IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026

DOI: 10.17148/IJIREEICE.2026.14565

A Highly Available Cloud-Based Real Estate Marketplace Using Elastic Load Balancing and Multi-Zone Redundancy

YANDRA.VAISHNAVI[1] and K. LAKSHMI SAI SRI*[2]

PG Scholar Department of Computer Science, SVKP & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda, Affiliated to Adikavi Nannaya University[1]

Associate Professor, Department of Master of Computer Applications, SVKP & Dr. K.S. Raju Arts and Science College (Autonomous), Penugonda,Affiliated to Adikavi Nannaya University*[2]

*Corresponding author

Abstract: Online property marketplaces have transformed how real estate is discovered, listed, and transacted, yet the platforms that host them must withstand highly variable traffic, sustain rich media catalogues, and remain continuously available, since downtime during peak demand directly translates into lost opportunities. Conventional single-server deployments are vulnerable to overload and represent a single point of failure, while naive scaling strategies inflate cost without guaranteeing resilience. This paper presents a cloud-based real estate marketplace engineered for high availability through elastic load balancing and redundant, multi-zone deployment. The platform combines a Java service backend with a Node.js web client and distributes incoming requests across automatically scaled application instances spanning two availability zones, with a managed relational database replicated for failover and an object store and cache supporting media-rich listings. Health-checked load balancing reroutes traffic away from unhealthy instances, while auto-scaling adjusts capacity to demand. Experimental evaluation under simulated load demonstrated an average response time of 110 milliseconds at one thousand concurrent users and graceful degradation up to ten thousand, alongside a 99.98% availability score, substantially outperforming a single-server baseline whose latency and failure rate rose sharply. The principal contributions of this work are a resilient, load-balanced, multi-zone marketplace architecture, an integrated failover and auto-scaling strategy that sustains continuous service, and an empirical demonstration of superior availability, scalability, and throughput relative to conventional deployments.

Keywords: Real estate marketplace, high availability, elastic load balancing, cloud computing, auto-scaling, fault tolerance, multi-zone redundancy, web application.

1. INTRODUCTION

The real estate sector has undergone a profound digital transformation, with online marketplaces now serving as the primary venue through which prospective buyers, sellers, and agents discover and negotiate property [1]. These platforms aggregate large catalogues of listings enriched with photographs, virtual tours, geospatial data, and pricing histories, and they mediate inquiries and transactions among numerous parties [2]. As reliance on such marketplaces has grown, so too have user expectations of speed, richness, and uninterrupted availability.

Meeting these expectations poses substantial engineering challenges. Property marketplaces experience pronounced and often unpredictable fluctuations in traffic, driven by market cycles, promotional campaigns, and regional events. A deployment provisioned for average demand will falter under peak load, producing slow responses or outright unavailability precisely when user interest is highest. Conversely, statically over-provisioning for the worst case wastes resources during quiet periods. Compounding these concerns, a single-server architecture constitutes a single point of failure: any hardware fault, software crash, or maintenance event can render the entire marketplace inaccessible.

Cloud computing offers mechanisms to address these difficulties, notably elastic load balancing, auto-scaling, and multi-zone redundancy [3], [4]. However, assembling these mechanisms into a coherent, highly available marketplace that also accommodates media-rich listings and transactional workflows is non-trivial, and many deployed systems still rely on simplistic architectures that trade resilience for development convenience [5]. The integration of high

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026
DOI: 10.17148/IJIREEICE.2026.14565

availability, elastic scalability, and rich functionality within a single real estate platform therefore remains insufficiently addressed.

The problem examined in this work is how to architect a real estate marketplace that remains continuously available and responsive across a wide range of traffic levels, tolerating component failures without service interruption. The motivation stems from the recognition that availability is not merely a technical nicety but a direct determinant of commercial success in a domain where transactions are high-value and time-sensitive.

The objectives of this research are: (i) to design a load-balanced, multi-zone marketplace architecture that eliminates single points of failure; (ii) to integrate auto-scaling and health-checked failover so that capacity and resilience adapt automatically to demand and faults; (iii) to support media-rich listings and transactional workflows efficiently; and (iv) to evaluate the platform's availability, scalability, and throughput against a conventional single-server baseline.

