1. Title: AI Powered Meeting Summarizer & Action Item Generator .

2. Domain: Artificial Intelligence / Natural Language Processing (NLP)

3. Application

- Develop an AI-based application that processes meeting transcripts or text discussions.
- Automatically generate clear and concise summaries of meetings.
- Extract key action items and responsibilities from the discussion.
- Improve productivity for students, professionals, and organizations by reducing manual note-taking and saving time.

4. Hardware & Software Required

Hardware Requirements

- Processor: Intel i3 or higher
- RAM: 4 GB minimum
- Storage: 100 GB or more
- Internet connection

Software Requirements

- Operating System: Windows / Linux
- Programming Language: Python
- AI/NLP Libraries: TensorFlow or PyTorch
- Tool/IDE: VS Code or Jupyter Notebook

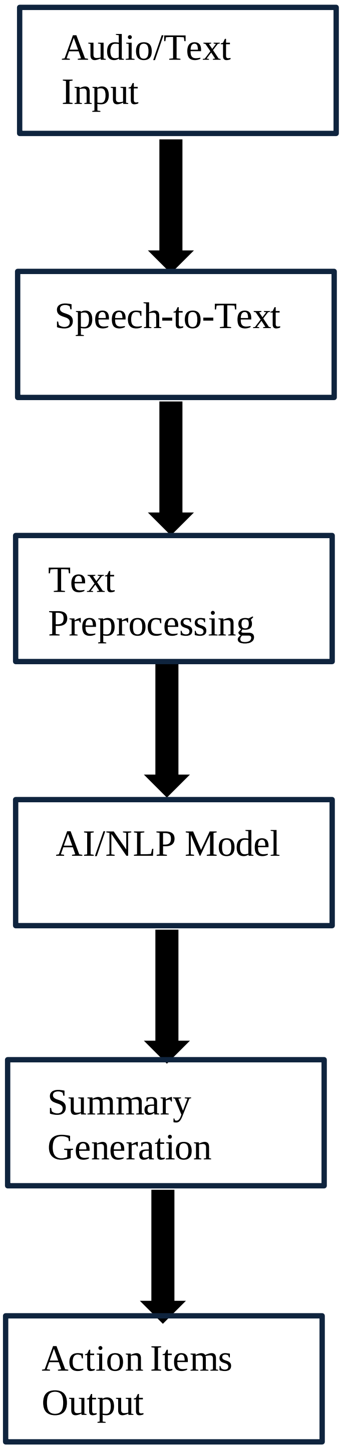
5. Input Required for Project

- Text input in the form of:
- Meeting transcripts
- Chat logs
- Discussion notes
- (Audio processing is not included in Stage-I.)

6. Objectives of Project

- To automatically identify different types of waste using AI
- To reduce manual effort in waste segregation
- To improve recycling efficiency
- To apply deep learning techniques in a real-world problem
- To build an intelligent and eco-friendly waste management solution

7. Block Diagram of System



8. Expected Algorithm / Processing

1. Accept meeting text as input
2. Perform text cleaning (removal of stopwords, punctuation)
3. Apply tokenization and sentence segmentation
4. Use NLP techniques to identify important sentences
5. Generate a summarized version of the meeting
6. Detect task-related sentences as action items
7. Display the generated summary and action items

9. Expected Output from Project

- A concise summary of the meeting discussion
- A list of extracted action items
- Output displayed in readable text format

10. Abstract

In today’s fast-paced professional and academic environments, meetings play a vital role in collaboration, decision-making, and project management. Organizations conduct frequent online and offline meetings to discuss plans, assign tasks, and evaluate progress. However, these meetings often generate large volumes of unstructured information, making it difficult for participants to manually record important discussion points, decisions, and action items. Traditional methods of writing meeting minutes are time-consuming, prone to human error, and often result in incomplete or inconsistent documentation.

