RentIt: A Cloud-Based Property Rental System

Apurva Mane[1], Purva Patil[2], Nikita Patkar[3], Sonal Kadam[4]
[1,2,3]*Department of Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, India*
[4]*Assistant Professor, Department of Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, India*

***Abstract*—The traditional property rental process is largely broker-driven, manual, and fragmented, resulting in high costs, low transparency, and inefficient communication between tenants and property owners. To address these challenges, this paper presents RentIt, a cloud-based intelligent property rental system designed to automate and modernize rental management. The proposed platform integrates role-based authentication, AI- based document verification using Optical Character Recognition (OCR), Machine Learning-driven rental price prediction using Random Forest regression, secure online payment gateways, and real-time communication modules. Additionally, the system enhances user experience through interactive 3D property visualization and geospatial mapping integration. Built on a scalable cloud-native architecture using modern web technologies and Firebase services, RentIt ensures data security, real-time synchronization, and seamless deployment. The system eliminates intermediaries, reduces rental costs, improves transparency, and provides a reliable digital ecosystem for tenants and property owners. Experimental evaluation of the ML model demonstrates effective rental price estimation using MAE, RMSE, and R^2 metrics. The proposed solution contributes toward the digital transformation of the real estate rental sector.**

***Index Terms*—Cloud Computing, Property Rental System, Artificial Intelligence, Machine Learning, 3D Visualization, SaaS Architecture**

I. INTRODUCTION

Urbanization has led to a steady increase in the demand for rental housing, particularly in metropolitan regions where mobility for education and employment is high. Despite technological progress in other sectors, the property rental ecosystem still relies heavily on brokers, manual documentation, and fragmented communication channels. As observed in existing rental management platforms [1], [3], many systems provide basic listing and booking functionalities but do not fully eliminate inefficiencies such as high brokerage fees, delayed tenant verification, and lack of transparency in rental agreements. Property owners also face challenges in screening tenants and managing records efficiently, as highlighted in recent apartment management studies [4].

With the emergence of cloud computing, scalable and distributed systems have become feasible for real-time service delivery. The NIST definition of cloud computing emphasizes on-demand availability, resource pooling, and rapid elasticity as core characteristics of modern digital systems [6]. Furthermore, large-scale cloud architectures have demonstrated their ability to support scalable web applications with high availability and reduced infrastructure overhead [7]. These advancements provide an opportunity to redesign traditional rental workflows into centralized and cloud-native platforms. In addition to cloud infrastructure, intelligent automation techniques can further enhance reliability and trust in digital rental ecosystems. Role-Based Access Control (RBAC) models provide structured authorization mechanisms that restrict system access based on defined user roles [8], ensuring secure interactions between tenants, property owners, and administrators. Document authenticity verification can be automated using Optical Character Recognition (OCR) technologies, which have evolved significantly in document analysis applications [9]. Moreover, machine learning-based recommendation and prediction models, such as matrix factorization techniques [10], have been widely applied in recommender systems to improve decision-making processes. Similar approaches have been explored in

rental portals incorporating collaborative filtering and visualization frameworks [2].

Motivated by these technological advancements, this paper presents *RentIt*, a cloud-based intelligent property rental management system designed to integrate secure authentication, automated tenant verification, predictive rental price estimation, and interactive property visualization within a unified architecture. The platform incorporates role-based authentication, AI-driven document validation, Random Forest-based rental prediction, secure online payment processing, real- time chat communication, and geospatial property mapping. Unlike many existing systems that focus on isolated features, RentIt centralizes the entire rental lifecycle—from listing and verification to agreement generation and payment—within a scalable cloud-native framework.

By minimizing manual intervention and reducing reliance on intermediaries, the proposed system aims to improve transparency, lower operational costs, and enhance trust between stakeholders.

II. LITERATURE REVIEW

Existing rental management solutions can generally be grouped into mobile-oriented platforms, conventional web portals, and feature-enhanced systems incorporating recommendation or visualization components. Mobile-based rental applications have improved accessibility and user reach; how- ever, several implementations rely on tightly coupled backend infrastructures that may restrict scalability and performance under growing user demand [1].

Traditional web-based rental portals have succeeded in digitizing property listings and facilitating communication between tenants and landlords. Studies such as [3] demonstrate how centralized web systems streamline booking and record management. Nevertheless, these platforms often depend on monolithic architectures and lack distributed cloud integration, which limits their adaptability and long-term scalability.