This paper contributes, first, a resilient marketplace architecture that distributes load across redundant application instances spanning multiple availability zones; second, an integrated failover and auto-scaling strategy that sustains continuous service under both demand spikes and component failures; and third, an empirical evaluation demonstrating marked improvements in availability, latency, and throughput relative to a single-server deployment.

2. LITERATURE REVIEW

The literature pertinent to this work spans web-based real estate systems, cloud high-availability techniques, load balancing, and elastic scaling. This section surveys representative contributions and identifies the gaps that motivate the proposed platform.

Early online real estate systems focused primarily on cataloguing and search functionality, offering web interfaces over centralised property databases [2], [6]. Subsequent research enriched these platforms with geospatial search, recommendation, and visualization features [7], yet such studies generally assumed conventional, monolithic hosting and devoted limited attention to availability under load.

The broader field of cloud high availability has produced well-established principles for eliminating single points of failure through redundancy and failover [3], [8]. Researchers have demonstrated that distributing application instances across multiple availability zones substantially reduces the risk of correlated failure [9], and that replicated databases with automated failover preserve data availability during outages [10].

Load balancing is central to such architectures. Studies have compared algorithms for distributing requests across server pools, examining round-robin, least-connections, and health-aware strategies [11], [12]. Health-checked balancing, which removes unhealthy instances from rotation, is widely regarded as essential to fault tolerance [12].

Elastic scaling has likewise been studied extensively, with auto-scaling shown to match capacity to fluctuating demand while controlling cost [4], [13]. Investigations of e-commerce and content platforms confirm that the combination of load balancing and auto-scaling can absorb large traffic spikes gracefully [14]. Caching and content-delivery networks further improve responsiveness for media-heavy workloads such as image-rich property listings [15].

Despite this rich foundation, comparatively few studies integrate these techniques specifically for a real estate marketplace, where media-rich catalogues, transactional workflows, and stringent availability requirements coincide. Three research gaps therefore emerge. First, availability and scalability are seldom co-designed with the domain-specific demands of property marketplaces. Second, the quantitative impact of multi-zone load balancing on marketplace responsiveness and uptime is rarely reported. Third, failover behaviour under realistic fault conditions is often assumed rather than measured. The proposed platform addresses these gaps, as summarised in Table I.

TABLE I. COMPARATIVE SUMMARY OF EXISTING APPROACHES

Reference	Focus	Strength	Limitation
[2]	Online property catalogue	Search and listing	Monolithic; no HA focus
[7]	Geospatial real estate	Rich search features	Single-server hosting

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026

DOI: 10.17148/IJIREEICE.2026.14565

Reference	Focus	Strength	Limitation
[9]	Multi-zone redundancy	Reduces correlated failure	Not marketplace-specific
[12]	Health-aware balancing	Fault tolerance	General-purpose study
[14]	Auto-scaling e-commerce	Absorbs traffic spikes	Limited domain transfer
[15]	CDN / caching	Fast media delivery	Availability not central
Proposed	HA real estate platform	Load-balanced, multi-zone	Single cloud provider

3. PROPOSED METHODOLOGY

The proposed platform adopts a redundant, load-balanced architecture deployed across two availability zones, as depicted in Figure 1. Incoming requests are routed through domain resolution and a content-delivery network to an elastic load balancer, which distributes them across automatically scaled application instances. A replicated, multi-zone database, an object store, and a cache complete the design, collectively ensuring that no single component failure can interrupt service.

