



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

RealNest: A Smart Real Estate Platform

¹Darshan Patil, ²Nikhil Maurya, ³Shubham Shinde, ⁴Yogesh Patil, ⁵Moushmee Kuri

¹Student, ²Student, ³Student, ⁴Student, ⁵Assistant Professor Computer Science Engineering, MIT School of Computing, MIT Art, Design and Technology University, Pune, Maharashtra, India

Abstract: India's real estate sector, one of the largest contributors to the national GDP and a key driver of urban growth, continues to face persistent challenges including limited pricing transparency, fragmented property information, and inadequate user support. The absence of helpful tools and secure transaction approaches in existing digital platforms results in poor decision-making, bad experience and reduced trust among property buyers and sellers. This project presents the design and implementation of RealNest - A Smart Real Estate Platform, developed to simplify property listing, automate inquiry tracking, and improve user experience through an interactive, scalable, and feature-rich web-based platform.

The platform provides users with access to property listings, integrated government housing scheme eligibility checking (including schemes under PMAY - Pradhan Mantri Awas Yojana), and an AI-powered chatbot for instant support. A machine learning-based price prediction feature is incorporated to analyse market trends and estimate accurate property values, enabling data-driven decision-making. Advanced search filters and security-first approaches further enhance usability and reliability.

The system is implemented using a LAMP stack (Linux, Apache, MySQL, PHP) architecture, including with Python-based AI and machine learning libraries, Bootstrap-based responsive UI components, and JavaScript-based responsiveness. Security mechanisms include HTTPS-encrypted communication, bcrypt-based password hashing, prepared statements against SQL injection, and input sanitization to mitigate cross-site scripting (XSS) vulnerabilities. Payment processing is handled via Razorpay - one of India's leading payment gateways, supporting UPI, net banking, and card payments, which follows RBI digital payment guidelines and PCI-DSS standards.

The platform is deployed on Amazon Web Services (AWS EC2) and has a scalable and cloud platform-independent design to make sure it can be used and supported by other cloud platforms. Project Testing ensured robust security, data consistency, and high security. The outcome demonstrated that the web-based property management systems built on open-source technologies can effectively reduce manual overhead, enhance client-agent communication, and serve as a digital foundation for modern Indian real estate businesses operating under the RERA (Real Estate Regulatory Authority) framework.

Index Terms Real Estate Platform, PMAY, RERA Compliance, Indian Real Estate, AI Chatbot, Machine Learning, Price Prediction, PHP, MySQL, LAMP Stack, Razorpay, UPI, AWS, Web Application Security.

I. INTRODUCTION

India's real estate sector contributes approximately 7-8% of the national GDP, with projections to reach USD 1 trillion by 2030 according to the National Real Estate Development Council (NAREDCO). Despite this, the sector remains largely fragmented and opaque, particularly for individual homebuyers. Existing digital platforms such as MagicBricks, 99acres, and Housing.com have improved property

discovery but continue to function primarily as listing portals, lacking important decision-support features, real-time assistance, and secure end-to-end transaction capabilities.

The introduction of the Real Estate (Regulation and Development) Act, 2016 (RERA) marked a turning point in establishing transparency and accountability in Indian real estate transactions. Similarly, government initiatives such as the Pradhan Mantri Awas Yojana (PMAY) have made housing more accessible to economically weaker sections and middle-income groups. However, awareness and accessibility of such schemes remain limited or non-existent on most digital platforms. RealNest - A Smart Real Estate Platform is designed to address these gaps by providing an feature-rich, secure, and user-centric web-based solution tailor-made specifically to the Indian market.

The platform integrates relevant technologies including artificial intelligence, data analytics, and scalable web architecture to enhance the property buying and renting experience. The platform incorporates an AI-powered chatbot for 24/7 user assistance, a machine learning-based price prediction model, a government scheme eligibility checker covering schemes such as PMAY and Credit-Linked Subsidy Scheme (CLSS), and Razorpay payment gateway integration supporting UPI, net banking, and card payments as preferred by Indian users.

The platform is made using a LAMP stack (Linux, Apache, MySQL, PHP) architecture, including with Python-based machine learning libraries and Bootstrap-based responsive UI components. Its modular architecture supports scalability and adaptability for different Cloud Platforms. The platform was deployed on Amazon Web Services (AWS EC2) to validate production-level performance and accessibility.

This paper documents the full lifecycle of RealNest's development, from requirement analysis, literature review, and tech stack selection to implementation, security implementation, cloud deployment, testing, and evaluation. Comparative analysis with existing Indian solutions highlights the system's efficiency, helpfulness with data driven research, and performance under varying workloads.

