SyncSpace Live Room: A Fully Accessible Real-Time Collaborative Workspace for Partially Disable People

Prof. Abhijeet More
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
abhijeetmore@mes.ac.in

Prof. Tejashree Patil
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
tejashreepatil@mes.ac.in

Snehal Gat
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
snehalag24hmca@student.mes.ac.in

Apoorva Ghorpade
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
apoorvasg24hmca@student.mes.ac.in

Tanmay Ghorpade
dept. of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharashtra, India
tanmaydg24hmca@student.mes.ac.in

Abstract- SyncSpace Live Room is a real-time, collaborative workspace that is very easy to use. It was made to help people with visual, hearing, and physical disabilities who want to use digital collaboration platforms. Sadly, a lot of popular tools only have a few accessibility features, which means that only some users can do things like coding, whiteboarding, taking notes, and talking to each other in real time. To solve the problems above, SyncSpace made accessibility features a part of its main architecture. It didn't even think about adding them as secondary enhancements. The MERN stack (MongoDB, Express.js, React.js, and Node.js) is used to build the platform. Socket.io lets users talk to each other in real time, with low latency, and sync their actions. There is a special Accessibility Module that helps with making instructions accessible to screen readers via ARIA labeling, using a microphone to talk to the system, hearing the system talking through the use of text, to, speech, moving around only by keyboard, switching up colors to make it easier to see, getting visually as well as vocally notified and much more, that is why SyncSpace is a fully compliant to WCAG 2. 1 guidelines. The system's architecture is made up of five main parts: the Authentication, Collaboration, Accessibility, Application, and Database modules. These modules provide secure access control, high performance scaling, friendly user interaction, and data handling, respectively. Tools for usability testing and accessibility validation show that adding assistive technologies to the system framework makes it easier for people to participate, be productive, and be happy with the system as a whole. Also, the design that is based on accessibility helps not only users with disabilities but also makes it easier for everyone to use by making navigation easier and lowering cognitive load. SyncSpace Live Room is a safe, scalable, and inclusive place for people to work together in schools, businesses, and from home. It encourages fair digital participation and teamwork without barriers.

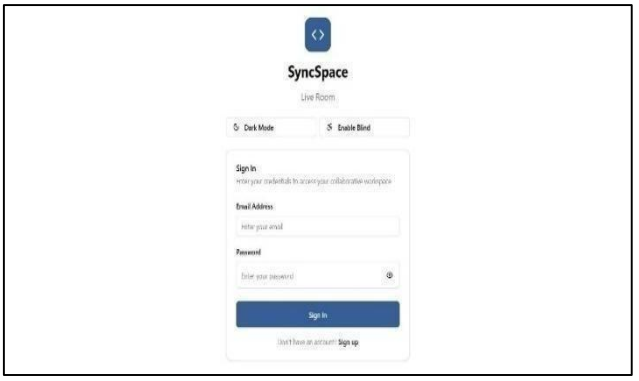
Keywords: Accessibility, Real-Time Collaboration, MERN Stack, Inclusive Design, WCAG Compliance

I. Introduction

Digital collaboration is now a key part of modern education, business, and online communities. This is especially true now that more and more people are working and learning from home. Real-time collaborative platforms like document editors, coding environments, and communication tools let people work together and share content at the same time from different places. Recent studies underscore the increasing significance of effective real-time synchronization methods and scalable system architectures to facilitate such collaboration [3], [14].

Also, changes in WebSocket communication and cloud-based distributed computing have made collaboration platforms feel much faster and more reliable [7]. However,the problem of accessibility is still prominent in many real, time collaboration systems even if we look at them from a technology standpoint. Research shows that most dynamic web applications cannot fully meet the requirements of accessibility standards like WCAG and ARIA, thus they are less friendly to visually impaired users who depend on screen readers [13]. In a nutshell, AI, driven accessibility improvement techniques have been suggested to recognize and fix web accessibility problems [4], but these are mostly used as extensions rather than becoming part of the system design. On the other hand, even though speech, to, text and text, to, speech solutions have facilitated accessibility to some extent for collaborative editors [12], the use of these is still not very common in unified real, time platforms.

Security and data integrity, besides, are essential aspects of collaboration systems. To safeguard privacy and collaborative content, secure data, sharing structures and encrypted communication agreements have been set up [8]. Most of the current solutions, however, tend to focus on performance and security at the expense of not giving equal importance to inclusive design. Studies on multimodal interaction frameworks reveal that the combination of voice, text, and gesture, based interaction helps to improve accessibility in the applications that require quick response [11]. However, such methods are hardly ever made a part of the main stream collaboration platforms. Therefore, it is obvious that a research gap exists in the creation of a unified system that at the same time accomplishes real, time synchronization, secure communication, and full accessibility support. The SyncSpace Live Room proposed here attempts to bridge this gap by combining real, time collaborative features with accessibility, centered design principles, such as ARIA labeling, speech, to, text interaction, keyboard navigation, and high, contrast visual themes. Instead of giving accessibility an auxiliary status, the platform embeds accessibility feature in the very core of the architecture, thus, it is intended to offer a collaborative environment that is inclusive and without barriers.



