International Journal for Novel Research in Economics , Finance and Management
www.ijnrefm.com
ISSN:
3048-7722
Volume 4, Issue 3, May-June-2026, PP: 01-06

Language Transcription Using Genai

***Karthiban .R , Rajashakthikumar.R,**
Sabaripriya .S , Selvaganesh. L, Suganthi .R
Sri Shakthi Institute of Engineering And Technology
,Coimbatore Tamil Nadu India
Corresponding Author
E-Mail Id: rkarthiban@siet.ac.in

***Abstract* – Online Conferencing Application is designed to be a communication platform in real time, which allows users to connect via video, audio, and chat from anywhere .The primary objective of this project is to build a system that is fast, secure, and user-friendly for online meetings, virtual classrooms, and remote collaboration.React.js is used to create the frontend, which gives a responsive and interactive user interface for smooth communication. Node.js is used for the backend, which handles server-side operations, user authentication, and data exchange in an efficient manner .In order to have an instant communication in real-time, Socket.io is brought in which makes it possible to have a live chat, notifications, and user synchronization without any waiting time. This set of modern technologies is what makes the system stable, with low latency, and scalable .To sum up, the Online Conferencing Application is an efficient, dependable, and userfriendly tool for virtual communication, which is in line with the increasing demand of online communication in this digital era. With the increasing need for remote communication and online collaboration, this application provides an effective solution that integrates multiple technologies to deliver high-quality video, audio, and data transmission over the internet. The frontend of the application is built using React.js, which provides a responsive and interactive user interface for smooth navigation and dynamic updates. The backend is implemented with Node.js and Express.js, which handle API requests, authentication, and socket connections. The system uses Socket.IO to maintain low-latency bidirectional communication between the client and server, ensuring real-time synchronization of audio, video, and messages among all participants. The Online Conferencing Application is a comprehensive real-time communication platform designed to support seamless interaction through video, audio, chat, and intelligent language processing. The system aims to provide a fast, secure, and user-friendly environment for virtual meetings, online classes, and remote collaboration.The frontend of the application is developed using React.js, which provides a responsive and dynamic user interface, along with capabilities for capturing live audio and video streams. The backend is implemented using Node.js and Express.js, handling API requests, authentication, and real-time data processing efficiently. Socket.IO (WebSockets) is used to enable low-latency, bidirectional communication, ensuring smooth synchronization of messages, media streams, and user activities.The application includes several advanced features such as a chatbox for real-time messaging, a live participant list to monitor users in a meeting, active speaker identification, automatic video recording, recording history, and meeting history for future reference. Security is enhanced through unique passwords for each meeting room, along with administrative controls that allow the host to manage participants, including removing users during an ongoing session.A key innovation in this system is the integration of a real-time voice translation module (VIDEO-TRANSLATE). In this feature, user speech during a video call is captured in the React frontend and transmitted to the Node.js backend. The backend leverages the Groq API to perform Speech-to-Text (STT), converting audio into text, and then translates the text into a user-selected target language. The translated text is sent back to the client and displayed as live subtitles. Additionally, the browser's built-in SpeechSynthesis API is used to convert the translated text into speech, enabling users to hear the translated output instantly. This feature supports multilingual communication and enhances accessibility for diverse users.Overall, the Online Conferencing Application is a scalable, reliable, and feature-rich platform that meets modern communication needs. By integrating real-time conferencing with intelligent voice translation, the system provides an advanced and inclusive digital communication experience.**

Keywords: **Real-time communication ,Virtual meeting ,Screen sharing ,Video conferencing.**

I. INTRODUCTION

In today's digital era, online conferencing applications have become an essential part of communication and collaboration across the world [1]. With the increasing demand for remote work, online education, and virtual meetings, there is a need for a reliable and secure platform that allows users to connect in real time [2]. The Online Conferencing Application is designed to provide high-quality video, audio, and chat communication, making it a complete solution for virtual interaction [3].The system focuses on creating a user-friendly and efficient platform that supports real-time collaboration among participants located in different geographical regions [4]. It ensures smooth performance even under varying network conditions by implementing adaptive bitrate technology [5]. The frontend of the application is developed using React.js, which provides a dynamic, scalable, and responsive interface suitable for modern web applications [6]. The component-based structure of React allows developers to manage complex interfaces with ease and ensures seamless user experiences [7].For the backend, Node.js is employed to handle server-side operations such as data management, authentication, and connection handling [8]. Node.js offers event-driven, nonblocking architecture, which supports fast processing and makes it ideal for real-time applications [9]. To enable instant communication between users, Socket.io is integrated for managing live data transfer, allowing instant updates during video calls, messaging, and file sharing [10].The

