Large Language Models

Walter van Heuven
waltervanheuven.net

Data club
4 March 2025

Large Language Models

Generative AI - Large Language Models (LLMs)

- Tokenizer, Prompting, Tools.
- Examples.
- Disadvantages of paid closed-source LLMs.

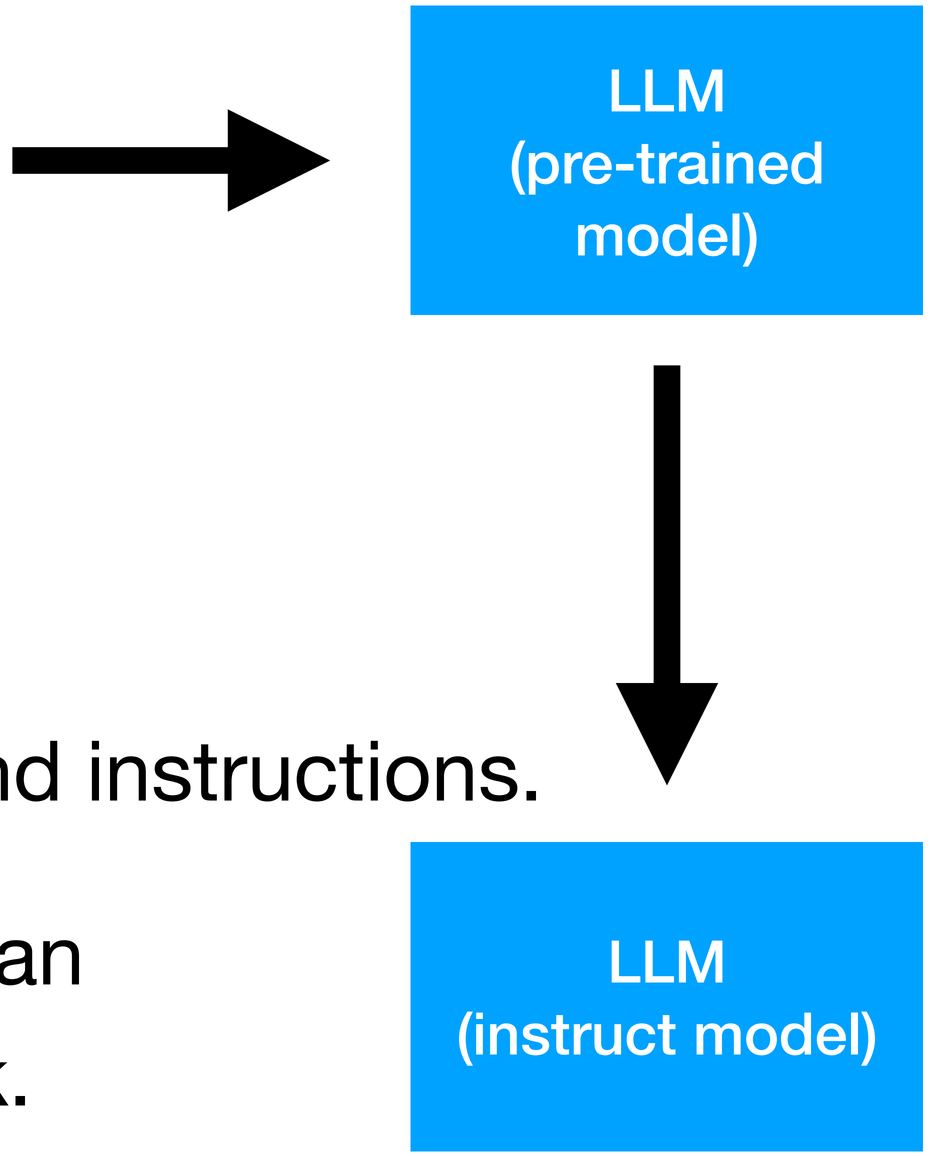
Running LLMs locally

- Apps and frameworks.
- Retrieval Augmented Generation (RAG).

Running LLMs through an API (Application Programming Interface)

- Local models.
- ChatGPT, Claude, Gemini, etc..
- Microsoft Azure.

Large Language Models



Pre-training

- Tokens.
- Predict next-word.

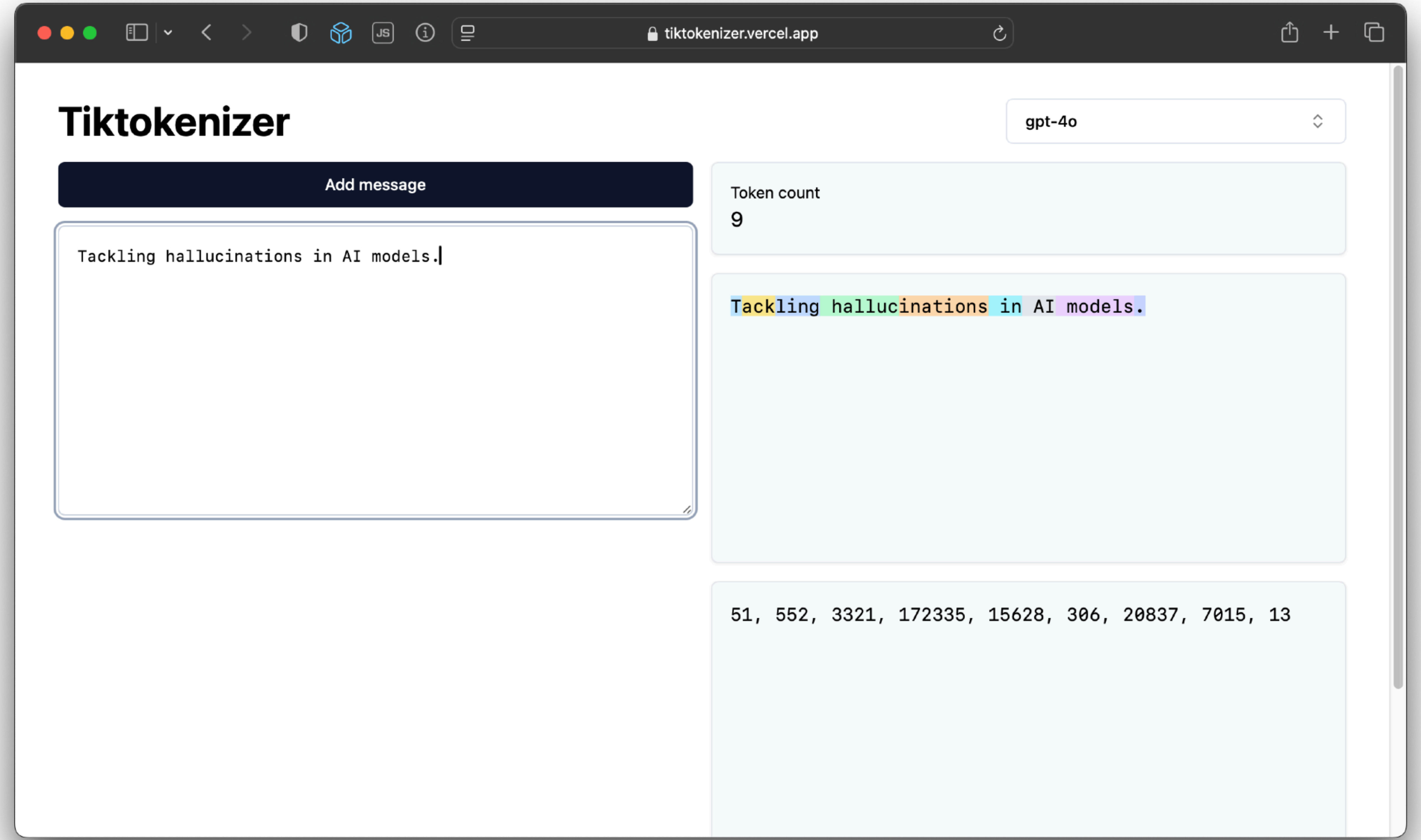
Post-training

- Fine-tuning to respond to questions and instructions.
- Reinforcement Learning (RL) with human feedback. RL without human feedback.

LLM model (LTM) - Context window (working memory)

Tokens

https://tiktokenizer.vercel.app/



Tiktokenizer

gpt-4o

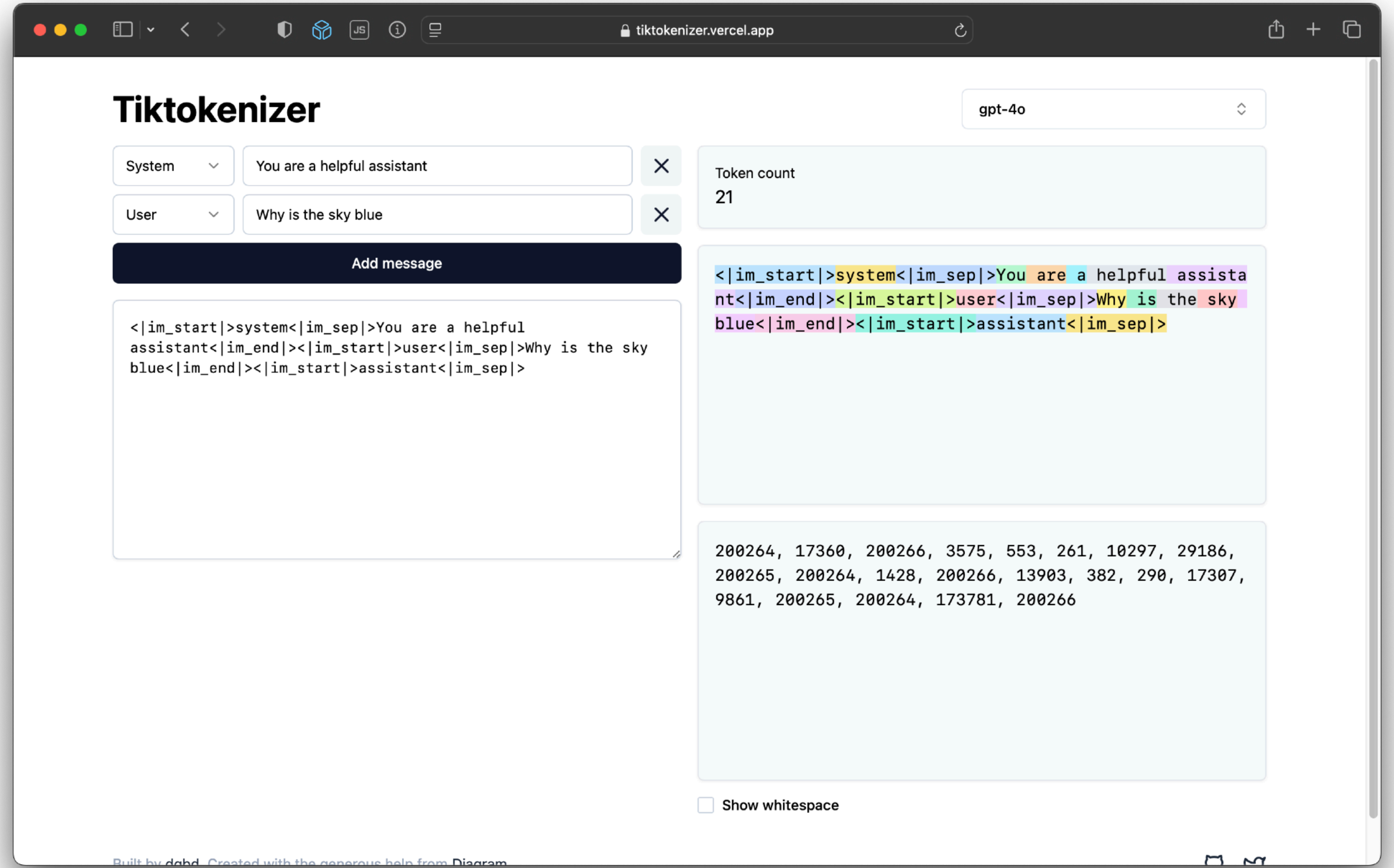
Token count
9

```
Tackling hallucinations in AI models.
```

```
Tackling hallucinations in AI models.
```

```
51, 552, 3321, 172335, 15628, 306, 20837, 7015, 13
```

Tokenizer



System

You are a helpful assistant

User

Why is the sky blue

```
<|im_start|>system<|im_sep|>You are a helpful
assistant<|im_end|><|im_start|>user<|im_sep|>Why is the sky
blue<|im_end|><|im_start|>assistant<|im_sep|>
```