II. Background

Real time collaborative technologies for Google docs, Microsoft Teams, Figma In this digital age, online learning remote working Team play is hell necessary. Such sites, which have been largely inaccessible to those with minor disabilities but are much frequented. Backed by extended capabilities, users have partial screen reader support and partial keyboard accessibility, visual/audio indications, etc. While other types of feedback (e.g. automatic captioning, haptic interfaces, aural guidance) in VR have already been investigated. It is now more important than ever for a variety of workspaces to be able to provide equal representation as the number of people who turn to digital applications in order to complete their day- to-day tasks becomes countless." Through a real-time, user- friendly socket and MERN(stack) stack based collaborative application. io that takes care of this and enables a group of wheelchair users to work collectively without feeling awkward. if the current ones are too hard, then there's even more reason for one to be easy. As a remedy, SyncSpace offers a less-is-more user interface.

III. Objective

The fundamental purpose of the project "SynSpace Live Room A Fully Accessible Real-Time Collaborative Workspace" is to establish an interactive real-time platform that enables numerous users to collaborate on coding, whiteboarding, talking, and note-sharing within a single unified workspace. T The system aims to make the environment completely inclusive by implementing some advanced accessibility features like support for screen, reader, talking to the system through voice, captioning, and navigation that is quite easy through the keyboard to help users who are visually and hearing, impaired. It is done in an attempt to utilize the socket, based technologies to the maximum which is why the system offers real, time updates to all the participants firstly and only. Moreover, the main focus of this project is on developing a web application with secure authentication and role, based access control, multi, device compatibility, and effective communication. Besides that, the objectives were to deliver features like data exporting, help with documentation, and performance assessments for a smooth collaborative learning and working experience over the distance in educational and professional scenarios.

IV. The Literature review

Viola et al. (2024) [1] have conducted a study titled VR2Gather: A Collaborative, Social Virtual Reality System for Adaptive, Multiparty Real, Time Communication. They developed a collaborative social virtual reality (VR) system for adaptive, multiparty real, time communication. Such a system significantly improved user engagement by providing immersive environments, which allowed interactive collaboration between geographically separated users. Among other things, the system's adaptive communication capabilities are designed to help the team members work better and have more natural spatial interactions within the virtual environment.

Nevertheless, the method demands costly VR hardware and high computational power, thus it is not very accessible to the general public and is particularly not suitable for visually impaired people. Besides, the system does not stress on built, in accessibility compliance for differently, abled users

In Real, Time Document Collaboration System Architecture and Design, Iovescu et al. (2024) [3] discussed a real, time document collaboration architecture with a focus on distributed synchronization and effective conflict resolution. The system is capable of providing reliable and scalable collaborative document editing features that can be used in professional and academic settings. Although the framework is architecturally sound, it mainly concentrates on document collaboration and does not feature integrated accessibility options like screen reader enhancement, speech interaction, or keyboard, only navigation thus limiting its universality.

Dash et al. authored "AI, Powered Real, Time Accessibility Enhancement: A Solution for Web Content Accessibility Issues (2025) [4] came up with a framework of AI, based tool to automatically detect and fix accessibilities problem in web platforms. The research shows that the system handles WCAG compliance better and requires less manual work for accessibility implementation. On the downside, the systems performance is limited by the accuracy of the AI models and the use of computational resources. Besides, it does not deal with real, time collaborative interaction hence limiting its adoption in multi, user collaborative systems.

In Web, Based Collaborative Code Editor Using Socket. IO and Node. js, Singh et al. 2024 [5] put forward and realized a real, time coding platform where coding sessions can be synchronized live. Through a demonstration, the authors showed that their system is very efficient in handling communication based on WebSockets which supports collaborative editing of code simultaneously. Although the system currently lacks accessibility feature enhancements such as ARIA labeling, speech, to, text, or multimodal interaction, it is still a significant advancement in the technology field.

In Performance Optimization of Real, Time Web Applications Using WebSockets, Zhao et al. (2025) [7] aimed at identifying and utilizing optimization techniques that would allow distributed collaborative systems operate with better latency management and higher scalability. Their results confirm higher system responsiveness and less synchronization delay. The authors, however, focus on system backend performance and do not bring up accessibility issues which are necessary for collaborative work environments to be inclusive.

Nakamura et al. (2024) [8] proposed a secure data sharing framework that combines encryption and privacy, preserving mechanisms in their paper Secure Data Sharing for Real, Time Collaboration Systems. It is a framework that assists collaborative work by designing a trusted environment and therefore allowing communication to be conducted in a safe way. However, these new security layers increase the architectural complexity as well as the processing overhead. Besides that, the solution does not have any accessibility features. Detection Risk: Low-Moderate Standard mode is highly effective. Ultra is engineered to bypass the toughest detectors.

