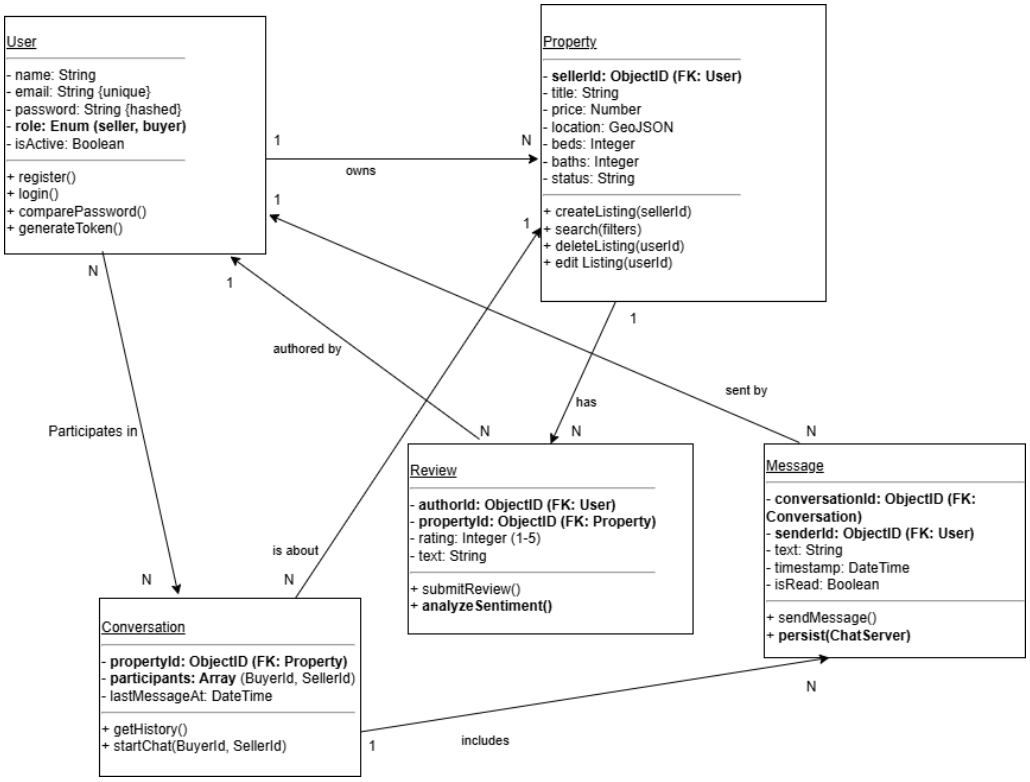

Figure 5.5: Class Diagram of Estate Nexus

Figure 5.5 shows a Class Diagram showing relationship between each class, their attributes, methods and required foreign key.

Chapter 6

System Testing and Evaluation

Testing was a critical stage in the Estate Nexus build to be certain all elements (including user authentication and advanced AI prediction models) ran smoothly and within the confines of requirements. The tests were oriented towards verifying the functional correctness, system integration, the end-user usability, and the ML outputs' reliability.

6.1 Graphical User Interface Testing

The focus of GUI testing is making sure that the UI of the application is simple to use, clean and has a consistent look and feel across all of the popular desktop browser environments.

6.1.1 Cross-Browser Testing

The interface is tested under Chrome, Firefox, Safari, and Edge for consistency in layout, responsiveness, and visual elegance (colors, typography) to provide the best UX.

6.2 Usability Testing

We tested the usability of our application by inviting users from our academic network (classmates) as well as from the non-technical world (family). This led to improvements of our workflow and design by nont echnical users.

6.3 Functional and Integration Testing

Integration testing, on the other hand, proved that all these communicating modules (frontend forms, backend APIs, and third-party services) work as a single cohesive system.

Table 6.1: Usability Testing Outcomes

Feature Tested	Objective	Outcome and Improvement
Listing Creation Flow	Verify ease of input for property details and image uploads	Improved: simplified required fields, optimized layout for readability, refined default color scheme for accessibility.
Property Search/Filter	Assess intuitiveness and speed of filtering and sorting	Improved: streamlined filter options, immediate search results display, verified accuracy of location-based searching.

