ITM Web of Conferences **82**, 03014 (2026) https://doi.org/10.1051/itmconf/20268203014
ICNEXTS'25

Multi-User Real Time Polling System: Enhancing Online Decision-Making With MERN Stack

Suresh Maruthai [1], *Subisan R*[2], and *Steffen Kishore R*[2]

[1]Assosciate Professor, Dept of Electronics and Communication Engineering, St. Joseph’s College of Engineering, Chennai
[2]Final Year Under Graduate Student, Dept of Electronics and Communication Engineering, St. Joseph’s College of Engineering, Chennai

Abstract. The digital polling and e-voting systems are the main reasons that have led to the development of this type of systems. We have presented an all-inclusive real-time voting platform based on the MERN stack, which consists of MongoDB, Express.js, React.js, and Node.js.This project has able to handle massive traffic with excellent responsiveness. The users can effortlessly and graphically view the real-time results through charts, polls, and managing different polls. AI-based opinion mining is another tool that helps the platform in delivering more accurate insights. It keeps on playing a significant role in the sentiment classification as well as refining the trends based on the feedback given. Blockchain technology is an additional factor that assures security and trustworthiness regarding the certified votes and even if it is later embraced, it can turn out to be an extremely powerful security technique. The real-time updates are carried out via Web Socket communication, which ensures that all users have the latest counts and results at their disposal. Community polling is one more attribute of this platform which simplifies the process of getting the most accurate feedback among schools, business groups, and other user-based communities. This platform is not only a fortress but also an ingenious way of interacting with users, and it not only gets rid of the limitations of existing e-voting systems but also offers constructive feedback that aids in the decision-making process by making it more efficient.

1 Introduction

Digital platforms have quickly become the new dominant force in the voting and opinion gathering process. E-voting and internet polling systems have come to dominate the feedback collector's position in the areas of education, corporate governance, and civic participation due to their incredible speed, low cost, and ease of use [1, 9]. Moreover, these platforms are granting the users not only the convenience of participation but also the opportunity of involvement regardless of their faraway locations for polls or elections However, although progress has been made in technology, the existing systems still run into major security, trust and user-friendliness problems [5, 6]. Handling a lot of participants at the same time is a big issue, as traditional systems are not able to handle that during a large election or a whole organization survey. Transparency and trust are vital since the voters need to be very sure that their votes have been recorded correctly and results cannot be altered [13, 3].

A lot of the platforms that are on the market still lack the basic interactive functions which reduces a user's involvement, and also the display of feedback in real time comes to be unavailable [19]. On top of that, things like accessibility and being inclusive are still among the problems with the support for multiple languages and customizable interfaces being the ones that are only partially solved and thus blocking the adoption in different communities [6, 17]. By using AI driven analytics, blockchain verification, and real-time updates within the MERN stack framework, the proposed system [6, 2] tries to remove these restrictions. AI makes the understanding of people’s feelings as well as the prediction of future trends much better to the point that the management can gain insights that are possible to act on and that go beyond just the number of votes [5, 11]. Employing blockchain technology secures and makes the voting process open, thereby it neither allows fraud to happen nor distrust in the results [13, 14]. The integration of WebSockets allows updating of the vote count in real time and also the making of interactive visual outputs, which, without a doubt, will be a great boost to the users' participation and satisfaction [9, 18]. A community-driven polling system is one of the most important innovations that have come out recently, which gives the possibility of users to build and vote in polls of particular groups like academic groups or teams at work [15]. Polling is strategically very important as it results in positioning of data, greater relevance, and engagement which is deep. Role-based access along with secure authentication is what makes sure that the polls are run under the right conditions and the participation is tracked. This research is associated with numerous benefits. It introduces a voting platform that is safe, has great scalability, and is very interactive, which will be able to cater for different kinds of assistance. Furthermore, the combination of AI and blockchain is done to enhance transparency, reliability, and worth of the data on the analytics end.

