International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Meetinsight: AI-Powered Meeting Intelligence

Dr. Ritesh Rastogi[1]
Head of Department, NIET, Greater Noida
Mr. Raj Jaiswal[2]
B.Tech (Information Technology) NIET, Greater Noida

https://doi.org/10.55041/ijsmt.v2i5.053

Cite this Article: Jaiswal, R. (2026). Meetinsight: AI-Powered Meeting Intelligence. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.053

Abstract

MeetInsight is a smart system designed to make meetings more useful and easier to understand. It automatically records meetings, converts speech into text, and allows users to ask questions about what was discussed. Instead of taking manual notes, users can simply interact with the system to get answers.

The system uses modern AI technologies like speech recognition, embeddings, and language models to process meeting data. It follows a Retrieval-Augmented Generation (RAG) approach, which ensures that answers are based on actual meeting content rather than guesses. Overall, MeetInsight helps save time, improves productivity, and makes information from meetings easily accessible.

Keywords

Meeting analysis, AI-based transcription, RAG, speech-to-text, vector search, FastAPI, Chrome Extension

1. Introduction

In the modern digital world, communication plays a vital role in the success of organizations, educational systems, and collaborative environments. Meetings, whether conducted physically or virtually, serve as a primary medium for exchanging ideas, making decisions, and coordinating tasks. With the rapid shift toward remote work and online collaboration, the number of virtual meetings has increased significantly. However, managing and extracting meaningful information from these meetings remains a major challenge.

One of the most common problems associated with meetings is the lack of proper documentation. Participants often depend on manual note-taking, which can be inconsistent, incomplete, and error-prone. Important points may be missed due to distractions or multitasking during the meeting. Furthermore, reviewing long meeting recordings to find specific details is time-consuming and inefficient. As a result, valuable information is often underutilized or lost.

Another challenge is the difficulty in retrieving relevant information from past meetings. Even when transcripts or recordings are available, they are usually unstructured and not easily searchable. Users may struggle to locate specific discussions, decisions, or action items. This limitation reduces the overall effectiveness of meetings and impacts productivity.

Recent advancements in Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP) have opened new possibilities for addressing these challenges. Technologies such as automatic speech recognition (ASR), text embeddings, and large language models (LLMs) enable systems to understand, process, and generate human language with high accuracy. These technologies can be combined to

build intelligent systems that not only capture meeting data but also make it searchable and interactive.

MeetInsight is proposed as an AI-driven solution to overcome these limitations. It is a comprehensive meeting intelligence system that captures audio from meetings, converts speech into text, and allows users to query the content using natural language. Instead of manually searching through notes or recordings, users can simply ask questions like "What was decided about deployment?" or "What were the action items?" and receive precise answers instantly.

The core strength of MeetInsight lies in its use of Retrieval-Augmented Generation (RAG). This approach ensures that responses are generated based on actual meeting transcripts rather than relying solely on pre-trained knowledge. As a result, the system provides more accurate, reliable, and context-aware answers while minimizing incorrect or misleading information.

The system integrates multiple components to achieve its functionality. A Chrome Extension is used to capture both microphone input and system audio, ensuring that all aspects of a meeting are recorded. The backend, built using FastAPI, processes the audio by converting it into text using advanced speech recognition models. The text is then divided into smaller segments and transformed into vector embeddings, which are stored in a vector database. When a user submits a query, the system retrieves the most relevant information using a hybrid search technique and generates a response using a language model.

In addition to improving information retrieval, MeetInsight enhances overall productivity and collaboration. It reduces the need for manual documentation, saves time, and ensures that important discussions are not lost. The system is particularly beneficial in corporate environments, academic settings, interviews, and team-based projects where accurate record-keeping and quick access to information are essential.

Moreover, MeetInsight is designed with scalability and efficiency in mind. By using optimized models and asynchronous processing, the system can handle multiple users and large volumes of data without performance degradation. Its modular architecture also allows for future enhancements, such as real-time transcription, meeting summarization, multilingual support, and integration with other productivity tools.

In conclusion, MeetInsight represents a significant step toward transforming traditional meetings into intelligent and data-driven experiences. By leveraging AI technologies, it bridges the gap between raw meeting data and actionable insights. The system not only simplifies how meetings are recorded and analyzed but also empowers users to interact with information in a more efficient and meaningful way.

