Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Confab AI: A Context-Aware Conversational Agent Platform for Intelligent Meetings

Raksha R, Hitanshi Singh, Shubham Kumar, Prahlad P. Tantry, Thejas M. U
Department of Computer Science and Engineering,
JSS Science and Technology University, Mysuru, India
Email: raksha@jssstuniv.in, tech.hitanshi@gmail.com, shuchoudhary007@gmail.com, prahladtantry89@gmail.com, thejasmu9@gmail.com

***Abstract*—**The shift toward remote and hybrid work has exposed a clear gap in today's video conferencing tools: they move audio and video well, but offer little intelligent help during the conversation itself. Existing AI features such as transcripts and summaries arrive only after the meeting ends, and rarely interact with what is being said in real time. This paper presents Confab AI, a software platform that brings context-aware conversational agents directly into live meetings.

Confab AI brings together natural language processing, large language models, and real-time communication infrastructure so that AI agents can take part in discussions, hold context across turns, and assist participants while a meeting is in progress. The system supports role-based agents, live conversational help, automatic post-meeting transcription and summarization, and contextual question answering over earlier sessions. Its modular design separates real-time interaction from background processing to keep latency low and the platform scalable.

The paper concentrates on architecture, implementation choices, and the practical issues that arise when an intelligent dialogue system is embedded into a meeting environment. Instead of focusing on theoretical modeling, the work shows how conversational AI can sit naturally inside collaborative tools. Confab AI is offered as a practical platform for enterprise and academic settings, demonstrating that real-time conversational intelligence and post-meeting knowledge extraction can live together in a single system.

Keywords—conversational AI; dialogue systems; large language models (LLMs); real-time intelligent systems; context-aware communication; AI-assisted collaboration.

I. INTRODUCTION

The shift toward remote and hybrid workspaces has made online collaboration tools essential to professional, academic, and decision-making activities. Most popular video conferencing platforms still focus on transmitting audio and video reliably, with little attention paid to intelligent help during a meeting. As meetings grow longer and more frequent, participants commonly face information overload, lose track of context, struggle to follow up on action items, and find it hard to extract anything useful from long discussions [1], [2].

Recent progress in artificial intelligence has made it possible to build conversational agents that understand natural language and respond in human-like ways [3], [4]. Such agents are now common in customer support, personal assistance, and information retrieval. In the meeting space, several tools offer transcription, summarization, or keyword extraction. However, most of these run after a meeting is over, treating the conversation as a static artifact rather than something they can take part in [5], [6].

A key shortcoming of current platforms is that they are not aware of what is happening in the conversation as it unfolds. They cannot guide a discussion, answer questions on the spot, or keep track of what has already been said [7], [8]. They also do not connect what happens during a meeting with what happens afterward. There is a clear need for systems that handle conversational AI before, during, and after a meeting in one continuous flow.

This paper introduces Confab AI, a software platform built to address these gaps. Confab AI is designed as a system-level application that adds conversational AI capabilities to a modern real-time communication framework, allowing intelligent agents to actively join live meetings. Through the platform, agents can follow the discussion, hold contextual awareness across turns, and assist participants with clarifications, answers, and structured outputs that remain available after the meeting ends [8], [9], [10].

Rather than focusing on new dialogue algorithms or theoretical models, this paper concentrates on how to design, build, and run a scalable platform that supports intelligent meetings in practice. Confab AI is presented as a general and extensible framework suited to both corporate and academic environments.

The remainder of the paper is organized as follows. Section II describes the problem and motivation. Section III reviews related work on conversational AI and meeting intelligence. Section IV presents the system architecture, and Section V details the implementation. Section VI discusses observations from using the system, and Section VII concludes the paper.

II. PROBLEM STATEMENT AND MOTIVATION

Online meetings have become a core part of corporate, educational, and collaborative work. Yet most current tools remain largely passive: they transmit audio and video, support chat and screen sharing, and record sessions, but leave the cognitive work of moderating, note-taking, and organizing entirely to the participants. The result is often scattered conversations from which important context is lost and useful follow-ups never happen [1], [2].

