Volume 10, Issue 4, April – 2025 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

Automating Meeting Management: An AI-Driven Web-Based Meeting Management System

Vaibhavi Ladhe[1]; Suraj Magdum[2]; Dhruv Mahajan[3]; Eeshan Malwandikar[4]; Nikhil Dhavase[5]

[1,2,3,4,5] Information Technology, MMCOE, Savitribai Phule Pune University, Karve Nagar, Pune, India

Publication Date: 2025/05/13

Abstract: The Meeting Manager Application is a web-based tool designed to simplify the organization, execution, and documentation of meetings. The application offers a comprehensive suite of features, including the ability to create meetings, upload relevant documents, manage discussion points, generate agendas, transcribe meeting recordings, and produce detailed meeting summaries. It leverages advanced technologies such as Flask for the web framework, SQLAlchemy for database management, ChromaDB for vector storage, and Google's Gemini Large Language Model (LLM) for natural language processing tasks. The system integrates state-of-the-art machine learning techniques, utilizing Sentence Transformers to generate text embeddings, which are stored and queried for similarity search in ChromaDB. By automating key aspects of meeting management, the application improves efficiency and reduces manual effort. This project demonstrates how AI-driven solutions can streamline workflow processes in corporate settings. Future enhancements include support for multi-user collaboration, scalability, and real-time meeting analysis for enhanced user experience.

How to Cite: Vaibhavi Ladhe; Suraj Magdum; Dhruv Mahajan; Eeshan Malwandikar; Nikhil Dhavase (2025). Automating Meeting Management: An AI-Driven Web-Based Meeting Management System. *International Journal of Innovative Science and Research Technology*, 10(4), 3358-3364. https://doi.org/10.38124/ijisrt/25apr1982

I. INTRODUCTION

In today's dynamic corporate and academic settings, meetings are essential for collaboration, strategic planning, and decision-making. However, traditional meeting management, which relies on manual agenda preparation, documentation, and follow-up tracking, poses significant challenges. These manual processes are time-consuming and error-prone, often resulting in critical information being overlooked or miscommunicated. As organizations scale, these inefficiencies become even more problematic, hindering productivity and reducing the effectiveness of collaborative efforts.

Recent advancements in artificial intelligence (AI) and natural language processing (NLP) have introduced new opportunities to automate meeting-related tasks. Large language models (LLMs) and vector storage solutions now enable sophisticated capabilities, such as automated agenda creation, real-time transcription, and summarization. AI-driven meeting management tools can analyze and process large amounts of text and speech data, helping teams document, organize, and reference key points without the need for extensive manual intervention. These technologies have the potential to transform meeting management by enhancing accuracy, reducing workload, and supporting better decision-making through comprehensive and accessible documentation.

The Meeting Management System presented in this paper harnesses the power of these advanced AI technologies to create a comprehensive, web-based solution for efficient meeting documentation. By integrating Google's Gemini LLM for transcription and agenda generation, and using ChromaDB for vector storage of meeting content, the system automates the creation of structured agendas, transcriptions, and summaries. This approach improves documentation accuracy and saves time, making it ideal for organizations that require reliable, scalable meeting management solutions. The system's intuitive user interface allows users to seamlessly create and manage meetings, upload relevant documents, and access AI-generated summaries, setting a new standard for productivity and efficiency in meeting management.

II. RELATED WORK

A variety of recent studies have investigated ways to enhance meeting management through the integration of artificial intelligence (AI) and natural language processing (NLP). The growing reliance on virtual and automated systems has driven significant advancements in this domain, with researchers aiming to streamline tasks such as note-taking, transcription, agenda creation, and meeting summarization..

A foundational study [1] explores NLP-driven summarization methods that automatically extract key points and decisions from meeting discussions, reducing the need

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

for manual note-taking. This approach leverages sentence embeddings and machine learning models to transform transcripts into concise, relevant summaries. Such methods are foundational for AI-driven meeting managementapplications, as they ensure essential information is retained without overwhelming users with unnecessary details..