Token count
21

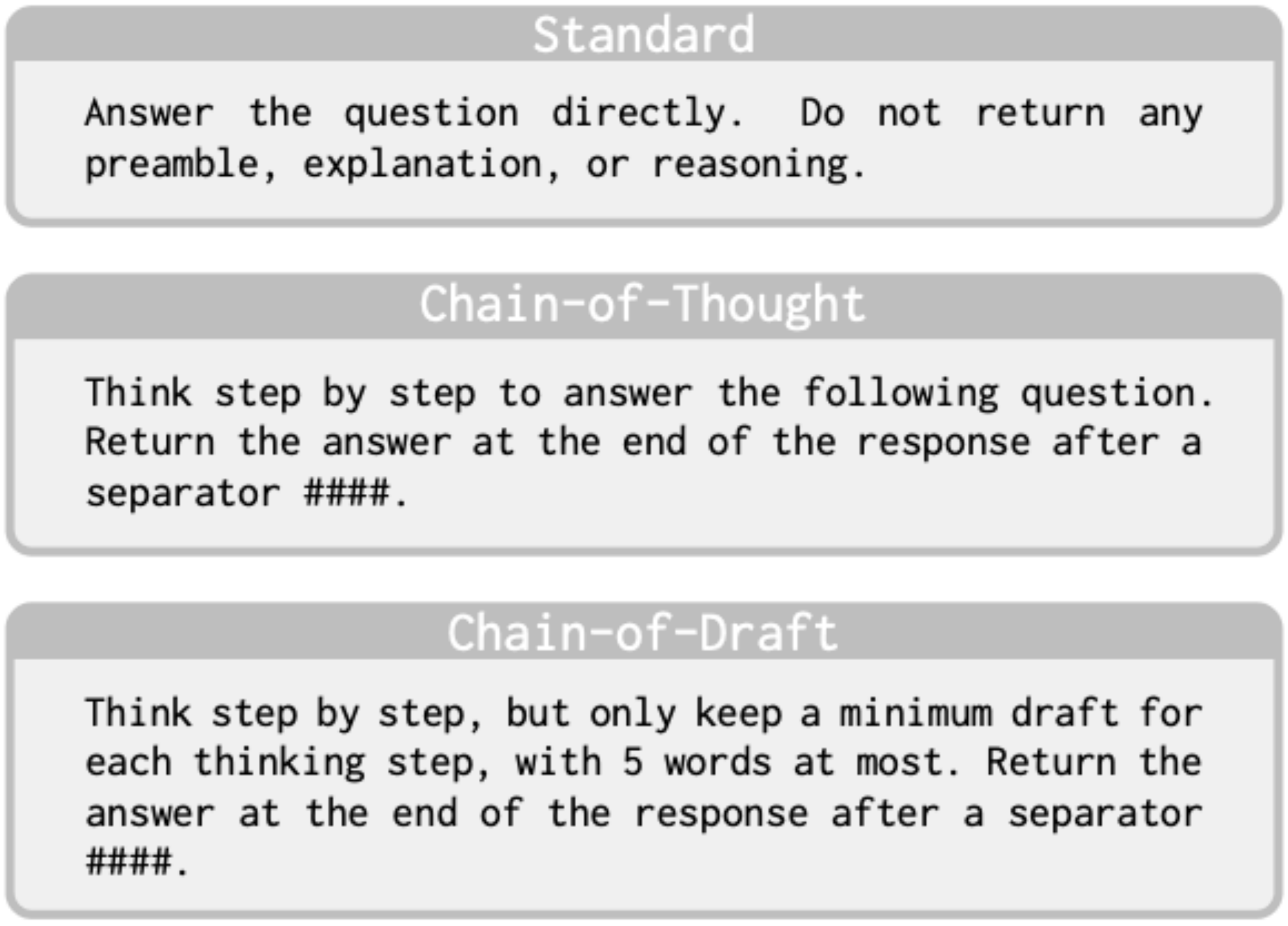
```
<|im_start|>system<|im_sep|>You are a helpful assista
nt<|im_end|><|im_start|>user<|im_sep|>Why is the sky
blue<|im_end|><|im_start|>assistant<|im_sep|>
```

```
200264, 17360, 200266, 3575, 553, 261, 10297, 29186,
200265, 200264, 1428, 200266, 13903, 382, 290, 17307,
9861, 200265, 200264, 173781, 200266
```

Prompting

Prompt and history of chat (context window) important to improve response.

Chain of Draft: Thinking Faster by Writing Less (Xu et al., arxiv)



Standard

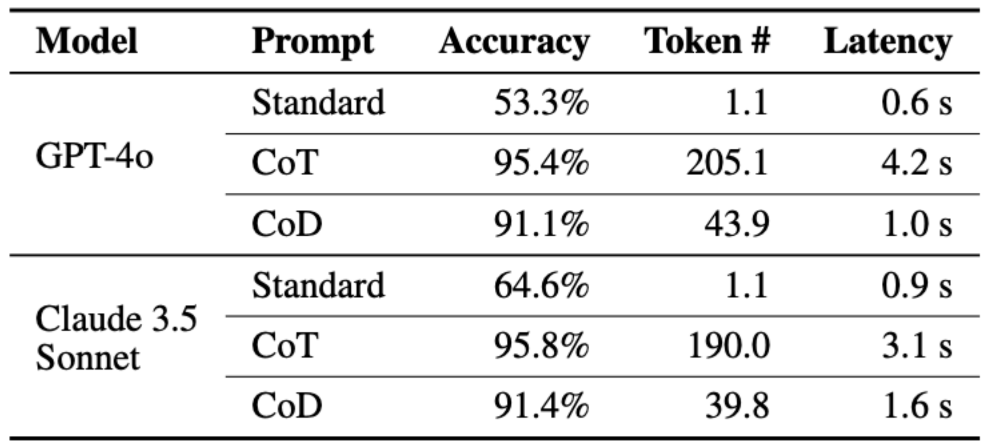
Answer the question directly. Do not return any preamble, explanation, or reasoning.

Chain-of-Thought

Think step by step to answer the following question. Return the answer at the end of the response after a separator ####.

Chain-of-Draft

Think step by step, but only keep a minimum draft for each thinking step, with 5 words at most. Return the answer at the end of the response after a separator ####.



Model	Prompt	Accuracy	Token #	Latency
GPT-4o	Standard	53.3%	1.1	0.6 s
	CoT	95.4%	205.1	4.2 s
	CoD	91.1%	43.9	1.0 s
Claude 3.5 Sonnet	Standard	64.6%	1.1	0.9 s
	CoT	95.8%	190.0	3.1 s
	CoD	91.4%	39.8	1.6 s

Table 1: GSM8K evaluation results.

Humanity's last exam:

https://scale.com/leaderboard

Tools

LLMs get much more powerful with tools:

- Programming language (Python, Javascript): Claude, ChatGPT

- Web search (ChatGPT)

- Data analysis (ChatGPT - Python data analysis and visualisation)

- Artifacts (Claude)

Examples

Example 1. Claude 3.7

1 Create an energy usage calculator. You enter daily electricity and gas usage in kWh per day and it shows total usage per year.

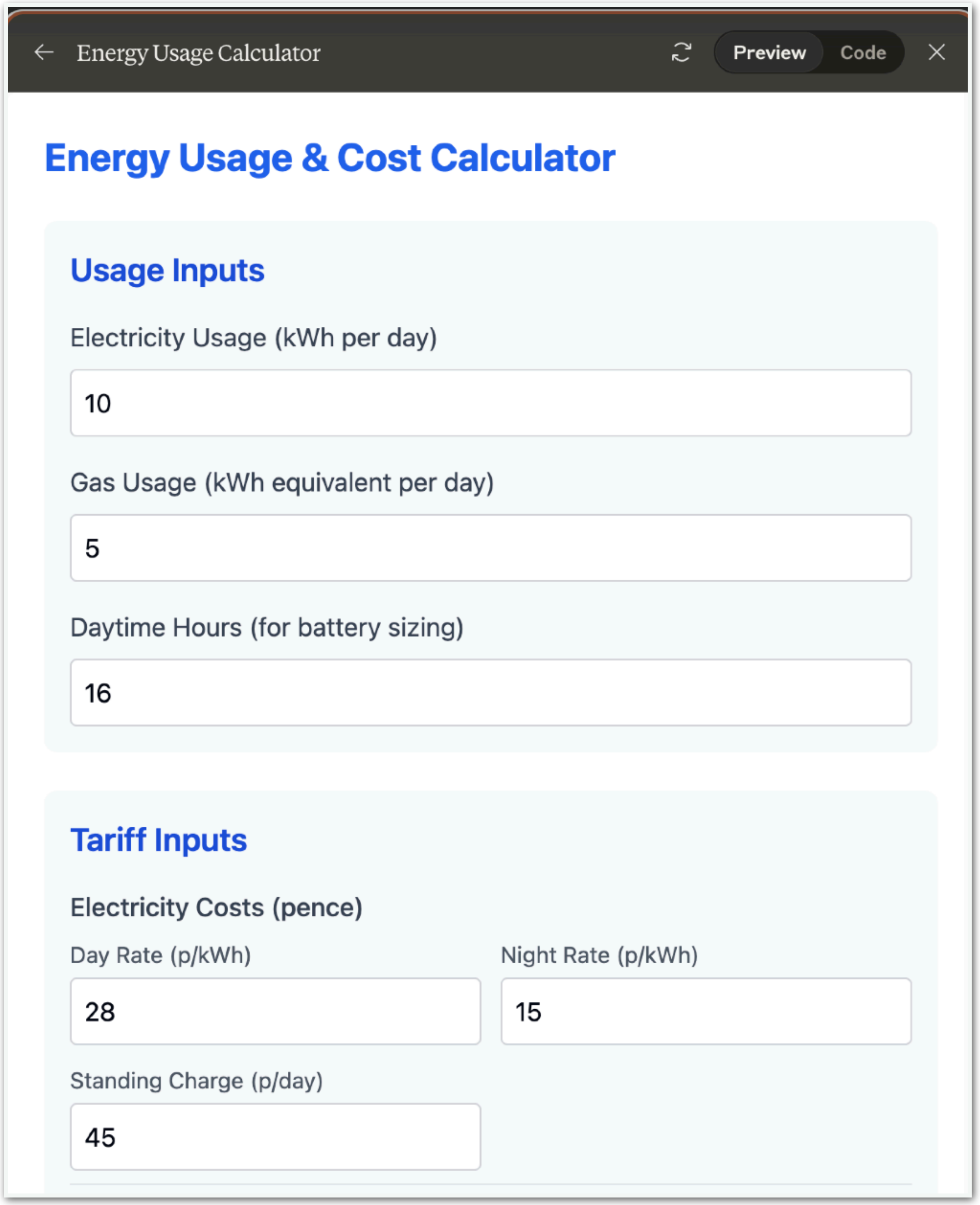
In addition, it should indicate the size of battery that would enable cheap recharging at night and using the battery during the day.

2 Add ability to calculate costs of energy use and cost of battery and amount of years to repay.

Cost needs to be entered using kWh in pounds (pence) for Day and Night for electricity as well as standing charge per day. And for gas unit rate and standing charge.

3 Keep recommended battery size but also allow user to enter size and price of battery

Example 1. Claude 3.7



Energy Usage & Cost Calculator

Usage Inputs

Electricity Usage (kWh per day)
10

Gas Usage (kWh equivalent per day)
5

Daytime Hours (for battery sizing)
16

Tariff Inputs

Electricity Costs (pence)

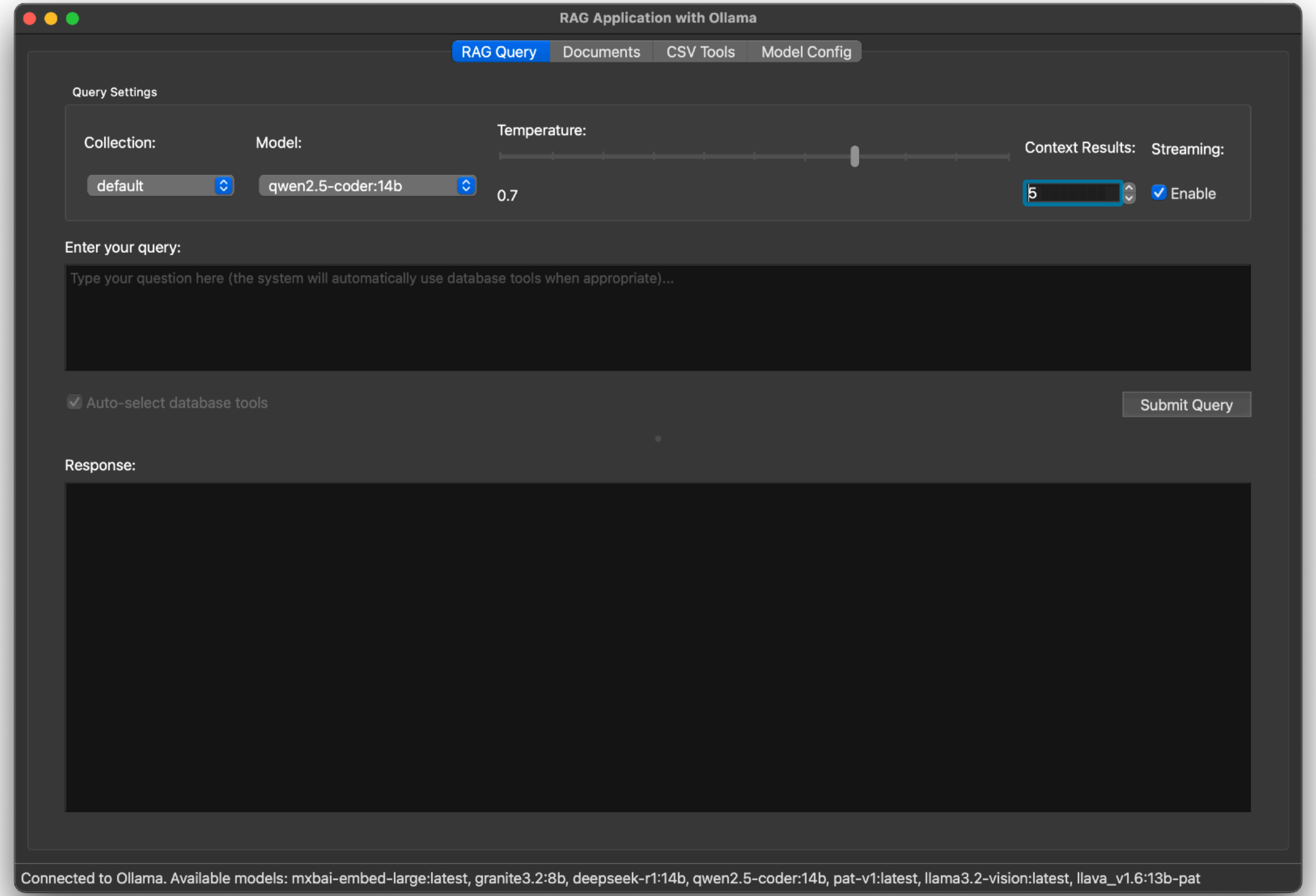
Day Rate (p/kWh)	Night Rate (p/kWh)
28	15

Standing Charge (p/day)
45

Example 2. Claude 3.7

Create a python app (PyQT) to do RAG using Pydantic API calling Ollama models. For the RAG request file(s), e.g. folder content of PDF and text files. Enable options such as chunk size. Convert PDF to markdown before processing. Use markitdown package. Also make use of tools for database content. In particular enable search through database (csv file), use the header in the csv file to identify key fields for the tool. For example, if the database has names, email addresses and office rooms create tools to search for names and email addresses, rooms, etc. Make sure app is easy to use and adaptable in terms of AI models used. Make a plan and then create the Python scripts.