6.3.1 Core CRUD Functionality (Sell/Manage Listings)

- **Create Listing:** Confirmed sellers can post listings with full property details. Listings are live immediately in the buyer search feed.
- **View/Edit/Delete:** Sellers can view, edit or delete listings, and any modifications are reflected in real time for buyers.
- **Separation of Accounts:** The security policies require that a user must only be able to edit or delete their own listings.

6.3.2 Integration of AI Components

- **Sentiment Analysis Trigger:** Analyze Sentiment is a button which reads the reviews and activates the sentiment analysis in order to refresh the AI Dashboard.
- **Recommendation Engine:** The selection of a property triggers the AI Recommendation Engine and returns similar listings.
- **Input Validation:** Validation and dropdowns are used to clean and format user input for AI models.

6.3.3 Geo-Location Accuracy

- **Map Integration:** Checked that the location of the chosen property on the interactive map is precisely where buyers are shown.

6.4 Software Performance Testing

Performance evaluation Performance evaluation was focused on the velocity and solidity of the software element.

6.4.1 Model Performance and Stability Testing

- **Price Prediction Benchmarking:** We vetted the model output against the observable market data to be confident that the valuations were in the ballpark.
- **Prediction Sensitivity Test:** Made sure that perturbations of inputs (like size of house/number of bedrooms) resulted in reasonable change in predicted price SOLD.
- **Load Testing:** They performed latency tests on AI model APIs to ensure they were responsive in a multi-user scenario.

6.5 Compatibility Testing

6.5.1 Cross-Browser Compatibility

During representative UI tests, the application was also tested on other browsers (Chrome, Firefox, Safari, Edge) for consistent look and feel behavior.

6.6 Exception Handling

Fail gracefully through comprehensive application-wide exception handling.

6.6.1 Data Validation and User-Facing Error Reporting

- **Input Validation and Feedback:** Check client and server side validation for required fields. Error messages are also specific, friendly, and are displayed close to the field.
- **Error Handling (AI Failure):** Simulate AI model failures: friendly messages without crashing app.
- **Authentication Exceptions:** Verify that non-authorized users cannot access protected routes (ie, Edit Listing page).

6.7 Load Testing

6.7.1 System Performance Under Load

- **API Response Times:** Tracked and optimized API latency (notably AI model endpoints) under synthetic concurrent user load.
- **Database Query Performance:** Validate that search queries and list retrievals have acceptable response times even with growing amounts of data..

6.8 Security Testing

6.8.1 Authentication and Authorization

- **Role-Based Access Control:** Check that users are only able to view and change data that is pertinent to their role (Buyer/Seller).
- **Protected Routes:** Verified that all unauthorized access attempts to protected pages (such as seller dashboard, edit listings) are redirected.
- **Session Management:** Test secure session management and logout.

6.9 Installation Testing

6.9.1 Deployment and Setup Verification

- **Environment Configuration:** Verified that all environment variables and configuration files are properly set up in deployment.
- **Setup Environment:** Confirm all environment variables and configuration files are correctly configured in the deployment.
- **Install Dependencies:** Make sure that all the dependencies are installed properly in the production environment.
- **Starting services:** Verify that all services (web server, database, AI model servers) start up properly and communicate as desired.

6.10 Test Cases

Here is the test cases executed for the sysyem. This guarantees that the system should perform as what are expected under various circumstances.

6.10.1 User Login

Table 6.2: Login Test Cases

Test Case ID	Description	Initial Condition	Status
T1.1	Test successful login for all user roles (Buyer, Seller)	Valid network connection, user has valid credentials	Pass
T1.2	Verify successful login redirects user to dashboard	Successful login credentials provided	Pass
T1.3	Test unsuccessful login attempts (incorrect password or username)	Application online	Pass
T1.4	Test input validation for Email/Username field	User enters malformed email	Pass
T1.5	Verify password input is masked during entry	N/A	Pass

