



The University of Texas at Austin  
Machine Learning Laboratory

# OLLAMA + RAG

---

## MLL SUMMER ACADEMY [LAB 2]

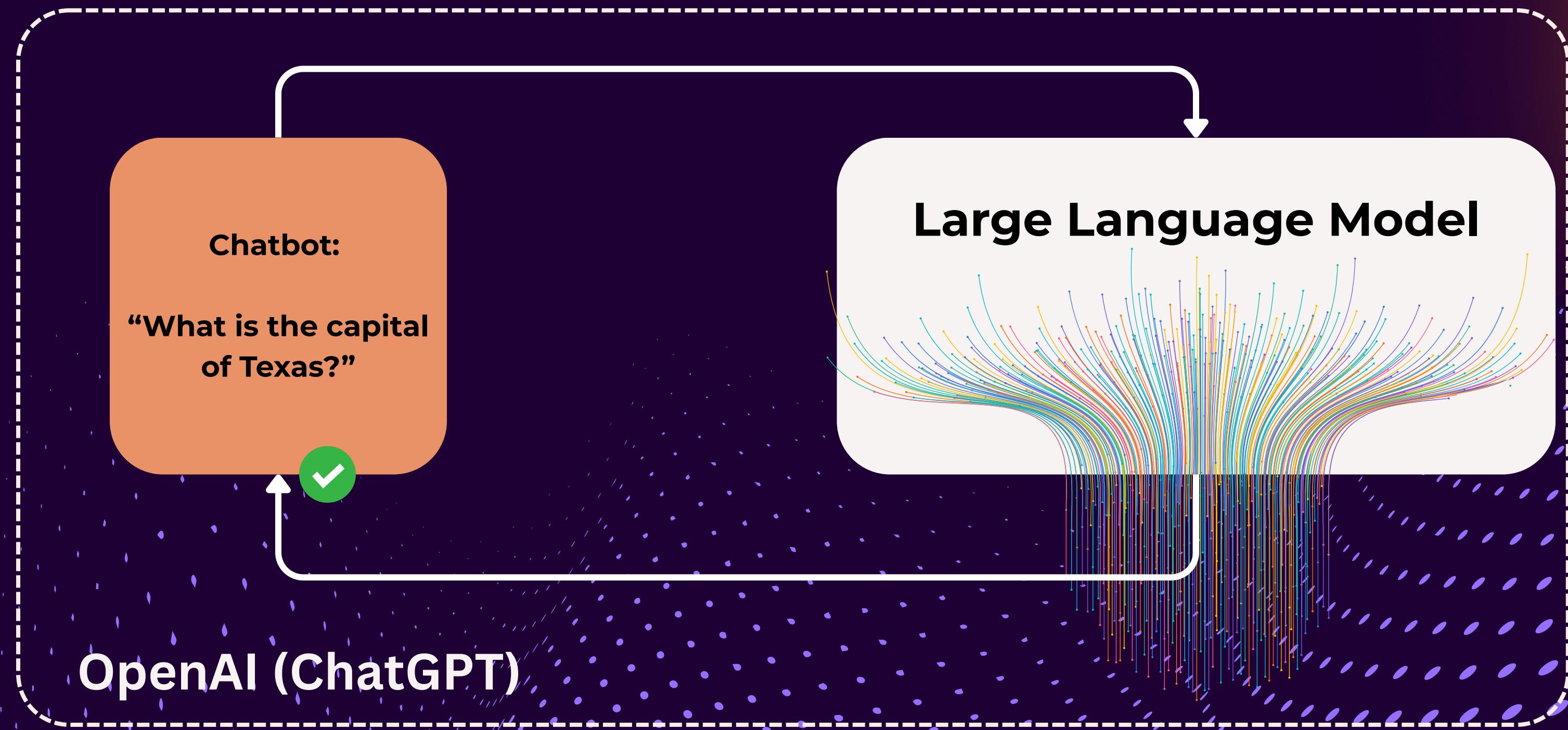
IFML

Institute for Foundations of  
MACHINE LEARNING

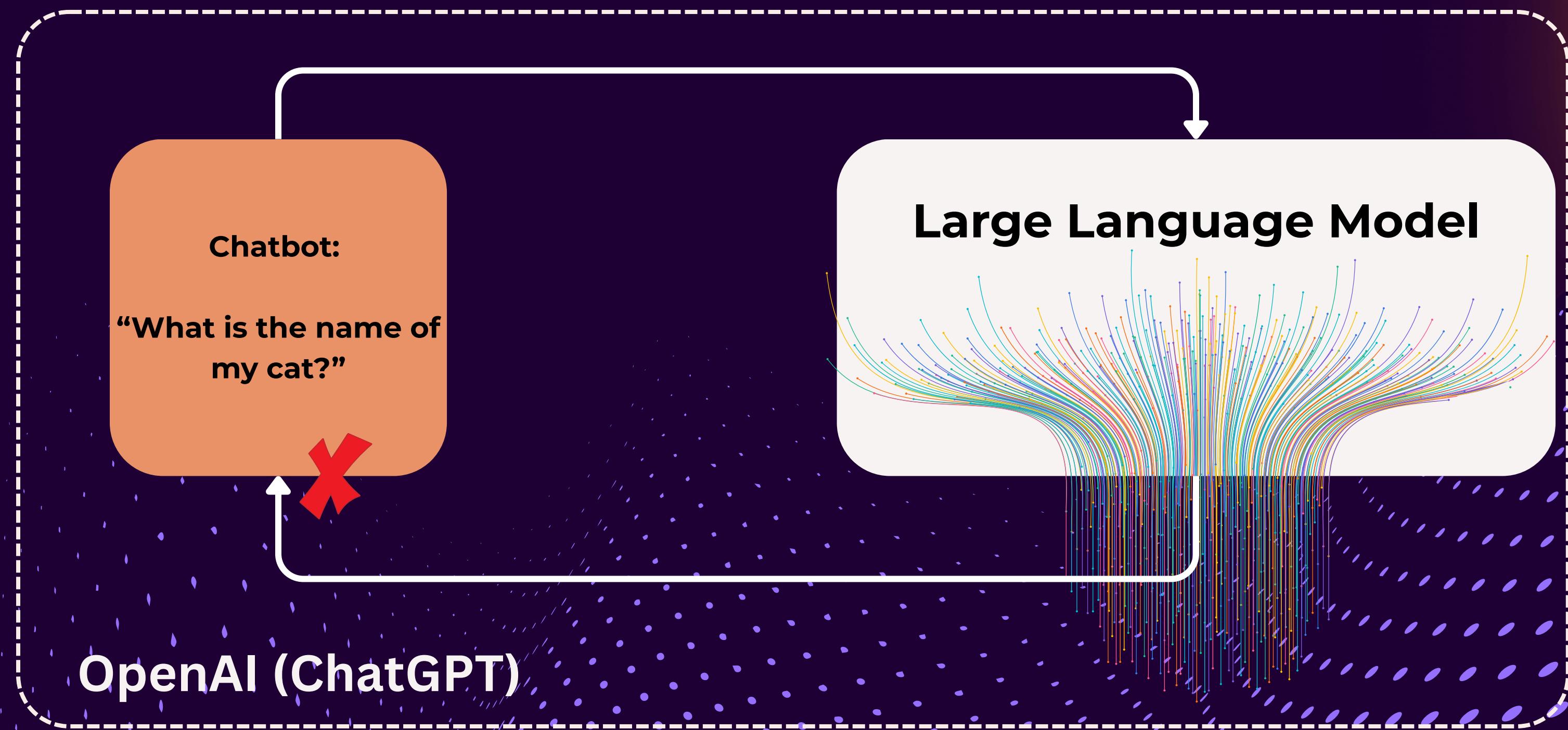
JUNE 2025

# RETRIVAL AUGMENTED GENERATION

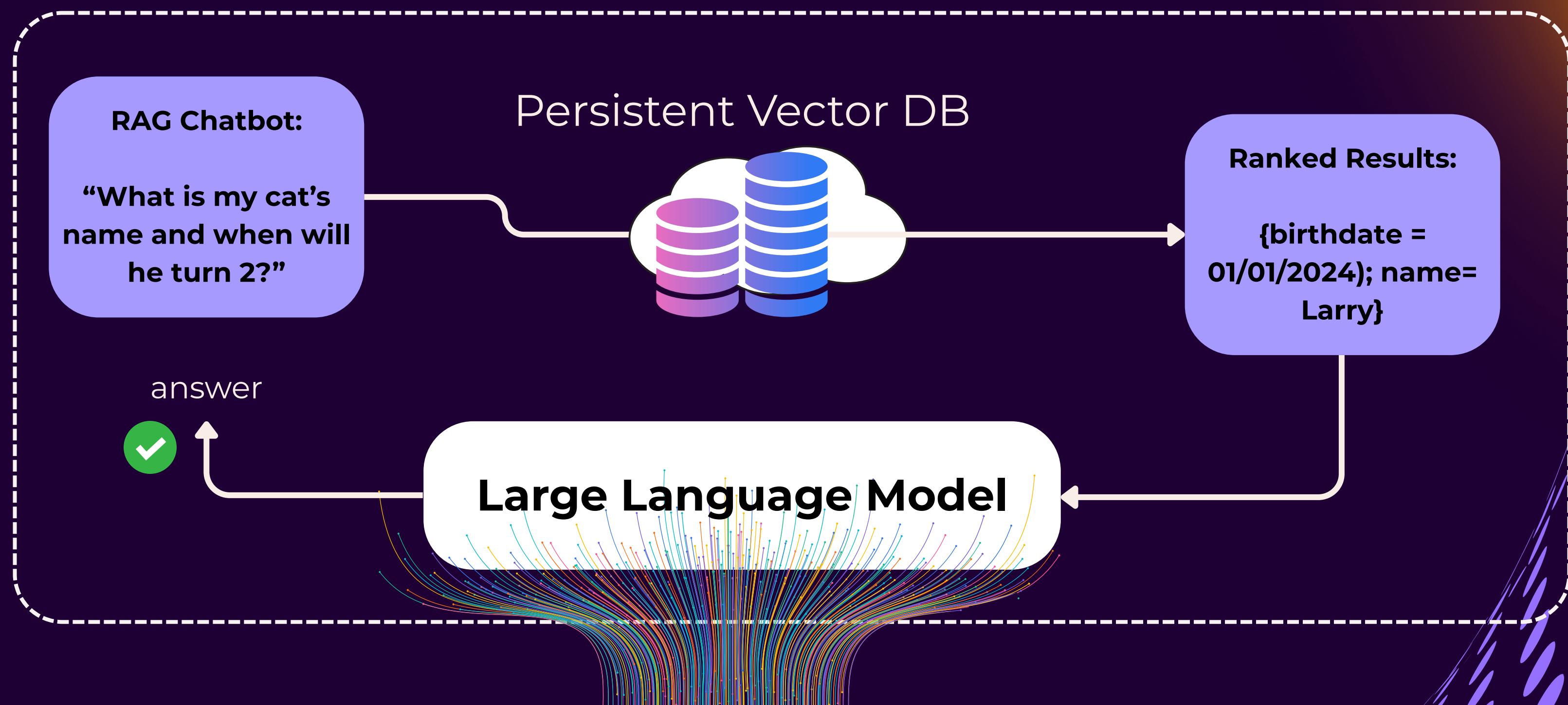
a technique that enhances large language models by retrieving relevant information from external data sources to generate more accurate, relevant, and grounded responses.



# RETRIVAL AUGMENTED GENERATION



# ADDING CONTEXT & PERSONALIZATION



# BUT WAIT... WHAT ABOUT THE COST?

Even in the RAG setup shown, there are costs (via an API) for the LLM to translate the embedding from our RAG database into text for our chat response.

**Also what if I don't want to upload all of my personal files for the database either?**

01

## **It's possible to take RAG "offline"**

Using open source LLM's and software tools like Ollama, you can run these models on your own hardware!

02

## **Increase the privacy of what you share**

The cloud is just someone else's computer, no matter how large it is. Great for many things, but you should have a choice. Running locally means no internet required!