Example 2. Claude 3.7



RAG Application with Ollama

Query Settings

Collection: default

Model: qwen2.5-coder:14b

Temperature: 0.7

Context Results: 5

Streaming: ✓ Enable

Enter your query:

Type your question here (the system will automatically use database tools when appropriate)...

✓ Auto-select database tools

Response:

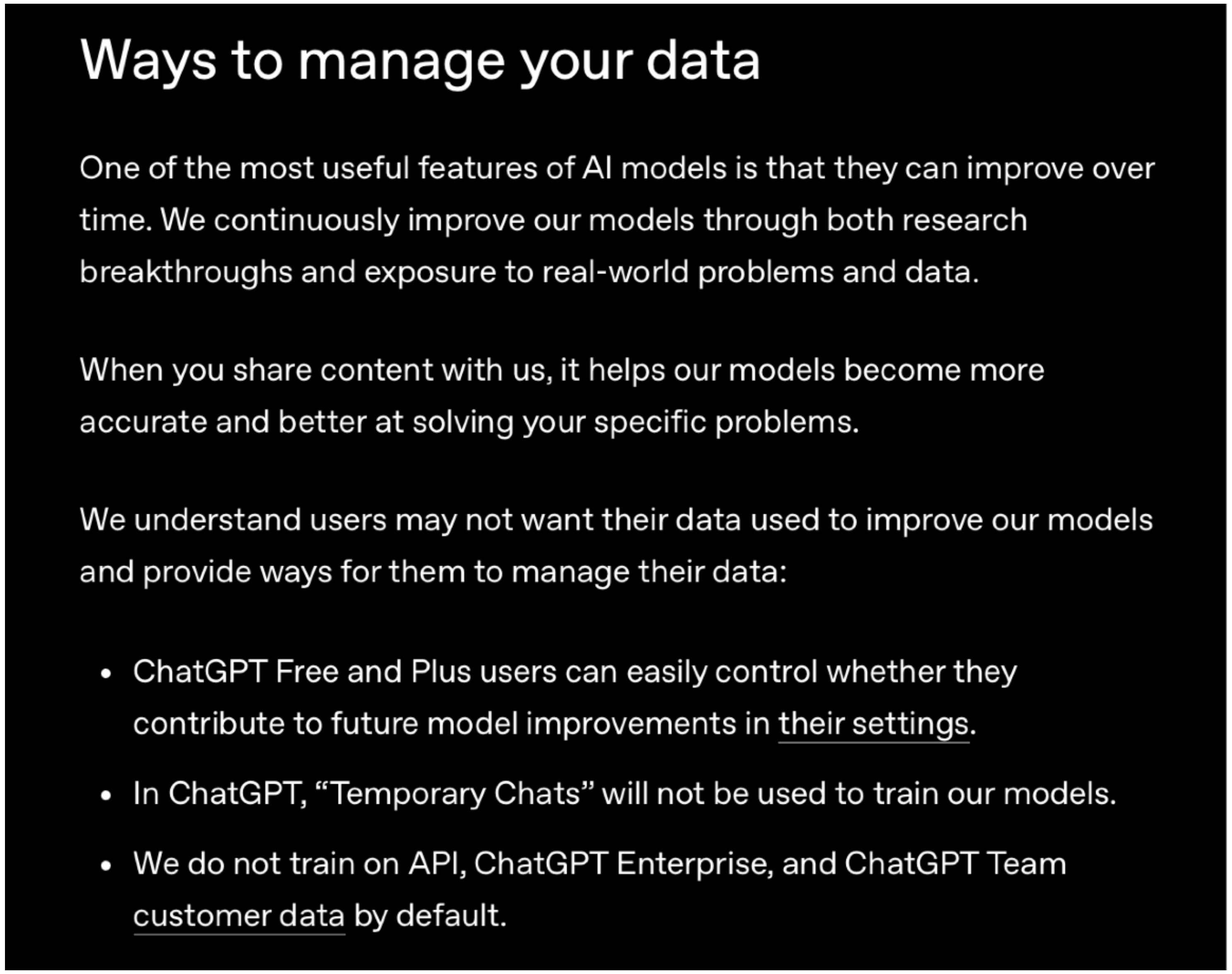
Connected to Ollama. Available models: mxbai-embed-large:latest, granite3.2:8b, deepseek-r1:14b, qwen2.5-coder:14b, pat-v1:latest, llama3.2-vision:latest, llava_v1.6:13b-pat

Using LLMs

- **Large Language Models** (LLMs) can be used in closed-sourced apps (or websites) such as **ChatGPT**, **Copilot**, **Claude**, **Gemini**, **Perplexity**, **Grok**, **DeepSeek**, **Mistral**, etc.

- **Disadvantages**
  - Input/output can potentially be used for training.
  - Limited control over the model.
  - Expensive (e.g. £20 or more per month for ChatGPT).

OpenAI Privacy



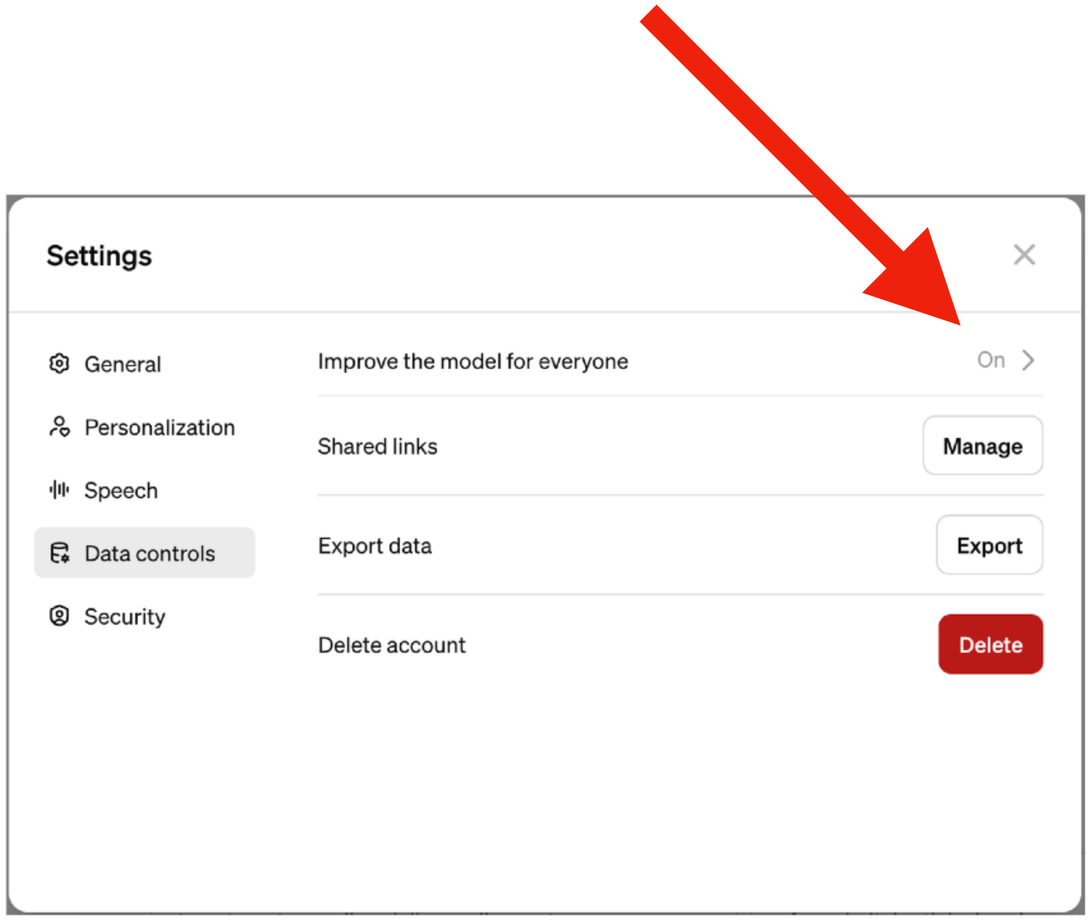
Ways to manage your data

One of the most useful features of AI models is that they can improve over time. We continuously improve our models through both research breakthroughs and exposure to real-world problems and data.

When you share content with us, it helps our models become more accurate and better at solving your specific problems.

We understand users may not want their data used to improve our models and provide ways for them to manage their data:

- ChatGPT Free and Plus users can easily control whether they contribute to future model improvements in their settings.
- In ChatGPT, "Temporary Chats" will not be used to train our models.
- We do not train on API, ChatGPT Enterprise, and ChatGPT Team customer data by default.



