International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real-Time Collaboration Platform for Remote Work: Chat, Video Conferencing, Document Editing

Somesh Alone[1], Lakhan Jadhav[2], Kaustubh Deshpande[3], Pratik Papanwar[4], Dipti Aghor[5]

[1,2,3,4]B.Tech Student, Computer Science & Engg. Department, MGM's College of Engineering, Nanded.
[5]Guide, Asst. Prof., Computer Science & Engg. Department, MGM's College of Engineering, Nanded.

Abstract

In the era of remote work, effective collaboration tools are essential for ensuring productivity and seamless communication among geographically dispersed teams. Real-time collaboration is significantly popular during the COVID-19 pandemic period. Various communication activities like meetings, seminars, video conferencing, and instant messaging were carried out during the COVID-19 pandemic period. In this research paper, we will design and implement real-time collaboration tools by merging chat, video conferencing, and document editing tools into a single platform. The frontend can be created in EJS, JavaScript, and CSS so that it can be dynamic and up to date with node.js and MongoDB, you can build a powerful backend to handle all aspects of data creation, management, and server-side logic. Real-time video and chat functionality are implemented by using the Agora API, while WebSocket-based synchronizations ensure consistent and concurrent document editing. Our tool addresses common challenges remote teams face, such as communication delays and lack of integrated features, by enabling real-time interactions with minimal latency. This research highlights the potential of custom-built tools to enhance productivity in remote environments and for the future development of scalable and intelligent collaboration platforms.

Keywords: Real-Time Communication, Remote Collaboration, Video-conferencing, Chat, Document editing.

1. Introduction

Remote work is now common practice, making online tools essential for teams to work together effectively. In the office, people could easily communicate and collaborate face-to-face. However remote teams face challenges like delays and lack of centralized tools. Currently Zoom, Google Meet, Google Docs, and Slack are the most popular tools. However, using multiple tools can be confusing and expansive. There is a need for an all-in-one solution that provides chat, video conferencing, and document editing in a single platform. This research focuses on creating such a platform using technologies like EJS, JavaScript, CSS, Node.js, and MongoDB.

Remote teams often struggle with using multiple tools for chat, video conferencing, and document editing, which makes collaborations difficult and time-consuming. This tool can be expansive, especially for small

teams, and integrating them takes extra effort. Real-time collaboration features in current tools often face delays and synchronization issues causing confusion and reducing productivity. Additionally, many tools are not compatible or scalable with different devices, limiting their usability for diverse teams. To solve the problem, there need for an all-in-one platform that combines these features, supports real-time updates, is easy to use, and works well for teams of any device.

Objectives:

1. Design and develop a user-friendly application that integrates chat, video conferencing, and document editing into one platform.
2. Enabling real-time collaboration by implementing synchronized document editing, chat, and video features with minimal delays.
3. Ensure cross-platform combability so the application can support all devices.

2. Literature Review

2.1 Overview of existing collaboration tools

The different categories of collaboration tools used by remote teams lead to improved productivity and communications. Zoom is a widely used tool that offers video conferencing as well as features like screen sharing and breakout rooms, but it does not have built-in document editing or enhanced chat capabilities. With high substitution cost and the versatile interfacing with Microsoft Office tools, Microsoft Teams offers an even more comprehensive solution that also integrates chat video calls, document editing, or document sharing tools (new users struggle to adapt to the complex interface) Slack is an instant-messaging-and-team-communication-focused platform built for sharing and third-party integrations, though it doesn't do native video conferencing or document editing. Google Workspace combines tools like Google like Google Docs, Meet, and Chat, but its features are spread across multiple applications, leading to fragmented workflows.

2.2. Challenges with current solutions

While current collaboration tools like Zoom, Slack, Microsoft Teams, and Google Workspace are effective in specific areas, they come with serval challenges. One of the major issues is the lack of integrations, as most tools focus on single features like video conferencing, chat, or document editing. Moreover, real-time collaboration features often face synchronization delays, especially during document editing, which can cause confusion and disrupt productivity. Many tools are expensive, making them less accessible for small teams or startups.