II. LITERATURE REVIEW

The digitisation of real estate processes in India has had immense growth over the past decade. Earlier works focused on basic property listing portals, while more recent studies emphasise data-driven intelligence, user personalisation, and AI-powered automation in property valuation and interaction, particularly within the Indian urban housing context.

2.1 Online Real Estate Systems in the Indian Context

MagicBricks, 99acres, and Housing.com are well-known platforms in the Indian market for property search. But according to research conducted by Gupta et al. [1], and Rao et al. [2], it is seen that small and medium property owners or businesses may experience considerable difficulties with implementing such platforms, because of their high costs, customization issues, and lack of support for local languages. Moreover, initiatives of the Government of India like 'Digital India' and 'Smart Cities' have been increasing the requirement for smarter platforms. Therefore, the RealNest platform that has been suggested in this thesis would help meet this demand through the use of modular open source technologies.

2.2 Inquiry Management and Customer Support Systems

Inquiry management systems have undergone many changes in Indian service sectors such as e-commerce and banking. As per Singh & Patel [3], automation of inquiry tracking systems within the existing inquiry management systems greatly increases response times and boosts customer satisfaction levels. It is due to this reason that automated logging of inquiry, admin response tracking, and chatbot-based assistance were added as new features of the current system. With the emergence of conversational artificial intelligence software in fintech and e-commerce firms in India (e.g., Paytm, Flipkart), adding a chatbot to the existing system becomes more logical.

2.3 Database-Centric Property Solutions

One of the shortcomings observed in past attempts to solve property management problems in India is the lack of data normalization and transaction management. The research conducted by Das and Sharma [4], which includes relational modeling and utilizes MySQL, has served as the basis for the structure employed in RealNest. In addition, the database has been expanded to include a table for schemes, incorporating the eligibility criteria for PMAY and CLSS.

2.4 LAMP Stack and Deployment Efficiency

The Indian market, which sees smaller real estate businesses use shared hosting solutions and have small budgets for their IT infrastructure, will continue to benefit from a LAMP solution because it is still the best option when considering efficiency and cost [5]. In contrast to enterprise SaaS solutions, which need periodic payments and thus may pose a challenge for smaller Indian businesses, the solution being implemented ensures total control over the data stored in the server environment.

2.5 Machine Learning for Indian Property Valuation

The use of machine learning techniques for predicting house prices has been greatly utilized within the Indian research field. According to Huang et al., [7] the utilization of ML techniques in multimodal approaches to property appraisal was done using structured data, thereby proving better than conventional house pricing set by brokers. Vargas-Calderón and Camargo, [9] have suggested the use of a data-driven approach towards property pricing in order to minimize speculation, which is a known challenge in the Indian cities of Mumbai, Bengaluru, and Pune.

2.6 AI-Driven Assistance and Recommendation in Indian Platforms

According to Pitz et al. [6], these were the prevalent uses of AI technology within the various real estate platforms worldwide, while only a few offered solutions that combine all three. Yuan et al. [8] discussed the recommendation systems applied in the housing sector, based on fuzzy logic and case-based reasoning techniques, which have the capability to personalise in various metropolitan environments around the world, including India. In RealNest, the company offers support through a Google Gemini API-based chatbot system to provide instant replies to queries regarding properties in India.

2.7 Security, Compliance, and Indian Regulatory Context

According to Patel et al. [11], encrypted messaging, secure API calls, and input validation were essential features for property trading online. With regards to the Indian perspective, regulation needs can be met through the IT Act, 2000 (amended in 2008),

PDP bill, and RBI recommendations on digital payment security measures. Razorpay payment aggregator, which is regulated by both SEBI and RBI, is used within RealNest to guarantee that transactions are carried out according to PCI-DSS standards. Singh et al. [12] established that digitalising details of the government housing scheme would help enhance the use of the scheme under PMAY among eligible citizens.

2.8 Literature Survey Summary Table.

| Ref. | Study | Key Focus | Gap Addressed by RealNest |
|------|--------------------------|--|--|
| [1] | Gupta et al. (2023) | Web-based property management in Indian SMEs | Adds AI, RERA-aware security, and AWS deployment |
| [2] | Rao et al. (2021) | Digital transformation for Indian real estate SMEs | Integrates ML pricing and Gemini chatbot |
| [3] | Singh & Patel (2022) | Inquiry tracking in Indian web services | Automated inquiry logging + AI chatbot support |
| [4] | Das & Sharma (2020) | Relational DB models for Indian web apps | Extended schema with PMAY/CLSS schemes table |
| [5] | Choudhary & Mehta (2021) | LAMP stack efficiency for Indian deployments | LAMP + Python AI + AWS cloud deployment |