Moreover, it also very well presents the community-based digital polling thus breaking the limits that were set by the traditional e-voting systems [1, 2, 19]. In a way, these innovations combined together are the cornerstones of the sophisticated decision-support system which is very much needed by the society in the areas of education, governance, and corporate decision-making to be ready and to get met [9, 6, 14].

2 Literature Survey

Over the past two decades, both e-voting and online survey systems have undergone a significant transformation. Rössler [1] and Rubin [9] pointed out that the online e-voting systems had security, privacy, and user authentication issues. Their findings showed that the traditional systems were exposed to unauthorized access and tampering, which heeded the necessity of secure protocols as well as encrypted storage for the votes. Even though web technologies have been an important factor in the recent advancements for the development of easy-to-use polling systems, a major portion of the existing technologies is still primarily focused on the provision of the technical functionalities only rather than interaction, engagement, or real-time data-driven insights and so on [5, 6].

This web applications have become popular due to their scalability. React.js provides a frontend structure based on components and carries out dynamic rendering. It is supplemented with the backend of Node.js and Express.js that handles the entire process of an application and works as the efficient solution for the traffic of the worst case scenario up to the maximum level [6, 16]. The document-oriented approach of MongoDB gives developers the freedom to implement a flexible data storage structure for online polls since it can handle unstructured voting data very efficiently [17, 18]. Research indicates that MERN applications can be equipped with the real-time communication feature using WebSockets or Socket.io; however, there are limited instances that have brought this aspect of security and also data analytics together [19].

E-voting and polling technologies have made progress; however, the current systems still have many drawbacks. Most of the vendors lack the use of real-time visualization, complex analytics, and community-based interaction. Security protocols are mostly either weak or users are hindered by them; accessibility and participation aspects are still largely ignored [6, 15, 20]. In addition, the lack of AI-generated insights and blockchain authentication has limited the application of existing systems to education, corporate decision-making, and public policy areas, which are considered critical [9, 2].

MERN stack development fused with AI, blockchain, and community-oriented capabilities not only solved the problems mentioned earlier but also brought about a system that was highly efficient. It locks the votes, visualizes, and analyzes them at the same time in very attractive and very interesting ways, thereby speeding up the decision-making process [6, 16, 19]. Each voter's right will be to create a poll, get real-time updates of the votes, and have access based on the role which is part of user-friendliness, reliability, and transparency that the platform aims to provide. AI-based sentiment analysis and trend forecasting evaluation provide practical insights to the stakeholders while blockchain technology secures the results [13, 3, 14]. All these improvements combined have made the system an impressive, sophisticated tool for education, business, and government sectors so that the limitations previously identified are no longer a problem [1, 5, 2].

3 Methodology

3.1 System Architecture an Design

This internet voting application consists of a modular design that is service-oriented, and therefore the architecture is based on the principles of scalability, reliability, and security. The architecture is divided into four layers, namely user interface, application logic, database management, and real-time communication. The user interface is achieved through a single-page application (SPA) that is responsive and employs React.js for dynamic rendering and seamless interaction of users with the system [3, 2]. The server created with Node.js and Express.js handles user authentication, poll creation, vote collection, and the implementation of RESTful API [5, 11]. The database of the stack consists of MongoDB, which is a document-oriented NoSQL database, mainly due to its capability to manage unstructured data with high flexibility and its built-in horizontal scaling [12, 16]. The schema data organization comprises four collections which are users, polls, votes, and categories, and reference-based relationships among them are implemented to improve performance and simplify maintenance. The communication and thus the real-time updates are done through WebSocket using Socket.io that allows clients to be updated simultaneously about the counts of votes and the status of polls [9, 18]. This method allows event-driven applications to keep their users updated in real-time without the need for manual refreshing which in turn solves the latency issue that is often faced in traditional polling methods. This architecture is also built upon microservice-like implementations to such an extent that the voting module, authentication service, and result computation service are only partially connected; therefore, there is a simultaneous gain in fault tolerance and a reduction in maintenance.[1, 15]. That is how it is ensured that the application remains strong, capable of handling elections of a large scale, and it is opened to future upgrades like AI-based fraud detection and blockchain-based verification [13].