2.3 Research gaps

Despite the availability of various collaboration tools, there is a significant gap in providing unified platforms that integrate all essential features such as chat, video conferencing, and real-time document editing. Existing solutions like Zoom, Slack, and Google Workspace focus on focus on individuals' functionalities, requiring teams' multiple tools to meet their needs. Additionally, many current platforms experience delays in synchronization during real-time collaboration and often struggle with scalability and cross-platform compatibility.

3. Methodology

The application structure utilized a client-server architecture, which ensures efficient communications between the user interface and backend services:

- **Frontend Development:** The front end of the application is created with EJS, JavaScript, and CSS to

offer a very basic and convenient user interface. That encompasses chatting windows for direct messages, a video conferencing interface with screen-sharing and similar features, and a collaborative document editor that updates in real-time It is a responsive design, so it looks good on desktops, tablets, and smartphones. The focus is on creating a smooth and user-friendly interface, allowing users to easily navigate and interact with platform features.

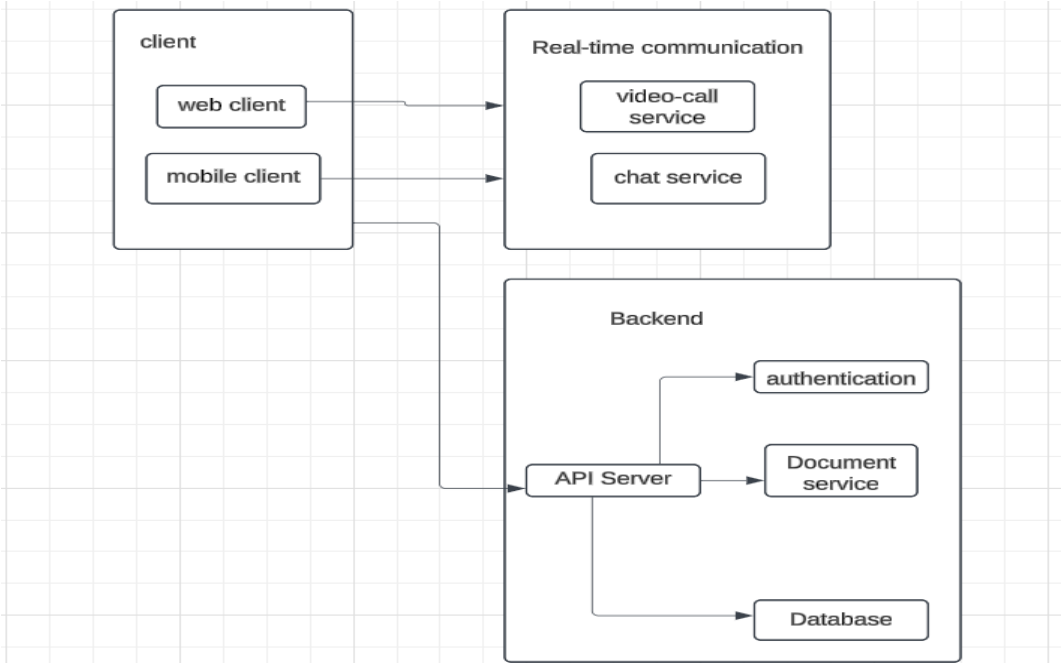

Fig1: System Architecture

- **Backend Development:** The backend is created in Node.js and will handle all the core processes of the application. This service handles user authentications, data storage such as chat history and document version in MongoDB, and updates in real-time with WebSocket technology. Agora APIs for video and chat services integrate with low latency. Built to be secure, scalable, and efficient, allowing multiple users to work together in real-time without latency. It can also mean that it can smooth communications between the front end and the server which makes real-time collaborations possible.

