



ISSN: 2814-1709

CTICTR 4(1): 94 - 106 (June 2025)

Received: 08-03-2025

Accepted: 04-06-2025

<https://doi.org/10.61867/pcub.v3i1a.201>

DESIGN AND IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE-DRIVEN VIDEO CONFERENCING APPLICATION

**Eweoya Ibukun
Adigun Taiwo
Omola Israel C.
Adewuyi Joseph
Idepefo Felix
Sodiq Kazeem
Oladipo Sunday
Ojenike Abimbola**

¹eweoyai@babcock.edu.ng

School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Design and Implementation of Artificial Intelligence-Driven Video Conferencing Application

Eweoya Ibukun¹, Adigun Taiwo², Omola Israel Chukwuyem³, Adewuyi Oluwaseyi⁴,
 Idepefo Felix⁵, Sodiq Kazeem⁶, Oladipo Sunday⁷, Ojenike Abimbola⁸
^{1,2,3}Software Engineering Department, Babcock University, Ilishan, Ogun State
^{4,5,7}Computer Science Department, Babcock University, Ilishan, Ogun State
⁶Department of Computer Engineering, Yaba College of Technology, Yaba, Lagos State
⁸Olabims Integrated Services Limited, Ajah, Lagos State

¹eweoyai@babcock.edu.ng

^{1,2,3,4,5,7} School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Abstract

In modern digital communication, video conferencing has become an essential tool across various sectors; including work, education, and general communication. Traditional video conferencing systems have come a long way from their inception; however, they are yet to fully integrate artificial intelligence to provide necessary features such as real-time and post-meeting transcripts and automated summaries of the meeting. Hence, this study aims to modernize the video conferencing landscape by implementing features such as the previously stated real-time and post-meeting transcripts and automated emails containing the summary of the meeting. It still implements the core features required of any video conferencing application, such as the ability to create and schedule meetings, screen sharing, and audio and video communication. Natural Language Processing (NLP) models played a significant role in the implementation of the system through the Large Language Model (LLM) and Speech-To-Text (STT) models for the transcription of the meeting. In this study, Claude-3.5-Haiku and the OpenAI Whisper model were used for the LLM and STT models, respectively. The front end was done using React js and Tailwind CSS for a modern responsive user interface, while Python and FastAPI were used to manage backend functionalities, including database management and integration with the third-party APIs. The results highlight the effectiveness of integrating AI into video conferencing systems, improving user experience through enhanced automation and real-time processing. Future recommendations include further optimizing AI-driven features and expanding compatibility with additional third-party services to maximize system performance and adaptability.

Keywords: Video Conferencing, AI, Natural Language Processing, Large Language Models, Speech-to-Text, FastAP

1.0 INTRODUCTION

The COVID-19 pandemic caused significant changes in the process by which people work and learn, pushing individuals to shift towards online methods to address the issue of distance. This led to a dramatic rise in the use of video conferencing platforms such as Zoom, Microsoft Teams, and Google Meet. The tools are now essential for collaboration and communication in the work and educational sectors [1].

The driving technology behind modern video conferencing applications is the Web Real Time Communication (WebRTC). Previously, video chats were so-called photo chats due to the low bandwidth of the channels, and a video stream was never sent, but it was an array of images at certain intervals. The first video chats worked using Flash technology. But it was not very secure, leading to the need for a standard or technology that would describe all kinds of processes for establishing a secure connection. WebRTC has become such a standard; it is a technology that allows you to establish a connection between two or more clients, allowing for the transfer of both audio and video stream data as well as text messages and media using a browser [2].

Video conferencing has come a long way from its inception: from low-bandwidth, low-quality video streams with few participants to 720p to HD video streams. Even with these advancements, it leaves many things to be desired, such as higher-quality video streams; a way to easily capture what was said during the meeting; an accurate transcript of the meeting; a digital whiteboard with real-time editing and collaboration capability; and much more.

Advancement in artificial intelligence can help address such needed features not present in most video conferencing solutions, such as speech recognition for transcription and the use of a LLM (Large Language Model) to generate summaries for quick reviews. This study aims to utilize these advancements to help address the challenges associated with utilizing AI models such as Whisper and Claude, respectively, for these issues.

The Whisper model was developed by OpenAI; it is an advanced speech recognition system designed to handle diverse languages and noisy environments. Whisper Architecture is an encoder-decoder transformer architecture; it processes speech as a sequence-to-sequence problem. The audio input is transformed into a log-mel spectrogram, which is processed by the encoder to extract meaningful patterns. Text is then generated through its decoder, using the attention mechanisms to pay attention to crucial parts of the audio [3].

GPT and other large language models (LLMs) process and produce text that is human-like by applying deep learning algorithms. Their basis is the Transformer Model neural network architecture, which understands the relationships between words in a phrase regardless of their placement by using mechanisms like self-attention. In order to train the model and modify the millions or even billions of parameters, it needs a great deal of text data, learning patterns, context, syntax, and meaning. Through iterative learning, it improves over time by predicting the subsequent word in a sequence based on the context of earlier words. After training, LLMs can use the information and patterns they've discovered in the data to produce logical text, respond to queries, or carry out activities [4]. Hence, this work leverages on AI, to intelligently provide post-meeting summary unlike many existing platforms.

2.0 LITERATURE REVIEW

In this literature review, the current state of video applications, their effectiveness, use cases, limitations, and areas of improvements are explored. It covers the theoretical review of past bodies of work and the technology behind them, using this to improve the existing bodies.

