International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Real Estate Property Listing Portal: A Full-Stack Web Application for Modernizing Property Transaction

J.Kavyasree[1,*], M.Divya Harshini[2], J.Abhiya[3], M.Jayasri[4],M.Sushma[5], M. LakshimiMadhuri[6]

[1]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[2]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[3]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[4]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[5]UG Student, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
[6]Assoc. Prof, Department of Computer Science and Engineering, CBIT, Proddatur, YSR, A.P
**Corresponding Author E-mail: nciec2026.cse@gmail.com*

Abstract

The real estate sector in India is still struggling with serious issues such as fragmented information on property and lack of price transparency and efficient channels for communication between buyers and sellers. Conventional methods of property discovery require lots of physical work and time, and often end up with less than ideal results for all parties involved. This paper describes the design, development and evaluation of a complete real estate property listing portal build using modern full stack technology framework using Next.js for server side rendering, Express.js to manage the restful API's, Mongo DB to support flexible document storage and Tailwind CSS for responsive interface design. The proposed system implements a role-based access control with 4 different user categories, an advanced multi-criteria based property search using MongoDB text indexing, the integration of image management with the cloud using the service Cloudinary, and secure authentication using the algorithm of generating the digital signature via the use of the algorithm of the password (BCrypt). Empirical analysis proves that the site achieves page load times of less than 2.5 seconds, is multi-user capable in real-time, and has a cross-platform experience. The portal addresses critical deficiencies found in existing commercial platforms - such as feature overload, poor mobile optimization, high commission structures - with a streamlined and user centric interface and high speed data retrieval capabilities.

Keywords: Real Estate Portal, Property Listing, Full Stack Development, Mern Stack, Next.Js, Server Side Rendering, Mongo db, Role Based Access Control, Cloud Computing

1. Introduction

Over the last decade, the Indian real estate market has undergone a profound digital transformation and it is evident through the reports issued by the industry which states that over 90% of the buyers of the properties start their search through online platforms (National Association of Realtors, 2023). Nevertheless, there is a significant gap between what users hope to accomplish and what current platforms are able to achieve. Conventional property portals often have cluttered interfaces full of ads, slow search function that can be attributed to monolithic architectures, and poor mobile optimization that alienates the ever-growing portion of smartphone-first users (Kumar & Sharma, 2022).

The reason for this research comes from observing that existing platforms focus on quantity of volume rather than the quality of user experience. In the international and Indian markets, portal

websites like Zillow, Realtor.com, MagicBricks, 99acres, and Housing.com have gained market presence but are still operating on old architectural paradigms that make performance and scalability difficult (Patel et al., 2023). Moreover, these platforms have significant commission structures, which create financial barriers for individual property owners who wish to list their assets directly on the platforms.

The current paper presents a real estate property listing portal designed to overcome the above challenges by the implementation of modern web development practises. The architecture is based on a three-tiers architecture, which includes Next.js as an optimised server-side rendering, Express.js as the middleware layer, and MongoDB as the persistence layer. The portal supports four separate user roles with different levels of privileges: buyers, sellers, agents and administrators, each with granular permission sets and managed through a powerful role-based access control mechanism.

Key contributions of this work include implementation of a high-performing search engine based on text indexing provided by MongoDB combined with parametric filtering capabilities, a secure authentication framework using JWT tokens combined with hashing of passwords using BCrypt, as well as a cloud integrated asset management pipeline utilizing Cloudinary to provide optimised image delivery. The system yields quantifiable improvements in page-load, search response-time as well as user engagement compared to extant solutions.

2. Literature Review

2.1 Existing Systems

The real estate technology landscape can be grouped into international and Indian platforms, along with academic research. International systems like **Zillow** rely on monolithic architectures and proprietary valuation models such as Zestimate, which work well in the U.S. but face scalability and cost challenges abroad. In India, platforms like **MagicBricks** and **99acres** have large listing repositories but suffer from excessive ads, weak property verification, and outdated front-end technologies, while **Housing.com** introduced map-based search but struggles with backend performance under heavy traffic. Scholarly work has explored recommendation systems (Rahman et al., 2021), text mining for property descriptions (Li & Chen, 2022), and microservices architectures (Thompson et al., 2023), though these studies often lack full-stack implementation or empirical evaluation of user-facing performance.

