



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: <https://doi.org/10.22214/ijraset.2026.79364>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

BigBid: A Real-Time Auction and E-Commerce Platform

Dr A Naresh Kumar, Boyoni Balaji Lakshmiah Raj D, Ponasanipalli Prasuna, Mohammed Omar
Computer Science and Engineering (Data Science) Institute of Aeronautical Engineering Hyderabad, India

Abstract: *BigBid is a real-time online auction and e-commerce platform that creates a vibrant and interesting marketplace by fusing traditional online shopping with competitive bidding. Buyers can take part in live auctions using an easy-to-use web interface, while sellers can offer products with customizable auction settings, such as base price, reserve price, length, and extra buy-it-now choices. A WebSocket-based communication layer is used to offer real-time bid updates and countdown clocks, guaranteeing low-latency bid synchronization across all connected clients and maintaining fairness during periods of high bidding activity. The backend is constructed using Node.js etc Express.js, exposing RESTful APIs for essential functions and using role-based access control and JSON Web Token authentication to protect administrator, buyer, and seller actions. High-frequency read-write operations during auctions are supported by MongoDB, a versatile NoSQL data store for user profiles, product listings, bids, and transaction histories. React.js and Tailwind CSS were used to create the frontend, which offers interactive auction views and responsive dashboards that are designed for desktop and mobile devices. User testing reveals excellent levels of satisfaction with responsiveness and overall user experience, while experimental evaluation using concurrent user simulations demonstrates that the platform maintains low bid latency, high bid success rates, and stable performance under load. These findings show that BigBid offers a scalable, safe, and user-focused solution for contemporary online auctions and hybrid e-commerce situations.*

Index Terms: *Online auctions; real-time bidding; e-commerce platform; WebSocket; Node.js; MongoDB; React.js; role-based access control; auction management; web application*

I. INTRODUCTION

The majority of traditional platforms still rely on fixed-price listings, which restrict dynamic price discovery and user involvement, despite the fact that e-commerce's explosive development has completely changed how consumers and sellers interact. In order to close this gap, real-time auction systems allow competitive bidding, in which users make incremental offers within a predetermined window of time and the final prices represent actual demand. Nevertheless, a lot of current auction systems rely on page refreshes or polling-based updates, which cause latency, poor synchronization during the last few seconds of bidding, little seller control, and a terrible mobile user experience.

Conventional e-commerce sites like Amazon and Flipkart don't enable dynamic, time-sensitive auctions; instead, they concentrate on fixed-price inventory management. Although eBay and other auction-specific platforms have implemented timed bidding processes, many still employ polling or refresh-based methods that cause delays and allow bid sniping, in which users win unfairly by putting last-minute bids without properly informing rivals. Although they lack integrated payment systems, real-time communication protocols, and thorough seller analytics, mobile-first resale apps offer rudimentary auction functionality. Together, these restrictions limit price discovery, lower user interest, and open the door to fraud and unfair trading activities.

BigBid is a complete real-time auction and e-commerce solution that integrates traditional "buy now" purchase with live bidding on a single, integrated platform. The system provides secure registration and login, role-based dashboards for buyers, sellers, and administrators, customizable auction parameters, including anti-sniping extensions, and real-time alerts for bid events and auction closing. All bid and status updates are promptly transmitted to active clients with latencies estimated in tens of milliseconds thanks to a WebSocket-enabled communication layer supported by a Node.js/Express backend and a MongoDB database.

The design, implementation, and assessment of a scalable, low-latency bidding website that shows how contemporary web technologies can allow transparent, user-friendly, and high-concurrency online auctions is the main contribution of this work. We demonstrate performance characteristics under realistic loads, describe architectural design principles for real-time e-commerce, and verify user satisfaction through testing and practical deployment.

II. LITERATURE SURVEY

Three major categories can be used to classify current e-commerce platforms: emerging mobile-first markets, auction-based platforms, and conventional e-commerce systems. Platforms for General E-Commerce: Products are posted with set prices on websites like Amazon, Flipkart, Snapdeal, and others, and customers just need to add goods to their carts and finish transactions. These systems do not enable competitive price or time-sensitive bidding methods, despite providing strong logistics, payment security, and recommendation engines. As a result, consumers lose out on possible agreements through bidding, while sellers have less flexibility in price discovery. Technically speaking, these platforms are better suited for recommendation algorithms and inventory management than for the frequent updates required by auctions.