2.1 Historical Background of the Research

Video Conferencing can be traced back to as early as April 7, 1927, when a one-way video call took place between the cities of Washington and New York. The call happened through a telephone circuit; between the AT&T president then; sending video to the U.S. Secretary of Commerce then, who received the video [5]. Also, a video call using a picture-phone prototype in 1956.

The prototype used standard analog public switched telephone network (PSTN) telephone lines to send still photos every two seconds [6].

Widespread use of video conferencing did not really start until the 1980s, when businesses like Mac led the way in computer technology. Businesses and individuals alike began considering video conferencing as an affordable alternative to in-person meetings after webcams were introduced. Additionally, it gave them the freedom to meet anywhere in the world, which improved the worldwide brand identification of their company.

Online video conversations were very popular by 2003, when Skype, which quickly rose to prominence as one of the most widely used video conferencing services worldwide. This was due to the fact that it was a free cross-platform service that enabled simultaneous messaging and voice communication between several users. Skype was based on the concept of decentralized “peer-to-peer,” which was used by Kazaa (a previous system they created). The way Skype worked was different from the videoconferencing application since it doesn’t use any of the signaling protocols or the standardized audio codecs [7]. It was based on a peer-to-peer architecture, where the only central element was an authentication server; audio used ILBC, ISAC, or IPCM codecs developed by Global IP Solutions [8]. The audio codecs were designed to be resistant against packet loss. Skype used both UDP and TCP for the transport of the encoded media, making it robust to work behind firewalls.

Upgrades were made, such as the FaceTime app, which came out in 2010. It was a video chat app between Apple smartphones and later also Mac computers. 2011: Microsoft bought Skype and relaunched it as Skype for Business in 2015, and then merged it with Microsoft Teams in 2017. WhatsApp Incorporated video calls by 2016 and 2018; it allowed group video calls.

Even with the cloud services available, it was important to develop technologies that would allow Internet browsers to play multimedia content in a standard way. Before then, to use video conferencing sites, you would have to have installed extensions, which made it difficult to use. By 2011, Google made the source code of the WebRTC project available for developers. [9]. By providing a consistent, standardized set of functionalities, WebRTC enabled audio and video to be accessible across browsers. It started being added to various browsers. This led to the “click and enter the conference” paradigm. In other words, the ability to launch a video conference session from a browser without requiring the installation of extensions specific to each platform and operating system or the execution of third-party programs.

Prior to the COVID-19 epidemic, video conferencing was not particularly common. Due to the global population's mass confinement, video conferencing has essentially become the primary means of maintaining visual communication with family and coworkers. Zoom swiftly established itself as one of the leading choices on a mass level. The quantity of Zoom meeting minutes grew 2500-fold between January and April 2020 [10]. Notably, the number of users and subscribers on other platforms also increased. By 2024, Microsoft Teams had 320 million users, up from 20 million in 2019 [11]. Every business that provided video conferencing services had to swiftly add new features and enhancements. For instance, Teams only allowed video conferencing for four individuals at the start of the pandemic. He later increased the number of participants to nine, and in July 2020, he permitted the simultaneous presentation of up to 49 persons. Zoom promptly fixed the security flaws that had previously drawn criticism. Video conferencing swiftly became a part of everyday life, both in the workplace and in personal and family contexts.

Following a two-year epidemic, video conferencing technologies matured and became widely used in 2024. It has become commonplace for many people to meet both personally and professionally

by taking part in a video chat or video conference. Cloud services, many of which are free or have extremely low fees per use for individuals or businesses, have largely supplanted costly hardware-based videoconferencing systems [12] [13].

2.2 Theoretical Review

The technology being utilized for the application, WebRTC is a peer-to-peer technology that provides web browsers with the ability to communicate in real time (RTC (Real Time Communication)). WebRTC is built into almost all modern browsers; this enables them to communicate with each other instead of just to web servers. When the connectivity is established, browsers can share audio and video streams from the microphone and camera, respectively. It also allows for them to share files, messages, and more, and it does so in a very fast way, "peer-to-peer." [14]. It works by creating an `RTCPeerConnection` instance that connects to another `RTCPeerConnection` instance in another browser or to an endpoint implementing the required protocols. The communication is controlled through the exchange of control messages (the signaling protocol) over a signaling channel provided by any means, but usually a script in the webpage, for example, using `WebSockets` or `XMLHttpRequest` [2].

AI is defined as "the simulation of human intelligence in machines that are programmed to think and learn like humans." [15], AI refers to the theory and development of computer systems with the ability to perform tasks that historically required human intelligence, such as decision-making, speech recognition, and pattern identification. It is an umbrella term encompassing a wide range of technologies, i.e., machine learning, deep learning, and natural language processing (NLP) [16].

AI is an emerging field where lots of innovations take place every day across various fields, such as computer vision, which enables computers to understand visual data (images, videos); supervised learning, training machines on labeled data; unsupervised learning, training machines on unlabeled data; and reinforcement learning, where machines try to maximize the cumulative reward in a dynamic environment. The technology has found its way into every sector, be it medicine, finance, education, construction, and so on and so forth.

Natural Language Processing (NLP) is the focus here. According to [17], NLP is "a branch of artificial intelligence that focuses on enabling computers to interact, understand, interpret, and generate human language." It contains a wide group of techniques and algorithms for processing and analyzing natural languages, such as text and speech.