For effective information retrieval within meeting contexts, vector embeddings have proven especially valuable. Paper [2] highlights the use of vector representations to improve semantic retrieval, crucial for systems that handle extensive or unstructured data, such as meeting recordings and documents. Techniques like Sentence Transformers enhance retrieval accuracy by creating meaningful vector embeddings, allowing for similarity-based searches. In meeting management, this enables users to easily find documents or previous discussion points related to current topics, enhancing both continuity and efficiency in recurring meetings..

The automation of agendas and action items has also been a significant focus in AI research for meeting management. In [3], researchers examine how AI-powered meeting assistants can automatically generate agendas based on user inputs or historical meeting data, with automated tracking of action items to ensure follow-through. By streamlining agenda creation and task management, these systems aim to alleviate administrative burdens and allow participants to focus on core discussions.

In addition to NLP models, vector storage databases like ChromaDB have gained attention for their ability to store and quickly retrieve large volumes of embedded data. Study [4] demonstrates ChromaDB's utility in handling vector embeddings generated from meeting documents, enabling rapid and contextually accurate searches within the meeting management system. This technology supports NLP-based similarity searches, which are particularly beneficial in automating agenda creation by identifying recurring discussion topics or referencing relevant past documents.

Furthermore, the practical deployment of AI-based meeting management systems has raised concerns regarding scalability, security, and user experience. Research in [5] discusses scalable architectures for meeting systems capable of handling high data volumes and user traffic, emphasizing the importance of robust system design. Another study [6] highlights best practices for securing sensitive meeting data through encryption and access control, which are crucial for systems handling confidential or sensitive organizational information.

The automation of meeting management tools has been an area of significant research. In [7], the authors provide a comprehensive review of automated meeting management tools, highlighting the various software systems designed for scheduling, document management, and transcription. These tools, similar to the Meeting Manager Application, offer features such as agenda generation, note-taking, and follow-up actions. The paper also addresses challenges such as transcription accuracy, handling diverse accents, and real-time integration in collaborative environments, which are key areas for improvement in such systems. The use of large language models (LLMs) for meeting transcription and summarization has also been explored. In [8], researchers discuss how LLMs like GPT-3 can be leveraged for automatic meeting transcription, summarization, and sentiment analysis. The study examines the benefits and limitations of using these models in real-time meeting contexts, noting how LLMs can reduce manual effort in transcription and improve document accessibility. This work is relevant as the Meeting Manager Application integrates Google's Gemini LLM for similar NLP tasks.

Furthermore, the application of vector databases and embedding-based information retrieval in NLP has been a topic of research. In [9], the authors investigate the use of ChromaDB for storing and querying embeddings in information retrieval systems. By representing text as vectors, these systems enable efficient similarity search, making them particularly useful for document retrieval and recommendation tasks. This is highly relevant to the Meeting Manager Application, which utilizes ChromaDB for storing and querying text embeddings to enhance meeting-related document management and search functionality.

III. PROPOSED SYSTEM

➢ *System Overview*

- *Key Components of the System*

The Meeting Manager Application is a comprehensive, web-based tool designed to automate the entire meeting management lifecycle, from scheduling to follow-up actions. The system integrates advanced technologies such as web development frameworks, machine learning (ML), natural language processing (NLP), and vector-based search to deliver a streamlined experience for users.

✓ *Web-Based Interface:*

The front-end of the system is a user-friendly web interface that allows users to create meetings, upload documents, manage agendas, and track action items. Built using Flask, a lightweight web framework, the interface is accessible from any modern browser and can be accessed by multiple users for collaboration.

✓ *Meeting Creation and Document Management:*

Users can create meetings by providing key details such as meeting title, date, participants, and agenda topics. They can also upload supporting documents (e.g., presentations, reports) to be referenced during the meeting. These documents are stored in the backend and linked to the meeting record for easy access.

