Python Backend Developer Skill Test

Objective: Build a FastAPI backend for PDF processing, LLM summarization, and cloud-ready APIs from scratch.

Core Skills Evaluated

FastAPI Development (REST APIs, JWT auth)

LLM/Cloud Integration (OpenAI/Gemini, AWS)

PDF Processing (PyPDF2/pdfplumber)

Database (PostgreSQL/MongoDB)

Code Quality (Testing, Docker, Docs)

Task Instructions

1. Repository Setup

- Create a **new GitHub/GitLab repo** named `pdf-llm-backend`.

---

2. Core Features to Implement

A. PDF Upload & Text Extraction

- **Assume:** Sample PDFs (e.g., invoices, research papers) are provided by users.
- **Endpoints:**
  - `POST /upload`: Accept PDF files → save to local storage/S3 → extract text.

    - `GET /documents/{doc_id}`: Return extracted text + metadata (filename, upload time).
- **Libraries:** Use `PyPDF2` or `pdfplumber` for extraction. (Layout extraction preferred)

B. LLM Summarization

- **Assume:** Use OpenAI's GPT-4 or Gemini API (mock keys allowed for testing).
- **Endpoint:**
    - `POST /summarize/{doc_id}`: Generate a 2-sentence summary of extracted text.
    - `POST /Query/{doc_id}/{Question}`: Get contextual response for the question from the document.
- **Security:** Store API keys in environment variables.

C. Database & Pagination

- **Assume:** PostgreSQL/MongoDB for storage.
- **Pagination:** Add `GET /documents?page=1&limit=10`.

---

3. Additional Evaluation Tasks

- **Dockerize:** Add a `Dockerfile` for containerization.
- **Auth:** Implement JWT for `/summarize` endpoint.
- **AWS:** Deploy to Lambda/API Gateway (submit screenshot).
- **CI/CD:** Github Actions / CI-CD Pipeline can be managed to push the code to Infra
- **RAG:** Implemented RAG for Q/A session

Evaluation Criteria

Area	Weight	Key Metrics
API Functionality	30%	Endpoints work, PDF/LLM integration

Code Quality	25%	Clean code, tests, docs
Error Handling	20%	Graceful API failures, validation
Git Practices	15%	Atomic commits, descriptive PRs
Bonus Tasks	10%	Docker/AWS implementation

Submission Guidelines

1. **Repository:** Share the GitHub/GitLab link.
2. **README.md:** Include:
   - Setup instructions (env variables, dependencies).
   - Assumptions (e.g., "Used mock PDFs from `./samples`").
3. **Video Demo (2 mins):** Show:
   - PDF upload → text extraction → LLM summary.
   - Error handling (e.g., invalid PDFs).

Notes for Candidates

- **Testing:** Use `pytest` (5+ tests) or Postman.
- **Assume:** No frontend required—focus on APIs.

For submission & queries: careers@pente.ai | Video: Unlisted in YouTube or drive link