iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** V **Month of publication:** May 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.82889

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue V May 2026- Available at www.ijraset.com*

# BidSphere: A Secure, Real-Time Web-Based Smart Online Auction Platform with Role-Based Access Control and High-Concurrency Bidding Architecture

Suguna S[1], Ravin Kumar K[2], Selva Kumar N[3], Surya L[4]
*Department of Computer Science and Business Systems, SSM Institute of Engineering and Technology (Autonomous), Dindigul, Tamil Nadu, India, Affiliated to Anna University, Chennai — 600 025*

***Abstract:*** ***BidSphere is an enterprise-grade, secure web-based online auction platform designed and developed to modernize traditional offline auction frameworks using real-time distributed bidding technology and a highly available software stack. The system supports fine-grained role-based access control explicitly classified into three discrete tiers: Administrators, Auctioneers, and Bidders. To preserve transactional integrity across high-concurrency event states, the backend layer implements high-speed state-machine tracking coupled with microsecond-accurate data reconciliation. A decoupled architecture cleanly separates the dynamic UI layer from data persistence routines, isolating long-duration archiving tasks from high-velocity operational loops and reducing cross-component query contention to nominal background levels. Comprehensive testing criteria — including functional unit tests, cross-component integration pipelines, and thorough User Acceptance Testing — were executed to ensure full coverage against operational failures. Experimental logs confirm structural stability, low latency responses, and immediate interface synchronization across concurrent multi-user environments, making BidSphere ideal for scalable localized deployment configurations.***

***Keywords:*** ***online auction, real-time bid synchronization, role-based access control, cryptographic authentication, state machine logic, WebSocket, Node.js, decoupled architecture, concurrent bidding, user acceptance testing.***

## I. INTRODUCTION

The emergence of digital economies and high-speed web services has fundamentally transformed traditional transactional workflows across global business landscapes. Traditional physical auctions, which have historically operated as a primary paradigm for high-value asset liquidation and fair market price discovery, are increasingly constrained by rigid environmental limitations — geographic reach, time-bound schedules, and manual bid management. BidSphere — Smart Online Auction Platform represents a modern software response explicitly engineered to establish an open, highly secure, scalable, and low-latency digital marketplace ecosystem that overcomes these structural constraints.

The platform employs a decoupled stack architecture in which the primary event broadcasting thread routes structural data states through highly optimized non-blocking transport buffers. By systematically distributing message broker queues away from operational table writes, the backend environment safely isolates long-duration archiving storage tasks from high-velocity operational loops. As a result, cross-component query contention is reduced to nominal background levels, maintaining a flawless, deterministic client execution speed across real-world enterprise deployments.

### *A. Traditional Auction Systems*

Early electronic auction architectures developed in the late 1990s primarily relied on standard monolithic multi-tier systems with synchronous database write loops. Using HTTP/1.0 configurations required a complete client-side page reload to display updated item valuations, making real-time bidding technically infeasible. These legacy systems were further constrained by the absence of persistent bi-directional communication channels, resulting in poor concurrency handling and significant user experience degradation under simultaneous bidding scenarios.

©IJRASET: All Rights are Reserved 

### B. Challenges in Real-Time Auction Platforms

Modern online auction platforms face unique challenges across security, concurrency, and performance dimensions. Simultaneous bid submissions from multiple clients must be resolved with microsecond-accurate timestamping to guarantee transactional integrity and prevent race conditions. Authentication mechanisms must enforce strict role separation to prevent unauthorized listing creation, bid manipulation, or administrative overreach. Additionally, user interfaces must reflect bid state changes in real time without full page reloads, necessitating persistent bi-directional transport protocols.

### C. Modern Real-Time Web Solutions

With the introduction of modern protocol backbones such as HTML5 WebSockets and asynchronous event-driven I/O loop engines, developers resolved the performance limits of legacy polling architectures by establishing persistent bi-directional transport pipes between clients and servers. These technologies enable server-push notifications that instantly propagate bid updates across all connected clients, delivering the sub-second synchronization that high-stakes auction environments demand. BidSphere leverages these foundational advances alongside cryptographic authentication and state-machine-driven bid validation to deliver a production-grade auction platform.