03

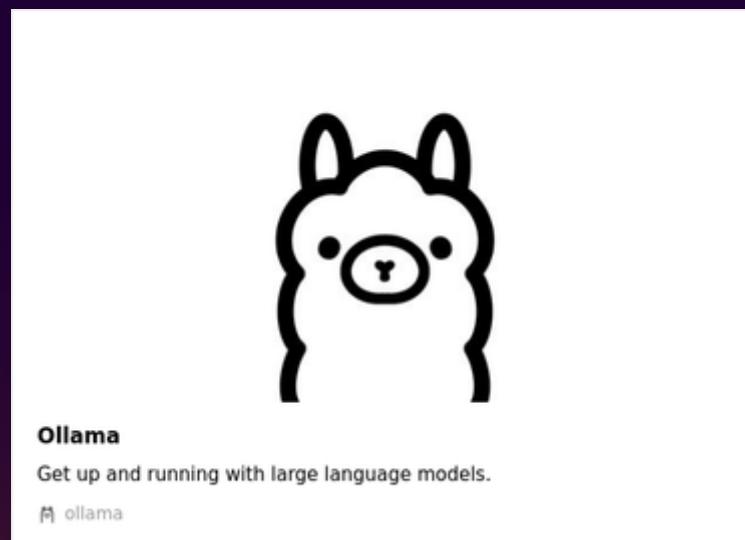
## **Adapts and improves as you use it**

Over time, each interaction you have with your system, and the more files you add, the richer the content and quality of responses will be.



# OUR LAB

You all have the skills and ability to create this kind of system. And best of all, you can refer back to this guide later and create a system like this on your own computer to use anytime and anywhere.



A large screenshot of the Ollama website, overlaid with a dark brown rectangular shape. The top navigation bar includes links for "Discord", "GitHub", and "Models", along with a search bar and "Sign in" and "Download" buttons. The main content area features a large llama logo, the tagline "Get up and running with large language models.", and a paragraph about running various models locally. A prominent "Download" button with a downward arrow is at the bottom, and a note below it states "Available for macOS, Linux, and Windows".



# OUR BUILD PLAN

We'll work through these steps together as a group, each building upon the next. By the end of the lab, you'll have downloaded 2 different model files, embedded a few PDF's, and have a working UI to ask questions and receive answers about information from those documents.

**01**

## SETUP OLLAMA SERVER & LOAD MODEL FILES

We'll do this through the command line in terminal on your Ubuntu machine in class. We're going to start with the **phi4-mini** model.

**02**

## UPLOAD PDFS OF YOUR CHOICE

Build your library - these could be anything, but the larger the file the longer it will take for our system to parse and embed the PDF. You have about 2.5 GB to work with on your Lab Machine.

**03**

## REVIEW THE PROGRAM & RUN IT!

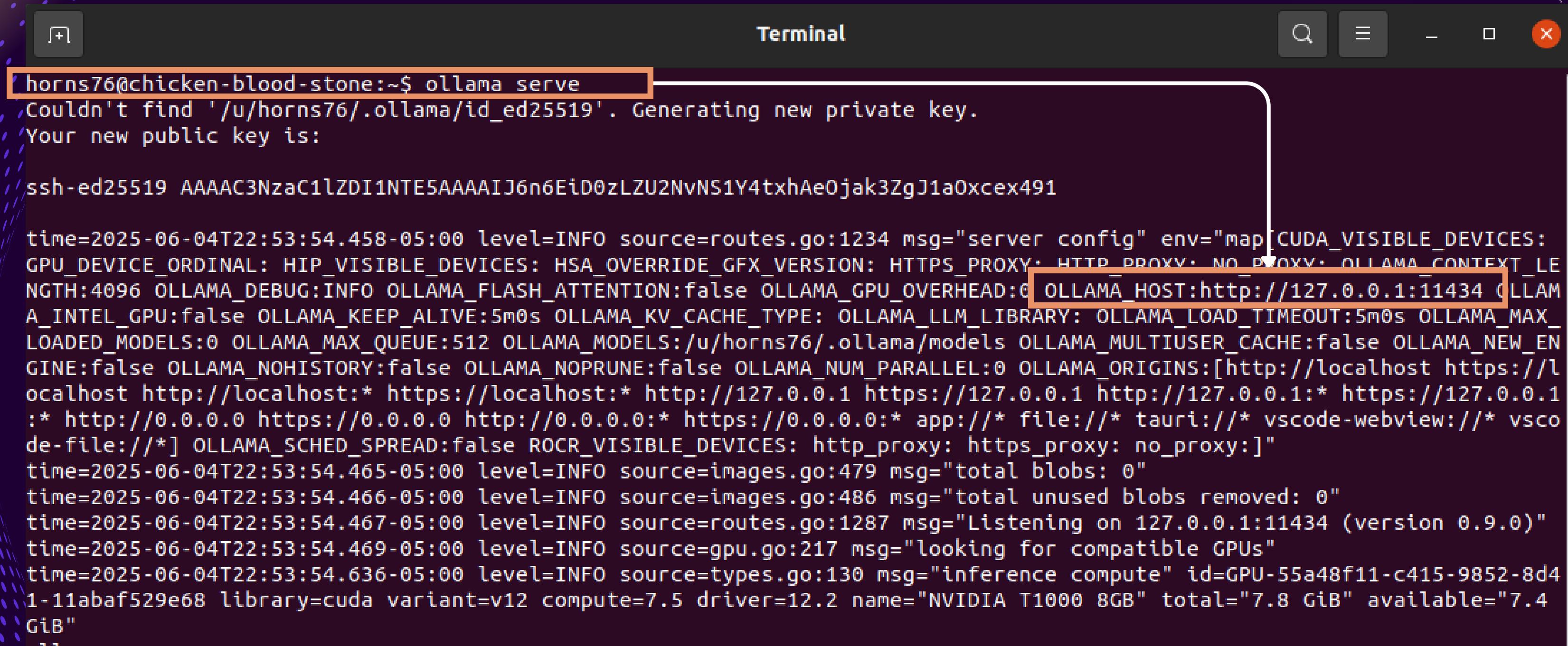
Launch the `rag_pipeline.py` file and start to chat!