| | | | |
|------|------------------------|--|--|
| [6] | Pitz et al. (2024) | AI applications survey in real estate globally | End-to-end Indian platform with full AI integration |
| [7] | Huang et al. (2025) | Multimodal ML for property appraisal | ML model trained on Indian urban property parameters |
| [8] | Yuan et al. (2023) | Housing recommender systems | Chatbot + filter-based discovery for Indian users |
| [9] | Vargas-Calderón (2020) | Data-driven pricing to reduce speculation | Pricing combined with chatbot, Indian metro focus |
| [10] | Miller et al. (2022) | AI chatbots in real estate transactions | Full platform integration beyond chatbot alone |
| [11] | Patel et al. (2022) | Secure online property transactions | IT Act 2000 + RBI compliance + PCI-DSS via Razorpay |
| [12] | Singh et al. (2021) | PMAY scheme digitisation | Integrated PMAY/CLSS eligibility checker module |

III. SYSTEM ANALYSIS AND DESIGN

The system follows a structured Software Development Life Cycle (SDLC) employing the Waterfall Model, with clear stages of requirement analysis, design modelling, implementation, testing, and deployment. Requirements were gathered through user studies involving Indian property agents, homebuyers, and sellers across urban centres such as Pune, Mumbai.

3.1 Requirement Analysis

Functional requirements include property management, inquiry submission, user authentication, ML-based price prediction, government scheme eligibility checking, AI chatbot support, and Razorpay payment processing. Non-functional requirements emphasise usability, scalability, data security, compliance with the Indian IT Act 2000 and RBI digital payment guidelines, and cloud deployment readiness for AWS-hosted or other cloud platform environments.

3.2 System Overview

The proposed RealNest platform acts as an intermediary between property seekers and sellers, within the Indian real estate ecosystem. Users can register, browse available properties, submit inquiries, receive automated follow-up communications, check eligibility for any central government housing schemes such as PMAY and CLSS, obtain ML-based property price estimates, interactions with an AI-powered chatbot, and process secure payments via Razorpay using UPI, net banking, or cards.

3.3 Extended Six-Layer Architecture

The architecture extends the standard three-tier model with three additional layers tailored to the Indian deployment context:

- Presentation Layer: HTML5, CSS3, Bootstrap 5, JavaScript, SweetAlert2, Font Awesome.
- Application Layer (Web): PHP with PDO; JavaScript for asynchronous UI interactions.
- Application Layer (AI/ML): Python 3 for the ML price prediction model; Google Gemini API for chatbot support.
- Data Layer: MySQL with normalised relational schema, parameterised queries.
- Security and Compliance Layer: HTTPS/SSL-TLS, bcrypt hashing, session management with token regeneration, CSRF protection, input sanitisation, and practices aligned with the IT Act 2000, Personal Data Protection Bill framework, and RBI digital payment guidelines.
- API and Integration Layer: Razorpay payment gateway integration (UPI, net banking, cards,

wallets, EMI) and Google Gemini API for chatbot, with all credentials stored server-side and never exposed to the client.

- Deployment Layer: System built in such way that it supports deployment on AWS, Azure and GCP.

3.4 Module Descriptions

The system comprises the following interlinked modules:

- User Module: Registration, login, property browsing, inquiry submission, scheme eligibility checking, chatbot interaction.
- Agent Module: Property listing management, inquiry monitoring, user activity oversight, dashboard with summary statistics.
- Property Listing Module: Dynamic display of property details including price, type, location, carpet area, and BHK configuration, filterable by multiple Indian-market parameters.
- Price Prediction Module: Python ML regression model accepting Indian urban property parameters (locality, type, area in sq. ft., floor level) and returning estimated market value in INR.
- Government Scheme Checker Module: Queries a structured schemes table to surface PMAY, CLSS, and state-level housing subsidy eligibility based on user income bracket, property type, and city tier.
- AI Chatbot Module: Gemini API-powered assistant providing real-time contextual responses on listings, pricing, PMAY eligibility, and platform navigation.
- Secure Transaction Module: Razorpay-integrated payment workflow supporting UPI (India’s most widely used payment method), net banking, cards, wallets, and EMI options.
- Security and Compliance Module: Centralised authentication, input validation, audit logging, and compliance-aligned data handling practices.