Recent research has introduced more advanced features such as recommendation algorithms and augmented visualization to enhance property discovery. For instance, collaborative filtering and AR-based visualization techniques have been explored to improve user decision-making and search accuracy [2]. Similarly, real-time apartment management systems have focused on improving transaction handling and billing automation [4]. While these contributions advance usability and system interaction, critical processes such as identity verification and secure authentication remain partially manual or externally managed.

Furthermore, emerging student-focused rental ecosystems incorporate geolocation services and filtered search mechanisms to simplify accommodation discovery [5]. However, many of these systems treat verification, authentication, and payment workflows as separate modules rather than integrating them into a cohesive architectural framework.

Overall, the existing body of literature indicates a degree of architectural fragmentation across rental platforms. Most systems emphasize either listing management, recommendation logic, or billing automation in isolation. Few solutions combine role-based access control, structured document management, managed authentication services, and scalable cloud-native deployment within a unified architecture. To address these gaps, RentIt adopts a distributed design built on Firebase-managed services and stateless RESTful APIs, enabling secure, scalable, and fully integrated rental lifecycle management.

III. SYSTEM DESIGN

The proposed platform, *RentIt*, is conceived as a cloud- native rental management system that streamlines and integrates the entire property rental lifecycle within a single digital environment. In contrast to conventional broker-centric models and isolated web portals [1], [3], the system is designed to reduce manual intervention by consolidating core rental operations into a unified framework.

The architecture brings together secure user authentication, structured property listing management, AI-assisted tenant document verification, booking workflow automation, integrated payment handling, and interactive 3D visualization capabilities. While prior systems have addressed selected components such as recommendation logic or visualization enhancements [2], and others have focused on transaction management [4], these functionalities are often implemented as independent modules rather than as a cohesive architectural solution.

To overcome such fragmentation, RentIt adopts a modular and layered architectural design. Each layer is responsible for a specific operational domain, enabling clear separation of concerns and easier system maintenance. This design sup- ports horizontal scalability, secure data handling, and flexible integration of intelligent processing modules. By leveraging cloud-native services and distributed backend orchestration, the platform ensures reliability, extensibility, and consistent performance under dynamic user demand.

A.Architecture

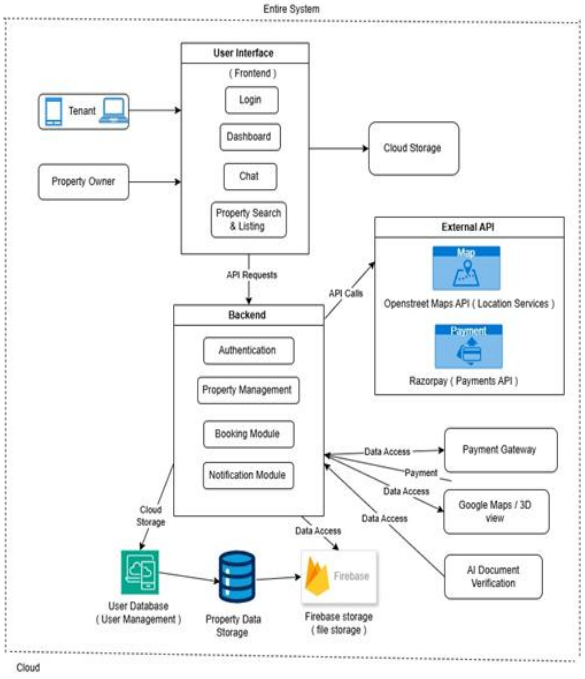

Fig. 1. Cloud-Based Architecture of RentIt System

The overall architecture of RentIt follows a cloud-native, layered design intended to support scalability, modularity, and secure service orchestration. The system is structured into four primary layers: the Frontend Layer, Backend Application Layer, Cloud Database and Storage Layer, and External API Integration Layer. This layered separation enables independent management of presentation logic, business processing, persistent storage, and third-party service interaction. Communication between layers is facilitated through stateless RESTful APIs over HTTPS, ensuring secure data exchange and token-based authentication mechanisms aligned with modern cloud computing principles [6], [7].

B.Frontend Layer

The frontend layer is developed using React.js and serves as the primary user interaction interface. It provides role- specific dashboards for tenants, property owners, and administrators, enabling structured access to system functionalities. The interface supports property discovery, dynamic filtering, booking initiation, secure document upload, real-time chat communication, and transaction tracking. A responsive design strategy ensures consistent performance across desktop and mobile environments, improving usability and accessibility.

