

AI Meeting Minutes Summariser – Complete Interview Preparation Guide

1. Project Introduction – Long Answer

This project is an end-to-end AI-powered Meeting Minutes Summariser designed to automatically convert raw meeting audio or transcripts into structured, actionable meeting documentation. It integrates speech recognition, speaker diarization, natural language processing, summarization, and report generation into a unified pipeline. The objective is to eliminate manual note-taking, reduce human error, and ensure important decisions, action items, and deadlines are accurately captured.

2. Problem Statement & Motivation

In modern organizations, meetings are frequent and lengthy. Manual documentation is inefficient, inconsistent, and often incomplete. This project automates the extraction of meaningful information from meetings, improving productivity, accountability, and record-keeping.

3. High-Level Architecture Explanation

The system follows a pipeline-based architecture where each stage performs a specific transformation. The flow is: Input File → Transcription → Speaker Diarization → Named Entity Recognition → Summarization → Report Generation → UI. A central MeetingContext object carries all information across stages.

4. Why Pipeline Architecture Was Chosen

A pipeline architecture ensures modularity, scalability, and fault tolerance. Each component is independent and can be replaced or upgraded without impacting the rest of the system.

5. MeetingContext – Design Justification

MeetingContext acts as a single source of truth containing transcript text, speaker segments, entities, summaries, and metadata. This eliminates redundant variables and simplifies data flow across the pipeline.

6. Speech-to-Text (ASR) – Deep Explanation

Audio transcription is performed using OpenAI Whisper and Faster-Whisper models. Whisper was chosen for its robustness to accents, background noise, and multilingual capabilities. Chunk-based processing prevents memory issues for long meetings.

7. Speaker Diarization – In-Depth

Speaker diarization identifies who spoke and when. This is achieved using WhisperX for alignment and PyAnnote for speaker segmentation. Diarization is optional due to its computational cost.

8. Named Entity Recognition (NER)

NER extracts structured information such as people names, dates, and action items from transcripts. spaCy models combined with rule-based logic are used to ensure accuracy and flexibility.

9. Summarization Strategy – Long Answer

Summarization condenses long transcripts into concise summaries using transformer-based models. Chunk-aware summarization avoids token limits while preserving context.

10. Report Generation & Output

The system generates professional reports using ReportLab and Python-Docx. Reports include summaries, speaker contributions, action items, and key dates.

11. UI Design (Streamlit)

The Streamlit UI is intentionally lightweight and focuses on orchestration rather than computation. This ensures clean separation of concerns and maintainability.

12. Error Handling & Robustness

Each pipeline stage includes defensive error handling. Failures are logged and displayed gracefully without crashing the application.

13. Performance Optimization Decisions

Optimizations include chunking, optional diarization, and model size selection. These balance accuracy, speed, and resource usage.

14. Comparison with Traditional Solutions

Unlike simple summarization tools, this project integrates multiple AI components into a cohesive system. It reflects real-world production pipelines.

15. Security & Privacy Considerations

The system processes data locally by default, ensuring sensitive meeting information remains private.

16. Scalability & Extensibility

The modular design allows easy extension such as multilingual support or analytics dashboards.

17. Deployment Strategy

The application can be deployed on Streamlit Cloud, HuggingFace Spaces, or internal servers.

18. Common Interview Questions & Answers

Q: Why Whisper? A: It is robust to noise and accents.

Q: What is diarization? A: Identifying who spoke when.

Q: Why MeetingContext? A: Centralized data flow.

19. System Design Interview Explanation

Explain the pipeline flow, justify modularity, and discuss scalability and trade-offs.

20. Optimization Interview Discussion

Discuss chunking, optional diarization, CPU vs GPU trade-offs, and memory handling.

21. Resume Talking Points

- Built an end-to-end AI meeting summarization system.
- Integrated ASR, diarization, NLP, and reporting.
- Designed scalable pipeline architecture.

22. Why This Project Stands Out

This project demonstrates real-world AI system design, not just model usage.

23. Final Interview Summary

This project showcases strong fundamentals in AI, NLP, system design, and software engineering.