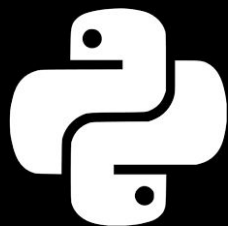


# LLMs for me



**Local LLM Development**

[llmsfor.me](https://llmsfor.me)



Myles Harrison,  
AI Consultant & Trainer



**January 28th,  
2025**

*NLP from scratch* 

# Agenda

**01**

**Introduction & Motivation**

**02**

**Ollama**

**03**

**GUI Applications for Local LLMs**

**04**

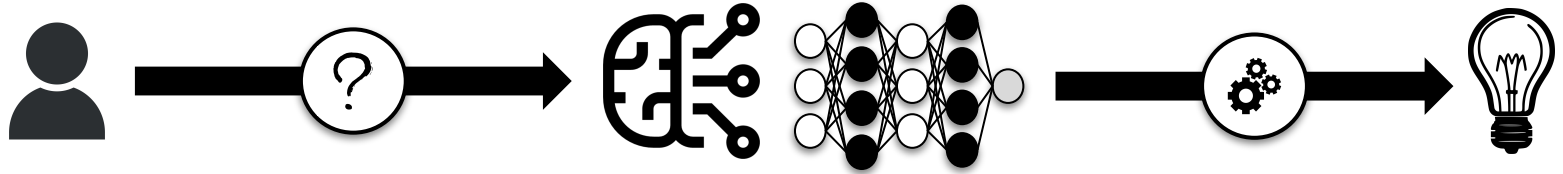
**Conclusion**



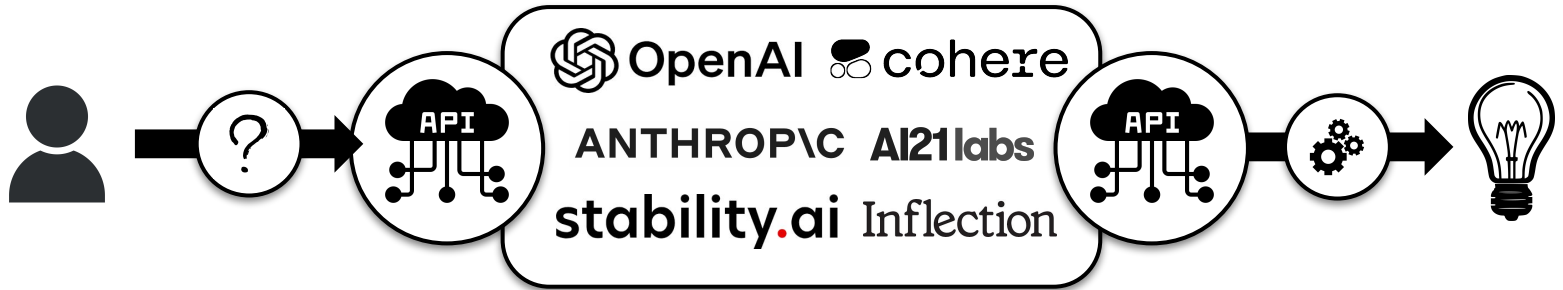
**MOTIVATION**

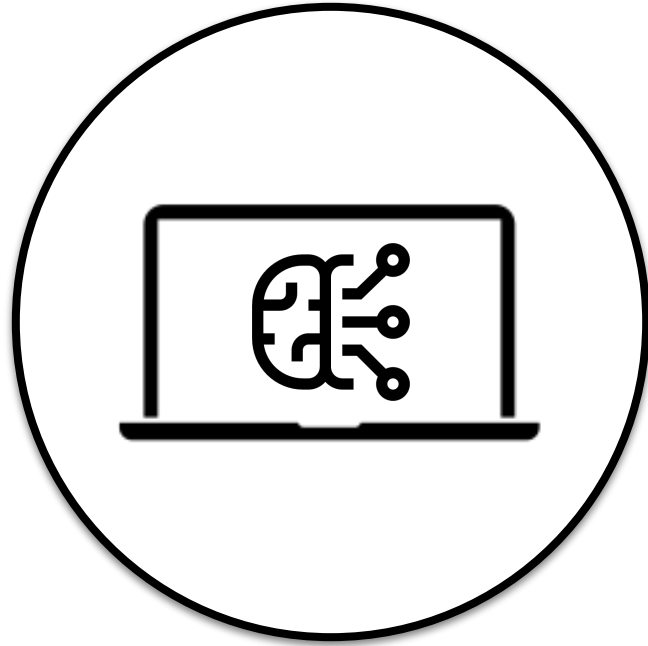
# The "Double Blackbox"