C. Backend Application Layer

The backend layer is implemented using Node.js with the Express.js framework and operates as the central business logic controller. This layer manages authentication, authorization, property lifecycle operations, booking state transitions, digital agreement generation, and integration with AI-based verification services. Role-Based Access Control (RBAC) mechanisms are enforced to restrict resource access based on defined user roles, ensuring secure and structured interaction between tenants, owners, and administrators [8]. The backend also coordinates secure routing and request validation for all API interactions.

D. Cloud Database and Storage Layer

Persistent data management is handled through Firebase Fire store, which stores structured information including user profiles, property metadata, booking transactions, payment logs, and verification records. Firebase Storage manages un-structured assets such as property images, uploaded identification documents, digital rental agreements, and 3D property models. The adoption of managed cloud services enables automatic scalability, high availability, and real-time data synchronization across distributed clients [6].

E.AI-Based Document Verification Module

To reduce manual verification overhead and improve trust, RentIt integrates an AI-driven document validation module. Users upload identification documents through a secure interface, after which Optical Character Recognition (OCR) techniques are applied to extract textual information from scanned documents [9]. Extracted data is analyzed for structural consistency, readability, and completeness

before being classified into approval states. This automated verification workflow minimizes fraudulent registrations and ensures that only validated tenants can proceed with booking and payment operations.

F.Machine Learning Rental Recommendation Module

The rental estimation component utilizes a Random Forest regression model to generate data-driven rental price predictions. Property attributes such as area, number of bedrooms, geographic location, amenities, and infrastructure proximity are processed as input features. Ensemble-based prediction techniques, similar to matrix factorization and recommender system methodologies [10], improve predictive robustness and reduce variance. The estimated rental value is presented to both tenants and property owners during listing and search activities to support informed decision-making.

G. Booking and Payment Engine

The booking engine governs the reservation workflow. When a tenant selects a property, the system generates a booking request and validates tenant verification status before proceeding. Payment processing is integrated through secure third-party APIs, enabling encrypted transaction handling. Upon successful payment confirmation, booking records are updated and a digital rental agreement is generated and stored securely within the cloud environment. This integrated flow reduces reliance on external intermediaries and improves transparency across transactions.

H. 3D Property Visualization Module

To enhance property evaluation prior to booking, RentIt in- corporates an interactive 3D visualization framework. Property layout parameters are converted into spatial coordinates and rendered using WebGL-based techniques. The module per- forms mesh construction, texture mapping, lighting simulation, and camera navigation control to enable virtual property exploration. Similar visualization enhancements have demonstrated improved engagement in rental discovery platforms [2]. This immersive representation assists users in understanding spatial configurations without requiring physical site visits.

I. Security and Scalability Features

Security and scalability are fundamental design considerations within the RentIt architecture. Multi-layer protection mechanisms include JWT-based session management, HTTPS- encrypted communication, and strict role-based authorization policies [8]. Cloud-managed services ensure elastic resource allocation, automated backup strategies, and horizontal scaling under fluctuating workloads [7]. The modular architecture further supports future integration of emerging technologies such as blockchain-based smart contracts and IoT-enabled property monitoring systems.

IV. SYSTEM ARCHITECTURE

The RentIt platform adopts a distributed cloud-native architectural paradigm designed to unify user interaction, backend orchestration, intelligent analytics, and visualization services within a cohesive digital rental ecosystem. The architectural design emphasizes scalability, modular deployment, secure communication, and automated workflow management in alignment with established cloud computing principles [6], [7]. By minimizing reliance on manual processes and intermediary dependencies, the system ensures streamlined rental lifecycle management.

At the user interaction level, the architecture provides structured, role-based access for tenants, property owners, and administrators. The web-based interface serves as the system's entry layer, enabling property exploration, listing creation, booking initiation, document submission, and real- time communication. The separation of presentation logic from backend services enhances maintainability and ensures cross- device accessibility.

The application processing layer forms the operational core of the system. Implemented using Node.js and Express.js, this layer coordinates authentication, session control, property listing workflows, booking state transitions, agreement generation, and AI module invocation. Role-Based Access Control (RBAC) policies regulate system access to ensure that each user category interacts only with authorized re-sources [8]. Communication between the frontend and backend components is facilitated through stateless RESTful APIs operating over HTTPS, supporting secure and scalable service orchestration.

Persistent data management is handled through managed cloud services. Firebase Fire store maintains structured datasets including user profiles, property metadata, booking records, transaction logs, and verification outcomes. Firebase Storage is utilized for handling unstructured assets such as uploaded identity documents, property images, digital agreements, and 3D models. The adoption of managed cloud storage enables automatic synchronization, elastic resource allocation, and high availability [6].

