iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71621

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Eye Webinar: A Web-Based Communication Platform for Seamless Remote Education

Shaswat Agrahari[1], Aditi Pandey[2], Arpita Singh[3], Subham Thakur[4], Dr. Sunil Kumar Verma[5]
[1, 2, 3, 4]*B.Tech, Student,* [5]*Head of department, Department of CSE, Feroze Gandhi Institute of Engineering and Technology, Rae Bareli, U.P.*

Abstract: As our world grows increasingly digital, the ability to communicate remote is essential, whether for personal or professional reasons. Remote meetings are often limited to time, cost, and logistical constraints—the latter of which can be exacerbated during global disruptions. To address some of these issues, we present Eye Webinar, a real-time video conferencing web application that enables quality communication in a browser through video and audio with multiple users. Eye Webinar takes advantage of WebRTC, a free, open-source, technology that allows peer-to-peer real-time compressed audio, video, and data to be sent using simple JavaScript APIs. Built on the MERN stack (MongoDB, Express.js, React.js, Node.js), we employed current best practices in web developing to create a reliable and scalable conferencing application. Our key features include live video / audio calls, direct chatting, screen sharing, file sharing, and user authentication. Eye Webinar seeks to provide high-quality, cross-device compatible communication with easy-to-use interfaces to make to the product more accessible for collaborative use in education, corporate communication, emergency services, and remote assistance. This research paper will explain the technical rationale behind our application design, development approach, and the social impact of providing a new secure, extensible web-based conferencing application.
Keywords: WebRTC, React.js, Node.js, Express.js, MongoDB, Socket.io, JWT, HTTPS, REST APIs, Media Servers, Material-UI, Crypto, Bcrypt, Real-Time Communication, File Sharing, Screen Sharing, Audio/Video Calls, Cross-Device Support, Secure Authentication, Scalable Architecture.

I. INTRODUCTION

In an increasingly interconnected world, real-time communication has become an indispensable tool for personal, educational, and professional engagements. The demand for seamless, efficient, and user-friendly live conferencing applications has grown exponentially, fueled by the global shift towards remote work, virtual collaboration, and online learning. This research paper explores the development, architecture, and impact of a real-time live conferencing web application designed to address the challenges of modern communication. Real-time live conferencing applications enable users to engage in video, audio, and text-based interactions over the internet. These platforms integrate advanced technologies such as WebRTC (Web Real-Time Communication), scalable cloud infrastructure, and adaptive streaming protocols to provide low-latency, high-quality communication experiences. The scope of these applications extends beyond simple video calls, encompassing features such as screen sharing, interactive whiteboards, live polling, and integration with third-party productivity tools.

The objective of this research is to investigate the design and implementation of a robust and scalable live conferencing application. The study examines critical aspects, including user experience (UX), security, bandwidth optimization, and multi-platform compatibility. Additionally, the research delves into the challenges faced during the development process, such as managing network latency, ensuring end-to-end encryption, and providing accessibility for diverse user demographics.

This paper aims to contribute to the field by providing insights into best practices, innovative features, and emerging trends in real-time communication. By leveraging a combination of theoretical frameworks and practical implementation strategies, the research seeks to address existing limitations and propose solutions for enhancing the efficiency and inclusivity of live conferencing applications in various domains.

II. RELATED HISTORY

The history of real-time live conferencing software is based on developments in networking technologies and communication protocols. From text-based systems in the early years to AI-powered video conferencing, every breakthrough has impacted people's virtual connectivity in a substantial way.

1) Early Text-Based Communication (1960s–1990s)

The communications started with networks such as ARPANET and Internet Relay Chat (IRC) through which the users were able to send instant text-based messages to each other. These initial developments paved the way for synchronous computer-mediated communication.

2) Development of Voice over IP (VoIP) (1990s)

The 1990s witnessed the emergence of Voice over IP (VoIP), which allowed the conveyance of voice signals via internet protocols. Software's like Skype, introduced in 2003, made this technology mainstream by providing scalable, inexpensive audio calls.

3) Emergence of Video Conferencing (2000s)

With increasing bandwidth on the internet and enhanced video compression, video conferencing products such as Webex and Polycom came into existence in the 2000s for business communications. They were later replaced by consumer platforms like Skype, bringing personal video calls.

4) Introduction of WebRTC (2011)

In 2011, Google launched Web Real-Time Communication (WebRTC)—an open-source initiative that allows for real-time peer-to-peer audio, video, and data exchange between web browsers without the use of plugins. WebRTC quickly emerged as the core technology for browser-based conferencing with low latency and ease of implementation.