3.1. Key Features:

The application incorporates several critical features designed to enhance user experience and engagement:

- User authentication: Provide login or register to our application so that only authenticated users can use the application or Platform.
- User-Friendly Interface: Design user friendly interface any user can easily use these applications.
- Real-time chat: Our application provides real-time chat functionality that users can chat in real- time.
- Video conferencing: With screen-sharing capabilities, it can offer high-quality video and audio calls.
- Real-time document editing: It can provide real-time document editing features with live updates.
- Supporting all devices: The application can be supported for all devices like PCs, laptops, Smartphones, etc.

3.2. Developments tools and Technologies:

The following tools and technologies are used to create or develop the real-time collaboration tool:

3.2.1. Frontend development:

- EJS (Embedded JavaScript): EJS is a popular template engine for node.js and web development. It allows you to generate dynamic HTML content by embedding JavaScript code with HTML templates.
- JavaScript: Provide interactive and real-time functionality for the website.
- CSS: It can be used to design a responsive or user-friendly website.

3.2.2. Backend development:

- Node.js: It is a run time environment allowing to execution of JavaScript code on the server side, outside the web browser.
- Express.js: It is the node.js web application framework that can be used for developing web or mobile applications and managing routes efficiently.
- MongoDB: It is a database that is used to store user login information and document data.

3.2.3. Real-time communication:

- WebSocket: It is a communication protocol for use in client-server communication. It can be user real-time chat, document editing or other live updates scan show the user.
- Agora API: It can be used for implementing high-quality video and audio functionality for video conferencing and chatting.

  Tiny cloud API: The tiny cloud API can be used for document editing features like bold, italic, etc. It can be easy to use the formatting of the text in a document in real-time.

3.2.4. TCP/IP protocols:

TCP/IP between frontend and backend ensures. Everyone should be able to log in, transfer data, query API, and so on. It enables real-time communication for document editing, video calls, and chats with protocols like WebSocket and Agora SDK. It also ensures error handling and proper routing of packets of data for smooth user interactions.

4. Working Flow

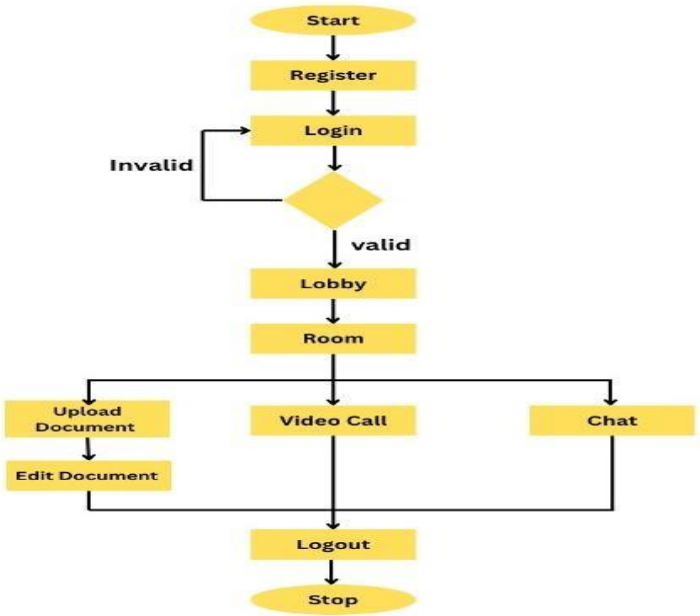

Fig2: Workflow

The real-time collaboration tool was successfully developed with features like chatting, document editing, and video conferencing. This application allows us to have multiple users work together in real-time. Users are able to log in and register securely, and access the application without any issues. Video calls and chat worked smoothly, allowing real-time communication without delays. Document uploading and editing are smoothly working and the users can upload documents and edit the documents in real time and it can show the updates visible to all participants.

4.1. User authentication:

Objective: The user be securely logged in and registered to the system.

Process: New users can be registered by providing their details, which are secularly stored in the database. Only authorized users can access or log in to the platform. If the credentials are valid then access is granted otherwise show an error message.

The system can store user data in a secure backend (e.g., MongoDB or MongoDB Atlas cloud).

4.2. lobby

Objective: It provides an interface for creating or joining the room.

