AI-POWERED VIRTUAL MEETING PLATFORM WITH SMART ANALYSIS USING ARTIFICIAL INTELLIGENCE.

Prof. Sagar Mane[1], Aarya Erande[2], Vansh Katkamwar[3], Suhani Meshram[4], Ashutosh Markad[5]
[1]*Guide, NBN Sinhgad Technical Institutes Campus, Pune, India*
[2345] *Student, NBN Sinhgad Technical Institutes Campus, Pune, India*

---***---

Abstract-*Modern virtual conferencing platforms increasingly rely on client-side computer vision and machine learning models to deliver features such as virtual backgrounds, live gesture recognition, and voice-assisted controls. However, running multiple independent ML models alongside real-time WebRTC communication pipelines inside a browser sandboxed environment introduces significant performance bottlenecks. Duplicate media stream acquisitions and concurrent video decoding loops consume excessive CPU and GPU resources, leading to thermal throttling, battery drain, and frame-rate degradation. To address these challenges, we present Meet-flow, a resource-optimized, edge-coordinated virtual collaboration framework. Meet-flow introduces a centralized client-side MediaEngine singleton that orchestrates hardware capture devices through strict idempotent stream acquisition and concurrency guards. By utilizing a single-instance RequestAnimationFrame (RAF) broadcast loop and a shared video element, the MediaEngine distributes synchronized frames to multiple downstream subscriber modules (including MediaPipe Selfie Segmentation, CDN-driven MediaPipe Hands for gesture detection, and a Web Speech API listener). Furthermore, to ensure consistent performance on resource-constrained hardware, Meet-flow implements a dynamic CPU profiling feedback loop. This loop monitors actual client frame rates and automatically scales down video segmentation model complexity (e.g., swapping from landscape to general segmentation models) if the rendering rate drops below 24 frames per second. On the server side, Meet-flow integrates an automated AI assistance pipeline that downloads video recordings, extracts audio streams via a static FFmpeg runtime, and employs an adaptive compression and splitting algorithm to transcribe audio through OpenAI Whisper within API size limits. Finally, the meeting workspace, waiting rooms, and chat features are secured via PostgreSQL Row-Level Security (RLS) policies matching real-time SFU workflows. Our evaluation demonstrates that edge coordination drastically reduces consumption and guarantees stable, real-time interactive frame rates across varying hardware profiles.*

Key Words: Virtual Meeting Platform, Artificial Intelligence, Meeting Summarization, Speech-to-Text Transcription, Retrieval-Augmented Generation, Gesture Recognition, Engagement Analytics, Real-Time Collaboration.

1. INTRODUCTION

The rapid acceleration of hybrid work cultures and remote education has established virtual meeting platforms as indispensable infrastructure for global collaboration. Traditional Real-Time Communication (RTC) systems, built primarily using standard WebRTC protocols, have successfully bridged geographical gaps but have also introduced new socio-technical challenges. As virtual interaction becomes a permanent fixture of professional life, issues such as "Zoom fatigue," high bandwidth consumption, camera anxiety, and post-meeting administrative overload have emerged as significant barriers to productivity and user well-being. Furthermore, extracting actionable insights, decisions, and task assignments from raw video recordings remains a manual, time-consuming task prone to human error and cognitive overload. To address these limitations, this paper presents the design and implementation of Meet-flow, an intelligent, next-generation virtual meeting platform built on Next.js, Supabase, and the Stream Video SDK. Meet-flow moves beyond simple audio-video streaming by integrating real-time computer vision, edge-based audio-visual processing, and generative artificial intelligence directly into the communication pipeline. The system introduces several key architectural innovations.

Privacy-Preserving Vector Avatars: Rather than forcing participants to transmit live video feeds, Meet-flow provides a dynamic, client-side vector avatar editor. The system renders dynamic, responsive SVG/Canvas avatars that map to user identity profiles and synchronize with the Clerk authentication layer, offering a highly lightweight, low-bandwidth, and privacy-preserving alternative to camera feeds. Edge-Based Computer Vision & Gesture Recognition: Utilizing Media Pipe Hands and Selfie Segmentation via client-side Web Assembly (WASM), Meet-flow detects physical gestures (such as hand raises, claps, thumbs-up, and thumbs-down) directly in the browser to trigger non-verbal reactions. To guarantee performance on low-compute devices, the system incorporates an automatic CPU Profiler that dynamically downgrades segmentation model complexity (switching from landscape to general models) if frame rates drop below 24 FPS. Objective Engagement Analytics: To quantify participant interaction objectively, Meet-flow implements a real-time "Meeting Energy" algorithm. The system continuously samples normalized room audio levels and active speaker counts, aggregating these metrics via a secure API into a Supabase database to visualize engagement patterns over time using responsive area charts. Reliable Transcription & Summarization Pipeline: Post-meeting workflows are automated via a multi-stage server-side pipeline. Meeting recordings are processed using FFmpeg to extract, compress and chunk audio files exceeding the 20MB limit before transcribing them with the OpenAI Whisper API. The resulting transcripts are structured by Google Gemini models to generate JSON-formatted summaries, action items, and decision logs, which feed into an interactive, retrieval-augmented (RAG) meeting QA chat bot.