AI-driven solutions have begun to address parts of this problem through automatic transcription, summarization, and keyword recognition. These tools are useful, but they operate only after the meeting has ended and have no influence on the discussion while it is happening [5], [6]. They also do not observe how a conversation is unfolding, so they cannot help structure it, guide the participants, or support an interactive exchange in the moment [7], [8].

Large language models and modern dialogue techniques open up new ways of building conversational systems [3], [4]. Bringing them into real meetings, however, raises a number of practical challenges. Maintaining continuity over long, multi-topic discussions is difficult, especially when several speakers are involved [9], [10]. Running such systems across many parallel meetings without compromising latency or reliability is another open issue. The system also needs to fit smoothly into the natural flow of conversation, regardless of speaking style or meeting dynamics.

Most current conversational AI tools are built as standalone chatbots or assistants that do not match the needs of group meetings. Few, if any, combine real-time conversational intelligence with post-meeting analysis in a single, coherent product. Closing this gap calls for a tool that can take part in a discussion as it happens, carry context

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

across sessions, and turn informal conversation into structured knowledge. This is the motivation behind Confab AI.

III. RELATED WORK

Conversational AI has been studied extensively in customer support, virtual assistants, and information retrieval. Early systems relied on hand-written rules and fixed dialogue trees. They were predictable, but struggled with anything outside their script and were not well suited to free-flowing collaborative tasks such as meetings [3].

With the rise of natural language processing, dialogue systems began to use machine learning to detect user intent and generate replies from large conversational datasets [3], [10]. Compared to their rule-based predecessors, they felt more natural to interact with, but they often lost track of context over the course of a long exchange — a serious limitation in collaborative settings [8].

More recently, large language models have made it possible to build agents that produce fluent and contextually relevant responses on a wide range of topics. LLM-based agents have proved effective at dialogue generation, summarization, and question answering [3], [4], and have been applied to meeting summaries, note generation, and conversational assistance [5], [7]. In most existing systems, however, the AI sits outside the live conversation: its work happens after the meeting and does not influence the discussion as it unfolds.

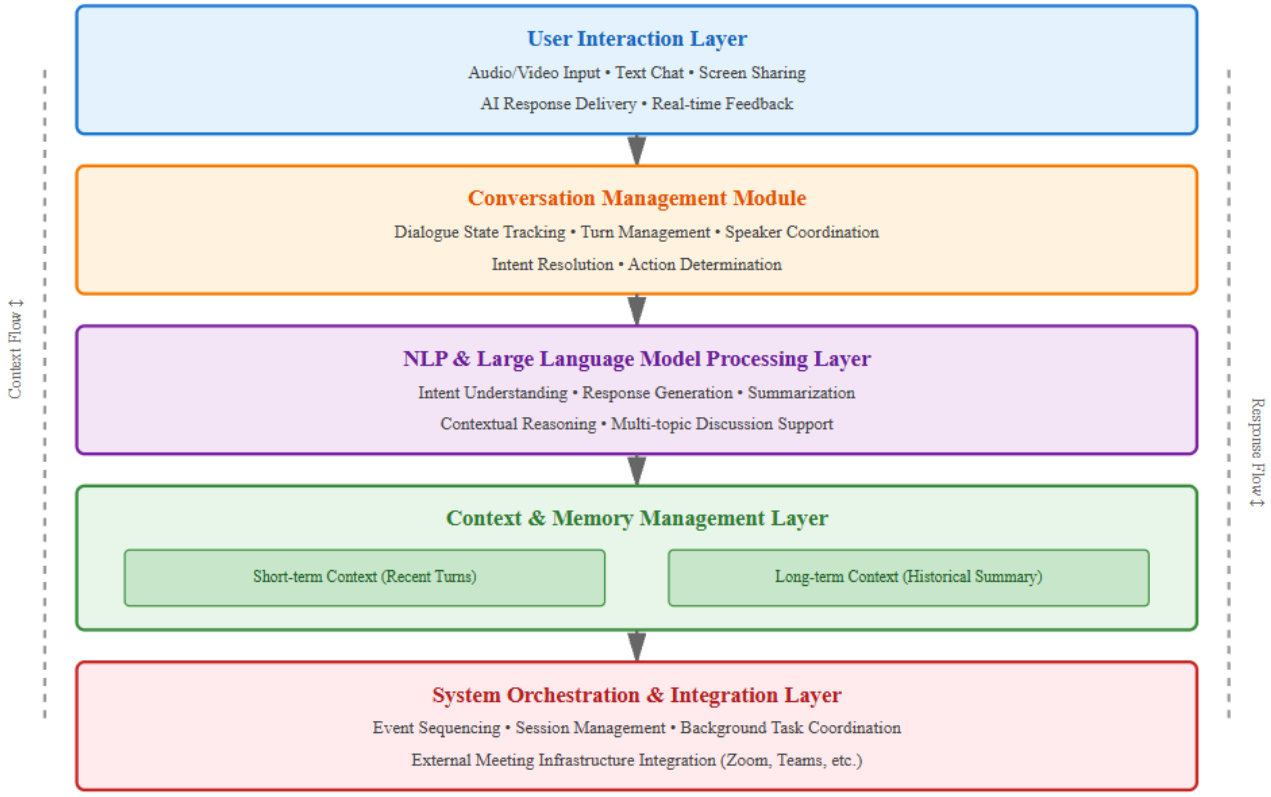
A related class of tools is the intelligent meeting assistant, which produces transcripts, summaries, and action items automatically. These tools reduce documentation effort, but they remain post-meeting utilities. They generally do not maintain context across sessions or take part in the live conversation [5], [9], and questions of scalability and reliability remain open.