To strengthen system trust and automate verification procedures, the architecture incorporates an intelligent document validation module. Uploaded identification documents undergo Optical Character Recognition (OCR) processing to extract textual information [9]. Extracted data is evaluated against predefined structural and consistency checks before being categorized into approval states. This automated verification mechanism reduces manual screening efforts and enhances system reliability.

For predictive rental estimation, the architecture integrates a machine learning component based on a Random Forest regression model. Property features such as spatial area, lo- cation characteristics, amenities, and furnishing attributes are processed as input variables. The predicted rental value is computed as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \quad (1)$$

where $T_i(x)$ denotes the output of the i^{th} decision tree within the ensemble. Ensemble-based learning approaches, widely applied in recommender systems and predictive analytics [10], improve estimation robustness and support data- driven pricing transparency.

An interactive visualization module further enhances user decision-making. Property layouts are translated into spatial coordinate structures, followed by mesh construction, texture mapping, lighting simulation, and camera navigation control. Rendering is executed using WebGL-based frameworks such as Three.js, enabling real-time virtual exploration. Similar visualization enhancements have demonstrated improved engagement and usability in rental discovery platforms [2].

The architecture also integrates external service interfaces for secure payment processing and geospatial mapping. En- crypted third-party payment APIs ensure protected financial transactions, while mapping services provide location-aware property representation.

Security considerations are embedded across all architectural layers. JWT-based session management, encrypted HTTPS communication channels, and strict role-based authorization enforcement collectively safeguard data confidentiality and integrity [8]. The modular and distributed design further supports horizontal scalability and future extensibility toward advanced technologies such as smart contract frameworks and IoT-enabled rental monitoring systems.

Overall, the RentIt system architecture establishes a tightly integrated yet modular environment that coordinates user inter- action, data persistence, intelligent analytics, and visualization services. This cohesive design enables efficient rental work- flow automation while maintaining scalability, transparency, and secure transactional governance.

V. METHODOLOGY

The RentIt platform follows a structured methodological framework aimed at digitizing and automating the complete rental lifecycle. The methodology combines cloud computing infrastructure, Artificial Intelligence (AI)-based verification, Machine Learning (ML)–driven rental prediction, and inter- active 3D visualization to create an integrated operational workflow. Each stage of the methodology is designed to reduce manual intervention while ensuring secure and scalable service delivery.

A. Data Collection

The data acquisition phase begins with property owners registering on the platform and submitting structured property information, including geographic location, spatial dimensions, number of rooms, available amenities, and visual assets. Tenants create user profiles and upload identity documents required for verification. These datasets collectively form the structured and unstructured input repository used for subsequent processing, prediction, and validation tasks.

B. System Workflow

The operational workflow follows a sequential yet

modular execution model. Initially, property data and user credentials are collected and securely stored within the cloud environment. Uploaded tenant documents are forwarded to the AI-based verification module for automated validation. Only after successful verification are tenants granted access to booking and transaction services, ensuring controlled system access aligned with secure authentication principles [8].

Once verified, tenants may explore property listings using search and filtering functionalities. The predictive module simultaneously evaluates property features and generates rental estimations to guide pricing transparency. Upon selection of a property, the booking engine validates eligibility and initiates secure payment processing. After successful transaction confirmation, a digital rental agreement is generated and archived in the cloud database. The visualization module enables tenants to inspect property layouts virtually prior to final confirmation.

C. AI-Based Document Verification Algorithm

The document verification mechanism incorporates an AI- assisted Optical Character Recognition (OCR) pipeline to extract textual content from uploaded identity documents [9]. Extracted textual data is transformed into structured features, including format consistency, completeness indicators, and confidence scores. These features are supplied to a supervised classification model that determines verification status.

Based on probabilistic thresholds, documents are categorized into Approved, Rejected, or Requires Review states. This automated approach reduces dependency on manual inspection while improving verification consistency and onboarding security prior to financial transactions.

D. Machine Learning Prediction Algorithm

The rental prediction component employs a data-driven regression methodology. Property attributes such as area, room count, amenity presence, furnishing status, and locational characteristics are preprocessed through cleaning and normalization procedures. Missing values are addressed using statistical imputation techniques, categorical attributes are numerically encoded, and extreme outliers are filtered using the Interquartile Range (IQR) method.

The refined feature matrix is provided to a Random Forest regression model. Each tree within the ensemble independently produces a rental estimate based on learned patterns from historical property data. The final predicted rental value is computed as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \quad (2)$$

where $T_i(x)$ represents the output of the i^{th} decision tree. Ensemble-based learning techniques enhance predictive robustness and reduce variance compared to single-model approaches [10]. The estimated rental value is displayed during listing and search operations to support informed decision-making.

