Project Proposal: AI Audio Notes (Lite)

Objective: To create a simplified version of NotebookLM that allows users to upload notes and receive a high-quality, synthesized "Audio Brief" summarizing the key takeaways.

---

1. High-Level Concept

The user provides text-based data (PDFs, text files, or links). The system uses a Large Language Model (LLM) to transform that data into a **natural, conversational script**, which is then converted into a **human-like audio file** (kind of like a podcast) for the user to listen to.

---

2. Frontend Development (The Interface)

The goal is a clean, minimal interface that focuses on file handling and playback.

- **File Upload Zone:** A drag-and-drop area for **PDF, .docx, and .txt** files.
- **Processing State:** A visual indicator (progress bar or spinner) to show when the AI is "Reading" and "Recording the Brief."
- **Audio Player:** A custom audio widget that includes:
    - **Playback Speed:** Options for 1.25x and 1.5x speed.
    - **Download Button:** To save the brief for offline listening.

---

3. Text Processing

This layer extracts the meaning from the documents and prepares it for audio.

- **Parsing:** We can use a library like PyPDF2 or LangChain to extract raw text from uploaded files.
- **Context Management:** If the notes are long, use **Recursive Character Text Splitting** to ensure the most important parts aren't cut off by the AI.
- **The "Scriptwriter" Prompt:** This is the most important part. You must instruct the LLM specifically for audio. It is important that we get the prompt right. Might take some trial and error to get the ideal prompt to convert the key concepts of the doc into a good conversation
- **Tech Recommendation: Gemini 1.5 Flash** (highly recommended for its massive context window and low cost).

---

4. Audio Processing (The Standout Feature)

This layer turns the text script into a professional-sounding voice/s.

- **Synthesis (TTS):** Convert the LLM's script into audio. For a "Lite" version, a single high-quality voice is best.
- **Voice Selection:** Choose a **Neural Voice**. These voices have natural pauses and emotional inflection, making them sound less like a robot.
- **Audio Formatting:** The output should be optimized in **MP3 or AAC** format to ensure the file size is small enough for fast streaming.
- **Tech Recommendation:** * **ElevenLabs API** (Highest quality "human" voices).
    - **OpenAI TTS-1** (Fastest and very cost-effective).

---

5. Summary Workflow (Step-by-Step)

1. **Input:** User uploads notes.
2. **Extract:** System cleans and extracts the text.
3. **Synthesis:** LLM converts the text into a **conversational script**.
4. **Vocalize:** The script is sent to the **TTS API**.
5. **Output:** The system returns an **Audio Player** to the frontend.

Some side notes:

- Maybe we can give user an option if they want a single voice (like a person briefing them about the notes), or two people (like two people discussing the notes as if it were a podcast topic)
- Select which part of the notes to summarise
- ELI5 feature (simplifies complex content into cheat sheets)
- Different types of summary (ELI5, motivational, gossip style etc)