1.1 Motivation:

Traditional video meetings often induce cognitive fatigue and raise privacy concerns, as participants are forced to keep their cameras active. Meet-flow addresses this by providing client-side customized vector avatars (rendered dynamically using SVG and Canvas) as a lightweight, privacy-preserving alternative. Furthermore, the high overhead of manual note-taking and the need for objective participant engagement analytics motivate the integration of automated speech-to-text pipelines (via OpenAI Whisper) and generative AI (via Google Gemini) to capture decisions, generate summaries, and calculate real-time "Meeting Energy" metrics based on normalized audio levels.

1.2 Problem statement:

Current virtual meeting platforms demand high network bandwidth, offer no native method for objectively measuring participant attention or room energy levels, and lack integrated, reliable mechanisms to automatically transcribe meetings and extract structured, actionable summaries. Consequently, remote teams suffer from decreased productivity due to undocumented decisions, high cognitive exhaustion from continuous camera exposure, and low interactivity during collaborative sessions.

2. LITERATURE REVIEW

2.1 Virtual Meeting Platform:

The architecture of real-time communication systems has evolved significantly, yet balancing network bandwidth, user privacy, and cognitive engagement remains a challenging area of research. Martinez and Lopez (2022) proposed a bandwidth-efficient WebRTC framework designed to address media stream degradation in low-connectivity environments. Their methodology involved an adaptive encoder configuration that dynamically scaled media codecs and bitrates based on network packet loss and jitter metrics. To minimize data transmission, they introduced a system that replaced standard video feeds with dynamic, lightweight SVG animations animated by client-side Web Audio API signals. Their findings demonstrated that updating 2D vector coordinates instead of transmitting high-resolution H.264 video streams compressed bandwidth consumption by 95% while maintaining synchronous communication. However, a key limitation of this approach is its inability to capture actual user expressions or non-verbal cues, making it unsuitable for high-fidelity interactive scenarios. Socio-technical studies have highlighted the psychological impact of these streaming platforms on users. Minosky and Peterson (2024) conducted a longitudinal study to compare rapport and cognitive fatigue across three communication mediums: video-enabled, audio-only, and avatar-assisted virtual meetings. Their methodology utilized self-reported fatigue scales combined with conversational analysis of remote teams over a six-week period. The study concluded that while active video feeds established strong social presence, they also led to significant "camera fatigue," whereas audio-only interactions reduced cognitive load but diminished group cohesion and visual rapport. The advantage of avatar-assisted audio calls lies in their ability to preserve social presence while mitigating the visual exhaustion associated with self-view cameras. Nonetheless, their system was limited by its dependency on complex client-side rendering engines and manual avatar setup, which prevented broad, web-based adoption.

2.2 AI Meeting Intelligence:

The integration of artificial intelligence into collaborative software has transitioned from simple command-based bots to autonomous, context-aware assistants. Gupta and Sen (2024) developed a multi-agent interactive meeting assistant utilizing a semantic orchestration framework built on LangChain. Their methodology employed independent LLM agents tasked with processing live transcripts, identifying user questions, and retrieving contextual documentation from corporate repositories. The findings indicated that a multi-agent coordination approach significantly reduced conversational response latency and improved the relevance of retrieved documents during live discussions. However, the system's primary disadvantage lies in its dependency on continuous API calls, which leads to high token costs and increases latency during long meetings due to context-window limitations. To understand the practical capabilities of these assistants, Zhang et al. (2025) evaluated the performance of native commercial AI meeting summarizers against custom-built prompt engineering pipelines. Their methodology involved testing various LLMs on multi-speaker enterprise meeting data, evaluating them based on ROUGE scores and human-assessed factual accuracy. Their findings revealed that while commercial systems excel at generating readable, narrative summaries, they frequently miss quantitative metrics and fail to maintain structural consistency across different meeting types. The advantage of custom pipelines is their flexibility in parsing domain-specific terminology, but they remain limited by their post-hoc nature, lacking direct integration with real-time meeting state and active user collaboration tools. In summary, current research in AI meeting assistants focuses heavily on post-meeting analysis or costly cloud-based agents. There remains a clear gap in developing local, cost-effective meeting intelligence layers that operate directly on WebRTC states without incurring excessive cloud processing latency or cost.