International Journal for Novel Research in Economics , Finance and Management
www.ijnrefm.com
ISSN:
3048-7722
Volume 4, Issue 3, May-June-2026, PP: 01-06

application also emphasizes data security and privacy, which are major concerns in In today's digital era, online conferencing applications have become an essential part of communication and collaboration across the world online communication platforms [11]. End-toend encryption, secure login authentication, and encrypted storage are implemented to ensure the safety of user information [12].

Moreover, the system includes essential collaboration features such as screen sharing, meeting recording, and whiteboard interaction to enhance productivity and teamwork [13].In addition, the application is designed to operate efficiently across multiple devices and browsers, providing flexibility and accessibility for users around the globe [14]. Overall, the Online Conferencing Application serves as a fast, safe, and effective tool for realtime communication and collaboration in educational, corporate, and personal environments [15]. The Online Conferencing Application is a comprehensive web-based platform developed to enable users to conduct virtual meetings, conferences, and collaborative discussions in real time. With the increasing need for remote communication and online collaboration, this application provides an effective solution that integrates multiple technologies to deliver high-quality video, audio, and data transmission over the internet. Include text-based chat, voice activity detection to highlight active speakers, participant count tracking, and admin control features such as muting participants or ending the meeting.

The application also ensures scalability and security, employing HTTPS protocols and room-based isolation of communication channels. The primary objective of this project The system allows a host (admin) to create a unique meeting room that other participants can join using a generated room ID. To enhance privacy and control, the application includes an authentication and approval mechanism where the admin can approve or deny participants before they join the meeting. Real- time communication between users is achieved using WebRTC (Web Real-Time Communication) integrated with Socket.IO for signaling and peer connection establishment. The frontend of the application is built using React.js, which provides a responsive and interactive user interface for smooth navigation and dynamic updates. The backend is implemented with Node.js and Express.js, which handle API requests, authentication, and socket connections.

The system uses Socket.IO to maintain lowlatency bidirectional communication between the client and server, ensuring real-time synchronization of audio, video, and messages among all participants. Additional functionalities is to provide a secure, efficient, and user-friendly conferencing platform that can be utilized for online education, business meetings, and remote collaboration. The system eliminates the geographical barriers to communication, enabling effective teamwork and information exchange in a digital environment.In conclusion, the Online Conferencing Application demonstrates the potential of integrating modern web technologies to build a real-time, scalable, and reliable communication system that supports both personal and professional virtual interactions.

II. LITERATURE SURWAY:

In today's digital age, online conferencing applications have become key tools for communication. They support the rapid growth of remote work, online education, and virtual collaboration. Modern conferencing systems rely on real-time web technologies like WebRTC.This technology allows direct transmission of video, audio, and data between browsers without needing extra software. For creating responsive and userfriendly interfaces, React.js is popular. Its modular design and efficient rendering through the virtual DOM allow for quick updates. On the backend, Node.js has become well-liked for its event-driven, non-blocking I/O architecture. This feature manages many user requests while maintaining stable performance. Recent studies highlight the importance of real-time data transfer tools like Socket.io.

This tool enables instant, twoway communication between clients and servers. Socket.io works well for live chat, notifications, and sync, making it perfect for interactive conferencing systems. By combining React.js for the frontend, Node.js for backend tasks, and Socket.io for realtime communication, developers can build scalable, secure, and high-performing conferencing platforms that enable smooth collaboration and improve user experience in today's connected digital world. There has been a rise in research into webconferencing and online meeting technologies, especially during and after the COVID19 pandemic. For example, a systematic review on WebRTC (which underpins many conferencing systems) found 83 unique peer-reviewed publications across IEEE, ACM, Springer etc, showing steady growth in real-time communication research. Many studies emphasise the adoption in educational contexts: e-learning shifted rapidly to synchronous conferencing tools, and investigations into perceptions, usability and technology acceptance followed.

