Rationale

To be truly honest, I try to limit the use of AI in my own work as much as possible since I have found in the past that it makes me passive. I have mostly used it (ChatGPT) as a search engine or to write pieces of code from clear instructions. But, given the nature of this project and my obvious time constraint, I have leaned into AI as much as possible. Since I'm unfamiliar with most recent forms of codex and I didn't have the time to fully experiment as I would've liked to, I limited my own scope to the use of GitHub co-pilot, directly accessible via VScode.

I am aware that the purpose of this project is to detail my entire process as a builder, so for the sake of complete transparancy, here I will document from my planning process up until my final submission.

Here is how I initially planned my time:

1. Setup Rag framework (1hr)
2. Improve details (1hr)
3. Add initial data(CV) & experiment (0.5hr)
4. Enhance with more data (2hr)
5. Experiment & test (1.5hr)
6. Further improvements (2hr)
7. write-ups & clean-up (1hr)

As you can expect, things didn't work out completely as planned. I got lost in the prototyping and debugging and forgot to fully track the time as I planned, but if I have to guess I'd say the actual building time was definitely within budget. The write-ups and video might've taken me over slightly.

The data:

I decided to initially limit the dataset to just three core forms (my CV, a personal intro, and a video) because I wanted to maximize the system's effectiveness with minimal input. From past experience, I've learned that RAG systems highly depend on data quality; it's a classic case of "garbage in, garbage out." By starting with clean, high-value, and well-structured sources, I ensured the agent could deliver coherent, accurate, and relevant responses before introducing more varied or complex data. This approach allowed me to validate the pipeline, optimize retrieval accuracy, and establish a strong foundation for future expansion.

I used a short, real intro I wrote myself (spelling mistakes and all) because it’s the rawest version of my voice. This was used by an LLM to construct a persona which was used in every prompt.

My CV was added as a PDF as is to extract the text which was cleaned and chunked effectively.

The video is a short I found on YouTube that my church posted about a guiding philosophy I strongly identify with: “Faithfulness, not fame”. This adds a meaningful personal dimension that purely textual sources might not capture.

Finally, I included this document itself to offer direct insight into my problem-solving approach and to serve as a queryable artifact of my thought process.

Here is a complete list of prompts:

ChatGPT:

- I got the different tones from ChatGPT asking for unique tones to answer questions in
- modified chunking: how should i modify this structure: import os from typing import List from pypdf import PdfReader from pydub import AudioSegment import tempfile from assemblyai_utils import transcribe_audio_assemblyai SUPPORTED_TEXT = [".txt"] SUPPORTED_PDF = [".pdf"] SUPPORTED_AUDIO = [".mp3", ".wav", ".ogg", ".m4a"] import re import unicodedata def clean_pdf_text(text: str) → str: # Remove common bullet characters bullets = ["•", "*", "·", "o", "●", "▪", "■", "▶", "‣", "○"] for b in bullets: text = text.replace(b, " ") # Remove non-printable/control characters text = ''.join(c

```
for c in text if c.isprintable()) # Fix hyphenation at line breaks (e.g., "hy- #
phen" → "hyphen") text = re.sub(r"(\w+)-\n(\w+)", r"\1\2", text) # Normalize
whitespace (multiple spaces/tabs/newlines) text = re.sub(r"[ \t]+", " ", text)
text = re.sub(r"\n{2,}", "\n\n", text) # collapse multiple newlines # Remove
empty lines text = '\n'.join([line for line in text.splitlines() if line.strip()]) #
Unicode normalization text = unicodedata.normalize("NFKC", text) return text
def extract_text_from_pdf(file_path: str) → str: reader = PdfReader(file_path)
text = "" for page in reader.pages: text += page.extract_text() or "" text =
clean_pdf_text(text) return text def extract_text_from_audio(file_path: str) →
str: # Use AssemblyAI for transcription return
transcribe_audio_assemblyai(file_path) def save_uploaded_file(uploaded_file,
save_dir: str) → str: ext = os.path.splitext(uploaded_file.name)[1].lower()
save_path = os.path.join(save_dir, uploaded_file.name) with open(save_path,
"wb") as f: f.write(uploaded_file.getbuffer()) return save_path def
chunk_text_by_paragraphs(text: str, chunk_size: int = 500, overlap: int = 100) -
> list: # Try paragraph splitting first paragraphs = [p.strip() for p in
text.split('\n\n') if p.strip()] # If most of the text is in one paragraph, fall back to
fixed-size overlapping chunks if len(paragraphs) <= 1 or max(len(p) for p in
paragraphs) > chunk_size * 2: chunks = [] start = 0 while start < len(text): end
= min(start + chunk_size, len(text)) chunk = text[start:end].strip() if chunk:
chunks.append(chunk) if end == len(text): break start += chunk_size - overlap
return chunks return paragraphs def extract_text_from_file(file_path: str) →
list: ext = os.path.splitext(file_path)[1].lower() if ext in SUPPORTED_TEXT: with
open(file_path, "r", encoding="utf-8") as f: text = f.read() elif ext in
SUPPORTED_PDF: text = extract_text_from_pdf(file_path) elif ext in
SUPPORTED_AUDIO: text = extract_text_from_audio(file_path) else: text = ""
return chunk_text_by_paragraphs(text)
```

- I also got suggestions from here for the different tones, but I can’t find that chat

Co-Pilot:

(There was no way to automatically export so I manually copy and pasted.)