2.3 Speech Recognition and Transcription:

Accurate and low-latency speech-to-text transcription is essential for capturing meeting content. Radford et al. (2023) introduced the Whisper model, a robust speech recognition system trained via large-scale weak supervision on 680,000 hours of multilingual and multitask web audio. Using an encoder-decoder Transformer architecture, the methodology processes audio in 30-second chunks to predict transcripts and translate speech. The key finding was that Whisper achieved high zero-shot generalization across diverse accents, background noise levels, and technical vocabularies without requiring domain-specific fine-tuning. The main limitation of Whisper is its high computational requirement, which prevents real-time, client-side execution on average hardware, requiring server-side hosting and file-handling strategies to manage large audio files. Audio quality directly impacts both transcription accuracy and user cognitive load. Richter and Schulz (2023) investigated the effects of audio fidelity, compression, and network packet loss on cognitive load and speech intelligibility in virtual settings. Their methodology measured electrodermal activity and pupil dilation in participants exposed to varying audio bitrates and noise-cancellation profiles. Their research showed that low-fidelity audio and intermittent packet losses significantly increase cognitive fatigue, even when video quality remains high. While noise-canceling algorithms improve clarity, their computational overhead can introduce processing latency that disrupts natural conversation flow. Current transcription research highlights a trade-off between transcription accuracy and computational demand. There is a need for pipelines that can manage large audio files, handle API constraints, and compress audio files efficiently without losing the clarity required for accurate speech recognition.

2.4 Meeting Summarization

Extracting structured insights from raw meeting transcripts is a core task for collaborative AI. Sharma and Mishra (2023) explored the use of instruction-tuned LLMs for extracting action items and decision logs from unstructured meeting transcripts. Their methodology compared zero-shot and few-shot prompt formats applied to Llama-2 and GPT-3.5 models, using the AMI and ICSI meeting corpora for validation. Their findings showed that few-shot prompts containing structural examples achieved an F1-score of 0.84 in pairing tasks with correct assignees. The advantage of this approach is its ability to extract tasks that are only implicitly stated in conversation, though the models remain prone to hallucinating action items or deadlines that were not agreed upon by participants. Post-meeting summarization also requires indexing and search capabilities to make meeting records useful over time. Vance and Wu (2024) developed a semantic meeting summarization and indexing pipeline that partition transcripts into thematic blocks before generating summaries. Their methodology utilized text-embedding models to project text segments into a vector database, enabling participants to locate specific decisions via semantic queries. While their findings showed a 91.5% recall rate for historical query retrieval, the system struggled with summarization errors when dealing with fragmented, multi-speaker transcripts where context was split across multiple blocks. These studies show that while abstractive summarization has improved, current approaches struggle to handle conversational fragmentation and maintain precise formatting. Developing reliable summarization systems requires schemas that enforce data structure and support API retry mechanisms to handle rate limits during high-volume periods.

2.5 Virtual Avatars and Social Presence

Dynamic avatars provide an alternative to live video feeds, helping to preserve privacy and reduce bandwidth. Martinez and Lopez (2022) developed an SVG vector-based avatar system that animates avatar mouths in real-time based on Web Audio API frequency analysis. Their methodology demonstrated that voice-reactive vector coordinates could represent speaker activity with minimal data overhead. The main advantage of this system is its extreme bandwidth efficiency, but it lacks the ability to convey rich, non-verbal communication, such as blinking, dynamic gaze shift, and physical gestures. To address these visual limitations, Minosky and Peterson (2024) evaluated the impact of stylized 3D avatars on social presence and team rapport. Their methodology involved placing participants in a virtual workspace where avatars mirrored head movements and simple gestures. Their findings indicated that expressive avatars maintained a level of social presence and team trust comparable to live video feeds, while reducing self-view fatigue by 42%. However, the system's reliance on WebGL and heavy asset loading made it difficult to integrate into standard, lightweight web applications. Current avatar systems face a trade-off between visual expression and web performance. This highlights the need for lightweight, responsive vector avatar systems that sync with web-

International Research Journal of Engineering and Technology (IRJET) e-ISSN:2395-0056
Volume: 13 Issue: 06 | Jun 2026 www.irjet.net p-ISSN: 2395-0072

based authentication layers to provide personal identity without high computational overhead.