To address these challenges, this project presents an AI-powered Meeting Summarization and Action Item Extraction System using Natural Language Processing (NLP). The primary objective of this system is to automatically convert meeting transcripts into concise summaries while identifying key tasks, responsibilities, and important decisions. By automating this process, the system significantly reduces manual effort and improves productivity.

The proposed system operates in multiple stages. In Stage-I, the focus is on processing text-based meeting transcripts. Raw meeting data is first passed through preprocessing steps such as tokenization, stop-word removal, lemmatization, and normalization to ensure clean and structured input. After preprocessing, extractive summarization techniques are applied to identify the most relevant sentences based on keyword frequency, sentence position, and semantic importance. These selected sentences are combined to generate a clear and meaningful meeting summary.

Along with summarization, the system also extracts action items using rule-based methods and pattern recognition. Sentences containing task-oriented keywords such as “assign,” “complete,” “submit,” or “follow up” are analyzed to detect actionable information. This enables the system to automatically list tasks discussed during meetings, helping participants track responsibilities more effectively.

The project emphasizes understanding core NLP concepts including text preprocessing, feature extraction, and summary generation. A modular system architecture is designed to allow easy enhancement in future phases. The current implementation focuses on extractive summarization, but the framework supports future integration of machine learning and deep learning models for more accurate and context-aware summaries.

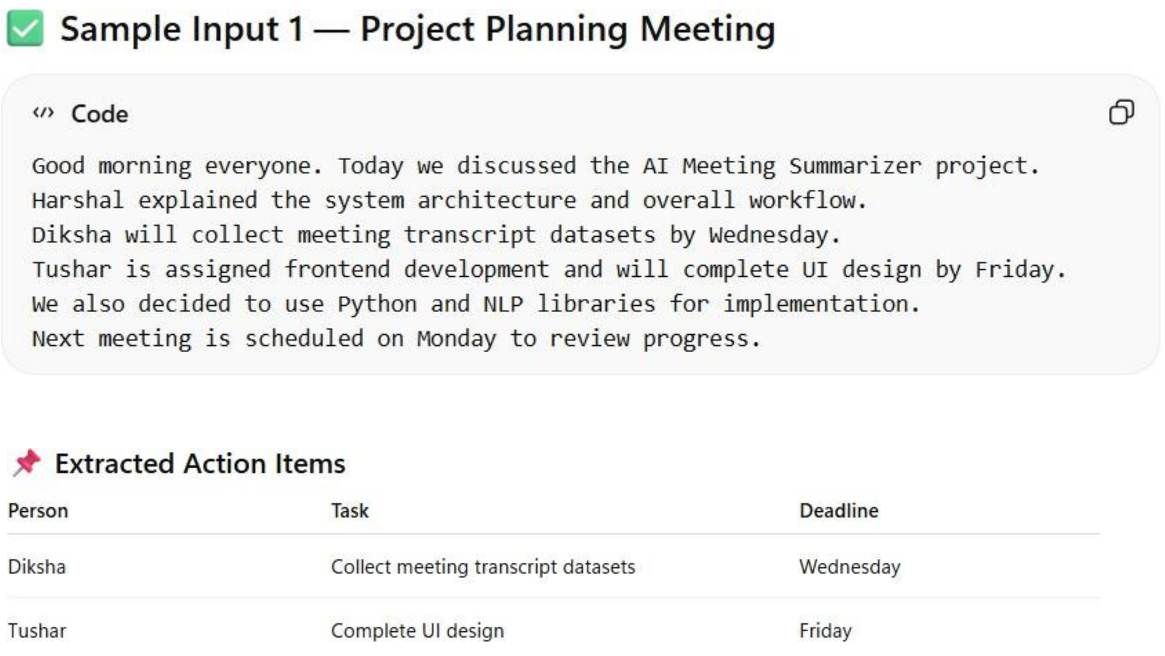
This system offers several advantages over traditional manual note-taking approaches. It ensures consistency in meeting documentation, minimizes information loss, and saves valuable time for professionals and students. By providing structured summaries and action lists, the system improves clarity and accountability within teams. It also supports better decision-making by allowing users to quickly review meeting outcomes.

