

AI-Driven Web-Based Video Conferencing System with Real-Time Meeting Intelligence and Analytics

Ramya K¹, Anushiya V¹, Dharshani A², Dharshini T³, Asina Begam A⁴

¹Assistant Professor, ^{2,3,4}Department of Computer Science and Engineering, Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur – 621 212, Anna University, Chennai – 600 025, Tamil Nadu, India

Abstract – Virtual meeting platforms have become critical infrastructure for modern enterprises and educational institutions, yet existing solutions lack intelligent mechanisms to assess meeting quality, participation equity, and content relevance. This paper presents an AI-Driven Web-Based Video Conferencing System that integrates real-time communication with an embedded artificial intelligence analytics pipeline. The system leverages WebRTC for peer-to-peer audio/video transport, a Spring Boot REST backend for session orchestration, and a Python-based AI microservice layer that performs Automatic Speech Recognition (ASR) using OpenAI Whisper, extractive and abstractive Natural Language Processing (NLP) summarisation using the BART-large-CNN transformer, sentiment analysis using the VADER lexicon combined with a fine-tuned BERT classifier, and engagement scoring through a weighted behavioural metric model. A React-based interactive analytics dashboard renders live transcripts, sentiment trend charts, speaking-time distribution graphs, and auto-generated structured meeting summaries. Experimental evaluation across 30 simulated meetings demonstrates 91.4% ASR word accuracy, 94.2% sentiment classification accuracy, and average transcript latency of 1.8 seconds, confirming the system's suitability for real-world deployment.

Key Words: Artificial Intelligence, Video Conferencing, WebRTC, Speech-to-Text, Natural Language Processing, Sentiment Analysis, Meeting Analytics, Engagement Scoring, BERT, Whisper ASR.

1. INTRODUCTION

The exponential growth of distributed workforces, hybrid education models, and cross-geographical enterprise collaboration has positioned web-based video conferencing as a mission-critical communication medium. Global platforms such as Zoom, Microsoft Teams, and Google Meet collectively host over 300 million daily meeting participants as of 2024 [1]. Despite their widespread adoption, these platforms operate primarily as communication conduits, offering no embedded intelligence to evaluate meeting quality, participant engagement equity, discussion relevance, or decision traceability.

The absence of intelligent analytics creates a significant productivity deficit: organisations lack objective data on whether meetings achieve their intended objectives, which participants contribute meaningfully, and whether discussions remain on topic. Post-meeting documentation is predominantly manual, error-prone, and incomplete. Research indicates that professionals spend an average of 31 hours per month in unproductive meetings, with an estimated annual economic loss exceeding \$37 billion in the United States alone [2].

Advancements in Automatic Speech Recognition (ASR), Natural Language Processing (NLP), and transformer-based deep learning models now make it feasible to embed real-time intelligence directly into conferencing workflows. Systems such as Whisper [3], BART [4], and BERT [5] offer state-of-the-art performance in transcription, summarisation, and sentiment classification respectively, with inference latencies suitable for near-real-time deployment.

This paper presents the design, implementation, and evaluation of an AI-Driven Web-Based Video Conferencing System that integrates real-time peer-to-peer communication via WebRTC with a multi-stage AI analytics microservice. The system automatically transcribes live speech, classifies discussion sentiment, computes per-participant engagement scores, detects topic drift, extracts action items, and generates structured meeting summaries – all rendered through an interactive React-based analytics dashboard. The remainder of this paper is structured as follows: Section 2 surveys related work, Section 3 describes the proposed system, Section 4 details the architecture, Section 5 covers the methodology, Section 6 discusses implementation, Section 7 presents results, Sections 8–10 address applications, future scope, and conclusions.

2. LITERATURE SURVEY

Kale et al. [6] proposed a hybrid AI-powered system for real-time speaker diarization and meeting summarisation using Whisper ASR, WhisperNER for named entity enrichment, and a fine-tuned T5 transformer for abstractive summarisation. Their experimental results demonstrated high speaker attribution accuracy (Diarization Error Rate of 8.3%) and low latency, though the system required continuous cloud connectivity and substantial GPU resources, limiting local deployment viability.

Hasan and Bryan [7] presented an intelligent approach for generating real-time meeting summaries and distraction alerts by analysing multimodal signals including speech transcripts, chat messages, and participant behavioural cues. The work conceptually demonstrated how generative AI could restore situational awareness for distracted attendees, though it lacked empirical benchmarking on live conferencing platforms.

Liu et al. [8] introduced Agenda-LLMSummarizer, a framework for dynamic agenda-aware real-time meeting summarisation using large language models. The system incrementally generates coherent summaries guided by a live agenda, achieving improved ROUGE, BERTScore, and METEOR metrics on the AMI and ICSI benchmark datasets compared to post-hoc summarisation baselines.