PART 1:  
LAUNCH & USE  

---

**OLLAMA**

# LAUNCH OLLAMA SERVER



A screenshot of a terminal window titled "Terminal". The window shows the command "ollama serve" being run and its output. The output includes a warning about generating a new private key, a public key, and a long log of server configuration details. A red box highlights the public key and the server's listening address.

```
horn$76@chicken-blood-stone:~$ ollama serve
Couldn't find '/u/horns76/.ollama/id_ed25519'. Generating new private key.
Your new public key is:

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIj6n6EiD0zLZU2NvNS1Y4txhAe0jak3ZgJ1a0xcex491

time=2025-06-04T22:53:54.458-05:00 level=INFO source=routes.go:1234 msg="server config" env="map[CUDA_VISIBLE_DEVICES: GPU_DEVICE_ORDINAL: HIP_VISIBLE_DEVICES: HSA_OVERRIDE_GFX_VERSION: HTTPS_PROXY: HTTP_PROXY: NO_PROXY: OLLAMA_CONTEXT_LENGTH:4096 OLLAMA_DEBUG:INFO OLLAMA_FLASH_ATTENTION:false OLLAMA_GPU_OVERHEAD:0 OLLAMA_HOST:http://127.0.0.1:11434 OLLAMA_INTEL_GPU:false OLLAMA_KEEP_ALIVE:5m0s OLLAMA_KV_CACHE_TYPE: OLLAMA_LLM_LIBRARY: OLLAMA_LOAD_TIMEOUT:5m0s OLLAMA_MAX_LOADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:/u/horns76/.ollama/models OLLAMA_MULTIUSER_CACHE:false OLLAMA_NEW_ENGINE:false OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:0 OLLAMA_ORIGINS:[http://localhost https://localhost http://localhost:* https://localhost:* http://127.0.0.1 https://127.0.0.1 http://127.0.0.1:* https://127.0.0.1 :* http://0.0.0.0 https://0.0.0.0 http://0.0.0.* https://0.0.0.* app:///* file:///* tauri:///* vscode-webview:///* vscode-file:///*] OLLAMA_SCHED_SPREAD:false ROCR_VISIBLE_DEVICES: http_proxy: https_proxy: no_proxy:]"
time=2025-06-04T22:53:54.465-05:00 level=INFO source=images.go:479 msg="total blobs: 0"
time=2025-06-04T22:53:54.466-05:00 level=INFO source=images.go:486 msg="total unused blobs removed: 0"
time=2025-06-04T22:53:54.467-05:00 level=INFO source=routes.go:1287 msg="Listening on 127.0.0.1:11434 (version 0.9.0)"
time=2025-06-04T22:53:54.469-05:00 level=INFO source=gpu.go:217 msg="looking for compatible GPUs"
time=2025-06-04T22:53:54.636-05:00 level=INFO source=types.go:130 msg="inference compute" id=GPU-55a48f11-c415-9852-8d41-11abaf529e68 library=cuda variant=v12 compute=7.5 driver=12.2 name="NVIDIA T1000 8GB" total="7.8 GiB" available="7.4 GiB"
```