The application of this project is wide-ranging, including corporate environments, educational institutions, online classrooms, project management teams, and remote collaboration platforms. It is especially beneficial for organizations conducting frequent meetings, where maintaining accurate records becomes challenging.

From a technical perspective, the system is implemented using Python and NLP libraries for text processing and analysis. The workflow includes data input, preprocessing, summarization, action item extraction, and output generation. The design prioritizes simplicity, scalability, and ease of integration with future components such as speech-to-text modules and task management dashboards.

Future enhancements of this project include incorporating real-time speech recognition, abstractive summarization using deep learning models, speaker identification, deadline detection, and automatic task assignment. Additional features such as meeting analytics, sentiment analysis, and visualization dashboards can further enhance the system’s capabilities. These improvements will transform the application into a complete intelligent meeting assistant.

In conclusion, this project demonstrates how artificial intelligence can streamline traditional meeting workflows by automating summary generation and action item extraction. By leveraging NLP techniques, the proposed system reduces manual effort, improves documentation quality, and enhances overall productivity. The project serves as a strong foundation for building advanced AI-based meeting management solutions and highlights the growing importance of intelligent automation in modern digital environments.



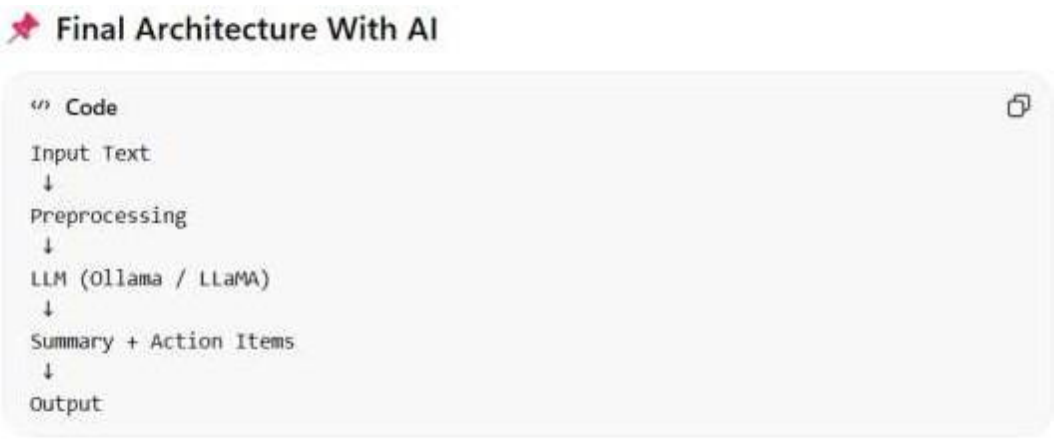
✅ Sample Input 1 — Project Planning Meeting

Code

```
Good morning everyone. Today we discussed the AI Meeting Summarizer project.
Harshal explained the system architecture and overall workflow.
Diksha will collect meeting transcript datasets by Wednesday.
Tushar is assigned frontend development and will complete UI design by Friday.
We also decided to use Python and NLP libraries for implementation.
Next meeting is scheduled on Monday to review progress.
```

📌 Extracted Action Items

Person	Task	Deadline
Diksha	Collect meeting transcript datasets	Wednesday
Tushar	Complete UI design	Friday