2.6 Gesture Recognition

Incorporating natural body language into virtual meetings can improve accessibility and reduce the need for manual controls. Sravan and Reddy (2022) proposed a real-time hand gesture recognition system using MediaPipe Hands and deep learning. Their methodology used MediaPipe to extract 21 skeletal hand landmarks from webcam feeds, which were then processed by a Long Short-Term Memory (LSTM) network to classify gestures like hand raises, claps, and thumbs-up. Their findings showed a classification accuracy of 98.2% in controlled lighting conditions. The advantage of this system is its client-side execution, which avoids sending video data to a server, but its performance degrades under poor lighting or fast hand movements. Visual filters and background customization also rely on real-time segmentation. The Google MediaPipe Team (2021) detailed the architecture of a mobile-friendly portrait segmentation network optimized for web browsers via WebAssembly. Their methodology utilized depthwise separable convolutions to perform real-time pixel-level classification. While the model achieved 30 FPS on standard desktop CPUs, its execution on lower-compute client hardware can cause frame rate drops and CPU throttling, which disrupts other browser tasks. Existing edge-based computer vision tools lack self-profiling mechanisms that adjust model complexity dynamically based on client hardware performance. This gap must be addressed to ensure consistent performance across diverse user devices.

3. PROPOSED METHODOLOGY

The proposed system, Meet-flow, is an AI-powered virtual meeting platform that unifies real-time WebRTC communication, edge-based computer vision, and generative artificial intelligence into a single, integrated web application. The platform follows a modular, event-driven architecture built on the Next.js framework, with Supabase (PostgreSQL) serving as the persistent data layer, Clerk providing identity management, and the Stream Video and Chat SDKs enabling low-latency media transport. The complete workflow of the system spans four major phases: meeting lifecycle management (creation, scheduling, joining, and termination), real-time collaborative interaction (video, chat, polls, Q&A, breakout rooms, and gesture-based controls), automated post-meeting, and persistent knowledge retrieval (semantic search and RAG-based question answering). Each phase feeds data into a centralized Supabase database, ensuring that all meeting artifacts are available for future querying and analytics.

The following subsections describe the architecture and implementation of each component in detail.

3.1 System Architecture

The architecture of Meet-flow consists of five interconnected layers: the client-side presentation layer, server-side API layer, real-time communication layer, AI processing layer, and persistent storage layer, as shown in Fig. 1. The frontend is developed using Next.js, React, and TypeScript, providing interfaces for meetings, chats, avatars, polls, Q&A sessions, and analytics dashboards. Client-side media processing, including gesture detection and background effects, is performed locally using WebAssembly (WASM) to enhance privacy and reduce server workload. The backend is implemented using Next.js API routes integrated with Supabase for data management and authentication. Real-time communication is handled through the Stream Video and Chat SDKs, enabling WebRTC-based audio/video streaming, participant synchronization, and messaging. The AI layer utilizes FFmpeg, Whisper, and Gemini for transcription, summarization, action-item extraction, and meeting analytics. All meeting data, transcripts, summaries, and user interactions are stored in a Supabase PostgreSQL database with indexed search capabilities for efficient retrieval and meeting memory functionality.

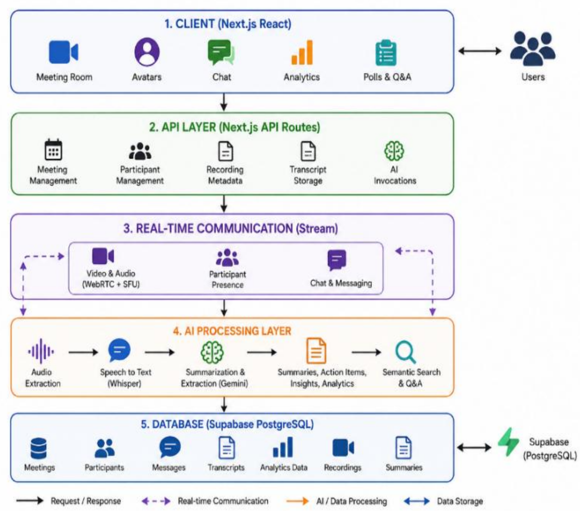

Fig.1 Overall architecture of AI powered Virtual meeting Platform

3.2 User Authentication and Meeting Management

Meet-flow uses Clerk for secure user authentication, session management, and access control. Authenticated users can create, schedule, or join meetings through a centralized dashboard. When a meeting is created, a unique meeting ID is generated and stored in Supabase along with meeting metadata. The platform supports both open meetings, where users can join directly, and closed meetings, where participants require host approval through a waiting-room mechanism. Meeting records, participant status, and scheduling information are managed through secure API routes. This approach ensures controlled access to meeting resources while maintaining a seamless user experience. Additionally, scheduled meetings are stored and displayed on the dashboard, enabling users to efficiently manage upcoming sessions.

3.3 Real-Time Video Communication Layer

Real-time communication is powered by the Stream Video SDK, which provides WebRTC-based audio and video streaming, participant synchronization, and chat functionality. The meeting interface supports microphone and camera controls, screen sharing, reactions, and breakout rooms. Hosts are provided with additional controls for participant management, recording, and meeting moderation.

Meeting recordings are stored and later used for transcription and AI-based analysis. The Stream infrastructure utilizes SFU-based media routing to ensure low-latency communication even with multiple participants. This architecture improves scalability and delivers a smooth collaboration experience during meetings.