### D. Research Objectives

The primary objective of this research is to design, implement, and validate a comprehensive, real-time online auction platform that overcomes the limitations of existing systems. Specifically, the BidSphere platform is engineered to:

1) Establish a cryptographic multi-tier authentication matrix with strict role-based access control for Administrators, Auctioneers, and Bidders.
2) Develop ultra-low-latency bid broadcasting pipelines using WebSocket-based persistent bi-directional transport.
3) Implement rigid role-based dashboards that present contextually appropriate views and controls to each user tier.
4) Ensure transactional integrity under high-concurrency bidding scenarios using state-machine validation and microsecond-accurate data reconciliation.
5) Validate system reliability through comprehensive unit, integration, and User Acceptance Testing (UAT) pipelines.

## II. LITERATURE REVIEW

A systematic review of existing literature was conducted to situate the BidSphere platform within current research trends in real-time distributed auction systems, web application security, and concurrent data management.

### A. First-Generation Electronic Auction Paradigms

Early electronic auction architectures developed in the late 1990s relied on monolithic multi-tier systems with synchronous database write loops.

The requirement for full page reloads under HTTP/1.0 prevented real-time bid display. Research in this period focused primarily on correctness of bid resolution rather than latency optimization, establishing foundational auction theory models — English, Dutch, and sealed-bid — that continue to inform contemporary platform design.

### B. Critique of Modern Real-Time Web Solutions

The introduction of HTML5 WebSockets and asynchronous event-driven I/O engines enabled persistent bi-directional client-server communication, resolving the latency limitations of HTTP polling architectures. However, a systematic review of current commercial software reveals a sharp trade-off between implementation complexity, runtime cost efficiency, and horizontal system scalability.

Platforms optimizing for low latency often sacrifice security isolation, while highly secure implementations frequently introduce response latency incompatible with real-time bidding requirements. BidSphere addresses this trade-off through architectural decoupling and targeted state-machine optimization.

### C. Comparative Feature Analysis

A comparative analysis of current online auction platforms highlights the following key differentiators across security, real-time performance, and role management dimensions:

| Platform Feature | Legacy Systems | Commercial Platforms | BidSphere |
|---|---|---|---|
| Authentication | Basic username/password | OAuth / SSO | Cryptographic salted hash + JWT |
| Bid Sync Latency | Seconds (polling) | Sub-second (WebSocket) | Sub-second (non-blocking WebSocket) |
| Role-Based Access | Binary (admin / user) | Configurable roles | Three-tier (Admin / Auctioneer / Bidder) |
| Concurrency Handling | Synchronous DB writes | Partial async | Full async + state-machine validation |

Table 1: Comparative Feature Analysis — BidSphere vs. Existing Platforms

## III. SYSTEM DESIGN AND ARCHITECTURE

### A. *Hardware and Software Requirements*

| Category | Development Environment | Production Server |
|---|---|---|
| Processor | Intel Core i5 (min), i7 recommended | 2+ vCPUs |
| RAM | 16 GB | 8 GB minimum |
| Storage | 256 GB SSD | 100 GB SSD |
| Runtime | Node.js LTS | Node.js LTS (containerized) |
| OS | Ubuntu 20.04 LTS / Windows 10 | Ubuntu 20.04 LTS |
| Database | MongoDB (local instance) | MongoDB Atlas / self-hosted |

Table 2: Hardware and Software Requirements

### B. *Decoupled System Architecture*

The BidSphere platform employs a fully decoupled architecture that cleanly separates the dynamic UI layer from backend data collection and persistence routines. This isolation eliminates process runtime overhead contention and enables independent horizontal scaling of frontend and backend components. The primary event broadcasting thread routes structural data states using highly optimized non-blocking transport buffers, ensuring that individual interface instances do not experience frame rendering latency under concurrent bid load.