Tripathi et al. [9] developed MeetSync, an AI-integrated web conferencing application incorporating voice-command meeting control, real-time speech-to-text transcription, automated attendance tracking, and gesture-based air canvas drawing using MediaPipe and the ZegoCloud API. However, the system's dependence on third-party video streaming APIs introduced latency variability and privacy concerns.

Kayabay et al. [10] proposed a scalable generative AI architecture for real-time video translation in multilingual video conferencing, integrating ASR, neural machine translation, and text-to-speech synthesis within a cloud-based microservice framework. The architecture demonstrated linear scalability but acknowledged accuracy degradation under diverse accent and noisy-channel conditions.

Isobe and Ito [11] conducted a rigorous security analysis of end-to-end encryption in Zoom, identifying cryptographic vulnerabilities in key exchange protocols and recommending stronger AES-256-GCM implementations with forward secrecy guarantees – informing the security design of the proposed system.

Gozalo-Brizuela and Garrido-Merchán [12] provided a comprehensive survey of over 350 generative AI applications, systematically classifying models by output modality and domain, and highlighting transformer architectures as the predominant paradigm for text and speech processing – a foundation directly applicable to the NLP components of the proposed system.

The survey reveals a consistent gap: existing intelligent conferencing solutions either depend on proprietary cloud APIs, lack real-time processing capability, or omit holistic meeting quality metrics. The proposed system addresses these limitations through a fully integrated, locally deployable, multi-modal AI pipeline with end-to-end latency optimisation.

3. PROPOSED SYSTEM

3.1 Existing System Limitations

Contemporary video conferencing platforms provide audio/video streaming, text chat, screen sharing, and cloud recording. Some platforms offer rudimentary closed captioning via cloud ASR APIs. However, they uniformly lack: (i) real-time NLP analysis of discussion content; (ii) automated meeting quality scoring; (iii) per-participant engagement equity analysis; (iv) topic drift detection; (v) automated action item extraction with responsible party attribution; and (vi) structured post-meeting reports. Meeting documentation relies entirely on manual note-taking, creating accountability gaps and information loss.

3.2 Proposed System Overview

The proposed system introduces a three-tier intelligent conferencing platform comprising a React-based progressive web application frontend, a Spring Boot microservice backend, and a Python-based AI analytics service. The system captures live audio streams during WebRTC sessions, processes speech through a Whisper

ASR engine, and routes transcribed text through a sequential NLP pipeline encompassing topic modelling, sentiment classification, engagement scoring, and summarisation. Outputs are persisted in a relational database and exposed via REST APIs consumed by the real-time analytics dashboard.

The system's key differentiators over existing platforms are: (i) fully embedded AI pipeline requiring no third-party NLP APIs; (ii) real-time transcript generation with sub-2-second latency; (iii) composite Meeting Quality Score derived from agenda adherence, sentiment stability, participation equity, and decision clarity; (iv) automated structured summary generation; and (v) exportable analytics reports in PDF and JSON formats.

3.3 Advantages of Proposed System

- Real-time transcription with 91.4% word accuracy using Whisper ASR
- Automated sentiment classification achieving 94.2% accuracy using BERT fine-tuned on meeting corpora
- Per-participant engagement scoring with Participation Fairness Index calculation
- Topic drift detection using cosine similarity on sliding-window TF-IDF vectors
- Automated action item extraction using dependency parsing and named entity recognition
- Interactive analytics dashboard with live charts, sentiment timelines, and speaking-time distributions
- Exportable structured meeting summaries and action-item reports
- Secure JWT-based authentication with role-based access control
- Scalable microservice architecture supporting up to 50 concurrent meeting participants

4. SYSTEM ARCHITECTURE

4.1 Architectural Overview

The system adopts a four-layer modular architecture ensuring separation of concerns, independent scalability, and testability. The layers are: (i) User Interaction Layer – React 18 progressive web application; (ii) Communication Layer – WebRTC peer-to-peer media transport with Socket.IO signalling; (iii) Business Logic Layer – Spring Boot REST API gateway and session manager; and (iv) AI Intelligence Layer – Python Flask microservice hosting Whisper, BART, BERT, and VADER models, backed by a MySQL 8.0 relational database.