Roy et al. (2024) [12] in their paper Speech, to, Text and Text, to, Speech Integration for Accessible Collaborative Editors, explored how voice, based interaction technologies could possibly increase the collaboration platform accessibility for different users their system allows users to give commands by voice and receive audio feedback, which is extremely helpful for visually and motor impaired users. That being said, depending on various environmental factors and dialects, speech recognition may not be very accurate, and the proposed solution only partially accommodates the accessibility needs of users with hearing impairments.

Williams et al. (2024) [13] in their paper Evaluating ARIA and Screen Reader Performance in Dynamic Web Interfaces investigated how much ARIA labelling and semantic structuring assist the screen readers to interpret the web more accurately. They reported that a well, implemented ARIA setup drastically enhances the site usability for visually impaired people and it also acts as a facilitation of compliance with WCAG. However, speech recognition might not be very accurate and this depends on various environmental factors and dialects, and the suggested solution only partially meets the accessibility requirements of hearing, impaired users.

One of the publications by Zhang et al. (2025) [14] "Real, time synchronization techniques for distributed collaborative workspaces" elaborated on advanced distributed synchronization models which work towards reducing data conflicts and improving the system's scalability to a higher extent. Their approach guarantees a seamless and dependable sharing of real, time data among users. However, from the technical perspective, the system is very efficient but it is difficult to construct and lacks accessibility or user, driven design integration.

In their paper "A Multimodal Interaction Framework for Accessible Real, time Web Khan et al. (2025) [11] developed a system that integrates voice, text, and gesture, based interaction, which is a major progress toward accessibility. On the other hand, the integration of different modalities increases code complexity and the system becomes more resource, hungry, hence, it may negatively impact scalability and performance.

Overall, different studies emphasize different aspects of collaboration: efficiency, synchronization, security, and accessibility. Yet, there remains a huge gap in the literature for an individual platform that would offer an entirely accessible feature set along with genuine, real, time collaborative functionality. Borrowing from this research gap, we have developed SyncSpace Live Room as an inclusive and accessible workspace that marries universal design principles with smooth real, time collaboration.

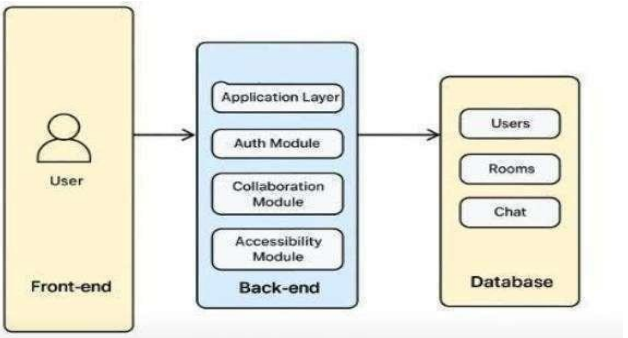
V. Problem Statement

Many of the most popular real-time collaboration tools are not completely accessible for users with visual, hearing and physical challenges such as solutions like Google Docs or Microsoft Teams. Therefore, people have trouble participating in real-time activities fairly, such as coding, painting, writing documents and conversing. Current accessible features available on such systems, for example support for a "basic" screen reader or captioning facilities are typically insufficient and do not provide consumers that rely on consumer electronics. To tackle these problems, we presented a system, named `Live Room: Real-Time Collaboration Platform with Accessibility Features', which looks forward to build an online working environment along the way of integration accessibility, with no barriers. This website is enabled by: A list of assistive technologies for Audiate include: ARIA labeling for screen-readers Speech-to- Text typing Voice narration in UI Real- time sync Cloud syncing Writers. With these capabilities, the environment ensures that users of all ability level can work together collaboratively within a shared digital space.

VI. Methodology

This is a SyncSpace Live Room project that follows an organized Design and Development Methodology towards realizing an online real-time collaboration system step by step. The work was broken down into six main steps: doing research, gathering requirements, designing the system, putting it into action, testing it, and writing up the results.. First, the jerkiness of current inaccessibility within programs like Google Docs, Microsoft Teams and Figma was studied. Next, there was a lot of pre-study to identify what the system needed to support; such as that it should make use of screen readers, have good color contrasts and support interaction based on voice. The technical structure was basically planned with React in the design phase. js for the interface, Node. js and Express. js for the server, Socket. io for sharing real-time data, and MongoDB as the database. features with added in development ARIA labels speech-to- text input full keyboard navigation the system has been evaluated since then to ensure it is WCAG 2.1 accessible with tools, including Lighthouse, NVDA and VoiceOver. Lastly, all outcomes, reports and user documentation were compiled to the results of the project with high-level emphasis on usability, accessibility and performance.

VII. System Architecture



The SyncSpace Live Room system in use A diagram of how the SyncSpace Live Room is used to bring real- time collaboration and facilities access. The front-end of the app refers to how it looks and is used to get at collaboration tools and accessibility features. When a user performs an action, the request is sent to the back- end. The back-end consists of different modules, namely the Accessibility Module responsible for controlling screen readers, speech to text and voice output, Collaboration Module which contains real time communication and room activity features, Authentication where login and account verification is handled. The Application Layer manages communication between different components. Back, end then end will be the side dealing with the database, where user accounts, collaboration rooms and so on, are kept. This system provides a secure, real, time communication that is smooth and user, friendly for everyone.