BEFORE

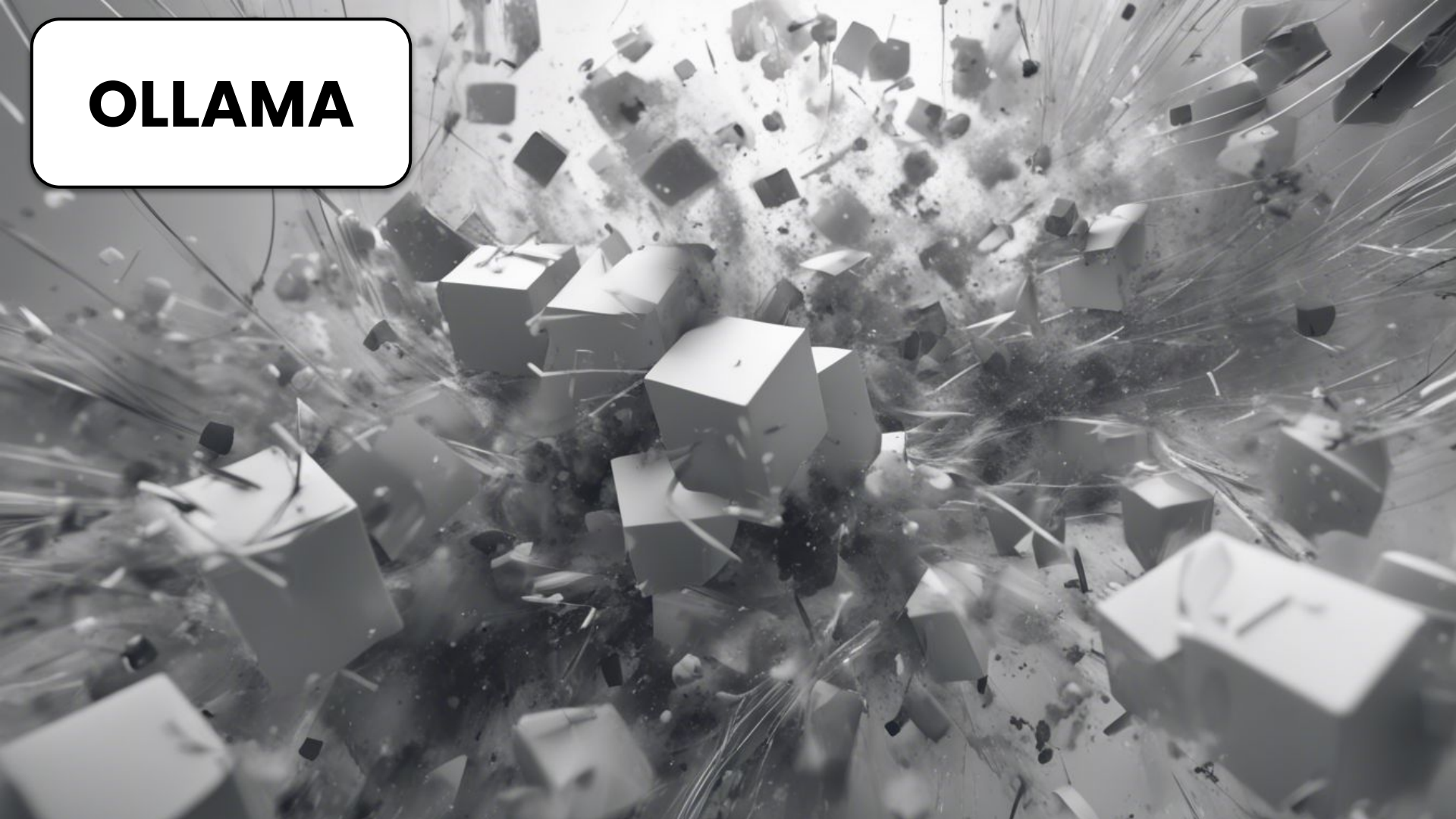


AFTER





**OLLAMA**



# What's Ollama?

Run LLMs locally, with models such as LLaMa 2, Mistral 7B, Phi, Orca, and many more!

Local web server with REST API, available in a Docker container

[ollama.com/blog/ollama-is-now-available-as-an-official-docker-image](https://ollama.com/blog/ollama-is-now-available-as-an-official-docker-image)

Libraries in Python, Java, C, Rust, Ruby, Langchain, etc.

Multimodal support (Ollama Vision) with LLaVa added for image interrogation, Object detection, text recognition, etc. added 2024/02/02

<https://ollama.ai/blog/vision-models>



## Get up and running with large language models, locally.

Run Llama 2, Code Llama, and other models.  
Customize and create your own.

Download ↓

Available for macOS & Linux  
Windows coming soon

*NLP from scratch* 



# How does Ollama run LLMs locally?

The logo for LLaMA C++ is displayed on a dark background. The text 'LLaMA' is in white, and the 'C++' is in orange. The 'C' is stylized with a flame-like shape above it.

license [MIT](#)

[Roadmap](#) / [Project status](#) / [Manifesto](#) / [ggml](#)

Inference of Meta's [LLaMA](#) model (and others) in pure C/C++



# llamafile: bringing LLMs to the people, and to your own computer

Dec. 14, 2023 | Stephen Hood

## Llamafile

One LLM, one executable file, 6 platforms.

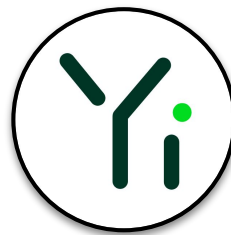
Combines llama.cpp with Cosmopolitan Libc

<https://github.com/Mozilla-Ocho/llamafile>



Introducing the latest Mozilla Innovation Project [llamafile](#), an open source initiative that collapses all the complexity of a full-stack LLM chatbot down to **a single file that runs on six operating systems**. Read on as we share a bit about why we created llamafile, how we did it, and the impact we hope it will have on open source AI.

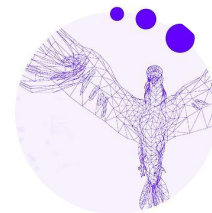
# What models can I run?



Qwen

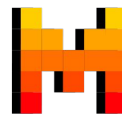


Gemma




stability.ai

and more...