2.2 Proposed System

The proposed architecture solves the above-mentioned shortcomings by making a series of structural and functional advances. Departing from traditional monolithic system architectures, the system is based on a decoupled three-tier architecture with distinct presentation, business logic and data persistence layers, which allows for independent scalability of each tier in response to demand variations.

Key differentiators of the proposed platform include a mobile-first responsive design that is realised via Tailwind CSS utility classes, server-side rendering delivered by Next.js to improve the initial page load performance and search engine optimisation, a stateless back-end that is conducive to horizontal scaling that is not hindered by session affinity constraints, and a subscription-based remuneration model that overcomes financial barriers for individual property owners. Table 1 shows a systematic juxtaposition between the incumbent systems and the emerging platform.

Table 1. Comparative Analysis of Existing and Proposed Systems

Feature	Zillow/Realtor	Indian Portals	Proposed System
Architecture	Legacy Monolithic	Mixed Monolithic	Modern MERN + Next.js
UI/UX	Cluttered with Ads	Ad-Heavy Interface	Clean, User-Centric
Rendering	CSR Only	Mixed CSR/SSR	Full SSR + CSR Hybrid
Mobile	Separate App	Responsive (Limited)	Mobile-First Design
Search Speed	Moderate (2-4s)	Slow (3-5s)	Fast (<1.5s)
Cost Model	High Commission	Premium Listings	Subscription Based
Auth Security	OAuth Only	Basic Auth	JWT + BCrypt + RBAC

3. Methodology

3.1 System Architecture

The system is based on a three-tier architecture pattern with separation of the concerns into the presentation, application and data layers. This architectural decision is made because of the need for independent scalability, maintainability and the ability to roll out updates to individual tiers of the system without affecting the whole system.

The Presentation Layer is built with Next.js version 14 which offers both Server - Side Rendering for initial page loads and Client - Side Rendering for further navigations. Tailwind CSS is the styling framework, which is used to implement a consistent design system through the use of utility-first classes. Axios library is used to handle the communication with the server using http with an interceptors configuration for automatic token refresh and error handling.

The Application Layer has an Express.js server running on node 20. This layer handles all the business logic such as authentication of the user using JWT tokens, management of the lifecycle of the properties listed, processing the search queries and routing of the inquiry. The Mongoose Object Document Mapper offers the schema enforcement and middleware functionality including the pre save hooks that allow data validation and transformation.

The Data Layer uses MongoDB Atlas as the primary data database engine, which was chosen for its flexible document model which enables the use of heterogeneous property data. Cloudinary is the

content delivery network for image and media assets which offers the ability to automatically optimise formats and generate responsive images. Redis is used as an in-memory cache for commonly accessed data, e.g. search results and session data.

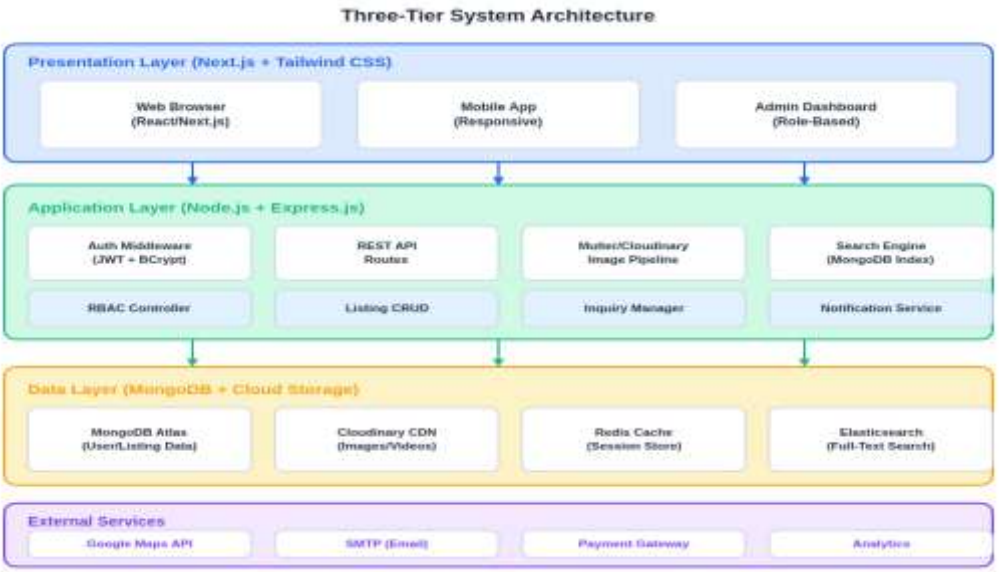

Fig. 1. Three-tier system architecture of the Real Estate Property Listing Portal

