Lesson 3

Querying AI Models

Using APIs

Cornell University
Systems Engineering

Dr. Tim Fraser
Assistant Teaching Professor

Summary

01

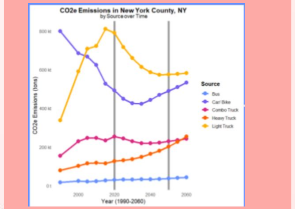
Discussion: Your Labs

02

Recap: What are LLMs?

03

Data Reporting with AI



04

What is Ollama?

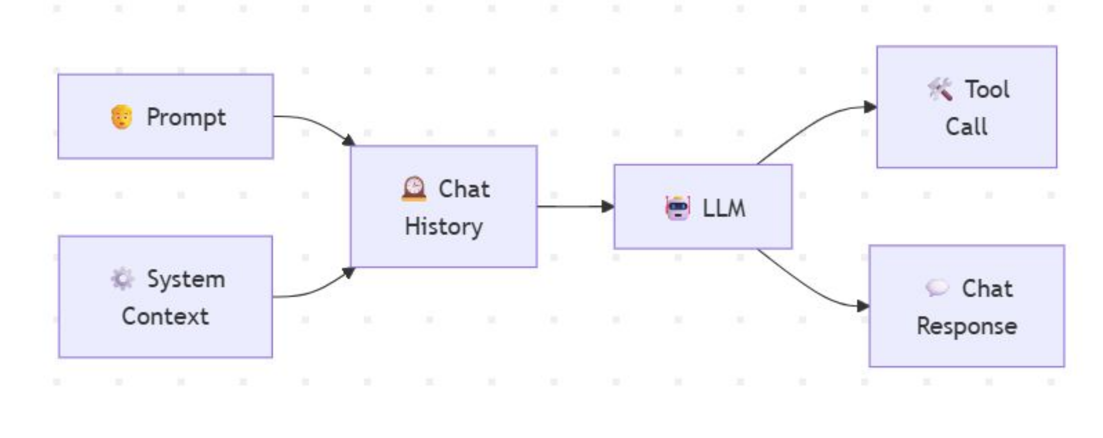
Discussion

- How did your lab go last week?
- What worked when prompting the AI?
- What didn’t work?
- Ideas for how you will use cursor next?

🤖 Recap: What are LLMs?

⚙️ How do LLMs work?

- 🧩 **System prompt**: sets rules, behavior, and constraints (usually hidden from users)
- 🗣️ **User prompt:** what you explicitly ask for
- 🕰️ **Chat history**: prior turns that shape context and continuity
- 🧮 Model combines all three → predicts next tokens until a stop condition