NLP leverages machine learning algorithms trained on unstructured data, usually text, to analyze how human language is structured to convey meaning. The data required for can be massive, ranging from phrases to entire books are used as training data. An NLP models identify patterns in the data, utilizing this to make predictions on what comes next.

A major innovation in NLP is the transformer model, which has applications ranging from language translation and speech recognition to agents like Chatbots. These models employ techniques like the self-attention mechanism and multi-head attention to the importance of different parts on the input and process it in parallel instead of sequentially. This enhances the model's ability to understand context, therefore generating more accurate outputs [18].

Based on a focus on the speech-to-text aspect of NLP, that is the ability of AI models to transcribe speech gotten from audio to text; to help in achieving these, the Whisper model was employed. Whisper is an Open-AI model, which is a very modern speech recognition system capable of performing its functions in noisy conditions and with diverse languages. Whisper architecture implements an encoder-decoder transformer architecture; it treats as a sequence-to-sequence problem where an audio input is transformed into a log spectrogram, for the encoder to extract meaningful patterns from. Then it generates text from its decoder, using the attention mechanisms to pay attention to crucial parts of the recording [3].

Advancement in NLP has led to development of effective language models such as Generative Pre-trained Transformer (GPT) series from 1-4, these models have been implemented into chat-bots such as Claude, other chatbots such as Gemini, Claude, Perplexity, Jasper and others. They all implementing their own various models this innovation was driven by the transformer architecture [19]. GPT and other large language models (LLMs) use deep learning techniques to process and generate human-like text. It uses mechanisms like self-attention to comprehend the relationships between words in a phrase regardless of where they are located. The model requires lots of text data, learning patterns, context, grammar, and meaning in order to train and adjust the millions or even billions of parameters. By using the context of previous words to predict the next word in a sequence, iterative learning helps it get better over time. Following training, LLMs can generate logical writing, answer questions, or perform tasks using the knowledge and patterns they have found in the data [4].

2.3 Review of Related Works

The work in [20] used Node.js and Python to create a video chat website with a built-in notes controller. The built-in note-taking function is the unique characteristics of the video chat service. During a meeting, users can download the files to enhance information sharing and teamwork. To ensure everyone can access the key topics covered during the conversation, these files are synchronized in real-time for each participant. The website incorporates WebRTC (Web Real-Time Conversation) technology to facilitate real-time video conversation. The feature-rich video chat website that is produced will offer a collaborative and user-friendly platform for teamwork and remote communication.

Also, [21] featured a conferencing app that facilitate audio and video conferencing; screen sharing; and real-time messaging. In the study, they utilized WebRTC technologies and socket programming to develop a group video chat. It also included screen sharing and real-time chat. The app used Firebase's real-time database to store user data and chats, JQuery for the front end, and Node.js and Express.js for the signaling server. WebRTC enabled them to establish a peer-to-peer connection, and they were able to transport SDP packets and ice candidates with the aid of sockets.

[22] developed a secure video chat application based on the WebRTC standard. Various analyses were performed on different technologies, leading to analysis of the theoretical aspect of WebRTC. They tried to address the security flaw of WebRTC where users private IP was exposed even when users utilized a VPN (Virtual Private Network). To address this, they implemented an architecture that consists of several components, including a loop that uses an open socket connection to listen, transmit, and reproduce audio and media streams between system participants, as well as a message sent to the signal and STUN servers to gather the information required to make a video conversation.

Research in [23] implemented a video conferencing application utilizing WebRTC. For a connection to be proven, a detection procedure must occur called signaling. But there is no exact implementation or definition in WebRTC. The goal was to build an application with proper multi-browser support, i.e., Chrome and Firefox. As a result, a signaling channel for video conferencing and chat between two peers was created and executed utilizing the Socket.io mechanism, the Node.JS platform, and Express.JS. The primary programming languages used are JavaScript; Axios (HTTP JSON) for making requests; and Vanilla JS for the front end. As a result of this study, a signaling channel has been established and put into use.

2.3 Strength and Weakness of existing system

The developed application has the following strengths, as it:

- (i) provides basic functionalities for guest users, that is users without an account,
- (ii) supports a large number of participants for instance, 100 members for zoom, and google meet,
- (iii) offers real time-messaging capabilities,
- (iv) provides user with the ability to raise their hand. share their screens,

(v) offers the ability to set up meeting and having a messaging room (for example, as in teams).

Furthermore, the following are the Weaknesses:

- (i) they do not offer some needed features and if they did its usually behind a paywall.
- (ii) higher-quality video streams.
- (iii) a way to easily capture what was said during the meeting; an accurate transcript of the meeting.
- (iv) a digital whiteboard with real-time editing and collaboration capability.

2.4 Challenges and Considerations in Implementing an Artificial Intelligence-Driven Video Conferencing

Before starting the development of the application, a feasibility study was conducted to determine the practicality and viability of the project. The feasibility study assessed the technical, economic, and operational feasibility of the research.

Technical feasibility was assessed by evaluating the availability of resources and the technical skills required to build the application. The development team has experience with the technologies used in the project, including Fast-API, React.js, Redis, and utilizing and deploying AI models. The team also evaluated the feasibility of running various models depending on the number of concurrent calls and or meetings happening.

Economic feasibility was assessed by analyzing the costs associated with developing and maintaining the application. The development team used open-source technologies, which reduced the cost of development. The project requires ongoing maintenance and periodic updates, which would incur additional costs. Based on these factors, the project was deemed economically feasible.

