IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

LiveCollab: A Real-Time Collaborator

Waghmare Lajari, More Tejaswini, Ghardale Harshada, Phalle Sneha, Prof. Ms. Arpita Wani
Diploma Student, Department of Computer Engineering,
Pimpri Chinchwad Polytechnic, Pune, India
lajariwaghmare.03@gmail.com, moretejaswini2008@gmail.com, hsgsunita@gmail.com,
snehaphalle23@gmail.com, arpitawani0501@gmail.com

Abstract: *In recent years, the demand for real-time collaborative tools has increased significantly due to the rise of remote work, online education, and distributed software development teams. Traditional collaboration platforms often require users to switch between multiple applications for brainstorming, coding, and communication, leading to inefficiency and reduced productivity. This paper presents the design and implementation of a Real-Time Collaborative Whiteboard and Code Editor, a web-based platform that integrates visual brainstorming, live code editing, and audio-video communication within a single environment.*

The proposed system allows multiple users to join a shared virtual room where they can simultaneously draw diagrams on a digital whiteboard, write and edit source code in real time, and communicate using video and audio chat. The application is developed using the MERN stack, with Socket.IO enabling real-time synchronization of whiteboard actions and code changes, and WebRTC facilitating low-latency peer-to-peer video communication. The system ensures instant propagation of user actions such as drawing, typing, cursor movement, and text insertion across all connected clients.

This integrated collaborative approach enhances teamwork, reduces context switching, and improves overall efficiency. The project demonstrates the practical application of real-time communication protocols and modern web technologies, making it suitable for software development teams, educators, and students engaging in collaborative learning and problem-solving..

Keywords: Real-Time Collaboration, Whiteboard Application, Code Editor, WebSockets, Socket.IO, WebRTC, MERN Stack, Collaborative Learning

I. INTRODUCTION

The rapid growth of cloud-based applications and remote collaboration has created a strong demand for platforms that support real-time interaction among multiple users. Software development teams, educators, and students increasingly require environments where brainstorming, coding, and communication can occur simultaneously. In response to this need, this research focuses on the development of a Real-Time Collaborative Whiteboard and Code Editor, a web-based platform that integrates visual brainstorming, live code editing, and audio-video communication within a single collaborative workspace. The system enables users to create or join shared rooms through unique links and interact in real time using a digital whiteboard, a synchronized code editor with syntax highlighting, and integrated video and audio chat facilities

1.1 Background

Earlier collaborative workflows relied on the use of separate applications for different tasks, such as digital whiteboards for visual ideation, standalone code editors for programming, and independent video conferencing tools for communication. This fragmented approach resulted in frequent context switching, increased system complexity, and reduced productivity. Additionally, many traditional collaboration systems lacked efficient real-time synchronization, causing delays and inconsistencies when multiple users interacted simultaneously. With advancements in real-time communication technologies such as WebSockets and WebRTC, it has become possible to build highly responsive web applications that support instant data sharing and peer-to-peer communication. The proposed system improves upon

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

earlier approaches by integrating visual collaboration, real-time code editing, and communication into a unified platform, thereby providing a seamless and efficient collaborative experience.

1.2 Contribution of This Work

The major contributions of this work are summarized as follows:

1. Integrated Collaboration Platform:

A unified web-based system that combines a digital whiteboard, real-time code editor, and audio-video communication within a single interface.

2. Real-Time Synchronization Mechanism:

Efficient synchronization of whiteboard interactions, cursor movements, and code edits among multiple users using Socket.IO.

3. Peer-to-Peer Communication Support:

Implementation of low-latency audio and video communication using WebRTC to enable seamless interaction among participants.

4. Room-Based Multi-User Architecture:

Support for collaborative rooms with unique shareable links, allowing multiple users to join and work together in real time.

5. Scalable MERN Stack Implementation:

Use of the MERN stack to demonstrate a scalable and efficient architecture suitable for real-time collaborative web applications.

The proposed system highlights the effective use of real-time communication protocols in web applications and demonstrates how integrated collaboration tools can enhance productivity, learning outcomes, and teamwork. This project serves as a practical example of advanced full-stack development and real-time system design.