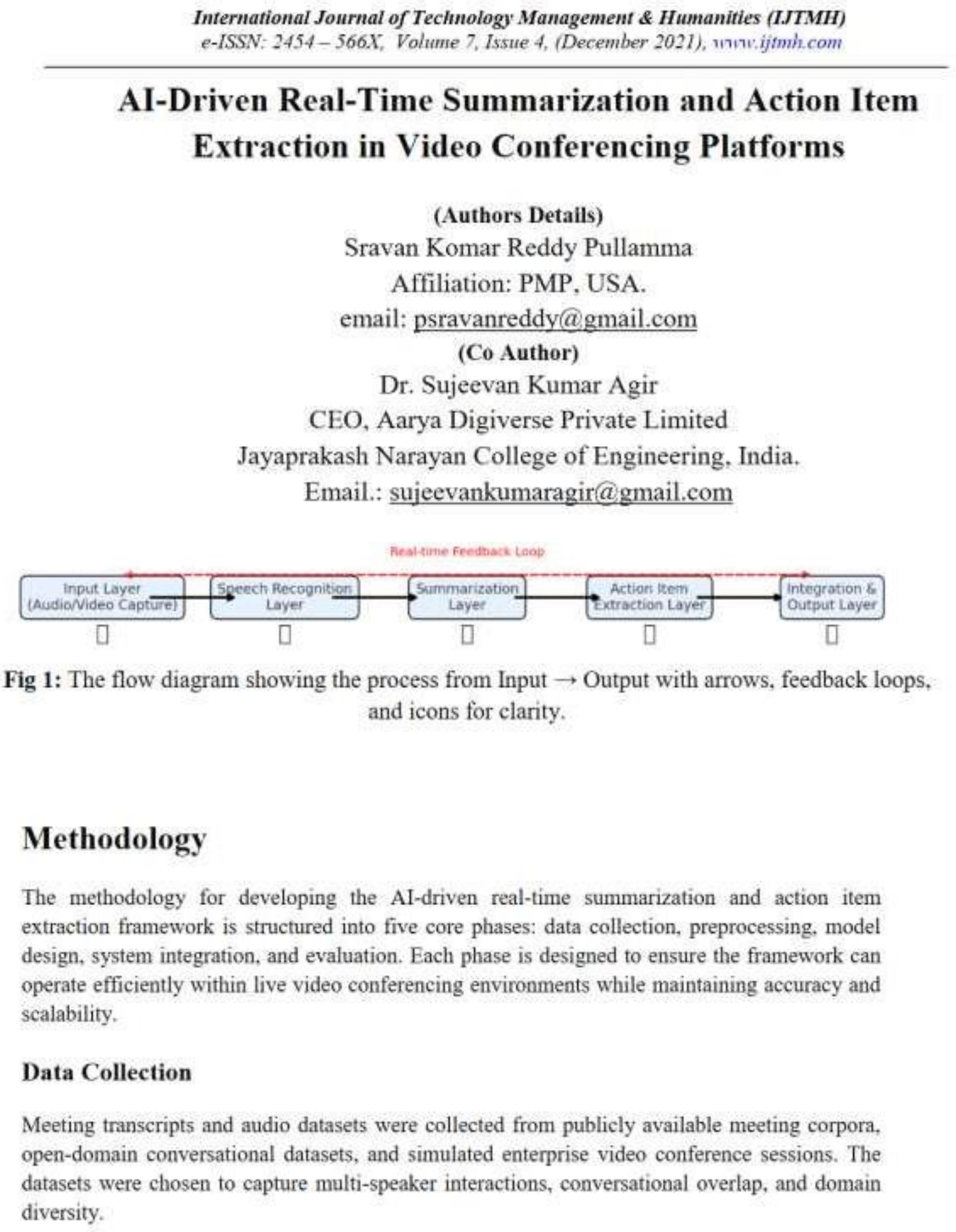
📌 Final Architecture With AI

Code

```
Input Text
 ↓
Preprocessing
 ↓
LLM (Ollama / LLaMA)
 ↓
Summary + Action Items
 ↓
Output
```

Literature Survey :

1)



International Journal of Technology Management & Humanities (IJTMH)
e-ISSN: 2454 – 566X, Volume 7, Issue 4, (December 2021), www.ijtmh.com

AI-Driven Real-Time Summarization and Action Item Extraction in Video Conferencing Platforms