Operational feasibility was assessed by evaluating the practicality of the application in real-world scenarios. The application was designed to be user-friendly and intuitive, making it accessible to a broad range of users. The transcription and summary functionality allows users to ascertain what was said during the meeting. The use of email communication provides an easy way to reach out to users with summary of the meeting. Based on these factors, the project was deemed operationally feasible.

The feasibility study showed that the project is viable and can be developed within the available resources. The development team has the required technical skills, and the infrastructure is in place to support the application. With the technical, economic, and operational feasibility established, the research was deemed feasible, and development proceeded.

3.0 METHODOLOGY

The methodology for the AI-driven video conferencing application involves several essential steps. The Agile software development model provides for flexibility as well as continuous improvement, and thus, a division of the work into several sprints making use of this model. The application has three aspects: the UI (front-end) and the back-end which will connect to the database, and lastly, the AI services that the application calls on.

3.1 Requirement Analysis

Requirement analysis is crucial to the creation of the application. The many features that were employed to accomplish the research goals and objectives are analyzed at this stage. Functional and non-functional criteria make up the system requirements.

Functional requirements are the features that should be implemented in the system. They describe what the system or software does, in Table 1 the functional requirements are shown with their priority to the system.

Table 1: Functional Requirements

S/N	System Requirement	Priority
1	The system shall enable users the ability to register	1-high
2	The system shall enable users to login.	1-high

3	The system shall enable users to create a meeting	1-high
4	The system shall enable users to schedule a meeting time	1-high
5	The system shall enable users to set the meeting private	1-high
6	The system shall send invites to invitees emails when they are part of private email.	1-high
7	The system shall enable users to join meetings by visiting the meeting link.	1-high
8	The system shall enable only invited members to join a private meeting.	1-high
9	The system shall enable meeting host remove a participant from a meeting.	2-medium
10	The system shall enable up to 10 participant in a meeting	1-high
11	The system shall enable participants to share their screens	2-medium

Non-functional requirements are the attributes and properties that represent the quality expectation of the software, in Table 2 the non- functional requirements are shown with their priority to the system.

Table 2 Non Functional Requirements

S/N	System Requirement	Priority
1	The system shall complete tasks asynchronously	1 – High
2	The system shall be able to scale to match the server workload.	1 – High
3	The system shall have the possibility to process multiple requests at once	1 – High
4	The system shall be available to users 24/7	1 – High
5	The system shall be secure meeting data by encrypting the content of the file.	1 – High
6	The system shall have a modern user interface	1 – High

3.2 System Design

The system requirements and the selected main functionalities served as the basis for the system architecture and database schema that was created. React.js was used for the front-end because it provides a faster and smoother user experience due to its small bundle size, reducing page load time and improves overall usability. While for the backend fast-api framework was employed because of its blazing fast performance and ease of development.

A use case serves as an example of a system functionality unit. The use-case diagram's primary goal is to assist development teams with visualizing a system's functional needs, including how "actors"—the people who will interact with the system relate to key procedures and how various use cases relate to one another. The use case diagram for our software solution in Figure 1.

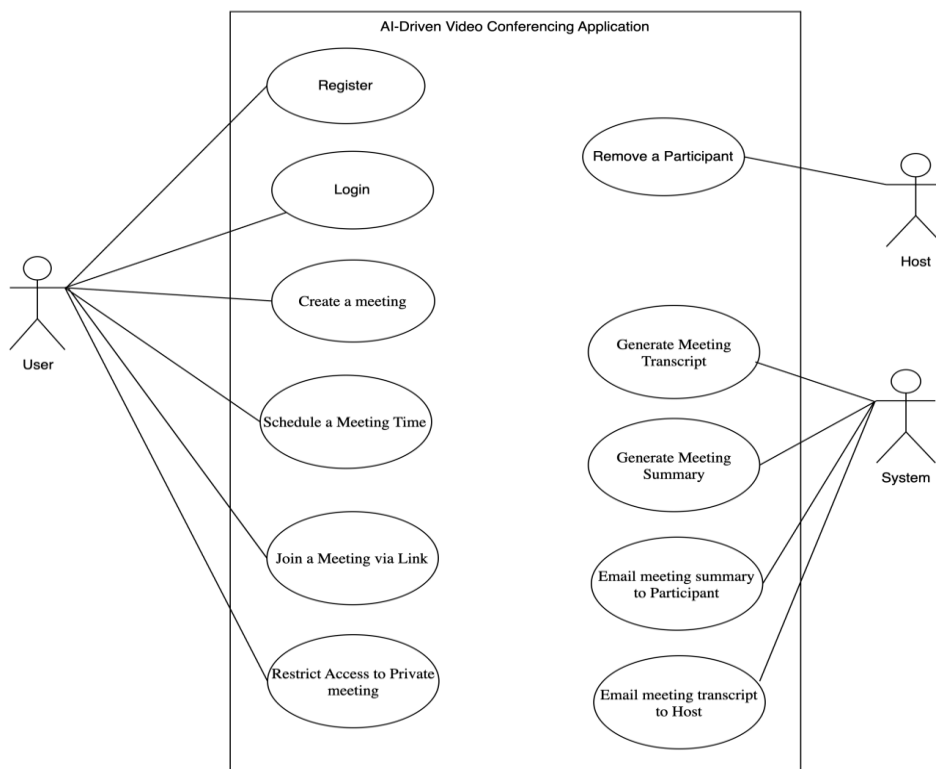


Figure 1: Use case Diagram for Artificial Intelligence-Driven Video Conferencing Application

The sequence diagram in Figure 2 illustrates the flow of interactions between different actors and components in the AI-driven video conferencing application for scheduling, hosting, and summarizing a meeting. The key actors and components include the Meeting Host, Participant, Application, Transcription Engine, and LLM (Large Language Model).