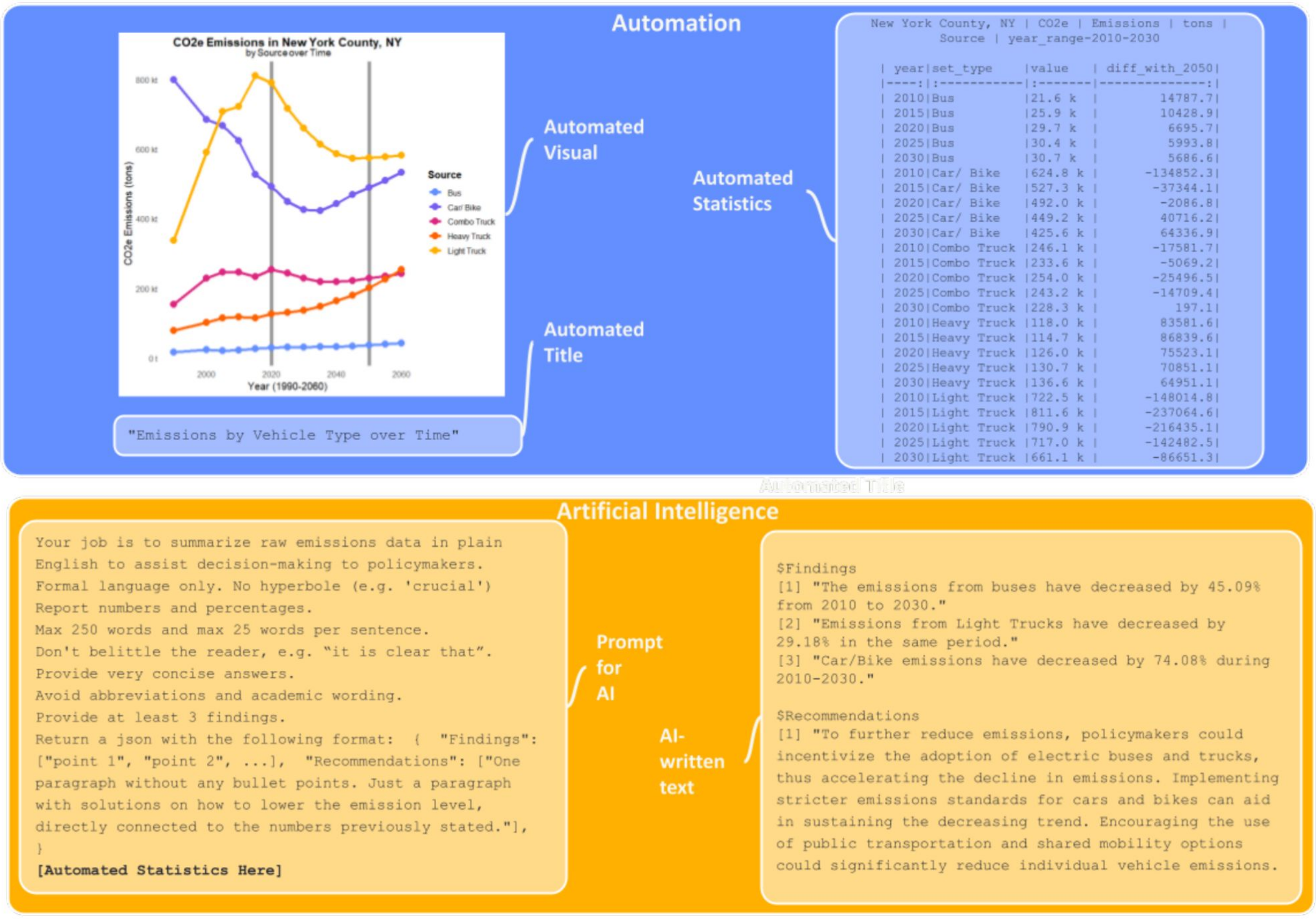
🌐Example LLM Providers

- OpenAI's GPT series
- Google's Gemini
- Cursor's Composer

🤖 Data Reporting with AI

Components

- Your job
- Internal structure
- Syntax
- Output format
- Summarize Stats!



```
"Emissions by Vehicle Type over Time"
```

```
New York County, NY | CO2e | Emissions | tons |
        Source | year_range-2010-2030

| year|set_type     |value    | diff_with_2050|
|----:|:------------|:--------|--------------:|
| 2010|Bus          |21.6 k   |        14787.7|
| 2015|Bus          |25.9 k   |        10428.9|
| 2020|Bus          |29.7 k   |         6695.7|
| 2025|Bus          |30.4 k   |         5993.8|
| 2030|Bus          |30.7 k   |         5686.6|
| 2010|Car/ Bike    |624.8 k  |      -134852.3|
| 2015|Car/ Bike    |527.3 k  |       -37344.1|
| 2020|Car/ Bike    |492.0 k  |        -2086.8|
| 2025|Car/ Bike    |449.2 k  |        40716.2|
| 2030|Car/ Bike    |425.6 k  |        64336.9|
| 2010|Combo Truck  |246.1 k  |       -17581.7|
| 2015|Combo Truck  |233.6 k  |        -5069.2|
| 2020|Combo Truck  |254.0 k  |       -25496.5|
| 2025|Combo Truck  |243.2 k  |       -14709.4|
| 2030|Combo Truck  |228.3 k  |          197.1|
| 2010|Heavy Truck  |118.0 k  |        83581.6|
| 2015|Heavy Truck  |114.7 k  |        86839.6|
| 2020|Heavy Truck  |126.0 k  |        75523.1|
| 2025|Heavy Truck  |130.7 k  |        70851.1|
| 2030|Heavy Truck  |136.6 k  |        64951.1|
| 2010|Light Truck  |722.5 k  |      -148014.8|
| 2015|Light Truck  |811.6 k  |      -237064.6|
| 2020|Light Truck  |790.9 k  |      -216435.1|
| 2025|Light Truck  |717.0 k  |      -142482.5|
| 2030|Light Truck  |661.1 k  |       -86651.3|
```

```
Your job is to summarize raw emissions data in plain
English to assist decision-making to policymakers.