There is also growing work on challenges: low-bandwidth and unstable network conditions, security/privacy issues, "Zoom fatigue" and user-experience problems have been flagged. Language transcription, which involves converting speech into text and vice versa, has been extensively studied in the fields of Artificial Intelligence, Natural Language Processing (NLP), and Speech Processing. This section reviews existing research related to Speech-to-Text (STT), Text-to-Speech (TTS), and real-time speech translation systems.Early research in speech recognition focused on Automatic Speech Recognition (ASR) systems, which convert spoken language into text. Traditional approaches used statistical models such as Hidden Markov Models (HMM) and Gaussian Mixture

International Journal for Novel Research in Economics , Finance and Management
www.ijnrefm.com
ISSN:
3048-7722
Volume 4, Issue 3, May-June-2026, PP: 01-06

Models (GMM). However, these methods faced limitations in handling natural conversations, noise, and multiple languages .With advancements in deep learning, modern STT systems now use neural networks such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and transformer-based models, significantly improving transcription accuracy and robustness .Recent studies highlight the growing importance of real-time speech transcription systems. These systems process live audio streams and convert them into text with minimal delay. However, challenges such as latency, audio segmentation, and maintaining context still exist. Research shows that techniques like Voice Activity Detection (VAD) help balance transcription accuracy and delay in real-time systems .In addition to transcription, speech translation systems have gained attention. These systems combine STT and Machine Translation (MT) to convert speech from one language into text in another language.

Traditional systems follow a pipeline approach (STT → Translation → TTS), but recent research explores end-to-end models that directly translate speech across languages, reducing latency and improving efficiency .Text-to-Speech (TTS) technology has also evolved significantly. Earlier rule-based systems have been replaced by neural speech synthesis models, which generate natural and human-like speech. Modern TTS systems focus on improving speech quality, expressiveness, and speed using deep learning techniques . Studies also evaluate trade-offs between synthesis speed and audio quality to ensure real-time usable.Recent research emphasizes the integration of STT and TTS into real-time communication systems, such as video conferencing and virtual assistants. Technologies like WebRTC and browser-based speech APIs enable low-latency communication, making live transcription, subtitles, and voice translation feasible in practical applications .

III. PROPOSED FRAMEWORK:

The proposed framework for the Online Conferencing Application is designed to provide a reliable, secure, and real-time communication system that supports seamless interaction through video, audio, and chat. In today's digital environment—where online meetings, virtual classrooms, and remote collaborations have become essential—this framework aims to offer an efficient, scalable, and user-friendly platform for smooth and uninterrupted communication. The architecture integrates modern web technologies such as React.js, Node.js, Socket.io, WebRTC, and MongoDB to ensure flexibility, performance, and highlevel security. The frontend of the application is built using React.js, a popular JavaScript library that enables the development of dynamic and interactive user interfaces. React.js plays acrucial role in enhancing the overall user experience by ensuring quick rendering, responsive layouts, and modular design.Its component-based structure allows developers to reuse UI elements efficiently, reducing redundancy and improving maintainability. Users benefit from an intuitive interface that supports features like screen sharing, chat messaging, and video conferencing with minimal latency. Additionally, React 's virtual DOM improves the speed of updates and transitions, ensuring that changes appear instantly during live sessions. This helps participants experience real-time responsiveness, which is vital for smooth and engaging meetings. On the backend, the application leverages Node.js, a high-performance JavaScript runtime that excels in handling multiple connections simultaneously. Its eventdriven, non-blocking I/O model ensures fast and efficient processing, making it ideal for applications requiring realtime updates. The backend handles key functionalities such as user authentication, session management, and data synchronization, ensuring that users remain securely connected throughout the meeting. Node.js also manages communication between the client and server, maintaining continuous updates for all active participants. To achieve real-time communication, the framework integrates Socket.io, which enables instant bidirectional data exchange between clients and servers. This ensures that chat messages, participant status changes, and meeting updates are reflected immediately without requiring page refreshes.