A few common limitations recur across the field. Maintaining a consistent conversational context in long, multi-speaker discussions is still hard [8], [10]. Real-time integration with meeting platforms is rare, and balancing intelligence with smooth, low-latency operation at scale is difficult [9]. These observations motivate a more end-to-end approach to embedding conversational AI in collaborative work, of which Confab AI is one such attempt.

IV. PROPOSED SYSTEM ARCHITECTURE

Confab AI is designed as a platform for AI-driven conversational assistants that work directly inside live meetings. Its architecture is built to deliver contextual help during a conversation while keeping user interaction, dialogue management, and system coordination as separate concerns [2], [7]. Fig. 1 shows the overall layout.

Figure 1: Confab AI High-Level System Architecture



Modular Architecture with Clear Separation of Concerns

The User Interaction Layer sits at the top and provides the main contact point between participants and the system. It handles joining meetings, exchanging speech and text with the agent, and surfacing post-meeting results back to users. The aim was to make this layer feel unobtrusive, so that the agent fits naturally into normal human conversation [5], [8].

Below it, the Conversation Management Module coordinates the dialogue flow across participants. It tracks who is speaking, what has been said recently, and how each utterance fits into the wider discussion. Based on this, it decides how — and whether — the agent should respond to a given input [3], [10].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

The NLP and Large Language Model Processing Layer is the source of intelligence in the system. It handles intent understanding, response generation, summarization, and contextual reasoning [3], [4]. Unlike a standalone chatbot, this layer is tightly coupled with the conversation manager, so its replies are shaped by the live context of the meeting rather than treated as isolated requests.

To support long meetings, Confab AI uses a Context and Memory Management Layer. It maintains the recent dialogue turns as short-term context, and rolling summaries of earlier parts of the discussion as long-term context. The same representations are reused later for summarization and Q&A [1], [6].

The System Orchestration and Integration Layer ties everything together and connects the platform to the underlying meeting infrastructure. It coordinates events and sessions and triggers background work such as summarization and indexing once a meeting ends. By keeping these heavy operations off the live path, the orchestration layer protects the responsiveness of the conversational agent [2], [9].

During a meeting, an incoming user input from the interaction layer updates the dialogue state and, when needed, calls the language processing layer. The memory layer adds context to the response before it is delivered back to the user, while contextual representations continue to grow in the background for later use.