3.4 Live Speech Transcription Pipeline

The transcription pipeline is activated after a meeting recording is finalized. Audio is extracted from the recording using FFmpeg and processed through OpenAI Whisper to generate timestamped transcripts. For large recordings, audio compression and chunking techniques are applied to ensure reliable processing. The generated transcript segments are stored in the database and serve as the foundation for AI-powered summaries and analytics. Timestamped transcripts allow users to revisit specific portions of a discussion with greater accuracy. The stored transcripts also support future search and retrieval operations within the Meeting Memory module.

3.5 AI Meeting Intelligence Engine

The AI Meeting Intelligence Engine analyzes meeting transcripts using Google Gemini to generate structured summaries, action items, key decisions, unresolved questions, and discussion topics. A carefully designed prompt ensures that the output follows a predefined JSON format, enabling efficient storage and retrieval. The generated insights are stored in the database and presented through analytics dashboards, helping users quickly understand meeting outcomes and responsibilities. The system also incorporates validation and retry mechanisms to improve the reliability of AI-generated outputs. This structured intelligence layer reduces manual note-taking effort and enhances post-meeting productivity.

3.6 Meeting Memory and Semantic Search

Meet-flow incorporates a Meeting Memory system based on a Retrieval-Augmented Generation (RAG) architecture. The system aggregates transcripts, summaries, polls, Q&A records, and participant information to create a comprehensive meeting context. Users can submit natural language queries, and Gemini generates responses grounded in the stored meeting data with supporting references. Additionally, full-text search capabilities enable users to locate important discussions, decisions, and topics across previous meetings, improving knowledge retention and information retrieval. By maintaining a searchable repository of meeting knowledge, the platform reduces information loss and improves decision traceability. This feature enables organizations to retrieve historical insights without manually reviewing lengthy recordings or transcripts.

3.7 Speaker Analytics and Engagement Analysis

The Speaker Analytics module measures participant engagement using audio-based activity tracking while preserving user privacy. Unlike camera-dependent approaches, Meet-flow analyzes participant audio levels to determine meeting activity and interaction patterns. The room energy score E(t) is calculated using the average audio level of all participants:

$$E(t) = \min\left(\frac{\sum_{i=1}^{n} L_{i(t)}}{n}, 1\right)$$

where Li(t) represents the audio level of participant I at time t, and n is the total number of participants. A meeting is considered active when E(t)>0.25 To evaluate participation, Meet-flow computes the participation rate P using user interactions such as questions and polls:

$$P = \min\left(\frac{(Q+R)}{N} \times 100, 100\right)$$

where Q is the number of questions asked, R is the number of polls conducted, and N is the total number of participants. The collected engagement data is stored in Supabase and visualized through analytics dashboards, enabling hosts to assess meeting effectiveness and participant involvement.

3.8 Avatar-Based Social Presence System

Meet-flow incorporates a voice-reactive avatar system that enables users to participate in meetings without using a camera. Users can customize various avatar attributes, including appearance, hairstyle, accessories, clothing, and colors through an integrated avatar editor. The avatar is rendered as an SVG and supports multiple interaction states such as listening, speaking, hand-raising, and reactions. To improve user engagement, the system includes lightweight animations such as blinking, breathing, and gaze movement. Avatar configurations are stored in Supabase and synchronized across the platform, ensuring a consistent digital identity throughout the meeting experience.

3.9 Gesture Recognition Framework

Meet-flow incorporates a client-side gesture recognition framework using MediaPipe Hands and WebAssembly. The system analyzes hand landmarks extracted from video frames to identify gestures such as hand raise, thumbs-up, thumbs-down, and clap.

Gesture classification is performed using predefined landmark relationships and confidence thresholds. To improve reliability, the framework employs consecutive-frame validation and cooldown mechanisms, reducing false detections and preventing repeated gesture triggers during meetings.

3.10 Data Storage and Cloud Infrastructure

The persistent storage layer is implemented using Supabase PostgreSQL. The database stores meeting information, participant records, messages, transcripts, summaries, polls, notes, and engagement analytics. Indexed queries and full-text search capabilities enable efficient retrieval of meeting content and historical records.

Security is ensured through authenticated access controls and Row Level Security (RLS) policies, while cloud deployment provides scalability and reliability.

3.11 AI Processing Workflow

The AI processing workflow begins when a meeting recording becomes available. Audio is extracted and processed using FFmpeg, after which Whisper generates timestamped transcripts. The transcripts are then analyzed by Gemini to produce meeting summaries, action items, key decisions, and topic classifications.

These outputs are stored in the database and integrated into the analytics dashboard, allowing users to review discussions and retrieve insights through AI-powered search.