Formal language only. No hyperbole (e.g. 'crucial')
Report numbers and percentages.
Max 250 words and max 25 words per sentence.
Don't belittle the reader, e.g. "it is clear that".
Provide very concise answers.
Avoid abbreviations and academic wording.
Provide at least 3 findings.
Return a json with the following format:  {   "Findings":
["point 1", "point 2", ...],   "Recommendations": ["One
paragraph without any bullet points. Just a paragraph
with solutions on how to lower the emission level,
directly connected to the numbers previously stated."],
}
[Automated Statistics Here]
```

```
$Findings
[1] "The emissions from buses have decreased by 45.09%
from 2010 to 2030."
[2] "Emissions from Light Trucks have decreased by
29.18% in the same period."
[3] "Car/Bike emissions have decreased by 74.08% during
2010-2030."

$Recommendations
[1] "To further reduce emissions, policymakers could
incentivize the adoption of electric buses and trucks,
thus accelerating the decline in emissions. Implementing
stricter emissions standards for cars and bikes can aid
in sustaining the decreasing trend. Encouraging the use
of public transportation and shared mobility options
could significantly reduce individual vehicle emissions.
```

Figure 3. Components for a Block of Content with Generative AI, NY

🧮 What is Ollama?

- **An open-source software for serving many different large-language models.**
- Primary solution if you want to choose your model or protect your data.
- 1 consistent API for serving and querying models
- Cloud models also available, giving faster responses.
- Cloud model usage limits:
  - **Free**: Light usage—chat, quick questions, trying out models
  - **Pro:** Day-to-day work—RAG, document analysis, and coding tasks
  - **Max:** Heavy, sustained usage—coding agents, batch processing, and data automation