Socket.io plays a crucial role in maintaining the fluidity of live interactions, reducing delays, and enhancing responsiveness. Furthermore, the system employs WebRTC (Web Real-Time Communication) for peer-to-peer media streaming, which allows direct video and audio transmission between participants.WebRTC enhances the quality of live communication by minimizing latency, reducing server dependency, and supporting adaptive bitrate streaming. This ensures that users experience clear video and audio, even under varying network conditions. For data management, the framework uses MongoDB, a NoSQL database known for its scalability, flexibility, and highspeed performance. MongoDB stores user profiles, meeting details, and chat histories securely in a structured yet WebRTC enhances the quality of live communication by minimizing latency, reducing server dependency, and supporting adaptive bitrate streaming.

This ensures that users experience high-quality video and audio, even under fluctuating network conditions. For data management, the proposed framework integrates MongoDB, a powerful NoSQL database renowned for its scalability, flexibility, and performance efficiency. MongoDB securely stores user profiles, meeting details, and chat histories in a structured yet adaptable format. Its document- oriented data model enables rapid retrieval and efficient management of large volumes of information, which is crucial for real-time conferencing systems that handle multiple concurrent sessions. The combination of WebRTC and MongoDB ensures seamless synchronization between media transmission and data storage, improving

International Journal for Novel Research in Economics , Finance and Management
www.ijnrefm.com
ISSN:
3048-7722
Volume 4, Issue 3, May-June-2026, PP: 01-06

overall system responsiveness. Furthermore, the integration of React.js for the frontend, Node.js and Socket.io for the backend, and WebRTC for live interaction provides a modern, secure, and interactive communication platform. In addition, this framework can be enhanced with AI-based features such as real-time language translation, background noise suppression, and automatic meeting transcription, making virtual communication more intelligent and user-friendly. Overall, the system offers a reliable, scalable, and high- performing solution that ensures smooth and immersive conferencing experiences for users worldwide.

The proposed framework for the language transcription system is designed to enable real-time speech recognition, translation, and audio playback within the Online Conferencing Application. It follows a client-server architecture where the React.js frontend captures live audio from users during a video call and transmits it to the Node.js and Express.js backend through WebSockets or HTTP protocols. The backend processes the received audio using the Groq API, which performs Speech-to-Text (STT) to convert spoken language into text and then translates it into a user-selected target language. The translated text is sent back to the frontend, where it is displayed as live subtitles for better understanding. Additionally, the system uses the browser's SpeechSynthesis API to convert the translated text into speech, allowing users to hear the output instantly. This framework ensures low latency, efficient data processing, and seamless multilingual communication, making it highly suitable for real-time virtual.

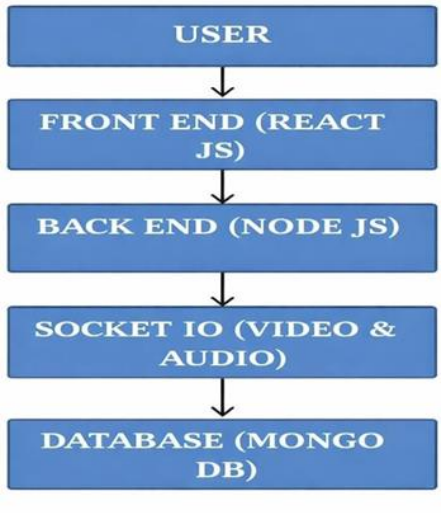

Fig:1 proposed framework diagram

This diagram illustrates the architecture of a full-stack web application and how different components interact with each other. The process starts with the user, who accesses the application through a browser or device. The request is handled by the front end, built using React, which is responsible for creating an interactive and responsive user interface. The front end sends user actions and data to the back end.The back end, developed using Node.js, processes these requests, applies business logic, and manages communication between the client and the database. For real-time features such as video calls, chats, or live updates, Socket.IO is used, enabling instant bidirectional communication between the client and server.Finally, all application data is stored in MongoDB, a flexible NoSQL database that efficiently handles large volumes of structured and unstructured data. This layered architecture ensures scalability, real-time interaction, and smooth data flow throughout the application.

