KRID Dev Candidate Task

Scenario

A multinational materials-science company wants us to automatically extract structured information from patents and enable R&D scientists to “chat” with the extracted content.

Build a small, client-facing “demo prototype” of the system. You will receive **10 example patents**. You may **hard-code** or **manually extract** parts of the data if needed—the goal is to convincingly demonstrate the end experience using realistic technology and UX. AI-assisted programming is encouraged for this task.

Demo Prototype Minimum capabilities

- **Professional UI** where a user can:
    - Browse or select among the provided patents.
    - Ask a natural-language question.
    - Receive an **LLM-generated answer grounded in the patent content**, with **clear citations** (patent ID/page/section).
- **No visible bugs**; modern UX conventions (loading states, error messages, empty states).
- **Local run**: The system must run locally and be share-screen demoable.

Implementation notes (flexible, not prescriptive)

- Data layer can be a simple store (JSON/SQLite/files).
- You may use any LLM (mocked or real). If mocked, show how grounding/citations would work.
- Retrieval/extraction can be simplified, but the UX must **always** show source references.
  Provide a short **README** with setup/run steps and the tech stack.

Inputs provided

- Ten example patents to be used within the demo.

What we evaluate

- UX quality and polish under time constraints.
- Smart scoping choices (what you built vs. mocked).
- Sensible use of AI for coding/acceleration.
- Understanding of potential demo pitfalls (where your demo prototype might fail) to mitigate during the client call.