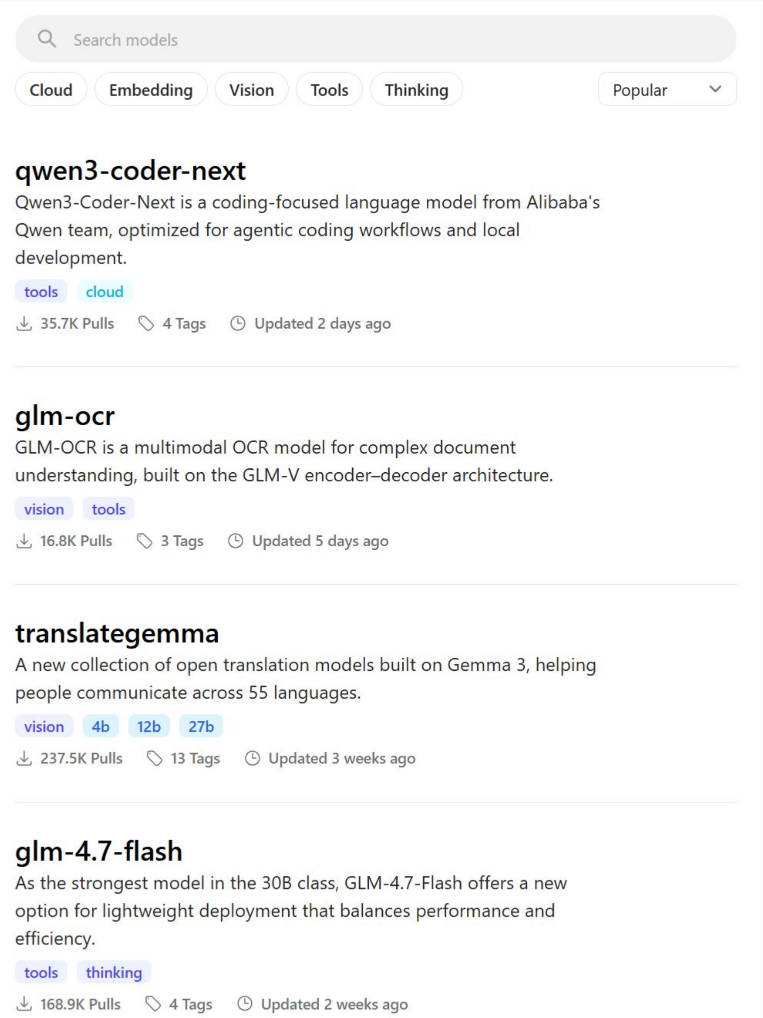
Learn more about Ollama Cloud pricing

🧮 What is Ollama?

>100s of models to choose from, some with extra functions!

User process:

- Search for models
- Download model
- Serve Model
- Query Model



qwen3-coder-next

Qwen3-Coder-Next is a coding-focused language model from Alibaba's Qwen team, optimized for agentic coding workflows and local development.

tools cloud

35.7K Pulls 4 Tags Updated 2 days ago

glm-ocr

GLM-OCR is a multimodal OCR model for complex document understanding, built on the GLM-V encoder–decoder architecture.

vision tools

16.8K Pulls 3 Tags Updated 5 days ago

translategemma

A new collection of open translation models built on Gemma 3, helping people communicate across 55 languages.

vision 4b 12b 27b

237.5K Pulls 13 Tags Updated 3 weeks ago

glm-4.7-flash

As the strongest model in the 30B class, GLM-4.7-Flash offers a new option for lightweight deployment that balances performance and efficiency.

tools thinking

168.9K Pulls 4 Tags Updated 2 weeks ago

🧮 Why use Ollama?

- Ollama does not record, log or train on any prompt or response data.
- Can run as many models as your hardware supports.
- Cloud models have concurrency limits by plan.
- All cloud requests are encrypted in transit.
- Made for offline use → cloud features optional.