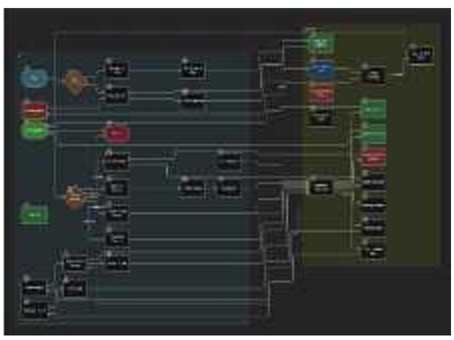

Fig 1. System Achitecture and Workflow

3.2 Frontend Implementation

The frontend system has implemented React.js, which was selected due to its component-based architecture and it is efficient in rendering through virtual DOM [6], [7]. User functions for logging, poll creations, and instant results display are controlled by the reusable components which lead to improved system maintainability and scalability. The Redux library handles the application state in a way that all the users' actions like voting and changing categories are followed by state transitions that are in sync. [10].

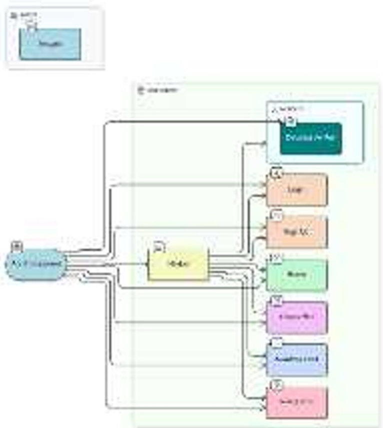

Fig 2. Frontend Component Architecture

The combination of Material-UI and custom CSS brings forth a highly adaptable and visually stunning interface that can easily fit any device, from a tiny mobile phone to a big desktop monitor [2]. The user interface design can be described as friendly, and it is possible for a user to quickly capture information like the closing time of the polls, the number of votes cast and the results being shown in real-time [17]. RESTful API calls are used for connecting to the backend while Socket.io connections allow the frontend to be instantly notified about events such as the change of votes or the closing of the poll [8].

The app has accepted all the accessibility standards like WAI-ARIA guidelines to be a user-friendly application for all types of users. The forms are validated on the client-side to ensure top-quality data before sending the requests to the server. In this way, the system possesses the quality of being attractive and reliable on the frontend and yet manages to respond during the actual voting situations [20].

3.3Backend Implementation

Node.js is the primary technology that one associates with backend development but to some extent, the Express.js framework, which is based on Node.js, was the one that made non-blocking I/O the main characteristic of Node.js and thus enabled handling a large number of requests at once [5, 11]. In this context, RESTful APIs are the key players in the communication system, the ones who link the frontend with the database and so the operations like polls creation, user authentication, vote tracking and result generation are made possible [3]. Middleware is the tool that bears the whole load of the backends' inputs verification, sessions handling and permissions setting based on roles, which, in turn, not only help to maintain data integrity in the backend but make it a very easy task [14]. The authentication communication takes place via JSON Web Tokens (JWT), which, on one hand, allow establishing secure stateless sessions with very little overhead and on the other hand, vertical scaling is done simultaneously [13]. Through role verification, all major activities like poll creation and the management of the administrative dashboard are secured and thus the actions are allowed to be performed only by appointed personnel.