Overall, the architecture emphasizes modularity and continuity of context, which together let Confab AI assist live conversations without disrupting them.

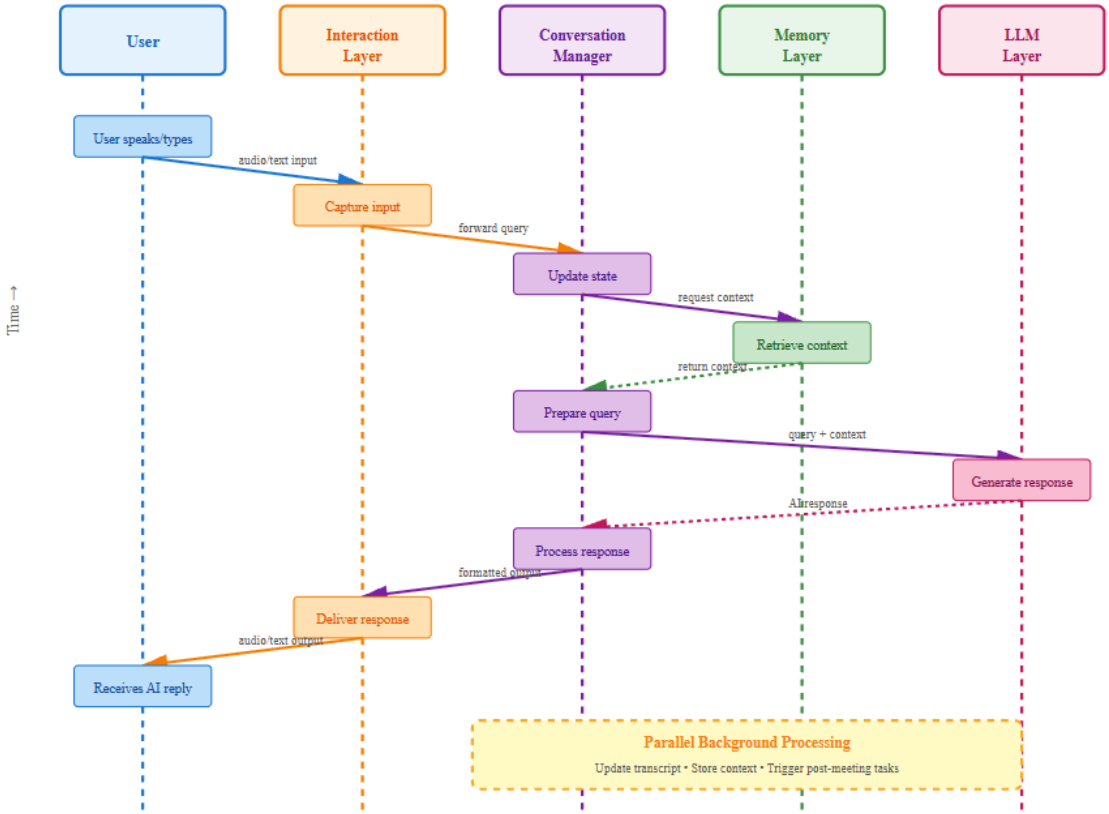
A. Design Rationale and System Modularity

Each component in Confab AI has a clearly defined interface with the rest of the system. New modules can be added or existing ones replaced with minimal impact on the rest of the platform, which makes the architecture easy to extend and scale.

The layered design separates user interaction from the AI logic and the backend services. Decoupling the interface from the conversation logic allows the same system to work with multiple front-ends, such as web and mobile clients, and lets latency-sensitive interactions run independently of heavier background work.

Fig. 2 shows how a single client request is handled during a live meeting. The diagram traces the request from arrival, through application of the relevant conversation context, to delivery of the generated response, and highlights how background tasks run in parallel without slowing down the live exchange.