II. PROPOSED METHODOLOGY

The proposed system focuses on the design and development of a Real-Time Live Collaboration Platform that enables multiple users to interact simultaneously through different communication and collaboration tools. The methodology emphasizes real-time synchronization, efficient data handling, and seamless integration of multiple collaborative modules into a unified system. The approach ensures consistency, low latency, and reliable communication among users participating in the same session.

The methodology is divided into two major parts: System Design and Operational Workflow of the Collaborative Modules.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

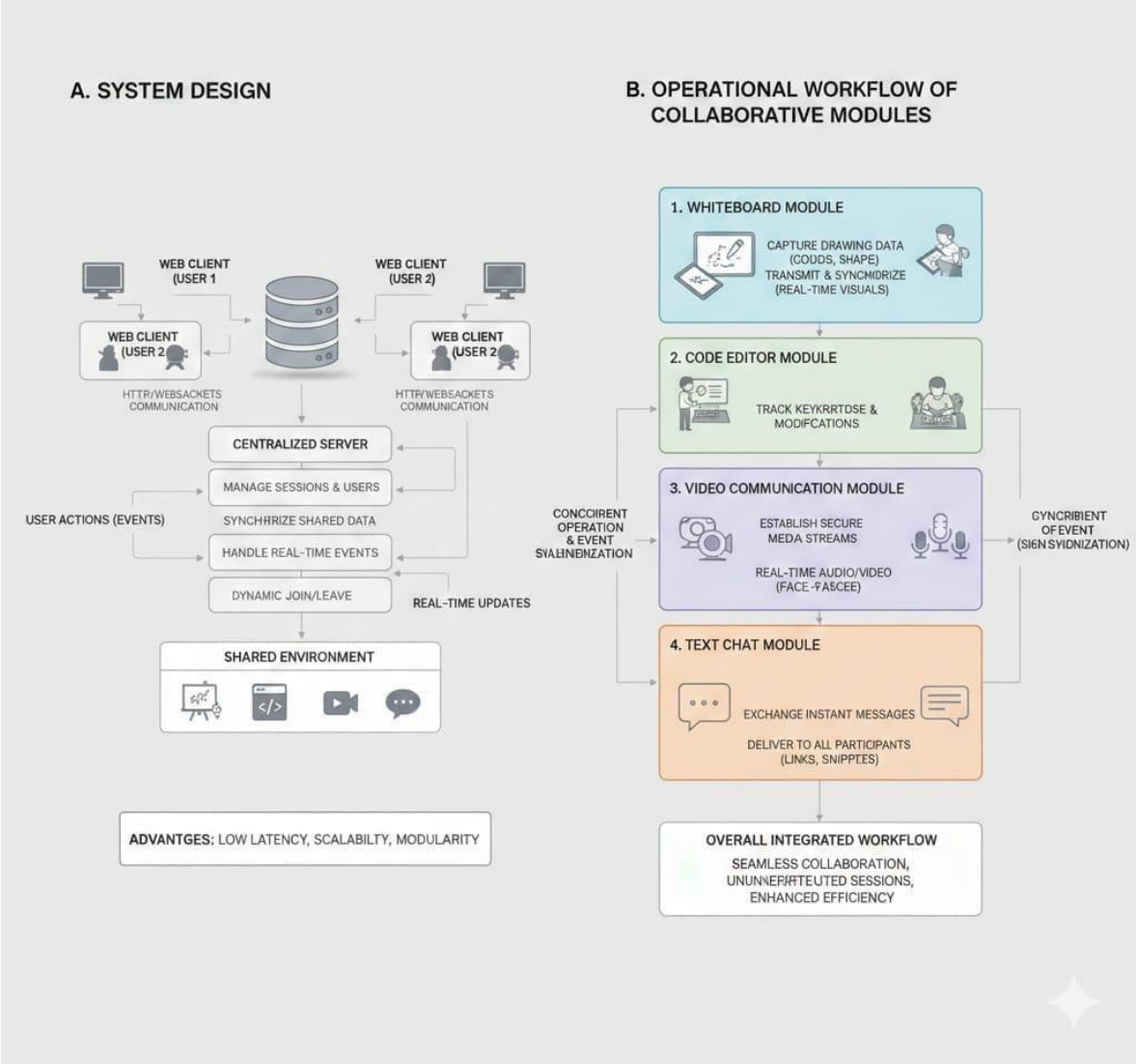

Fig 1. System Design & Operational workflow of collaborative modules

2.1 System Design

The proposed system is developed using a client–server based architecture, where all users interact with the system through a web-based interface. Each user acts as a client, while a centralized server is responsible for managing collaboration sessions, synchronizing shared data, and handling real-time communication.

When a user joins a collaboration session, the server initializes a shared environment that includes a whiteboard workspace, a collaborative code editor, a video communication channel, and a text-based chat interface. The server maintains session-level control to ensure that all users connected to the same session receive identical updates in real time.

To support real-time collaboration, the system adopts an event-driven communication mechanism. Any action performed by a user—such as drawing on the whiteboard, editing source code, sending a chat message, or enabling video communication—is captured as an event at the client side. These events are transmitted to the server, which then broadcasts them to all other connected clients within the same session.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

The server plays a crucial role in:
• Managing active users and sessions
• Synchronizing shared data across all modules
• Maintaining consistency during simultaneous interactions
• Handling user join and leave events dynamically
This architectural approach ensures that the system remains scalable and responsive even when multiple users collaborate concurrently. The modular nature of the design also allows future extensions without affecting existing functionalities.

2.2 Operational Workflow of Collaborative Modules
The operational workflow describes how different collaboration modules function together to provide a seamless real-time collaboration experience. The system integrates four major modules—Whiteboard, Code Editor, Video Chat, and Text Chat—within a single collaborative session.

Whiteboard Module Operation