5) Integrated Platforms Growth (2010s–2020s)

The decade has also seen the ubiquitous use of integrated platforms like Zoom, Microsoft Teams, and Google Meet that integrate video conferencing and collaboration capabilities. The global pandemic of COVID-19 accelerated the use of these platforms, driving how institutions work remotely.

6) Emerging Trends and Future Directions (now)

Contemporary conferencing apps are developing along with the incorporation of Artificial Intelligence (AI)to enable real-time transcription, virtual backdrops, and smart moderation. Augmented Reality (AR), Virtual Reality (VR), and 5G technologies are going to make virtual communication even more immersive and seamless,

III. OBJECTIVES

1) To design and implement a scalable architecture that supports real-time web conferencing with high availability.
2) To evaluate and integrate cutting-edge technologies that enable seamless, low-latency live communication.
3) To identify and mitigate challenges related to latency, scalability, and security to ensure a robust user experience.
4) To conduct comprehensive performance and usability testing to validate the effectiveness of the application.

IV. RELATED WORK

Numerous studies and existing platforms such as Zoom, Microsoft Teams, and Google Meet have been extensively analyzed in the domain of real-time web conferencing. These platforms primarily utilize WebRTC and complementary technologies to facilitate seamless and low-latency live communication. Prior research has explored key challenges including network latency optimization, video and audio compression techniques, data encryption for secure communication, and scalable infrastructure design. These works provide a strong foundation for this study by highlighting critical performance bottlenecks and security considerations, informing the architecture and technology choices in the current project.

V. REQUIREMENT ANALYSIS

A. Hardware Requirements

1) For Development:

A development machine with a minimum of 8GB RAM, a quad-core processor, and internet access.

A stable internet connection for video/audio testing and API requests.

2) For Deployment:
A server with a minimum of 16GB RAM, 100GB of storage, and a reliable internet connection to handle user traffic.
Optionally, a cloud hosting solution like AWS, Google Cloud, or Digital Ocean for better scalability and availability.

B. Software Requirements

1) Front-End
React.js for building the user interface.
HTML5, CSS3, and JavaScript (ES6+) for web development.

2) Back-End
Node.js for server-side processing.
Express.js for routing and API management.

3) Database
MongoDB for flexible, scalable data storage.

4) Real-Time Communication:
WebRTC for video/audio streaming.
Socket.io for real-time messaging and event handling.

5) Security
JWT for user authentication.
HTTPS for secure communication.

VI. METHODOLOGY

The development of the Real-Time Live Conferencing Web Application will follow the agile software development methodology. This approach will allow for iterative progress, frequent testing, and constant user feedback, ensuring that the final product meets user needs and expectations.

1) *Front-End Development (React.js)*: The front-end will be developed using React.js, focusing on creating a responsive, interactive user interface. React.js's component-based architecture will allow for modular development, making it easy to add features like meeting rooms, user login pages, video controls, and real-time chat interfaces. The platform will be fully responsive, ensuring compatibility with various devices such as desktops, laptops, tablets, and mobile phones.
2) *Back-End Development (Node.js and Express.js):* The back end will be managed using Node.js and Express.js. These tools will handle API requests, manage user sessions, and interact with the database. APIs will be built to handle essential tasks like user authentication, meeting scheduling, data storage, and real-time communication. Node.js's event-driven architecture will ensure the back end can handle many simultaneous users without performance degradation.
3) *Real-Time Communication (WebRTC & Socket.io):* WebRTC will serve as the backbone for real-time audio and video communication, enabling peer-to-peer streaming with minimal latency. Complementing this, Socket.io will handle real-time event-driven interactions, including live chat messaging, notifications, and user presence updates (e.g., when participants join or leave meetings), ensuring smooth and synchronous user experiences
4) *Database Management (MongoDB):* MongoDB will be used to store user credentials, meeting details, chat logs, and other relevant information. The database will be designed to scale dynamically, ensuring that the platform can handle increasing amounts of data as more users join.
5) *Security and Privacy:* Ensuring the security and privacy of user data is a key concern. The platform will use JWT-based authentication to verify users and control access to meetings. All data transfers will be encrypted using HTTPS to prevent unauthorized access. Sensitive information, like user credentials and chat logs, will be securely encrypted in the database to protect against data breaches.