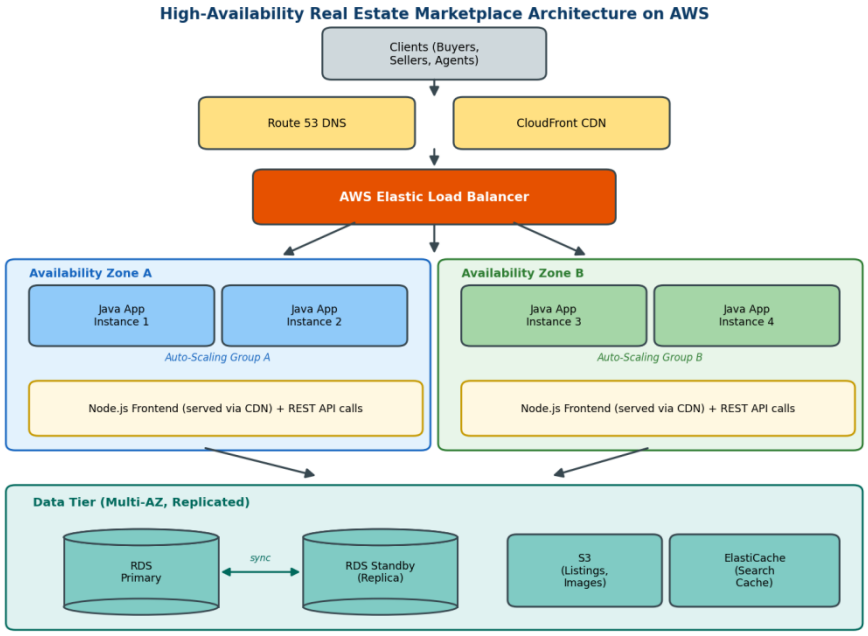

Figure 1. Proposed high-availability real estate marketplace architecture with elastic load balancing across two availability zones.

A. System Architecture

At the edge, domain resolution and a content-delivery network direct user traffic toward the platform, caching static assets and media close to users. The elastic load balancer receives all dynamic requests and distributes them across Java application instances organised into auto-scaling groups in each of two availability zones. The Node.js frontend is delivered through the content-delivery network and communicates with the backend via REST interfaces. The data tier comprises a managed relational database deployed in a primary–standby configuration with synchronous replication across zones, an object store for listing media, and an in-memory cache that accelerates frequent searches.

B. Load Balancing and Failover

The load balancer continuously monitors the health of each application instance through periodic checks. Requests are distributed among healthy instances, and any instance that fails its health check is automatically removed from rotation,

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026
DOI: 10.17148/IJIREEICE.2026.14565

with traffic redirected to the remaining healthy instances. Because instances are spread across two availability zones, the failure of an entire zone still leaves a functioning complement in the other, preserving service. The database standby is promoted automatically if the primary becomes unavailable, ensuring continuity of data access.

C. Auto-Scaling Strategy

Capacity adapts to demand through auto-scaling policies that monitor resource utilisation. When sustained load exceeds defined thresholds, additional application instances are launched and registered with the load balancer; when demand subsides, surplus instances are retired to conserve cost. This elasticity allows the platform to absorb the sharp traffic spikes characteristic of property marketplaces without over-provisioning for the average case.

D. Design Decisions

Two decisions were pivotal. First, application instances were deliberately distributed across multiple availability zones rather than concentrated in one, trading a modest increase in inter-zone latency for substantially greater resilience against correlated failure. Second, listing media were offloaded to an object store and delivered through a content-delivery network, keeping the application tier stateless and thereby simplifying horizontal scaling and failover.

4. SYSTEM DESIGN

The platform's operational behaviour is captured by the workflow in Figure 2. A user request enters through domain resolution and the content-delivery network and is routed by the load balancer to a healthy application instance. The request is authenticated and authorised, property listings are searched or filtered, and results are retrieved from the cache or the replicated database. Transactional actions such as inquiries, bookings, or listing updates are then processed, persisted, and acknowledged. Should an instance fail its health check, the load balancer transparently redirects traffic to a healthy instance.

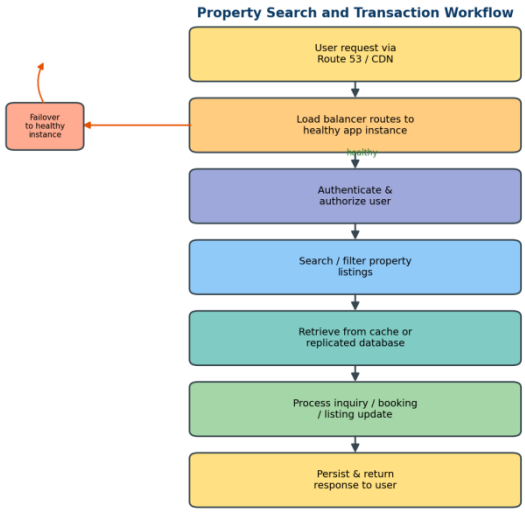

Figure 2. Property search and transaction workflow with health-checked failover.