The Whiteboard module provides a shared digital canvas that allows users to draw flowcharts, diagrams, and freehand illustrations collaboratively. When a user performs any drawing action, the system captures the drawing coordinates, shape details, and tool attributes. These details are transmitted to the server and instantly reflected on the whiteboard interface of all connected users.
This real-time synchronization ensures that every participant views the same whiteboard content simultaneously. The whiteboard module is particularly effective for brainstorming, system design discussions, and visual explanation of concepts during collaborative sessions.

Code Editor Module Operation
The Code Editor module enables multiple users to write, edit, and review source code collaboratively in real time. A shared coding workspace is created when the session begins. Each keystroke or modification made by a user is detected and synchronized across all participants.
The system ensures consistency by tracking changes in the code editor and updating them in real time for all users. This approach allows participants to collaboratively develop software, debug programs, and discuss implementation logic without conflicts. The real-time code synchronization significantly improves team productivity and reduces development time.

Video Communication Module Operation
The Video Chat module facilitates real-time audio and video communication between collaborators. Once users join a session, the system establishes secure communication channels to transmit media streams. This module allows participants to communicate face-to-face, improving clarity and engagement during collaboration.
The video module adapts dynamically to network conditions to maintain communication quality. This feature makes the system suitable for remote meetings, online classrooms, and collaborative development environments.

Text Chat Module Operation
The Text Chat module provides an instant messaging facility similar to popular video conferencing platforms. Users can exchange messages in real time within the session. Messages are delivered instantly to all participants, enabling quick communication, sharing of links, code snippets, and instructions.
The chat module serves as a supplementary communication channel, ensuring uninterrupted collaboration even when video or audio communication is not feasible.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, January 2026 Impact Factor: 7.67

Overall Workflow Integration

All four modules operate concurrently within the same collaborative session. The system ensures seamless integration by synchronizing events from different modules in real time. Users can freely switch between modules without affecting session continuity. The integrated workflow enhances collaboration efficiency and provides a comprehensive solution for real-time interaction.