VII. PROPOSED SYSTEM ARCHITECTURE DIAGRAM

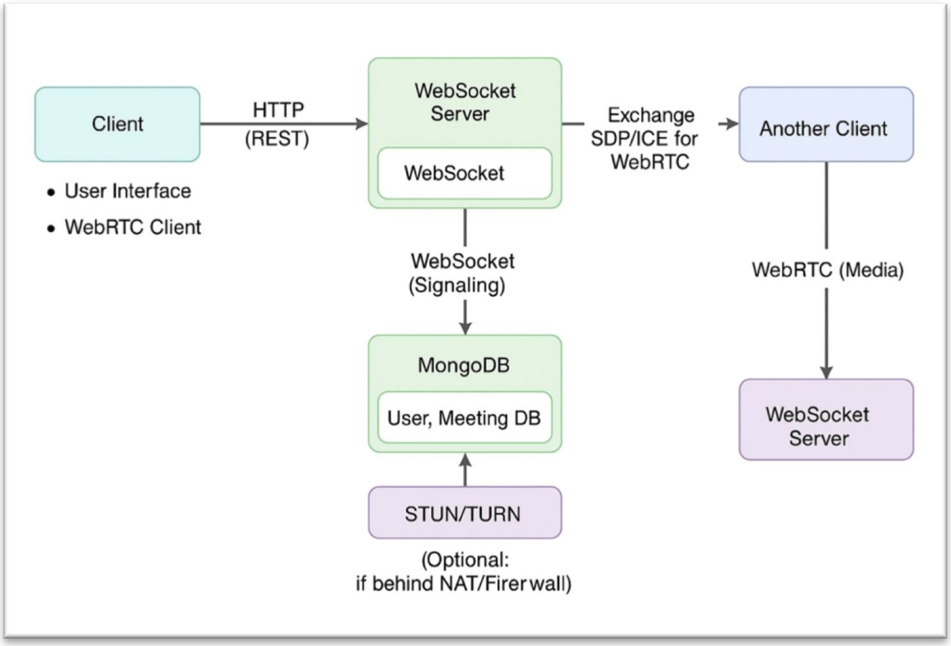

Fig. 1: Architecture diagram of the Eye Webinar application showing client-server interaction, WebRTC signaling, and data flow between components.

VIII. CLASS DIAGRAM (UML)

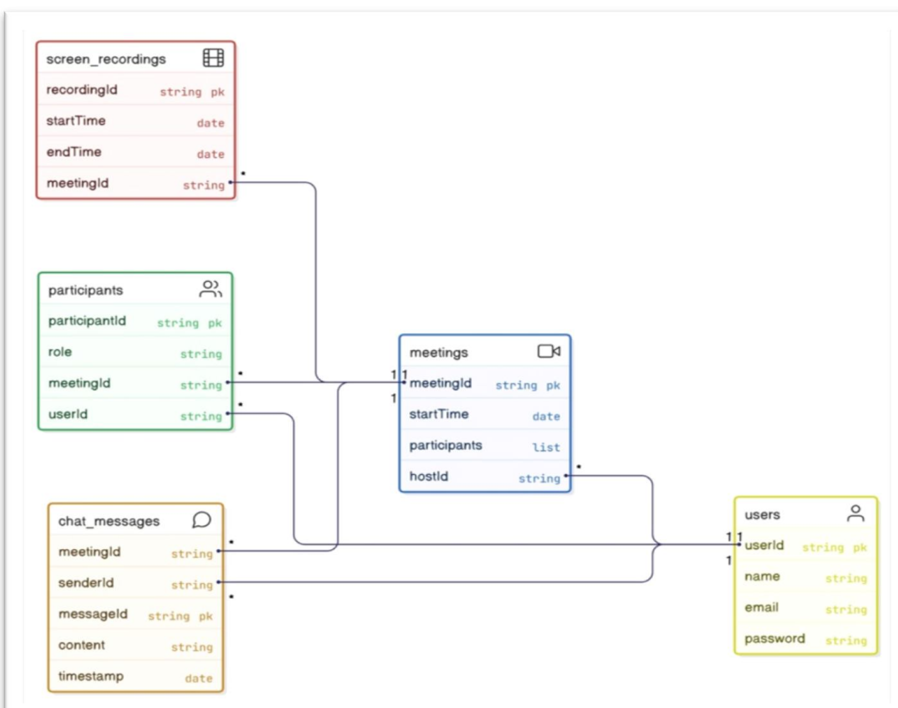

Fig. 2: UML class diagram illustrating the key classes and their relationships in the Eye Webinar system.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