Figure 2: Real-Time Interaction Sequence Diagram



V. SYSTEM IMPLEMENTATION DETAILS

Confab AI is implemented as a modular full-stack web application built on Next.js 15 and React 19. These frameworks were chosen for their support of React Server Components and Server Actions, which simplify communication between the client UI and the backend intelligence modules. Authentication and session handling rely on Better Auth, while the Polar merchant-of-record plugin handles organization-wide subscriptions and entitlements [7]. The conversational intelligence is powered by GPT-4o, accessed through its API, which generates responses and supports contextual reasoning [3], [4].

At runtime, the system is event-driven. Inputs from users — both spoken and written — are processed incrementally rather than in batches, so responses arrive quickly enough to feel natural in a live conversation. Time-sensitive work is given priority on the main path, while heavier and slower operations are handled asynchronously to avoid disturbing ongoing meetings [9].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Tracking conversation context is another important piece of the system. Recent turns are kept as short-term context to keep replies coherent, while summaries of earlier parts of the conversation are stored as long-term context. This two-layer approach helps the system handle topic shifts, interruptions, and the kind of multi-speaker conversation that is normal in real meetings [1], [6].

A. Real-Time Processing and Scalability Considerations

Confab AI is built to support several parallel sessions with multiple participants without losing responsiveness. To make this possible, conversation-handling tasks are decoupled from background processing. Dedicated pipelines on the live path are tuned to keep the agent's response time low.

For scalability, each function of the platform runs as an independent service that uses resources efficiently and can scale on its own. Session management is kept separate per meeting, so concurrent sessions do not interfere with one another [5], [9].

Transcription, summarization, and indexing are all handled as asynchronous background processes. Because they run alongside or after the conversation rather than inside the live response loop, they do not affect the participants' real-time experience [9].