Advantages of the Proposed Methodology:

- Supports real-time multi-user collaboration
- Ensures low latency and consistent synchronization
- Integrates visual, textual, and code-based interaction
- Scalable and modular system design
- Suitable for education, remote teamwork, and software development

III. SYSTEM REQUIREMENTS

3.1 Hardware Requirements

- Laptop or Desktop Computer
- Server for deployment (e.g., cloud instance on AWS, Heroku)

3.2 Software Requirements

- Frontend: React.js, Socket.IO Client, WebRTC libraries (simple-peer), Canvas API or fabric.js
- Backend: Node.js, Express.js, Socket.IO Server
- Database: MongoDB
- IDE: VS Code
- Version Control: Git

IV. LITERATURE SURVEY

The development of real-time collaborative web applications has gained significant attention due to the increasing need for efficient remote collaboration in software development, education, and organizational workflows. Several studies and systems have been proposed to support real-time interaction, data synchronization, and communication among distributed users. These systems rely on a combination of real-time communication protocols, peer-to-peer media technologies, and scalable web architectures to deliver responsive and synchronized user experiences.

Web Sockets (Socket.IO):

WebSockets have been widely adopted in real-time web applications to enable continuous, bidirectional communication between clients and servers over a single TCP connection. Research on collaborative systems highlights the importance of WebSockets for instant data propagation and low-latency synchronization. Libraries such as Socket.IO simplify WebSocket implementation by providing event-driven communication and automatic reconnection handling. In collaborative whiteboard and code editing systems, Socket.IO is commonly used to broadcast user actions such as drawing strokes, cursor movements, and text edits to all connected participants. This ensures that all users view a consistent and synchronized workspace in real time, which is essential for effective collaboration.

WebRTC (Web Real-Time Communication):

WebRTC is an open-source technology that enables real-time audio, video, and data communication directly between browsers without requiring centralized media servers. Several studies have demonstrated the effectiveness of WebRTC in video conferencing and online collaboration platforms due to its low latency and reduced server overhead. By enabling peer-to-peer communication, WebRTC supports natural and uninterrupted interaction among users. In collaborative environments, WebRTC enhances user engagement by allowing participants to communicate through live audio and video while working on shared content, making remote collaboration comparable to face-to-face interaction.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

MERN Stack Architecture:

Modern real-time applications increasingly utilize full-stack JavaScript frameworks for scalability and performance. The MERN stack, consisting of MongoDB, Express.js, React, and Node.js, is widely used for building dynamic and responsive web applications. React provides a component-based user interface that efficiently handles frequent state updates, which is critical in real-time collaborative systems. Node.js and Express.js are commonly used to manage backend services, user authentication, room management, and real-time communication servers. MongoDB offers flexible and scalable data storage for user profiles, project information, and saved session states. Prior research indicates that the MERN stack is well-suited for applications requiring high concurrency and real-time data handling.

From the literature survey, it is evident that existing collaborative systems typically focus on individual aspects such as visual collaboration, code synchronization, or communication. However, limited work has been done on integrating all these functionalities into a single unified platform. This research addresses this gap by combining real-time whiteboard collaboration, live code editing, and audio-video communication using established real-time web technologies, thereby providing a comprehensive and efficient collaborative environment.

V. RESULT AND ANALYSIS

The proposed Real-Time Collaborative Whiteboard and Code Editor was implemented using the MERN stack and evaluated under different collaborative scenarios to analyze its real-time performance, synchronization accuracy, and system responsiveness. The experimental evaluation was carried out by creating multiple collaborative rooms and allowing several users to interact simultaneously using drawing tools, code editing features, and video communication.