Figure 2: Sequence Diagram for Artificial Intelligence-Driven Video Conferencing Application

3.3 Database Design

An Entity-Relationship Diagram (ERD) is a visual representation of a database's structure, illustrating entities (tables), their attributes (columns), and the relationships between them. It helps in designing, analysing, and understanding the data model of a system. Figure 3 illustrates the Entity-Relationship Diagram (ERD) for a system managing users, rooms, room memberships, and meeting extracts. The diagram highlights the relationships between the entities and their attributes.

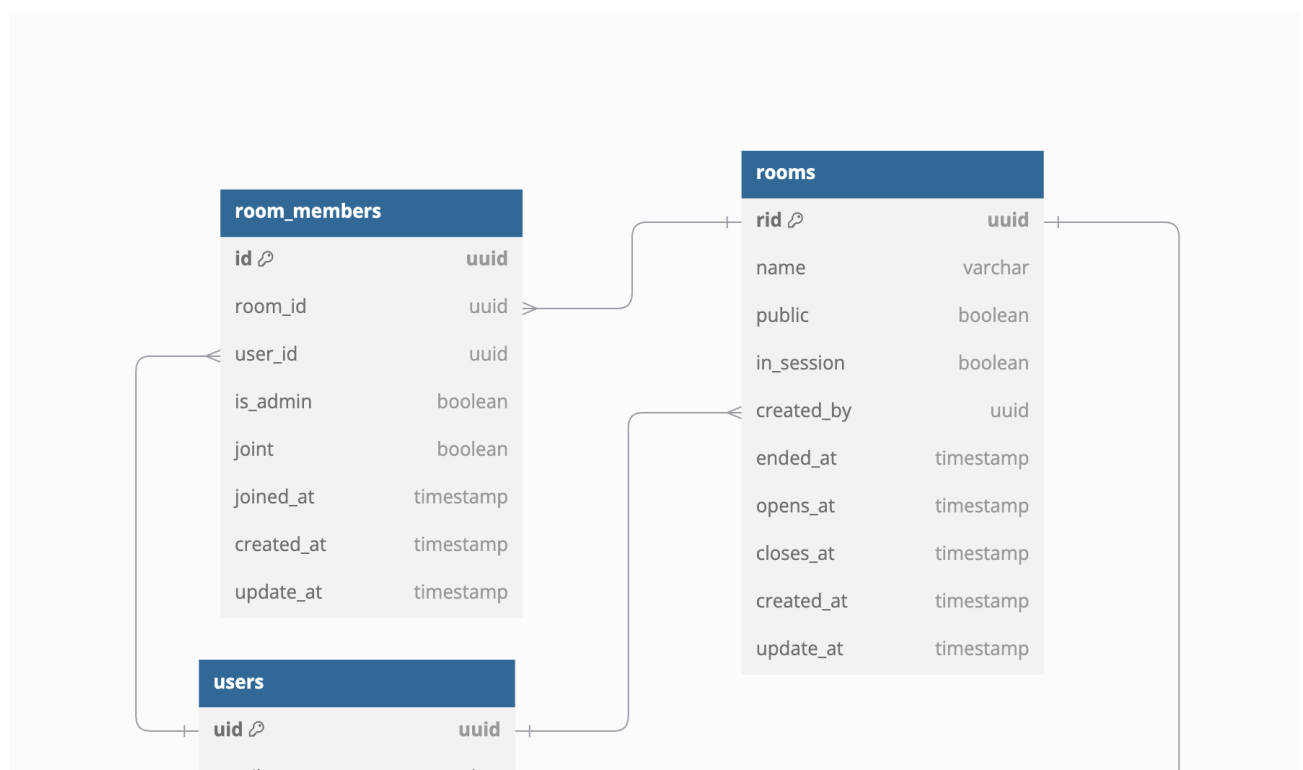


Figure 3: Entity Relationship Diagram for Artificial Intelligence-Driven Video Conferencing Application

4.0 SYSTEM DEVELOPMENT TOOL

In order to develop the proposed system, a number of development tools and technologies were utilized. These tools and technologies were chosen to ensure efficient development, smooth functionality, and a high-quality user experience. For the Implementation of the Artificial Intelligence-Driven Video Conferencing Application, the following tools were used- Python, Javascript, FastApi, React.js, Celery, Redis, PostgreSQL, AssemblyAI and StreamJS.

5.0 SYSTEM IMPLEMENTATION AND RESULT

The system implementation phase involved translating the system design and requirements into a functional and operational application using the FastAPI and React. The development process was broken down into modules and components, and thorough testing and deployment were performed to ensure the system was fully functional and met the requirements of the stakeholders. Below are some of the key pages on the Artificial Intelligence-Driven Video Conferencing Application.

Figure 4 is the dashboard page which displays an overview of Upcoming meeting, who created them whether it is a public or private meeting, how many known participant, and when it will hold, it also allows users search and filter meetings based on various fields.