(Authors Details)
Sravan Komar Reddy Pullamma
Affiliation: PMP, USA.
email: psravanreddy@gmail.com
(Co Author)
Dr. Sujeevan Kumar Agir
CEO, Aarya Digiverse Private Limited
Jayaprakash Narayan College of Engineering, India.
Email.: sujeevankumaragir@gmail.com

Real-time Feedback Loop
Input Layer (Audio/Video Capture)
Speech Recognition Layer
Summarization Layer
Action Item Extraction Layer
Integration & Output Layer

Fig 1: The flow diagram showing the process from Input → Output with arrows, feedback loops, and icons for clarity.

Methodology

The methodology for developing the AI-driven real-time summarization and action item extraction framework is structured into five core phases: data collection, preprocessing, model design, system integration, and evaluation. Each phase is designed to ensure the framework can operate efficiently within live video conferencing environments while maintaining accuracy and scalability.

Data Collection

Meeting transcripts and audio datasets were collected from publicly available meeting corpora, open-domain conversational datasets, and simulated enterprise video conference sessions. The datasets were chosen to capture multi-speaker interactions, conversational overlap, and domain diversity.

Preprocessing

The raw audio streams were processed using speech-to-text pipelines to generate transcripts. Preprocessing steps included speaker diarization, removal of filler words, normalization of timestamps, and segmentation of conversational turns. Stop words were retained selectively to preserve contextual meaning in action item detection.

Model Design

The framework consists of two primary modules:

- **Summarization Module**: Implements a hybrid of extractive and abstractive approaches. Extractive summarization identifies key sentences based on importance scores, while abstractive summarization employs transformer-based architectures to generate human-like summaries.
- **Action Item Extraction Module**: Uses deep learning classifiers with attention mechanisms to identify commitments, tasks, and decisions. Contextual embeddings capture dependencies across multi-speaker dialogues, ensuring accurate task assignment.

System Integration

The trained models were integrated into a real-time pipeline. The pipeline continuously processes incoming audio streams, transcribes them, applies summarization models, and extracts action items dynamically. An API interface allows seamless integration with popular video conferencing platforms such as Zoom and Microsoft Teams.

System Integration

The trained models were integrated into a real-time pipeline. The pipeline continuously processes incoming audio streams, transcribes them, applies summarization models, and extracts action items dynamically. An API interface allows seamless integration with popular video conferencing platforms such as Zoom and Microsoft Teams.