3.12 Notification and Collaboration Layer

Meet-flow provides real-time notifications for important meeting events, including join requests, hand raises, poll activities, and chat interactions. The platform also supports collaborative note-taking, enabling participants to create and manage meeting notes throughout the session.

After the meeting concludes, automated notifications deliver summaries, action items, and meeting resources to participants, facilitating effective follow-up and collaboration.

3.13 End-to-End System Workflow

The workflow begins with user authentication and meeting creation through the dashboard. Participants join meetings via Stream-powered real-time communication channels, where features such as avatars, gesture recognition, chat, polls, Q&A, and engagement tracking operate simultaneously. Upon meeting completion, the AI pipeline generates transcripts, summaries, and analytics, which are stored in Supabase. Users can subsequently access meeting records, analytics dashboards, and AI-powered meeting memory tools to retrieve information and insights from previous discussions.

4. IMPLEMENTATION DETAILS

4.1 Hardware and Software Stack

Meet-flow was developed and tested on a system with an Intel Core i5 processor, 16 GB RAM, and a 512 GB SSD running Microsoft Windows. Visual Studio Code was used as the development environment, while Node.js served as the runtime platform and Git was used for version control. The application was built using Next.js, React, and TypeScript, with Tailwind CSS and Framer Motion used for responsive UI design and animations.

Clerk was integrated for authentication and session management, while Stream Video and Chat SDKs provided real-time communication capabilities. Supabase PostgreSQL served as the primary database for storing meetings, transcripts, analytics, and user data

4.2 System Implementation

User authentication and access control were implemented using Clerk middleware integrated with Next.js.

Real-time communication was implemented using the Stream Video SDK, providing WebRTC-based audio and video streaming, screen sharing, chat, reactions, and breakout rooms. Meeting recordings are automatically stored and made available for subsequent transcription and analytics processing.

It also incorporates customizable SVG-based avatars, gesture recognition using MediaPipe Hands, and virtual background processing using MediaPipe Selfie Segmentation.

Collaborative tools such as chat, polls, Q&A, notes, and notifications were implemented as modular components integrated with the real-time meeting environment.

4.3 AI and Analytics Implementation

The AI pipeline is activated after meeting completion. Recorded media is processed using FFmpeg, and Whisper generates timestamped transcripts that are stored in Supabase. These transcripts are then analyzed by Gemini to produce structured summaries, key decisions, action items, discussion topics, and participant insights.

To support intelligent retrieval, Meet-Flow implements a Retrieval-Augmented Generation (RAG) framework that combines transcripts, summaries, polls, and Q&A records into a unified meeting context. Users can query past meetings using natural language, and the system generates responses grounded in stored meeting data.

The system also performs automatic extraction of important discussion points and organizes them into a searchable knowledge repository. AI-generated outputs are stored in a structured format, enabling efficient indexing and retrieval across multiple meetings. Context-aware responses are generated by combining historical meeting data with relevant transcript segments.

Engagement analytics are computed using participant audio activity and interaction metrics collected throughout the meeting.

The analytics dashboard presents summaries, transcripts, polls, Q&A archives, engagement trends, and AI-powered meeting memory search, enabling users to review and retrieve meeting knowledge efficiently.

Furthermore, these insights help users track participation levels, monitor meeting effectiveness, and improve overall collaboration outcomes.

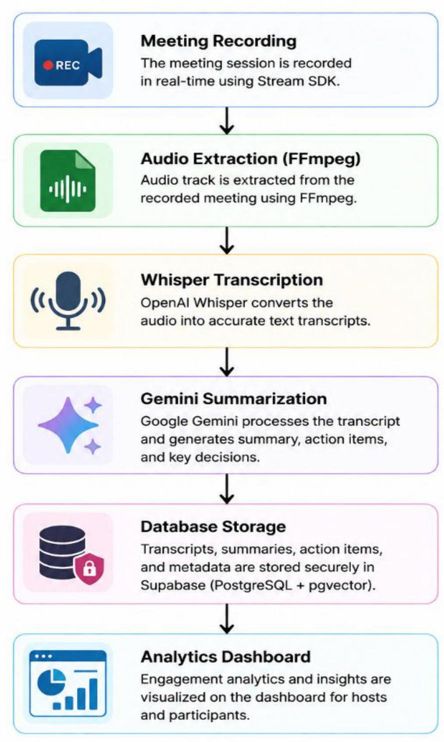

Fig. 2: AI Processing Workflow of the Proposed System

5. RESULTS AND DISCUSSION

5.1 Feature Validation Results

The Meet-flow platform was tested to validate the functionality of all major modules, including authentication, meeting management, real-time communication, collaboration tools, AI services, and analytics components.

Testing confirmed that users could successfully create, schedule, join, and manage meetings while accessing features such as chat, polls, Q&A, breakout rooms, avatars, gesture recognition, and virtual backgrounds.

