Task 1: Handwritten Notes Digitization & Indexing

- Objective: Build a pipeline to digitize sample handwritten medical notes.
- Steps:
  - Use AWS Textract (free tier) or GCP Document AI trial.
  - Upload 3–5 sample handwritten images/PDFs (candidates may create dummy handwritten notes).
  - Extract text and clean it.
  - Store structured output in a local DB (MongoDB, PostgreSQL, or vector DB like FAISS/Chroma).
  - Build a simple search API (FastAPI/Flask/Node.js) to retrieve notes by keyword.
- Deliverable:
  - GitHub repo with code + README
  - Small demo (CLI or REST API endpoint)

Task 2: Clinical Record Summarization with an LLM

- Objective: Summarize digitized notes into structured JSON.
- Steps:
  - Use a free-tier LLM API (e.g., OpenAI free trial, Hugging Face models, or Ollama locally).
  - Batch process digitized notes.
  - Convert them into JSON format with fields: Patient, Diagnosis, Treatment, Follow-up.
  - Ensure consistency across multiple inputs.
- Deliverable:

    - Script in repo
    - Example JSON outputs

Task 3: Mini RAG Agent

- Objective: Build a small retrieval-augmented generation (RAG) pipeline.
- Steps:
    - Ingest at least 10 notes into a vector DB (FAISS/Chroma/Pinecone free-tier).
    - Use LangChain (Python) or LlamaIndex for query processing.
    - Build a chatbot/REST API that answers:
        - “Which patients had X diagnosis?”
        - “What treatment was prescribed most frequently?”
- Deliverable:
    - REST API or chatbot
    - Sample queries + outputs in README

Task 4: Cloud-Native Deployment

- Objective: Deploy one of the above services as a microservice.
- Steps:
    - Containerize with Docker.
    - Deploy on AWS Lambda/ECS Free Tier or GCP Cloud Run Free Tier.
    - Provide a working endpoint + short usage doc.
- Deliverable:
    - GitHub repo (with Dockerfile + deployment script)

    - Public endpoint (if possible, otherwise local Docker run instructions)

Expected Outcome (48–72 hrs)

- A GitHub repo with 4 modules (Tasks 1–4),
- Sample data + outputs,
- A running API/chatbot (local or cloud),
- Documentation showing thought process, AI-tool usage (Copilot/Tabnine optional), and design decisions.
    -