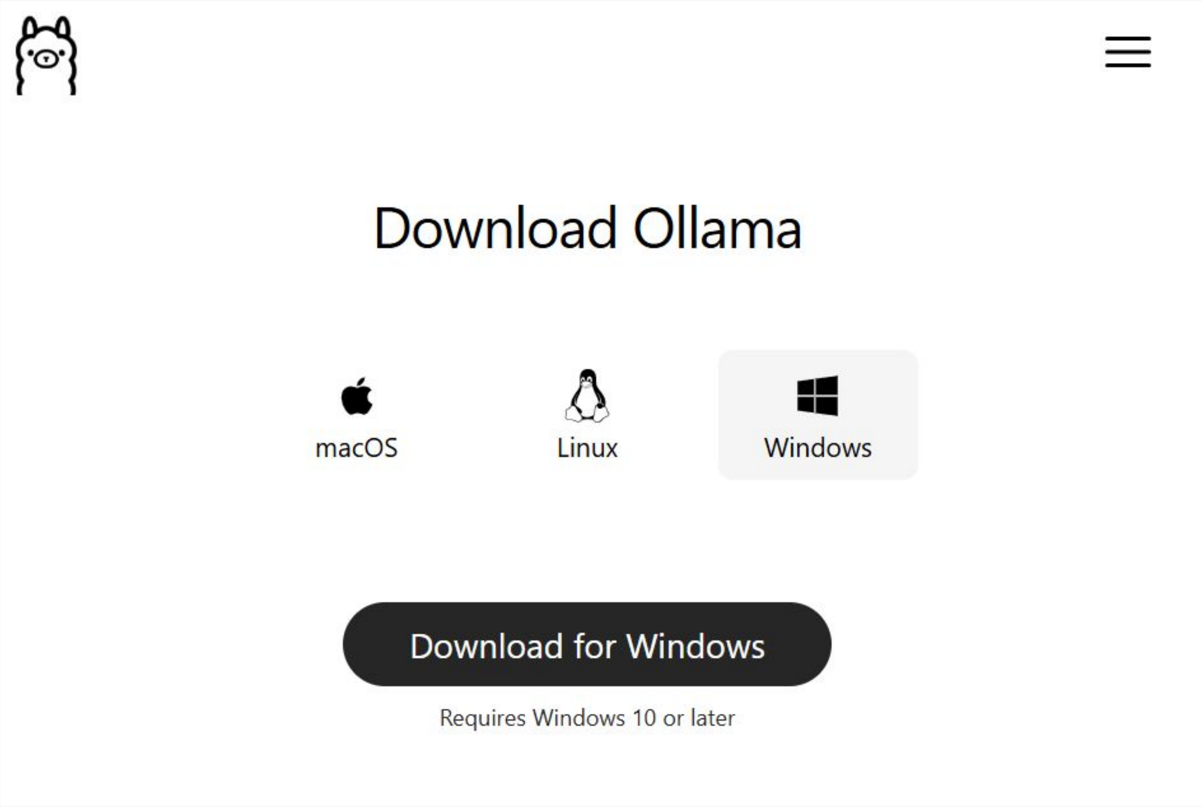
Learn more about Ollama Cloud pricing

Install Ollama

Please go ahead and install Ollama from this link. **ollama.com/download**

You will need to add Ollama to **PATH**. Cursor can help - OR, for an imperfect solution, use your **.bashrc** file.

See mine here: https://github.com/timothyfraser/dsai/blob/main/.bashrc



Download Ollama

Requires Windows 10 or later

Install Ollama

Please go ahead and install Ollama from this link.
ollama.com/download

You will need to add Ollama to **PATH**. Cursor can help - OR, for an imperfect solution, use your **.bashrc** file.

See mine here:
https://github.com/timothyfraser/dsai/blob/main/.bashrc

How to customize your .bashrc file

```
export PATH="$PATH:/c/Users/tmf77/AppData/Local/Programs/Ollama"
alias ollama='/c/Users/tmf77/AppData/Local/Programs/Ollama/ollama.exe'
```

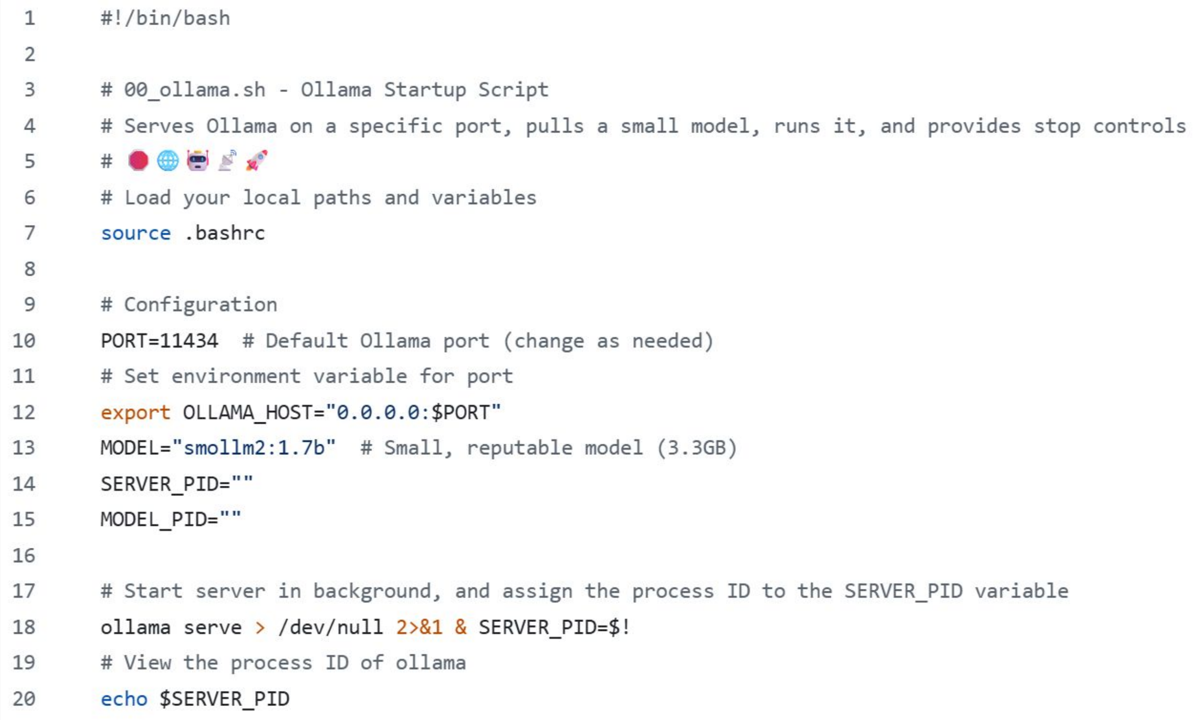
How to load your .bashrc file

```
source ./.bashrc
```