# VERIFY THE OLLAMA SERVER RUNNING ON YOUR MACHINE

The terminal window displays two main sections of output:

**Logs (Top Section):**

```
horn$76@chicken-blood-stone:~$ ollama serve
Couldn't find '/u/horn$76/.ollama/id_ed25519'. Generating new private key.
Your new public key is:

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAIJ6n6EiD0zLZU2NvNS1Y4txhAe0jak3ZgJ1a0xcex491

time=2025-06-04T22:53:54.458-05:00 level=INFO source=routes.go:1234 msg="server config" env="map[CUDA_VISIBLE_DEVICES: GPU_DEVICE_ORDINAL: HIP_VISIBLE_DEVICES: HSA_OVERRIDE_GFX_VERSION: HTTPS_PROXY: NO_PROXY: OLLAMA_CONTEXT_LENGTH:4096 OLLAMA_DEBUG:INFO OLLAMA_FLASH_ATTENTION:false OLLAMA_GPU_OVERHEAD:0 OLLAMA_HOST:http://127.0.0.1:11434 OLLAMA_INTEL_GPU:false OLLAMA_KEEP_ALIVE:5m0s OLLAMA_KV_CACHE_TYPE: OLLAMA_LLM_LIBRARY: OLLAMA_LOAD_TIMEOUT:5m0s OLLAMA_MAX_LOADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:/u/horn$76/.ollama/models OLLAMA_MULTIUSER_CACHE:false OLLAMA_NEW_EGINE:false OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:0 OLLAMA_ORIGINS:[http://localhost https://localhost http://localhost:* https://localhost:* http://127.0.0.1 https://127.0.0.1 http://127.0.0.1:* https://127.0.0.1:* http://0.0.0.0 https://0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app:///* tauri:///* vscode-webview:///* vscode-file:///*] OLLAMA_SCHED_SPREAD:false ROCR_VISIBLE_DEVICES: http_proxy: https_proxy: no_proxy:]"
time=2025-06-04T22:53:54.465-05:00 level=INFO source=images.go:479 msg="total blobs: 0"
time=2025-06-04T22:53:54.466-05:00 level=INFO source=images.go:486 msg="total unused blobs removed: 0"
time=2025-06-04T22:53:54.467-05:00 level=INFO source=routes.go:1287 msg="Listening on 127.0.0.1:11434 (version 0.9.0)"
time=2025-06-04T22:53:54.469-05:00 level=INFO source=gpu.go:217 msg="looking for compatible GPUs"
time=2025-06-04T22:53:54.636-05:00 level=INFO source=types.go:130 msg="inference compute" id=GPU-55a48f11-c415-9852-8d41-11abaf529e68 library=cuda variant=v12 compute=7.5 driver=12.2 name="NVIDIA T1000 8GB" total="7.8 GiB" available="7.4 GiB"
ollama
```

**Command Usage (Bottom Section):**

```
clear
[GIN] 2025/06/04 - 22:55:00 | 200 | 2.205976ms | 127.0.0.1 | HEAD | "/" "/api/tags"
[GIN] 2025/06/04 - 22:55:00 | 200 | 6.958279ms | 127.0.0.1 | GET | "/" "/api/tags"
[GIN] 2025/06/04 - 23:02:14 | 200 | 40.695μs | 127.0.0.1 | HEAD | "/" "/api/tags"
[GIN] 2025/06/04 - 23:02:14 | 200 | 1.286267ms | 127.0.0.1 | GET | "/" "/api/tags"
[GIN] 2025/06/04 - 23:02:27 | 200 | 46.172μs | 127.0.0.1 | HEAD | "/" "/api/tags"
[GIN] 2025/06/04 - 23:02:27 | 200 | 689.978μs | 127.0.0.1 | GET | "/" "/api/tags"
[GIN] 2025/06/04 - 23:03:06 | 200 | 36.713μs | 127.0.0.1 | HEAD | "/" "/api/tags"
[GIN] 2025/06/04 - 23:03:06 | 200 | 462.503μs | 127.0.0.1 | GET | "/" "/api/tags"

horn$76@chicken-blood-stone:~$ ollama
Usage:
  ollama [flags]
  ollama [command]

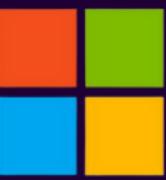
Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show       Show information for a model
  run        Run a model
  stop       Stop a running model
  pull       Pull a model from a registry
  push       Push a model to a registry
  list       List models
  ps         List running models
  cp         Copy a model
  rm         Remove a model
  help       Help about any command

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

Use "ollama [command] --help" for more information about a command.
horn$76@chicken-blood-stone:~$ ollama list
NAME   ID   SIZE   MODIFIED
horn$76@chicken-blood-stone:~$ |
```