E. Booking and Payment Workflow

The booking workflow is initiated when a tenant selects a property. The system verifies document approval status before enabling payment processing. Secure third-party APIs handle encrypted transaction execution. Upon confirmation, booking status is updated, and a digital rental agreement is generated and securely stored in the cloud repository. This structured workflow ensures transactional transparency and reduces intermediary dependency.

F. 3D Property Model Generation

To provide enhanced spatial understanding, structural property parameters such as room dimensions and layout configuration are transformed into coordinate-based representations. A three-dimensional model is generated through mesh construction, texture application, lighting simulation, and camera navigation configuration. The rendered environment enables users to explore the property virtually prior to finalizing their booking decision.

G. Cloud Integration

All operational data—including user credentials, property metadata, booking transactions, verification records, and agreements—are managed through cloud-native storage ser- vices. Managed cloud infrastructure enables real-time synchronization, elastic scaling, and high availability in accordance with established cloud computing standards [6].

H. Evaluation

The performance of the rental prediction model is assessed using standard regression evaluation metrics:

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- Coefficient of Determination (R^2)

These metrics quantify predictive accuracy, error magnitude, and model fit, ensuring objective assessment of estimation reliability.

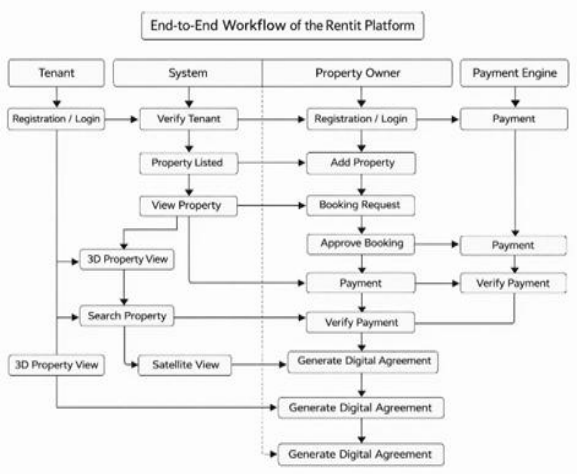

Fig. 2. End-to-End Workflow of the RentIt Platform

The end-to-end workflow of the RentIt platform is summarized in Fig. 2. The process progresses through the following stages:

1) Property owner registration and property data submission.
2) Tenant profile creation and document upload.
3) AI-driven document verification.
4) Property exploration and filtering.
5) Machine learning–based rental estimation.
6) Interactive 3D visualization of selected property.
7) Booking initiation and eligibility validation.
8) Secure payment processing.
9) Digital rental agreement generation and storage.

VI. IMPLEMENTATION

The RentIt platform was implemented as a fully functional full-stack cloud-based web application. The implementation phase focused on translating the conceptual architecture into a scalable and operational system integrating user interfaces, backend orchestration, intelligent verification modules, predictive analytics, and secure cloud services. A modular development strategy was adopted to ensure maintainability, extensibility, and efficient performance under real-time usage conditions.

A. Frontend Implementation

The frontend layer was developed using React.js with a component-based architecture to enable reusable and maintain- able UI elements. Responsive design principles were applied to ensure compatibility across desktop and mobile devices.