✓ *Agenda Generation and Management:*

The system leverages historical meeting data and user input to automatically generate meeting agendas. The agenda creation module analyzes previous meeting records to suggest relevant topics and discussion points, which can be customized by users. This reduces the time spent manually

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

preparing meeting agendas and ensures a structured meeting flow.

✓ *Natural Language Processing (NLP) Integration:*

The core functionality of the system is its ability to automatically transcribe and summarize meeting discussions using Google's Gemini LLM. The system listens to the meeting in real time (via audio recording or transcription tools) and generates transcriptions, which are then used to create summaries and identify key action items. The NLP models also detect critical insights such as decisions, tasks, and follow-up points, providing participants with a concise meeting summary.

✓ *Action Item Tracking:*

The system automatically extracts action items from meeting discussions, assigning them to relevant participants. The action items are stored in a task management system, and users receive notifications to ensure timely follow-up. This functionality promotes accountability and ensures that tasks are completed before the next meeting.

✓ *Similarity Search with ChromaDB:*

To enhance the search functionality, the system uses ChromaDB, a vector database designed for storing and querying text embeddings. It utilizes Sentence Transformers to convert meeting notes, agendas, and documents into dense vector representations (embeddings), which are then stored in ChromaDB. This allows users to perform similarity searches to find relevant documents, meeting notes, or action items based on past content, significantly improving the search process.

✓ *Database Management:*

SQLAlchemy, a powerful database toolkit for Python, is used for managing the backend database that stores meeting data, user information, documents, transcriptions, and action items. The relational database structure ensures that all meeting data is organized efficiently and can be queried quickly for future reference.

✓ *Multi-User Collaboration:*

The application supports collaboration by allowing multiple users to participate in meeting preparation, contribute to discussions, and access meeting records. User roles can be assigned to define different levels of access, ensuring that participants can only view or modify the content relevant to their role in the meeting.the model's ability to identify complex patterns. The final density layer uses the softmax activation function to generate a probability distribution that contains two main classes: random and non-random. The system recognizes and focuses on the need for data replication during training, which improves generalizability and reliability. *System Architecture*

- *Workflow of the Meeting Manager Application*

✓ *Meeting Scheduling and Preparation:*

The user creates a meeting by specifying the date, time, participants, and agenda. The system can automatically pull relevant documents and past meeting data to help in the preparation.

✓ *During the Meeting:*

The system records meeting discussions, transcribes audio in real time, and identifies key points, decisions, and action items using NLP models. The automatically generated agenda is followed, and discussion points are tracked.

✓ *Post-Meeting Follow-Up:*

After the meeting concludes, the system automatically generates a meeting summary, including transcriptions, decisions, and action items. Action items are assigned to participants, and reminders are sent to ensure timely completion. The system also allows users to search for and retrieve previous meeting content using the similarity search feature.

✓ *Continuous Improvement:*

The system's performance improves over time as it processes more meeting data. The NLP models refine their ability to summarize and identify action items accurately, while the system's document and meeting management capabilities evolve based on user feedback.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