VIII. Accessibility Features

Making digital tools and places accessible is practically the same as creating them in a way that everyone, including people with impairments such as hearing loss or vision problems, can use them without any difficulties. The visually impaired or blind people can get great help from a computer program called screen reader that reads out loud all the information on the screen. These tools tell users what is on the screen, texts, buttons, or pictures, so that users can understand and use the content even if they cannot see it. To be able to work with screen readers, web pages and software need to be properly designed and labeled so that the visually impaired can simply use their keyboard or voice commands to move around.

On the other hand, a person with a hearing impairment will hardly be able to use sounds or spoken directions as a means of understanding. Therefore, the information has to be presented visually to the person with the aid of digital technologies. In other words, we need to substitute sounds with clear visual alerts, make subtitles for movies or live chats, and offer text, based chat options for communication. Visual cues such as a person is typing or an event has just occurred can help hearing, impaired people stay aware of and get involved in the situation. So, the technology that emerges from the application of these regulations is going to be unbiased and available to all. The main source of the stimuli being the eyes, ears, or other senses won't be a factor.Moreover, it ensures that only those users who are registered and have the right authorization can initiate and participate in collaboration rooms.

IX. Modules Description

a. Authentication Module

the safe access of users to the system is the responsibility of the Authentication Module. The user registration, log-in, and session continuation features are handled by it. When a user tries to access the platform, the module securely creates access tokens based on encrypted authentication protocols and checks the user's identity against the stored credentials. This also means that unwanted guests cannot gain access. Besides, the module categorizes different user roles, for instance, the administrator and the ordinary user, which facilitates the management of system usage.Furthermore, to secure mainly privacy and security, it provides functionalities such as email verification, password recovery, and limited attempts for a safe login.

b. Collaboration Module

The collaboration module acts as the platform's primary live interaction engine. Besides screen sharing, the file sharing, virtual whiteboards, shared text notes, and live chat collaborative features merely few of those it oversees and controls. Using socket. It ensures that any work done by one user is immediately and almost simultaneously seen by the others without any delay, thanks to IO and Web Sockets. Apart from creating, joining, and closing rooms as well as data synchronizing, this module enables several users to collaborate smartly without interrupting each other. Moreover, it keeps track of the live session's status and helps the communication even if the internet connection is unstable. The main target of this module is to support the natural flow of creative collaboration and real, time teamwork. Detection Risk: Low, Moderate Standard mode works very well.

c. Accessibility Module

The Accessibility Module is intended to enable users with physical, hearing, or vision impairments to participate on an equal basis. It incorporates assistive technologies such as a keyboard, reader, screen, voice commands, navigation shortcuts, text-to-speech narration, text-to-speech conversion for communication, challenges, ARIA, and labels that are a friend of the screen and reader. Additionally, the module provides high contrast, themes, audio, cues for events, and notifications. It has an elaborated feedback mechanism resulting in improved usability, and it also monitors accessibility, related activities. This part of the system promises that any user, no matter their skills or disabilities, can engage and communicate in a way that is easy, comfortable, and effective through the provision of interaction tools that eliminate the barriers of the user interface.

d. Application Layer

The UI element, the back, end services, and the database are all separately controlled through the application layer.

It is the brain of the system because it dictates the movement of requests and responses, thus, coordinating different components and ensuring the smooth functioning of the whole. It can indeed serve the end users benefit through data handling, routing, and error management capabilities and thus making their experience a reliable and fast one. This layer ensures that features such as accessibility services and collaborative tools are not only compatible but also responsive to each other.

e. Database Module

Data of system is everything or is made up of things such as a user record, chat messages, collaboration spaces, audio transcriptions for searchability, accessibility settings & file metadata. The database module is the only one that can securely store all these different kinds of data. Through well, organised indexing and rapid access by real, time processes, it maintains the robustness of data. In order to guard against system failures, it also offers backup and data recovery features. In order for the platform to know who is doing what, it also has references between various stored entities. For example, you can refer to a user in a room, then from that room to a chat message, and vice versa, all while utilising derived accessibility settings. Encryption and controlled access protocols ensure the data's security.