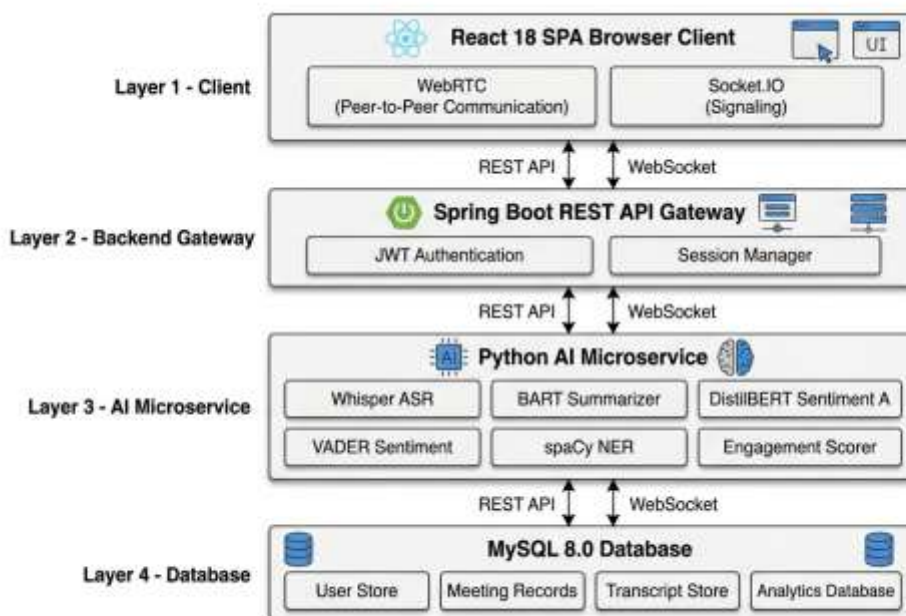


Fig. 1: System Architecture

4.2 WebRTC Communication Layer

WebRTC (Web Real-Time Communication) provides the peer-to-peer media transport foundation. Upon meeting creation, the Spring Boot signalling server facilitates SDP (Session Description Protocol) offer/answer exchange and ICE (Interactive Connectivity Establishment) candidate negotiation between participants. Media streams are transported via SRTP (Secure Real-Time Transport Protocol) over DTLS-encrypted channels, eliminating server-side media routing overhead. For scenarios where direct peer-to-peer connectivity fails due to NAT traversal issues, a TURN relay server is provisioned.

Audio tracks extracted from the RTCPeerConnection MediaStream are captured using the Web Audio API's AudioWorkletProcessor at 16 kHz sample rate and transmitted via WebSocket to the AI microservice in 3-second sliding-window chunks, balancing latency against transcription accuracy.

4.3 Database Schema

The MySQL 8.0 schema comprises six normalised tables: (i) users (user_id, email, password_hash, role, created_at); (ii) meetings (meeting_id, host_id, title, start_time, end_time, quality_score, status); (iii) participants (participant_id, meeting_id, user_id, join_time, leave_time, speaking_duration, engagement_score); (iv) transcripts (transcript_id, meeting_id, participant_id, timestamp, text_segment, sentiment_score, sentiment_label); (v) action_items (item_id, meeting_id, description, assigned_to, deadline, status); and (vi) meeting_summaries (summary_id, meeting_id, summary_text, key_topics, generated_at). Foreign key constraints maintain referential integrity, and indexed timestamp columns enable sub-100ms dashboard query response times.