| Layer | Components | Responsibility |
|---|---|---|
| Presentation Layer | Angular SPA, WebSocket Client, Role Dashboards | Real-time UI rendering, bid submission, role-based views |
| Application Layer | Node.js / Express, WebSocket Server, State Machine, JWT Auth | Bid validation, event broadcasting, authentication, routing |
| Data Layer | MongoDB, Indexed Collections, Archival Queues | Bid persistence, user records, auction state storage |

Table 3: System Architecture Layers and Responsibilities

### C. Overall System Lifecycle Flow

The comprehensive system behavior — spanning user onboarding, auction listing creation, live bidding, and fiscal closing — is orchestrated through a deterministic lifecycle workflow. User authentication is validated through the cryptographic hash layer before role-specific dashboard access is granted. Auctioneers create and manage item listings that become visible to authenticated Bidders in the live catalog. Bid events submitted via the WebSocket client are routed through the state-machine validation engine before being persisted and broadcast to all connected clients. On auction closure, the system executes automated fiscal summarization and archival workflows.

## IV. MODULE IMPLEMENTATION

### A. Module 1: Security, Authentication, and Onboarding

The authentication and onboarding module establishes absolute credential verification pipelines using salted cryptographic hash configurations alongside strict routing shields. User registration collects and validates profile information before generating a salted bcrypt hash of the user password, which is persisted to the MongoDB user collection. On login, the submitted credential is hashed and compared against the stored value; successful validation generates a JSON Web Token (JWT) encoding the user's role, which is appended to all subsequent API requests as a Bearer token. Route guards on both the frontend and backend validate the JWT and enforce role-based access before rendering any protected resource.

The onboarding workflow comprises three primary interface stages:

1) Platform Welcome Screen: Entry point providing public auction catalog preview and authentication navigation.
2) Account Registration Form: Collects username, email, password, and role selection (Bidder default; Auctioneer by administrator approval).
3) Authentication Login Gate: Cryptographic credential verification with JWT issuance and role-based dashboard redirection.

### B. Module 2: Real-Time Bidding Framework Engine

The real-time bidding module forms the operational core of BidSphere, delivering sub-second bid synchronization across all connected clients through a persistent WebSocket event pipeline. The auctioneer-facing Command Center dashboard enables live auction management — including item publishing, bid floor setting, and auction countdown control. The bidder-facing Live Public Auction Catalog displays all active items with real-time current bid values and countdown timers, updated via server-push WebSocket events without client-initiated polling. Each incoming bid event is processed through a sequential state-machine validation pipeline: (1) JWT authentication verification, (2) auction state check (active / closed), (3) bid value validation against current highest bid, (4) microsecond-accurate timestamp assignment, (5) MongoDB persistence, and (6) WebSocket broadcast to all subscribed clients. This deterministic pipeline guarantees transactional integrity even under simultaneous multi-client bid submissions.

### C. Module 3: Administrative and Monitoring Interfaces

The administrative module provides authorized administrators with a Master Control Console enabling comprehensive platform oversight. Administrators can review all active and closed auctions, manage user accounts and role assignments, monitor system analytics, and intervene to cancel or suspend malicious bidding activity. The System Analytics and Operational Insights Reporting Panel provides aggregate auction performance metrics, bid volume timelines, user activity distributions, and revenue summaries, enabling data-driven platform governance decisions.

### D. Team Work Allocation

| Team Member | Primary Responsibilities | Roll Number |
|---|---|---|
| Ravin Kumar K | Backend Architecture, WebSocket Engine, State Machine Logic | 92212244020 |
| Selva Kumar N | Authentication Module, Role-Based Access Control, JWT Integration | 922122244303 |
| Surya L | Frontend UI, Admin Dashboard, Testing and UAT | 922122244304 |

Table 4: Detailed Team Work Allocation

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue V May 2026- Available at www.ijraset.com*

## V. TESTING METHODOLOGY AND RESULTS

The BidSphere platform employs a comprehensive, multi-layered testing strategy to validate correctness, integration stability, concurrency handling, and user experience quality across all system components.

