MPJ MINI PROJECT REPORT

TITLE: AI-Powered Meeting Summarizer for Enhanced Corporate Productivity

Team Member:


| Sr.No. | Name | Roll No. | PRN |
|---|---|---|---|
| 1 | Ahad Shaikh | 33 | 1032222742 |
| 2 | Manoj Nayak | 22 | 1032221958 |
| 3 | Siddhi Murkute | 31 | 1032222655 |
| 4 | Vansh Patel | 26 | 1032222327 |
| 5 | Vishvesh Modcoicar | 9 | 1032220882 |


## 1. Abstract


Meetings are essential in corporate environments but often result in information overload, missed action items, and inefficient follow-ups. This project introduces an AI-powered Meeting Summarizer that automates transcription, summarization, and task extraction from meeting recordings. The system leverages OpenAI's Whisper for transcription, Google's Gemini API for summarization and deadline extraction, and Vaadin for a user-friendly interface. All processed data is stored in MongoDB for easy retrieval and task tracking. The solution ensures that key decisions, action items, and deadlines are systematically captured, improving accountability and productivity.


## 2. Introduction

In corporate settings, meetings consume significant time, yet critical details are often lost due to manual note-taking inefficiencies. Automating meeting summaries can drastically reduce human error and ensure actionable insights are captured. This project presents an end-to-end AI system that:

- Converts meeting recordings (audio/video) into structured text.
- Extracts key discussion points, action items, and deadlines.
- Provides a well-formatted summary with task assignments.
- Stores data in MongoDB for future reference.

## 3. Problem Statement

Manual meeting summaries are:

- Time-consuming – Employees spend extra hours reviewing recordings.
- Error-prone – Human transcription may miss critical details.
- Inconsistent – Different note-takers capture varying levels of detail.

- Task leakage – Action items and deadlines are frequently overlooked.

An automated AI summarizer solves these challenges by ensuring accuracy, consistency, and task accountability.

## 4. Objectives

- Develop a transcription module using Whisper for high-accuracy speech-to-text conversion.
- Implement Gemini API for summarization, action item extraction, and deadline detection.
- Design a Vaadin-based UI for easy upload and summary retrieval.
- Store and manage summaries in MongoDB for scalability.
- Ensure the system is scalable, secure, and user-friendly.

## 5. Literature Review

| Ref | Authors | Title | Venue / Year | Key Contributions |
|---|---|---|---|---|
| [1] | V. Rennard, G. Shang, J. Hunter, M. Vazirgiannis | Abstractive Meeting Summarization: A Survey | TACL, 2023 | - Comprehensive survey of abstractive meeting summarization<br>- Reviews datasets, encoder-decoder models, evaluation metrics<br>- Highlights challenges: dialogue structure, coherence, external knowledge integration |
| [2] | Z. Deng, S. Yoon, T. Bui, F. Dernoncourt, et al. | Aspect-based Meeting Transcript Summarization | IEEE Big Data, 2023 | - Two-stage summarization with weakly supervised classification<br>- Focus on aspect-specific summaries<br>- Improved performance on AMI dataset |
| [3] | Md T. R. Laskar, X.-Y. Fu, C. Chen, S. Bhushan TN | Building Real-World Meeting Summarization Systems using LLMs | arXiv, 2023 | - Evaluates GPT-4, PaLM-2, LLaMA-2 (open vs closed-source)<br>- Performance vs cost/privacy trade-offs<br>- Real-world applicability insights |
| [4] | L. P. Kumar, A. Kabiri | Meeting Summarization: A Survey of the State of the Art | arXiv, 2022 | - Surveys extractive, abstractive, hybrid techniques<br>- Challenges in informal speech, overlapping dialogue<br>- Dataset and evaluation discussion |