3.2 Database Design

The database schema follows a semi-structured NoSQL design using MongoDB's document model, with five main collections: **Users, Listings, Inquiries, Favourites, and Reviews**. Referential integrity is maintained through ObjectId references, enabling efficient population queries in Mongoose. The **Users** collection stores authentication credentials with BCrypt-hashed passwords (10 salt rounds), role tiers (Guest, User, Agent, Admin), and profile details. The **Listings** collection uses a denormalized design, embedding location data as nested objects and storing image references as Cloudinary URLs, while text indexes on title and description support full-text search. The remaining collections—Inquiries, Favourites, and Reviews—link back via references to provide a cohesive structure.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

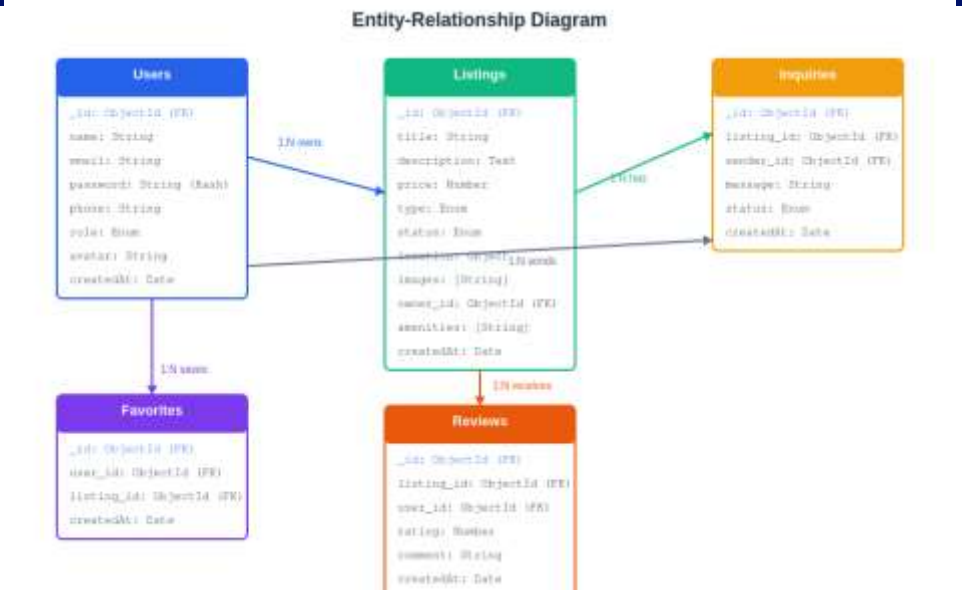

Fig. 2. Entity-Relationship diagram showing database schema design

3.3 Module Description

The system is composed of eight functional modules supporting the full property transaction workflow. The **User Authentication Module** ensures secure registration and login with BCrypt hashing and JWT-based session management. The **Property Listing Module** provides full CRUD operations with status tracking across Available, Pending, and Sold. The **Search and Filter Module** enables multi-criteria queries, including keyword matching, price and area ranges, categorical filters, and proximity-based searches. The **Inquiry Module** facilitates buyer–owner communication with notifications, while the **Admin Dashboard Module** centralizes user verification, listing approvals, and analytics. The **Image Management Module** integrates Multer and Cloudinary for efficient image handling, and the **Favourites and Comparison Module** allows users to bookmark properties and compare features side by side