Settings

Improve the model for everyone — On

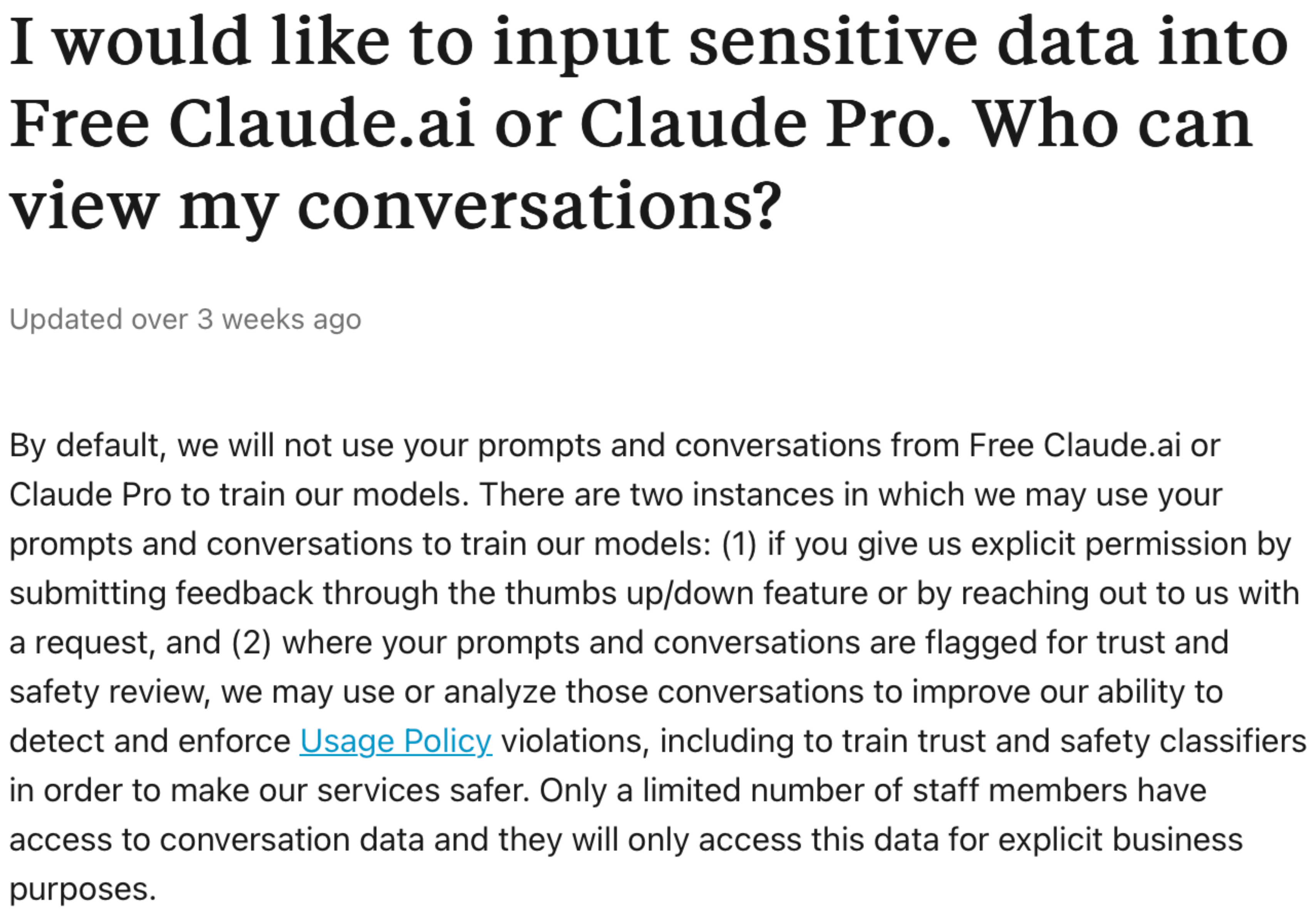
Shared links — Manage

Export data — Export

Delete account — Delete

https://openai.com/consumer-privacy/

Claude Privacy



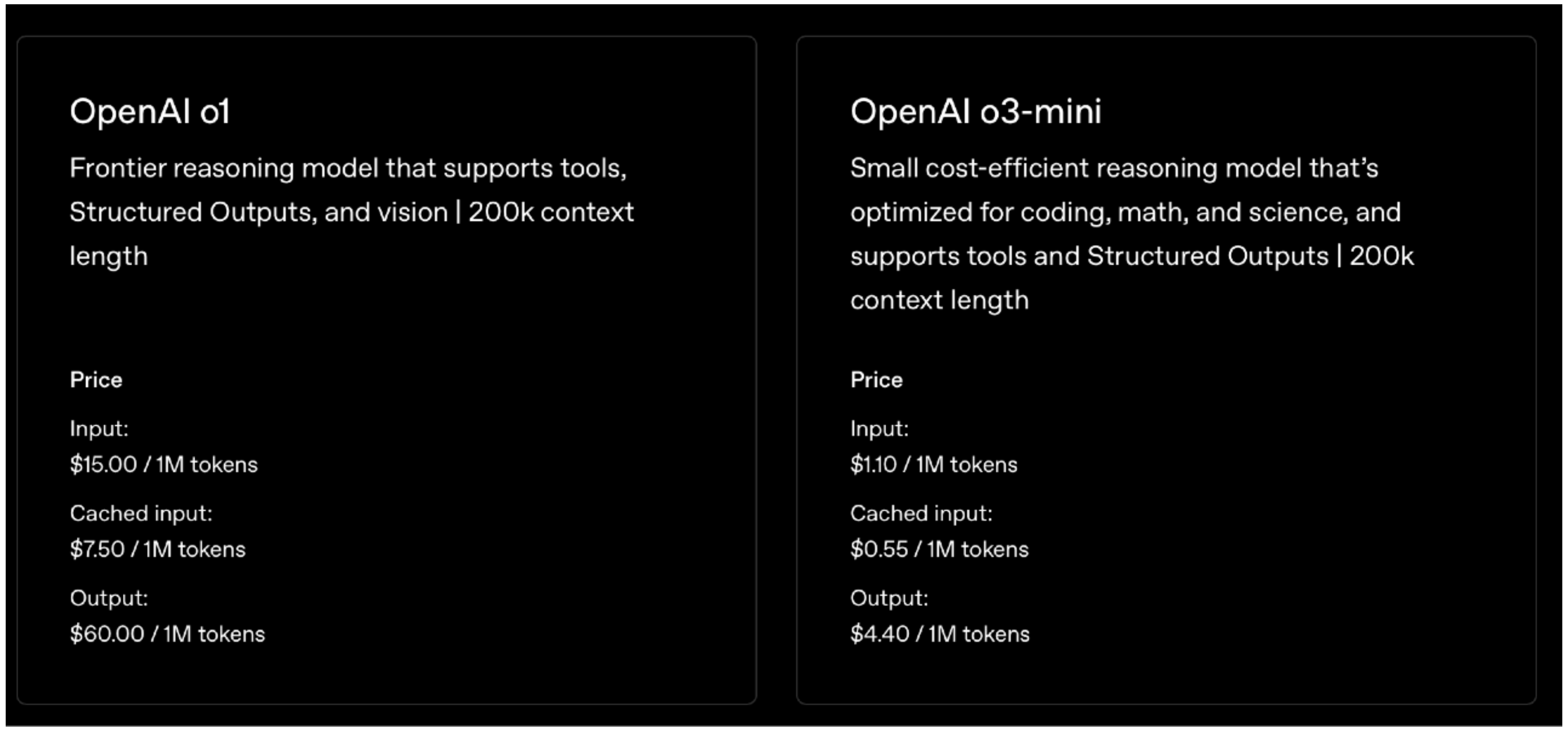
I would like to input sensitive data into Free Claude.ai or Claude Pro. Who can view my conversations?

Updated over 3 weeks ago

By default, we will not use your prompts and conversations from Free Claude.ai or Claude Pro to train our models. There are two instances in which we may use your prompts and conversations to train our models: (1) if you give us explicit permission by submitting feedback through the thumbs up/down feature or by reaching out to us with a request, and (2) where your prompts and conversations are flagged for trust and safety review, we may use or analyze those conversations to improve our ability to detect and enforce Usage Policy violations, including to train trust and safety classifiers in order to make our services safer. Only a limited number of staff members have access to conversation data and they will only access this data for explicit business purposes.

https://support.anthropic.com/

Pricing example



OpenAI o1

Frontier reasoning model that supports tools, Structured Outputs, and vision | 200k context length

Price

Input:
$15.00 / 1M tokens

Cached input:
$7.50 / 1M tokens

Output:
$60.00 / 1M tokens

OpenAI o3-mini

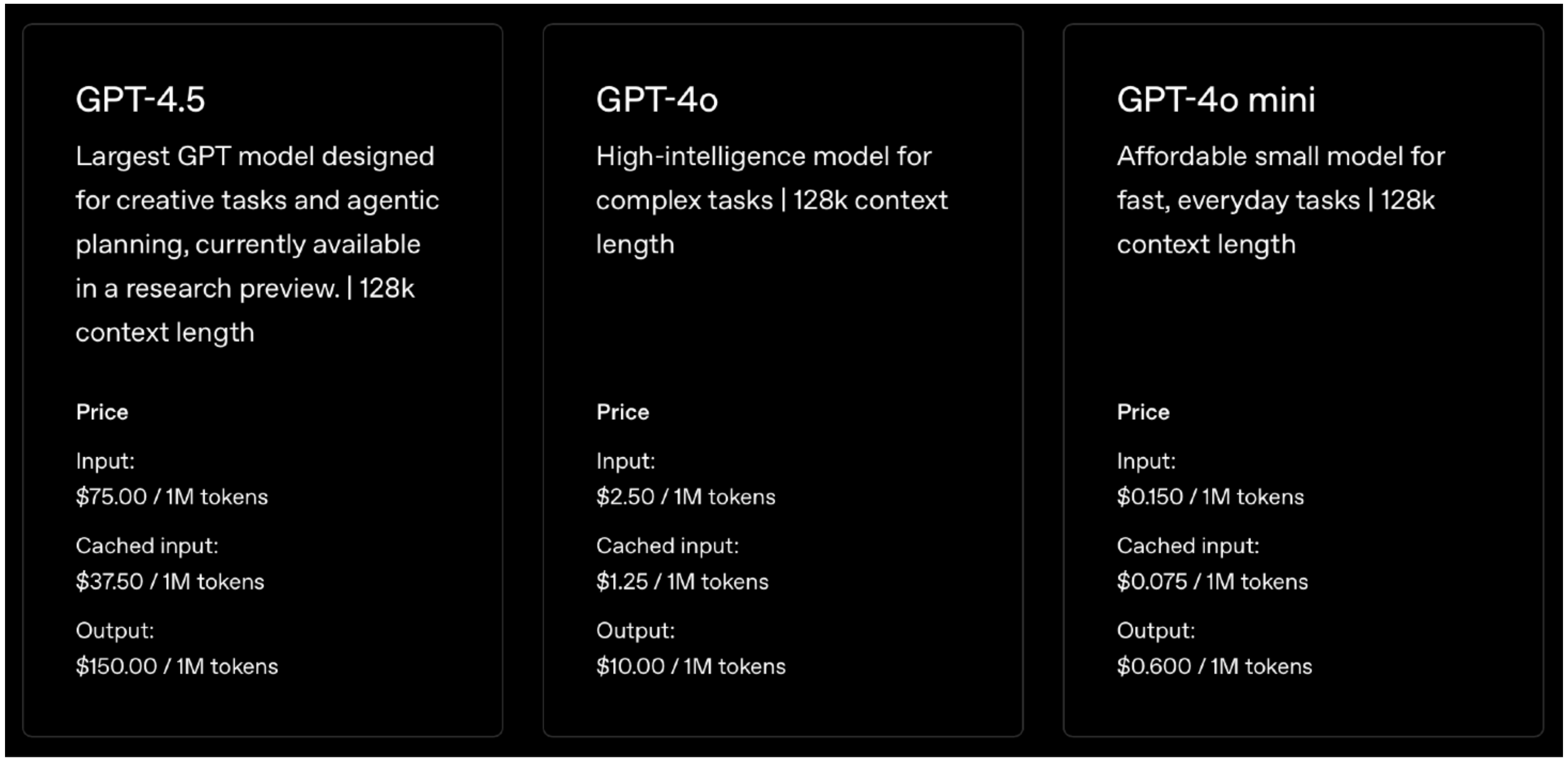
Small cost-efficient reasoning model that's optimized for coding, math, and science, and supports tools and Structured Outputs | 200k context length

Price

Input:
$1.10 / 1M tokens

Cached input:
$0.55 / 1M tokens

Output:
$4.40 / 1M tokens



GPT-4.5

Largest GPT model designed for creative tasks and agentic planning, currently available in a research preview. | 128k context length

Price

Input:
$75.00 / 1M tokens

Cached input:
$37.50 / 1M tokens

Output:
$150.00 / 1M tokens

GPT-4o

High-intelligence model for complex tasks | 128k context length

Price

Input:
$2.50 / 1M tokens

Cached input:
$1.25 / 1M tokens

Output:
$10.00 / 1M tokens

GPT-4o mini

Affordable small model for fast, everyday tasks | 128k context length

Price

Input:
$0.150 / 1M tokens

Cached input:
$0.075 / 1M tokens

Output:
$0.600 / 1M tokens

Running LLMs locally

Requirements for LLM inference

Powerful recent **PC/Mac** with lots of memory (>= 16 Gb).

PC: Dedicated GPU (NVIDIA RTX series, e.g. A4000).

RTX 3060
8 Gb: £270

RTX A4000
16 Gb: £900

Mac: Apple Silicon (M1 or later).

8 GPU cores

10-40 GPU cores

Mac Studio
76 GPU cores

How much memory needed?

LLMs memory requirements depend on the number of **parameters** in the model and number of **bytes** used for each parameter.

Llama3.3: 70B, Llama3.1: 8B, 70B, 405B, Phi4: 14B, QwQ: 32B

Weights can be quantized (reduce precision) to for example 6-bits or 4-bits reduce memory requirements.

e.g. in ggml **Q4_K**: 4 bits per weight. With Q4_K, a 7B parameter model requires ~4GB.