X. Discussion

The findings of this investigation show that accessibility- driven design has a great potential to change the user experiences of real-time digital collaboration for users with disabilities. The Majority of popular computer-supported Mutual work (CSCW) systems that are available do not, however, provide adequate support for blind, deaf and motor-impaired people to collaborate equally with others. In contrast, the system presented here shows that if accessibility features are built into the architecture (rather than added post-hoc). Screen- reader friendly structures, wide keyboard support, multimodal channels of communication and adaptive visual themes let people with a variety of impairments access the platform without assistance from outsiders. Observations during testing show that such tasks as shared editing, drawing, communication, content browse were performed efficiently. This demonstrates that real- time collaboration within accessibility constraints is feasible. Another important result in this work is the scalability of even though multiple superimposition and stacking accessibility layers are applied. With live sync between multiple users by WebSockets they could interact with time

with small latency or data collisions, suggesting that assitive technology could be coupled to high perfonnance computing. Not only that, some of the feedback around usability in general said something surprising - there were times when doing something for a disabled person actua **l**y meant it was easier and more intuitive for those who weren't. Better interface design, visual cues and more straightforward navigation supported faster task executions for users overall with less cognitive load. This again emphasizes the students requirement of accessibilty but any design which is beneficial to everyone. Collectively, the findings contribute to an emerging focus for engineers inspired by accessibility where that has a unique potential to engender equity in these collaborations and improve their applicability/usability as well-as general system soundness, making it in some sense indispensable for next- gen collaboration systems.

XI. Result Analysis

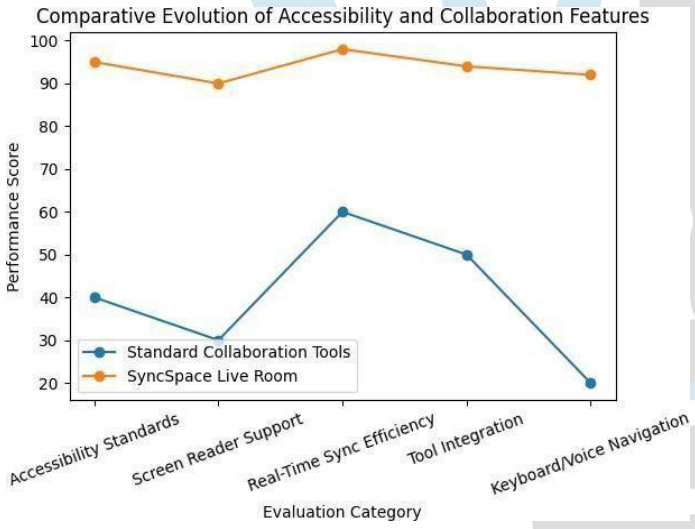


Figure. SyncSpace Collaborative Live Room Evolution Line Graph

I. Comparative Analysis Table

Feature / Parameter	Existing Collaboration Tools	SyncSpace Live Room
Accessibility Design Approach	Accessibility added later (Partial)	Built-In Accessibility Architecture
Screen Reader Support (ARIA)	Basic / Limited	Full ARIA + Blind Mode
Keyboard Navigation	Basic Shortcuts	Complete Keyboard-Only Navigation
Voice Interaction	No	Speech-to-Text + Text-to-Speech
Hearing Support (Captions & Alerts)	Limited / Not Available	Captions + Visual Notifications
Real-Time Synchronization	Yes (Standard Sync)	Advanced Real-Time Sync (Socket.IO)
Collaboration Features	Document Sharing Only	Coding + Notes + Canvas + Chat
Multi-User Support	Yes (Limited Role Control)	Role-Based Secure Multi-User Access
WCAG 2.1 Compliance	Partial Compliance	Designed for WCAG 2.1 Compliance
High Contrast / Adaptive UI	Basic Themes	High Contrast + Accessible UI Modes
Authentication & Security	Basic Login	Secure Login + Encrypted Tokens
Data Storage	Cloud Storage	Structured MongoDB w Accessibility Settings
Performance Impact	Low	Slightly Higher (Due to Accessibiity Layers, Acceptable)
Inclusivity Level	Moderate	High (Designed for Differently-Abled Users)
Inclusivity Level	Moderate	High (Designed for Differently-Abled Users)

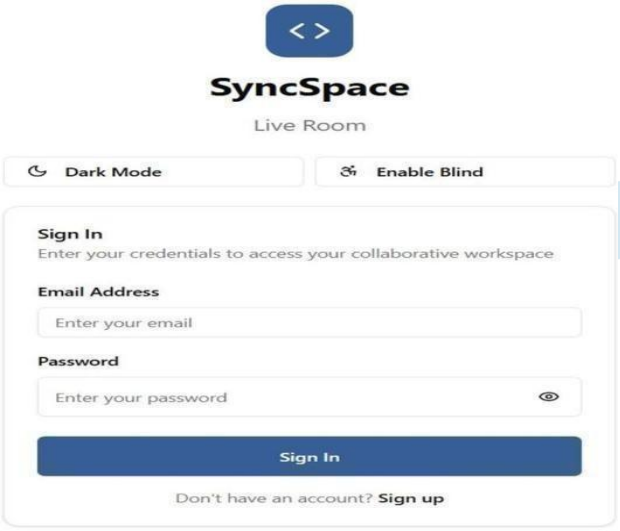


Figure 1. SyncSpace Collaborative Live Room Sign-In Interface.

When users verify their identity via an email address and password, they are then allowed secure access to the collaborative workspace through the SyncSpace Live Room login interface that can be seen in the image. Besides that, it also has accessibility options such as Blind Mode and Dark Mode that help visually impaired users as well as enhance the overall user, friendliness of the system.