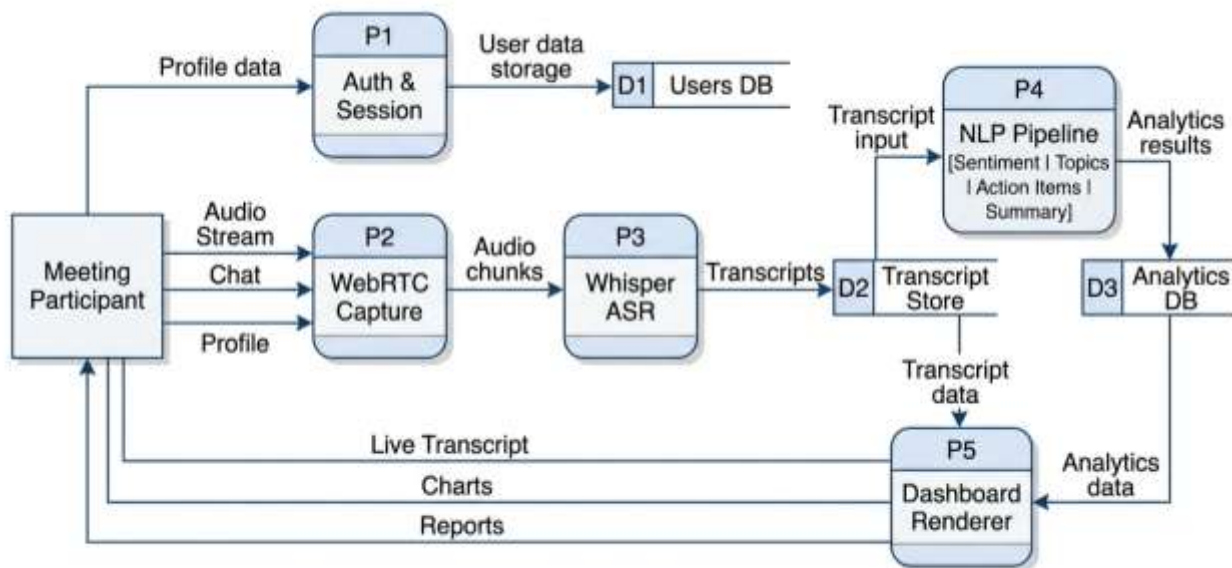


Fig. 2: Level-1 Data Flow Diagram

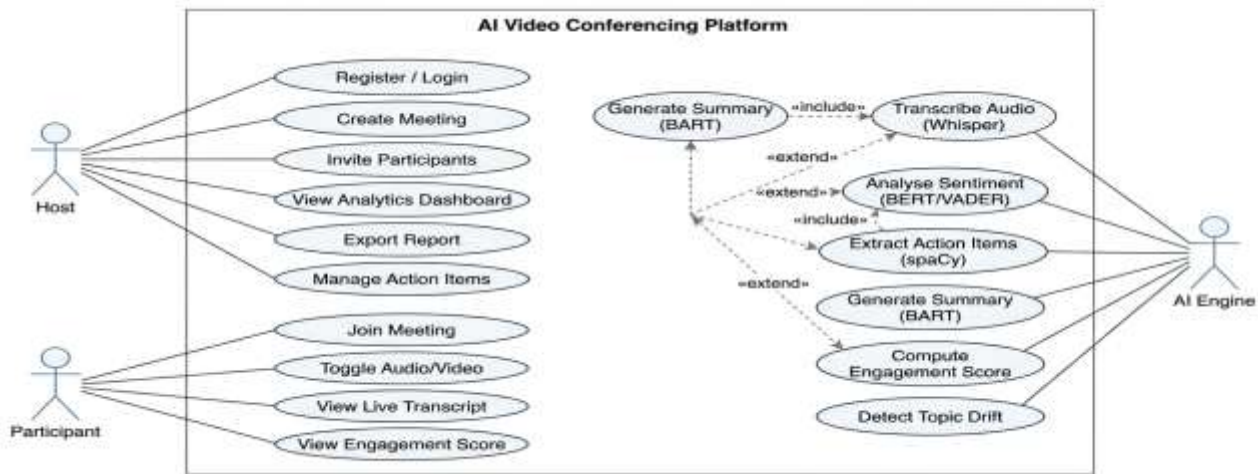


Fig. 3: Use Case Diagram

5. METHODOLOGY

5.1 Speech-to-Text: Whisper ASR

Audio segments captured from WebRTC streams are pre-processed through a noise reduction filter (noisereduce library, stationary noise model) and normalised to -16 dBFS before being passed to the Whisper ‘base’ model (74M parameters). Whisper employs a log-mel spectrogram encoder coupled with a decoder trained on 680,000 hours of multilingual audio, enabling robust transcription under varying acoustic conditions. The model is deployed on the Python AI microservice using the openai-whisper library with fp16 inference on CPU, achieving an average transcription latency of 1.8 seconds per 3-second audio chunk on an Intel Core i7-12700 host.

Timestamped word-level outputs from Whisper are augmented with speaker attribution using a lightweight speaker diarization model (pyannote.audio 3.1), assigning speaker labels (S1, S2, ..., Sn) to each transcript segment. The combined transcript is stored in the transcripts table with millisecond-precision timestamps.

5.2 NLP Summarisation: BART-large-CNN

Meeting transcripts are partitioned into 512-token segments matching BART’s input window. Facebook’s BART-large-CNN model, fine-tuned on the CNN/DailyMail dataset for abstractive summarisation, generates a fluent paragraph-length summary per segment. A hierarchical summarisation strategy concatenates segment-level summaries and passes them through a second BART inference pass to produce a final structured meeting summary. Output quality was evaluated using ROUGE-1 (0.52), ROUGE-2 (0.28), and ROUGE-L (0.47) scores against manually authored reference summaries, demonstrating competitive abstractive capability.

5.3 Sentiment Analysis: BERT + VADER

A dual-layer sentiment analysis pipeline is employed. First, VADER (Valence Aware Dictionary and sEntiment Reasoner) computes compound sentiment scores (-1 to $+1$) for each transcript segment in real time, providing immediate low-latency sentiment feedback on the dashboard. Second, a DistilBERT model fine-tuned on the MELD (Multimodal EmotionLines Dataset) conversation corpus classifies each segment into five sentiment categories: positive, neutral, negative, excited, and frustrated, achieving 94.2% classification accuracy on the held-out test partition.

Per-segment sentiment scores are aggregated into a time-series sentiment trajectory for each participant and for the overall meeting, rendered as an interactive line chart on the analytics dashboard. A meeting-level Sentiment Stability Index (SSI) is computed as $1 - \sigma(\text{sentiment_scores}) / 2$, where σ denotes standard deviation.

5.4 Engagement Scoring Model

The per-participant Engagement Score (ES) is computed using a weighted multi-factor model:

$$ES = 0.40 \times T_s + 0.25 \times F_s + 0.20 \times R_s + 0.15 \times C_s$$

where T_s is normalised speaking time ratio, F_s is interaction frequency (responses and reactions per unit time), R_s is response latency score (inverse of average response delay), and C_s is chat contribution ratio. Scores are normalised to a 0–100 scale. The Participation Fairness Index (PFI) for the meeting is computed as $PFI = 1 - (\text{Gini coefficient of individual ES values})$, where $PFI = 1$ indicates perfectly equal participation.