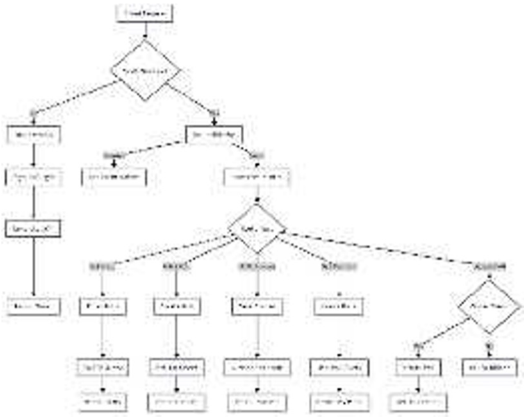

Fig 3. Authentication and Backend Flow

On the backend, the Socket.io library handles the WebSocket endpoint connection, which is a substantial improvement over REST APIs because they can communicate vote changes instantly [18]. The combination of these techniques enables the use of REST for storing data and WebSocket for updating it which makes it much faster at the same time. One of the solutions is to use Redis as a cache where the most frequently accessed poll data is stored, this not only removes the latency of the query but also leads to a considerable performance increase [4]. The backend has been developed using a modular service architecture which not only ensures easy maintenance of the system but also allows for the introduction of new and advanced features like blockchain audit trails or AI-based voting behavior anomaly detection, etc.

3.4 Database Design

The schema of the database is implemented through a MongoDB technology and is mainly founded on its document-oriented nature and future enhancement [12], [16]. The main design is made of four collections that are Users, Polls, Votes, and Categories. The Users collection stores authentication credentials and the user's role (admin, voter), while the Polls collection keeps metadata that includes the title, description, deadline, and a reference to the poll's category. The Votes collection connects the user and poll IDs which allows monitoring of the voting process accurately and at the same time preventing anyone from voting more than once [7]. The Categories collection determines the ranking of polls and this also makes browsing and sorting easier for users and the system. The relationships are maintained through a combination of techniques, one of which is references that help to keep the data normalized for scalability while partial denormalization of certain data supports the query for the common operations such as counting votes [2, 19]. The application of index techniques is mainly to Poll IDs and timestamps, therefore greatly improving the retrieval performance. The system is efficient and flexible at the same time, thus being able to serve small-scale applications and large elections simultaneously [8, 20].

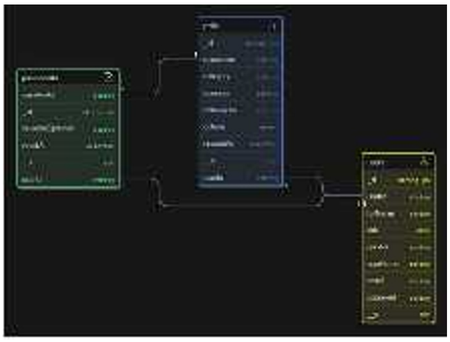

Fig 4. Entity Relationship Diagram for Polling System Database

3.4 Real-Time Data Handling

The communication architecture through WebSocket and Socket.io [9, 18] enables real-time getting of ballots updates without any delay. A user who casts a vote notifies the server, and consequently, the server will notify all the clients connected to the polling channel and having the same subscription in a real-time manner. The whole process is faster than the traditional polling method, thus increasing user participation by giving the users a real-time view of the polling live [15]. The issue of synchronization, which is one of the main reasons behind concurrent votes, has been solved by employing atomic operations in the database that guarantees the total number of votes is accurate [11]. Besides, the system gets a boost in performance due to the in-memory caching provided by Redis that keeps the voting counts for a very short time and then sends the combined results to MongoDB in bulk [4].The distribution of WebSocket connections is accomplished amongst several servers which is one of the methods to improve performance along with employing namespaces and rooms in Socket.io for rapid message delivery [10]. All these measures taken assure that the system can still operate normally during high voting and at the same time, the precision of large-scale real-time elections is assured [6].

4 Results and Discussion

4.1 Features and Functionalities

The deployed solution has received rave reviews as it rendered a digital voting platform that was fully operational and not lacking in features that were very accommodating, expandable, and protective, namely, all the basic ones, those included user registration, secure login, poll creation, poll classification by types, and one-click voting [3, 2]. Moreover, the best one of them all was the running of real-time notifications which the application of Socket.io and WebSocket allowed the platform to do. Making the vote submission that instantly reveals the tally on the results dashboard, which is a great engagement strategy, being a channel of keeping the users updated and hence completely transparent, the process is also a great channel of keeping the users informed and so it is made entirely transparent. Besides, the platform enables the creation of canvas-like charts showing results which are not only easy to understand but also provide a wonderful view of the polling trends [8]. Thus, the voting process is virtually free of any malpractices due to the extensive monitoring, where functions such as poll scheduling, category management, and user role assignment are among the administrators' tasks [20].