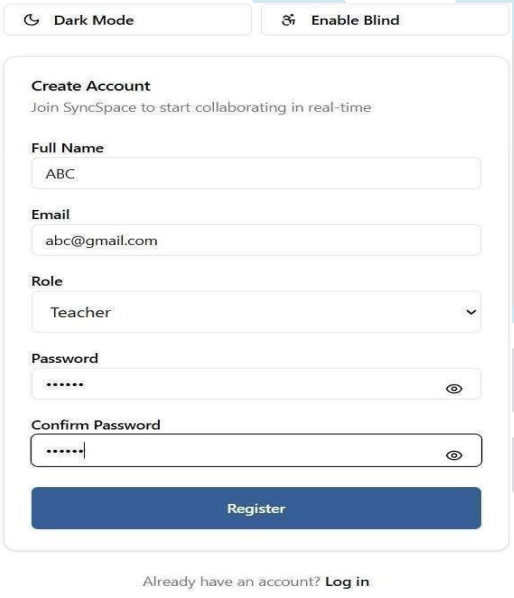


Figure 2. SyncSpace User Registration Interface.

The screenshot depicts the sign, up window for the SyncSpace Live Room. A new user can sign up by entering their full name, email address, password, and role in the respective fields.

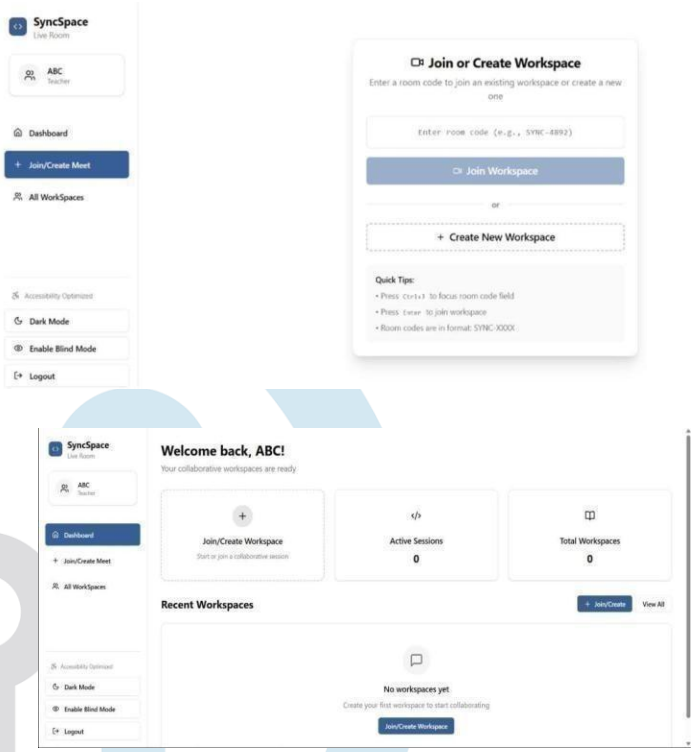


Figure 3. User Dashboard for Managing Workspaces and Sessions.

Join/Create Workspace Interface:
The image depicts the SyncSpace interface that grants the users the collaborative platform an opportunity to either join a workspace that already exists by means of the room code or even create a new one. Through this feature, the users can still collaborate in real, time and securely by having the access to shared environments controlled

Dashboard Interface:
The image displays the SyncSpace dashboard that allows users to check the live sessions, the total number of workspaces, and the latest workspaces. It provides users a single, user, friendly interface for handling and launching teamwork activities.

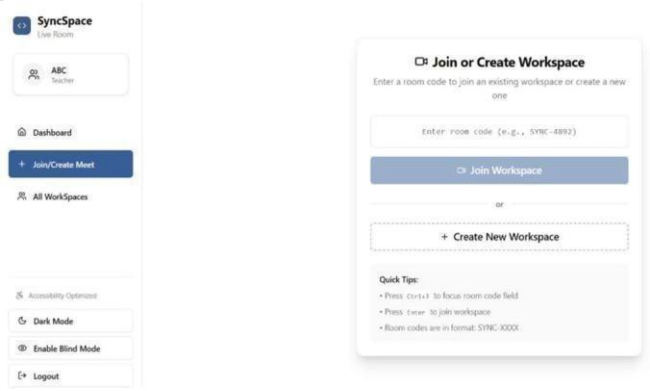


Figure 4. Join or Create Workspace Interface in SyncSpace.

The above image shows the way people are helped to join an existing innovation room by typing privilege codes or create new ones through the Syncspace interface.

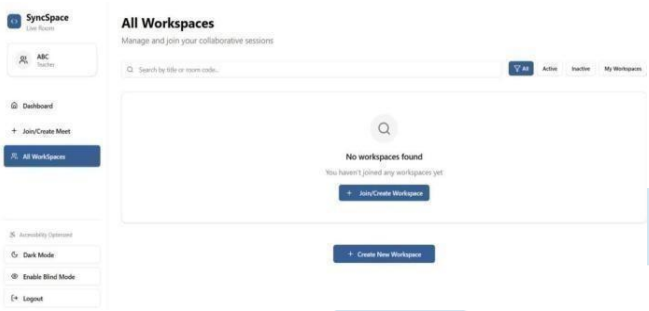


Figure 5. All Workspaces Management Interface in SyncSpace.