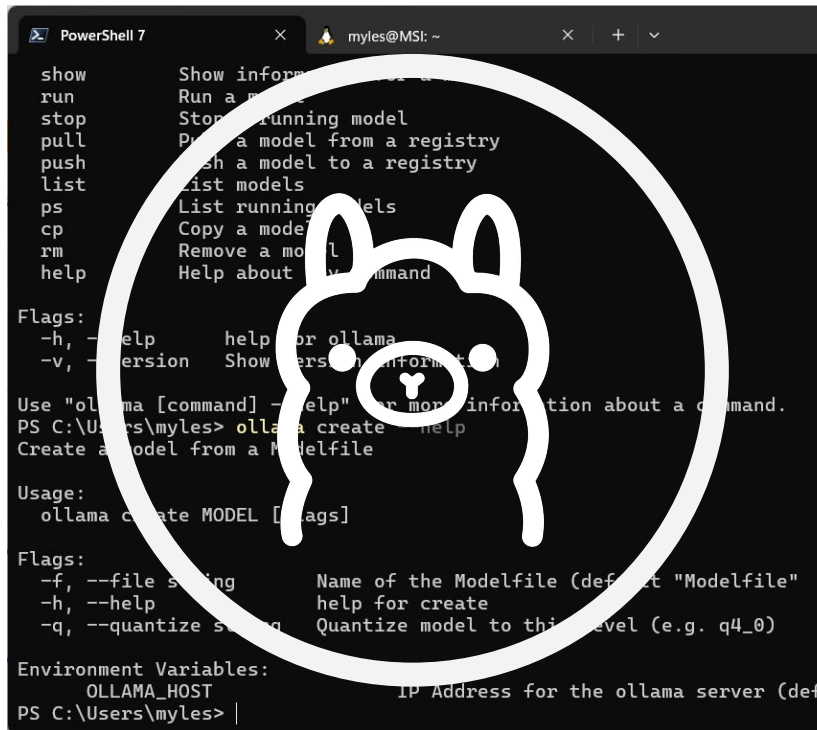


MISTRAL  
AI\_

*NLP from scratch* 

# Working with Ollama on the Command Line

- **List, Download, and Run Models:** Run `ollama list` to view models, `ollama pull <model_name>` to download, and `ollama run <model_name>` to drop into an interactive session.
- **Serve a Model:** Use `ollama serve <model_name>` to host a model locally for API usage (default port is 11434)
- **List model attributes:** Use `ollama show <model_name>` to show details about a particular model
- **Create your own model (Advanced):** Use `ollama create` to create your own Ollama model from a ModelFile



```
PowerShell 7
myles@MSI: ~

show      Show information about a model
run       Run a model
stop      Stop running model
pull      Pull a model from a registry
push      Push a model to a registry
list      List models
ps        List running models
cp        Copy a model
rm        Remove a model
help      Help about this command

Flags:
  -h, --help            help for ollama
  -v, --version         Show version information

Use "ollama [command] --help" for more information about a command.
PS C:\Users\myles> ollama create --help
Create a model from a Modelfile

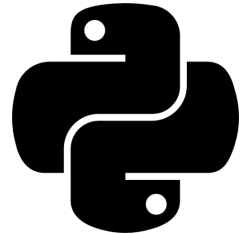
Usage:
  ollama create MODEL [flags]

Flags:
  -f, --file string      Name of the Modelfile (default "Modelfile")
  -h, --help             help for create
  -q, --quantize string  Quantize model to this level (e.g. q4_0)

Environment Variables:
  OLLAMA_HOST            IP Address for the ollama server (default "localhost:11434")
PS C:\Users\myles> |
```

# Ollama in Python

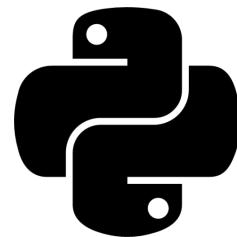
```
from ollama import Client
client = Client(
    host='http://localhost:11434',
    headers={'x-some-header': 'some-value'}
)
response = client.chat(model='llama3.2', messages=[
    {
        'role': 'user',
        'content': 'Why is the sky blue?',
    },
])
```



<https://github.com/ollama/ollama-python>

# Ollama API in Python

- **Chat:** `ollama.chat(model='llama3.2', messages=[{'role': 'user', 'content': 'Why is the sky blue?'}]])`
- **Generate:** `ollama.generate(model='llama3.2', prompt='Why is the sky blue?')`
- **List:** `ollama.list()`
- **Show:** `ollama.show('llama3.2')`
- **Create:** `ollama.create(model='example', from_='llama3.2', system="You are Mario from Super Mario Bros.")`
- **Copy:** `ollama.copy('llama3.2', 'user/llama3.2')`
- **Delete:** `ollama.delete('llama3.2')`
- **Pull:** `ollama.pull('llama3.2')`
- **Push:** `ollama.push('user/llama3.2')`
- 