https://mlabonne.github.io/blog/posts/Introduction_to_Weight_Quantization.html

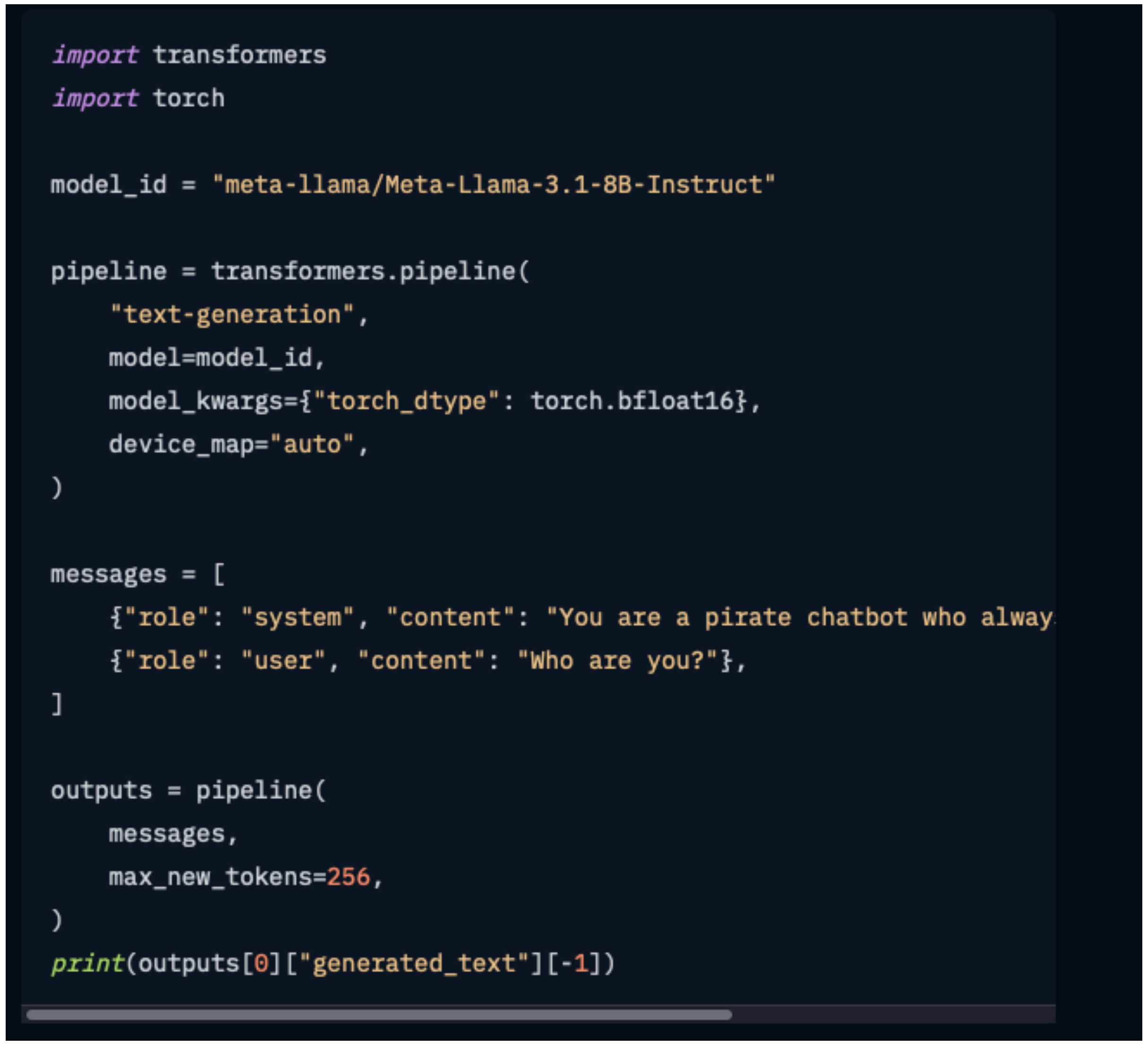
Calculator (memory, costs): https://llm-dev-tools.streamlit.app/

LLMs in Python

Hugging Face: https://huggingface.co/

Using **Python** and the **transformers** and **torch** libraries.

https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct



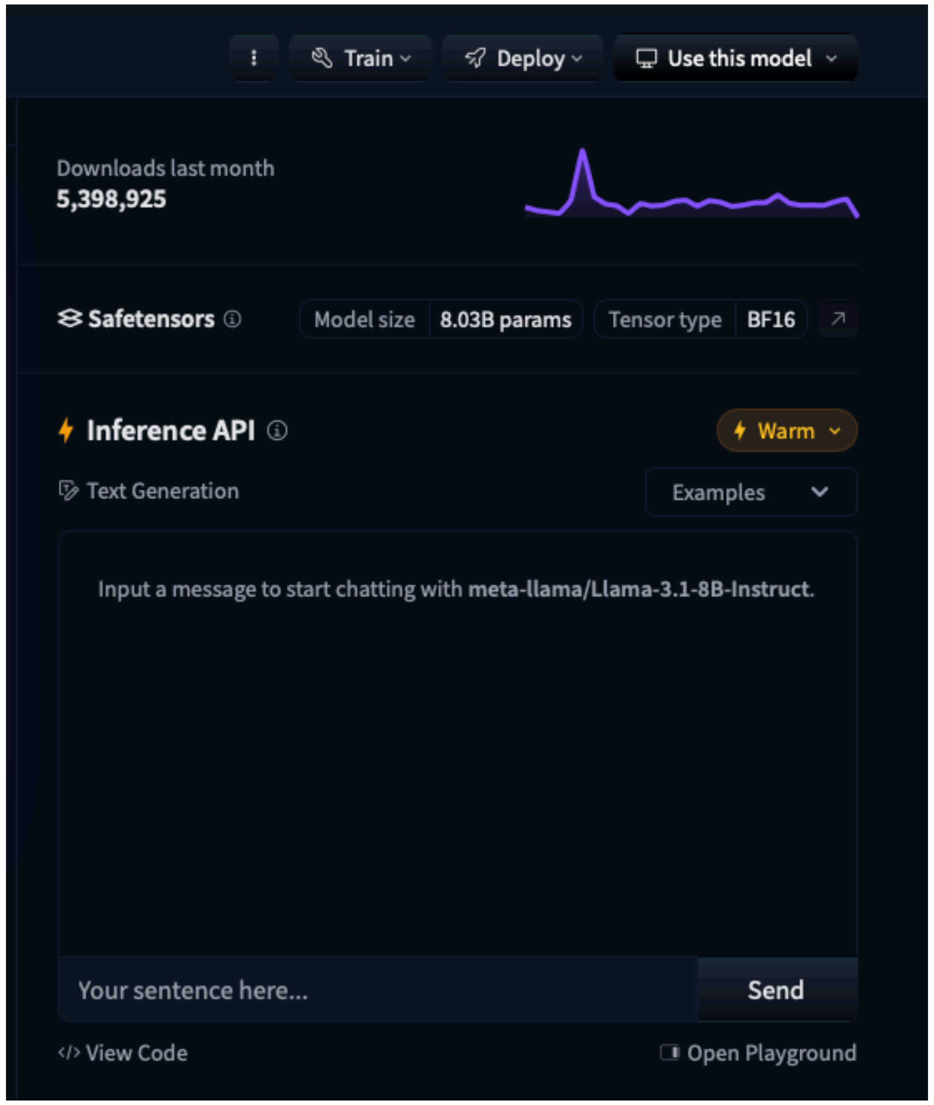
```python
import transformers
import torch

model_id = "meta-llama/Meta-Llama-3.1-8B-Instruct"

pipeline = transformers.pipeline(
    "text-generation",
    model=model_id,
    model_kwargs={"torch_dtype": torch.bfloat16},
    device_map="auto",
)

messages = [
    {"role": "system", "content": "You are a pirate chatbot who alway
    {"role": "user", "content": "Who are you?"},
]

outputs = pipeline(
    messages,
    max_new_tokens=256,
)
print(outputs[0]["generated_text"][-1])
```



Downloads last month
5,398,925

Safetensors | Model size 8.03B params | Tensor type BF16

Inference API

Input a message to start chatting with meta-llama/Llama-3.1-8B-Instruct.

llama.cpp

llama.cpp (https://github.com/ggerganov/llama.cpp) written by Georgi Gerganov enables LLM inference with minimal setup and state-of-the-art performance on a wide range of hardware - locally and in the **cloud**.

It uses **ggml** (tensor library for machine learning).

llama.cpp is written in plain C/C++. Supports a wide range of backends: e.g. CPU, Metal, CUDA (requires **CUDA toolkit**), Vulkan (requires **ROCm**).

Supports many models: e.g., LLaMA, Mistral, BERT, Deepseek, Qwen, Phi, GPT-2, LLaVA, Qwen2-VL

llama-cli, llama-server

MLX

MLX: array framework for machine learning on **Apple Silicon** (some support for Linux and Windows)

https://github.com/ml-explore/mlx

Closely follows **NumPy**.

https://github.com/ml-explore/mlx-examples

MLX-LM (run models, serve LLMs through HTTP, fine-tuning, merging model, etc.). Python mlx-lm module.

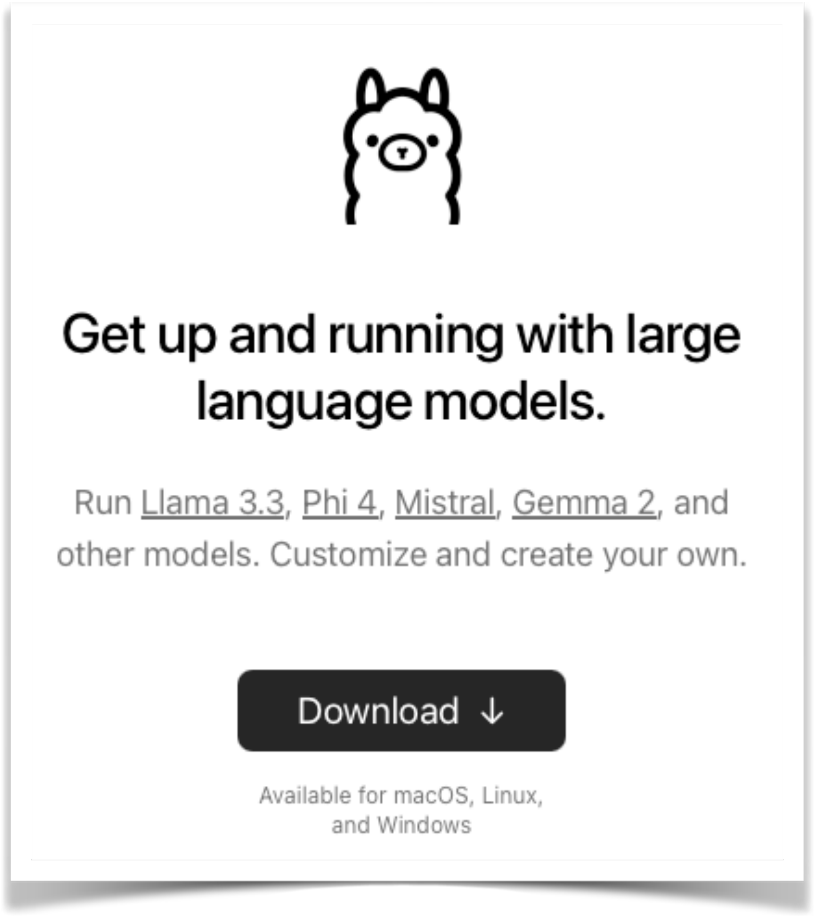
Models available on hugging-face: https://huggingface.co/mlx-community

MLX-VLM Vision Language Models (VLMs)

(e.g. LLaVA, Qwen2-VL, Phi3-Vision).

Ollama

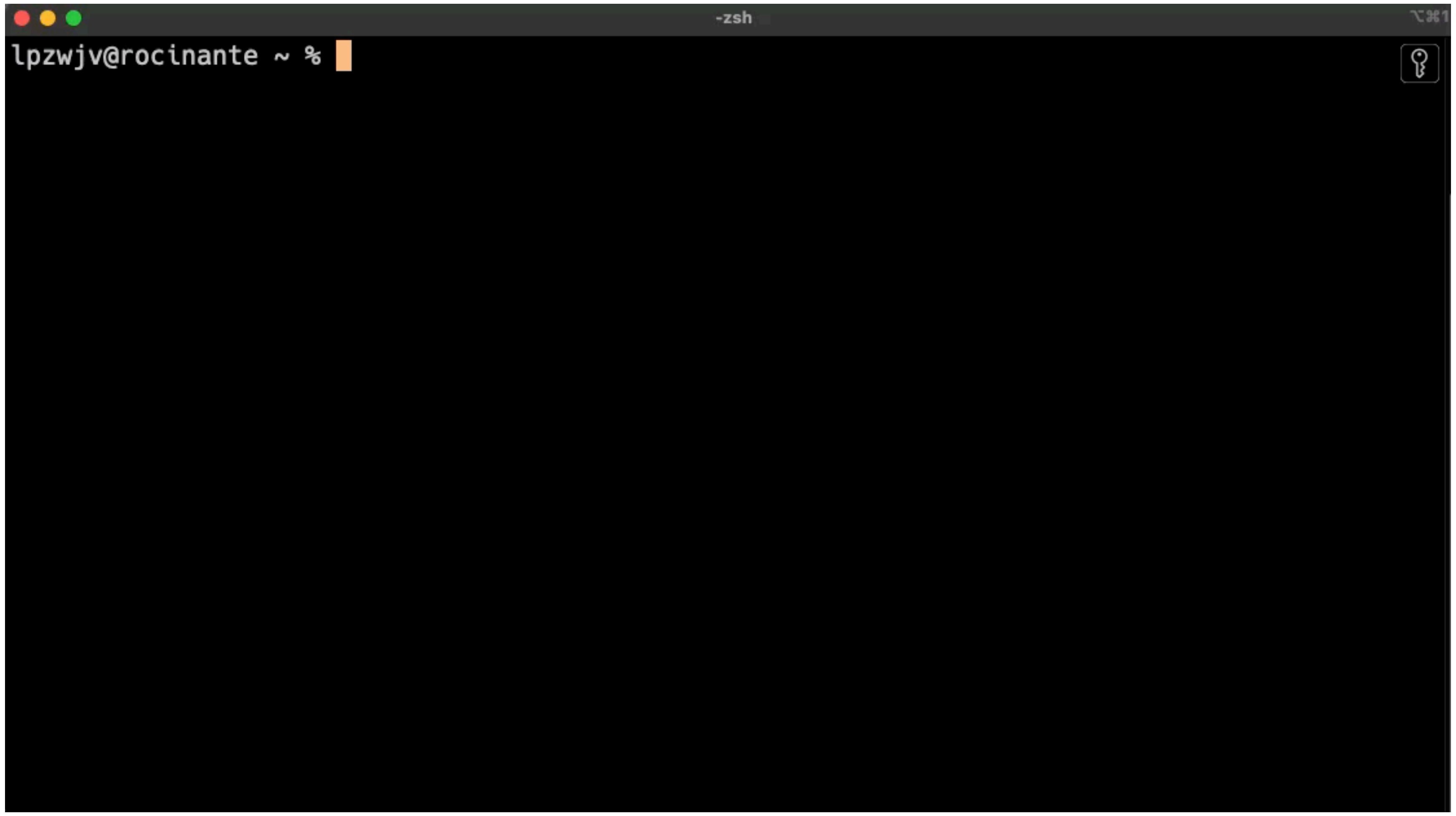
- https://ollama.com/ (macOS, Linux, Windows)
- On macOS you can install it through brew (https://brew.sh/).
- Next, pull model (e.g. llama3.2, Phi-4, gemma 2, etc.)