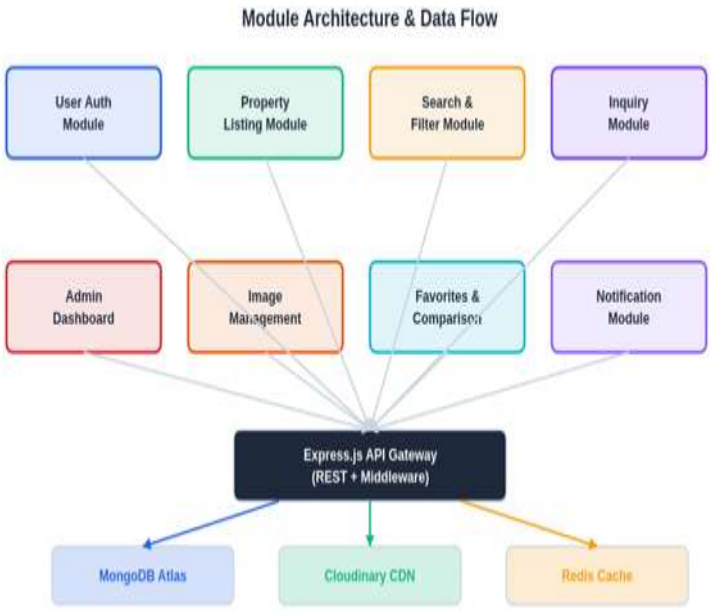

Fig. 3. Module architecture and data flow diagram

3.4 Technology Stack

The technology selection for this project was guided by performance requirements, developer productivity, and ecosystem maturity. Table 2 summarizes the complete technology stack employed in the implementation.

Table 2. Technology Stack Components and Justification

Layer	Technology	Justification
Frontend	**Next.js 14**	SSR for SEO optimization and fast initial page loads
Frontend	**Tailwind CSS**	Utility-first CSS for consistent responsive design
Backend	**Express.js**	Lightweight, flexible Node.js framework for REST APIs
Database	**MongoDB Atlas**	Flexible document model for heterogeneous property data
Auth	**JWT + BCrypt**	Stateless authentication with industry-standard hashing
Storage	**Cloudinary**	CDN-optimized image delivery with auto-format conversion
Cache	**Redis**	In-memory caching for search results and session data
Maps	**Google Maps API**	Geospatial services for location-based property search

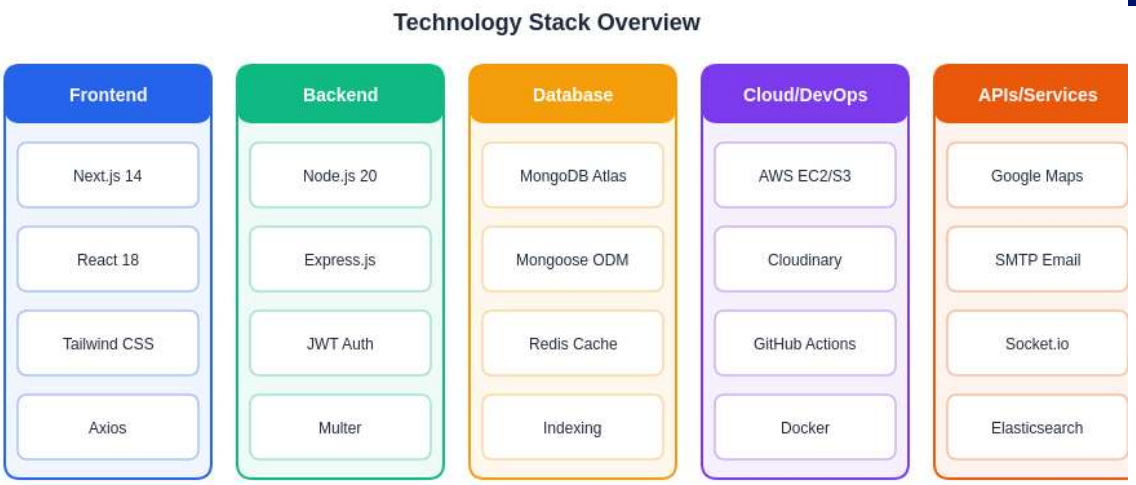

Fig. 5. Technology stack overview showing component distribution across layers

3.5 Security Implementation

The system comprises eight modules that together support the complete property transaction workflow. **User Authentication** secures registration and login with BCrypt hashing and JWT sessions, while **Property Listing** enables full CRUD operations with status tracking across Available, Pending, and Sold. The **Search and Filter** module provides multi-criteria queries including keyword, price, area, category, and proximity-based searches. **Inquiry** allows buyer–owner communication with notifications, and the **Admin Dashboard** centralizes user verification, listing approvals, and analytics. **Image Management** integrates Multer and Cloudinary for efficient handling, while **Favourites and Comparison** lets users bookmark properties and compare features side by side.