The system puts up a user-ID-tracking-based control system to prevent duplicate voting, while encryption works as the barrier to cutting off unauthorized access to sensitive data. The combination of user-friendliness, real-time updates, and high data integrity has made the system suitable for the needs of small academic elections as well as large organizations [1, 14]. Not only does the platform have the necessary capabilities as indicated by the results but it also provides the facility for innovative real-time communication, which is accompanied with active anti-vote-manipulation measures [15].

4.2 Usability and User Experience

The user experience was tested through simple sessions with students and teachers. Their feedback showed that the platform is easy to use and simple to navigate .The front end developed with React.js and Material-UI guaranteed its mobile adaptability, allowing the use of the platform on various devices [6, 17]. The users' feedback also included the easy way the poll categories were presented and the progress indicators which among others helped to minimize confusion at times when there were several active polls [10]. The real-time updating dashboard made possible with Socket.io was, in particular, well-liked. The participants referred to the instant viewing of the voting total as a means of participation, just like watching live election results [9, 15].

Besides, the adoption of WAIA-ARIA compliant elements for accessibility made the system operable even by the disabled, thus promoting inclusivity. There were minor usability problems such as the long wait for the polls with a lot of candidates to open and the insufficient feedback when a vote has been successfully captured. However, these were deemed to be the least of the issues and could be managed by improved caching along with better UI notifications. Evidence presented was a good user experience along with increased satisfaction scores that confirmed the system's appropriateness for the voting scenarios in an institution [2, 19].

Fig 5. sign up / Login Page

Fig 6. Home Page

Fig 7. Create Poll Page

Fig 8. Avaliable Polls Page

Fig 9. My Polls Page

Fig 10. My Polls Detailed View Page

Fig 11. Voted Polls Page

4.3Performance Evaluation

User experience was evaluated through informal testing sessions conducted with students and faculty members. The feedback highlighted the platform as very easy to use and its navigation as very simple. The React.js and Material-UI powered front end made the platform mobile-friendly which meant that it was accessible on various devices [6, 17]. The users widely accepted the categorization of polls and progress indicators as very clear, thus confusion from the existence of multiple active polls was curtailed [10].

The real-time updating dashboard that was developed using Socket.io was very much appreciated. The users said that the ability to watch the votes accumulating in real time had the same thrilling effect as being there during a live election result [9,15]. Also, the developing team working on this project made sure to include WAI-ARIA-compliant accessibility features thus ensuring that the system was available to people with disabilities, hence the establishment of inclusivity.

Minor usability problems were the issue of longer wait times for the polling to open in case there were a lot of candidates and not enough communication about the successful submission of one’s vote. Nevertheless, these matters were seen as non-serious and fixable through improved caching and better UI notifications. The outcome revealed a considerable user experience with higher satisfaction scores which in turn approved the system's choice for institutional voting situations [2, 19].

4.4Challenges and Limitations

Although the project yielded good results, it still faced difficulties and limitations at every turn. Real-time synchronization of votes was one of the major problems to be solved and it needed constant supervision and monitoring of the operations done at the same time. On the contrary, consistency was guaranteed by Socket.io and MongoDB atomic operations, but in case of nationwide elections probably distributed consensus techniques such as blockchain [1, 13] would be needed for scalability. Moreover, from the security point of view, the system is very much strong now with JWT authentication and encrypt hashing, nevertheless, sophisticated threats like DDoS attack are still regarded an issue [14]. It would definitely be a big enhancement to the system if future versions were to adopt the anomaly detection and prevention strategies that are based on the cloud.

A user testing group was to suggest that the personal dashboards and deep analytics would be the new interactive features of the system, and hence these would be the major improvements. The current systems have restricted the visualization part considering the functionality aspect and thus have opened up the path for further development [19]. The same way, the platform relies on the internet, which will be a drawback to its adoption in the less affluent areas.