# INSTALL OUR MODEL:

## phi4-mini



Microsoft Research

The image shows two terminal windows side-by-side. The top terminal window displays logs from the 'ollama serve' command, showing various configuration details and GPU usage. The bottom terminal window displays the help output for the 'ollama -h' command, listing available commands like 'serve', 'create', and 'run'. Both terminals have specific log entries highlighted with orange boxes. The bottom terminal also shows a progress bar for a file download.

```
hornsn76@chicken-blood-stone:~$ export OLLAMA_MODELS="/u/hornsn76"
hornsn76@chicken-blood-stone:~$ ollama serve
time=2025-06-04T23:12:06.898-05:00 level=INFO source=routes.go:1234 msg="server config" env="map[CUDA_VISIBLE_DEVICES:GPU_DEVICE_ORDINAL: HIP_VISIBLE_DEVICES: HSA_OVERRIDE_GFX_VERSION: HTTPS_PROXY: HTTP_PROXY: NO_PROXY: OLLAMA_CONTEXT_LANGUAGE:0 OLLAMA_DEBUG:INFO OLLAMA_FLASH_ATTENTION:false OLLAMA_GPU_OVERHEAD:0 OLLAMA_HOST:http://127.0.0.1:11434 OLLAMA_INTEL_GPU:false OLLAMA_KEEP_ALIVE:5m0s OLLAMA_kv_CACHE_TYPE: OLLAMA_LLM_LIBRARY: OLLAMA_LOAD_TIMEOUT:5m0s OLLAMA_MAX_LOADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:/u/hornsn76 OLLAMA_MULTIUSER_CACHE:false OLLAMA_NEW_ENGINE:false OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:0 OLLAMA_ORIGINS:[http://localhost https://localhost http://localhost:80 https://localhost:80 http://127.0.0.1 https://127.0.0.1 http://127.0.0.1:80 https://127.0.0.1:80 http://0.0.0.0 https://0.0.0.0 http://0.0.0.0:80 https://0.0.0.0:80 app:///* tauri:///* vscode-webview:///* vscode-file:///*] OLLAMA_SCHED_SPREAD:false OLLAMA_VISIBLE_DEVICES: http_proxy: https_proxy: no_proxy:]"
time=2025-06-04T23:12:06.902-05:00 level=INFO source=images.go:479 msg="total blobs: 0"
time=2025-06-04T23:12:06.902-05:00 level=INFO source=images.go:486 msg="total unused blobs removed: 0"
time=2025-06-04T23:12:06.904-05:00 level=INFO source=routes.go:1287 msg="Listening on 127.0.0.1:11434 (version 0.9.0)"
time=2025-06-04T23:12:06.904-05:00 level=INFO source=gpu.go:217 msg="looking for compatible GPUs"
time=2025-06-04T23:12:07.027-05:00 level=INFO source=types.go:130 msg="inference compute" id=GPU-55a48f11-c415-9852-8d41-11abaf529e68 library=cuda variant=v12 compute=7.5 driver=12.2 name="NVIDIA T1000 8GB" total="7.8 GiB" available="7.3 GiB"
[GIN] 2025/06/04 - 23:17:38 | 200 |      65.8μs |   127.0.0.1 | HEAD | /api/show
[GIN] 2025/06/04 - 23:17:38 | 404 |  3.148608ms |   127.0.0.1 | POST | /api/show
time=2025-06-04T23:17:39.272-05:00 level=INFO source=download.go:177 msg="downloading 3c168af1dea0 to 16 155 MB part(s)
"
hornsn76@chicken-blood-stone:~$ ollama -h
Large language model runner

Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show       Show information for a model
  run        Run a model
  stop       Stop a 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 any command

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

Use "ollama [command] --help" for more information about a command
hornsn76@chicken-blood-stone:~$ ollama run phi4-mini
pulling manifest
pulling 3c168af1dea0: 63% [███████████| 1.6 GB/2.5 GB 119 MB/s 7s]
```