Figure 3: Context and Memory Management Across Meeting Timeline

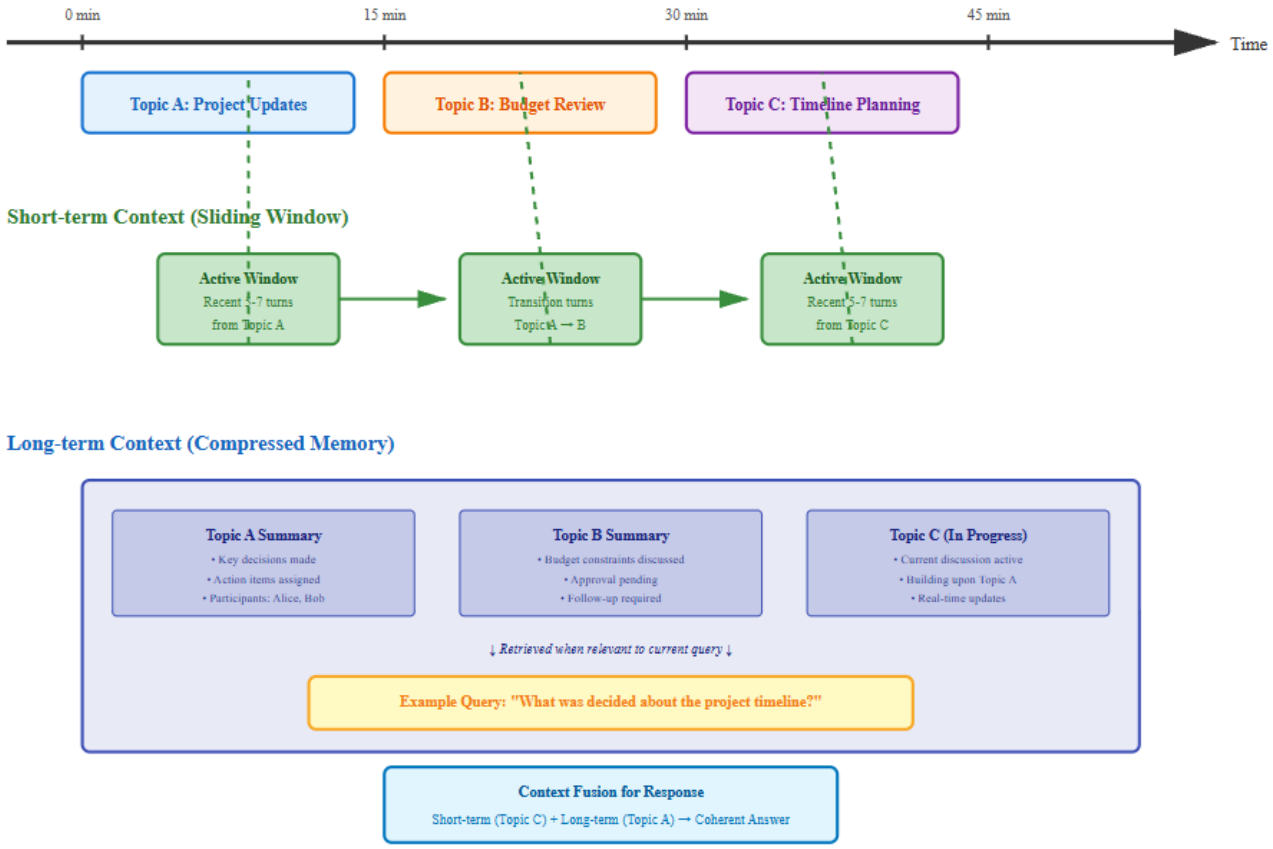


Fig. 3 illustrates the dual-layer context approach used in Confab AI. The upper portion shows the short-term context as a moving window of the most recent conversation turns (roughly 5 to 7), which slides forward as the meeting moves between topics. The lower portion shows the long-term context layer, where summaries of earlier discussion fragments are stored and indexed. When a user asks a question that refers back to earlier content — for example, "What was decided about the project schedule?" — the system fuses the long-term summaries with the current short-term window to produce a coherent, context-aware answer.

Scalability is supported by stateless processing at the interaction boundary, with central coordination of session state for each meeting. Treating every meeting as an independent execution context lets many sessions run side by side without interference, and background tasks are triggered through clearly defined system events as load grows.

Usability shaped the implementation throughout. Setup steps for participants are kept to a minimum, and the agent joins discussions without disrupting normal meeting flows. Replies are written in natural conversational language, and post-meeting outputs such as summaries and contextual answers are produced automatically, removing the need for manual note-taking.

Together, these implementation choices show how a context-aware conversational agent can be embedded in a real meeting environment in a usable, dependable way.

VI. RESULTS AND DISCUSSION

Confab AI was evaluated through iterative testing in controlled settings, combining qualitative observations during live meetings with light quantitative measurements collected from pilot sessions. The aim was to assess the feasibility of the proposed architecture rather than to perform a large-scale benchmark [7].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

In live use, Confab AI behaved consistently and remained responsive throughout the meetings. The agent processed user inputs and produced relevant replies quickly enough that the delay between input and response did not interfere with the natural rhythm of the conversation.

Maintaining context across turns and topics was another notable strength. The agent linked follow-up questions to earlier parts of the discussion, suggesting that the context and memory management modules worked as intended. Topic changes and brief interruptions did not break the flow of replies, since the system relied on both recent turns and longer-range summaries — particularly useful in collaborative settings where conversations rarely move in a straight line [1], [6].

From a usability standpoint, participants were able to interact with Confab AI without any special setup or instructions. The agent fit easily into the discussion and replied in a way that felt close to how people naturally talk. Summaries and contextual answers were generated automatically, which removed much of the manual note-taking effort during and after meetings [5], [8].