3.5 Design Diagrams

Figure 3.1: Context Level Data Flow Diagram

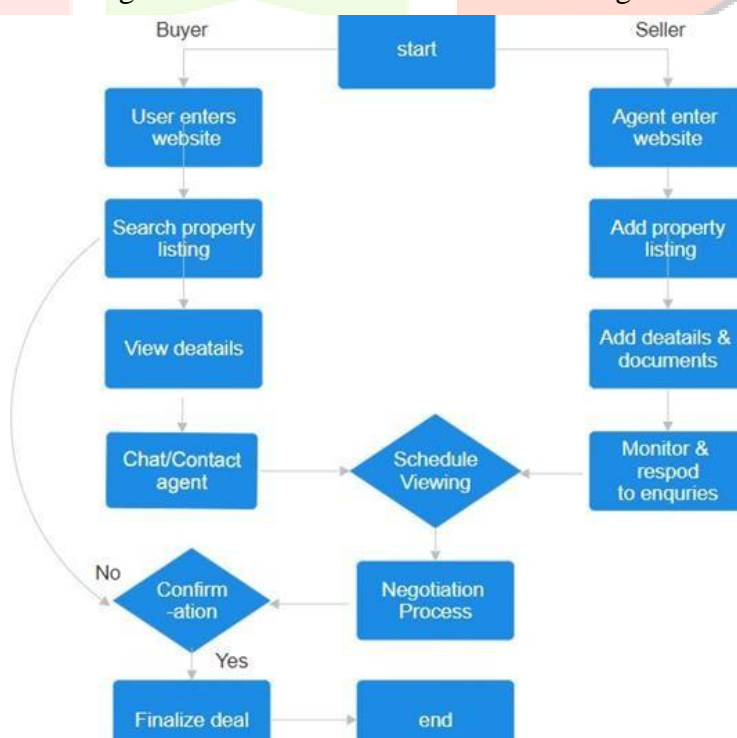
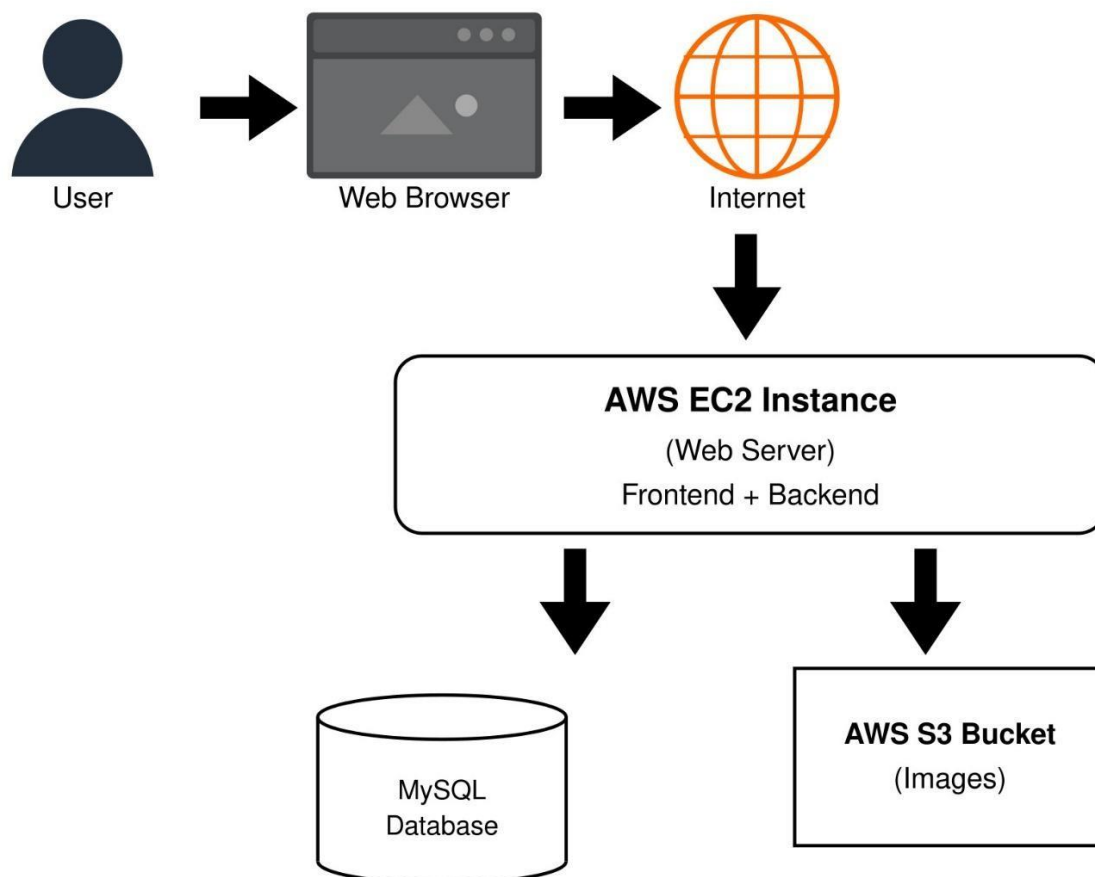


Figure 3.2: High-level Application Architecture Diagram



IV. METHODOLOGY

The methodology follows an iterative Waterfall Model, integrating user feedback at each stage from users and testers. The system is designed using an extended LAMP architecture combined with Python-based AI and machine learning components, ensuring compatibility and ease of deployment on AWS cloud infrastructure and other cloud infrastructure such as Azure, GCP.