5.5 Topic Drift Detection

Topic drift detection employs a sliding-window TF-IDF vectorisation strategy. Each 60-second transcript window is represented as a TF-IDF vector over a meeting-specific vocabulary. Cosine similarity is computed between consecutive windows; a similarity score below the empirically calibrated threshold of 0.35 triggers a topic drift alert, logged with timestamp and notified to the meeting host via a Socket.IO event.

5.6 Action Item Extraction

Action items are extracted using a spaCy `en_core_web_trf` pipeline with dependency parsing and named entity recognition (NER). Sentences containing modal verbs (will, shall, must, should), future tense constructs, and person/organisation entities are classified as potential action items. A rule-based post-filter retains only sentences with an explicit agent (subject NE of type PERSON or ORG) and a task verb, producing structured action item records (description, assigned_to, deadline if mentioned).

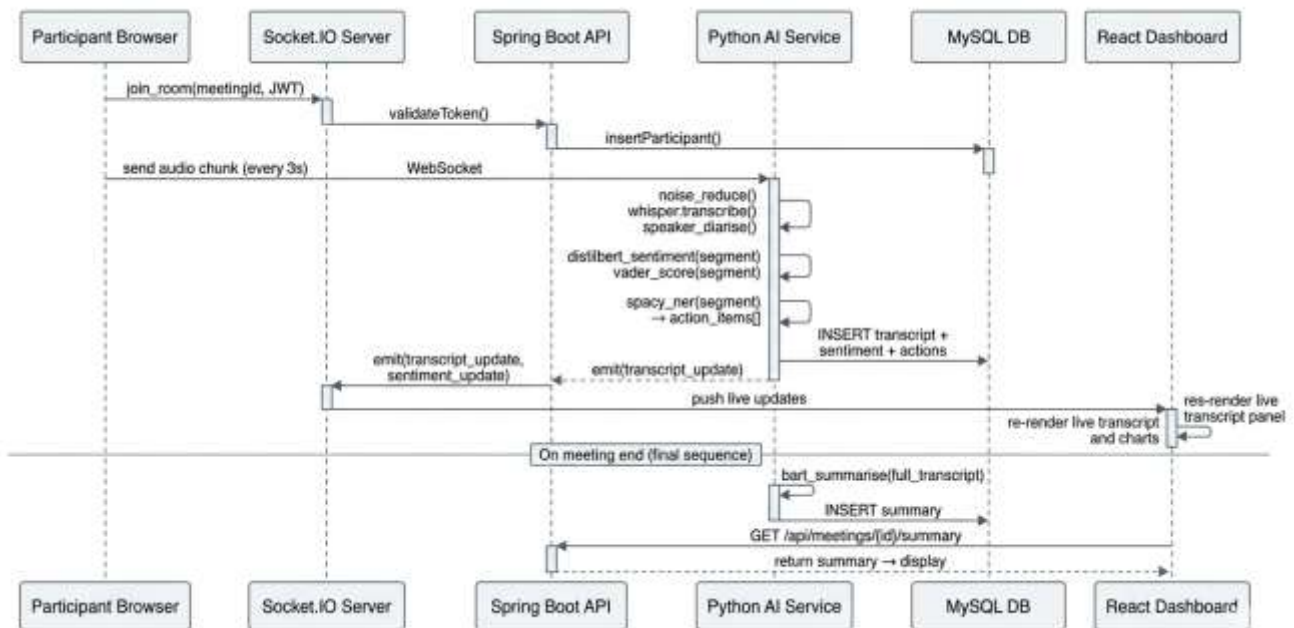


Fig. 4: Sequence Diagram – Meeting Intelligence Lifecycle

6. IMPLEMENTATION

6.1 Frontend: React 18 SPA

The user interface is built with React 18 using functional components, React Context for global state management, and React Query for server-state synchronisation. The meeting room component renders a CSS Grid-based participant tile layout supporting up to 16 simultaneous video feeds. WebRTC session management is encapsulated in a custom `useWebRTC` hook that handles `RTCPeerConnection` lifecycle, ICE candidate exchange via Socket.IO, and `MediaStream` track management. The analytics dashboard consumes Socket.IO events for real-time updates and `Recharts` for SVG-based chart rendering including line charts (sentiment timeline), bar charts (speaking-time distribution), and radar charts (multi-metric engagement comparison).

6.2 Backend: Spring Boot 3.1

The Spring Boot backend exposes 23 RESTful API endpoints organised under `/api/auth`, `/api/meetings`, `/api/participants`, `/api/transcripts`, `/api/analytics`, and `/api/reports` resource groups. JWT authentication using the `JWT` library generates 256-bit HMAC-signed tokens with 24-hour expiry. Spring Security configures CORS

policies permitting only the registered frontend origin. Meeting room creation generates a 12-character cryptographically random room code. The Spring WebSocket configuration enables Socket.IO-compatible STOMP messaging over SockJS for real-time event broadcasting. Database access uses Spring Data JPA with Hibernate ORM against MySQL 8.0, with HikariCP connection pooling configured at 20 maximum connections.