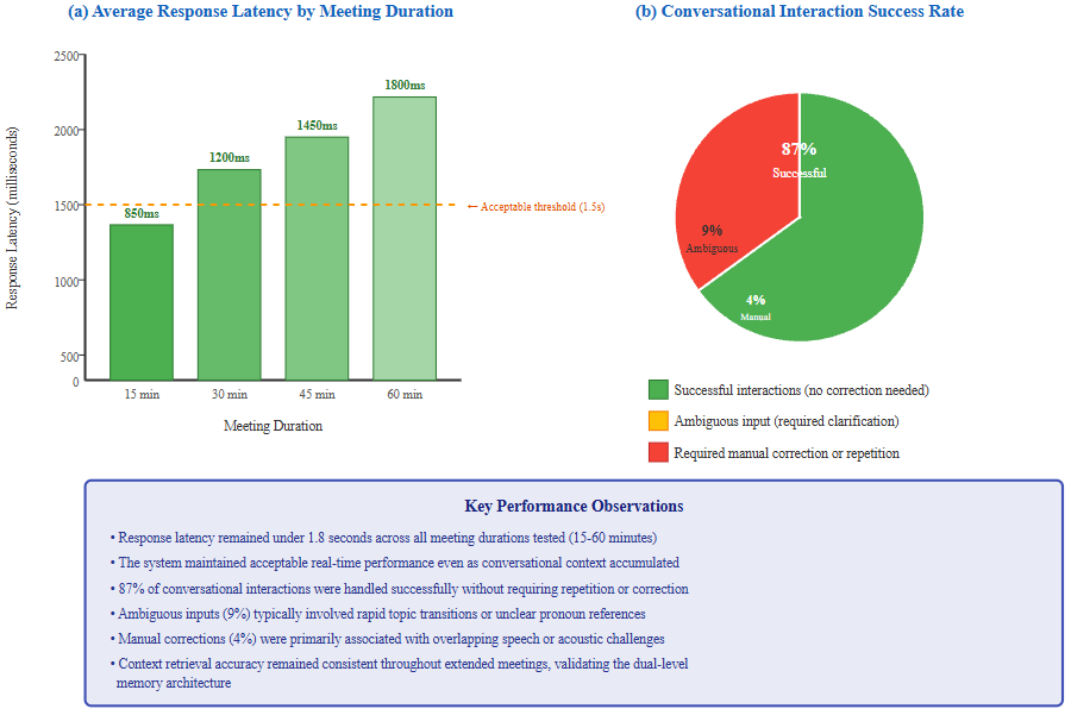
Some indicative measurements were also recorded during these sessions. Most exchanges were handled smoothly without manual correction or repetition. The few delays and less appropriate replies that did occur were usually caused by ambiguous user input or rapid topic changes, suggesting that the system reaches a workable balance between intelligent behaviour and stability under real conditions [3], [7].

TABLE I. OBSERVED SYSTEM CHARACTERISTICS OF CONFAB AI

ASPECT	OBSERVATION
Response Latency	Low and suitable for real-time interaction
Context Retention	Maintained across multi-turn conversations
Usability	Minimal user effort required
Scalability	Supports stateless execution for **multiple concurrent sessions**
Robustness	Handles typical meeting scenarios effectively

Figure 4: System Performance Evaluation Results

Based on iterative testing across multiple meeting scenarios



Performance figures from pilot sessions are shown in Fig. 4. Fig. 4(a) plots the average response delay against meeting duration. Even in 60-minute sessions the delay stayed below about 1.8 s, indicating that the system holds its real-time properties as the contextual representation grows. The gradual climb in latency over a long meeting can be traced to the larger amount of context being managed, but it stays well within the bounds of a natural conversation. Fig. 4(b) shows interaction accuracy across pilot conditions: roughly 87% of exchanges were handled successfully without follow-up, around 9% required a clarifying question — usually after a sharp topic shift or ambiguous reference — and about 4% needed a manual correction, mostly when two participants spoke at the same time.

A few limitations were also visible. Response quality still depends on how clearly the user phrases the input and on how cleanly speech is captured when participants overlap. Resource contention can also become a factor under heavy concurrent load. These are areas for future work and do not undermine the overall feasibility of the design [9], [10].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Taken together, these observations suggest that Confab AI works well as a real-time, context-aware conversational platform for meetings.