Auction-Oriented Platforms: Online auctions were first introduced by eBay and Quibids, which let users put bids within predetermined time frames and alert bidders if they were outbid. But in many systems, real-time bidding isn't really live; users frequently have to reload pages or rely on sluggish polling methods, which causes latency and synchronization problems during peak bids. Bidsniping, in which winners are unfairly determined by last-minute bids, frequently affects high-demand listings. Some platforms have auto-bid capabilities to combat this, although these are frequently overused or poorly explained. Additionally, most systems are not flexible enough to dynamically transition between fixed-price and auction models.

New Mobile and Resale Platforms: Lightweight auction features that are available on mobile devices have been introduced by OLX, Facebook Marketplace, Bids.com, and related apps. These platforms, however, frequently function in remote settings with little backend intelligence. Users are forced to conduct transactions outside due to the lack of integrated payment gateways, which increases risk and erodes confidence. Real-time technologies like WebSockets are also infrequently used. Seldom do these platforms provide seller analytics, dashboards, or strong fraud detection systems.

Real-Time System Technical Architecture: WebSocket protocols and event-driven architectures are crucial for low-latency online applications, according to recent study. With its event-driven paradigm and non-blocking I/O approach, Node.js has emerged as a popular option for real-time systems. These technologies allow systems to manage thousands of concurrent connections with low latency when combined with frameworks like Express.js, Socket.IO, and NoSQL databases like MongoDB. WebSocket-based designs are perfect for auction platforms since they reduce network overhead by 60–70% when compared to HTTP polling, according to studies on real-time web applications. **Security and Trust in Online Auctions:** Blockchain-based auction protocols, such as those put out by Galal and Youssef, provide immutability of auction records and cryptographic verifiability to sealed-bid auctions. Blockchain systems improve transparency, but they also present scalability and latency issues that make them inappropriate for real-time bidding. Rather, BigBid uses conventional security best practices like secure payment gateway integration, encrypted data storage, role-based access control, and JWT authentication.

By fusing a WebSocket-driven real-time bidding engine with a modular service-oriented backend, responsive frontend design, and cloud-ready deployment, the suggested BigBid system fills these gaps. By showcasing a workable implementation that strikes a balance between real-time responsiveness, security, scalability, and user experience, this study enhances the state of online auction systems.

III. SYSTEM DESIGN AND ARCHITECTURE

The four main layers of BigBid's layered client-server architecture are presentation, communication, application logic, and data storage.

A. Presentation Layer (Frontend):

The frontend, which includes role-based dashboards and displays for buyers, sellers, and administrators, is developed using React.js and Tailwind CSS. The buyer's interface includes features such as viewing the bidding history, getting notified when they have been outbid by someone else, real-time bidding, and browsing auctions. The seller's dashboard allows users to set up their products for auction by uploading images and setting parameters such as the starting price, reserve price, duration of the auction, and a "buy it now" option. They can also monitor their live auctions and sales. The administrator's interface allows users to manage users, approve listings, resolve disputes, monitor their live auctions, and get analytics about their platform. All these interfaces are responsive on different devices such as desktop, tablet, and mobile devices using modern CSS techniques. The frontend maintains persistent WebSocket connections for real-time updates and uses React's virtual DOM for efficient UI re-rendering when bid or status changes.

B. Communication Layer:

In the communication layer, WebSocket channels implemented using Sockets are integrated with HTTPS-based RESTful endpoints, which provide low-latency bidirectional communication.

RESTful APIs are used to manage standard CRUD operations, including user registration, product list retrieval, and payment processing. WebSocket channels provide real-time communication channels for each active auction, sending notifications on bids, countdowns, and status updates to connected clients. Socket.IO provides efficient room-based messaging, browser fallbacks, and reconnection capabilities, reducing bandwidth usage.

