Meeting Insight Generator

AI-Powered Meeting Insight Generator

Purpose

In modern organizations, meetings happen frequently — stand-ups, reviews, planning sessions, interviews, and brainstorming discussions.
While meetings contain valuable information, much of it is often lost due to:

- Long and unstructured conversations
- Lack of proper documentation
- Missed action items and unclear decisions
- Time-consuming manual note-taking

An AI-powered system that can **listen to meetings, understand conversations, and generate structured insights** can significantly improve productivity and decision-making.

Problem Statement

Design and develop an **AI-based Meeting Insight Generator** that can analyze a meeting's **audio recording or transcript** and automatically generate:

- A concise meeting summary
- Clearly defined action items
- Important decisions taken during the meeting
- Key discussion points

The system should help users quickly understand **what happened**, **what was decided**, and **what needs to be done next**, without listening to the entire meeting again.

Objective

The objective of this challenge is to evaluate the participant's ability to:

- Build an end-to-end **AI-powered application**
- Convert unstructured meeting conversations into **structured insights**
- Apply NLP, speech-to-text, and LLM reasoning effectively
- Deliver a **user-friendly and deployable solution**

Functional Requirements

Your application must be able to:

1. Accept **meeting input** in one of the following formats:
    - Meet Recordings URL
    - Audio file (recorded meeting)
    - Text transcript (manual or pre-generated)
2. Convert audio input into text (if audio is provided)
3. Analyze the meeting content and generate:
    - **Meeting Summary** – overall context and outcome
    - **Key Discussion Points** – main topics covered
    - **Decisions Made** – confirmed conclusions or approvals
    - **Action Items** – tasks, responsibilities, or next steps
4. Display the results clearly in the application interface

Input & Output Details

Input

- Meet Recordings URL **or**
- Audio file (MP3/WAV) **or**
- Text transcript of a meeting

Optional inputs:

- Meeting title
- Meeting type (stand-up / planning / review / discussion)

Output

The application should generate structured output such as:

- **Summary**
- **Action Items**
- **Decisions**
- **Key Topics Discussed**
- **And if any additional segment required**

Expected Application Flow

1. User uploads an audio file or pastes a transcript
2. Audio is converted to text (if applicable)
3. AI processes the conversation
4. Insights are extracted and structured
5. Results are displayed in the UI
6. (Optional) User downloads the summary or report

Technical Constraints & Guidelines

- **Backend must be implemented in Python**
- Participants are free to choose any frontend technology
- Open-source or free-tier tools are strongly encouraged
- The solution should focus on **clarity, usefulness, and insight quality**
- Notebook-only solutions are **not accepted**

Suggested Tech Stack (Free-First)

- **Speech-to-Text:** OpenAI Whisper / AssemblyAI (free tier) / local models
- **NLP / LLM:** OpenAI GPT / HuggingFace models / open-source LLMs
- **Backend:** Python (FastAPI / Flask)
- **Frontend:** Streamlit / React / HTML + CSS
- **Deployment:** Streamlit Cloud / Render / Hugging Face Spaces

---

Deployment Expectations

- The application must be **deployed**
- It should be accessible via a **public URL**
- Judges should be able to:
    - Open the link
    - Upload input
    - View generated insights without local setup

Evaluation Criteria

Participants will be evaluated on:

Criteria	Description
Problem Understanding	How well the solution addresses meeting insight challenges

Insight Quality	Relevance and clarity of summaries, decisions, and action items
AI Usage	Effective use of speech-to-text and NLP/LLMs
Application Design	User-friendly interface and flow
Deployment & Accessibility	Publicly accessible, working application

---

Deliverables

- Deployed application link
- Source code repository
- Demo Video Link
- Short documentation explaining:
  - Application flow
  - Tools and models used
  - Example input and output