Get up and running with large language models.

Run Llama 3.3, Phi 4, Mistral, Gemma 2, and other models. Customize and create your own.

Available for macOS, Linux, and Windows



```
lpzwjv@rocinante ~ %
```

LM Studio

https://lmstudio.ai/

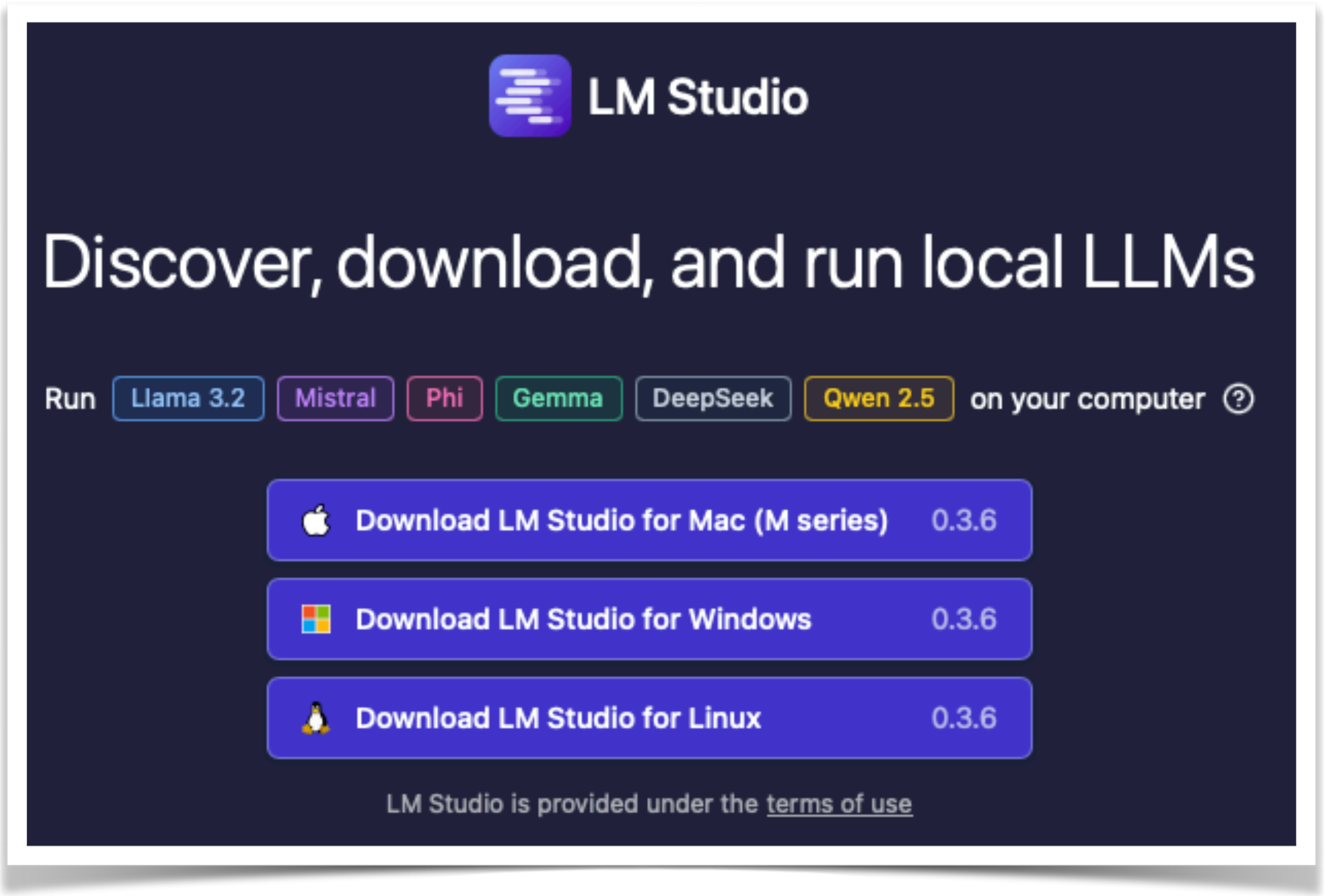
Windows, Mac, Linux.

Run LLMs locally.

llama.cpp and MLX support.

Lots of features but

complex interface.



LM Studio

Discover, download, and run local LLMs

Run Llama 3.2 Mistral Phi Gemma DeepSeek Qwen 2.5 on your computer

Download LM Studio for Mac (M series) 0.3.6
Download LM Studio for Windows 0.3.6
Download LM Studio for Linux 0.3.6

LM Studio is provided under the terms of use

Open WebUI

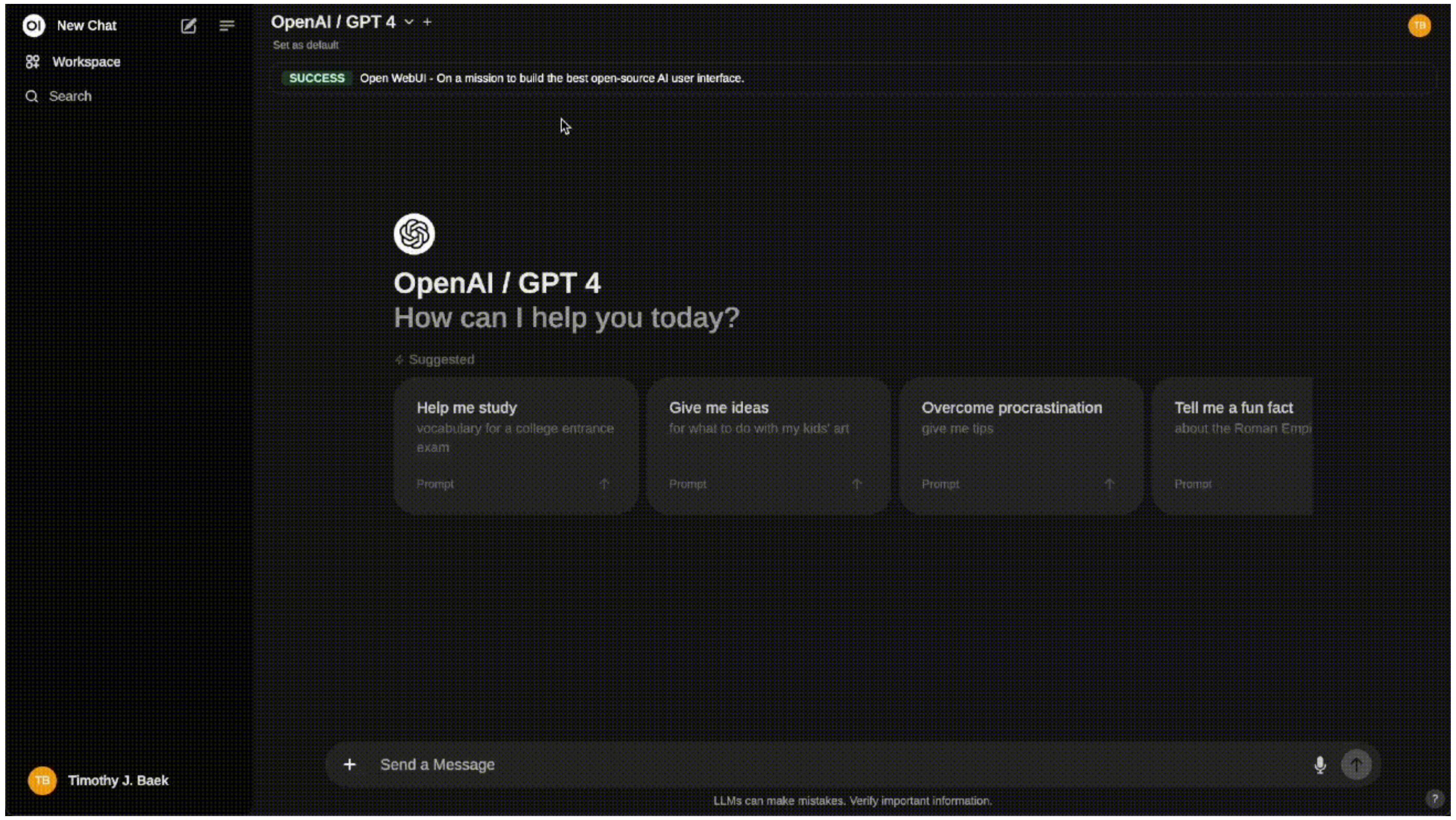
https://openwebui.com/

Nice chat interface. Web browsing.
Local LLMs. Ollama support.

RAG support. Installed through Docker.

Ignite curiosity wherever you are

Open WebUI is an extensible, self-hosted AI interface that adapts to y workflow, all while operating entirely offline.



SUCCESS Open WebUI - On a mission to build the best open-source AI user interface.

OpenAI / GPT 4
How can I help you today?

LLMs can make mistakes. Verify important information.

Linux/macOS:
Use OrbStack rather than Docker.

LibreChat

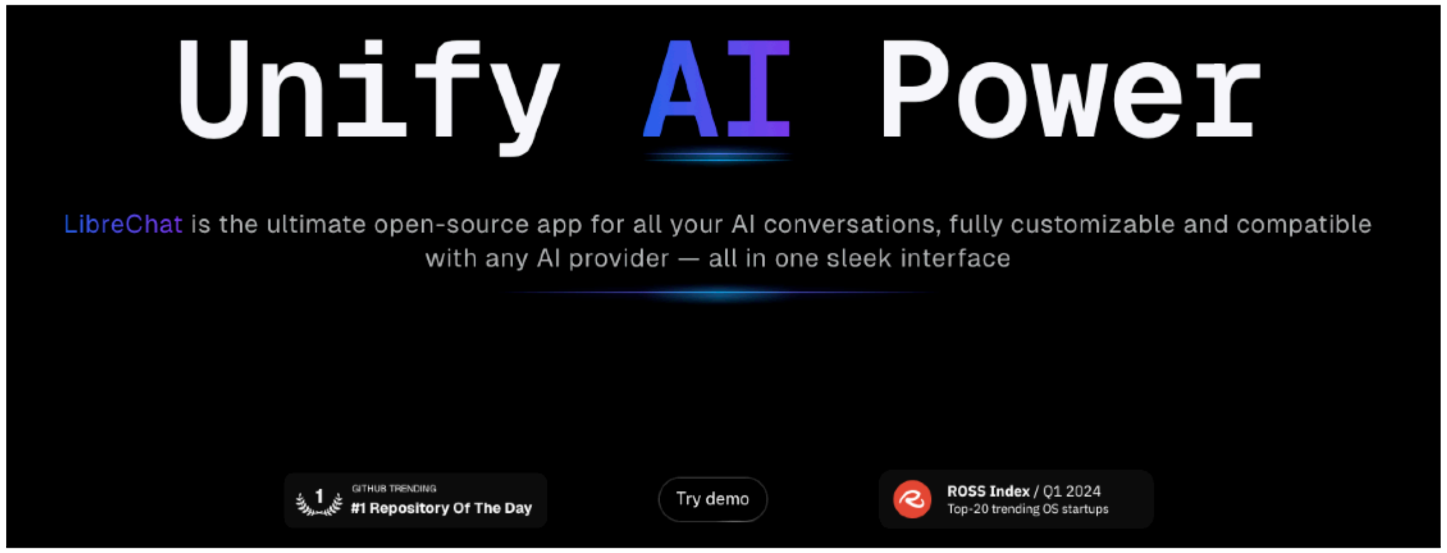
https://www.librechat.ai/

http://localhost:3080/login

Nice chat interface.