Fig:2 Block diagram for Language Transcription

The video call app flow diagram and proposed frame work is shown by Fig 1he proposed framework for the Online Conferencing Application provides a secure, reliable, and real-time communication system supporting video, audio, and chat. The frontend, developed using React.js, ensures an interactive and responsive interface for smooth meetings with screen sharing and real-time messaging. Node.js powers the backend, efficiently handling authentication, data processing, and server operations. Socket.io enables instant communication between users and servers for seamless updates in calls and chats. WebRTC supports peer-to-peer streaming for high- quality media performance, while MongoDB securely stores user and meeting data, making the entire system scalable, stable, and efficient for modern virtual collaboration.

IV. RESULTS AND DISCUSSION:

The online conferencing application functioned properly and enabled real-time chat, audio, and video communication. Joining and creating meetings was made simple and seamless by the React.js frontend. The Node.js backend effectively handled connections, data, and user login. Socket IO integration made it possible to stream audio and video in high definition with minimal latency an to automatically record meetings on video for later

International Journal for Novel Research in Economics , Finance and Management
www.ijnrefm.com
ISSN:
3048-7722
Volume 4, Issue 3, May-June-2026, PP: 01-06

viewing. MongoDB was used to securely store all user and meeting data. All things considered, the project turned out to be a safe, dependable, and quick platform for remote collaboration and online meetings. The developed Online Conferencing Application successfully enables realtime video, audio, and text communication between multiple users through a web-based platform. The system allows users to create and join virtual meeting rooms using unique IDs, with an integrated admin approval mechanism for secure participation. Using technologies such as React.js, Node.js, Express.js, and Socket.IO, the application provides smooth performance with low latency and stable connectivity. Features like voice activity detection, chat messaging, and screen sharing enhance interactivity and collaboration among participants. Testing results show that the application performs efficiently under normal network conditions, offering clear audio, good video quality, and quick synchronization between users. Overall, the project meets its objective of providing a secure, scalable, and user-friendly online conferencing solution suitable for education, business, and remote communication purposes.

.

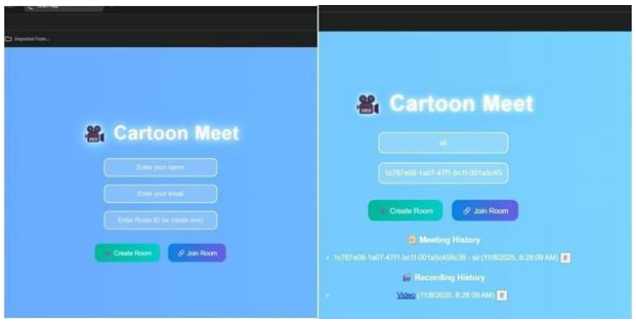

Fig:3 User Interface to select visualization period The video call application discussion and final result is shown by Fig 4

The Online Conferencing Application efficiently enabled real-time chat, audio, and video communication. React.js provided a smooth interface for meetings, while Node.js managed data and authentication. Socket.IO supported high-quality streaming and automatic recording, and MongoDB securely stored data, creating a reliable, fast, and secure platform for online collaboration.

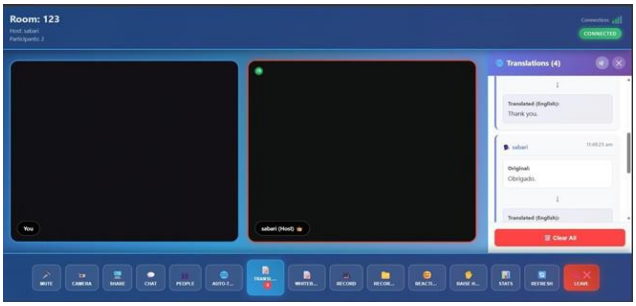

Fig: 4 Autodetection and language Transcription

The implementation of the Online Conferencing Application with the integrated language transcription and voice translation module produced effective and reliable results in real-time communication scenarios. The system successfully captured live audio from users and converted it into text using the Groq API for Speech-to-Text (STT). The transcribed text was accurately translated into the user-selected target language and displayed as live subtitles with minimal delay.The Text-to-Speech (TTS)

functionality using the browser's SpeechSynthesis API worked efficiently, allowing users to hear the translated output instantly. The integration of WebSockets (Socket.IO) ensured low-latency communication between the frontend and backend, resulting in smooth synchronization of audio, video, and text data.Additional features such as chatbox, live participant list, active speaker identification, automatic video recording, recording history, meeting history, and admin controls (like removing participants) functioned as expected. The system also maintained secure access through unique passwords for each meeting room.Overall, the application demonstrated good performance in terms of speed, accuracy, usability, and scalability, making it suitable for real-time multilingual communication in virtual meetings.