🧮 How to serve Ollama!

Customize and run this shell script in git bash!

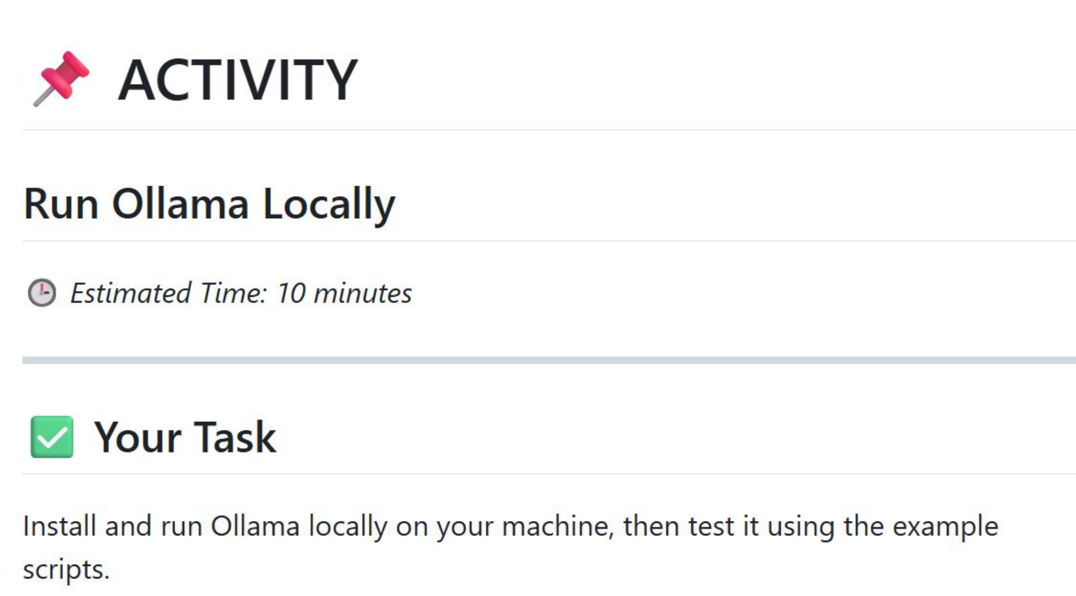
https://github.com/timothyfraser/dsai/blob/main/03_query_ai/01_ollama.sh



```bash
#!/bin/bash

# 00_ollama.sh - Ollama Startup Script
# Serves Ollama on a specific port, pulls a small model, runs it, and provides stop controls
# 🛑 🌐 🤖 📡 🚀
# Load your local paths and variables
source .bashrc

# Configuration
PORT=11434  # Default Ollama port (change as needed)
# Set environment variable for port
export OLLAMA_HOST="0.0.0.0:$PORT"
MODEL="smollm2:1.7b"  # Small, reputable model (3.3GB)
SERVER_PID=""
MODEL_PID=""

# Start server in background, and assign the process ID to the SERVER_PID variable
ollama serve > /dev/null 2>&1 & SERVER_PID=$!
# View the process ID of ollama
echo $SERVER_PID
```

ACTIVITIES

🌐ACTIVITY



📌 ACTIVITY

Run Ollama Locally

🕒 *Estimated Time: 10 minutes*

✅ Your Task

Install and run Ollama locally on your machine, then test it using the example scripts.