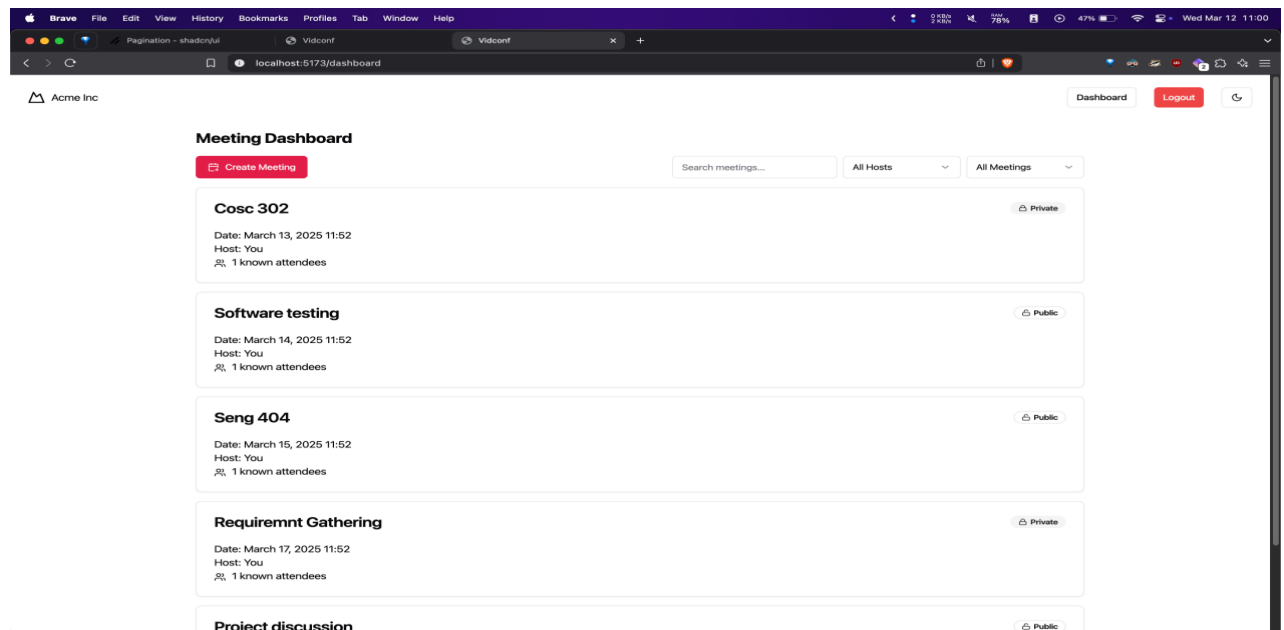
**Figure 4: Dashboard Page**

Figure 5 is the Create Meeting Page which allows users to create and schedule new meetings. It includes fields such as the title, the date, the publicity of the meeting and if private the list of allowed participants.

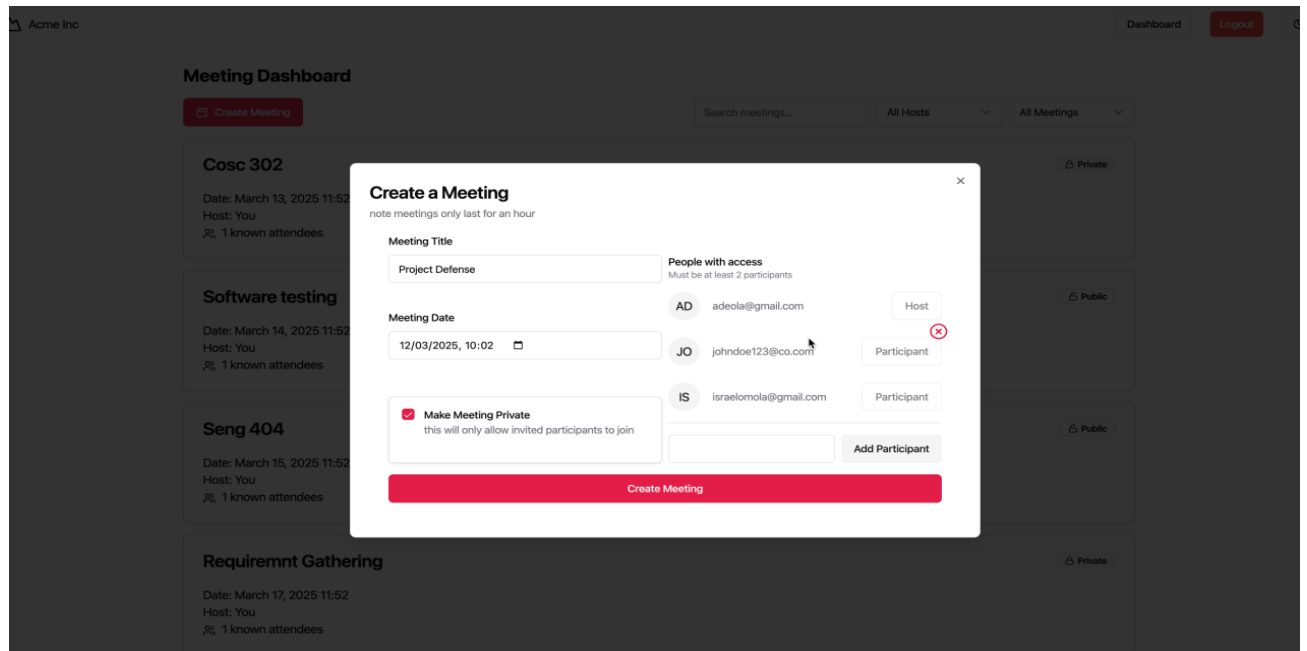


Figure 5: Create Meeting Page

Figure 6 shows the Transcription Page which displays real-time transcription when enabled using AssemblyAI, shown at the bottom of the screen.