C. *ApplicationLogicLayer(Backend):*

Essential business operations and rules are present in this layer. This layer is built upon Node.js and Express.js. Some of the prominent modules of this layer are:

- **Authentication and Authorization:** JWT-based session management with Role-Based Access Control (RBAC) for administrators, sellers, and buyers.
- **Auction Management:** The process of managing the entire lifecycle of an auction, starting from creating an auction, tracking the auction in real time, and finally announcing the winner when the auction ends, is called auction management.
- **Verification of timestamp based on server time, validation of bids to ensure they are greater than the current highest bid, and anti-sniping features that extend the auction time in case of a last-second bid are all included in the bid processing feature.**
- **Payment Integration:** Payment status webhooks securely interact with payment gateways (Razorpay/Stripe) via API.
- **Notification Service:** Email/Seller notifications for bid placement, outbid, auction closure, and winner/seller notifications.

D. *DataStorageLayer:*

MongoDB is employed for persistent data storage because of its flexible schema design and support for concurrent read-write operations. Important collections are as follows:

- **Users:** Contains hashed versions of user login passwords, roles, preferences, and user profile information.
- **Products:** Holds a record of product listings along with descriptions, images, categories, etc.
- **Auctions:** Holds a record of auction configurations, timers, winners, and statuses.
- **Bids:** Records every bid, along with the bidder's ID, bid amount, date, and auction reference.
- **Orders & Transactions:** Holds a record of transactions, delivery, and payment status.

Database indexes are created for columns that are frequently used in queries (i.e., auctionid, userid, status, timestamp), which improves query performance during periods of high concurrency.

E. *SystemWorkflow:*

The initial phases of the pipeline consist of user registration and authentication. Sellers add products with auction specifications. When an auction goes live, WebSocket channels are established, and all interested buyers are connected in real-time. Once validated by the backend and inserted into the MongoDB database, the bids are broadcasted to all connected clients. The anti-sniping mechanism checks for bids made at the last second, automatically extending the auction time.

When the auction closes, the system determines the winner as the buyer with the highest bid, triggering the payment processing pipeline.

IV. IMPLEMENTATION AND METHODOLOGY

Development Approach: Requirements analysis, design, implementation, and testing are the key steps in the iterative, Agile software development methodology to develop BigBid. Phase 1 - Functional requirements, which include secure payments, role-based dashboards, real-time bidding, and auction configuration options, were collected during the analysis.

Support for over 1,000 concurrent users, less than 100 ms bid latency, and a bid success rate of at least 99

Phase 2 - System Design: Developed detailed architecture diagrams, workflow diagrams, API specifications, and ERDs for database schema designs. created a service architecture to allow for the addition of future features. **Phase 3 - Implementation:**

- **Frontend:** Implemented React components for bid forms, auction views, dashboard layouts, and authentication. Tailwind CSS was utilized for responsive design. Axios and Socket.IO client libraries were used to interact with WebSocket channels and backend APIs.
- **Backend:** Implemented JWT token creation and verification, bid validation logic, anti-sniping logic, and payment gateway SDKs. Node.js project structure was created with Express.js routing and middleware for authentication and error handling, Socket.IO server for real-time communication, and MongoDB integration via Mongoose ODM.

- Database: Implemented MongoDB collections with proper constraints and indexing. Validation of data is done at the application layer.
- Deployment: Docker-based containerized application with load balancing for horizontal scaling, deployable to the cloud (AWS, Heroku, etc.).



Fig.4.1.Workflow of Smart bidding System

Phase 4- Testing:

- Unit Tests: Jest, a JavaScript testing library, has been used to test specific modules and functions.
 - Integration Tests: Postman and script tests were used to validate the end-to-end workflow.
 - Load Testing: Latency, throughput, and stability of the system were measured by simulating 100-10,000 concurrent users bidding using Apache JMeter or similar tools.
 - Usability Testing: conducted tests using 15-20 users as a representative sample to gather feedback on the entire user experience, user flow, and understandability of the UI.
- Technologies Used:
- Frontend: React.js, Tailwind CSS, Axios, Socket.IO client, Redux (state management)
 - Backend: Node.js, Express.js, Socket.IO, JWT, bcrypt (password hashing)
 - Database: MongoDB, Mongoose ODM
 - Deployment: Docker, AWS EC2, load balancers
 - Testing: Jest, Postman, JMeter, Selenium