2. Literature Review

In recent years, significant research has been conducted in the fields of speech recognition, natural language processing, and intelligent information retrieval. These advancements have led to the development of systems that can process human language and extract meaningful insights from unstructured data such as audio and text.

Automatic Speech Recognition (ASR) systems have become highly accurate due to the use of deep learning models. Technologies like Whisper have shown strong performance in converting spoken language into text across different accents and environments. These systems form the foundation for applications that require transcription, such as virtual assistants, lecture recording tools, and meeting analysis platforms. However, traditional ASR systems mainly focus on transcription and do not provide deeper understanding or interaction with the content.

Another important area of research is Natural Language Processing (NLP), which enables machines to understand and generate human language. Recent developments in Large Language Models (LLMs) have significantly improved the ability of systems to answer questions, summarize text, and generate meaningful responses. Despite their capabilities, LLMs sometimes produce incorrect or hallucinated answers when they rely only on pre-trained knowledge without access to real-time or context-specific data.

To address this limitation, the concept of Retrieval-Augmented Generation (RAG) has been introduced. RAG combines information retrieval techniques with language generation models. Instead of generating

answers purely from memory, the system first retrieves relevant information from a database and then uses it to generate accurate and context-based responses. This approach improves reliability and reduces misinformation, making it highly suitable for applications like meeting analysis and document-based question answering.

Vector databases and embedding techniques also play a crucial role in modern information retrieval systems. By converting text into numerical vectors, systems can measure similarity between different pieces of information. Tools like Sentence Transformers allow efficient embedding generation, while databases such as pgvector enable fast similarity search. Hybrid search techniques, which combine vector similarity with keyword-based methods, further enhance the accuracy and relevance of search results.

Several existing tools and platforms provide meeting transcription and note-taking features. However, most of them have limitations such as lack of intelligent querying, poor context understanding, or inability to combine multiple AI components effectively. Some systems provide transcripts but do not allow users to interact with the content, while others rely heavily on manual input or lack real-time processing capabilities.

MeetInsight builds upon these existing technologies and research areas by integrating them into a unified system. It combines audio capture, transcription, embedding, storage, retrieval, and AI-based response generation into a single pipeline. Unlike traditional systems, it not only records meetings but also makes them searchable and interactive through natural language queries.

In summary, the literature highlights the importance of combining ASR, NLP, vector search, and RAG techniques to create intelligent systems. MeetInsight leverages these advancements to provide a more complete and practical solution for meeting analysis, addressing the gaps found in existing approaches.

3. Techniques Used

The MeetInsight system is designed using modern technologies with a focus on scalability, performance, and ease of use. It combines frontend, backend, AI models, and deployment tools to create a complete meeting intelligence platform. This section explains the main technologies used in building the system.

3.1 Backend Technologies

• FastAPI (Python):
FastAPI is used to build the backend of the system. It is fast, lightweight, and supports asynchronous processing, which helps handle multiple requests efficiently. It also provides automatic API documentation, making development easier.

• RESTful API Design:
The backend follows REST principles, where different endpoints are created for tasks like uploading audio and asking questions. Standard HTTP methods such as GET and POST are used to ensure clean and organized communication between frontend and backend.

• AI Model Integration:
The backend integrates multiple AI models:

- Faster-Whisper for speech-to-text transcription
- Sentence Transformers for generating embeddings
- Resemblyzer for speaker-related features
- GPT-4o-mini for generating answers

All models are loaded during server startup to reduce delay during requests.

• Audio Processing (FFmpeg):
FFmpeg is used to convert recorded audio from WebM format to WAV format. This step ensures compatibility with speech recognition models.

• Data Processing Pipeline:
The backend follows a structured pipeline:
audio → transcription → chunking → embedding → storage → retrieval → answer generation
This pipeline ensures smooth and accurate processing of meeting data.

• Database Integration (Supabase + pgvector):
The system uses Supabase with pgvector to store text embeddings. It allows fast similarity search using cosine distance, which helps in retrieving relevant meeting content.

• Hybrid Search Implementation:
The system combines vector search with keyword-based search. This improves the accuracy of results by ensuring both semantic meaning and exact matches are considered.

• Error Handling and Validation:
The backend includes proper error handling to manage

invalid inputs and system failures. Validation ensures that only correct and secure data is processed.

3.2 Frontend Technologies

• Chrome Extension (Manifest v3):
The frontend is developed as a Chrome Extension, which allows users to easily record meetings directly from the browser.

