**PDF Processing Pipeline**

This pipeline extracts information from PDF documents using Large Language Models (LLMs) and image processing techniques. It generates summaries, flashcards, and relevant image search queries for each page of the PDF.

**Setup and Running the Pipeline**

1. Clone this repository to your local machine.
2. Install the required dependencies:

**pip install PyMuPDF google-cloud-vision openai python-dotenv Pillow requests**

1. Set up your environment variables by creating a .env file in the project root with the following content:

**OPENAI\_API\_KEY**=your\_openai\_api\_key

**GOOGLE\_APPLICATION\_CREDENTIALS**=E:\Task\iconic-mariner-434611-r4-2231884d3d3d.json"

**GOOGLE\_API\_KEY**= ”AIzaSyBZzdxwFrrf0r8xH2TGKGAxY48EveeLiNE"

**GOOGLE\_CSE\_ID**= "630be4de38aeb4409"

1. Place your PDF file in the project directory or update the pdf\_path variable in the main script.
2. Run the main script:

**python main.py**

**Dependencies and API Keys Required**

* OpenAI API key for GPT-4 access
* Google Cloud Vision API credentials
* Google Custom Search API key and engine ID

**Approach and Design Decisions**

1. PDF Processing:
   * Used PyMuPDF to convert PDF pages to images for better compatibility with the Google Vision API.
2. Text Extraction:
   * Employed Google Cloud Vision API for accurate text extraction from images.
3. LLM Integration:
   * Utilized OpenAI's GPT-4 for generating summaries, flashcards, and search queries due to its advanced understanding of medical content.
4. Image Search:
   * Implemented Google Custom Search API to fetch relevant image URLs based on generated queries.
5. Error Handling:
   * Added basic error handling for JSON parsing and API responses to improve robustness.
6. Output Structure:
   * Organized results into separate JSON files for summaries, flashcards, and search queries with image URLs for easy access and readability.

**Prompt Engineering Strategy**

1. Page Summarization:
   * Prompt focuses on key cardiac physiology concepts and conduction system details.
   * Specifies a word limit to ensure concise summaries.
2. Flashcard Generation:
   * Requests a specific number of flashcards to ensure consistency across pages.
   * Emphasizes important medical concepts and diagrams.
   * Specifies the desired JSON output format for easy parsing.
3. Search Query Generation:
   * Asks for a short, general query to find relevant medical images.
   * Specifies the desired length and focus areas (cardiac anatomy, physiology, conduction system).