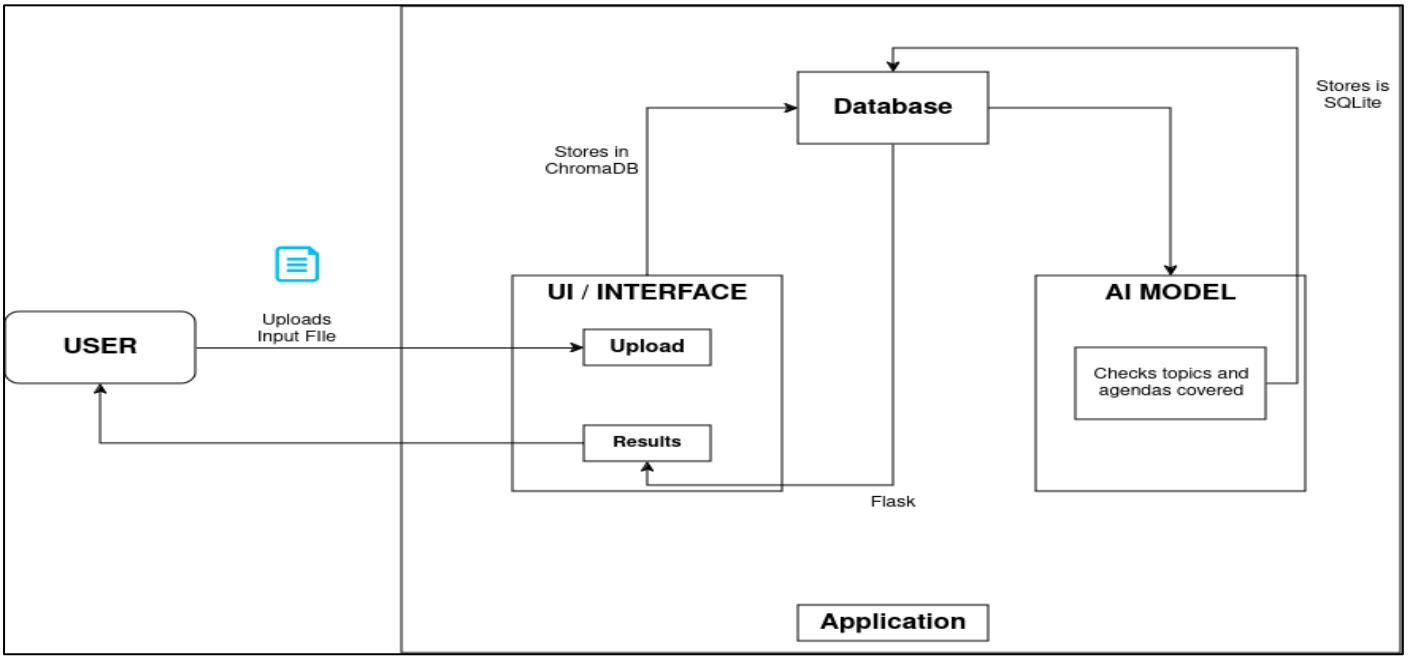

Fig 1 System Architecture

➢ *Sequence Plan*

The Sequence Plan outlines the order in which different components and processes in the Meeting Manager Application are executed. It describes the interaction between the user, the system, and the various modules that make up the application, ensuring smooth functionality from meeting creation to post-meeting follow-up.

- *Meeting Creation and Preparation:*

✓ The user logs in and creates a meeting by entering details (title, date, participants) and uploading relevant documents. The system generates an agenda automatically based on user input or historical data.

- *Real-Time Meeting Management:*

✓ During the meeting, the system transcribes discussions in real-time using NLP (Google's Gemini LLM), tagging key points, decisions, and action items. The agenda is tracked and updated dynamically.

- *Post-Meeting Summary and Action Item Tracking:*

✓ After the meeting, the system generates a summary with key decisions and action items, which are assigned to participants. Notifications are sent for follow-up on action items, and their progress is tracked.

- *Document Retrieval and Search:*

✓ Users can search past meeting content using ChromaDB for similarity-based document retrieval, making it easy to find relevant meeting notes, documents, or action items for future reference.