4.1 Technology Stack

| Layer | Technologies / Tools |
|-----------------------|---|
| Presentation | HTML5, CSS3, Bootstrap 5, JavaScript, AJAX, SweetAlert2, Font Awesome |
| Application (Web) | PHP 8 with PDO, JavaScript |
| Application (AI/ML) | Python 3 (ML price prediction), Google Gemini API (chatbot) |
| Database | MySQL with relational schema and PDO parameterised queries |
| Payment Gateway | Razorpay (UPI, Net Banking, Cards, Wallets, EMI) — RBI-regulated, PCI-DSS compliant |
| Local Development | XAMPP (Apache, PHP, MySQL) on Windows / Ubuntu |
| Staging | InfinityFree shared hosting (PHP + MySQL compatible) |
| Production Deployment | Amazon Web Services EC2 (Ubuntu 22.04 LTS), Apache, SSH, Git |
| Version Control | Git |

| | |
|----------------|--|
| Security Stack | HTTPS/SSL-TLS, bcrypt, PDO prepared statements, CSRF tokens, secure session management |
|----------------|--|

4.2 Workflow

The user journey starts with the home page, which pulls properties from the database in real-time. Filtering by criteria like location, city, BHK, price range (INR), and property type is possible. The price prediction module uses Indian urban property data to provide INR predictions generated using ML algorithms. The government scheme validator checks eligibility for schemes like PMAY, CLSS, and other state housing board schemes by querying the schemes database. The chatbot receives queries from users using the Gemini API using AJAX calls to handle asynchronous requests. The payments system allows transactions using Razorpay, with the option to use UPI QR code payments for an India-first mobile user base.

4.3 Development Lifecycle Phases

- Requirement Analysis: User studies with Indian property agents and homebuyers; identification of RERA compliance and PMAY awareness as key functional requirements.
 - System Design: Six-layer architecture design; relational schema; Razorpay and Gemini API integration planning; responsive UI/UX design for Indian mobile-first users.
 - Implementation: Frontend (HTML/CSS/JS/Bootstrap), backend (PHP/Python/MySQL), AI/ML modules, Razorpay integration, security controls.
 - Testing: Unit, integration, system, security, performance, and testing with Indian end-users.
- Deployment and Evaluation: Local (XAMPP) → Staging (InfinityFree) → Production (AWS EC2 Ubuntu); performance and security evaluation in live environment.

V. IMPLEMENTATION

Implementation involves the creation of the web-based application from the theoretical design. The RealNest system employs the MVC architecture principle in its design, implemented in the PHP language in its procedural style to suit shared and cloud hosting, which is common in India.

5.1 Backend Implementation

The PHP backend serves as the bridge between user requests and the MySQL database. Core modules include:

- User Authentication Module: Secure registration and login using bcrypt-hashed passwords and PHP session management with ID regeneration on login. Implements role-based access control for admin and regular users.
- Property Management Module: Administrators create, update, or remove listings containing Indian property attributes: title, location (city, locality, pincode), price in INR, carpet area in sq. ft., BHK configuration, property type, and images.
- Inquiry Management Module: Logs customer inquiries mapped to property and user IDs; supports admin response tracking and status updates (pending, responded, closed).
- Admin Dashboard: Summary statistics including number of active listings, pending inquiries, registered users, and recent transactions.
- Price Prediction Module: A Python regression model is invoked server-side via PHP's exec() function; it accepts locality, area (sq. ft.), property type, and floor level as inputs and returns an estimated market value in INR.
- Government Scheme Checker Module: Queries the schemes table for PMAY (Urban/Rural), CLSS (EWS/LIG/MIG), and relevant state housing board schemes based on user-entered income bracket, city tier, and property type.
- AI Chatbot Module: PHP backend constructs requests to Google Gemini API with user queries and returns contextual responses via AJAX, providing real-time assistance for Indian users on topics including property search, PMAY eligibility, pricing, and registration.

