

Technical Assessment Brief: AI Engineer

Goal: Build a tiny RAG system that ingests one PDF, converts it to clean text, splits it into chunks, stores chunks + embeddings in a vector store, and answers user questions grounded in the doc with explicit citations to retrieved chunks.

Must-haves

1. Ingest

Choose **one**:

- **Gemini text extraction** (recommended): Use Gemini to extract clean plaintext from the uploaded file (PDF/images).
- **Local parser**: Use a PDF parser (e.g., `pypdf`, `pdfplumber`) to get plaintext. Document which path you chose and why.

2. Chunk & Embed

- Reasonable chunk size with overlap.
- Embeddings via Gemini Embeddings (or other preference) with a consistent dimensionality

3. Store

- **Postgres + pgvector** (recommended) **or** other vector storage.

4. Retrieve & Answer

- Embed the query; top-k search; compose a grounded answer with **sources**
- If evidence is insufficient, say "I don't know."

5. Hygiene & DX

- `README`, `.env.example`, no hard-coded secrets.
- Scripted run commands; basic logging & error messages.

Deliverables

- Source repo with instructions.
- **30-minute demo** covering architecture, a live query flow, and limitations/next steps.