Fig. 3. Homepage with Property Search and Listing Interface

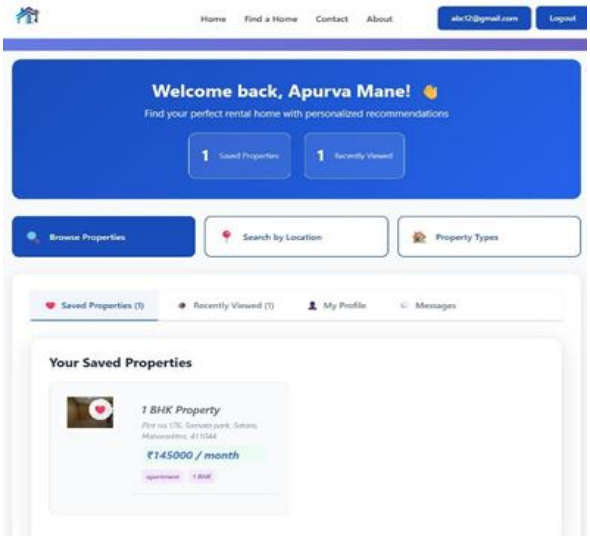

Fig. 4. Tenant Dashboard Interface

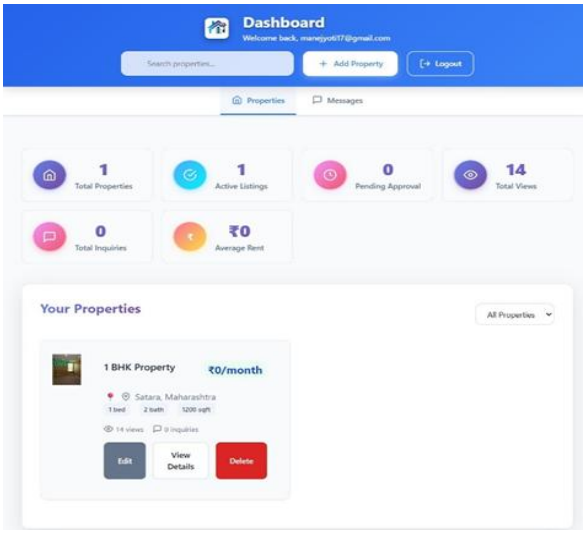

Fig. 5. Property Owner Dashboard Interface

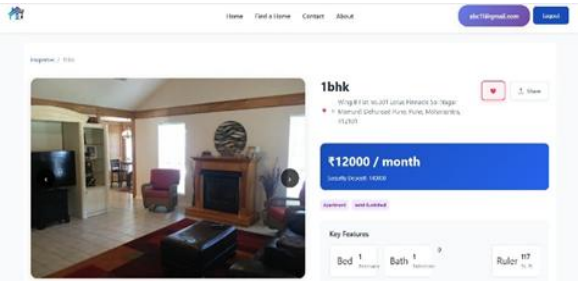

Fig. 6. Property Overview Section

Fig. 7. Amenities and Description Section

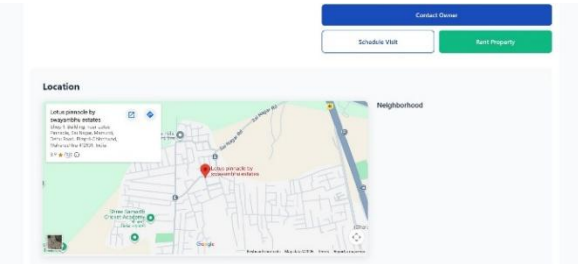

Fig. 8. Booking and Action Controls

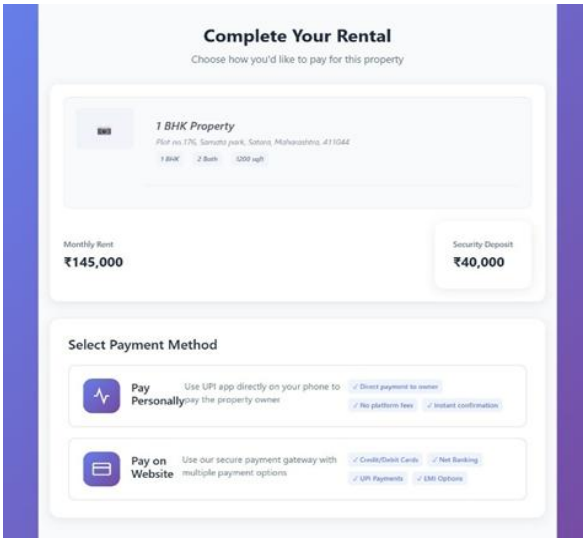

Fig. 9. Rental Payment Selection Interface

Role-based dashboards were implemented to provide customized interfaces for tenants, property owners, and administrators. Tenants were able to browse and filter properties, upload verification documents, initiate booking requests, and interact with 3D property visualizations. Property owners were provided with listing management capabilities, booking request monitoring, and agreement tracking features. Administrative controls were implemented to oversee verification statuses and platform activity.

State management was handled efficiently to ensure smooth user experience during property search, booking transitions, and payment confirmations.

B. Backend Implementation

The backend services were implemented using Node.js with the Express.js framework. The backend handled authentication, authorization, property CRUD (Create, Read, Update, Delete) operations, booking state transitions, digital agreement generation, and communication with external APIs.

JWT-based authentication mechanisms were implemented to secure user sessions. Role-based access control logic was enforced at the API level to restrict endpoint access based on user privileges. RESTful API endpoints were structured in a stateless manner to ensure scalability and compatibility with distributed cloud deployment.

The backend also coordinated interactions between the AI verification module, rental prediction engine, and payment gateway integration.