- Please setup a chatbot deployed on streamlit. This chatbot needs to be based on a rag structure, so also setup a vector store that is easiest. Use groq for an llm

- what env should i activate for my project?
- where can i get the logs
- The database needs to change to be dynamic and persistent
- please attach the db to the rag structure
- Now i want to expand the database to make it more dynamic. files that are dropped in the frontend should be embedded and stored persistently. Also add a multi-modal framework that can support .pdf and audio
- does the RAG structure utelise the entire database?
- what similarity measeure is used to retrieve content
- Integrate assemblyAI as transcription for audio data in the db
- is intro.txt also integrated into the db?
- how can i check this?
- It seems intro.txt is not integrated, only CV.pdf is?
- yes, i want checks to ensure that all the filesa are included and the database should be persistent
- So since im using streamlit its important to not re-embed something to spare time and also display all embedded files in frontend as a check
- Now use intro.txt to format a user role/prompt template so the answers from the llm sounds like this person. Make this a process and not hardcoded. Also add "intro.txt" as a config
- does an llm contrcut the persona from intro.txt or does it use intro.txt directly as prompt input?
- Okay I rather want intro.txt to be used by an llm to contruct a persona. This created persona is then prepended to the chatbots input
- what is DB_DIR
- where is it set?
- how does the content get chunked before embedding
- yes please implement chunking by context (usually paragraphs)

- how many chhunks are retrieved when searching
- does the way the content is chunked respond to the way the chunks are retreived
- I want the text that is retrieved to correspond exactly to the awy its chunked and embedded
- i dont think the pdf file uses newlines, what is another way of chunking you can add to this
- yes please update as suggested
- what other ways of cleaning a pdf for characters like bullet points are there
- yes please implement as much as you think necessery
- Move the drag and drop files into a section called "Enhance database". Here i want the user to be able to answer questions that get ambedded and added to the databse
- move this content you just added to the sidebar
- add a spinner when Add Q&A is clicked and clear the input boxes afterwards
- now move the entire enhance the database section to below the main promt and answer section and I want to suggested questions to popup. These should be modified after an answer is generated and repopulate
- but you didnt change any code...
- no, i believ the code wasnt changed in the code base. please check that the deseired changes were implimented
- yes please re-apply the changes
- the suggested questions should only pertain to the enhace database(Q&A) section
- i want the questions to be populated by the llm and the persona should be added as context. These questions should appear at the enhance database section at the bottom of tehe page
- now the ui is loading forever

- now Add a "context switch" mode that changes the prompt template to different tones depending on what is chosen
- use this and improve the UI stylistically: {

  ```
  "colors": {
      "primary_text": "#000000",
      "accent_text": "#4A6572",
      "background": "#FFFFFF"
      },
      "fonts": {
          "heading": "Century Gothic, sans-serif",
          "body": "ArialMTPro-Regular, Arial, sans-serif"
      }
  }
  ```
- can you make it less jarring?
- All boxes should correspond to the send Button's styling
- All text should be that same nlue/grey, also on teh sidebar. And the main title can be slightly darker
- the context switch text doesnt match yet
- i can't see my radio button text
- i still can't
- change all black input fields/button to a more suitable colour scheme
- change the success backdrop to be darker so the font colour can be seen
- Add the reponse answer text to a box that makes it stand out more
- please update my requirements.txt accordingly
- use pydub where you usually used pyaudio
- where is pyaudio used
- is audioop used

- the pydub isnt working, so For audio file handling, replace pydub with:
- any alternative you believe will work well with streamlit
- yes please clean up then any unnecessery reqs or imports
- the embedded_files.txt and persona_prompt.txt should not be added to the knowledge base
- How do i add an image of myself
- Add the image where you think it's fitting
- The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.
- add a seperator before the Enhance the Knowledge base section
- please update my readme and add a section explaining the system design choices
- add a afrontend features section
- Add debugging throughout such that someone new can easily understand the system if they turn debugging on
- where do i set the debug environment variable
- and where can i find the logs
- i can't find my logging output
- Please print debugging to a file and dont show anything related on the UI
- okay only log necessery info like what the query was, what was retrieved and actions that are duccessful or failed
- okay only log necessery info like what the query was, what was retrieved and actions that are duccessful or failed
- please go thorugh all the files and check for unnecesery logging
- what is this watchdog stuff
- but these are crowding my log file and its becoming massive
- add to my readme how to operate the debugging and update my gitignore

- edit the readme to be accurate with the system
- add to the readme a section explaining the different prompts used in the project