6.10.2 User Sign-up

Table 6.3: Sign-up Test Cases

Test Case ID	Description	Initial Condition	Status
T2.1	Test successful registration for both roles	User provides unique email, name, phone, password	Pass
T2.2	Verify password minimum length enforcement	User enters password < 6 characters	Pass
T2.3	Test registration with existing email	Email already registered	Pass
T2.4	Verify role selection mandatory	N/A	Pass

6.10.3 Create Property Listing (Seller)

Table 6.4: Property Listing Test Cases

Test Case ID	Description	Initial Condition	Status
T3.1	Test creation of new property listing with all fields	Authenticated seller, listing form available	Pass
T3.2	Verify property price accepts only positive numbers	User enters letters or negative number	Pass
T3.3	Verify map sets address fields correctly	Geolocation service active	Pass
T3.4	Verify uploaded images are associated	Image upload service functional	Pass

6.10.4 Real-Time Messaging Functionality

Table 6.5: Messaging Test Cases

Test Case ID	Description	Initial Condition	Status
T4.1	Verify message delivery between two users	Buyer sends message, appears instantly on seller screen	Pass
T4.2	Verify message history persistence	User logs out and back in	Pass
T4.3	Verify unread message badge updates	New message received on different page	Pass

6.10.5 Buyer Interaction and Search Experience

Table 6.6: Buyer Interaction Test Cases

Test Case ID	Description	Initial Condition	Status
T5.1	Test empty search results display gracefully	Impossible filter combination	Pass
T5.2	Verify property detail page loads all data correctly	Buyer clicks listing	Pass
T5.3	Verify 'Contact Seller' button disabled for owner	Seller viewing own listing	Pass

6.10.6 AI Feature Functional Verification

Table 6.7: AI Feature Test Cases

Test Case ID	Description	Initial Condition	Status
T6.1	Verify Sentiment Analysis correctly scores positive review	Strongly positive review added	Pass
T6.2	Verify Sentiment Analysis correctly scores negative review	Strongly negative review added	Pass
T6.3	Verify prediction output changes logically with input changes	User modifies key features (e.g., bedrooms)	Pass
T6.4	Verify Recommendation Engine returns similar properties	Buyer views Property A, engine returns 80%+ feature match	Pass

Chapter 7

Conclusions

The above goals have been met in this project by designing and implementing a user friendly, scalable and feature rich property listing system catering to the needs of the sellers as well as the buyers. The platform creates an open market which sellers to list their property and manage their listings, while buyers have a simplified browsing and evaluation experience.

The main features that differentiate our platform from the others are a combination of three different AI modules: The AI Recommendation System helps increase user engagement by actively recommending similar properties, thereby speeding up buyer's decision making.

Sentiment Analysis Dashboard: Performs sentiment analysis on the reviews for a property so the sellers and buyers can know in an instant how desirable the property is, and Price Prediction Model provides a trustworthy prediction of the property value based on the real market data, which may serve as a neutral point to facilitate the fair negotiation.

In the process, we learned a lot about how one might leverage more modern full-stack techniques to expose Python-focused ML pipelines (Random Forest, VADER, Logistic Regression) in the context of a dynamic web application. The project is intended to demonstrate how data science can be integrated into a marketplace to provide intelligent tools that make the user experience better and help make the market more efficient.

7.1 Future Improvements

While the platform as it now is is an Minimum Viable Product (MVP), there are a few key areas of improvement for the next professional version that will enable us to enhance security, scalability, user experience, and income potential.

7.1.1 Advanced AI Integration and Validation

To refine the intelligent features:

- **Prediction model tuning scaling:** Replace static geographic lookup tables with a dynamic geocoding service; span all Pakistan in the dataset. Build models for continuous latitude/longitude anywhere in the world.
- **Prediction Model enhancement:** Upgrade from a simple Random Forest model to sophisticated models (e.g., GBM or NN) to obtain better price prediction.
- **Sentiment Calibration:** Adapt the VADER lexicon or employ a domain-specific LLM that is pre-trained on real estate language to get more accurate sentiment analysis on property reviews.