## 6. Tools and Technologies Used

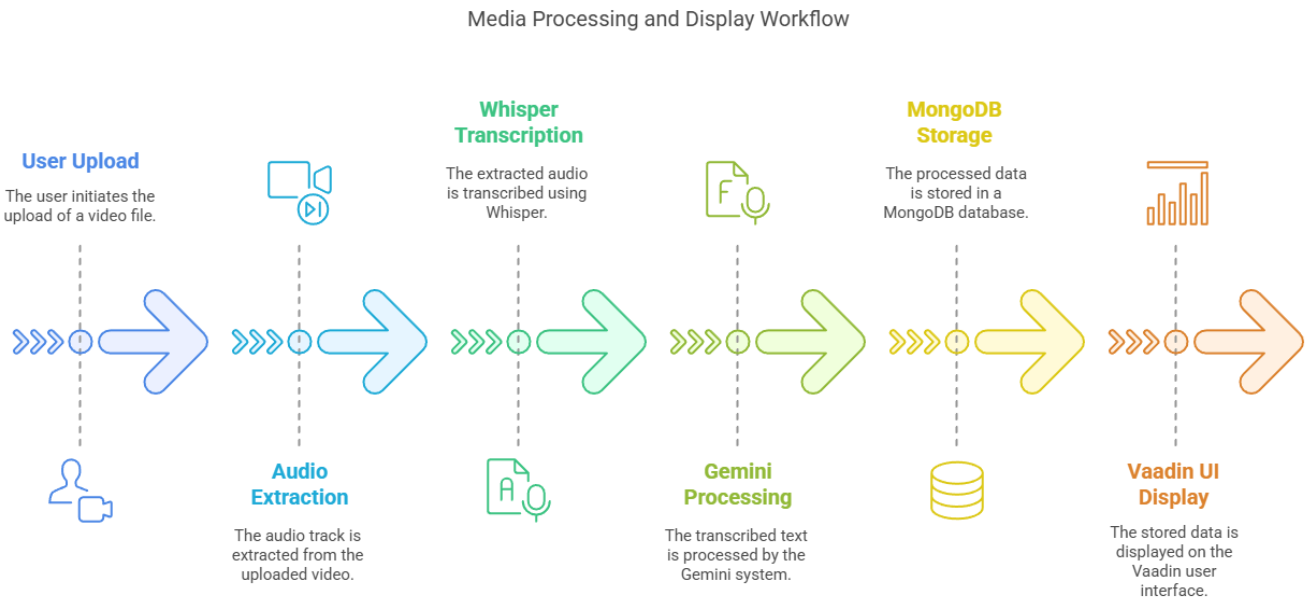
| Category | Technology |
|---|---|
| Transcription | OpenAI Whisper (Python) |
| Summarization | Google Gemini API |
| Frontend | Vaadin (Java-based web framework) |
| Database | MongoDB (NoSQL) |
| Backend | JAVA |

## 7. System Architecture

The system follows a modular pipeline:

- Input Module – User uploads a meeting recording (audio/video).
- Audio Extraction – If video, extract audio using moviepy/ffmpeg.
- Transcription – Whisper processes audio into text.
- AI Summarization – Gemini parses text, extracts key points, action items, and deadlines.
- Database Storage – Structured summary saved in MongoDB.
- Frontend Display – Vaadin UI renders summaries and tasks.

## 8. Workflow Diagram




## 9. Modules

- Video-to-Audio Conversion
  - Uses moviepy or ffmpeg to extract audio from video files.

    - Output: .mp3 file for transcription.
- Audio Transcription (Whisper)
    - Whisper model (medium-v2) transcribes audio with high accuracy.
    - Handles multiple speakers, background noise, and accents.
- Gemini-Powered Summarization & Task Extraction
    - Gemini API processes transcript with prompts like:
    - “Summarize this meeting, list action items, and extract deadlines.”
    - Returns structured JSON with:
    - Summary (concise meeting recap)
    - Action Items (tasks with assignees)
    - Deadlines (extracted dates)
- Vaadin Frontend
    - File upload interface.
    - Summary display panel.
    - Task management dashboard.
    - Built with Java for seamless integration with backend services.
- MongoDB Storage
    - Stores:
    - Raw transcripts.
    - Summarized outputs.
    - Action items with deadlines.
    - Enables querying past meetings by date, participant, or keyword.