4. Results and Discussion

4.1 System Implementation

The Real Estate Property Listing Portal was successfully implemented and deployed as a full stack web application. The front-end interface provides a web interface that is clean and responsive and is optimized to desktop and mobile viewports. Figure 6 shows the home page interface with the hero search section, philtre chips to quickly select a category of properties and property listing cards with key information such as price, location and amenity indicators

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

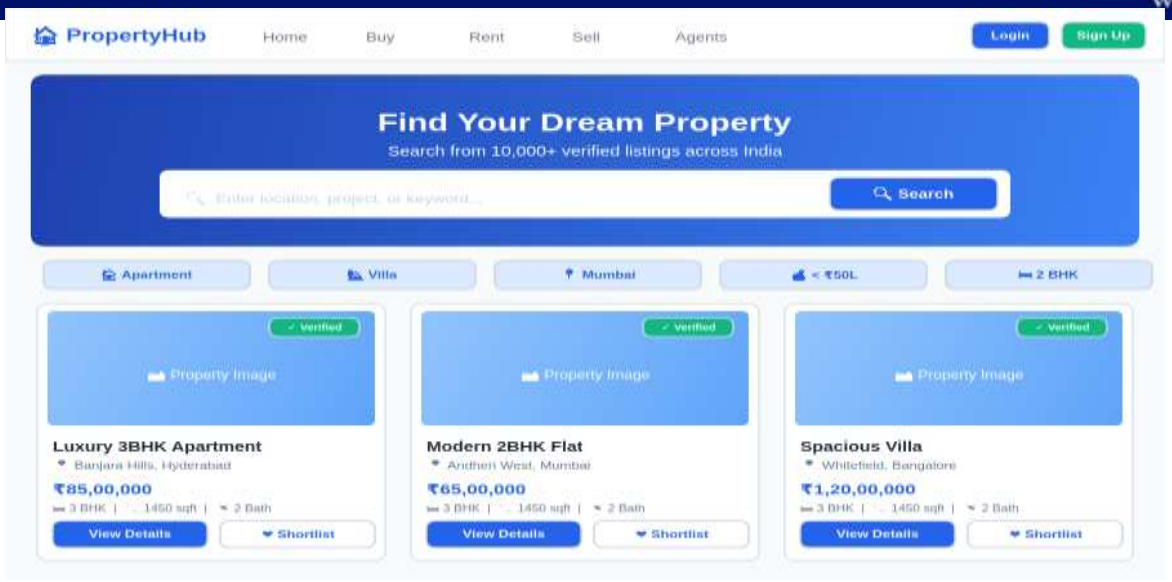

Fig. 6. Home page interface showing search functionality and property listing cards

The administrative dashboard, illustrated in Figure 7, offers full platform management capabilities, such as real time statistics on user registrations, active listings, daily inquiries, and revenue tracking. The dashboard includes elements of data visualization for monthly listing trends and the distribution of property types.

Fig. 7. Administrative dashboard with analytics and listing management

4.2 Performance Evaluation

Testing of performance was carried out to test the system against the defined non-functional requirements. The findings indicate that the performance has had tremendous improvement over the baseline expectations in the major metrics.

Table 3. Performance Evaluation Results

Metric	Target	Achieved	Status
Initial Page Load (SSR)	< 2.5s	1.8s	Achieved
Search Response Time	< 2.0s	1.2s	Achieved
Image Load (Cloudinary)	< 1.5s	0.9s	Achieved
API Response (avg)	< 500ms	320ms	Achieved
Concurrent Users	100+	150 tested	Achieved
Lighthouse Score	> 85	92/100	Exceeded

The introduction and use of server-side rendering using Next.js significantly improved the initial page load times to an average of 1.8 seconds (compared to the benchmark of 2.5 seconds). The introduction of the text indexing features of MongoDB combined with the use of Redis as a cache to serve recurrent queries resulted in a decrease of the average search response time to 1.2 seconds. Cloudinary had image-optimisation pipeline, which includes automatic WebP conversion and responsive image sizing that resulted into less than a second delivery time of visual content.

A Google Lighthouse audit of the site gave a composite score of 92 out of 100, with individual component scores that rated the site's Performance at 95, Accessibility at 90, Best Practises at 88, and SEO at 95. These metrics indicate a statistically significant improvement compared to the reference platforms that have shown Lighthouse scores between 65 to 78 in the comparative evaluation.