# LOCAL CHAT 🔥

```
horns76@chicken-blood-stone:~$ ollama run phi4-mini
pulling manifest
pulling 3c168af1dea0: 100%
pulling 813f53fdc6e5: 100%
pulling fa8235e5b48f: 100%
pulling 8c2539a423c4: 100%
verifying sha256 digest
writing manifest
success
```

2.5 GB
655 B
1.1 KB
411 B

```
>>> hello, what is your name and what can you do?
```

Hello! I'm Phi developed by Microsoft. I am an advanced AI assistant designed to help with a wide range of tasks including answering questions about various topics such as science, history, technology, language translation, data analysis, creative writing assistance (like generating poems or stories), providing summaries for lengthy texts like articles and books.

If you have any specific request that you're curious about, feel free to ask!

```
>>> Send a message (/? for help)
```

To exit chat, type: /bye

PART 2:

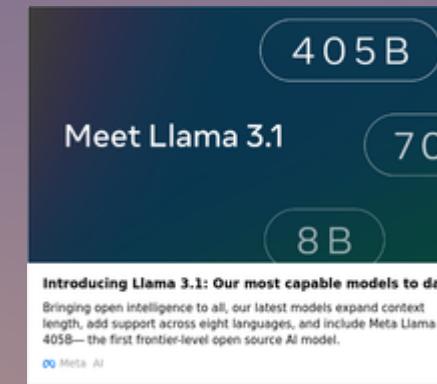
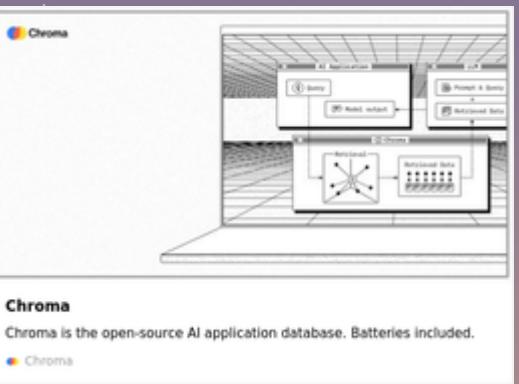
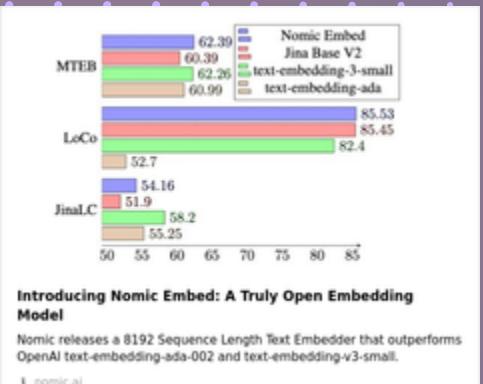
---

RAG BUILD

# OUR SYSTEM COMPONENTS

AI can automate and accelerate the incident response process, enabling faster identification, containment, and recovery from cyberattacks.

## references:



## 01

nomic-embed-text

### EMBEDDING:

This model is going to embed the parsed text from our RAG PDF files. We'll also use this model to embed our query before searching our vector database

## 02



### VECTOR DATABASE:

We're going to use the lightweight and powerful ChromaDB library for our local database.

## 03



### LLM:

Llama3.1 is an open-source family of reasoning models built using a dataset developed and maintained by Meta.

# OUR PLAN

---

Since this is going to be a terminal based RAG system, we're going to build each component in sequential order, one after the other.

- 1: `pdf_loader.py`
- 2: `pdf_manager.py`
- 3: `embedder.py`
- 4: `retriever.py`
- 5: `ollama_runner.py`
- 6: `chat_loop.py`
- 7: `rag_pipeline.py`

load & chunk PDFs  
detect & track new PDF's  
send chunks for embedding  
ChromaDB storage, retrieval, & reranking  
build prompt, send to LLM  
handle input / output for chat  
bring it all together

# CLONE THIS REPO:

```
git clone https://github.com/IFML-UT/MLLAcademy-2025_0llamaRAG.git
```