C. Database Implementation

Firebase Firestore was utilized as the primary NoSQL database for structured data storage. Collections were designed to manage user profiles, property metadata, booking trans- actions, payment records, and document verification results. Firestore's real-time synchronization capability enabled instant updates across active client sessions.

Firebase Storage was configured to handle unstructured data, including uploaded identity documents, property images, generated rental agreements, and 3D model assets. Access control rules were implemented to ensure that only authorized users could retrieve sensitive files.

D. AI Module Implementation

The document verification module was implemented using an OCR-based processing pipeline. Uploaded identification documents were securely transmitted to the verification ser- vice, where textual information

was extracted using an OCR engine.
Extracted data fields were parsed and transformed into structured validation features such as format matching, data completeness checks, and confidence scoring. A supervised classification model evaluated these features to determine authenticity status. Based on predefined confidence thresholds, documents were automatically categorized into Approved, Rejected, or Requires Review states.
The verification result was stored in the database and directly influenced booking eligibility, ensuring that only verified tenants could proceed with financial transactions.

E.Machine Learning Model Implementation
The rental prediction component was implemented using a Random Forest Regressor trained on structured property attribute datasets. Feature preprocessing included normalization, categorical encoding, and outlier filtering prior to model training.
The trained model was serialized and integrated into the backend environment, enabling real-time rental estimation during property listing and search operations. Feature im- portance metrics were also computed to identify key pricing determinants such as location, area, and amenities.

F.3D Visualization Implementation
The 3D visualization module was developed using WebGL- based rendering through the Three.js framework. Property layout parameters were programmatically transformed into geometric meshes. Textures and lighting configurations were applied to simulate realistic spatial environments.
Interactive controls were implemented to allow users to rotate, zoom, and navigate within the virtual property space. Rendering performance was optimized to ensure smooth interaction across supported devices.

G. Deployment
The complete system was deployed using a distributed cloud hosting strategy. The frontend application was deployed on Vercel, enabling optimized static asset delivery and automatic build pipelines. The backend services were deployed on Ren- der, ensuring continuous server availability and secure API access.
Firebase services managed database and storage operations, providing real-time synchronization, scalable infrastructure, and secure file management. HTTPS encryption was enforced across all communication channels.
This deployment configuration enabled horizontal scalability, minimized infrastructure management overhead, and ensured reliable real-time performance for concurrent users.

VII. RESULTS AND ANALYSIS

The RentIt platform was evaluated across multiple dimensions, including predictive model accuracy, verification efficiency, system responsiveness, and overall user workflow performance. The evaluation demonstrates the effectiveness of integrating cloud-native infrastructure, AI-assisted document validation, machine learning–based rental estimation, and interactive visualization within a unified rental management environment.

A.Machine Learning Model Performance
The Random Forest regression model was assessed using an 80:20 train–test split strategy to evaluate generalization performance. Model accuracy was measured using Mean Ab- solute Error (MAE), Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2).

Table I Machine Learning Model Performance Metrics

Metric	Value
MAE	1850
RMSE	2650
R^2 Score	0.89

An R^2 value of 0.89 indicates that the model explains approximately 89% of the variance in rental pricing, reflecting strong predictive capability. The MAE and RMSE values demonstrate acceptable deviation levels between predicted and actual rental prices, confirming that the model can provide reliable price estimations suitable for practical deployment scenarios.
Feature importance analysis further indicated that location, property area, and available amenities were the most influential predictors in rental value estimation.

B.AI-Based Verification Performance
The document verification module was evaluated based on processing consistency and reduction in

manual verification effort. The OCR pipeline demonstrated high reliability in extracting structured textual data from uploaded identity documents. The classification component successfully categorized submissions into Approved, Rejected, and Requires Review states based on confidence thresholds.

Automated verification significantly reduced manual screening time and ensured that only validated users were permitted to proceed with booking and financial transactions, thereby improving system trust and onboarding efficiency.

System performance was analyzed in terms of responsive- ness, transaction integrity, and synchronization reliability. The distributed cloud deployment architecture enabled:

- Real-time synchronization of property listings and booking updates.
- Secure payment processing via encrypted third-party APIs.
- Low-latency property search and filtering operations.
- Smooth rendering of interactive 3D property models.

The integration of managed Firebase services ensured high availability, consistent data access, and scalable storage management across concurrent user sessions.

C. User Experience Evaluation

The inclusion of 3D visualization significantly enhanced spatial comprehension and user engagement. Tenants were able to assess room layout and property structure virtually, reducing dependency on physical visits and improving decision confidence.