IX. USER FLOW DIAGRAM / USE CASE DIAGRAM

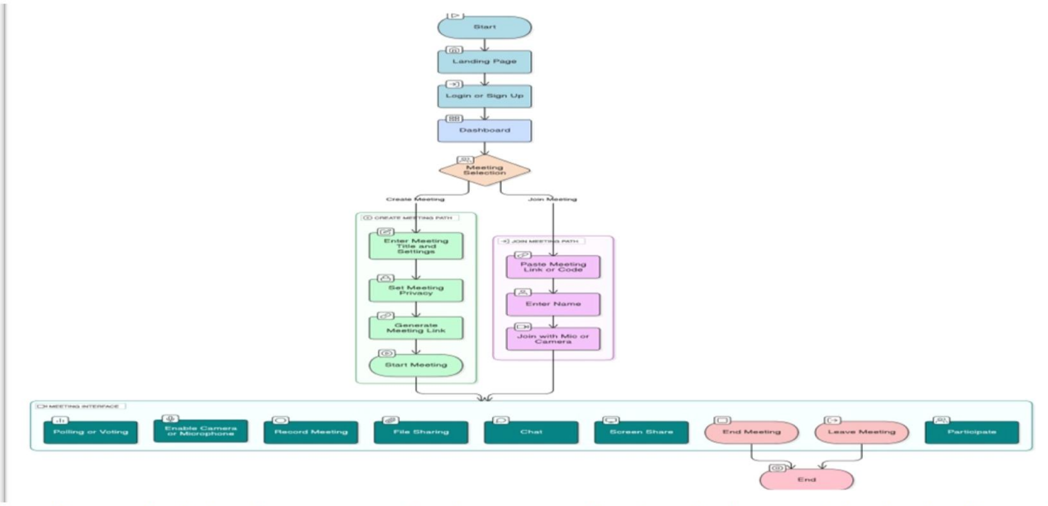

Fig. 3: User flow diagram depicting the sequence of actions a user takes from login to participating in a live webinar.

X. INDUSTRY/SOCIETY IMPACT

The Real-Time Live Conferencing Web Application has the potential to make a significant impact on both industry and society, especially as remote work, online education, and virtual collaboration become more prevalent.

1) *Educational Impact:* In the education sector, the platform can serve as a vital tool for virtual classrooms, enabling teachers and students to engage in real-time lessons, group discussions, and one-on-one interactions. The platform's ease of use ensures that even users with minimal technical knowledge can participate fully in online learning. This is particularly important for students in rural or underdeveloped areas who may not have access to traditional learning environments but can still receive quality education through virtual classes.
2) *Corporate and Professional Impact:* In the business world, the application provides a cost-effective solution for companies that need to conduct virtual meetings, webinars, and conferences. Unlike expensive subscription-based platforms, this application offers an affordable, customizable solution that small to mid-sized businesses can deploy to facilitate internal and external communication. The platform will also support team collaboration through real-time screen sharing, allowing teams to work on projects seamlessly, regardless of their physical locations.
3) *Societal Benefits:* Beyond education and business, the platform can also play a crucial role in maintaining personal connections by allowing families and friends to interact through virtual meetups, celebrations, and events. It can be particularly beneficial in situations where physical meetings are not possible due to geographical distances or public health concerns, like pandemics. By providing an easily accessible platform for remote communication, it can help combat social isolation and foster stronger community bonds.
4) *Contribution to Industry:*This project contributes to the technology ecosystem by demonstrating the power and viability of open-source, full-stack development tools. Built using the MERN stack and WebRTC, it showcases how secure, flexible, and scalable solutions can be developed without relying on costly proprietary technologies. The platform also emphasizes user data protection and system customizability, addressing key concerns in the current digital communication landscape, such as data privacy, user trust, and system ownership.

In conclusion, the Real-Time Live Conferencing Web Application will play a significant role in enhancing the accessibility, affordability, and quality of real-time communication, making it a valuable tool in various sectors, contributing to societal and professional advancement.

XI. CONCLUSION

Eye Webinar is a solution that solves problems brought about by the remote learning era by integrating a real-time web-based communication system. The system leverages modern web technologies such as WebRTC and MERN stack to provide audio/video transmission, messaging, and user interaction. Their development focuses on solving latency, scalability, data privacy, and accessibility issues which are very important in virtual learning setups.