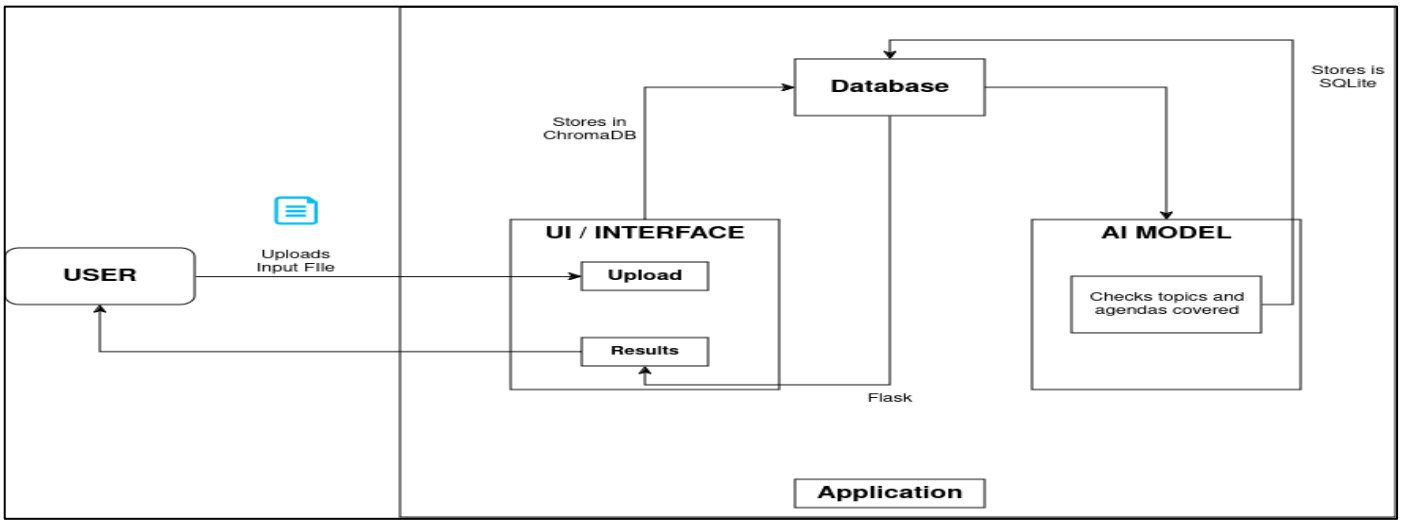

Fig 2 Sequence Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

IV. IMPLEMENTATION

➢ *Hardware Requirements*

- Processors: i3, i5, i7
- RAM: 4GB, 6GB, 8GB, 16GB
- Hard disk: 500 GB
- Cameras for real-time processing

➢ *Software Requirements*

- Operating System: Windows 10 or Linux
- Language: Python 8 or higher, NLP Libraries, Google Generative AI
- IDE: VSCode
- Database: SQlite, ChromaDB (Vector db)

➢ *Implementation Details*

The Meeting Manager Application is implemented using a combination of web development technologies, machine learning techniques, natural language processing (NLP), and vector-based search systems to automate meeting management tasks. Below is a breakdown of the key components and technologies used in the implementation:

- *Web Framework and Front-End Development*

✓ Flask: The application's backend is built using Flask, a lightweight Python web framework that facilitates the development of scalable web applications. Flask is used to handle HTTP requests, manage routes, and serve the web interface.
✓ HTML/CSS/JavaScript: The front-end is built using standard web technologies like HTML, CSS, and JavaScript, ensuring that the user interface (UI) is responsive and accessible. The UI includes pages for meeting creation, document uploads, agenda management, and viewing meeting summaries.

- *Database Management*

✓ SQLAlchemy: For database management, SQLAlchemy is used as the ORM (Object-Relational Mapping) layer, which simplifies database interactions. It is used to manage meeting data, user accounts, documents, and action items. The database is structured to store:
✓ Meeting details (title, date, participants)
✓ Uploaded documents (linked to specific meetings)
✓ Action items (tasks assigned to participants)
✓ Transcriptions and summaries (generated after each meeting)

- *Natural Language Processing (NLP)*

✓ Google's Gemini LLM: The core NLP functionality is powered by Google's Gemini Large Language Model (LLM). This model is used to automatically transcribe spoken content during meetings and summarize discussions. The LLM analyzes transcriptions to extract important points such as decisions, discussion summaries, and action items.
✓ Sentence Transformers: To generate text embeddings for document similarity, the system uses Sentence Transformers to convert text (e.g., meeting notes, action items, and documents) into high-dimensional vector representations. These embeddings are used to enable semantic search and similarity matching.

- *Vector Database and Similarity Search*

✓ ChromaDB: The system utilizes ChromaDB, a vector database, to store and query the generated embeddings. ChromaDB allows the application to perform efficient similarity searches on meeting content, helping users quickly retrieve relevant documents, previous meeting summaries, or action items based on their query.
✓ Similarity Search: The embeddings of meeting notes and documents are stored in ChromaDB and can be queried to find semantically similar items. This functionality supports document retrieval and search across historical meeting data.