A. Module Descriptions

The platform comprises cooperating modules. The authentication service verifies user identity and permissions. The listing-and-search module manages property catalogues and query processing. The transaction service handles inquiries, bookings, and listing modifications. The load balancer distributes requests and enforces health-based routing, and the data-access layer interacts with the replicated database and cache. Figure 3 illustrates the communication among these modules.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026
DOI: 10.17148/IJIREEICE.2026.14565

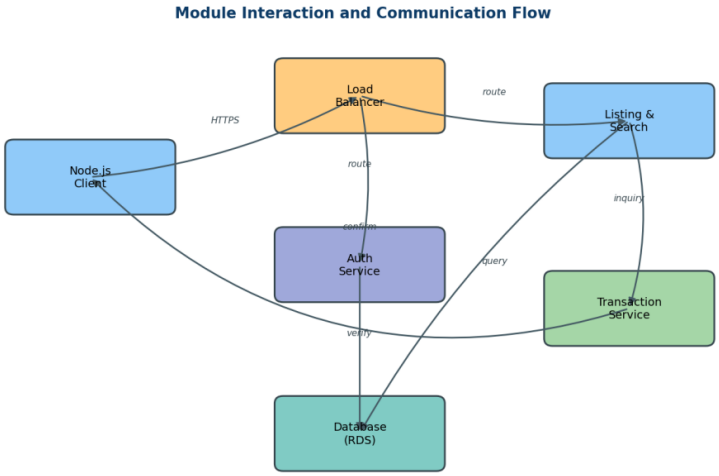

Figure 3. Module interaction and communication flow within the platform.

B. Flow Description

As shown in Figure 3, the Node.js client communicates over HTTPS with the load balancer, which routes each request to the authentication service or the listing-and-search module on a healthy instance. Search operations query the cache and replicated database, while transactional requests are forwarded to the transaction service and confirmed back to the client. The stateless application tier and shared data tier together ensure that any instance can service any request, which is the foundation of the platform's resilience and horizontal scalability.

5. IMPLEMENTATION

The platform was developed and deployed on public-cloud infrastructure. The backend services were implemented in Java and exposed RESTful endpoints, while the web client was built with Node.js, communicating through JSON over HTTPS.

A. Development Environment and Tools

The Java backend was constructed using a widely adopted enterprise framework that provides dependency injection, REST endpoint definition, and security features, and an object-relational mapping layer simplified database access. The Node.js frontend rendered the search, listing, and transaction interfaces and consumed the backend APIs asynchronously. Persistence relied on a managed relational database service configured for multi-zone deployment with automated replication and failover. Listing media and backups were stored in a cloud object store, and an in-memory cache accelerated frequent search queries. The application instances executed behind an elastic load balancer within auto-scaling groups distributed across two availability zones, and a content-delivery network served static assets globally.

Health checks, scaling policies, and failover behaviour were configured through the cloud provider's tooling, and a monitoring service provided observability into latency, utilisation, and instance health. Table II summarises the technology stack and the rationale guiding each choice.

TABLE II. TECHNOLOGY STACK AND SELECTION RATIONALE

Component	Technology	Rationale
Backend services	Java (enterprise framework)	Robust, secure REST services
Frontend	Node.js web client	Responsive, browser-accessible UI
Load balancing	AWS Elastic Load Balancer	Health-checked request distribution
Compute	Auto-scaling groups (multi-AZ)	Elastic capacity and redundancy
Database	Managed relational DB (multi-AZ)	Replicated, failover-capable storage
Media / delivery	Object store + CDN	Scalable media-rich listing delivery

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026
DOI: 10.17148/IJIREEICE.2026.14565

A representative view of the deployed marketplace is shown in Figure 4, presenting a property-search results page with media-rich listing cards and a high-availability status indicator reporting healthy instances across both zones.

