

PART 3: SYSTEM FAILURE REASONING

1. What specific transcription errors are most likely to occur and why?

The transcription errors most likely to occur in this recording are:

- Misrecognition of words due to emotional tone, fast speech, or unclear pronunciation (e.g., “Data Science”, “Business Analytics”).
- Incorrect or missing word boundaries, especially during overlapping phrases or filler words like “like”, “so”.
- Language/code-switching confusion, e.g., Tamil phrases mixed with English
- Dropped context or meaning when pauses or tones are not captured properly.

2. How would these errors affect downstream summarization?

- Key points about career options (Data Science course vs MBA) could be misrepresented.
- Important terms or phrases may be missed, reducing relevance in TF-IDF or other summarization methods.
- Misrecognized words may cause incorrect conclusions about decisions or advice.
- Emotional context (stress → calm) might be lost, affecting sentiment-aware summaries.

3. Which component in the pipeline amplifies errors the most?

The speech-to-text (STT) component amplifies errors the most because:

- It is the first semantic conversion step, and every downstream module depends on its output.
- Any mis-transcribed words propagate to language detection, normalization, and summarization, distorting meaning.
- Once a critical word (e.g., “MBA” vs “Data Science”) is misrecognized, the summary may become misleading.

4. Why would a single LLM call fail for this use case?

- Token limits can be exceeded if the full audio is transcribed and sent at once.
- Noisy or partially incorrect transcription makes the LLM prone to generating inaccurate summaries.
- Errors cannot be isolated or corrected incrementally, so a single LLM call risks amplifying mistakes.
- It creates a single point of failure for the entire summarization.

5. What tradeoffs exist between accuracy, cost, and latency?

- Higher accuracy requires robust STT models and large LLMs → increases cost and processing time.
- Lower-cost systems may reduce model complexity → risk losing nuance, context, and emotional cues.
- Real-time or low-latency processing prioritizes speed → may sacrifice deep understanding or accuracy.
- There is a balance for small recordings like yours (60 seconds), moderate STT + chunked summarization is best.

PART 7: FAILURE DISCLOSURE AND REFLECTION

1. If my system fails in production, it will most likely fail first because *speech-to-text transcription because emotional tone, overlapping speech, or code-switching*

2. What assumptions did you knowingly make?

- Users speak clearly and at a moderate pace.
- Audio quality is good enough for transcription.
- Conversation length is short (~60 seconds) or can be handled via chunking.
- Summaries can be generated accurately from mostly correct transcription, without relying on full generative models.

3. What monitoring signals would you add to detect silent failure?

- Transcription confidence scores drop below a threshold.
- Recognized word count suddenly drops or is unusually low.
- Summaries are empty or extremely short.
- Language detection mismatches (e.g., Tamil phrases misidentified).
- Chunk processing failures or skipped segments.

PART 8: COST AND ETHICAL RESPONSIBILITY

1. Estimate monthly cost for processing 1,000 similar conversations.

Estimated monthly cost is very low because:

- No paid APIs are used
- Processing runs on local CPU
- No GPU dependency

2. Which component dominates cost and why?

The speech-to-text component dominates cost because:

- It is computationally intensive
- It processes raw audio and analyzes the signal continuously.
- Downstream components (summarization, language detection) are lightweight in comparison.

3. If this system is used in a mental health or counseling context:

Mandatory safeguards:

- Clear disclaimer that summaries are not medical advice.
- Human review required before taking any action.
- Secure storage of audio and text.
- User consent and privacy protection.

Accountability:

- The system owner or deploying organization is responsible for outcomes.
- Users or the automated system itself are not accountable.