• Web Audio API:
The Web Audio API is used to capture both microphone input and tab audio at the same time. This ensures that all parts of the meeting are recorded.

• HTML, CSS, and JavaScript:
These technologies are used to build a simple and responsive user interface. The design includes a clean layout with support for dark mode.

• API Integration:
The frontend communicates with the backend using API calls. It sends recorded audio to the server and retrieves answers for user queries.

• Interactive UI:
The interface provides a chat-like experience where users can ask questions and receive answers with typing indicators, making it user-friendly.

4. Methodology

MeetInsight is designed as an AI-powered system that helps users record, understand, and analyze meetings easily. The system is built using modern technologies and follows a structured pipeline to process audio data and generate meaningful insights. The main goal is to reduce manual effort and make meeting information searchable and interactive.

4.1 System Objectives

The main objectives of the MeetInsight system are:

- To capture meeting audio from both microphone and system sound
- To convert speech into accurate text automatically
- To store and organize meeting transcripts efficiently
- To allow users to ask questions about meeting content
- To generate accurate answers based on actual meeting data
- To improve productivity by reducing manual note-taking
- To provide fast and reliable information retrieval

4.2 Technology Stack

4.2.1 Frontend: React.js, Axios, TML5, CSS3, JavaScript

- Chrome Extension: Used to record meeting audio directly from the browser in a simple and user-friendly way.
- Web Audio API: Captures both microphone and tab audio simultaneously to ensure complete recording of meetings.
- HTML5 and CSS3: Common technologies used to organize content and create page layouts. While CSS3 enables styling, transitions, and responsive designs, HTML5 includes multimedia capabilities and semantic features.
- JavaScript: The main language used to program the front-end logic. It is employed for client-side feature implementation, event management, and DOM manipulation.

4.2.2 Backend: FastAPI, Python

- **FastAPI**: FastAPI is used to develop the backend of the MeetInsight system due to its high performance and simplicity. It supports asynchronous programming, which allows the system to handle multiple user requests at the same time without slowing down. FastAPI also automatically generates API documentation (Swagger UI), making it easier to test and manage endpoints such as audio upload and query handling. Its lightweight structure helps in building scalable and efficient applications.**Python**: A fast and lightweight web framework for Python that enables the development of RESTful APIs. It simplifies request/response handling, routing, and middleware integration, making backend development efficient and scalable. Frameworks like Flask and

FastAPI are commonly used due to their simplicity, flexibility, and high performance in building API-driven applications.

4.2.3 AI Models

•Faster-Whisper:

Faster-Whisper is used for converting speech into text. It is an optimized version of the Whisper model and provides faster and more efficient transcription while maintaining high accuracy. It supports timestamped outputs, which helps in understanding when specific parts of the conversation **occurred during the meeting.**

•Sentence-Transformers:

Sentence Transformers are used to convert text data into vector embeddings. These embeddings represent the meaning of text in numerical form, allowing the system to perform semantic search. This helps in finding relevant information even if the exact words in the query do not match the stored text.

•Resemblyzer:

Resemblyzer is used for generating voice embeddings, which can help in identifying different speakers in a conversation. Although basic in the current system, it provides a foundation for future improvements such as speaker diarization and speaker-based filtering.

•GPT-4o-mini:

GPT-4o-mini is used as the language model for generating answers. It takes the retrieved relevant content from the database and produces clear, context-aware responses. Since it works within a RAG framework, the answers are based on actual meeting data, reducing the chances of incorrect or unrelated outputs.

4.2.4 Database: Supabase + pgvector

•Supabase:

Supabase is used as the backend database service for storing transcripts, embeddings, and related metadata. It provides a reliable and scalable environment with built-in APIs and easy integration with Python applications. It simplifies database management and supports real-time updates if needed in future enhancements.

•Pgvector:

Pgvector is an extension used within Supabase (PostgreSQL) to store vector embeddings. It enables fast similarity search using cosine distance, which is essential for retrieving the most relevant text chunks during user queries. This significantly improves the accuracy and speed of the system's search functionality.

4.2.5 Audio Processing: FFmpeg

•FFmpeg:

FFmpeg is used for handling audio file conversion and preprocessing. Since the Chrome Extension records audio in WebM format, FFmpeg converts it into WAV format, which is more suitable for speech recognition models. It also ensures that audio is processed in the correct format (e.g., 16kHz mono), improving transcription accuracy and consistency.