Process: After secure login, the user is redirected to the lobby page. Here, the user can be either in a new meeting room or join the meeting by entering the room ID.

4.3. Room

Objective: Inside the meeting room it can be provided access to users' multiple features like chat, document editing, and video conferencing.

Process: After the user can start or join the meeting user is redirected to the room page. Here they provide multiple access where video conferencing, chat, and document editing feature. In video conferencing with smooth features like mute, unmute, and screen sharing. The chat feature allows the user can send messages in real-time.

4.4. Upload and Edit Document:

Objective: The user can upload documents and edit the document in real-time.

Process: After the user can start or join the meetings user is redirected to the room page. Here they provide the document editing or uploading options. The user clicks on the document. It goes to the upload document page where the user can upload the file. It can upload any type of file, such as docs, text, etc.

After successfully uploading the file, the user can click on the file user can edit the document. It can show the changes in the files to both users in real-time. It also provides document editing features like bold, italic,etc.

5. Results

The real-time collaboration tool was successfully developed with features like chatting, document editing, and video conferencing. This application allows us to have multiple users work together in real-time. Users are able to log in and register securely, and access the application without any issues. Video calls and chat worked smoothly, allowing real-time communication without delays. Document uploading and editing are smoothly working and the users can upload documents and edit the documents in real time and it can show the updates visible to all participants.