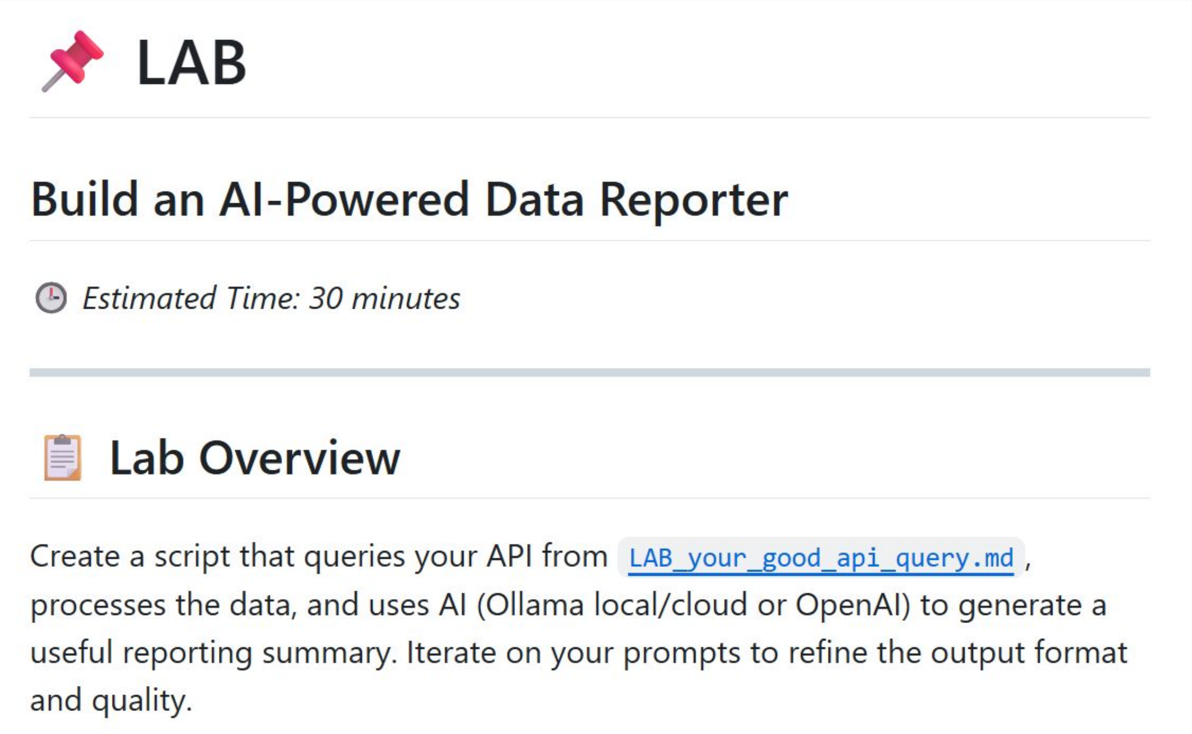
https://github.com/timothyfraser/dsai/blob/main/03_query_ai/ACTIVITY_ollama_local.md

🌐ACTIVITY

More coming on Wednesday!
See Github!

https://github.com/timothyfraser/dsai/blob/main/03_query_ai/README.md

🌐LAB



📌 LAB

Build an AI-Powered Data Reporter

🕒 *Estimated Time: 30 minutes*

📋 Lab Overview

Create a script that queries your API from `LAB_your_good_api_query.md`, processes the data, and uses AI (Ollama local/cloud or OpenAI) to generate a useful reporting summary. Iterate on your prompts to refine the output format and quality.

https://github.com/timothyfraser/dsai/blob/main/03_query_ai/LAB_ai_reporter.md

Happy coding!