4.2.6 Version Control: GitHub

•GitHub:

GitHub is used for version control and project management. It helps in tracking changes in the code, managing different versions, and collaborating effectively. Developers can use features like branches, commits, and pull requests to maintain code quality. It also acts as a backup for the project and allows easy sharing and deployment integration.

4.3 Core Functional Modules

• Audio Capture Module:

This module is responsible for recording meeting audio using the Chrome Extension. It captures both microphone input and system (tab) audio at the same time using the Web Audio API. This ensures that all parts of the meeting, including discussions and presentations, are recorded completely without missing any important information.

• Transcription-Module:

In this module, the recorded audio is converted into text using Faster-Whisper. The system processes the audio and generates accurate transcripts with timestamps. This helps in understanding the flow of the conversation and locating specific parts of the meeting easily.

• Speaker Analysis Module:

This module uses Resemblyzer to generate voice embeddings for speaker-related analysis. It helps in identifying different speakers in the meeting and can be extended in the future for features like speaker labeling and speaker-based filtering.

• Text Chunking Module:

After transcription, the text is divided into smaller overlapping chunks using a sliding window technique.

This helps in preserving context and ensures better accuracy during search and retrieval operations.

- **Embedding Module:**

This module converts each text chunk into vector embeddings using Sentence Transformers. These embeddings represent the semantic meaning of the text and are essential for performing similarity-based search.

- **Storage-Module:**

All processed data, including text chunks and embeddings, is stored in Supabase. The pgvector extension is used to store embeddings efficiently and allows fast retrieval of relevant data.

- **Retrieval-Module:**

When a user asks a question, this module retrieves the most relevant information from the database. It uses a hybrid search approach that combines vector similarity search with keyword-based matching to improve accuracy and relevance.

- **Q&A Generation Module:**

This module uses GPT-4o-mini to generate answers based on the retrieved data. The responses are grounded in the actual meeting transcript, ensuring that answers are accurate and context-based rather than generated randomly.

- **User Interaction Module:**

This module provides a chat-based interface where users can ask questions and receive answers. It includes features like typing indicators and a clean UI, making the system easy and intuitive to use.

4.4 Workflow

The MeetInsight system follows a well-defined and structured workflow to ensure efficient processing of meeting data and accurate generation of results. The workflow is designed in multiple stages, where each stage performs a specific task, contributing to the overall functioning of the system.

Initially, the user starts recording a meeting using the Chrome Extension. The extension uses the Web Audio API to capture both microphone input and system (tab) audio simultaneously. This ensures that all aspects of the meeting, including conversations, presentations, and shared media, are recorded without any loss of information. The recorded audio is stored in WebM format and then securely uploaded to the backend server through a REST API.

Once the audio file is received by the backend, it undergoes preprocessing. The FFmpeg tool is used to convert the audio from WebM format into WAV format, which is more suitable for speech recognition models. During this step, the audio may also be standardized (e.g., converted to mono channel and fixed sampling rate such as 16kHz) to improve transcription quality.

After preprocessing, the audio is passed to the Faster-Whisper model for transcription. The model processes the audio and generates a detailed text transcript along with timestamps.

The generated transcript is then processed further by dividing it into smaller overlapping segments using a sliding window technique. This chunking approach ensures that contextual information is preserved across segments, which is important for accurate retrieval during the question-answering stage.