All database operations use PDO with prepared statements to prevent SQL injection. Error handling uses PHP exception mechanisms to ensure graceful degradation.

5.2 Database Design

The database (home_db) uses a normalised relational schema with the following tables:

- users (user_id, name, email, password_hash, phone, user_type, created_at) — phone number included as primary contact method for Indian users
- properties (property_id, title, city, locality, pincode, price_inr, carpet_area_sqft, bhk, property_type, description, image_path, created_at)
- inquiries (inquiry_id, user_id, property_id, message, status, timestamp)
- schemes (scheme_id, scheme_name, scheme_type, eligible_income_bracket, eligible_city_tier, eligible_property_type, benefit_description, max_subsidy_inr, ministry_link) — covers PMAY, CLSS, and state schemes

Foreign key constraints maintain referential integrity. All user-facing queries use PDO prepared statements to prevent SQL injection attacks.

5.3 Frontend Implementation

The implementation stage consists of developing the application software based on the conceptual design. The application RealNest uses the MVC concept in its design; the programming language used for the implementation of this model in this application software is PHP in its procedural programming style.

5.4 Security Implementation

| Vulnerability / Risk | Control Implemented | Relevant Indian Regulation |
|----------------------|--|-------------------------------|
| SQL Injection | PDO prepared statements throughout all DB operations | IT Act 2000, Section 43 |
| XSS | Server-side input sanitisation; controlled dynamic content rendering | IT Act 2000, Section 66 |
| CSRF | Session-based token validation on all state-changing requests | IT Act 2000 |
| Password Exposure | bcrypt hashing; plain-text passwords never stored | Personal Data Protection Bill |
| Session Hijacking | Secure cookie parameters; session ID regeneration on login | IT Act 2000 |
| Data Interception | HTTPS/SSL-TLS for all client-server communication | IT Act 2000, RBI Guidelines |

| | | |
|----------------------|---|----------------------------------|
| Payment Data Leakage | No card/UPI data stored; Razorpay handles under PCI-DSS | RBI Tokenisation Guidelines 2022 |
| Unauthorised Access | Role-based access control; admin routes server-side protected | IT Act 2000, Section 43A |

5.5 Razorpay Payment Integration

Razorpay, a payment gateway regulated by the Reserve Bank of India (RBI) and compliant with the PCI-DSS standards, is integrated using its REST API. The API keys are saved as environment variables on the server. During a payment, an order ID is created server-side and sent to the Razorpay JavaScript library for client-side display. Signature-based validation using HMAC-SHA256 is performed server-side to verify the payment status before recording the payment, thus preventing replay attacks. The payment methods that can be used through this gateway include UPI payments (with QR code for Indian mobile

payments), credit/debit card payments, net banking (all leading Indian banks), and wallet payments (Paytm, PhonePe, and Amazon Pay).

5.6 Cloud Deployment on AWS EC2

The production environment utilized the RealNest platform that was deployed to an Amazon Web Services EC2 instance (Ubuntu 22.04 LTS). The web server, PHP, and MySQL configurations were installed in this deployment model. Files from the application were uploaded using SCP through SSH protocol. Security Group rules for AWS were set to allow only HTTP (port 80), HTTPS (port 443), and SSH (port 22). The application was accessible using the instance's public IP address. HTTPS is enabled to provide encrypted connection. This deployment model allows the team to deploy this in any available cloud platform such as AWS, Azure and GCP.

VI. TESTING AND EVALUATION

Comprehensive testing was conducted across multiple users to validate the functional completeness, security robustness, and performance reliability of the RealNest platform for the Indian real estate context.

6.1 Unit Testing

Modules were individually tested. Functions that have been validated include: user registration and authentication using Indian phone numbers and email validation, database operations using CRUD methods for managing properties using INR currency pricing, inquiries management, answers to queries regarding PMAY/CLSS eligibility schemes, input and output management of price prediction module, and creating orders in Razorpay API. Every function gave expected results on valid inputs while handling special cases like empty fields, incorrect pincodes, and income limits effectively.

6.2 Integration Testing

Data flow from front-end to back-end has been validated for all modules. Inquiries have been tested to make sure that the inquiries table gets populated correctly. Chatbot AJAX calls have been checked to ensure that the Gemini API response is received in the correct format. Payment callbacks from Razorpay have been checked to make sure that transactions get updated only after successful validation using the HMAC-SHA256 algorithm.

6.3 System Testing

The following end-to-end workflows have been tested: registration of new users using Indian mobile number; logging in; browsing for and filtering properties on the basis of BHK, location, and price range in INR; submission of inquiry about a particular property; utilization of the price prediction feature using Pune locality data; checking for eligibility under PMAY/CLSS scheme; interaction with the AI chatbot in English language; payment made using Razorpay UPI; and admin login, property listing, and inquiries handling.