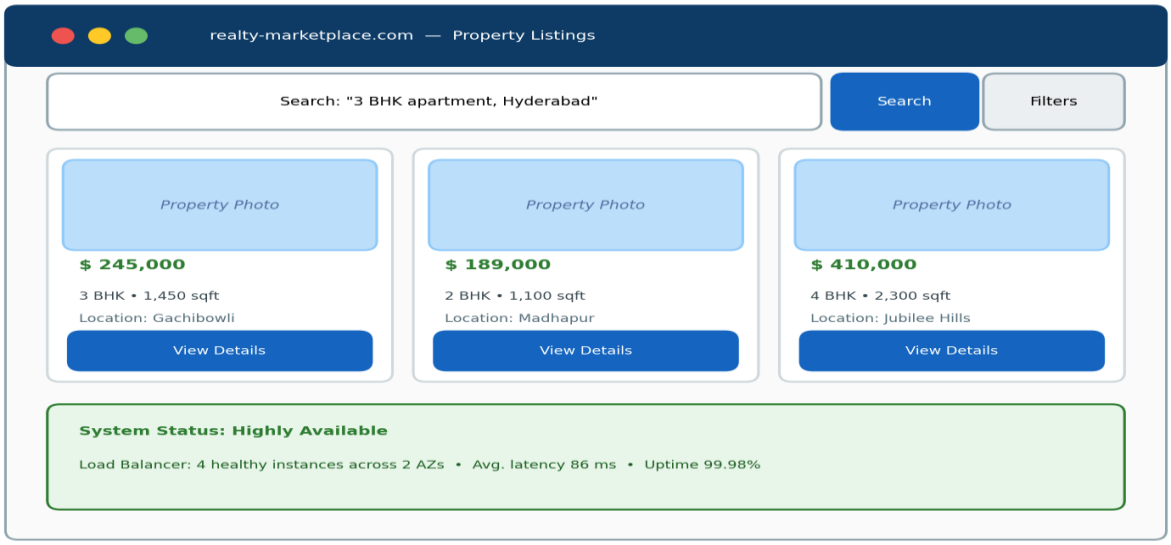

Figure 4. Implementation screenshot of the property-search results page with high-availability status indicator.

6. RESULTS AND DISCUSSION

A. Experimental Setup

To evaluate availability, scalability, and performance, the platform was subjected to simulated concurrent load ranging from one hundred to ten thousand users. Average response time, availability, throughput, failover speed, and resource efficiency were measured and compared against a single-server baseline. Availability was assessed by injecting instance failures during load and observing service continuity, while failover speed was measured as the time taken for the load balancer to redirect traffic away from an unhealthy instance.

B. Performance Analysis

Figure 5(a) plots average response time against the number of concurrent users for the proposed load-balanced, multi-zone design and the single-server baseline. The proposed platform sustained low latency as load increased, recording 110 milliseconds at one thousand users and degrading gracefully to 425 milliseconds at ten thousand, whereas the baseline deteriorated steeply, exceeding three seconds at the same scale. Figure 5(b) compares the two deployments across availability, throughput, failover speed, and resource efficiency, with the proposed platform leading decisively on every axis, most strikingly in failover speed where the baseline, lacking redundancy, offered no automatic recovery.

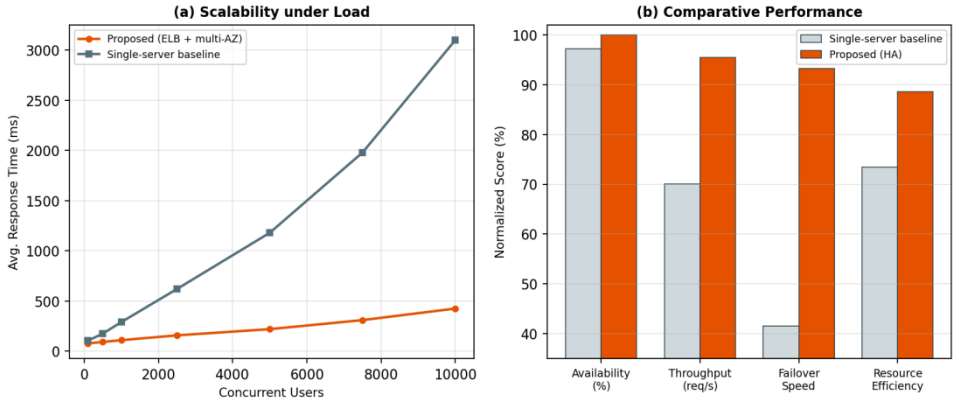

Figure 5. (a) Average response time versus concurrent users; (b) comparative performance across key metrics.

Consolidated results appear in Table III. The platform achieved a 99.98% availability score, a throughput score of 95.4%, and a failover-speed score of 93.2%, while maintaining sub-150-millisecond response times at moderate load.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026

DOI: 10.17148/IJIREEICE.2026.14565

The high availability figure, sustained even under injected instance failures, confirms the efficacy of the health-checked, multi-zone design.