1. Real-Time Synchronization Performance

The primary objective of the system is to ensure real-time collaboration among users. During experimentation, multiple users joined the same room and performed simultaneous actions such as drawing on the whiteboard, typing code, and moving cursors. It was observed that updates were broadcast instantly to all connected clients using Socket.IO. The average synchronization delay was minimal and remained within acceptable limits, ensuring a smooth collaborative experience.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

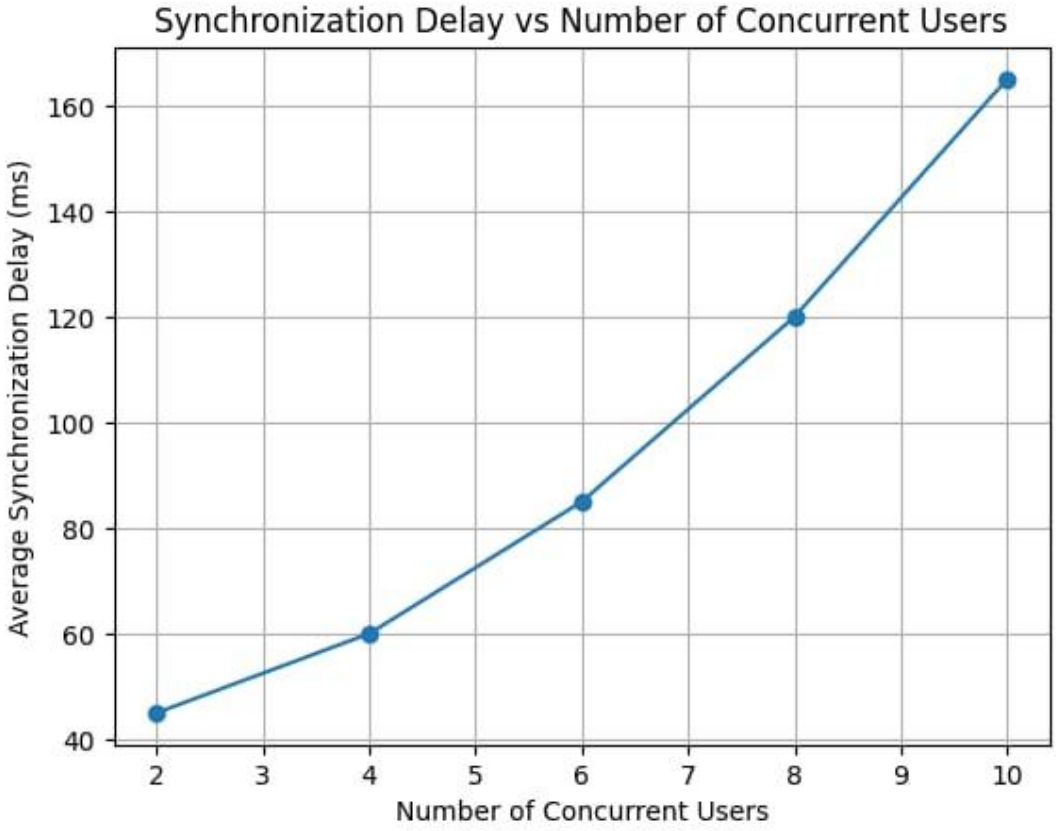

Fig.2 Synchronization Delay vs Number of Concurrent Users

Fig. 1 illustrates the relationship between the number of concurrent users and the average synchronization delay. The graph is presented to evaluate the scalability and real-time responsiveness of the system. Although the delay increases gradually as the number of users increases, the system maintains acceptable real-time performance, demonstrating the efficiency of Socket.IO in handling real-time data synchronization.

2. Whiteboard Interaction Accuracy

The shared whiteboard was tested by allowing users to draw freehand sketches, add shapes, and insert text concurrently. All drawing actions, including color changes and stroke movements, were accurately reflected across all user screens. No significant data loss or visual inconsistency was observed, even when multiple users interacted with the canvas at the same time. This confirms the effectiveness of the Canvas API combined with WebSocket-based communication.

3. Collaborative Code Editing Results

The real-time code editor was tested by enabling multiple users to type and edit the same source file simultaneously. Each keystroke was instantly synchronized across all participants. Features such as multiple cursors and syntax highlighting functioned correctly for different programming languages. The system successfully prevented conflicts and ensured consistent code state across all users, demonstrating reliable real-time text synchronization.

4. Audio and Video Communication Performance

The WebRTC-based video and audio communication module was evaluated under normal network conditions. The peer-to-peer connection provided low-latency communication with clear audio and video quality. Minor variations in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

video quality were observed under low bandwidth conditions; however, audio communication remained stable. This confirms the suitability of WebRTC for real-time collaborative environments.