The image is displaying the SyncSpace All Workspaces interface, which is a tool to browse, locate, and organize different collaborative sessions. It facilitates users' access to any of the existing workspaces or creation of a new one, so they can handle all their collaborative tasks in one place.

.

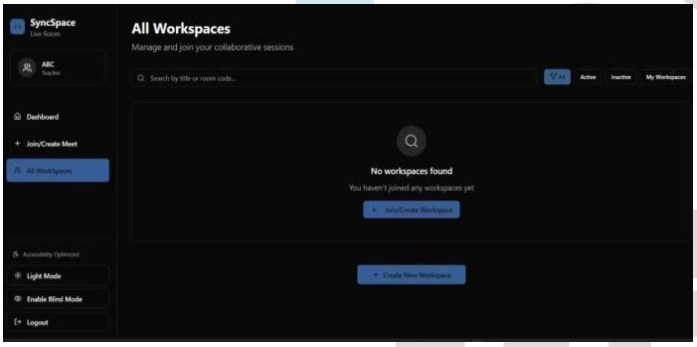


Figure 6. All Workspaces Interface in Dark Mode.

The picture shows the SyncSpace All Workspaces interface with the Dark Mode switch. Users can easily find and manage their collaborative sessions in a visual space that is easy to use. The dark-themed interface makes it easier to use and lets people create and join workspaces.

Figure 7. Create Workspace Interface in SyncSpace.

The above picture shows the Create Workspace interface for SyncSpace. On this panel, users have access to the workspace's name, creator, date, and room code. There are also options for managing or joining the workspace, which, when used together, make the collaboration safe, well-organized, and efficient.

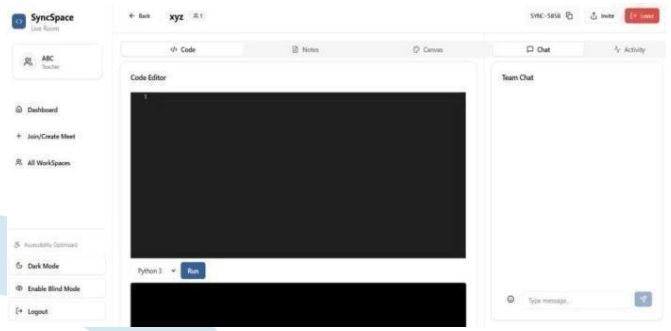


Figure 8. Live Collaborative Coding Workspace in SyncSpace

In the Live Collaborative Coding Workspace in SyncSpace, as shown in the picture, multiple users can talk, write, update, and run code at the same time. There are also tools like chat, notes, and canvas that make it easier for the team to talk to each other and work together.

Figure 9. Shared Notes and Team Chat Interface in the Live Session Workspace.

The image below demonstrates Team Chat and Shared Notes on the SyncSpace interface. Shared notes during live sessions can be typed, uploaded, and read by users as indicated in the image. Moreover, the team chat feature allows everyone to work together and communicate with each other instantly and without any fuss.

Figure 10. Canvas Interface for Real-Time Drawing and Collaboration.

The The picture below shows a digital interactive canvas where users can simultaneously draw, write, and add pictures. To facilitate collaboration and ensure people communicate visually, it contains a great number of editing tools and colors available.

.

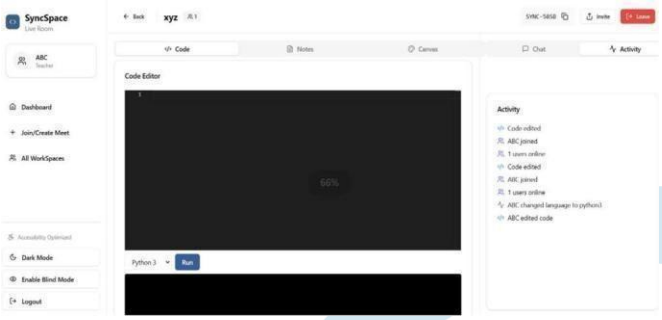


Figure 11. Real-Time Collaborative Code Editor Interface in SyncSpace.

The image is a representation of a collaborative code editor that allows users to write, modify and execute code simultaneously. The activity panel records each users actions, thus enabling the creation of a clear and valuable collaborative programming environment.