## 10. Key Benefits Delivered:

- Elimination of Manual Note-Taking
    - Reduces human effort and errors by automating transcription and summarization.
    - Ensures no critical discussion points or action items are missed, even in lengthy or multi-speaker meetings.
- Enhanced Task Tracking & Accountability
    - AI-extracted action items with deadlines ensure follow-ups are systematically tracked.
    - MongoDB storage allows for easy retrieval, filtering, and reporting of past meeting outcomes.
- Boost in Corporate Productivity
    - Saves hours of manual work per meeting, allowing teams to focus on execution rather than documentation.
    - Real-time insights (with future live processing support) enable faster decision-making.
- Scalable & User-Friendly Design
    - Vaadin’s Java-based UI provides an enterprise-ready frontend with minimal latency.

- Modular architecture allows for future integrations (e.g., Slack, Microsoft Teams, CRM tools).

## 11. Novelty

- Hybrid AI Pipeline – Combines Whisper (transcription) + Gemini (summarization) for superior accuracy.
- Deadline Extraction – Unlike generic summarizers, this system explicitly detects and tracks deadlines.
- Vaadin UI – Unlike common React/Dash solutions, Vaadin provides a Java-based, enterprise-friendly interface.
- MongoDB Optimization – Efficient schema design for quick retrieval of meeting data.

## 12. Challenges Faced and Solutions

| Challenge | Solution |
|---|---|
| Whisper's slow processing | Optimized batch processing & GPU acceleration |
| Gemini's unstructured output | Custom prompt engineering for JSON responses |
| Speaker diarization (who said what) | Future integration with PyAnnote for speaker ID |
| Vaadin-Python integration | REST API bridge between backend and frontend |

## 13. Future Scope

- Real-time Summarization – Live meeting transcription & summarization.
- Speaker Identification – Assign action items to specific participants.
- Slack/Teams Integration – Auto-post summaries in collaboration tools.
- Advanced NLP – Sentiment analysis to gauge meeting tone.

## 14. Conclusion

The AI Meeting Summarizer represents a transformative solution for modern corporate environments by eliminating the inefficiencies of manual note-taking, enhancing accountability in task tracking, and significantly boosting workplace productivity. By integrating cutting-edge technologies—OpenAI's Whisper for high-accuracy transcription, Google's Gemini API for intelligent summarization and deadline extraction, Vaadin for a seamless and responsive user interface, and MongoDB for scalable and structured data storage—the system delivers a robust, end-to-end meeting intelligence platform.

By bridging the gap between meeting discussions and tangible outcomes, this tool redefines how organizations capture, analyze, and act on meeting data. Future expansions—such as real-time processing, speaker diarization, and sentiment analysis—will further solidify its role as an indispensable asset for agile, data-driven workplaces.

## 15. References


1. Rennard, V., Shang, G., Hunter, J., & Vazirgiannis, M. (2023). Abstractive Meeting Summarization: A Survey. Transactions of the Association for Computational Linguistics, 11, 861-884
2. Deng, Z., Yoon, S., Bui, T., Dernoncourt, F., Tran, Q. H., Liu, S., Zhao, W., Zhang, T., Wang, Y., & Yu, P. S. (2023). Aspect-based Meeting Transcript Summarization: A Two-Stage Approach with Weak Supervision on Sentence Classification. Proceedings of the IEEE International Conference on Big Data, 636-645
3. MdTahmidRahmanLaskar, Xue-Yong Fu, Cheng Chen, Shashi Bhushan TN,Building Real-World Meeting Summarization Systems using Large Language Models: A Practical Perspective
4. Kumar, L. P., & Kabiri, A. (2022). Meeting Summarization: A Survey of the State of the Art. arXiv preprint arXiv:2212.08206.