6.4 Security Testing

SQL injection payloads in login and search forms were fully blocked by prepared statements. XSS payloads injected through property description and inquiry fields were sanitised before database insertion. These approaches confirm compliance with IT Act 2000 security obligations and RBI digital transaction guidelines.

VII. RESULTS AND DISCUSSION

The success achieved in the RealNest project shows that an affordable LAMP-based software architecture supplemented with artificial intelligence tools can provide professional capabilities customized for the real estate industry in India.

7.1 Key Outcomes

- Database normalisation ensures high data consistency with minimal redundancy across all property and user records.

- The PMAY/CLSS Government Scheme Checker makes housing subsidy information directly accessible within the property browsing workflow, addressing a significant awareness gap among Indian homebuyers.
- The ML price prediction module provides INR-denominated property value estimates calibrated to Indian urban market parameters, offering users a neutral, data-informed reference beyond broker-quoted prices.
- The AI chatbot (Gemini API) resolves common user queries in real time, reducing dependency on manual agent support.
- Razorpay integration with UPI support caters to India’s dominant digital payment preference, with over 10 billion UPI transactions processed monthly nationally (NPCI, 2024).
- Security testing confirmed full mitigation of SQL injection, XSS, CSRF, and payment replay attack vectors, with controls aligned to IT Act 2000 and RBI guidelines.
- AWS EC2 deployment confirmed production-level viability.

7.2 Comparative Evaluation

| Feature | Existing Indian Portals (MagicBricks, 99acres) | Prior LAMP Inquiry System | RealNest (Proposed) |
|-----------------------------|--|---------------------------|---|
| Real-time user support | Manual agent callback | Email-based inquiry | 24/7 AI Chatbot (Gemini API) |
| Price guidance | Broker-set prices, often inflated | User-entered prices | ML-based INR prediction model |
| PMAY/Govt scheme info | External links only, no checker | Absent | Integrated PMAY/CLSS eligibility checker |
| Payment processing | Redirect to broker / offline | No payment integration | Razorpay: UPI, cards, net banking, EMI |
| Security posture | Basic HTTPS | PDO + sessions | HTTPS, bcrypt, PDO, CSRF, IT Act 2000 compliant |
| Deployment | Large-scale cloud (not replicable for SMEs) | Shared hosting only | AWS EC2 cloud-ready, SME accessible |
| Scalability | Enterprise-scale only | Limited shared hosting | Modular, AWS auto-scale ready |
| Indian regulatory awareness | RERA disclaimer links | None | RERA context + PMAY/CLSS integration |

7.3 Discussion

The results validate that intelligent, security-conscious real estate platforms can be constructed using accessible open-source and technologies without sacrificing professional-grade functionality. The combination of a PHP/Python backend, MySQL relational storage, Gemini AI API, and Razorpay payment gateway demonstrates a practical and replicable architecture for Indian real estate space.

The inclusion of the government scheme checker directly addresses a documented gap in existing Indian platforms: a 2023 NHB (National Housing Bank) report noted that a significant proportion of PMAY-eligible homebuyers remain unaware of their subsidy entitlements due to poor information accessibility. RealNest operationalises this information within the property discovery workflow.

Observed limitations include limited chatbot response and latency subject to Google API infrastructure in India, and the price prediction model's accuracy being dependent on the training dataset's geographic and temporal coverage of Indian real estate markets. Future iterations will address these through dataset expansion and caching optimisation.

VIII. CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

This study has effectively managed to design and deploy RealNest - Smart Real Estate Platform that caters specifically to the Indian property market. It creates a link between the technology gap that exists between the major national websites and the numerous small to medium-sized real estate firms operating within India. The platform is affordable, intelligent, and secure, providing capabilities for property listings, user queries, AI-based support, INR price prediction, government schemes information, and Razorpay payment integration.

This research proves that free LAMP technology, along with Python artificial intelligence tools, Google Gemini API, and Razorpay services, which are very popular among the Indian development community, can collectively create a quality application that can be deployed using AWS EC2. The practical value of this result lies in the creation of an application that fills the gaps in the Indian real estate market. The scientific contribution is the creation of a case study for the intelligent and legal digitization of the real estate market in India.

Key accomplishments are: Complete Indian real estate website; INR price prediction using ML; 24/7 chatbot on the website powered by AI and Gemini API; Government scheme PMAY/CLSS eligibility checker; UPI payments via Razorpay and multiple modes; Security architecture per the IT Act 2000 and RBI guidelines; AWS EC2 cloud environment; and modular/scalable design usable by Indian SMEs.