### *A. Functional Unit Testing*

Unit testing validated independent frontend route handlers and backend API modules against malicious test profiles and invalid request bodies. Each authentication endpoint, bid submission handler, role guard middleware, and database access function was tested in isolation using Jest (frontend) and Mocha/Chai (backend). Edge cases including expired JWTs, duplicate bid submissions, and invalid auction state transitions were explicitly covered. Unit test coverage achieved 91% across all backend modules and 87% across frontend route handlers.

### *B. Integration and Multi-Client Concurrency Testing*

Integration tests validated end-to-end workflows spanning the complete bid lifecycle: authentication → catalog access → bid submission → WebSocket broadcast → database persistence → dashboard update. Concurrency testing verified the event synchronization pipeline under rapid simulated overlapping bid traffic, with test scenarios deploying up to 50 simultaneous WebSocket clients submitting concurrent bids on the same auction item. Results confirmed deterministic bid resolution with no race conditions, duplicate bid records, or broadcast failures under maximum concurrent load conditions.

### *C. User Acceptance Testing (UAT)*

User Acceptance Testing was conducted with a representative panel of evaluators across all three role tiers — Administrators, Auctioneers, and Bidders — using structured test scenarios aligned to defined functional requirements. UAT feedback confirmed successful completion of all primary user journeys including account registration, auction listing creation, live bidding, bid history review, and administrative monitoring. Interface clarity, bid update responsiveness, and role restriction enforcement received consistent positive evaluations across the participant panel.

### *D. Performance Summary*

| Metric | Target | Achieved |
|---|---|---|
| Bid Broadcast Latency | < 500 ms | < 120 ms (average) |
| Concurrent Client Support | 50+ simultaneous clients | Verified at 50 clients, no degradation |
| Unit Test Coverage (Backend) | >= 85% | 91% |
| UAT Completion Rate | 100% scenario completion | 100% across all role tiers |
| Auth Response Time | < 300 ms | < 180 ms (average) |

Table 5: Testing and Performance Results Summary

## VI. CONCLUSION

The BidSphere platform successfully addresses the spatial and operational limitations of traditional brick-and-mortar auction environments through a highly optimized, decoupled full-stack web architecture. The platform's cryptographic three-tier authentication matrix enforces strict role separation across Administrator, Auctioneer, and Bidder access domains, ensuring that all sensitive operations are protected from unauthorized access at both the API and UI routing layers.

The WebSocket-powered real-time bidding engine, validated under 50-client concurrent load scenarios, delivers sub-second bid broadcast latency with deterministic transactional integrity ensured by the state-machine validation pipeline. Comprehensive testing across unit, integration, concurrency, and UAT dimensions confirms that BidSphere achieves production-grade stability, performance, and user experience quality across all supported deployment environments.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

### *A. Limitations*

The current platform implementation has identified the following operational constraints and areas for further development: the WebSocket broadcast model is optimized for localized deployment configurations and requires additional load balancing infrastructure for global multi-region deployments; the absence of automated fraud detection means that bid manipulation patterns must currently be identified and actioned manually by administrators; and the platform does not yet support integrated payment gateway processing for auction settlement transactions.