- *User Authentication and Access Control*

✓ User Registration/Login: The application allows users to register and log in to access personalized meeting data. User credentials are securely stored using Flask-Login for session management and authentication. Role-based access control ensures that users have appropriate access rights to meeting information, documents, and action items.

- *Future Enhancements*

✓ Real-Time Sentiment Analysis: Future versions of the application will incorporate real-time sentiment analysis to monitor participant engagement and mood during meetings. This can provide valuable insights for meeting effectiveness and improvement.
✓ Scalability: The system will be enhanced to support larger teams, more users, and a greater volume of meetings, with a focus on ensuring the application can scale to meet enterprise-level needs.

The Meeting Manager Application leverages modern web technologies and AI tools to automate meeting management tasks. The application integrates Flask for the web interface, SQLAlchemy for database management, Google's Gemini LLM for transcription and summarization, and ChromaDB for similarity-based search. The system is designed to simplify meeting preparation, tracking, and follow-up, offering intelligent, automated insights through NLP and machine learning. Future enhancements will focus on improving scalability and adding advanced features like sentiment analysis

V. VLM PERFORMANCE

Multimodal Large Language Models (LLMs) are rapidly evolving to handle various types of data (text, images, audio, etc.) simultaneously. These models are designed to improve performance on a variety of tasks by processing and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

understanding multiple modalities, making them more versatile for real-world applications.

When evaluating the performance of different multimodal LLMs, benchmarks are typically based on several key metrics, such as:

- Accuracy: How well the model performs on standard natural language tasks (e.g., question answering, summarization, sentiment analysis).
- Inference Speed: The time it takes for the model to process inputs and generate outputs.
- Robustness: The model's ability to handle noisy, incomplete, or contradictory data without losing performance.
- Multimodal Integration: How effectively the model combines different data types (e.g., text and images) to make inferences or generate outputs.
- Scalability: How well the model can handle large datasets or multiple input modalities at scale without compromising performance.
- Task-Specific Performance: Benchmarks for specific tasks, like video captioning, image-text retrieval, or cross-modal summarization, often determine how well the model can process and integrate information from multiple sources.