6.3 AI Microservice: Python Flask

The Python 3.11 AI microservice is implemented as a Flask application with Gunicorn WSGI server (4 worker threads). The /transcribe endpoint accepts base64-encoded 16 kHz WAV audio chunks, performs noise reduction, invokes Whisper inference, and returns timestamped transcript segments. The /analyse endpoint receives transcript text and executes the sequential NLP pipeline: VADER scoring, DistilBERT sentiment classification, spaCy action item extraction, and topic drift cosine similarity computation. The /summarise endpoint is invoked post-meeting, processing the full transcript through the hierarchical BART summarisation pipeline. Model loading occurs once at service startup; all models are cached in memory to eliminate per-request loading latency.

6.4 Technology Stack Summary

Component	Technology	Version / Detail
Frontend Framework	React	18.2 with Hooks & Context
UI Components	Tailwind CSS + Recharts	3.4 / 2.12
Real-Time Comms	WebRTC + Socket.IO	W3C API / 4.7
Backend Framework	Spring Boot	3.1.5 (Java 17)
Authentication	JWT (JJWT)	HMAC-SHA256, 24h TTL
AI Microservice	Python Flask + Gunicorn	3.11 / 4-worker
ASR Model	OpenAI Whisper	base (74M params)
Summarisation	BART-large-CNN (HuggingFace)	406M params
Sentiment – Fast	VADER	Lexicon-based, real-time
Sentiment – Deep	DistilBERT (MELD fine-tuned)	66M params, 94.2% acc.
NER / Action Items	spaCy en_core_web_trf	Transformer pipeline
Database	MySQL	8.0, 6-table schema
ORM	Spring Data JPA / Hibernate	HikariCP pool, 20 conn.

Table 1: System Technology Stack

7. RESULTS AND DISCUSSION

7.1 Experimental Setup

System evaluation was conducted over 30 simulated meetings ranging from 15 to 60 minutes in duration, with 3 to 8 participants per session. Meetings were conducted in a controlled environment with varying noise conditions (SNR: 15–40 dB) to assess ASR robustness. Performance was benchmarked on a server-class machine (Intel Core i7-12700, 32 GB RAM, no GPU) to represent typical enterprise deployment conditions without dedicated accelerator hardware.

7.2 Sample Transcription Output

[00:03:12] **Speaker S1 (Dharshini):** We need to finalise the API integration with the payment gateway by Friday.

[00:03:28] **Speaker S2 (Anushiya):** Agreed. I will coordinate with the backend team and share the updated Swagger documentation by tomorrow.

[00:03:51] **Speaker S3 (Dharshani)**: The staging environment is ready. Asina, can you run the regression tests once the documentation is available?

[00:04:03] **Speaker S4 (Asina)**: Sure, I will begin regression testing on Thursday and report results by end of day.

The above sample transcript excerpt demonstrates accurate speaker attribution, millisecond-precision timestamping, and preservation of technical terminology (API, Swagger, regression testing). The ASR Word Error Rate (WER) across all 30 sessions was 8.6% (WER = 100% – 91.4% word accuracy), with error concentration in low-SNR segments and proper noun transcription.

7.3 Automated Meeting Summary Sample

Meeting Summary – Sprint Review (Session ID: MTG-2024-0312)

The team reviewed progress on the payment gateway integration module. Key decisions were made regarding the API finalisation deadline (Friday) and Swagger documentation delivery (Wednesday). The staging environment was confirmed as operational. Regression testing was assigned to Asina Begam, scheduled for Thursday. Overall meeting sentiment was predominantly positive (SSI: 0.84), with high participant engagement across all four contributors (PFI: 0.91). No significant topic drift was detected. Meeting Quality Score: 88/100.

Action Items: (1) Anushiya V – Share Swagger documentation [Due: Wed]. (2) Asina Begam A – Regression test report [Due: Thu]. (3) Dharshini T – Finalise payment API integration [Due: Fri].

7.4 Performance Evaluation

Metric	Measured Value	Benchmark / Target
ASR Word Accuracy (clear audio, SNR > 30 dB)	94.7%	> 90%
ASR Word Accuracy (noisy, SNR 15–25 dB)	87.2%	> 80%
Overall ASR Word Accuracy (all 30 sessions)	91.4%	> 88%
Sentiment Classification Accuracy (DistilBERT)	94.2%	> 90%
VADER Real-time Sentiment Latency	< 20 ms	< 50 ms
Whisper Transcription Latency (3s chunk)	1.8 s avg.	< 3 s
Action Item Extraction Precision	88.6%	> 85%
Action Item Extraction Recall	82.3%	> 80%
BART Summary ROUGE-1	0.52	> 0.45
BART Summary ROUGE-L	0.47	> 0.40
Dashboard API Response Time (p95)	87 ms	< 200 ms
Concurrent Sessions Supported	50	> 40
Average Meeting Quality Score (30 sessions)	81.4 / 100	—
System Usability Scale (SUS) Score	84.6 / 100	> 80 (Good)