# OpenAI compatibility

February 8, 2024

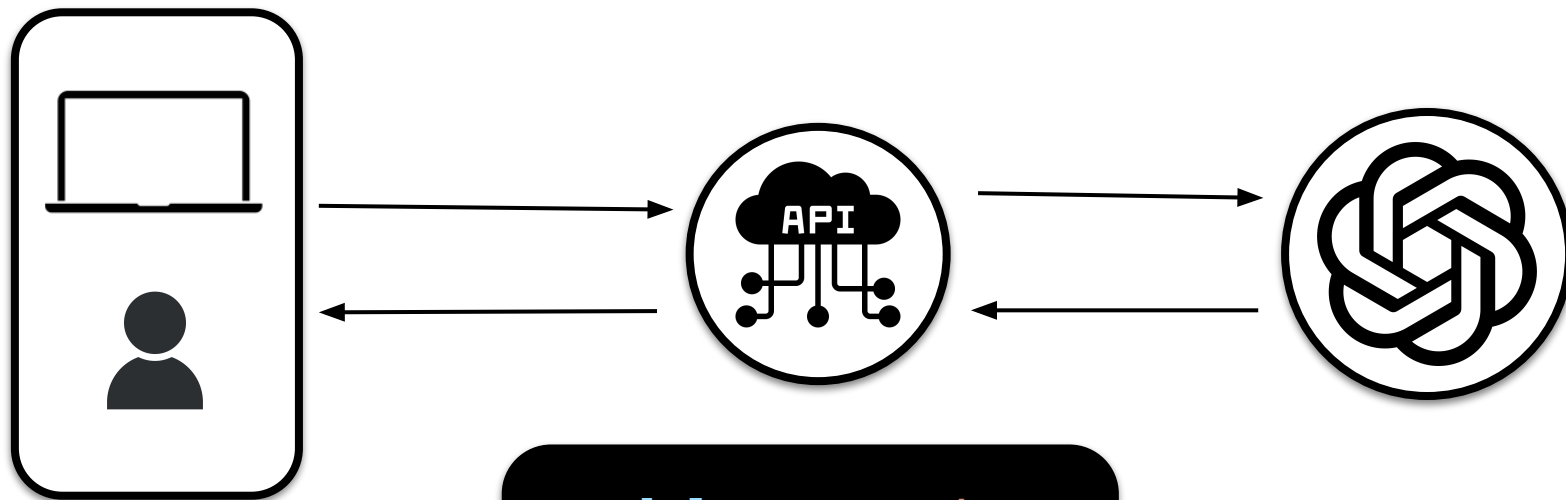
s/GPT/MyLLM



```
curl http://localhost:11434/v1/chat/completions \
  -H 'Content-Type: application/json' \
  -d '{
    "model": "llama2",
    "messages": [
      {
        "role": "user",
        "content": "Hello!"
      }
    ]
  }'
```

Ollama now has built-in compatibility with the OpenAI [Chat Completions API](#), making it possible to use more tooling and applications with Ollama locally.

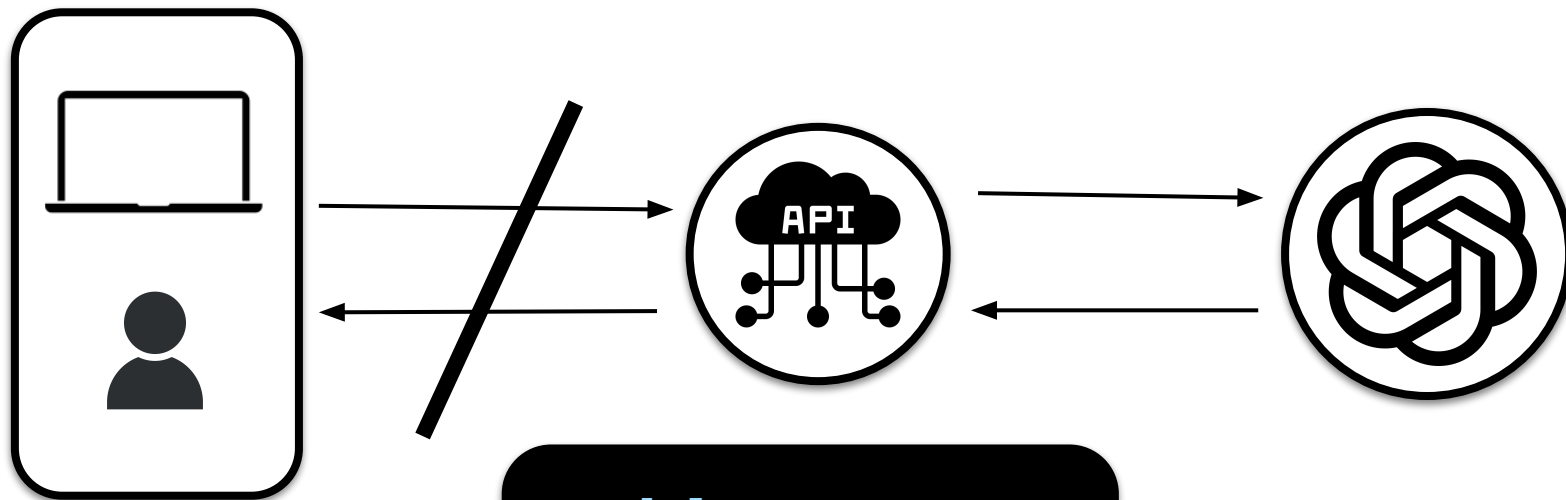
# Building an application with the OpenAI API



```
model = "gpt-4"  
client = OpenAI()
```