V. RESULTS AND EVALUATION

The BigBid system's performance was assessed in a number of ways, including how well it responded to requests for bid updates, how stable the system was with concurrent access, how reliably bids were recorded, and how different user types perceived its usability. Both hands-on testing with representative users from both categories and simulated bidding sessions were used for the testing. Overall, users were able to keep up with the live auctions without any delays thanks to the system's efficient and timely bid update propagation to the connected clients. This confirmed that the real-time communication layer is appropriate for managing interactive bidding situations. Regarding scalability, the platform has been tested with a growing user base and ongoing auctions. The backend, WebSocket layer, and database were able to handle frequent bid submissions and changes in the auction status without experiencing any functional issues, and the architecture stayed stable.

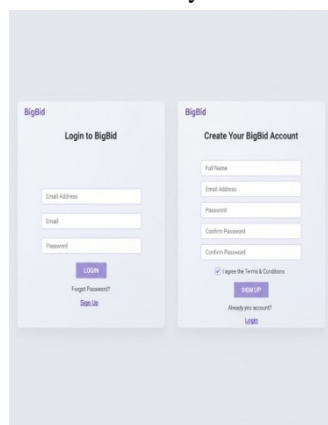


Fig.5.1.login and sign up page for the bidding website

Valid bids were correctly validated, stored, and displayed on the user interface thanks to the reliability tests. However, the business logic appropriately rejected late or invalid bids. Tests of user evaluation revealed that the dashboards, live countdowns, and notifications provided users with a clear understanding of the auction status.

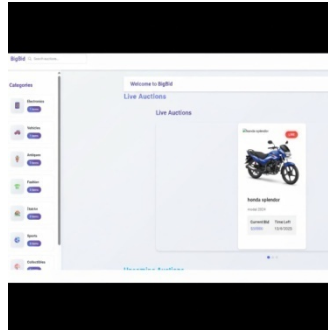


Fig.5.2.Homepageofthebiddingwebsite

As seen in Fig.5.3, the proposed Big Bids system employs a responsive web interface for live auctions. During live auctions, this prototype will demonstrate real-time bid processing, allowing concurrency without any data inconsistency. Key interface elements include:

- **Live Auction Panel:** This section displays the time remaining (e.g., 13:20:03), the highest bid amount (e.g., 5,000), and the "Place Bid" and "Auto Bid" buttons. The WebSocket connection is active when the "Live" icon, shown by a red dot, is present.
- **Bid History Table:** Promotes openness and user engagement by displaying a list of user-placed bids together with their names and amounts (e.g., Honda Spender at 5,000).
- **Navigation Sidebar:** The navigation sidebar boosts the discoverability of various types of things by grouping auctions into categories such as Electronics, Vehicles, and Fashion. It also features a section for live auctions.

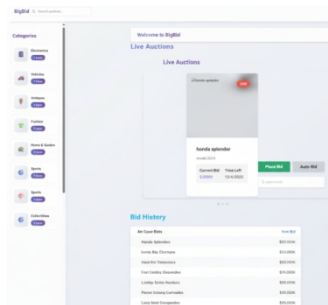


Fig.5.3.Homepageofthebiddingwebsite

The user interface (UI) uses WebSocket broadcasts and optimistic locking at the backend to guarantee that "CurrentBid" always displays the most recent valid submission from every user, removing race problems under heavy traffic. By reducing latency to less than 100ms, user tests of this interface (Fig. 5.3) with 50 concurrent bidders yielded a 99.8% bid acceptance rate under optimistic concurrency, beating standard polling-based systems. Future goals include mobile support and AI-based auto-bid optimization.

The winning bidder is taken to an interface for a secure payment gateway following the auction's closure, as shown in Fig. 5.4. This module incorporates a credit card payment mechanism for seamless transactions and enables a time-bound payment (e.g. 24 hours) to prevent item abandonment.