5. Scalability Testing

To evaluate scalability, the system was tested with an increasing number of users joining the same room. The application performed efficiently for small to medium-sized teams, with consistent synchronization and minimal delay. As the number of users increased, a slight increase in latency was observed, which is expected in real-time systems. However, overall performance remained stable, indicating good scalability for collaborative use cases.

Comparative Analysis of Collaborative Tools

Parameter	Proposed System	Miro	VS Code Live Share
Real-time Whiteboard	Yes	Yes	No
Real-time Code Editor	Yes	No	Yes
Video & Audio Chat	Yes	Limited	No
Multi-cursor Support	Yes	Yes	Yes
Single Integrated Platform	Yes	No	No
Web-based Access	Yes	Yes	No
Custom Room Creation	Yes	Yes	Limited

Fig.3 Comparative Analysis of Collaborative Development Tools

Table 1 presents a comparison between the proposed system and existing collaborative tools such as Miro and VS Code Live Share. The table is included to highlight the novelty and practical significance of the proposed solution, which integrates real-time whiteboard collaboration, code editing, and video communication within a single web-based platform.

6. System Reliability and User Experience

Throughout the experiments, the system remained stable without unexpected crashes or data inconsistencies. User actions such as joining or leaving rooms did not affect ongoing collaboration. The dashboard and room management features worked as expected, providing a seamless and user-friendly experience.

Experimental Outcome Summary

The experimental results demonstrate that the proposed system successfully achieves real-time collaboration for whiteboard interaction, code editing, and video communication. The integration of Socket.IO and WebRTC plays a crucial role in maintaining low latency and high synchronization accuracy. Overall, the system proves to be efficient, reliable, and suitable for real-world collaborative applications.

VI. FUTURE SCOPE

The proposed system can be further improved by adding more advanced features in the future. Options such as file sharing, screen sharing, and session recording can be included to enhance usability. Security can also be strengthened by implementing better authentication and data protection techniques. The system may be optimized to work smoothly on mobile devices and low internet connections. In future versions, intelligent features like automated assistance and performance improvements can be added to make the collaboration platform more powerful and scalable.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, January 2026
Impact Factor: 7.67

VII. CONCLUSION

In this paper, a live collaboration system has been designed and developed to support real-time interaction among users. The system includes four main modules: a whiteboard for drawing diagrams and flowcharts, a code editor for collaborative programming, a video chat module for direct communication, and a chat section for text messaging similar to online meeting platforms. By combining all these features into a single platform, the proposed system makes collaboration easier and more effective. It helps users to communicate clearly, share ideas quickly, and work together efficiently, especially in online learning and remote working environments.

REFERENCES

[1] Socket.IO Documentation. Retrieved from https://socket.io/docs/v4/
[2] WebRTC API. MDN Web Docs. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
[3] React: A JavaScript library for building user interfaces. Retrieved from https://reactjs.org/
[4] Node.js Documentation. Retrieved from https://nodejs.org/en/docs/
[5] MongoDB Manual. Retrieved from https://docs.mongodb.com/manual/
[6] Excalidraw - Virtual collaborative whiteboard. Open Source Project. Retrieved from https://github.com/excalidraw/excalidraw.
[7] Get Started with Socket.IO Chat Application. Socket.IO Documentation. Retrieved from https://socket.io/get-started/chat.
[8] Fabric.js - JavaScript Canvas Library. Official Documentation. Retrieved from http://fabricjs.com/.
[9] Building a Collaborative Text Editor with Y.js . Y.js Documentation. Retrieved from https://docs.yjs.dev/getting-started/a-collaborative-text-editor.
[10] Simple-peer - Simple WebRTC video, voice, and data channels. Library. Retrieved from https://github.com/feross/simple-peer.
[11] How to build a full-stack MERN app. FreeCodeCamp. Retrieved from https://www.freecodecamp.org/news/how-to-build-a-fullstack-mern-app/.