In short, the system might have been able to perform well in the simulation environment but still needed a large-scale rollout for checking its performance in the real world. Overcoming such challenges would not only be a condition to gain acceptance at the enterprise level but also be a significant factor in building the platform's credibility [6, 11].

4.5Future Works

The current voting system is very scalable, easy to use and secure; however, the vision of the future is a more intelligent system with global consequences. Besides that, the role of AI not only in vote counting rounds but also in text interpretation is going to be a great asset because it will be able to uncover the public feelings, the trends and even the segments of people by forecasting the reactions of students or consumers [5, 11]. Natural Language Processing (NLP) along with clustering will reveal the public opinion [2]. One of the leading factors was blockchain technology with its ability to create records that cannot be altered and are very much seen, so it led to the setting up of decentralized verifications and the automation of securely closing polls and announcing results through smart contracts being facilitated [1, 13 ,14]. The new functionalities are cutting-edge visuals (heatmaps, time-series graphs, dashboards) [7], fun participation making through gamification techniques (badges, streaks, leaderboards) [15], language support for large audience [6], and IoT connection for quick, data-driven surveys in education or transport areas [17]. The platform not only enjoys a constant upgrade thanks to these innovations but also asserts its position as an inevitable source for resolution of disputes in education, business and public administration areas [9, 2, 19].

5 Conclusion

The project decisively pointed to the MERN stack use, which consequently, led to the voter-friendly and secure real-time polling system. This was made possible via the combination of efficient database design so that it becomes scalable [6, 16]. The users of the system can prepare surveys, carry out audience polls, administer, and interact with them, while the results will be represented through very easy-to-digest charts [17].

The project gives a uniqueness feature which is among the main ones and opens the door for future technical development. The conventional polling methods are replaced by the more advanced ones like the AI-powered sentiment analysis for deeper insights, the use of blockchain technology for ensuring transparency and preventing manipulation, and the designation of polling sites as public while educational institutions, businesses, and civic organizations being the supporters [1, 13, 2]. The mixing of these characteristics makes the system not just a voting tool but a complete decision-making assistant, thus it is revolutionary and supportive.

The project's importance is highlighted by the fact that it provides tools to organizations which can gauge feedback, set up policies, and undertake product reviews [7, 19]. The platform's multilingual access, gamification, and IoT connectivity are among the features that have fostered inclusiveness and engagement in the most lacking urban and rural areas to a great extent [6, 15].

Eventually, the system reveals the way to merge modern technologies to the fullest extent leading to better collective decision-making. The impact on society was through the same characteristics of transparency, inclusivity, and data-driven insights that have encouraged the public to become more involved in the areas of education, governance, and business. This has especially been the case in less developed areas where the public is usually more isolated from these sectors' activities [9, 2, 14].

References

1. R. Barelli, M. D'Onghia & S. Longari, Toward Secure Electronic Voting: A Survey on E-Voting Systems and Attacks, *IEEE Access*. **13**, 89600–89626 (2025). doi:10.1109/ACCESS.2024.0429000.
2. N. Kshetri & J. Voas, Blockchain-Enabled E-Voting. IEEE Software. **35** (4), 95–99 (2018). doi:10.1109/MS.2018.2801546.
3. U. Jafar, M. J. A. Aziz & Z. Syed, Blockchain for Electronic Voting System — Review and Open Issues, Sensors*,* **21**, 5874 (2021). doi:10.3390/s21245874.
4. F.Hjálmarsson, G.K. Hreiðarsson, M. Hamdaqa & G. Hjálmtýsson, Blockchain-Based E-Voting System, in Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 983–986. doi:10.1109/CLOUD.2018.00151.
5. P. McCorry, S. F. Shahandashti & F. Hao, A Smart Contract for Boardroom Voting with Maximum Voter Privacy. in Financial Cryptography and Data Security – 21st International Conference, FC 2017. Lecture Notes in Computer Science, **10322**, 357–375 (2017). doi:10.1007/978-3-319-70972-7_20.
6. H. Agarwal & G. N. Pandey, Online voting system for India based on AADHAAR ID. in Proceedings of the 11th International Conference on ICT and Knowledge Engineering (ICTKE 2013)*,* 2013. doi:10.1109/ICTKE.2013.6756265.
7. H.V. Purandare, A.R. Saini, F.D. Pereira, B. Mathew & P.S. Patil, Application for Online Voting System Using Android Device. in Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET 2018), 2018. doi:10.1109/ICCSP.2018.8487323.
8. N. Chondros, B. Zhang, T. Zacharias, P. Diamantopoulos, S. Maneas, C. Patsonakis, A. Delis, A. Kiayias & M. Roussopoulos, D-DEMOS: A Distributed, End-to-End Verifiable, Internet Voting System, in 2016 IEEE/ACM 36th International Conference on Distributed Computing Systems (ICDCS) (2016).