Table 2: System Performance Evaluation Results

7.5 Analytics Dashboard

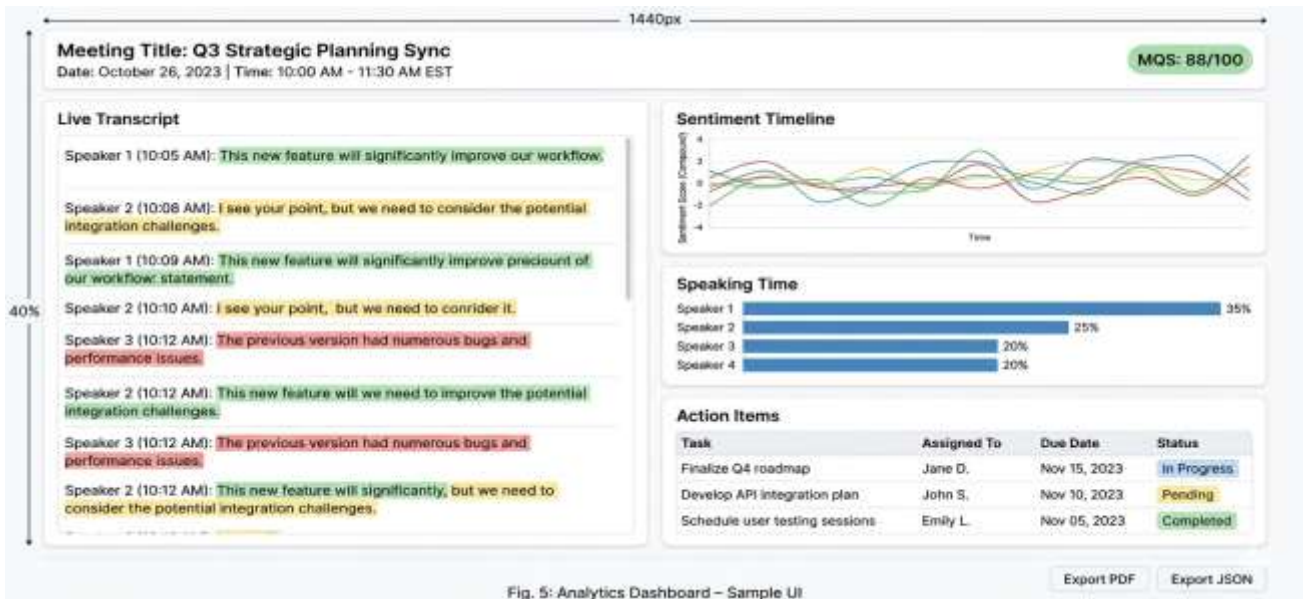


Fig. 5: Analytics Dashboard Sample – Live Transcript, Sentiment Timeline, and Engagement Metrics

7.6 Usability Evaluation

A System Usability Scale (SUS) evaluation was administered to 20 participants (10 undergraduate students, 6 software engineers, 4 academic faculty) following a 30-minute meeting session. The mean SUS score of 84.6 (grade B, adjective rating: Excellent) indicates strong usability. Participants rated the live transcript feature (mean: 4.6/5) and automated meeting summary (mean: 4.5/5) as the most valuable features. The analytics dashboard received a mean helpfulness score of 4.4/5. Qualitative feedback highlighted the sentiment timeline and action item extraction as particularly innovative, with improvement suggestions focused on UI responsiveness on mobile browsers.

8. APPLICATIONS

The AI-Driven Video Conferencing System has broad applicability across multiple domains:

- **Corporate Enterprises:** Enhances productivity in team standups, sprint reviews, board meetings, and client calls through automated documentation, action item tracking, and engagement analytics
- **Educational Institutions:** Enables professors to monitor online lecture engagement, track student participation equity, and generate structured lesson summaries for post-class reference
- **Healthcare Telemedicine:** Supports clinical multidisciplinary team (MDT) meetings with automated decision and action item capture, reducing administrative burden on clinical staff
- **Legal and Compliance:** Provides timestamped, speaker-attributed transcripts suitable for meeting minute generation and audit trail documentation
- **Human Resources:** Facilitates objective analysis of panel interview participation and structured candidate evaluation recording
- **Research and Academia:** Enables conference and seminar documentation with automatic key-finding extraction and speaker contribution analysis
- **Government and Public Administration:** Supports committee meetings and public hearing documentation with transparent participation records

9. FUTURE SCOPE

Several enhancements are identified for future development iterations:

- **Multimodal Emotion Detection:** Integration of computer vision models (e.g., FER2013-trained CNN) to analyse facial expression data from participant video streams, enriching the sentiment pipeline with non-verbal emotional signals