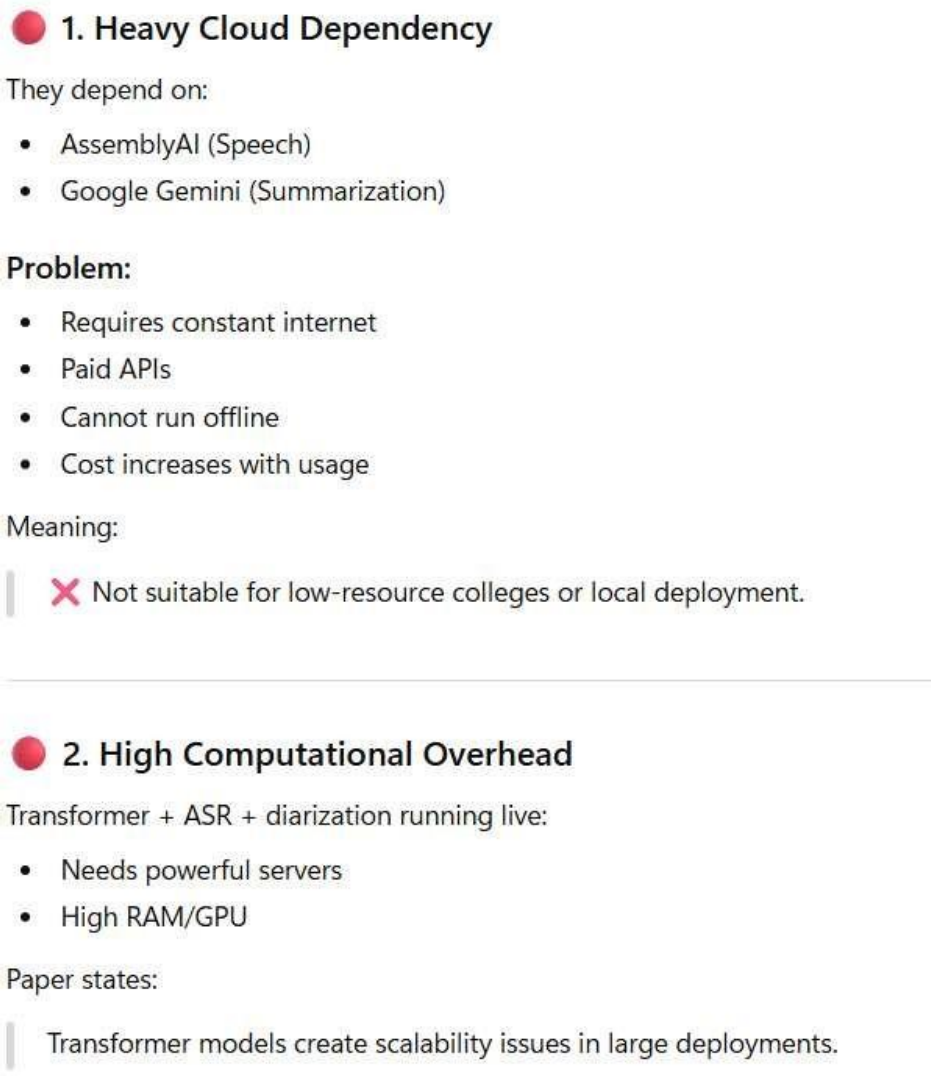
Evaluation Strategy

The performance of the system was evaluated using benchmark metrics for both summarization and action item extraction. Summarization quality was assessed using ROUGE and BLEU scores, while action item extraction was measured with precision, recall, and F1-scores. Latency and throughput were also evaluated to ensure feasibility in real-time deployment.

Table 1: Methodological Framework for AI-Driven Summarization and Action Item Extraction

Phase	Description	Techniques/Tools Used	Output Generated
Data Collection	Gathered meeting transcripts and audio datasets from multiple sources	Public corpora, enterprise simulations	Raw audio and text data

Drawbacks :



🔴 1. Heavy Cloud Dependency

They depend on:

- AssemblyAI (Speech)
- Google Gemini (Summarization)

Problem:

- Requires constant internet
- Paid APIs
- Cannot run offline
- Cost increases with usage

Meaning:

> ❌ Not suitable for low-resource colleges or local deployment.