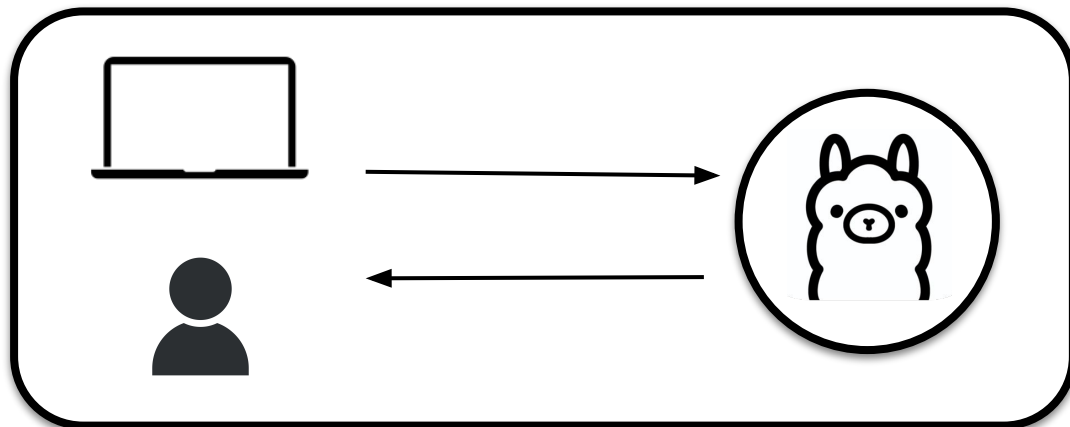
# Building an application with ~~the OpenAI API~~



```
model = " "
```

```
client = ?
```

# Building a local LLM application with Ollama



```
model = "llama3"  
client = OpenAI(base_url = 'http://localhost:11434/v1',  
                api_key='ollama')
```