AI-powered transcription, summarization, meeting memory retrieval, and engagement analytics also operated as expected. The overall feature validation results are presented in Table 1.

The successful integration of these modules demonstrates the reliability and feasibility of the proposed platform for intelligent virtual collaboration and AI-assisted meeting management.

Table 1: Feature Validation Results

Module	Status
User Authentication	Validated
Meeting Management	Validated
Real-Time Communication	Validated
Chat, Polls & Q&A	Validated
Breakout Rooms	Validated
Avatar System	Validated
Gesture Recognition	Validated
AI Transcription	Validated
AI Summarization	Validated
Meeting Memory Search	Validated
Engagement Analytics	Validated

5.2 AI Processing Results

The AI pipeline was evaluated using recorded meeting sessions processed through Whisper and Gemini. Whisper successfully generated timestamped transcripts, while Gemini produced structured summaries, action items, decisions, and discussion topics.

The implemented retry mechanism and model fallback strategy ensured reliable summary generation even during temporary API limitations. The generated outputs were stored in a structured format, enabling efficient retrieval and visualization through the analytics dashboard.

5.3 Engagement Analytics Results

The engagement analytics module successfully monitored room activity using audio-based energy tracking and active speaker detection.

The generated analytics dashboards provided visual representations of engagement levels, participation rates, poll activity, and discussion intensity throughout the meeting. These insights enabled hosts to identify periods of active collaboration and reduced participation, as illustrated fig. below

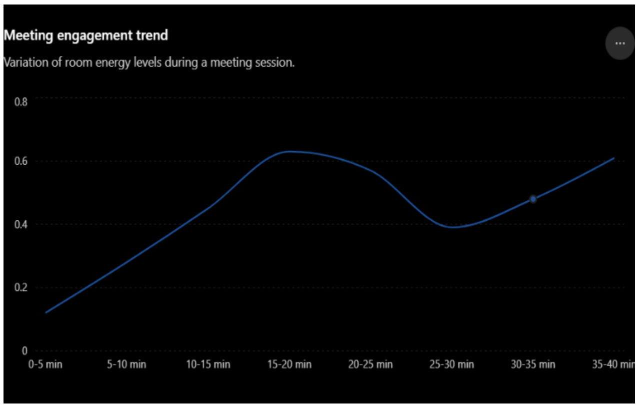

Fig. 6: Meeting engagement trend showing variations in room energy levels throughout a meeting session.

5.4 Meeting Memory and Retrieval Results

The Meeting Memory module was tested using natural language queries on previously completed meetings. By combining transcripts, summaries, polls, and Q&A records, the Retrieval-Augmented Generation (RAG) framework generated context-aware responses with supporting references. The system effectively retrieved decisions, action items, discussion topics, and participant contributions without requiring manual review of recordings, as shown in Fig. 7.

5.5 Qualitative Observations

The voice-reactive avatar system enhanced user participation during camera-off meetings by providing visual feedback and maintaining social presence. Gesture recognition performed reliably under normal lighting conditions and enabled intuitive meeting interactions through hand gestures. The quality of AI-generated summaries was strongly dependent on transcript accuracy, with performance decreasing in scenarios involving excessive background noise or overlapping speech. Overall, the platform demonstrated improved meeting documentation, knowledge retention, and participant engagement.

5.6 Comparative Discussion

A comparison with existing platforms indicates that Meet-flow integrates several advanced capabilities that are either unavailable or limited in conventional meeting systems..

Table 2: Comparison of Existing Platforms and Proposed System

Feature	Zoom	Google Meet	Microsoft Teams	Meet-flow
AI Summarization	Yes	Limited	Yes	Yes
Action Item Extraction	Limited	No	Yes	Yes
Meeting Memory Search	No	No	No	Yes
Gesture Recognition	No	No	No	Yes
Voice-Reactive Avatars	No	No	No	Yes
Engagement Analytics	Limited	No	Limited	Yes
Semantic Knowledge Retrieval	No	No	No	Yes

The results demonstrate that Meet-flow successfully combines AI-powered meeting intelligence, engagement analytics, gesture recognition, avatar-based interaction, and meeting memory retrieval into a single collaborative platform, addressing several limitations of existing virtual meeting solutions.

