leadtech;

Full-Stack Engineer Technical Task: AI-Powered CV Screener

1. Introduction & Goal

Welcome! We've designed this technical task to be a practical and interesting challenge. The goal is to build a simple, end-to-end prototype of an AI-powered tool for screening CVs.

This task will allow you to demonstrate your ability to integrate AI models, handle data processing pipelines, and build a functional user interface. We are most interested in your ability to learn, problem-solve, and deliver a working solution. Even if this is a new territory for you, we're excited to see your approach.

2. The Core Task

Your mission is to build a chat application that allows a user to ask questions about a collection of résumés (CVs). The project consists of three main parts:

1. Data Generation: Create a small dataset of 25-30 realistic-looking, fake CVs in PDF format.
2. Backend & AI Workflow: Develop a simple workflow to process these PDFs and make their content searchable and understandable by a Large Language Model (LLM). This is commonly known as a Retrieval-Augmented Generation (RAG) pipeline.
3. Frontend Chat Interface: Create a simple, functional web interface where a user can ask questions and get answers from the LLM based on the content of the CVs.

3. Requirements & Functionality

Core Requirements:

- **CV Generation:**
    - Generate 25-30 unique and fake CVs.
    - The CVs must be in PDF format.
    - They should appear realistic, including elements like an AI-generated photo, contact information, work experience, skills, and education sections.
    - The roles and languages used in the CVs are up to you.
    - **Use any LLM to generate the texts or images.**

- **RAG Workflow:**
    - The system must extract text from the provided PDF documents.
    - It must process and store this information in a way that an LLM can retrieve it.
    - Optional: Grounded on the data only from the CVs.
    - **Use any suitable tool or solution for this: Pinecone, LangChain, N8N, GCP...**

- **Chat Interface:**
    - A clean and simple user interface with a text input for questions and a display area for answers.
    - The LLM's responses must be based on the information contained within the CVs.
    - Each question can be treated as independent; you do not need to manage complex conversational history.
    - Optional: Source Indication: In the chat response, indicate which CVs were used as the source for the answer.
    - **Use tools like v0, bolt, lovable, or any other AI tool.**

4. Technical Guidelines & Suggestions

- **API Key:** You can choose any model you like suitable for the task. If necessary, you can start with free API keys from:
    - Google AI Studio: https://aistudio.google.com/apikey (free with limited use)
    - Openrouter: https://openrouter.ai/settings/keys (many models are free)
- **Creative Freedom:** You have complete freedom to choose your technology stack. The goal is a working product, not to follow a rigid set of rules. Don't feel the need to over-engineer the solution.
- **Hosting:** The final application does not need to be deployed or hosted online. Running it locally is perfectly fine.

5. Deliverables

Please provide the following:

- A Short Video Demonstration (3-10 minutes):
    - Record your screen using Loom or a similar tool.
    - Demonstrate the final application by asking it a few sample questions (e.g., "Who has experience with Python?", "Which candidate graduated from UPC?", "Summarize the profile of Jane Doe.").
    - Briefly walk us through your code and technical architecture, explaining the key decisions you made and the tools you chose.
    - Create an overview diagram of this complete workflow.

- Source Code :
    - Include the source code of the project.
    - (Optional but Recommended) Pushing your code to a public GitHub repository is a great way for us to see your work, but it is not a strict requirement.

6. Evaluation Criteria

We will be looking at your submission holistically, focusing on:

- Execution & Functionality: Does the application work as described?
- Thought Process: Your explanation of the architecture and technology choices.
- Code Quality: The clarity, structure, and readability of your code.
- AI Literacy: Your awareness of the relevant tools, models, and trends in the AI industry.
- **Learn & Adapt**: Your ability to tackle a new problem domain and produce a functional result is the most important factor.

7. Timeline

Please submit your deliverables within **2 days** from receiving this task.

If you have any questions, please don't hesitate to reach out to us. Good luck, and we look forward to seeing what you build!