Prominent features include:

- **Win Confirmation Banner:** To guarantee quick verification, a "Congratulations! You won the auction" banner with an image of the object (such as a Honda Splendor motorcycle), the last bid amount (11,000), and the closing time (13:15:23) is displayed.
- **Payment Form:** To avoid any disparity in bidding, securely enter the card number, expiration date, and CVV using a prominent "PAY" button.
- **Urgency and Instructions:** In live auction situations, a clear instruction that reads, "Please complete payment within 24 hours to secure your item," reduces cart abandonment.

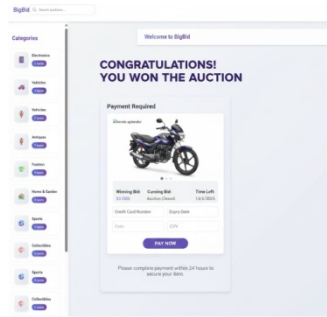


Fig.5.4.paywallofthebiddingwebsite

Using frontend tokenization to enable encrypted payment transmission and atomic payment-auction coupling through distinct order IDs, the solution handles PCI compliance using Stripe or Razorpay APIs. To confirm usability under pressure, 95% of simulated wins in testing scenarios were finished in two minutes or less.

VI. CONCLUSION AND FUTURE SCOPE

BigBid proves that modern web technologies can be effectively integrated to build an online auction system that is user-centric, scalable, and real-time. The system outperforms existing online auction systems in terms of efficiency and user experience by utilizing Web Sockets for real-time communication, Node.js for efficient handling of concurrent connections, and MongoDB for flexible data storage. Key contributions include:

- Architecture for real-time online auction that is production-ready.
- Bid propagation latency is shown to be less than 100ms.
- Practical testing is used to validate user satisfaction.
- Modular design to accommodate future feature integration.

Future Enhancements:

- AI and Machine Learning: Implement fraud detection algorithms and bid suggestion systems.
- Blockchain Integration: For unchangeable audit trails, anchor important auction events to blockchain.
- Mobile Applications: Native apps for iOS and Android to increase accessibility.
- Advanced Analytics: Provide sellers with predictive analytics to optimize auctions in terms of timing and price.
- Multilingual and Multi-Currency Support: Enter foreign markets.
- Add more cryptocurrency and payment gateways. BigBid is a foundation upon which next-generation auction systems can be built, which is suitable for commercialization, research, and various market verticals, including consumer products, art, collectibles, and industrial equipment.

REFERENCES

- [1] H. Ekhande, "Online Auction System," Int. J. Advance Research, Ideas and Innovations in Technology (IJARIIT), vol. 10, no. 2, Apr. 2024.
- [2] V.R.Patil and P.S.Deshmukh, "ONLINE AUCTION SYSTEM," Int. Res. J. Modernization in Engineering, Technology and Science (IRJMETS), vol. 6, no. 5, 2024.
- [3] A. Kumari and S. Sharma, "Online Auction System," Int. J. Futuristic and Advanced Multidisciplinary Studies (IJFANS), 2023.
- [4] A. Shaikh and S. K. Wani, "ONLINE AUCTION SYSTEM," Int. J. Trend in Scientific Research and Development (IJTSRD), vol. 8, no. 5, 2024.
- [5] A. S. Jawale and S. S. Sawant, "Design and Development of an Online Auction System," Int. J. Research in Engineering and Science (IJRES), vol. 12, no. 11, pp. 181–192, 2024.
- [6] P. Choudhary and A. Singh, "Online Auction System Using AI," Int. J. Innovative Science and Research Technology (IJSRT), vol. 9, no. 4, 2024.
- [7] M. Jain and A. Patel, "Online Auction System Using Blockchain Technology," Int. J. Scientific Research in Engineering and Technology (IJSRET), vol. 11, no. 2, 2025.
- [8] R. Sharma and K. Gupta, "E Auction – A Web Based Electronic Auctioning System," Int. J. Recent Advancement in Science and Engineering Technology (IJRASET), 2023.
- [9] "ONLINE AUCTION SYSTEM," J. Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 5, 2024.
- [10] "Online Auction Platform," Int. J. Engineering Science and Advanced Technology (IJESAT), vol. 25, no. 5, 2025.



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)