Multi-Agent Healthcare Assistant — Assignment (Associate Data Scientist)

Context (COVID-style scenario, *simulation only*): Build a proof-of-concept multi-agent system that accepts clinical artifacts (e.g., chest X-rays, ID docs), performs triage + treatment suggestion, and coordinates a mock e-pharmacy for doorstep delivery. **Strictly non-clinical**—for demonstration only.

Task (What you must build)

Create an end-to-end app with **4+ collaborating agents** and a **deployed, shareable link**.

> **Fresher track expectations:** Keep it simple, well-structured, and safe. Clear code, comments, and a working demo matter more than complex ML.

Core user flow:

1. Upload chest X-ray (PNG/JPG) + optional PDF report/ID → 2) System triages likely condition(s) & severity → 3) Suggests **non-prescriptive** medicine options (OTC only) with interactions & contraindication flags → 4) Match to nearest partner pharmacy with stock → 5) Offer optional **tele-consult escalation** (mock doctor roster) → 6) Place mock order & generate confirmation.

What we're testing for freshers: basic Python skills, clean modular design, safe reasoning, small data handling, and ability to deploy.

Required Agents (minimum)

- **Ingestion Agent**: Validates files, extracts text from PDFs (OCR), de-identifies PII.
- **Imaging Agent**: Runs a lightweight classifier on chest X-ray (use a tiny dummy model or rule-based stub; never claim diagnostic certainty). Output: {condition_probs, severity_hint}.
- **Therapy Agent**: Maps condition → **OTC** options (dummy formulary), checks age/allergy flags, basic interaction screen, and outputs advice text with safety disclaimers.

- **Pharmacy Match Agent**: Finds nearest store with stock; computes ETA & cost from **dummy inventory + geo**; reserves items.

- **Coordinator/Orchestrator**: Routes tasks, handles fallbacks, consolidates final plan; triggers **Doctor Escalation Agent** when confidence < threshold or red-flags present.

Safety/Scope Rules: No prescriptions; show “not medical advice.” Always surface **red flags** (e.g., chest pain + shortness of breath) → advise immediate care.

Tech/Deployment (must-haves)

- **Backend/UI (pick one):** Streamlit (recommended), Gradio, or FastAPI + simple HTML.

- **Agents:** Plain Python classes with clear JSON I/O (LangGraph optional). Keep logic readable.

- **Model:** Use a tiny/dummy `predict()` for X-ray (heuristic or pre-baked probabilities). No heavy training required.

- **Data:** Local CSV/JSON for pharmacies, stock, doctors, medicines, interactions (place under `/data`).

- **Deployment:** Streamlit Community Cloud or Render (free tier). **Submit public URL** + repo.

- **Observability:** Minimal event log (timestamped steps) visible in UI.

- **Docs:** README with how to run, architecture diagram (simple), limitations/safety.

Dummy Data (provide in repo under `/data`)

- **pharmacies.json**: `[{"id":"ph001","name":"MedQuick Andheri","lat":19.12,"lon":72.84,"services":["24x7","delivery"],"delivery_km":12}]`

- **inventory.csv**: `pharmacy_id,sku,drug_name,form,strength,price,qty`

- **doctors.csv**: `doctor_id,name,specialty,tele_slot_iso8601[]`

- **meds.csv** (OTC only):
  `sku,drug_name,indication,age_min,contra_allergy_keywords`

- **interactions.csv**: `drug_a,drug_b,level,note`

- **zipcodes.csv**: `pincode,lat,lon`

API/Schema Contracts

Upload → Ingestion Agent Output

```
JSON
{
  "patient": {"age": 45, "allergies": ["ibuprofen"]},
  "xray_path": "./uploads/x1.png",
  "notes": "cough, low-grade fever"
}
```

Imaging Agent Output

```
JSON
{"condition_probs": {"pneumonia": 0.42, "normal": 0.38,
"covid_suspect": 0.20}, "severity_hint": "mild"}
```

Therapy Agent Output

```
JSON
{"otc_options": [{"sku":"OTC001","dose":"500
mg","freq":"q8h","warnings":["contains paracetamol"]}],
"red_flags": ["SpO2 < 92%"]}
```

Pharmacy Match Output

```
JSON
{"pharmacy_id":"ph001","items":[{"sku":"OTC001","qty":10}],"eta_min": 45,"delivery_fee": 25}
```

Deliverables (submit all)

1. **Public app URL** (no login) and **GitHub repo**.
2. **README** with setup, simple agent diagram, and **limitations/safety**.
3. **Sample run**: screenshots + one sample order JSON.
4. **Tests**: at least 3 unit tests for agent hand-offs.
5. **Slide deck (5–8 slides)** for panel: problem framing, architecture, data mocks, demo flow, safety/limits, next steps.

Constraints & Ethics

- Prominent **disclaimer**: “Educational demo, not medical advice.”
- No storage of real PHI; treat all uploads as anonymous; mask IDs.
- Avoid prescription-only claims; route to doctor for anything uncertain.

Panel Presentation

- **Live review:** 1-hour call with a panel on **Friday**. You will present the solution (10–12 min demo + 10 min slides) followed by Q&A. Bring a backup recording or screenshots in case the live demo fails.
- **Deck template (suggested 5–8 slides):**
    1. Problem & scope (safety constraints)
    2. System architecture & agent roles

3. Data mocks (pharmacy, meds, doctors) & assumptions
4. Demo journey (upload → triage → OTC suggestions → pharmacy match)
5. Risks, red-flags, and escalation
6. Deployment, logs, and known limitations
7. Next steps if given 2 more weeks

Timeline

- **Start:** Today
- **Submission (App + Repo + Slides):** within 2 days
- **Panel Presentation:** On the 3rd day after assignment