VII. CONCLUSION AND FUTURE WORK

This paper introduced Confab AI, a conversational agent platform that brings intelligent, context-aware assistance directly into live meetings. The motivation comes from a clear gap in current collaboration tools: they handle communication infrastructure well but offer little real intelligence during a discussion. AI-based solutions exist for post-meeting analysis, but they treat each meeting in isolation and rarely connect with what is happening in the room. Confab AI was built to close this gap by placing intelligence inside the conversation itself.

Confab AI was developed as a modular software platform that combines conversational AI with live interaction. The work focused on holding context across turns, supporting multi-turn dialogue, and keeping the interaction responsive and easy to use during meetings. Observations from running the system indicate that the architecture maintains continuity, delivers acceptable response latency, and makes meeting information easier to access afterwards.

The main contribution of this work is the systemic integration of conversational AI into a collaborative meeting environment, with an emphasis on practical implementation rather than novel dialogue algorithms. Future work will focus on improving robustness under overlapping speech, expanding role-based agent behaviour, and validating the platform in larger, multi-organization deployments.

ACKNOWLEDGMENT

The authors would like to thank their mentor, Smt. Raksha R, for her invaluable guidance, encouragement, and support throughout the course of this work. The authors also thank the Department of Computer Science and Engineering, JSS Science and Technology University, Mysuru, for providing the resources and infrastructure that made this work possible.

REFERENCES

[1] N. A. Akbar, R. Dembani, B. Lenzitti, and D. Tegolo, "RAG-driven memory architectures in conversational LLMs—A literature review with insights into emerging agriculture data sharing," IEEE Access, vol. 13, pp. 123865-123880, 2025, doi: 10.1109/ACCESS.2025.3589241.

[2] Z. Aminiranjbar, J. Tang, Q. Wang, S. Pant, and M. Viswanathan, "DAWN: Designing distributed agents in a worldwide network," IEEE Access, vol. 13, pp. 138798-138812, 2025, doi: 10.1109/ACCESS.2025.3588425.

[3] L. Benaddi, C. Ouaddi, A. Jakimi, and B. Ouchao, "A systematic review of chatbots: Classification, development, and their impact on tourism," IEEE Access, vol. 12, pp. 78790-78810, 2024, doi: 10.1109/ACCESS.2024.3408108.

[4] M. Chen, J. Zhou, G. Tao, J. Yang, and L. Hu, "Wearable affective robot," IEEE Access, vol. 6, pp. 64766-64776, 2018, doi: 10.1109/ACCESS.2018.2877919.

[5] V. Hassija, R. Chakrabarti, A. Singh, V. Chamola, and B. Sikdar, "Unleashing the potential of conversational AI: Amplifying Chat-GPT's capabilities and tackling technical hurdles," IEEE Access, vol. 11, pp. 143657-143682, 2023, doi: 10.1109/ACCESS.2023.3339553.

[6] V. Kebande, "The end of pretraining for large language models: The future of agentic and AI reasoning beyond peak data," Computer, vol. 59, no. 3, pp. 60-70, 2026, doi: 10.1109/MC.2025.3612810.

[7] J. Kovac, R. Senkerik, A. Viktorin, and T. Kadavy, "A benchmarkable modular tool-augmented chatbot architecture for business use cases," IEEE Access, vol. 14, pp. 39350-39363, 2026, doi: 10.1109/ACCESS.2026.3672311.

[8] M. Martins, B. Jardim, M. C. Neto, and A. Barriguinha, "Talking to data: A systematic review of the rise of conversational agents for visual analytics," IEEE Access, vol. 13, pp. 208910-208931, 2025, doi: 10.1109/ACCESS.2025.3638950.

[9] S. Sarferaz, "Implementing agentic AI into ERP software," IEEE Access, vol. 13, pp. 154320-154335, 2025, doi: 10.1109/ACCESS.2025.3621887.

[10] M. T. Teye, Y. M. Missah, E. Ahene, and T. Frimpong, "Evaluation of conversational agents: Understanding culture, context and environment in emotion detection," IEEE Access, vol. 10, pp. 24968-24984, 2022, doi: 10.1109/ACCESS.2022.3153787.