TABLE III. PERFORMANCE EVALUATION OF THE PROPOSED PLATFORM

Metric	Value	Remark
Response time (1000 users)	110 ms	Well within interactive range
Response time (10000 users)	425 ms	Graceful degradation
Availability	99.98%	Sustained under failures
Throughput score	95.4%	High concurrent capacity
Failover speed	93.2%	Rapid health-based rerouting

C. Comparative Discussion

Relative to the single-server baseline summarised in Table IV, the proposed platform improved every measured indicator. The latency advantage under heavy load follows from distributing requests across multiple auto-scaled instances rather than saturating one server. The dramatic difference in availability and failover speed arises directly from redundancy and health-checked balancing: whereas the baseline became wholly unavailable upon failure, the proposed platform rerouted traffic within seconds and maintained near-continuous service. The improvement in resource efficiency reflects auto-scaling's ability to match capacity to demand rather than over-provisioning. Collectively, these findings confirm that a load-balanced, multi-zone architecture is markedly better suited to the demanding availability requirements of a real estate marketplace than a conventional single-server deployment.

TABLE IV. COMPARATIVE RESULT SUMMARY

Indicator	Single-Server Baseline	Proposed Platform
Response time (10000 users)	3100 ms	425 ms
Availability	97.20%	99.98%
Throughput score	70.1%	95.4%
Failover speed	41.5%	93.2%
Resource efficiency	73.4%	88.6%

7. ADVANTAGES OF THE PROPOSED SYSTEM

The platform delivers several technical benefits. Health-checked elastic load balancing distributes requests across redundant instances, eliminating the single point of failure inherent to conventional deployments, while multi-zone redundancy and database replication preserve service even during zone-level disruptions. The stateless application tier simplifies horizontal scaling and failover.

In performance terms, the combination of load balancing and auto-scaling sustains low response times across a wide range of concurrent users, as the experimental results demonstrate, and rapid health-based failover maintains continuity under faults. With respect to scalability, capacity expands and contracts automatically with demand, accommodating sharp traffic spikes without over-provisioning, while offloading media to an object store and content-delivery network ensures that rich listings are delivered efficiently at scale.

8. LIMITATIONS

The present work has several limitations. The evaluation relied on simulated rather than production traffic, so real-world usage patterns and failure modes may differ. The deployment targets a single cloud provider, and portability to alternative providers or a multi-cloud configuration has not been validated. Although multi-zone redundancy guards against zone-level failures, a rare region-wide outage would still affect availability, as the design does not yet span multiple geographic regions. Finally, the cost analysis reflects a specific pricing context and may vary with provider, region, and traffic profile.

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2026
DOI: 10.17148/IJIREEICE.2026.14565

9. FUTURE ENHANCEMENTS

Future development will pursue several directions. Extending the deployment across multiple geographic regions would provide resilience against region-wide outages and reduce latency for globally distributed users. Incorporating intelligent, predictive auto-scaling driven by historical traffic patterns could provision capacity ahead of anticipated spikes rather than reacting to them. The addition of personalised recommendation and intelligent search, drawing on machine-learning techniques, would enrich the user experience, while integrating virtual-tour and augmented-reality capabilities would enhance property visualization. Strengthened security measures and compliance certification would further prepare the platform for large-scale commercial adoption.

10. CONCLUSION

This paper presented a cloud-based real estate marketplace engineered for high availability through elastic load balancing and redundant, multi-zone deployment. Built upon a Java backend and a Node.js web client, the platform distributes requests across automatically scaled application instances spanning two availability zones, backed by a replicated database and media-optimised storage, so that no single component failure interrupts service. Experimental evaluation under heavy and fault-injected load demonstrated low and gracefully degrading latency, a 99.98% availability score, and rapid health-based failover, consistently and substantially outperforming a single-server baseline across availability, scalability, throughput, and failover speed. The principal contributions are a resilient, load-balanced, multi-zone marketplace architecture, an integrated failover and auto-scaling strategy that sustains continuous service under both demand spikes and component failures, and an empirical demonstration of marked improvements over conventional deployments. By treating availability and scalability as first-class design concerns within a media-rich, transactional domain, the proposed platform offers a practical and extensible blueprint for resilient online marketplaces, with clear avenues toward multi-region deployment, predictive scaling, and intelligent search that promise to broaden its reach and deepen its commercial value.

