Grünenthal Senior Data Scientist – AI Agent Case Study

About Grünenthal

Grünenthal is a privately-owned, mid-sized pharmaceutical company headquartered in Germany. We are global leaders in pain management and related diseases. Our core brands include Tramadol, Tapentadol, and Vimovo.

Our team works for the Global Operations division. We are responsible for Data & Analytics in manufacturing, supply chain, quality, and procurement. The mission of Global Operations is to ensure safe, efficient and reliable product supply to our patients.

Your Mission

You will build and deploy a simple working AI Agent with a chatbot UI for Grünenthal. The goal is to simulate how our teams could use a modern AI chatbot to answer real-world questions using:

- Tool 1: Neo4j graph database
- Tool 2: FDA Adverse Events API
- Tool 3: Grünenthal's financial report
- Agent + Chatbot UI

IMPORTANT:
- Spend no more than 3 hours.
- Use GenAI, pre-built code, and shortcuts.
- Focus on practical reasoning — not on building a production app.

Required Tools

1️⃣ Neo4j Sandbox Tool

Setup:
- Go to https://sandbox.neo4j.com/
- Create a Project using the Healthcare Analytics template.
- Note your Bolt URI, username (neo4j), and password.

Python example:

```
!pip install neo4j

from neo4j import GraphDatabase

URI = "bolt://44.192.43.48:7687"
AUTH = ("neo4j", "deeds-combs-rays")

driver = GraphDatabase.driver(URI, auth=AUTH)
driver.verify_connectivity()
print("✅ Connected to Neo4j!")

with driver.session() as session:
  results = session.run("MATCH (n) RETURN n LIMIT 5").data()
  print(results)
```

Question example: “Which manufacturers are connected to drugs which contain TRAMADOL in its name?”

Tip: Include the Neo4j schema as metadata. The drug names are capitalized in the Healthcare Analytics template.

2 FDA API Tool

Use the OpenFDA API: https://open.fda.gov/apis/drug/event/

Python example:

```
import requests

def get_adverse_events(drug_name: str, limit: int = 10):
    url = (
        f"https://api.fda.gov/drug/event.json?"
        f"search=patient.drug.medicinalproduct:{drug_name}&limit={limit}"
    )
    response = requests.get(url)
    return response.json()

print(get_adverse_events("TRAMADOL"))
```

Question example: “What are the top 10 most recent adverse events registered for a drug containing TRAMADOL in its name?”

3️⃣ Financial Report Tool — Two Options

Option A:
- Download a recent Grünenthal financial or annual report (or placeholder PDF).
- Use PyMuPDF or LangChain's PDF loader.

Example PDF reader:

```
!pip install PyMuPDF

import fitz

def read_pdf(path):
    doc = fitz.open(path)
    text = ""
    for page in doc:
        text += page.get_text()
    return text

pdf_text = read_pdf("Grunenthal_Annual_Report.pdf")
print(pdf_text[:500])
```

Option B:
- Use an Internet Search Tool (SerpAPI, Bing, Google).

Agent + Chatbot UI

Build your chatbot using frameworks such as Streamlit, Flask, or FastAPI. Use LangChain, LangGraph, AutoGen, CrewAI, Semantic Kernel or any other agent orchestration framework. The agent must:

- Decide which tool to use
- Run dynamic queries
- Return clear answers

Deployment

Deploy your chatbot publicly on Streamlit Community Cloud, Hugging Face Spaces, Render.com, Railway.app or your own favorite deployment service.

Document Your Top 10 Prompts

With your delivery, include:

- Your top 10 prompts you used while building and testing your agent.
- A short note on why each was helpful.

Submission

We must have visibility in your entire source code. Your solution must be deployed; we must be able to use the chatbot online.

Submit via email:

1. Chatbot link
2. Username and password (if needed)
3. Source code as ZIP or repo (GitHub or others)
4. Clear instructions (if needed)
5. Your top 10 prompts

Summary

Deliver a simple, working AI Agent that:

- Connects to Neo4j
- Calls the FDA API
- Answers questions about the latest Grünenthal annual report (PDF or Search)
- Uses GenAI to decide which tool to run
- Responds clearly in chat

Submit all documents within the deadline

Keep it practical — focus on smart shortcuts, not perfection!

Good luck!

The Global Operations Data & Analytics Team