The structured workflow—from user registration to agreement generation—simplified the rental process. Automation of document verification and rental estimation reduced operational delays and minimized manual coordination between tenants and property owners.

D. Analysis

The experimental evaluation indicates that the integration of AI-driven verification, ensemble-based rental prediction, and cloud-native service orchestration contributes to improved system efficiency when compared to traditional broker-driven rental models. Automated workflows reduce administrative overhead, enhance pricing transparency, and strengthen trans- actional reliability. Although the current implementation demonstrates strong performance, future improvements may include expanding the training dataset for enhanced predictive robustness, incorporating deep learning–based document validation techniques, and integrating dynamic pricing strategies influenced by temporal and seasonal demand patterns.

VIII. FUTURE SCOPE

Future enhancements of the RentIt platform may include expanding the training dataset to improve the accuracy of the rental price prediction model. The system can incorporate advanced deep learning techniques for more reliable document verification. Integration of dynamic pricing mechanisms based on market trends, seasonal demand, and location analytics can further improve rent estimation. Additionally, the platform can be extended to support government-approved digital lease agreements to ensure legal validity and compliance. Further development may also include mobile application support and smart contract–based rental agreements for enhanced security and automation.

IX. KEY FINDINGS

The experimental evaluation and system implementation revealed several important findings regarding the effectiveness of the proposed RentIt platform.

The integration of the Machine Learning-based rent prediction module enabled data-driven pricing and reduced inconsistencies in rental rates.

AI-based document verification significantly minimized the risk of fraudulent tenant registrations by automating identity validation.

Automation of document checks and booking workflows decreased dependency on manual verification processes.

The 3D property visualization feature allowed tenants to evaluate spatial layouts effectively without requiring physical visits.

Cloud deployment ensured real-time synchronization and scalability for handling multiple users and listings.

Role-based dashboards improved usability for tenants and property owners.

These findings indicate that the proposed system

enhances efficiency, trust, and accessibility within the property rental ecosystem.

X. CONCLUSION

This work introduced RentIt, a cloud-native rental management platform developed to address inefficiencies inherent in traditional property rental systems. By consolidating authentication, document verification, rental price estimation, transaction management, and immersive visualization into a unified framework, the platform redefines how rental workflows are executed in a digital environment.

The integration of AI-assisted document validation enhances onboarding security by minimizing manual screening and reducing fraudulent registrations. The Random Forest–based rental estimation model enables transparent, data-driven pricing, contributing to fairer and more informed decision-making for both tenants and property owners. Secure payment processing and automated agreement generation further streamline the transaction lifecycle while maintaining system integrity.

Experimental evaluation confirmed strong predictive performance and stable system responsiveness under realistic usage scenarios. The modular cloud deployment ensures scalability, real-time synchronization, and consistent availability across distributed users. The addition of interactive 3D visualization improves spatial understanding and user engagement, reducing reliance on physical property visits.

Collectively, these contributions demonstrate that integrating cloud infrastructure, intelligent verification mechanisms, predictive analytics, and immersive visualization can significantly modernize rental management processes. RentIt pro- vides a scalable, secure, and extensible foundation for future advancements in digital real estate ecosystems.

REFERENCES

[1] O. Iroju, A. A. Adeniran, and O. O. Ikono, “A secured mobile cloud-based house rental management system,” 2021. [Online]. Available: https://www.researchgate.net/publication/340926278

[2] S. M. Satapathy, R. Jhaveri, U. Khanna, and A. K. Dwivedi, “Smart rent portal using recommendation system visualized by augmented reality,” Procedia Computer Science, vol. 171, pp. 197–206, 2020.

[3] R. Rastogi, R. Jain, P. Mishra, and P. Kumar, “Rental house management system: An empirical approach with simulation,” Acta Scientific Computer Sciences, vol. 5, no. 4, pp. 88–102, 2023.

[4] J. M. Magno, R. S. Pelayo, and K. T. Soberano, “Apartment rental management system for real-time transaction and task organization,” International Journal of Multidisciplinary Research and Analysis, vol. 7, no. 7, pp. 3287–3294, Jul. 2024.

[5] S. Raddi, S. Matkari, S. Suryawanshi, R. Kumbhar, and S. Ambavade, “Smart rental ecosystem: An integrated and user-friendly platform for student residential discovery,” International Journal of Science, Engineering and Technology, vol. 13, no. 3, 2025.

[6] P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication 800-145, 2011.

[7] M. Armbrust et al., “A view of cloud computing,” Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.

[8] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.

[9] R. Smith, “An overview of the Tesseract OCR engine,” in Proc. Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.

[10] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.