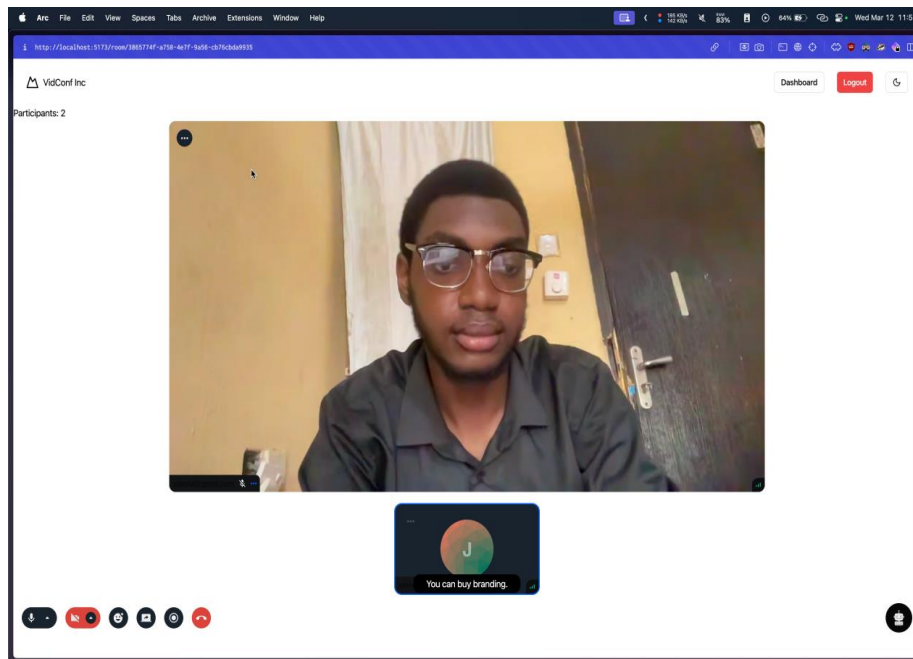


Figure 6: Transcription enabled

Figure 7 shows the AI Assistant Page which Integrates AI responses for user queries during meetings , all the user has to do is enable transcription and say GPT followed by the question which is then displayed to the user. There is a lamar that shows up when the AI assistant is called to indicate to the user that it is active.

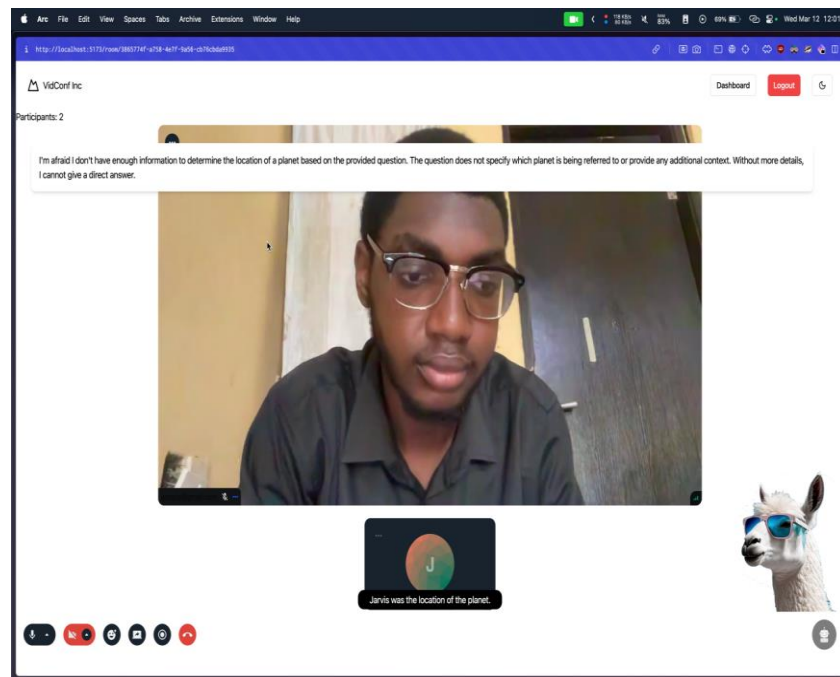


Figure 7: AI assistant Page

6.0 CONCLUSION

The system developed demonstrates a high degree of efficiency in handling messaging, notifications, and real-time communication. The combination of FastAPI, React, and other supporting technologies ensures smooth performance, scalability, and ease of maintenance. While the system currently meets its intended goals, there are opportunities for further optimizations, particularly in improving real-time data handling, reducing latency, and enhancing overall user experience.

7.0 RECOMMENDATION

To further enhance the system's functionality and reliability, several improvements can be made. Optimizing real-time data handling by refining WebRTC performance will help reduce latency and improve communication quality. Enhancing UI/UX by refining component interactions and improving accessibility will create a more user-friendly experience. Additionally, increasing system scalability by optimizing database queries and backend processing will ensure the platform remains efficient as user demand grows. Expanding integrations to include additional third-party services will also provide enhanced features, making the system more robust and adaptable to future needs.