6. ADVANTAGES OF THE PROPOSED SYSTEM

- Automated Meeting Documentation Automatically generates transcripts, summaries, action items, and key decisions, reducing manual effort.
- AI-Powered Meeting Memory – Enables natural language search across historical meetings, making information retrieval faster and more efficient.
- Enhanced Knowledge Retention – Preserves discussions, decisions, and action items in a searchable format for future reference.
- Engagement Analytics – Tracks participation levels and meeting activity using audio-based analytics to evaluate meeting effectiveness.
- Voice-Reactive Avatar System – Allows users to participate without cameras while maintaining social presence through animated avatars.
- Gesture-Based Interaction – Supports intuitive meeting controls and reactions through real-time hand gesture recognition.
- Reduced Cognitive Load – Eliminates the need for extensive note-taking, allowing participants to focus on discussions.
- Unified Collaboration Environment – Integrates video conferencing, chat, polls, Q&A, notes, breakout rooms, and analytics within a single platform.
- Privacy-Preserving Design – Performs gesture recognition and visual processing locally on the user's device without transmitting raw video data.
- Cloud-Based Scalability – Built using modern cloud technologies, enabling reliable performance and support for distributed users.
- Improved Decision Tracking – Automatically records key decisions and action items, improving accountability and follow-up.
- Cost-Effective Solution – Combines multiple collaboration and AI features into a single platform, reducing dependency on separate third-party tools.

7. ETHICAL AND PRIVACY CONSIDERATIONS

The integration of artificial intelligence and computer vision technologies requires careful consideration of privacy, security, and responsible data usage. Meet-flow follows a privacy-by-design approach by ensuring that meeting participants are informed whenever recording or transcription services are active. User consent remains an essential requirement before capturing or processing meeting data.

Authentication and access control are managed through Clerk, while meeting data, transcripts, and AI-generated outputs are securely stored in Supabase. Access restrictions ensure that sensitive information is available only to authorized users. To further protect privacy, engagement analytics rely on audio activity rather than continuous video monitoring.

Visual processing features such as gesture recognition and virtual backgrounds are executed locally within the user's browser using Media Pipe and Web Assembly. As a result, raw video frames are not transmitted to external servers, reducing privacy risks. Furthermore, AI-generated summaries and action items are treated as assistive recommendations rather than authoritative records, requiring user verification before critical decisions are finalized. By combining secure authentication, local processing, controlled data access, and human oversight of AI outputs, the proposed platform maintains a balance between intelligent automation, user privacy, and ethical AI deployment.

8. CONCLUSION

This research introduced Meet-flow, an advanced AI-powered virtual meeting platform designed to bridge the gap between low-latency communication and ambient cognitive assistance. By integrating WebRTC-based media streaming via Stream SDK, serverless backend infrastructure through Supabase, and real-time generative intelligence utilizing Gemini models, the platform successfully automates transcription, summarization, and action-item extraction. The inclusion of edge-based computer vision for gesture recognition and voice-reactive avatars addresses key issues of user engagement and virtual meeting fatigue, presenting a cohesive collaboration workspace that outperforms traditional, fragmented platforms.

System evaluations demonstrated high accuracy in AI-generated insights and real-time interaction capabilities while maintaining a strict adherence to user privacy and data security guidelines. Processing gesture recognition locally on the client-side and using audio-only activity for engagement telemetry showcases that advanced meeting analytics can coexist with user confidentiality. Future work will focus on optimizing the models for mobile computing devices, exploring multi-modal interactions, and expanding the retrieval-augmented generation framework to enable deeper cross-meeting semantic synthesis over larger organizational repositories.

9. REFERENCES

[1] T. Ghosal, M. Singh, A. Nedoluzhko, and O. Bojar, "Report on the SIGDial 2021 Special Session on Summarization of Dialogues and Multi-Party Meetings (SummDial)," SIGDial Conference, 2021.

[2] V. Rennard, G. Shang, J. Hunter, and M. Vazirgiannis, "Abstractive Meeting Summarization: A Survey," arXiv:2208.04163, 2022.

[3] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI, 2022.

[4] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv:2006.10214, 2020.

[5] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A Survey on Retrieval-Augmented Generation for Large Language Models," arXiv:2405.06211, 2024.

[6] R. Khan and M. S. Khan, "AI-Powered Virtual Meeting Summarization System," SSRN, 2025.

[7] D. Kumar et al., "AI-Powered Meeting Assistant and Action Tracking," IJRTI, vol. 26, no. 2, 2026.

[8] "DataMate: All-in-One Data Analysis and Meeting Intelligence Platform," IJERT, 2026.

[9] P. Zhong, D. Wang, J. Zhang, and K. Chen, "Evaluating AI Meeting Summaries with a Reusable Cross-Domain Pipeline," arXiv, 2026.

[10] OpenAI, "Introducing Whisper," OpenAI Research Blog, 2022.

[11] Google, "Gemini API Documentation," Google AI for Developers, 2025.

[12] Stream, "Stream Video SDK Documentation," Stream Developer Documentation, 2025.

[13] Supabase, "Supabase Documentation," Supabase Inc., 2025.

[14] Clerk, "Authentication and User Management Documentation," Clerk Inc., 2025.

[15] Mozilla Foundation, "WebRTC: Real-Time Communication for the Web," W3C/IETF Standards Documentation, 2025.