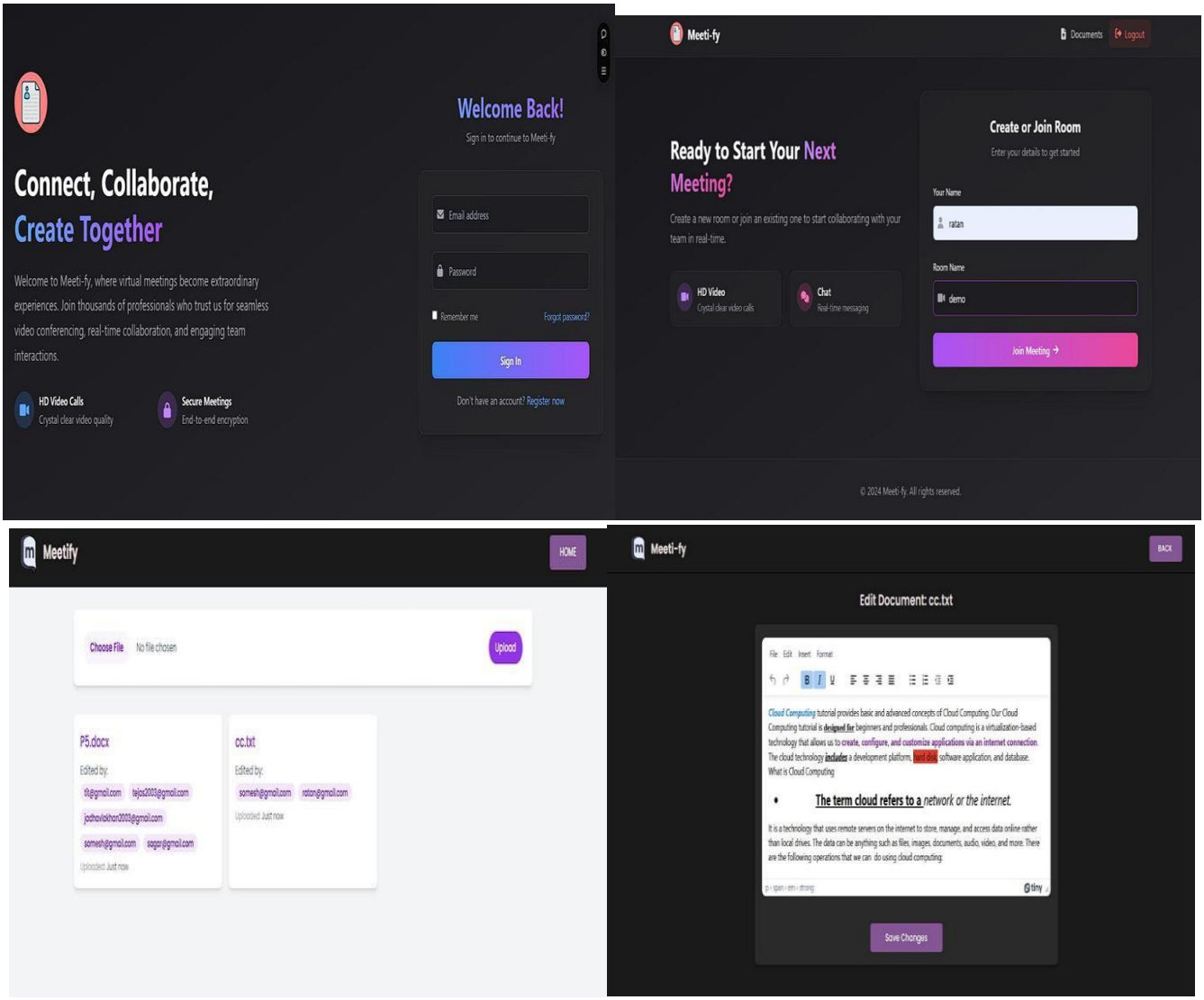

Fig3 : Result and Outcome

Conclusion

The developed real-time collaboration tool successfully addresses the key challenges faced by remote teams by integrating chat, video conferencing, and document editing into a single, unified platform. This all-in-one solution eliminates the need for multiple tools, enhancing workflow efficiency and reducing confusion. The use of robust technologies such as Node.js, MongoDB, and WebSocket ensures secure, scalable, and low-latency performance, while the Agora API provides high-quality video and chat functionality. Real-time synchronization across features ensures seamless interactions, making the tool highly effective for geographically dispersed teams. The application's responsive design and cross-platform compatibility further enhance its accessibility, ensuring usability across various devices and environments.

This research highlights the potential for custom-built tools to overcome the limitations of existing platforms, particularly in real-time collaboration and feature integration. By combining essential functionalities with minimal delays and an intuitive user interface, the developed application provides a scalable solution that can evolve with the future needs of remote work environments. The results demonstrate that leveraging modern web technologies can significantly enhance productivity,

communication, and collaboration among remote teams, paving the way for further advancements in integrated digital workspace solutions.

References

1. **Microsoft Teams: Integrated Collaboration Features** Microsoft Corporation. (n.d.). *Microsoft Teams: Collaboration Tools*. Retrieved from https://www.microsoft.com/en-us/microsoft-teams/collaboration
2. **Introducing Real-Time Collaboration Systems: Development of a Conceptual Scheme and Research Directions**
   Nunamaker, J. F., Reinig, B. A., & Briggs, R. O. (2009). *Introducing Real-Time Collaboration Systems*. Retrieved from https://www.researchgate.net/publication/237821066
3. **E-Collaboration: Concepts, Methodologies, Tools, and Applications** Kock, N. (Ed.). (2009). *E-Collaboration: Concepts, Methodologies, Tools, and Applications*. IGI Global.
4. **Slack: Enhancing Remote Communication** Slack Technologies. (n.d.). *Top Collaboration Tools for Remote Work*. Retrieved from https://slack.com/blog/collaboration/top-collaboration-tools-remote-work
5. **Real-Time Document Collaboration—System Architecture and Design** Zhang, Y., Wang, X., & Sun, J. (2022). *Real-Time Document Collaboration*. MDPI. Retrieved from https://www.mdpi.com/2076-3417/14/18/8356
6. **Understanding Real-Time Collaborative Programming** Chen, X., Han, J., & Smith, A. (2023). *A Study of Developers' Practices and Challenges*. ACM Digital Library. Retrieved from https://dl.acm.org/doi/10.1145/3643672
7. **Agora API: Low Latency Video and Chat** Agora. (n.d.). *Agora API Documentation*. Retrieved from https://www.agora.io/en/developer- resources/