V.CONCLUSION AND ENHANCEMENT

Online Conferencing Application was effectively created to offer a seamless and safe platform for chat, audio, and video communication in real time. A responsive and effective system design was guaranteed by using React.js for the frontend and Node.js for the backend. The user experience was improved by the integration of Socket IO, which allowed for automatic video recording and high-quality audio and video streaming. In terms of speed, dependability, and usability, the system did well. All things considered, this project demonstrates how cutting-edge web technologies can The Online Conferencing Application has been successfully designed and implemented to provide a reliable, real-time communication platform for users to connect through video, audio, and chat from remote locations.

The system integrates React.js for the frontend, Node.js and Express.js for the backend, and Socket.IO for real-time bidirectional communication. The inclusion of features such as admin approval, voice activity detection, screen sharing, and participant tracking enhances both security and user interaction during virtual meetings. Through extensive testing, the application demonstrated efficient performance with low latency, smooth streaming, and quick synchronization among participants under moderate network conditions. The user interface is intuitive, responsive, and easy to navigate, ensuring a positive user experience across devices. The project fulfills its objective of providing a secure, scalable, and userfriendly conferencing platform suitable for online

learning, business meetings, and remote collaboration. In conclusion, the developed system highlights the potential of integrating modern web technologies to create a cost-effective, real-time communication solution that meets the growing need for digital connectivity in today's world . In conclusion, the developed Online Conferencing Application with real-time language transcription and translation capabilities provides an advanced solution for modern communication needs. By integrating technologies such as React.js, Node.js, Socket.IO, Groq API (STT and translation), and SpeechSynthesis (TTS), the system delivers a seamless and interactive user experience.The addition of the VIDEO-TRANSLATE module significantly enhances the application by enabling users to communicate across different languages through live subtitles and instant translated audio.

This improves accessibility, inclusivity, and overall communication efficiency.The system is scalable, reliable, and capable of supporting various features required for virtual collaboration, including meeting management, recording, and participant control. Despite challenges such as handling noise and improving translation accuracy, the application performs effectively in real-time environments.Future improvements can include support for more languages, enhanced AI models for higher accuracy, and better noise handling techniques. Overall, the project successfully meets its objectives and represents a powerful tool for real-time, multilingual virtual communication.

REFERENCES

[1] H. Mahmoud and R. Abozariba, "A Systematic Review on WebRTC for Potential Applications and Challenges Beyond Audio Video Streaming," Multimedia Tools and Applications, Springer, 2024.
[2] Y. Li, Z. Zhang, H. Chen, and Z. Ma, "Mamba: Bringing Multi-Dimensional ABR to WebRTC," Proceedings of the ACM Multimedia Conference, 2023.
[3] R. B. Patel, A. Singh, and P. Sharma, "Design and Implementation of Video Conferencing System using WebRTC," IEEE Xplore, 2021.
[4] A. Chatterjee and D. Roy, "A Survey on Real-Time Communication Technologies: WebRTC and Alternatives," Springer Journal of Communication Systems, 2020.
[5] J. Chen, L. Zhao, and K. Wong, "Scalable Video Coding for Video Conferencing Applications," IEEE Transactions on Circuits and Systems for Video Technology, 2019.
[6] M. Gupta and S. Kumar, "QoS Management in Real-Time Video Communication," Computer Networks, Elsevier, 2022.
[7] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2025.
[8] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv preprint arXiv:2212.04356, 2024.
[9] A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," arXiv preprint arXiv:1609.03499, 2016.
[10] W. Xiong et al., "The Microsoft 2017 Conversational Speech Recognition System," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.
[11] V. Pratap et al., "Massively Multilingual Speech Recognition," Proceedings of Machine Learning Research, 2020.
[12] Mozilla Developer Network (MDN), "Web Speech API – SpeechSynthesis (Text-to-Speech)," 2026.
[13] Groq Inc., "Groq API Documentation – Speech-to-Text and AI Processing," 2026.