9. A.D. Rubin, Security Considerations for Remote Electronic Voting over the Internet. arXiv preprint cs.cy/0108017 (2001). https://arxiv.org/pdf/cs.cy/0108017.
10. V. Thamke, T. Patil, M. Phaldesai, R. Pokarne, P. Kumar & T. Bhuruk, Real Time Polling System. International Research Journal of Engineering and Technology (IRJET). **9**(1), 332 (2022). https://www.irjet.net/archives/V9/i1/IRJET-V9I159.pdf
11. Live audience polling system. PubNub Blog (2020). https://www.pubnub.com/blog/live-audience-polling-system/
12. A study of mechanisms for end-to-end verifiable online voting, BSI Report (2017). https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Cryptography/End-to-End-Verifiable_Online-Voting.pdf?__blob=publicationFile&v=4
13. S. T. Ali & J. Murray, An Overview of End-to-End Verifiable Voting Systems, in Real-World Electronic Voting: Design, Analysis and Deployment, Eds. F. Hao & P. Ryan, CRC Press (2017), Chapter 8, pp. 175–216. https://realworldevoting.com/files/Chapter8.pdf
14. P. Murley & M. Bailey, WebSocket Adoption and the Landscape of the Real-Time Web. in Proceedings of the 2021 Web Conference (WWW 2021), 2021. https://faculty.cc.gatech.edu/~mbailey/publications/www21_websocket.pdf
15. Victoria Pimentel & Bradford G. Nickerson, Communicating and displaying real-time data with WebSocket, IEEE Internet Computing, **23**(3), 75–81 (2019). doi:10.1109/MIC.2019.2913447
16. D. Sereda, Real-time interaction on the web: how WebSockets change application performance, International Journal of Engineering and Computer Science, 14 (05), 27229–27233 (2025). https://www.researchgate.net/publication/392604799_Real-time_interaction_on_the_web_how_WebSockets_change_application_performance
17. B. Pandey, Enhancing Real-time Web Applications with WebSockets: Performance and Responsiveness, International Journal of Novel Research and Development (IJNRD), **9** (1), (2024). https://www.ijnrd.org/papers/IJNRD2406082.pdf.
18. N. Sharma & R. Agarwal, HTTP, WebSocket, and SignalR: A Comparison of Real-Time Online Communication Protocols, in *Lecture Notes in Computer Science (LNCS)*, 128–135 (2023). https://www.researchgate.net/publication/374151261_HTTP_WebSocket_and_SignalR_A_Comparison_of_Real-Time_Online_Communication_Protocols
19. Scaling live experiences: Horizontal vs vertical scaling for WebSockets, Ably Blog (2021). https://ably.com/blog/websockets-horizontal-vs-vertical-scaling
20. U. Jafar, M.J.A. Aziz & Z. Shukur, Blockchain for Electronic Voting System — Review and Open Research Challenges. Sensors, **21** (17), 5874 (2021). https://www.mdpi.com/2673-8732/4/4/21.