The platform allows instructors and learners to participate live in classroom interactions, enhancing digital learning. Geographical and socio-economic barriers do not limit access to information. The Eye Webinar system is secure and highly responsive through the use of JWT authentication, HTTPS encryption, and a React backend, making it a reliable tool for educational institutions.

Also, inclusiveness drives the development of the application, which is geared towards ease of use for educators and learners. It accommodates a range of devices and internet connection speeds, broadening accessibility even further. This degree of adaptability and assurance supports educational institutions in sustaining lessons during routine times and in emergencies—like pandemics and enforced remote teaching.

As mentioned before, Eye Webinar is an easy-to-use educational platform that meets the specifications of modern remote teaching while supporting seamless future integrations, such as AI-assisted learning, AR/VR powered classrooms, and enhanced 5G functionalities. This transforms the framework of remote education from an isolated experience to a more unified, equitable, and dynamically innovative one, bridging technology and education.

XII. FUTURE SCOPE

Future research can focus on:

1) Add AI features like automated transcription, translation, and noise suppression for clearer communication.
2) Improve support for AR/VR to create immersive, interactive virtual meetings.
3) Optimize bandwidth and compression techniques for smooth performance in areas with poor internet.
4) Use predictive analytics to anticipate network issues and understand user behavior for better system responsiveness.
5) Integrate with third-party apps (calendars, CRMs, task managers) for smoother collaboration.
6) Implement stronger security measures, including end-to-end encryption and facial recognition.
7) Ensure cross-platform support with user-friendly interfaces for mobile, desktop, and smart devices.
8) Use blockchain for secure identity management and maintaining trustworthy meeting records.
9) Adopt microservices architecture to make the system scalable and easier to maintain.
10) Personalize user experience with machine learning that adapts to individual preferences.
11) Develop AI-powered virtual classrooms with automatic attendance, adaptive content, and engagement insights

XIII. ACKNOWLEDGEMENTS

We, Shaswat Agrahari, Aditi Pandey, Arpita Singh, and Subham Thakur, B.Tech students, would like to express our sincere gratitude to Dr. Sunil Kumar Verma, Head of the Department, Faculty of Computer Science and Engineering, Feroze Gandhi Institute of Engineering and Technology, Rae Bareli (U.P.), India, for his invaluable guidance and continuous support throughout the course of our research. His insightful suggestions and encouraging thoughts greatly contributed to the successful completion of this work.