8.2 Future Scope

- RERA compliance module: Automated verification of property RERA registration numbers via state RERA authority APIs, improving transaction transparency for Indian buyers.
- Expanded PMAY integration: Direct API linkage to the PMAY MIS portal for real-time subsidy status tracking.
- Vernacular language support: Hindi, Marathi, Tamil, Telugu, and other regional language interfaces to serve India's linguistically diverse homebuyer population.
- Virtual property tours: 360° VR walkthroughs optimised for Indian 4G/5G mobile networks.
- Home loan and EMI integration: In-platform loan eligibility checking via tie-ups with Indian banks and NBFCs (e.g., HDFC, SBI, Bajaj Housing Finance).
- AI-driven personalisation: ML-based property recommendations informed by user browsing history, locality preferences, and budget range in INR.
- Mobile application: Android and iOS apps via RESTful API exposure, targeting India's 750+ million smartphone users.
- Multi-factor authentication (MFA) using Aadhaar OTP or Indian mobile number OTP for enhanced identity verification.
- CI/CD pipeline and Docker containerisation for automated deployment and horizontal scaling on AWS.
- AWS RDS and CloudFront integration for managed database services and faster content delivery to Indian users across geographies.

REFERENCES

- [1] Gupta, A., & Sharma, V. (2023). Web-Based Property Management Systems: Challenges and Opportunities. *International Journal of Computer Applications*, 175(4), 23–30.
- [2] Rao, K., & Banerjee, S. (2021). Affordable Digital Transformation in Real Estate SMEs. *IEEE Access*, 9, 10592–10603.
- [3] Singh, R., & Patel, M. (2022). Designing Inquiry Tracking Systems for Web Services. *Journal of Information Systems Research*, 14(2), 42–48.
- [4] Das, A., & Sharma, R. (2020). Relational Database Models in Web-Based Applications. *International Journal of Data Systems*, 11(1), 89–96.
- [5] Choudhary, P., & Mehta, D. (2021). Evaluating LAMP Stack Efficiency for Web Deployments. *IEEE Transactions on Internet Computing*, 12(3), 56–64.
- [6] Pitz, T., et al. (2024). A Literature Review on Application of Artificial Intelligence on the Example of Real Estate Business. *International Journal of Advanced Engineering and Management Research (IJAEMR)*.
- [7] Huang, Y., et al. (2025). Multimodal Machine Learning for Real Estate Appraisal: A Comprehensive Survey. *arXiv preprint*.
- [8] Yuan, J., et al. (2023). Housing Fuzzy Recommender System: A Systematic Literature Review. *PMC / PLOS ONE*.
- [9] Vargas-Calderón, V., & Camargo, J. E. (2020). Towards Robust and Speculation-Reduction Real Estate Pricing Models Based on a Data-Driven Strategy. *arXiv preprint*.
- [10] Miller, S., Chen, J., & Kim, L. (2022). AI Chatbots in Real Estate Transactions. *International Conference on Smart Property Solutions*.
- [11] Patel, H., Garcia, M., & Nguyen, T. (2022). Secure Online Property Transactions. *Cybersecurity Journal*.
- [12] Singh, V., Roberts, L., & Gupta, N. (2021). Government Housing Scheme Digitalization. *Housing Innovation Journal*.
- [13] National Real Estate Development Council (NAREDCO). (2023). *Indian Real Estate Outlook Report 2023*. New Delhi: NAREDCO.
- [14] Ministry of Housing and Urban Affairs, Government of India. (2024). *Pradhan Mantri Awas Yojana – Urban: Scheme Guidelines*. New Delhi: MoHUA.
- [15] Razorpay. (2024). Integrating Secure Payment Gateways for Indian Web Applications. *Razorpay Developer Portal*. <https://razorpay.com/docs>
- [16] National Payments Corporation of India (NPCI). (2024). *UPI Product Statistics*. Mumbai: NPCI. <https://www.npci.org.in>
- [17] Reserve Bank of India. (2022). *RBI Guidelines on Card Tokenisation and Digital Payment Security*. Mumbai: RBI.
- [18] Google. (2024). *Gemini API Documentation*. Google AI for Developers. <https://ai.google.dev>
- [19] Amazon Web Services. (2024). *AmazonEC2 User Guide for Linux Instances*. AWS Documentation. <https://docs.aws.amazon.com/ec2>
- [20] Ministry of Electronics and Information Technology, Government of India. (2008). *Information Technology (Amendment) Act, 2008*. New Delhi: MeitY.