Notary

https://github.com/carz0987/Notary

Jaymond Lei (With the Assistance of ChatGPT)

Professor Sendag
EGR404
4 May, 2025

1. Introduction

In a world where meetings, lectures, interviews, and discussions frequently occur through audio and video, transcribing spoken content and generating clear summaries is a common but time-consuming task. Manual transcription is not only tedious, but it also lacks accessibility and scalability for users who need fast, intelligent extraction of key information.

This project, **Notary**, addresses this challenge by providing a lightweight, browser-accessible AI assistant capable of transcribing speech and summarizing content from both live microphone recordings and uploaded audio or video files. It also includes an embedded AI companion that allows users to query their transcribed notes interactively without hallucination or irrelevant responses.

2. Problem Statement

The problem at hand was how to enable users to:

- Convert spoken content into accurate, structured text.
- Automatically extract meaningful summaries from long transcripts.
- Ask intelligent questions about the notes and receive responses strictly based on the transcript.
- Perform all of this through a user-friendly, local-first web application using modern AI tools.

This project specifically aimed to solve issues with:

- Inconsistent access to transcription tools across formats.
- High latency or cost in cloud-based third-party summarization solutions.
- Lack of focus in AI chat tools that often “hallucinate” or answer beyond the scope of available data.

---

3. Methodology

The project was developed in iterative stages, following a structured, modular design.

3.1 Audio Capture and Upload

Users can either:

- Record new audio using their microphone via the browser.
- Upload existing `.mp3`, `.wav`, `.mp4`, or other audio/video formats.

To ensure compatibility and file size limits, all audio is automatically converted to `.mp3` using **pydub** prior to transcription.

3.2 Transcription

Audio input is passed to **OpenAI's Whisper API (whisper-1)**. This model provides state-of-the-art multilingual transcription capabilities. Only files under 25MB are accepted, as per OpenAI API limits.

3.3 Summarization

Once transcribed, the raw text is passed to **OpenAI GPT (gpt-4o)** via the new `responses.create()` API. The system is instructed to:

- Generate bullet-point summaries.
- Focus strictly on clarity and factual retention.

- Avoid speculation or fabrication.

3.4 Question Answering

The entire transcript and summary are saved to a local `.txt` file. An embedded Q&A interface allows users to ask questions about their notes. The system prompts GPT with both the saved notes and an instruction to only respond using that content, rejecting queries it cannot answer based on the text.

3.5 Web Interface

A user-facing UI was built using **Gradio**, featuring three tabs:

- **Record Audio**
- **Upload File**
- **Ask Your Notes**

Users can transcribe and summarize content, download it, and ask questions — all within a single session.

---

4. Tools and Technologies Used

Tool/Library	Purpose
`OpenAI Whisper`	Transcription of speech from audio/video
`OpenAI GPT-4o`	Summarization and structured Q&A
`Gradio`	Web UI and user input/output handling
`pydub`	Audio format conversion to `.mp3`
`Python`	Core programming language
`dotenv/os`	Environment variable handling for API keys

---

5. Results

The final product, **Notary**, was tested on various use cases including:

- Lecture recordings (~5–10 mins, .mp4)
- Voice notes (~2–3 mins, .wav)

✅ Achieved:

- Consistent, high-quality transcriptions across formats.
- Summaries that captured the essential information.
- Fast Q&A responses constrained by source material.
- Compatibility across browsers and operating systems.

📁 Output Example:

A single `.txt` file with:

```
📝 Transcript:
[Full transcription text]

📌 Summary:
• Key point one
• Key point two
• ...
```

---

6. Challenges and Design Decisions

6.1 File Size Limits

The Whisper API imposes a 25MB limit. To address this:

- All inputs were auto-converted to `.mp3`
- Transcription was aborted gracefully for oversized files

6.2 Accuracy of Summarization

`gpt-4o` was selected for its increased factuality and lower hallucination rate, as well as general flexibility.

6.3 Preventing Hallucinations in Q&A

Prompt engineering was key. Instructions were included in each Q&A call to restrict answers strictly to the notes provided. If the question was out of scope, the AI replied with:

> "That information is not available in the notes."

6.4 UI Simplicity

Tabs were used to isolate features: audio recording, file upload, and question answering. This reduced user confusion and allowed for modular development and testing.

---

7. Reflections and Future Work

This project was a valuable experience in building an end-to-end applied AI tool that is user-facing, privacy-conscious, and practically useful.

🔍 Future Improvements:

- Add support for segmenting large audio files for batch processing.
- Provide PDF/Markdown export in addition to `.txt`.
- Integrate RAG-based search using embeddings for richer Q&A.
- Optional local transcription for offline usage.

---

8. Conclusion

Notary offers a minimal yet powerful solution to a widespread real-world problem. Through intelligent design and integration of modern AI APIs, the project demonstrates how transcription, summarization, and search can be combined to turn raw speech into structured knowledge. The app serves both as a productivity tool and a technical showcase for conversational AI interfaces rooted in real data.

Appendix:

- Example file formats: `.mp3`, `.mp4`, `.wav`, `.m4a`
- OpenAI pricing: whisper-1 @ $0.006/min, gpt-4o input/output per token