Take Home Test - (Junior to Mid Level LLM Engineer)

Build a small FAQ/data chatbot over one DOSM dataset using **Dify OR n8n OR any open-source workflow tool** (no coding preferred).
If coding is needed, use **LangChain** (bonus: **LangGraph**).
Use **any open-source OR paid LLM API**.

Scope & Constraints

- **Dataset**: Pick **one** dataset from https://open.dosm.gov.my/ (e.g., CPI, unemployment, population, median income).
    - Do not ingest everything — focus on **5–10 files/pages** max OR **≤20k tokens**.
    - Provide a **data card**: source URL(s), refresh cadence, license, last updated.
- **Chatbot**: Answer user questions about that dataset with **citations**.
- **Workflow**: Use a **workflow builder** (Dify, n8n, etc.).
- **RAG**: Required — vector store + embeddings.
- **No-code first**: Prefer visual flows. Code only where needed (LangChain allowed; **LangGraph = bonus**).

Required Features

1. **Ingestion & Indexing**

- Ingest your chosen dataset (CSV/JSON/HTML/PDF).
- Document **chunking strategy** and embedding model.
- Export/commit your **data card**.

2. **RAG Answering with Citations**

- Retrieve relevant passages and ground answers with **explicit sources**.
- Low-confidence fallback: clarifying question OR refusal.

3. **Workflow Orchestration**

- Visual workflow showing: input → retrieval → LLM → response.
- Include **one extra action** (e.g., export CPI to CSV, summarize unemployment trend, compare states).

4. **Evaluation & Monitoring**

- Provide an **eval set (15 queries)** with expected notes.
- Report **latency (p50/p95)**, **retrieval hit-rate**, **hallucination rate**.
- Basic logging/metrics (built-ins are fine).

5. **Responsible AI**

- Show how you handle outdated/missing data, disclaimers, or off-topic queries.
- Include **one refusal test case**.

Bonus Features

- **LangGraph** for agent state handling, retries, or guardrails.
- **Visualization**: return a simple chart/CSV from the data.
- **Agentic AI with Model Context Protocol (MCP)**:
    - Can use ready-to-use mcp server in (eg. https://mcpservers.org/) or Built it using (eg. Fastmcp framework)
    - Define one or more **tools/actions** (e.g., `export_csv`, `log_summary`).
    - Wire them so the chatbot can **invoke tools via MCP schema**.
    - Document how you set up MCP in your flow (workflow nodes OR code).
- Streaming responses to client.

Deliverables (via GitHub)

1. **README.md**

    - Quickstart (≤10 min run).
    - Tool choice + model provider (API keys redacted).
    - Data card.
    - RAG design (chunking, embeddings, k).
    - Eval methodology + results.
    - Limitations & future work.
2. **Exports / Config**
    - Dify/n8n (or other) flow export JSON/YAML.
    - KB export or ingestion script (if used).
    - If coding: LangChain (bonus: LangGraph) files + requirements.
3. **Demo Video (4–6 min)**
    - Project overview.
    - Data ingestion process.
    - At least 2 Q&A demos with citations.
    - One failure/low-confidence case.
    - One extra action triggered.
    - Evaluation results.
4. **Eval Set**
    - `eval/queries.jsonl` with 15 queries.
    - `eval/results.jsonl` (or screenshots).

Submission

- **GitHub repo link** (public or access-enabled).

- Include the **demo video** file or repo-hosted link.