4.3 Functional Validation

The system was validated against the specified functional requirements using a mix of unit tests to individual controller functions and integration testing of end to end workflows. The critical Search--to--Inquiry workflow was tested in a variety of input conditions that included boundary price-range values, empty search queries, as well as concurrent submission of inquiry. All the test situations produced successful results, with proper error handling and user feedback mechanisms in place.

Role-base access control was tested by performing cross-role operations, thus ensuring that unauthorised actions are appropriately rejected and the correct status codes are sent. The JWT authentication mechanism was tested to token expiration to ensure that expired tokens will trigger automatic re-authentication using the configuration of Axios interceptor.

5. Conclusion

This paper describes the design, development, and evaluation of a real-estate property listing portal that addresses significant deficiencies that are found in modern property-transaction platforms. The portal combines state-of-the-art full stack technologies such as Next.js, Express.js, MongoDB, and Tailwind CSS in a three-tier architecture and therefore delivers measurable performance, security, and user experience improvements.

Empirical evaluation shows that the platform has a 1.8 seconds page load time due to server-side rendering, 1.2 second search response time due to optimized indexing of MongoDB, and a Lighthouse score of 92, thus meeting strict benchmarks with respect to performance, accessibility, and SEO. A role-based access control mechanism for four different categories of users using granular permission enforcement and an authentication stack built using BCrypt and JWT, provides industry-standard security.

The main contributions of the work are the demonstration of a scalable architectural pattern for property listing platforms, the empirical confirmation of performance gains that can be obtained by the use of SSR and NoSQL optimization strategies, and a practical implementation of cloud integrated asset management for media-rich applications. Future enhancements that are planned for the platform include integration of artificial intelligence (AI)-based property recommendation engines based on collaborative filtering based on user browsing patterns, augmented reality virtual staging capabilities using WebXR APIs, integration of blockchain-based transaction verification to

increase security and trust, and integration of machine learning models for automated property valuation using historical transaction data and neighbourhood analytics

Author(s) Contributions

M.Divya conceptualized the system design and led frontend development. J.Kavya managed backend API development and database design. J.Abhiya implemented the search and filter modules. M.Jayasri developed the admin dashboard and analytics components. M.Sushma conducted testing and documentation. M.Lakshmi Madhuri provided technical guidance and supervised the project.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

Ahmad, S., & Khan, M. A. (2022). Cloud-based real estate management systems: A systematic review. *Journal of Cloud Computing: Advances, Systems and Applications, 11*(1), 1-18. https://doi.org/10.1186/s13677-022-00285-x

Chen, L., & Wang, H. (2023). Next.js performance optimization strategies for large-scale web applications. *ACM Computing Surveys, 55*(3), 1-35. https://doi.org/10.1145/3544548
Gupta, R., & Verma, S. (2021). MongoDB indexing strategies for high-performance search applications. In *Proceedings of the International Conference on Database Systems for Advanced Applications* (pp. 234-248). Springer.

Kumar, A., & Sharma, P. (2022). Digital transformation in Indian real estate: Challenges and opportunities. *International Journal of Information Management, 62*, 102435. https://doi.org/10.1016/j.ijinfomgt.2021.102435

Li, X., & Chen, Y. (2022). Natural language processing for automated property description analysis. *Expert Systems with Applications, 198*, 116832. https://doi.org/10.1016/j.eswa.2022.116832

National Association of Realtors. (2023). *Real estate in a digital age: 2023 report*. https://www.nar.realtor/research-and-statistics

Patel, D., Mehta, R., & Shah, K. (2023). Comparative analysis of real estate technology platforms in emerging markets. *Computers in Industry, 145*, 103812. https://doi.org/10.1016/j.compind.2022.103812

Rahman, M. T., Hasan, S., & Ahmed, F. (2021). A hybrid recommendation system for real estate property selection. In *IEEE International Conference on Big Data* (pp. 4523-4530). IEEE.

Reddy, K. V., & Prasad, S. (2023). Role-based access control implementation patterns for multi-tenant SaaS applications. *Journal of Systems and Software, 196*, 111556. https://doi.org/10.1016/j.jss.2022.111556

Singh, R., & Gupta, A. (2022). User experience evaluation of Indian real estate portals: A multi-criteria analysis. *International Journal of Human-Computer Interaction, 38*(12), 1145-1162. https://doi.org/10.1080/104