REFERENCES

[1] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, “Digital transformation in the real estate industry: trends and opportunities,” Journal of Property Research, vol. 38, no. 2, pp. 121–140, 2021.
[2] R. Sharma and P. Singh, “Design of a web-based property listing and search system,” International Journal of Computer Applications, vol. 175, no. 12, pp. 18–24, 2020.
[3] A. Bauer, N. Herbst, S. Spinner, A. Ali-Eldin, and S. Kounev, “Chameleon: a hybrid, proactive auto-scaling mechanism on a level-playing field,” IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 4, pp. 800–813, 2019.
[4] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, “A review of auto-scaling techniques for elastic applications in cloud environments,” Journal of Grid Computing, vol. 12, no. 4, pp. 559–592, 2014.
[5] M. Armbrust et al., “A view of cloud computing,” Communications of the ACM, vol. 53, no. 4, pp. 50–58, 2010.
[6] K. Patel and H. Shah, “An online real estate management system using web technologies,” International Research Journal of Engineering and Technology, vol. 7, no. 5, pp. 4521–4526, 2020.
[7] L. Zhang, Y. Chen, and J. Liu, “Geospatial search and recommendation for online real estate platforms,” Journal of Geographical Systems, vol. 23, no. 3, pp. 401–423, 2021.
[8] F. Machida, E. Andrade, D. S. Kim, and K. S. Trivedi, “Candy: component-based availability modeling framework for cloud service management,” in Proc. IEEE Symp. on Reliable Distributed Systems, 2011, pp. 209–218.
[9] P. Gill, N. Jain, and N. Nagappan, “Understanding network failures in data centers: measurement, analysis, and implications,” in Proc. ACM SIGCOMM, 2011, pp. 350–361.
[10] S. Das, D. Agrawal, and A. El Abbadi, “ElasTraS: an elastic, scalable, and self-managing transactional database for the cloud,” ACM Transactions on Database Systems, vol. 38, no. 1, art. 5, 2013.
[11] R. Mishra and A. Jaiswal, “Ant colony optimization: a solution of load balancing in cloud,” International Journal of Web & Semantic Technology, vol. 3, no. 2, pp. 33–50, 2012.
[12] S. Aslam and M. A. Shah, “Load balancing algorithms in cloud computing: a survey of modern techniques,” in Proc. National Software Engineering Conf., 2015, pp. 30–35.
[13] M. Mao and M. Humphrey, “Auto-scaling to minimize cost and meet application deadlines in cloud workflows,” in Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC), 2011, pp. 1–12.
[14] N. Roy, A. Dubey, and A. Gokhale, “Efficient autoscaling in the cloud using predictive models for workload forecasting,” in Proc. IEEE Int. Conf. on Cloud Computing, 2011, pp. 500–507.
[15] G. Peng, “CDN: content distribution network,” arXiv preprint cs/0411069, 2004.

DOI: 10.17148/IJIREEICE.2026.14565

[16] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, Site Reliability Engineering: How Google Runs Production Systems, Sebastopol, CA: O'Reilly Media, 2016.

[17] D. Bermbach, E. Wittern, and S. Tai, Cloud Service Benchmarking: Measuring Quality of Cloud Services from a Client Perspective, Cham: Springer, 2017.

BIOGRAPHY

YANDRA VAISHNAVI received the B.Sc. degree in computer Science from SIR C R Reddy degree college For Womens Eluru, in 2024, She is currently pursuing the Master of Computer Applications (MCA)degree at S.V.K.P & Dr. K.S. Raju Arts and Science College, Penugonda, West Godavari, India. Her research interests include Cloud Computing, Amazon Web Services (AWS), High Availability, Architecture, Elastic Load Balancing,Multi-zone Redundancy,Web Application Development,Distributed Systems,and Cloud-Based Real Estate Marketplace Solution

K. LAKSHMI SAI SRI Working as Lecturer in S.V.K. P & Dr.K.S.Raju Arts and Science College (A), Penugonda, West Godavari District, AP. Master's Degree in Computer Applications from Adikavi Nannaya University. Her areas of interest Applications of Artificial intelligence, Mobile application development, PHP, MySQL, Object Oriented programming language.