XIV. APPENDICES [APPLICATION SCREENSHOTS]

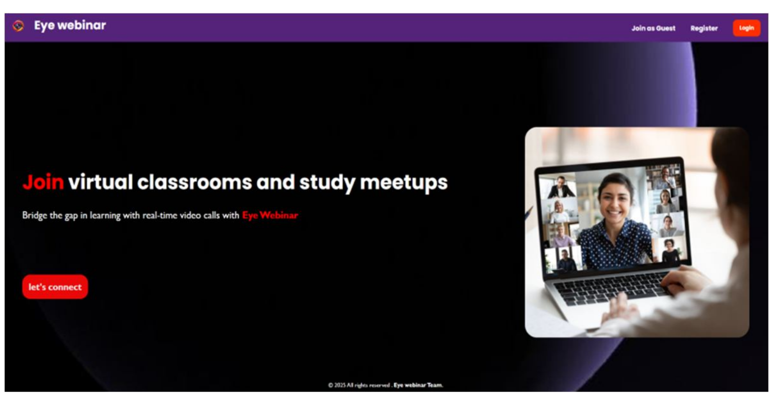

Fig. 4: Landing Page

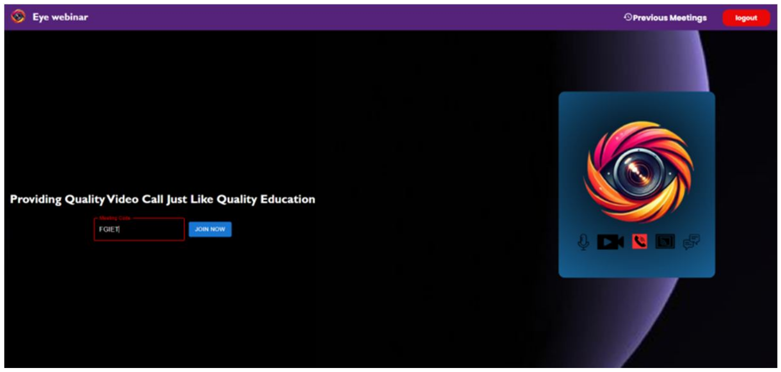

Fig. 4: Home Page

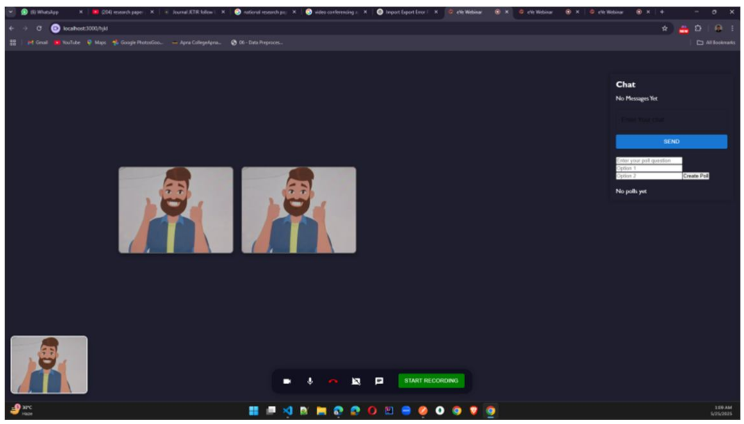

Fig. 4: Meeting Page

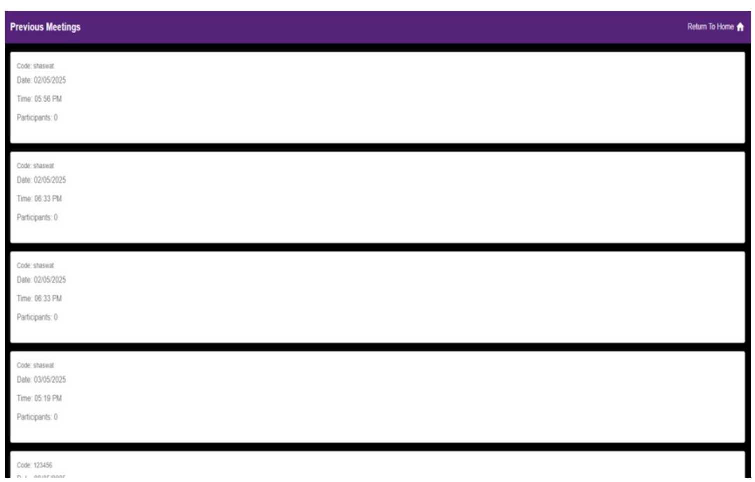

Fig. 4: Previous Meeting Page

REFERENCES

[1] Google. WebRTC: Real-Time Communication in Web Browsers. Available at: https://webrtc.org/ (Accessed January 2025)
[2] Zoom Video Communications. (2021). Technical White Paper. Available at: https://zoom.us/docs/doc/Zoom-Technical-White-Paper.pdf
[3] Socket.IO Documentation. Real-time communication in Node.js applications. Available at: https://socket.io/docs/v4/ (Accessed February 2025)
[4] Amazon Web Services (AWS). Building Scalable Real-Time Applications with AWS. Available at: https://aws.amazon.com/solutions/real-time-applications/ (Accessed May 2025)
[5] Kumar, S., & Verma, P. (2020). Real-Time Video Conferencing Application Using MERN Stack and WebRTC. International Journal of Computer Applications, 176(10), 15-22. DOI: 10.5120/ijca2020920446
[6] Patel, A., & Shah, N. (2021). Implementation of Video Conferencing System Using MERN Stack and Socket.IO. Proceedings of ICETET, pp. 342-348.

[7] Mozilla Developer Network (MDN). Using WebRTC with Node.js and React. Available at: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Using_WebRTC_with_Node.js_and_React (Accessed May 2025)
[8] Wojtowicz, P., & Stanek, W. (2019). Security Aspects of WebRTC. In: International Conference on Information and Communications Security (ICICS), pp. 123–134.
[9] AWS Documentation. Architecting for Real-Time Video Streaming on AWS. Available at: https://docs.aws.amazon.com/solutions/latest/video-streaming-on-aws/overview.html
[10] Ravindra, D., & Reddy, S. (2019). Design and Development of a Scalable Video Conferencing Web Application Using WebRTC and MERN Stack. International Journal of Computer Science and Information Technologies, 10(6), 255-260