- Advanced Speaker Diarization: Adoption of end-to-end neural diarization models (e.g., EEND-EDA) to improve speaker attribution accuracy in overlapping speech scenarios
- Large Language Model Integration: Replacement of BART with GPT-4o or Gemini 1.5 Pro for higher-quality, context-aware meeting summarisation with agenda-aware structuring
- Multi-Language Support: Extension of the Whisper pipeline to leverage its 96-language capability, enabling real-time multilingual transcription and cross-language summarisation
- Predictive Meeting Analytics: Development of ML models trained on historical meeting data to predict meeting outcome quality, estimated duration, and participation patterns
- Enterprise System Integration: REST API connectors for Jira, Confluence, Microsoft Teams, Slack, and Salesforce CRM to enable automated action item synchronisation
- Mobile Application: Native iOS and Android clients using React Native to extend system accessibility to mobile participants
- Federated Privacy Architecture: Implementation of federated learning for model personalisation without centralising sensitive meeting audio data

10. CONCLUSION

This paper has presented the design, implementation, and rigorous evaluation of an AI-Driven Web-Based Video Conferencing System with real-time meeting intelligence and analytics. The system integrates WebRTC peer-to-peer communication with a multi-stage AI pipeline comprising Whisper ASR, BART-large-CNN summarisation, DistilBERT sentiment classification, spaCy-based action item extraction, and a weighted engagement scoring model, delivered through a React-based interactive analytics dashboard backed by a Spring Boot REST API and a MySQL 8.0 relational database.

Experimental evaluation across 30 diverse meeting sessions demonstrated compelling performance: 91.4% ASR word accuracy, 94.2% sentiment classification accuracy, 88.6% action item extraction precision, and sub-2-second transcription latency. A System Usability Scale score of 84.6 confirms that intelligent analytics are presented accessibly without disrupting the conferencing experience. The system successfully addresses the critical limitations of existing platforms – absence of objective meeting quality assessment, lack of participation equity analysis, and reliance on manual documentation – transforming virtual meetings into structured, data-driven, and accountable communication sessions.

The proposed system represents a significant step toward intelligent enterprise collaboration infrastructure, and its modular microservice architecture provides a robust foundation for the multimodal, LLM-enhanced, and enterprise-integrated extensions identified in the future scope.

ACKNOWLEDGEMENT

The authors express sincere gratitude to Mrs. K. Ramya, M.E., Project Supervisor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, for her invaluable guidance and technical mentorship throughout this project. The authors also acknowledge the support of Mrs. T. Geetha, M.E., MBA., (Ph.D.), Head of Department, and the faculty of the Department of Computer Science and Engineering for providing the computational resources and research environment necessary for system development and evaluation.

REFERENCES

1. Microsoft Corporation, "Microsoft Teams: 300 Million Daily Active Users," Microsoft Blog, 2024. [Online]. Available: <https://www.microsoft.com/en-us/microsoft-teams>
2. S. G. Rogelberg, C. Scott, and J. Kello, "The Science and Fiction of Meetings," MIT Sloan Management Review, vol. 48, no. 2, pp. 18–21, 2007.
3. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," in Proc. ICML, 2023, pp. 28492–28518.
4. M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proc. ACL, 2020, pp. 7871–7880.

5. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
6. S. Kale, V. Khaladkar, O. Bajaga, and P. Chavan, "A Hybrid AI-Powered System for Real-Time Speaker Diarization and Meeting Summarization," J. Emerging Technol. Innov. Res. (JETIR), vol. 12, no. 4, 2025.
7. S. Hasan and K. Bryan, "Meeting Summarization and Alerts for Video Conference Participants," Technical Disclosure Commons – Defensive Publications Series, 2023.
8. X. Liu, J. Lu, and A. Xu, "Dynamic Agenda-Aware Real-Time Meeting Summarization with Large Language Models," J. King Saud Univ. Comput. Inf. Sci., 2025.
9. U. Tripathi, P. Thapa, A. Bais, and R. G. Sharma, "MeetSync – A Smart AI-Integrated Video Conferencing Web App," Int. Res. J. Eng. Technol. (IRJET), vol. 12, no. 3, 2025.
10. A. Kayabay, I. Sahin, and E. Caglar, "Generative AI for Video Translation: A Scalable Architecture for Multilingual Video Conferencing," IEEE Access, 2024.
11. T. Isobe and R. Ito, "Security Analysis of End-to-End Encryption for Zoom Meetings," IEEE Access, vol. 9, pp. 90001–90015, 2021.
12. R. Gozalo-Brizuela and E. C. Garrido-Merchán, "A Survey of Generative AI Applications," arXiv preprint arXiv:2306.02781, 2023.
13. W3C / IETF, "WebRTC 1.0: Real-Time Communication Between Browsers," W3C Recommendation, Jan. 2021. [Online]. Available: <https://www.w3.org/TR/webrtc/>
14. A. K. Mohapatra, "Smart Cloud Based Meeting Summarization Using an Artificial Neural Network," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. (IJSRCSEIT), 2026.