Teamified Coding Challenge – AI Data Engineering
AI Context Builder with Query Intent Recognition

Hi! 👋 Thanks for your interest in joining **Teamified** as a **Senior AI Data Engineer**.
This take-home challenge is designed to evaluate how well you can build prompt contexts from unstructured documents and generate intelligent responses using language models.

📌 Objective

You'll build a simple Python app that simulates part of a **Retrieval-Augmented Generation (RAG)** pipeline:

1. **Parse a PDF document** about Philippine History
2. **Chunk & embed the content** using vector similarity (with FAISS)
3. **Accept a user query** (in natural language)
4. **Retrieve the most relevant chunks**
5. **Build a context and pass it to an LLM**
6. **Print the LLM's final answer to the console**

📄 Input File

Download the public PDF file from NCCA: Philippine History Source Book

This document includes educational content about significant events, figures, and dates in Philippine history.

🧠 LLM Options (Choose One)

You may choose **one** of the following options for generating the final response:

✅ Option 1: Use OpenAI GPT

- Use GPT-3.5 (gpt-3.5-turbo) or later versions
- Read from this env variable: `TEAMIFIED_OPENAI_API_KEY`

- **Do not hardcode** your own key. We’ll insert our key when testing your code.

✅ Option 2: Use Any Free or Open LLM

- Use **any open or local language model** you're familiar with (e.g., hosted or on-device)
- We encourage you to be creative!
- Just make sure your script prints the actual model's response.

🛠️ What You Need to Build

Write a script that:

1. Loads and **chunks** the philippine_history.pdf file
2. **Embeds** each chunk into a FAISS index
3. Accepts a **user query**, like:
    - "When did the EDSA People Power Revolution happen?"
    - "Who is José Rizal and why is he important?"
    - "Tell me about the Spanish colonization of the Philippines."
4. Retrieves the **top-N most relevant chunks**
5. Builds a **context string** from them
6. Sends the context + question to your LLM
7. **Prints the LLM's response** clearly to the console

📦 Tech Requirements

- Python 3.8+
- Use FAISS (no Pinecone needed)
- You may use: LangChain, PyMuPDF, OpenAI SDK, etc.
- Include all dependencies in requirements.txt
- **BONUS:** Add as much unit tests as you can!

✅ Sample Output

```
$ python run.py

User Query: When did the EDSA People Power Revolution happen?


Retrieved Chunks:

- "The EDSA People Power Revolution occurred in February 1986..."

- "It led to the ousting of President Ferdinand Marcos..."


LLM Response:

"The EDSA People Power Revolution happened in February 1986 and marked
the end of Marcos' dictatorship in the Philippines."
```

🗂️ What to Submit

Please commit the following to your GitHub repo:

- README.md with:
  - Setup & run instructions
  - Notes on your LLM choice
- run.py or similar script
- requirements.txt
- .env.example with: **`TEAMIFIED_OPENAI_API_KEY`**
- We will provide the PDF separately — just make sure your code loads philippine_history.pdf from the local folder.
- Please make sure your GitHub repo is public.

⏰ Time Estimate

For a AI Data Engineer with strong Prompt Engineering background, this task should take about **3–4 hours**. But feel free to take your time. No rush! Focus on writing clean, modular, and readable code.

We’re excited to see how you approach this challenge. If you have additional ideas, feel free to implement them — we love initiative!

Good luck and enjoy the dive into history!
– The Teamified Engineering Team