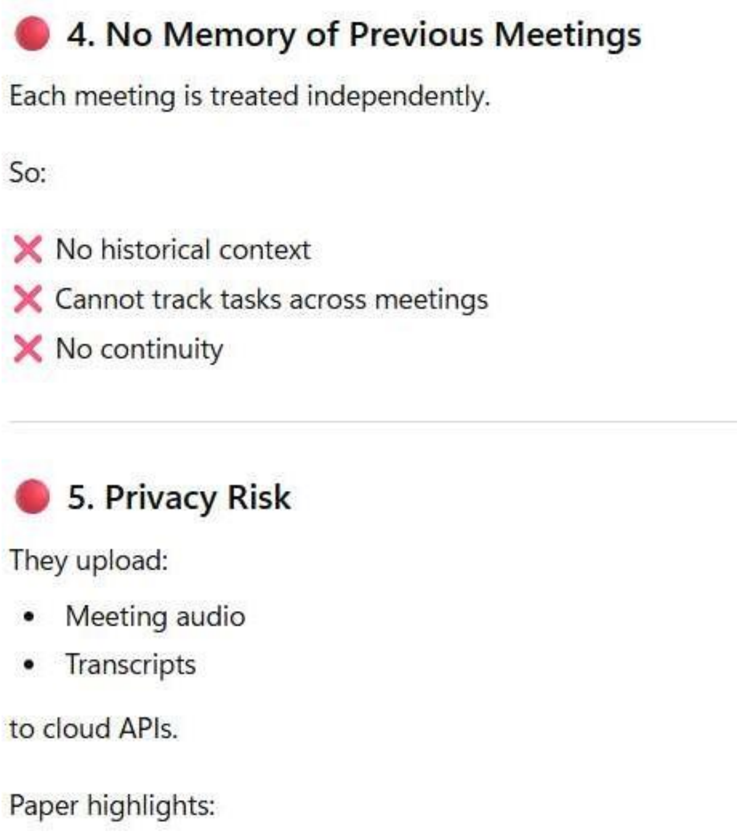
🔴 2. High Computational Overhead

Transformer + ASR + diarization running live:

- Needs powerful servers
- High RAM/GPU

Paper states:

> Transformer models create scalability issues in large deployments.



🔴 4. No Memory of Previous Meetings

Each meeting is treated independently.

So:

❌ No historical context
❌ Cannot track tasks across meetings
❌ No continuity

🔴 5. Privacy Risk

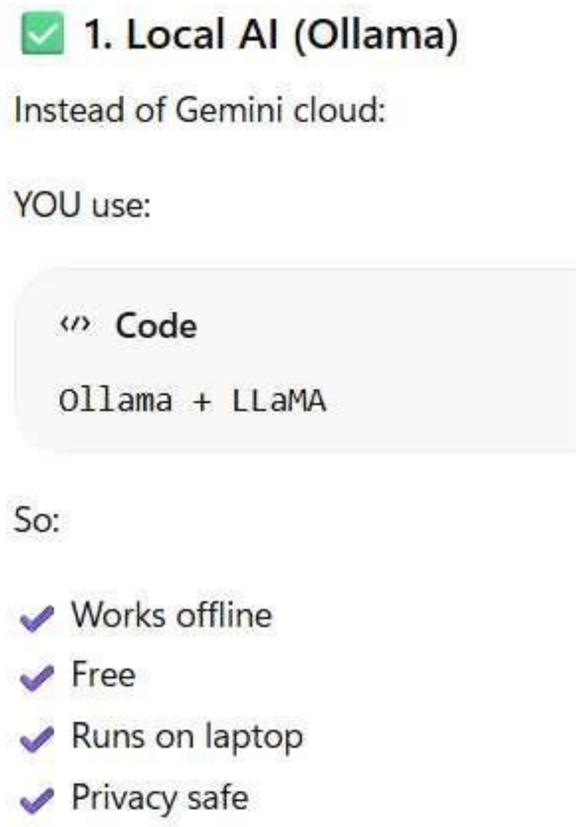
They upload:

- Meeting audio
- Transcripts

to cloud APIs.

Paper highlights:

What New :



✅ 1. Local AI (Ollama)