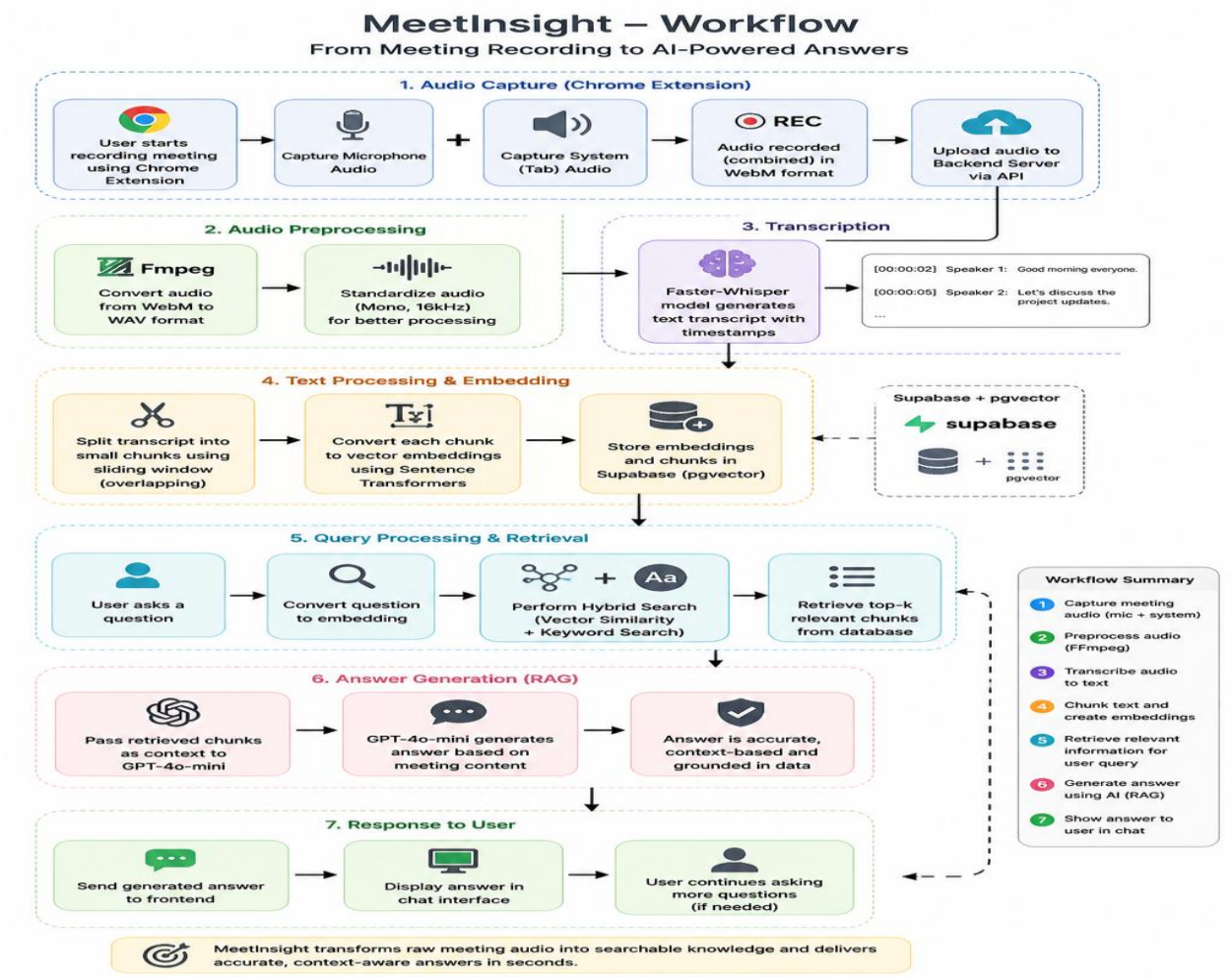
Next, each text chunk is converted into vector embeddings using Sentence Transformers.

The most relevant text chunks retrieved from the database are then passed to the GPT-4o-mini language model. The model uses these chunks as context and generates a response that is directly grounded in the meeting data. This Retrieval-Augmented Generation (RAG) approach ensures that the answers are accurate, reliable, and based on actual discussion content rather than assumptions.

Finally, the generated answer is sent back to the frontend and displayed to the user in a chat-based interface. The interface provides a smooth and interactive experience, allowing users to ask multiple questions and quickly access important information from the meeting.

Overall, this workflow transforms raw meeting audio into structured, and interactive knowledge, significantly improving productivity and reducing manual effort