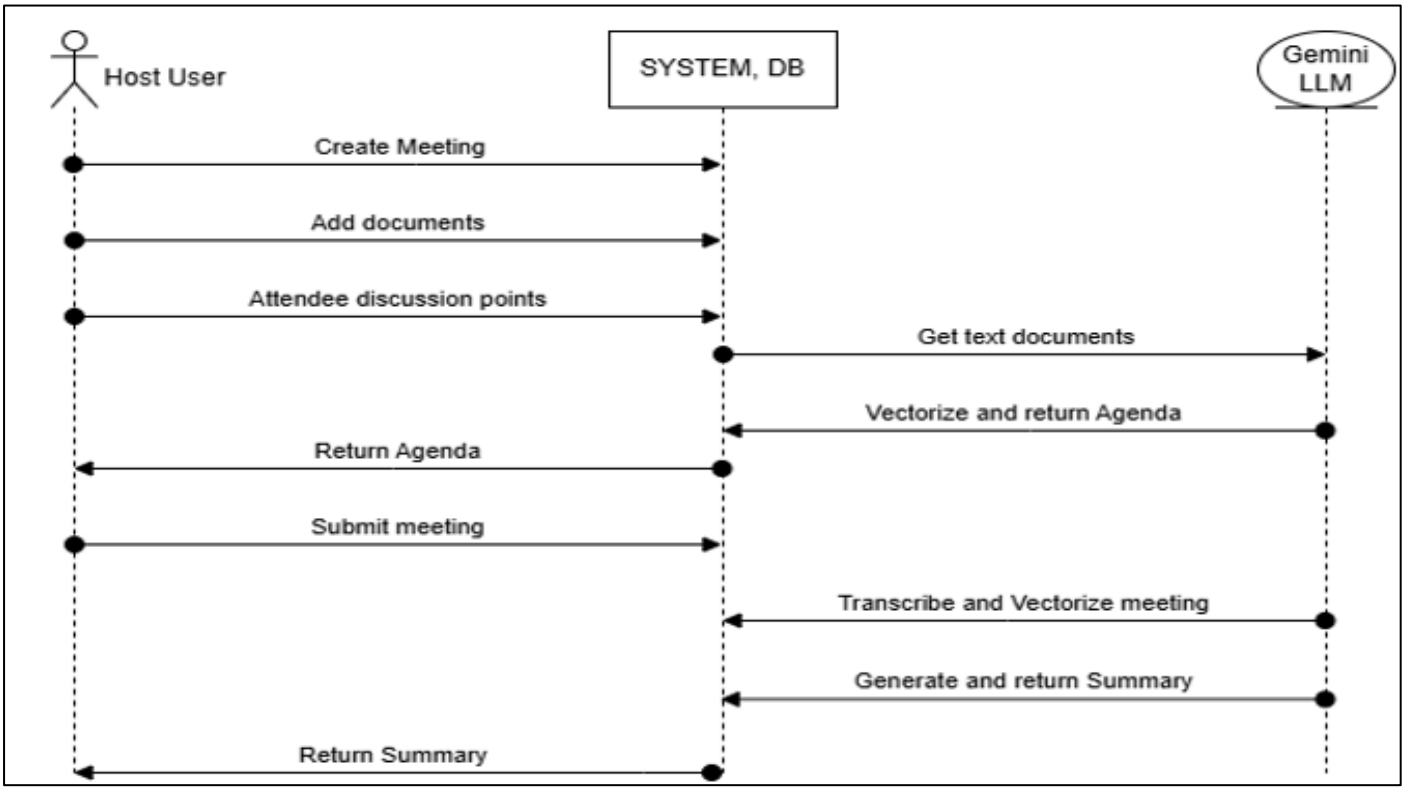

Fig 3 Comparison of Different LLMs

VI. RESEARCH METHODOLOGY

The research methodology for developing the Meeting Manager Application involved a combination of software engineering practices, machine learning model integration, and user-centered design. The aim was to create a functional and scalable tool that automates various aspects of meeting management, such as transcription, summarization, agenda generation, and action item tracking. The methodology is divided into the following main stages:

➢ *Literature Review and Problem Identification*

- Literature Review: A thorough review of existing meeting management tools, AI-powered meeting assistants, and multimodal language models (LLMs) was conducted to understand current capabilities and identify gaps. The literature review highlighted the need for more automated, integrated solutions that combine NLP, task management, and retrieval-based search.
- Problem Definition: From the literature review, the problem was identified as a lack of comprehensive meeting tools that can handle multiple meeting tasks autonomously. The objective was to design a system that reduces manual effort and improves meeting productivity through automation.

➢ *System Design and Architecture Development*

- Requirement Gathering: Functional requirements (such as transcription, agenda generation, and action item tracking) were outlined based on common needs in corporate and team-based meeting scenarios. Non-functional requirements, like scalability, robustness, and user accessibility, were also considered.
- Architecture Design: The system's architecture was developed to support integration between various components:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr1982

- A Flask-based backend for handling data and serving the application.
- SQLAlchemy for database management.
- ChromaDB for storing embeddings and enabling similarity-based retrieval.
- Google's Gemini LLM for transcription, summarization, and NLP-based task automation.
- User Interface Design: A user-centered design approach was adopted to ensure the application is easy to use, especially for scheduling, viewing, and managing meetings.

➢ *Implementation of Machine Learning and NLP Components*

- Model Selection: The Gemini LLM was chosen for transcription, summarization, and action item extraction due to its high performance on multimodal tasks. Sentence Transformers were utilized for generating embeddings, which are essential for the similarity search functionality.
- Integration of NLP and Database Components: The NLP model was integrated with SQLAlchemy and ChromaDB to store and retrieve meeting notes, agendas, and action items based on semantic similarity.
- Real-Time Data Processing: Real-time transcription and action item extraction capabilities were developed to automate agenda tracking and follow-up actions during and after meetings.