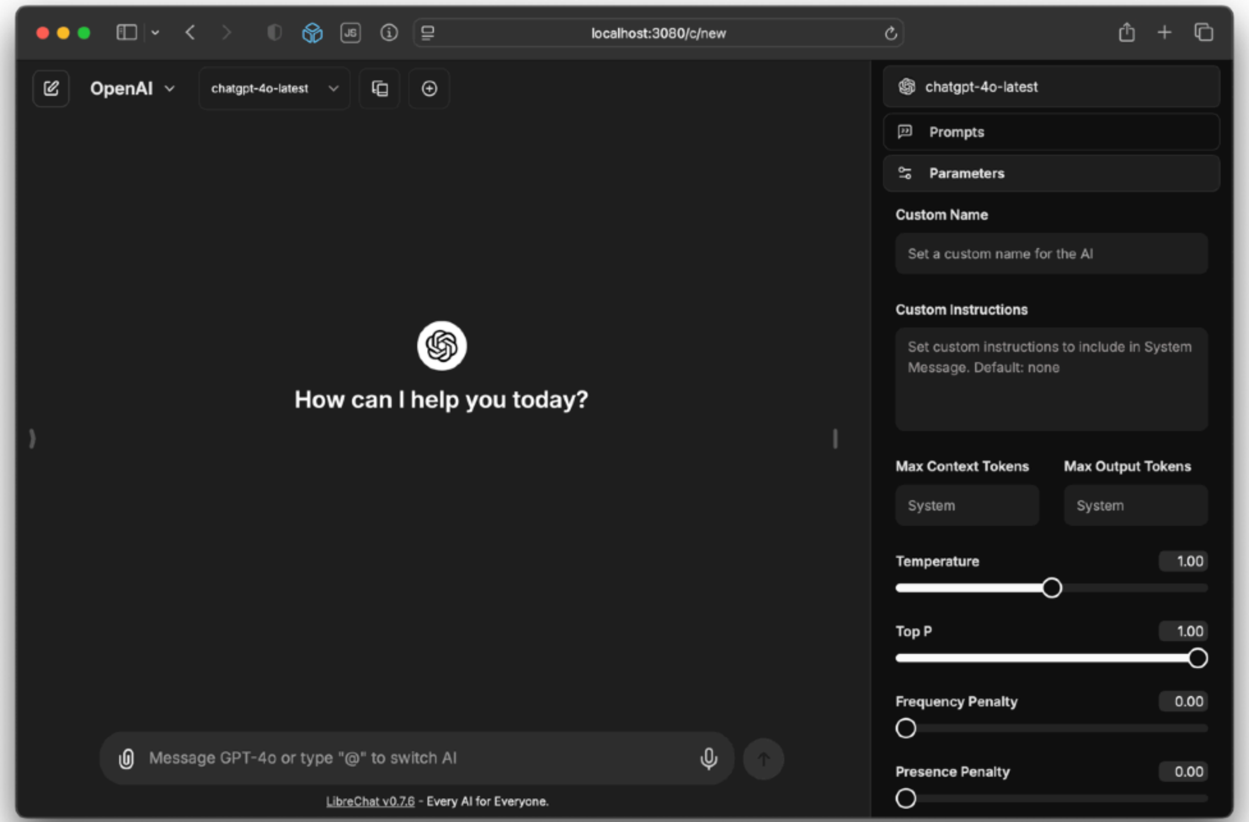
LLMs through API access.

Installed through Docker (or OrbStack).



Unify AI Power

LibreChat is the ultimate open-source app for all your AI conversations, fully customizable and compatible with any AI provider — all in one sleek interface



How can I help you today?

Message GPT-4o or type "@" to switch AI

LibreChat v0.7.6 - Every AI for Everyone.

Custom Name: Set a custom name for the AI

Custom Instructions: Set custom instructions to include in System Message. Default: none

Max Context Tokens: System

Max Output Tokens: System

Temperature: 1.00

Top P: 1.00

Frequency Penalty: 0.00

Presence Penalty: 0.00

Msty

https://msty.app/

macOS, Windows, Linux.

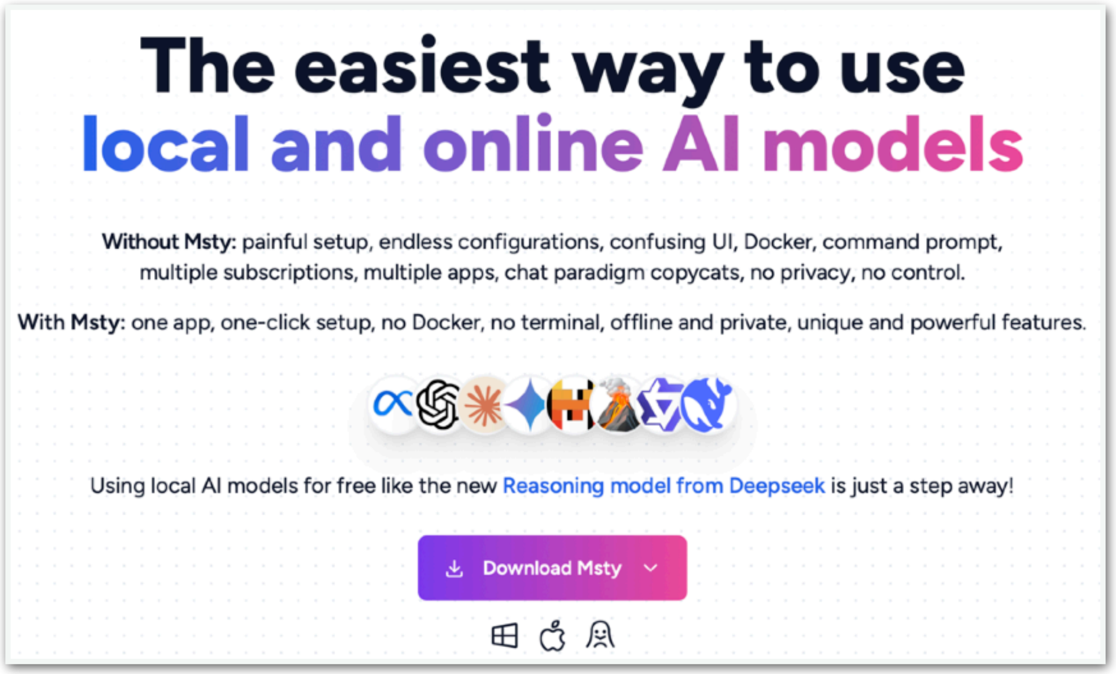
Local models, no need to install Ollama.

AI models through API.

Azure support (paid version).

Web search support.

RAG.

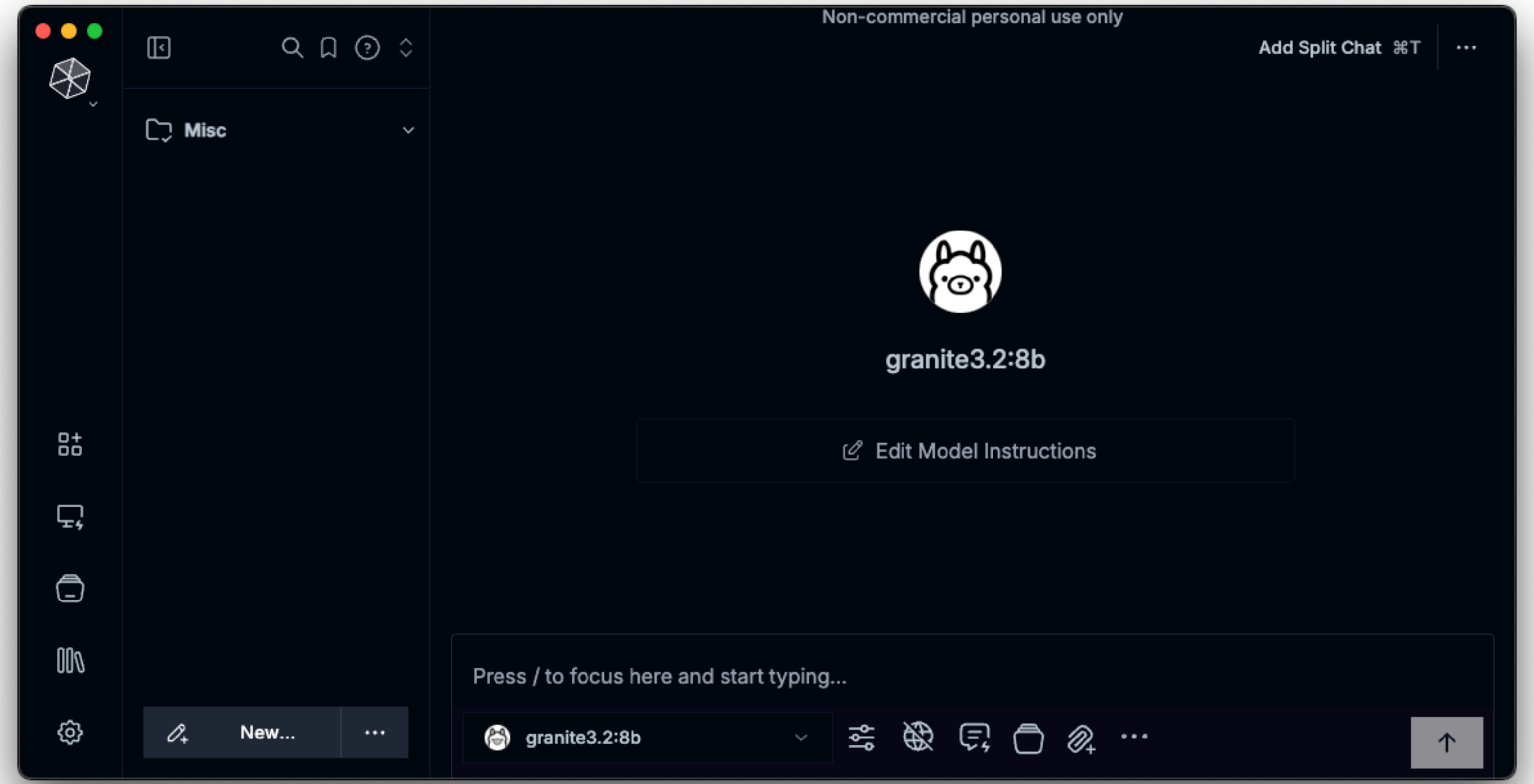


The easiest way to use local and online AI models

Without Msty: painful setup, endless configurations, confusing UI, Docker, command prompt, multiple subscriptions, multiple apps, chat paradigm copycats, no privacy, no control.

With Msty: one app, one-click setup, no Docker, no terminal, offline and private, unique and powerful features.

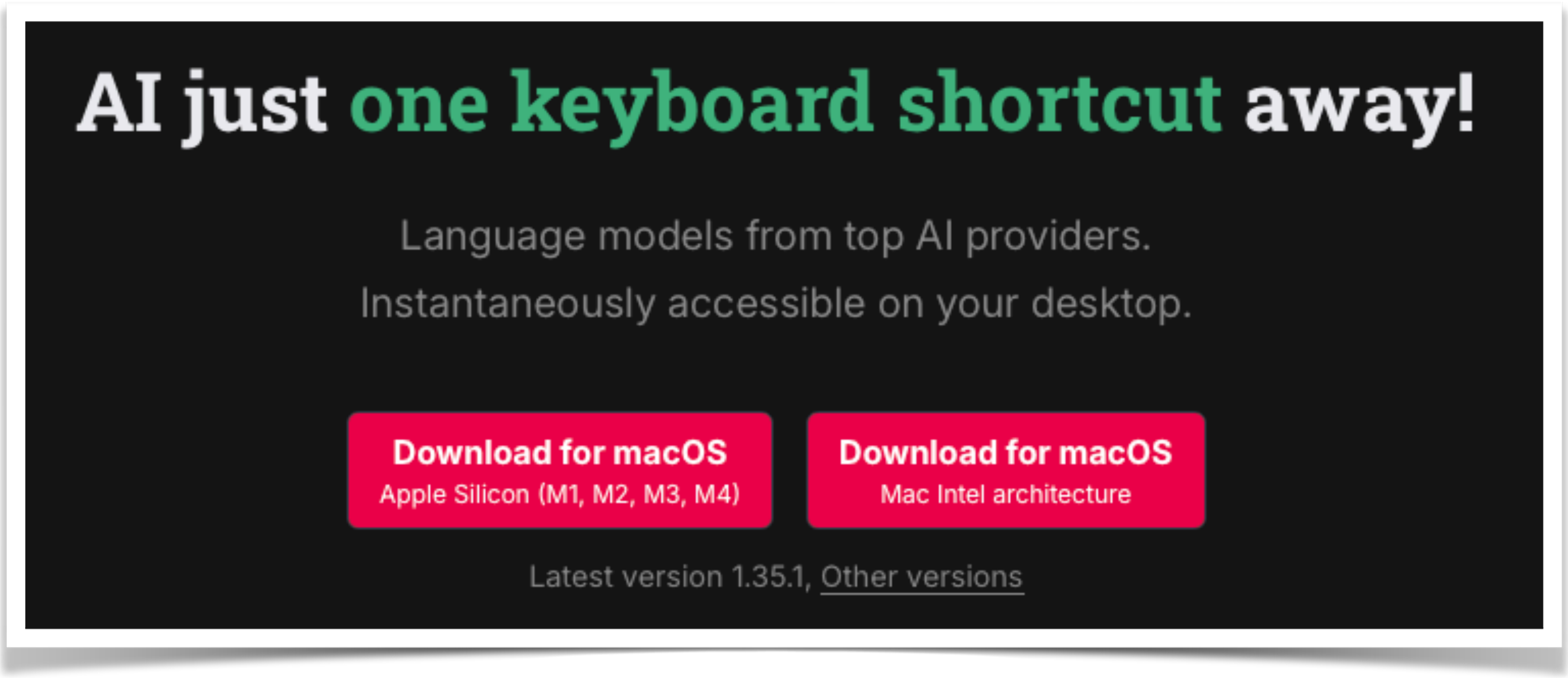
Using local AI models for free like the new Reasoning model from Deepseek is just a step away!