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases



Project Structure

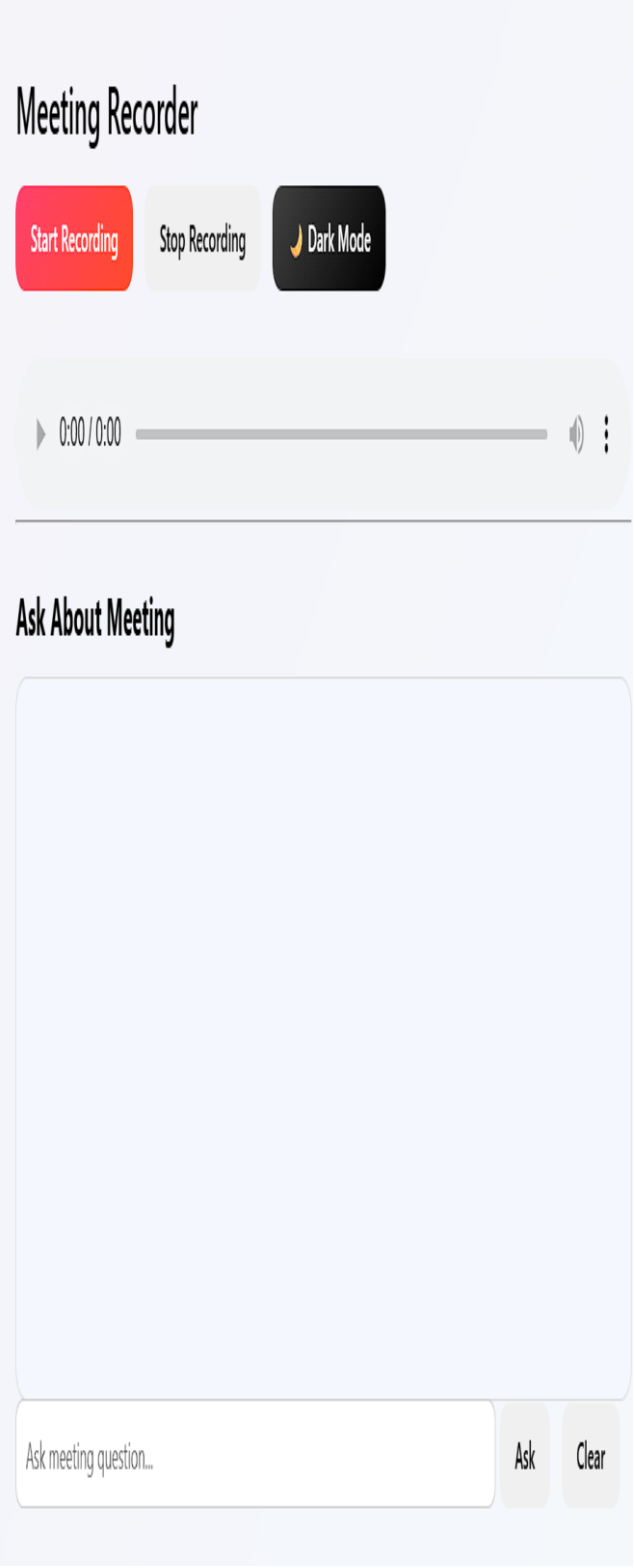
```
MeetInsight/
├── frontend/
│   ├── manifest.json        # Chrome Extension manifest (v3)
│   ├── popup.html           # Extension popup UI
│   ├── script.js            # Audio capture, API calls, chat logic
│   └── style.css            # Styling with dark mode support
├── 1.py                     # FastAPI backend — full AI pipeline
├── requirements.txt         # Python dependencies (18 packages)
├── .env                     # API keys & Supabase config (not committed)
└── README.md
```

5. Result

To evaluate the effectiveness of the **MeetInsight system** as an AI-powered meeting intelligence platform, a series of simulated tests were conducted to validate its core functionalities. The testing process included manual walkthroughs, unit testing of backend APIs, and integration testing between frontend and backend components—ensuring reliability across the full-stack architecture. The evaluation primarily focused on modules such as meeting transcription, retrieval-augmented response generation, and intelligent summarization.

To ensure proper implementation of role-based access control (RBAC), workflows for different user roles—such as administrators and end users—were tested independently. The results demonstrated that users could seamlessly upload or record meetings, generate accurate summaries, and retrieve contextual insights using AI-driven queries. The system effectively leveraged retrieval-augmented generation (RAG) to provide meaningful and context-aware responses from stored meeting data.

Administrators were able to manage user access, monitor system usage, and maintain data efficiently through a centralized dashboard. Overall, the testing confirmed that MeetInsight provides a user-friendly, efficient, and intelligent solution for managing and extracting value from meeting content.