7.1.2 Security, Infrastructure, and Scalability

- **APIs for Authentication Verification:** Offer services with Google Maps, phone number verification (OTP), and identity verification (paid) for your buyer and seller accounts to maintain safety and legitimacy on your end and help keep fraud at bay.
- **Data Backup and Management:** Employ robust backup mechanisms and scale out storage specifically for the database to withstand a possible growth to millions of listings and user interactions.

7.1.3 Monetization and Payment Integration

- **Payment Gateway:** Connect with trustworthy third-party payment service providers (such as local financial services or mobile wallets like Easypaisa and JazzCash) for support of verified listings, premium accounts and the possibility to charge a commission from transactions.
- **Premium Ads:** Add paid tier level for sellers to boost property visibility, similar to that on major property portals.

References

[1] M. Walacik. Property appraisal via lens of property registration abundance – real estate market asymmetry assessment. *International Journal of Strategic Property Management*, 28(6):393–410, 2024. Cited on p. 1.

[2] W. K. O. Ho, B.-S. Tang, and S. W. Wong. Predicting property prices with machine learning algorithms. *Journal of Property Research*, 38(1):48–70, 2021. Cited on pp. 1 and 6.

[3] J. Hausler, J. Ruscheinsky, and M. Lang. News-based sentiment analysis in real estate: a machine learning approach. *Journal of Property Research*, 35(4):344–371, 2018. Cited on p. 1.

[4] M. Xu, H. Zheng, and J. Wu. Are online-only real estate marketplaces viable? evidence from china. *Journal of Real Estate Finance and Economics*, 68(4):583–623, 2024. Cited on p. 1.

[5] M. Bailey, R. Cao, T. Kuchler, and J. Stroebel. The economic effects of social networks: evidence from the housing market. *Journal of Political Economy*, 126(6):2224–2272, 2018. Cited on p. 1.

[6] L. M. Soegianto, A. T. Hinandra, P. A. Suri, and M. F. Nadril. Comparison of model performance on housing business using linear regression, random forest regressor, svr, and neural network. *Procedia Computer Science*, 245:1139–1145, 2024. Cited on pp. 2 and 6.

[7] Ajay Kumar, Ankit Yadav, Amit Raj, and Vijay Kumar. Logistic regression based sentiment analysis system: Rectify. *2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, 2024. Cited on p. 2.

[8] Y. Zhangling and M. Dai. A real-time group communication architecture based on web-socket. *International Journal of Computer and Communication Engineering*, 1(4):408–411, 2012. Cited on pp. 2 and 10.

[9] P. Kurlat and J. Stroebel. Testing for information asymmetries in real estate markets. *Review of Financial Studies*, 28(8):2429–2461, 2015. Cited on p. 6.

[10] A. T. Nursal, M. F. Omar, M. N. M. Nawi, M. S. Khalid, and N. A. M. Zaki. Online reviews sentiment analysis based on high-rise residential projects in malaysia. *Journal of Electronic Systems*, 20(3S):257–266, 2024. Cited on pp. 6 and 10.

[11] T. Xu, Y. Zhao, and J. Yu. A real estate price index forecasting scheme based on online news sentiment analysis. *Systems*, 13(1):art. 42, 2025. Cited on p. 6.

[12] J. Li, Y. Wang, and C. Liu. Spatial effect of market sentiment on housing price: evidence from social media data in china. *International Journal of Strategic Property Management*, 26(1):72–85, 2022. Cited on p. 7.

[13] F. Ullah, S. Sepasgozar, and C. Wang. A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms. *Sustainability*, 10(9):3142, 2018. Cited on p. 9.

[14] Hafiz Muhammad Usman Ali, Qaisar Farooq, Azhar Imran, and Khalil M. El Hindi. A systematic literature review on sentiment analysis techniques, challenges, and future trends. *Knowledge and Information Systems*, 67:3967–4034, 2025. Cited on p. 9.

[15] A. Gharahighehi, K. Pliakos, and C. Vens. Recommender systems in the real estate market—a survey. *Applied Sciences*, 11(16):art. 7502, 2021. Cited on p. 9.

[16] H. Zhang. Residential real estate price prediction based on adaptive loss function and feature embedding optimization. *Humanities and Social Sciences Communications*, 12:art. 832, 2025. Cited on p. 10.