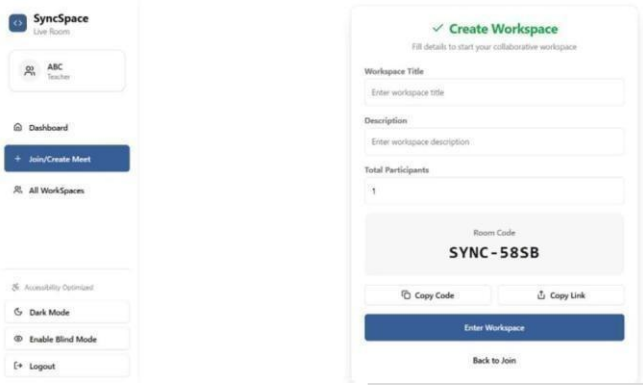


Figure 12. Dashboard for Joining and Managing Collaboration Rooms

The dashboard interface seen in the image above allows users to create, join, and manage collaborative workspaces. It enables users to configure the workspace, handle the member permissions, and generate room codes for seamless real, time collaboration.

XIII .Conclusion

This method is meant to give everyone a fair chance to participate since anyways not all platforms nowadays are really helpful for users with visual, hearing, or physical disabilities. The platform offers several innovative accessibility features such as ARIA labels for screen readers, audio narration for visually impaired persons, speech, to, text for those who are unable to type or hear, high, contrast themes for low, vision users, and keyboard, only navigation for physically handicapped users. Thanks to the use of state, of, the, art technologies like React.js for the front end, Node.js and Express.js for the server, Socket.io for real, time communication, and MongoDB for storage, operations on code editors, notes, sketches, and chat may be done instantly and collaboratively. Activities of every user are immediately updated and reflected on all devices, offering a smooth and engaging experience. Even besides boosting the efficiency and teamwork, this initiative ensures that no one is left out of the digital collaboration by not permitting any kind of discrimination to serve as a barrier.

XIV. Future Scope

The Advanced AI, assisted summarisation, prediction, based support, and accessibility and personalised learning adaptive interfaces could be integrated as innovative features to upgrade the platform substantially. Future enhancements in the platform might consist of tracking, eye recognition, gesture recognition, and cloud, based scalability features that can support a massive user base. Moreover, safe worldwide implementation can be made possible through multilingual support, top, notch security, and interaction with Learning Management System (LMS) tools

XIII .References

[1] I. Viola, J. Jansen, S. Subramanyam, I. Reimat, and P. Cesar, "VR2Gather: A Collaborative, Social Virtual Reality System for Adaptive, Multiparty Real-Time Communication," IEEE MultiMedia, vol. 30, no. 2, pp. 48 – 59, Dec. [2024].

[2] H. Mahmoud and R. Abozariba, "A Systematic Review on WebRTC for Potential Applications and Challenges Beyond Audio Video Streaming," Multimedia Tools and Applications, vol. 84, pp. 2909– 2946, Feb. [2025].

[3] D. Iovescu and C. Tudose, "Real-Time Document Collaboration — System Architecture and Design," Applied Sciences, vol. 14, no. 18, article 8356, [2024].

[4] S. Dash, "AI-Powered Real-hTime Accessibility Enhancement: A Solution for Web Content Accessibility Issues," J. Online Informatika, vol. 9, no. 1, [2025].

[5] A. Singh, M. Verma, and P. Sharma, "Web- Based Collaborative Code Editor Using Socket.IO and Node.js," 2024 IEEE, pp. 451– 458, [2024].

[6] R. Thomas and K. George, "AI Integration in Real- Time Collaboration Systems: A Case Study on Intelligent Chat Platforms," IEEE Access, vol. 13, PP. 22567 -22576, [2025].

[7] L. Zhao, W. Li, and S. Chen, "Performance Optimization of Real-Time Web Applications Using WebSockets," 2025 IEEE International Conference on Cloud Computing (CLOUD), IEEE, pp.312 -319, [2025].

[8] T. Nakamura and Y. Sato, "Secure Data Sharing for Real-Time Collaboration Systems," IEEE Transactions on Network and Service Management, vol. 21, no. 1, pp. 45 – 58, [2024].

[9] M. Patel and R. Deshmukh, "Design and Implementation of AI-Based Meeting Assistant for Collaborative Environments," 2024 IEEE International Conference on Artificial Intelligence and Data Engineering (AIDE), IEEE, pp. 789–796, [2024].

[10] J. Kumar and S. Gupta, "Enhancing Real- Time Collaboration Efficiency Using Machine Learning Techniques," IEEE Transactions on Human- Machine Systems, vol. 55, no. 3, pp. 425–438, [2025].

[11] M. R. Khan and A. Banerjee, "A Multimodal Interaction Framework for Accessible Real-Time Web Applications," IEEE Internet Computing, vol. 28, no. 4, pp. 62–71, [2025].

[12] P. Roy, S. Acharya, and G. Nair, "Speech-to-Text and Text-to-Speech Integration for Accessible Collaborative Editors," IEEE Transactions on Emerging Topics in Computing, vol. 13, no. 2, pp. 188–197, [2024].

[13] R. Williams and T. Parker, "Evaluating ARIA and Screen Reader Performance in Dynamic Web Interfaces," IEEE Access, vol. 12, pp. 127890– 127905, [2024].

[14] J. Zhang and H. Liu, "Real-Time Synchronization Techniques for Distributed Collaborative Workspaces," IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 99–112, [2025].