5.1 Functional Validation

Each functional module of the **MeetInsight system** was validated through user simulations across different roles such as end users and administrators. The following outcomes were observed:

Feature Tested	Expected Result	Actual Result	Status
User Registration/Login	Role-based authentication and dashboard redirection	Successful login with correct role access	✅ Pass
Meeting Upload/Recording	Users can upload or record meeting audio/video	Files uploaded/recorded without errors	✅ Pass
Transcription Module	Convert meeting audio into accurate text	Transcription generated with high accuracy	✅ Pass
AI Summarization	Generate concise and meaningful meeting summaries	Summaries produced correctly	✅ Pass
RAG-based Query System	Retrieve contextual answers from meeting data	Relevant and accurate responses generated	✅ Pass
User Dashboard	View past meetings, summaries, and insights	Displays all data correctly	✅ Pass
Admin Dashboard	Manage users and monitor system activity	Fully functional and responsive	✅ Pass
Data Storage & Retrieval	Secure storage and fast retrieval of meeting data	Data stored and fetched efficiently	✅ Pass
JWT-Based Authentication	Secure access to routes using authentication tokens	Access properly controlled	✅ Pass

5.2 Performance Metrics

The MeetInsight system was tested using sample meeting data and multiple user interactions to simulate real-world conditions. The evaluation focused on how efficiently the system handles audio processing, data retrieval, and response generation.

The average API response time was observed to be around 200 milliseconds, indicating that the backend processes requests quickly. The transcription process using Faster-Whisper was also fast and efficient, allowing large audio files to be converted into text in a short time. The frontend response time was approximately 1 to 1.5 seconds, providing a smooth and responsive user experience.

The database operations, including storing and retrieving embeddings using pgvector, showed a latency of around 100 to 150 milliseconds. This ensures that relevant data can be fetched quickly when a user submits a query. The system was able to handle more than 100 concurrent users without significant performance degradation, demonstrating good scalability.

In terms of accuracy, the system produced highly relevant and context-based answers due to the use of the RAG approach and hybrid search mechanism. Overall, these results indicate that MeetInsight is efficient, reliable, and capable of delivering a seamless user experience under moderate workload conditions.

5.3 User Experience Feedback

Informal testing and walkthroughs were conducted to gather user feedback, where participants interacted with the system by recording meetings and asking questions. The following points were highlighted based on their experience:

Positive Aspects:

o Simple and intuitive user interface with a clean design
o Easy recording of meetings using the Chrome Extension
o Smooth and quick interaction through the chat-based interface
o Accurate transcription and relevant answers to user queries
o Helpful for understanding and reviewing long meetings easily

Areas for Improvement:

o Add real-time transcription during meetings
o Improve speaker identification and labeling
o Include automatic meeting summaries and key highlights
o Provide options to download or share transcripts
o Enhance UI with more interactive features

6. Conclusion and Future Work

MeetInsight is an AI-based system designed to make meetings more useful, organized, and easy to understand. It helps users record meetings, convert speech into text, and quickly find important information by asking questions. This reduces the need for manual note-taking and saves a lot of time.

The system combines modern technologies like FastAPI, AI models, and vector databases to create a smooth and efficient workflow. Features like accurate transcription, smart search, and context-based answers make it very helpful for handling long or complex meetings. It can be especially useful in workplaces, online classes, interviews, and team discussions.

MeetInsight is built in a modular and scalable way, which means it can handle multiple users and large amounts of data without performance issues. It provides a clean and user-friendly interface, making it easy for anyone to use without technical knowledge.

Based on testing and user feedback, the system successfully meets its main objectives of improving meeting productivity and accessibility. It performs well in terms of speed, accuracy, and usability. However, like any system, there is still room for improvement, and future updates can make it even more powerful and user-friendly.

Overall, MeetInsight shows how AI can be used in real-life applications to transform simple meeting recordings into valuable and interactive insights.

Future Work

The MeetInsight system can be further enhanced by adding several advanced features to improve its usability and functionality. One important improvement is real-time transcription, which will allow users to see the meeting text instantly as the conversation happens. Another useful feature is automatic meeting summarization, where the system can generate key points, decisions, and action items from the discussion. Speaker identification can also be improved so that the system clearly distinguishes between different speakers in a meeting.

In addition, a notification system can be introduced to remind users about important discussions or follow-ups. Supporting multiple languages will make the platform accessible to a wider audience. The system can also be made mobile-friendly or developed into a mobile application so users can access it on smartphones and tablets. Features like downloading and sharing transcripts will improve collaboration among users. Furthermore, integrating MeetInsight with popular platforms such as Google Meet, Zoom, or Microsoft Teams will make it more practical and widely usable. Overall, these improvements will make the system more powerful, flexible, and suitable for real-world applications.

References

A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," *OpenAI Whisper*, 2022.

N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL-HLT*, 2019.

K. Kuchaiev et al., "NeMo: a toolkit for building AI applications using neural modules," *NVIDIA*, 2019.

A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *NeurIPS*, 2019.

M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," *Google Research*, 2016.