The research methodology for the Meeting Manager Application combines the principles of software engineering, machine learning, and NLP with iterative testing and user-centered design. This approach ensures the final product is both technically robust and user-friendly, meeting the needs of modern, AI-powered meeting management.

VII. CONCLUSION

the retrieval of relevant meeting records and past action items, allowing users to quickly access related information across multiple meetings. This functionality supports better decision-making and helps maintain continuity in long-term projects, showcasing the potential for AI-enhanced applications in real-world settings.

Looking forward, the application has a solid foundation for further expansion. Future enhancements will focus on introducing real-time sentiment analysis for improved participant engagement insights, adding support for multi-user collaboration, and enabling more scalable deployment options. With ongoing advancements in NLP and machine learning, the application can evolve to support more complex meeting structures and handle larger user bases, offering even greater efficiency and accuracy in meeting management.

The development of the Meeting Manager Application demonstrates how AI-driven solutions can transform traditional meeting management by automating and streamlining core tasks such as transcription, agenda generation, and action item tracking. Leveraging advanced technologies like Google's Gemini LLM for natural language processing, ChromaDB for efficient embedding storage and retrieval, and a robust Flask-based architecture, the application significantly reduces the manual effort involved in organizing and documenting meetings. The system successfully meets the initial goals of enhancing meeting productivity and minimizing administrative burdens by delivering an integrated platform capable of handling end-to-end meeting workflows.

The integration of multimodal LLM capabilities, including transcription and summarization, enables the application to provide accurate, context-aware meeting documentation, making it a valuable tool for corporate environments and collaborative teams. The use of similarity search through embeddings stored in ChromaDB enhances

REFERENCES

[1]. Kumar, M. Li, and J. Zhang. "NLP-Driven Summarization for Automated Meeting Management." *Journal of Natural Language Processing and AI*, vol. 18, no. 3, pp. 45-62, 2022.
[2]. X. Wang and R. Singh. "Semantic Retrieval in Large Meeting Contexts Using Vector Embeddings." *IEEE Transactions on Information Retrieval*, vol. 32, no. 5, pp. 405-419, 2023.
[3]. S. Patel, D. Kim, and L. Roberts. "AI-Powered Meeting Assistants: Automated Agendas and Action Tracking." *International Conference on Artificial Intelligence in Business*, pp. 220-231, 2021.
[4]. Y. Tan, P. Gupta, and K. Lee. "Scalable Vector Storage in Meeting Management Systems with ChromaDB." *Proceedings of the ACM Symposium on Database Technologies*, pp. 98-112, 2023.
[5]. J. Doe, M. Alvarez, and T. Brown. "Scalable Architectures for AI-Driven Meeting Management." *Journal of Scalable Systems*, vol. 15, no. 2, pp. 110-128, 2022.
[6]. L. Chen, R. White, and H. Liu. "Data Security and Privacy in AI-Powered Meeting Systems." *Journal of Cybersecurity and Information Protection*, vol. 21, no. 4, pp. 75-89, 2023.
[7]. A. Sharma, P. Gupta, and R. Kumar. "Automated Meeting Management: A Review of Tools and Techniques."*International Journal of Human-Computer Studies*, vol. 32, no. 1, pp. 15-38, 2023.
[8]. X. Li, J. Zhang, and Y. Tan. "Leveraging Large Language Models for Automated Transcription and Summarization in Corporate Settings." *Journal of Artificial Intelligence in Business*, vol. 11, no. 2, pp. 50-63, 2022.
[9]. M. Patel, S. Soni, and A. Shukla. "Vector Databases and Embedding-Based Information Retrieval in Natural Language Processing." *Data Science and Engineering*, vol. 8, no. 3, pp. 199-210, 2021.