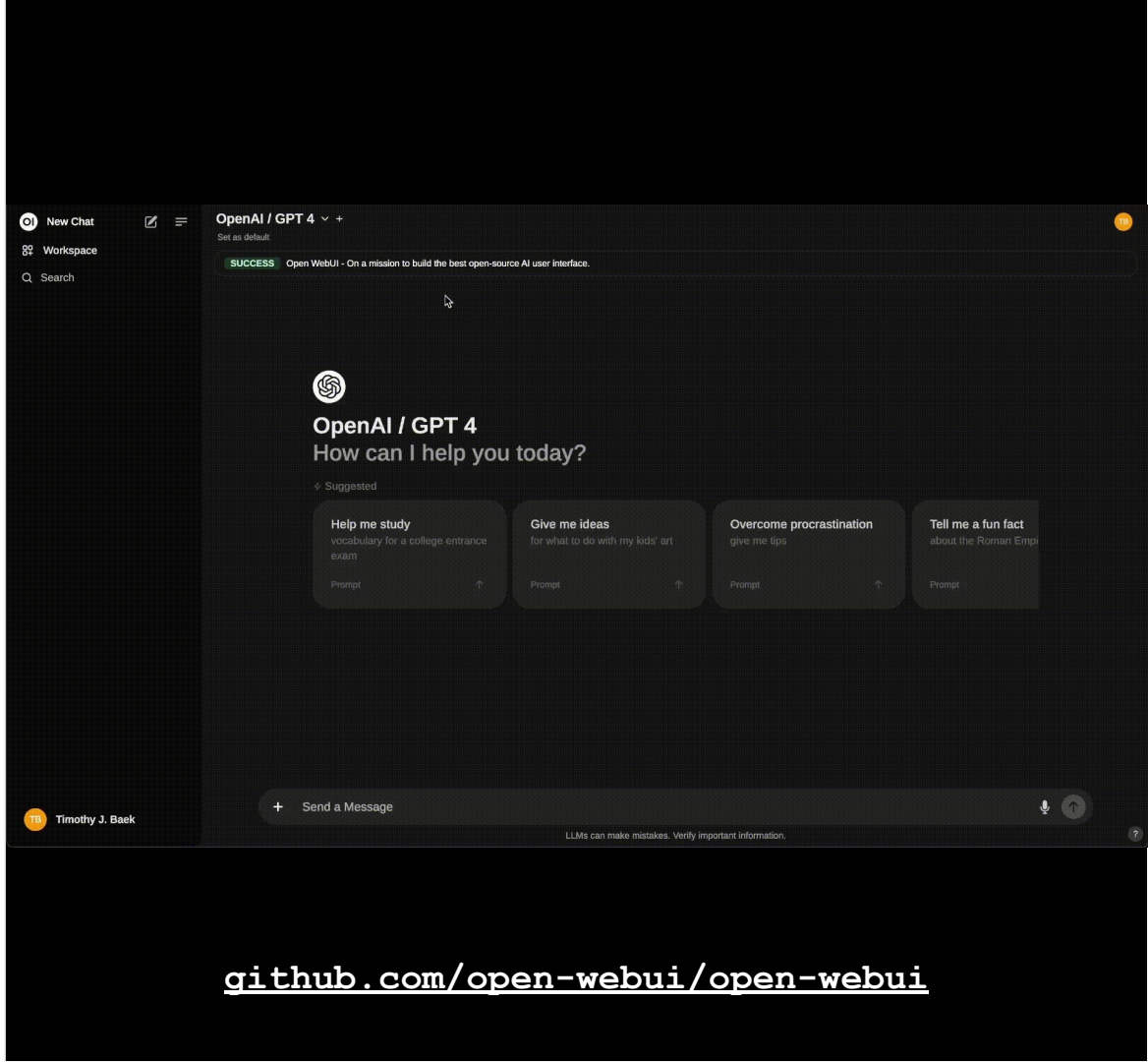
# **LOCAL LLM GUI CLIENTS**

# Local LLM GUI Clients with Ollama support

- Variety of local GUI clients available for LLMs with varying functionalities and focus
- **OpenWeb UI** (originally Ollama WebUI) has many features (RAG, annotation, agents, etc.) but requires using docker container
- **AnythingLLM** supports a wide range of models and frameworks, and can either run bundled Ollama models or with an external install
- **Jan** is a general UI for LLMs with desktop and server versions which can be integrated with Ollama
- **Chatbox** is another GUI client for multiple LLMs



# OpenWebUI Demo



[github.com/open-webui/open-webui](https://github.com/open-webui/open-webui)

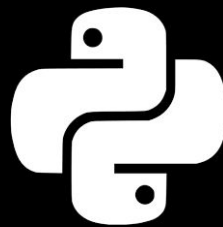
# End of Part 4

[LLMsfor.me](https://llmsfor.me)

PWYC Microcourse in LLMs and Generative AI  
January 2025

**Part 4 - Local LLM Development**

**Tuesday, January 28th, 2025**



[llmsfor.me](https://llmsfor.me)