Witsy

https://witsyai.com/

macOS, Windows, Linux.


AI just one keyboard shortcut away!

Language models from top AI providers.
Instantaneously accessible on your desktop.

Latest version 1.35.1, Other versions

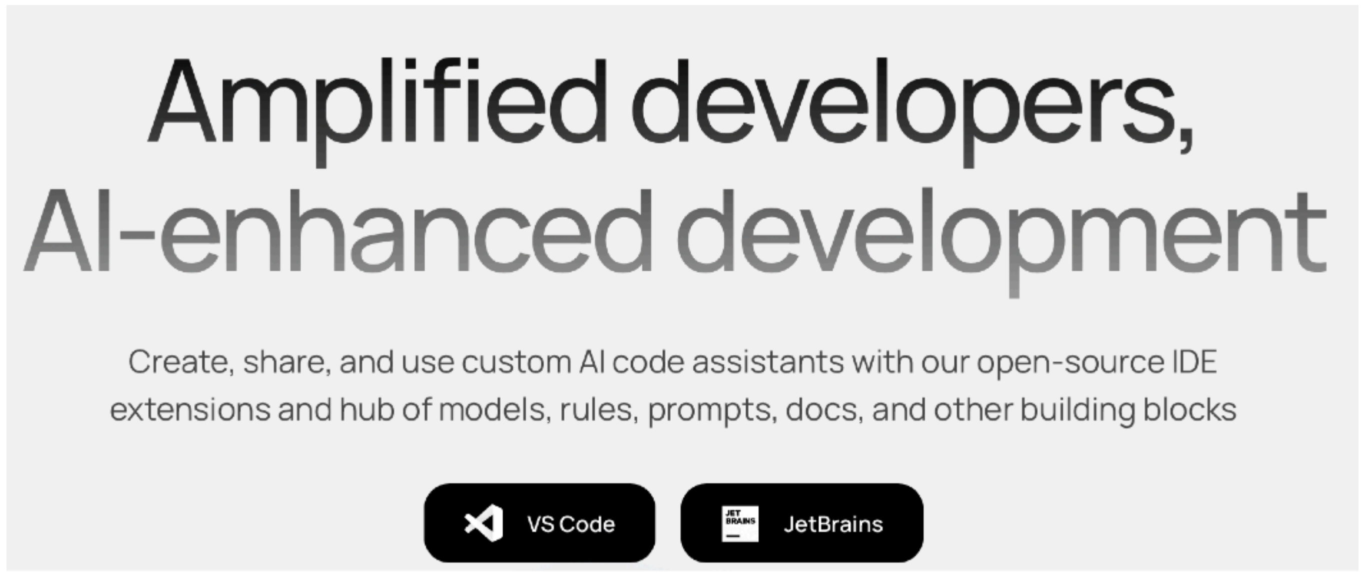
Support for Ollama running locally or remotely. Furthermore, you can access closed-sourced models by using an API keys.

Web search and RAG support.

Visual Studio Code

LLMs to support coding in VS Code

- [Copilot]
- [Continue]
- [llama-vscode]


Amplified developers, AI-enhanced development

Create, share, and use custom AI code assistants with our open-source IDE extensions and hub of models, rules, prompts, docs, and other building blocks

```
llama-server \
    -hf ggml-org/Qwen2.5-Coder-7B-Q8_0-GGUF \
    --port 8012 -ngl 99 -fa -ub 1024 -b 1024 \
    --ctx-size 0 --cache-reuse 256
```

LLMs through Microsoft Azure

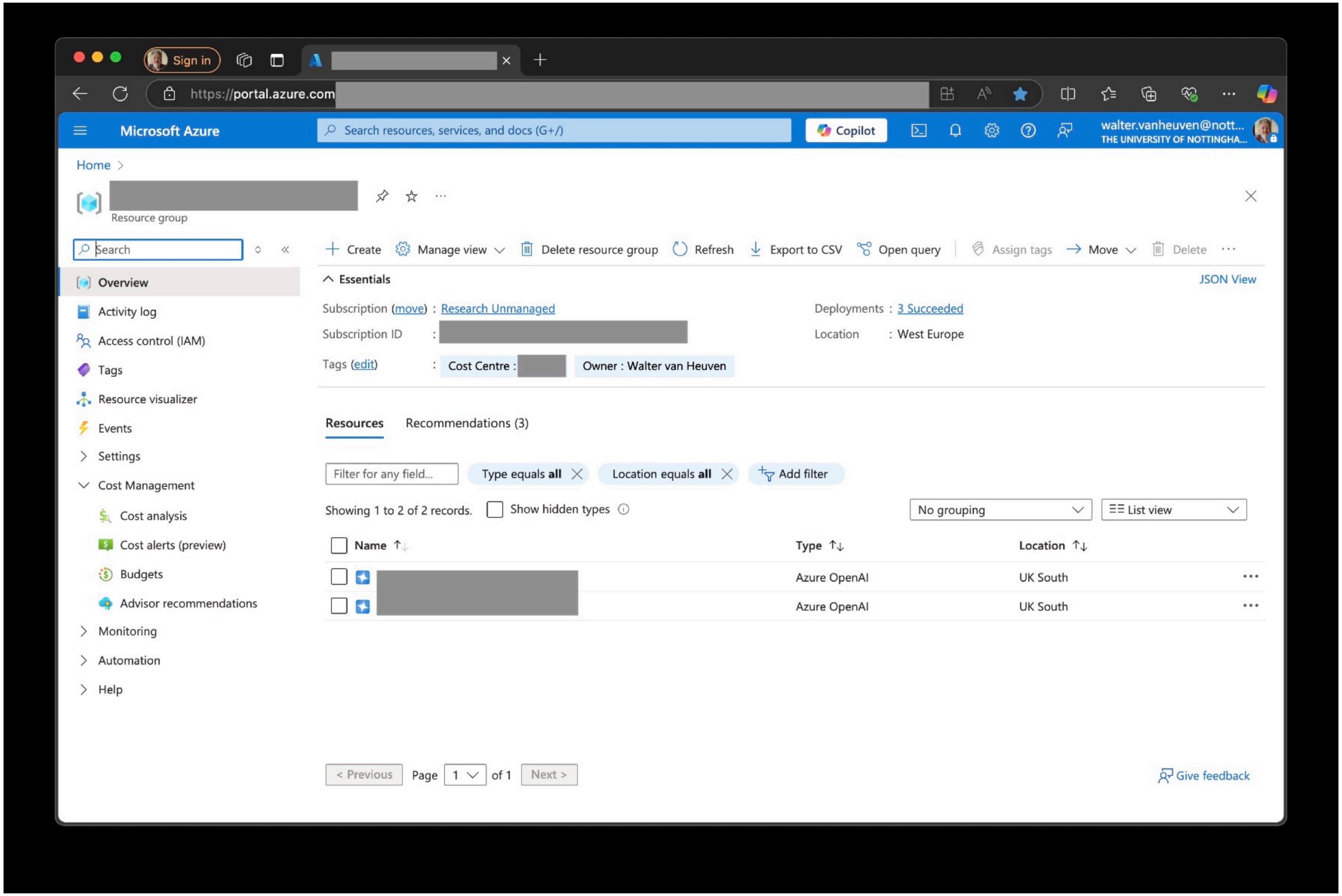
Advantage

- University approved platform.
- Data not used for training.
- Can use RTSG to pay for OpenAI API usage.

Issues

- Complex to setup.
- Does not use a pre-payment model.
- Need to monitor usage/costs to avoid surprises.

Microsoft Azure Portal



Sign in [redacted]

https://portal.azure.com[redacted]

Microsoft Azure

Search resources, services, and docs (G+/)

Copilot

walter.vanheuven@nott...
THE UNIVERSITY OF NOTTINGHA...

Home >

[redacted]
Resource group

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Events
- Settings
- Cost Management
  - Cost analysis
  - Cost alerts (preview)
  - Budgets
  - Advisor recommendations
- Monitoring
- Automation
- Help

+ Create | Manage view | Delete resource group | Refresh | Export to CSV | Open query | Assign tags | Move | Delete

Essentials — JSON View

Subscription (move) : Research Unmanaged
Subscription ID : [redacted]
Tags (edit) : Cost Centre : [redacted] Owner : Walter van Heuven
Deployments : 3 Succeeded
Location : West Europe

Resources | Recommendations (3)

Filter for any field... | Type equals **all** | Location equals **all** | Add filter

Showing 1 to 2 of 2 records. Show hidden types

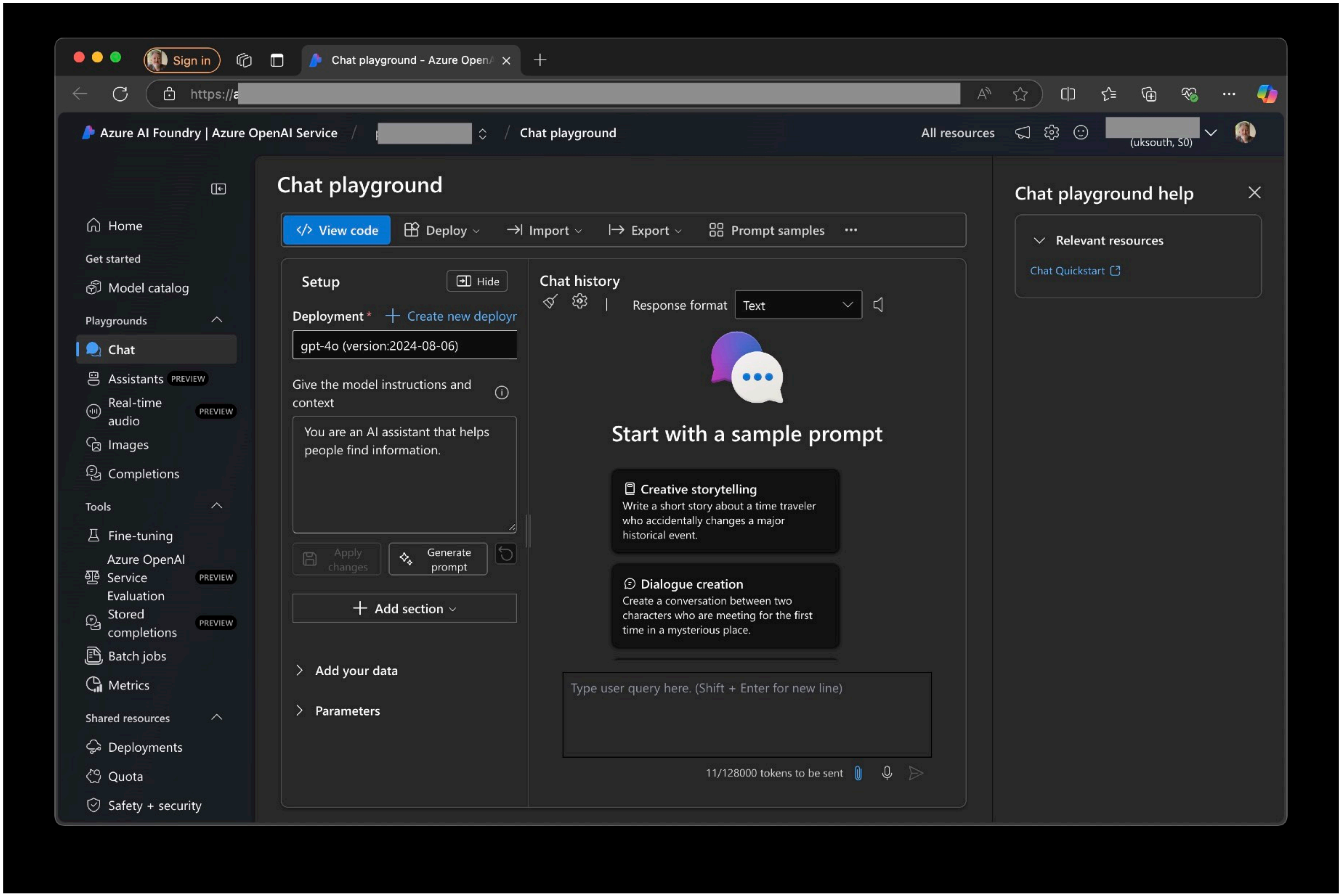
No grouping | List view

Name ↑↓	Type ↑↓	Location ↑↓
[redacted]	Azure OpenAI	UK South
	Azure OpenAI	UK South

< Previous Page 1 of 1 Next >

Give feedback

Azure AI Foundry



Chat playground

Deployment * : gpt-4o (version:2024-08-06)

Give the model instructions and context

You are an AI assistant that helps people find information.

Response format: Text

Start with a sample prompt

Creative storytelling
Write a short story about a time traveler who accidentally changes a major historical event.

Dialogue creation
Create a conversation between two characters who are meeting for the first time in a mysterious place.

Type user query here. (Shift + Enter for new line)

11/128000 tokens to be sent

Useful Resources

Andrej Karpathy's YouTube videos

- How I use LLMs

- Deep Dive into LLMs like ChatGPT

- Let's build GPT: from scratch, in code, spelled out

AI Explained videos

https://www.youtube.com/@aiexplained-official

Thank You