REFERENCES

- [1] K. A. Karl, J. V. Peluchette, and N. Aghakhani, "Virtual work meetings during the COVID-19 pandemic: The good, bad, and ugly," *Small Group Research*, vol. 53, no. 3, pp. 343–365, May 2022, doi: <https://doi.org/10.1177/10464964211015286>.
- [2] H. Fateh, A. Khan, A. Akash, R. Avinash, and C. Lokesh, "WebRTC Peer to Peer Learning," Mar. 2018. Accessed: Oct. 05, 2024. [Online]. Available: <https://www.ijert.org/research/webrtc-peer-to-peer-learning-IJERTV7IS030135.pdf>
- [3] A. Radford, J. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," Dec. 2022. Available: <https://cdn.openai.com/papers/whisper.pdf>
- [4] H. Naveed et al., "A Comprehensive Overview of Large Language Models," arXiv.org, Aug. 18, 2023. <https://arxiv.org/abs/2307.06435>
- [5] José Joskowicz, "Video Conferencing Technologies: Past, Present and Future," Dec. 2023, doi: <https://doi.org/10.36227/techrxiv.24669051.v1>.
- [6] "Picturephone - Engineering and Technology History Wiki," ethw.org, Feb. 2014. <https://ethw.org/Picturephone>
- [7] "Videoconferencing | communications," Encyclopedia Britannica, Mar. 24, 2024. <https://www.britannica.com/technology/videoconferencing>

- [8] Global IP Solutions, "GIPS Codecs Unmatched VoIP Clarity." Accessed: Oct. 24, 2024.[Online].Available: <http://www.gipscorp.alcatrazconsulting.com/files/english/datasheets/Codecs.pdf>
- [9] "Google release of WebRTC source code from Harald Alvestrand on 2011-05-31 (public-webrtc@w3.org from May 2011)," lists.w3.org. <https://lists.w3.org/Archives/Public/public-webrtc/2011May/0022.html>
- [10] B. Dean, "Zoom User Stats: How Many People Use Zoom in 2021?," Backlinko, Feb. 13, 2024. <https://backlinko.com/zoom-users>
- [11] S. Balasubramanian, "Microsoft Teams Statistics - By Revenue, Demographics, Usage (2024)," Desk365, Aug. 29, 2024. <https://www.desk365.io/blog/microsoft-teams-statistics/>
- [12] Zoom, "Video Conferencing, Web Conferencing, Webinars, Screen Sharing," Zoom Video, 2024. <https://zoom.us/pricing>
- [13] "Compare Pricing and Plans for Microsoft Teams Online Options | Microsoft Teams," www.microsoft.com. <https://www.microsoft.com/en-us/microsoft-teams/compare-microsoft-teams-home-options>
- [14] I. Hickson, "WebRTC 1.0: Real-Time Communication Between Browsers," www.w3.org. <https://www.w3.org/TR/webrtc/> (accessed Oct. 19, 2024).
- [15] R. Verma, F. Services, and Jammu And Kashmir, "The Evolutionary Landscape of Artificial Intelligence," International Journal of Multidisciplinary Research and Growth Evaluation, vol. 05, no. 05, pp. 582–585, Oct. 2024, Accessed: Oct. 19, 2024. [Online]. Available: https://www.researchgate.net/publication/384599786The_Evolutionary_Landscape_of_Artificial_Intelligence
- [16] Coursera Staff, "What is Artificial Intelligence? Definition, Uses, and Types," Coursera, Apr. 03, 2024. <https://www.coursera.org/articles/what-is-artificial-intelligence>
- [17] Mohana Murugan, "Natural Language Processing (NLP)," Apr. 03, 2024. https://www.researchgate.net/publication/379513103Natural_Language_Processing_NLP (accessed Oct. 19, 2024).
- [18] T. Bell and T. Olavsrud, "What is NLP? Natural language processing explained," CIO, Aug. 11, 2023. <https://www.cio.com/article/228501/natural-language-processing-nlp-explained.html>
- [19] Y. Liu et al., "Summary of ChatGPT-Related Research and Perspective Towards the Future of Large Language Models," Meta-Radiology, vol. 1, no. 2, Apr. 2023, doi: <https://doi.org/10.1016/j.metrad.2023.100017>.
- [20] S. D. Souza, U. R., and C. Fong Kim, "VCVERSE – A Video Conferencing Website with Controls," INTI Journal, vol. 2024, no. 1, Jul. 2024, doi: <https://doi.org/10.61453/intij.202412>.
- [21] S. Majhi, "Video Conferencing WebApp," International Journal for Research in Applied Science and Engineering Technology, vol. 9, no. 12, pp. 413–416, Dec. 2021, doi: <https://doi.org/10.22214/ijraset.2021.39277>.
- [22] E. Revyakina, "Development of a secure video chat based on the WebRTC standard for video conferencing," E3S Web of Conferences, vol. 389, p. 07017, 2023, doi: <https://doi.org/10.1051/e3sconf/202338907017>.
- [23] M. A. Saeed and Naktal Moaid Edan, "Design and Implementation of Peer-to-Peer Video and Chat Communication," Nov. 11, 2022. https://www.researchgate.net/publication/365319298Design_and_Implementation_of_Peer-to-Peer_Video_and_Chat_Communication (accessed Oct. 23, 2024).