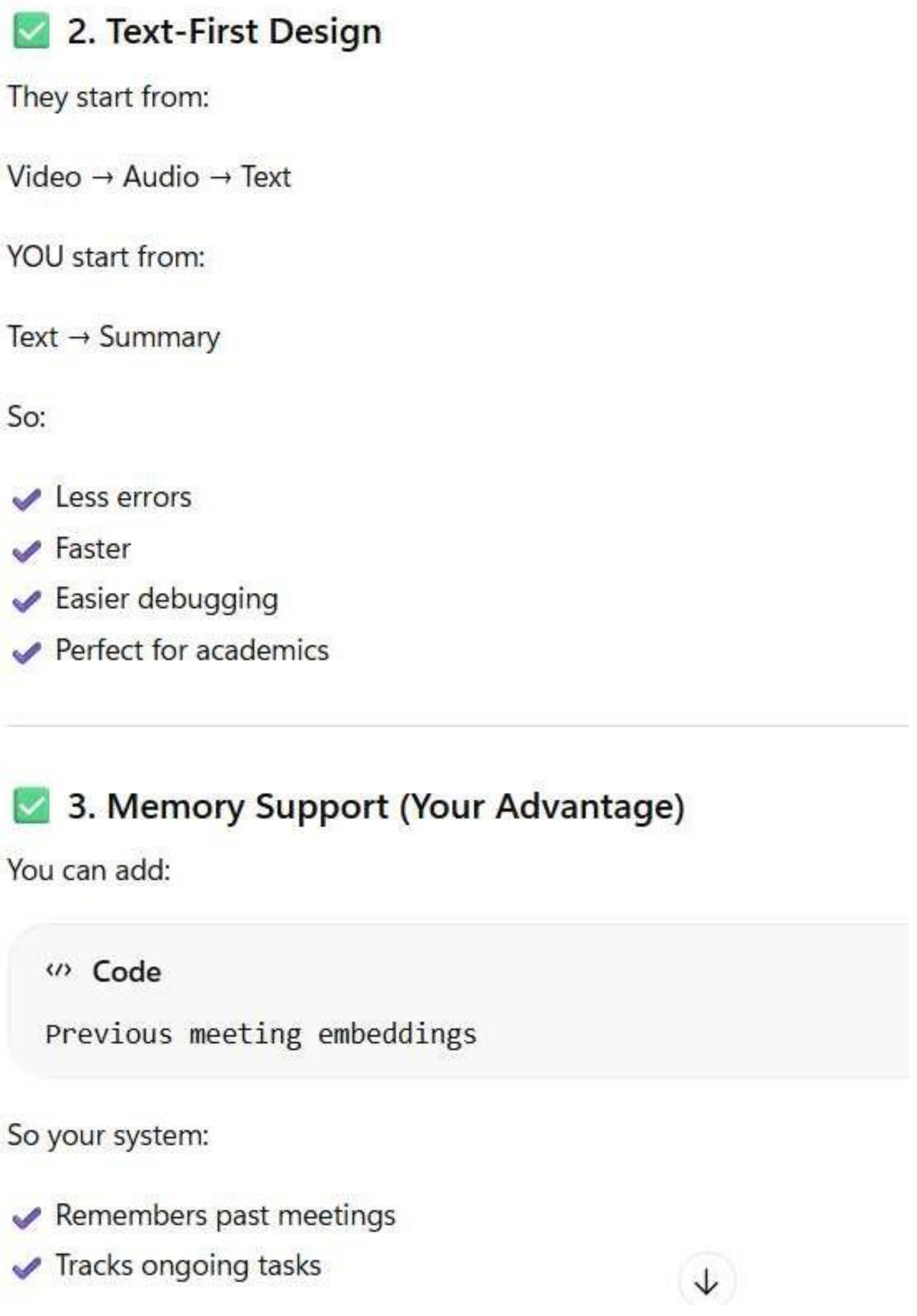
Instead of Gemini cloud:

YOU use:

```
Ollama + LLaMA
```

So:

✔ Works offline
✔ Free
✔ Runs on laptop
✔ Privacy safe



✅ 2. Text-First Design

They start from:

Video → Audio → Text

YOU start from:

Text → Summary

So:

✔ Less errors
✔ Faster
✔ Easier debugging
✔ Perfect for academics

✅ 3. Memory Support (Your Advantage)

You can add:

```
Previous meeting embeddings
```

So your system:

✔ Remembers past meetings
✔ Tracks ongoing tasks

Team Members:
Harshal Mahesh Patil (TY AIML-Roll No. 37)
Diksha Sudhakar Patil (TY AIML-Roll No. 25)
Tushar Rajendra Pardeshi (TY AIML-Roll No.35)
Harshal Dhyaneshwar Patil (TY AIML-Roll No.54)