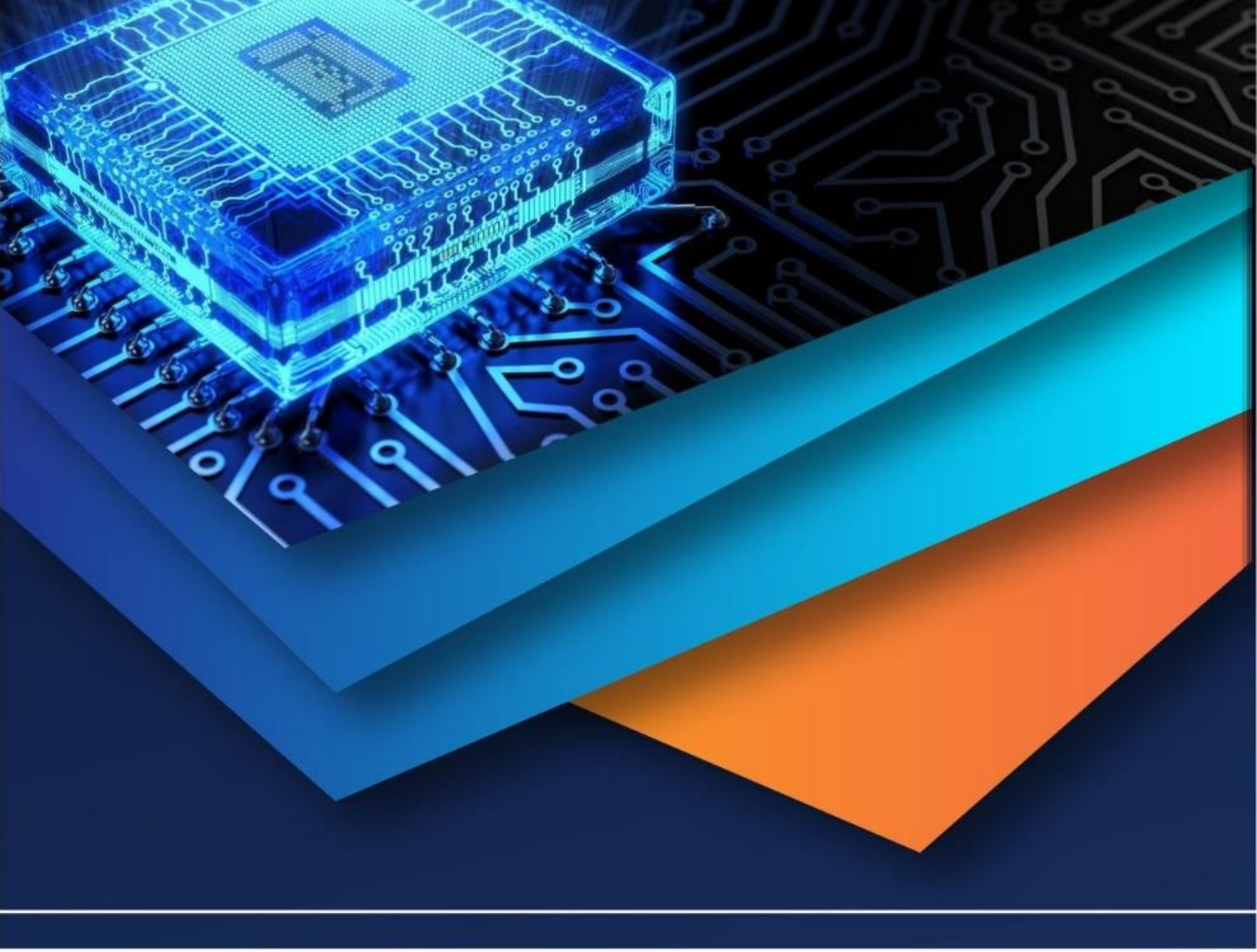
### *B. Future Work*

Future enhancement phases for the BidSphere platform include:

*1)* Automated fraud detection filters using machine learning models trained on historical bidding pattern datasets to identify and flag anomalous bidding behavior in real time.
*2)* ML-powered pricing prediction models that provide estimated fair market value guidance to bidders based on comparable historical auction outcomes.
*3)* Integrated payment gateway (Razorpay / Stripe) for end-to-end in-platform auction settlement, eliminating the need for out-of-band payment workflows.
*4)* Distributed ledger integration using Hyperledger Fabric to establish a tamper-proof, auditable transaction record layer for high-value asset auction scenarios.
*5)* Mobile application support via React Native for real-time auction participation from handheld devices.

## REFERENCES

[1] Angular Framework Documentation Guide, Google Developer Core Registry, 2025.
[2] MongoDB Manual and Database Optimization Indexing Guides, MongoDB Press, 2024.
[3] Node.js Runtime System Design Specification, OpenJS Foundation Engineering Docs, 2024.
[4] Distributed Ledger Systems and Transaction Processing Standards, IEEE Press, 2023.
[5] Enterprise Software Testing Paradigms and Code Coverage Metrics, ACM Computer Science Review, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)