# Lab 16. Getting started with LocalChat

# (To be Executed on your own Machine)

You get to decide the setup for these 3 main components:

- 1. **LLM**: the large language model provider used for inference. It can be local, or remote, or even OpenAI.
- 2. **Embeddings**: the embeddings provider used to encode the input, the documents and the users' queries. Same as the LLM, it can be local, or remote, or even OpenAI.
- 3. **Vector store**: the store used to index and retrieve the documents.

# Sequence 1. Installing Ollama

#### a) Installing on Linux

- i. Login to your linux machine as root user and launch the terminal.
- ii. Run the following command from terminal

```
curl -fsSL https://ollama.com/install.sh | sh
```

### b) Installing on Windows

i. Download the following installer for windows

```
https://ollama.com/download/OllamaSetup.exe
```

ii. Run the Installer and follow the instructions

# c) Installing on MacBook (M1/M2/M3 Chipset)

i. Download the Installer

```
https://ollama.com/download/Ollama-darwin.zip
```

ii. Extract the Installer and Run.

# Sequence 2. Installing LocalChat

- 1. Create a Python Environment with Python version 3.11 with pyenv or conda:
  - a. Using pyenv
    - Check the default version set by pyenv. The \* indicates that the system Python version is active currently.

```
$ pyenv versions

* system (set by /home/opc/.pyenv/version)
3 11 9
```

• If you see "system", this means that, by default, you are still using your system Python:

```
$ python -V
```

• Change the default global version to python 3.11.9 with:

#### b. With conda

Check default Python Version (No Active Environment):

```
$ Python -V
Python 3.12.6
```

• Activate base environment and check Python Version:

```
info@sangwan ~/labs$ conda activate base
(base) info@sangwan ~/labs$ Python -V
Python 3.12.8
```

• Create an environment with Python 3.11 version:

```
$ conda create -n LocalChat Python=3.11.9
```

• Activate New environment and check Python Version:

```
$ conda activate LocalChat
(LocalChat) info@sangwan ~/labs$ python -V
Python 3.11.9
```

2. Clone the LocalChat Repository and change to the downloaded directory:

```
git clone https://github.com/Sangwan70/local-chat
cd local-chat
```

```
[(LocalChat) info@sangwan ~/labs$ git clone https://github.com/Sangwan70/local-chat Cloning into 'local-chat'...
remote: Enumerating objects: 173, done.
remote: Counting objects: 100% (173/173), done.
remote: Compressing objects: 100% (146/146), done.
remote: Total 173 (delta 6), reused 173 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (173/173), 564.43 KiB | 3.87 MiB/s, done.
Resolving deltas: 100% (6/6), done.
[(LocalChat) info@sangwan ~/labs$ cd local-chat
(LocalChat) info@sangwan ~/labs$/local-chat$
```

3. Install Poetry for dependency management.

```
pip install -qU poetry
```

4. Setup for Windows (Skip to step 5 for Linux/Mac)

```
set PGPT PROFILES=ollama make run
```

5. Open another Tab of your terminal and **Start Ollama to** start a local inference server, serving both the LLM and the Embeddings:

```
ollama serve
```

If you get the following error, that would mean ollama is already running.

Error: listen tcp 127.0.0.1:11434: bind: address already in use

```
info@MacBook-Pro private-gpt % pip install -qU poetry
info@MacBook-Pro private-gpt % ollama serve
2024/12/05 22:14:54 routes.go:1197: INFO server config env="map[HTTPS_PROXY: HTTP_
PU_OVERHEAD:0 OLLAMA_HOST:http://127.0.0.1:11434 OLLAMA_KEEP_ALIVE:5m0s OLLAMA_LLM
QUEUE:512 OLLAMA_MODELS:/Users/info/.ollama/models OLLAMA_MULTIUSER_CACHE:false OL
IGINS:[http://localhost https://localhost http://localhost:* h
http://0.0.0.0 https://0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app://* file://*
ttp_proxy: https_proxy: no_proxy:]"
time=2024-12-05T22:14:54.539+05:30 level=INFO source=images.go:753 msg="total blob
time=2024-12-05T22:14:54.542+05:30 level=INFO source=images.go:760 msg="total unus
time=2024-12-05T22:14:54.542+05:30 level=INFO source=common.go:1248 msg="Listening
time=2024-12-05T22:14:54.542+05:30 level=INFO source=common.go:135 msg="extracting
ama3734876440/runners
time=2024-12-05T22:14:54.595+05:30 level=INFO source=common.go:49 msg="Dynamic LLM
time=2024-12-05T22:14:54.616+05:30 level=INFO source=types.go:123 msg="inference c
="21.3 GiB" available="21.3 GiB"
```

6. The default settings-ollama.yaml is configured to user Ilama3.2 LLM (~2GB) and nomic-embed-text Embeddings (~275MB)

By default, PGPT will automatically pull models as needed.

7. Back to previous Tab of your terminal, install LocalChat with:

```
poetry install --extras "ui llms-ollama embeddings-ollama vector-stores-qdrant"
```

```
. . .
                                                                                                             local-chat — info@sangwan — -zsh — 141×56
                                                                 ..bs/local-chat
((LocalChat) info@sangwan ~/labs/local-chat$ poetry install --extras "ui llms-ollama embeddings-ollama vector-stores-qdrant"
 Installing dependencies from lock file
Package operations: 114 installs, 16 updates, 5 removals
    - Removing build (1.2.2)
- Removing importlib-metadata (8.4.0)
- Removing pyproject-hooks (1.1.0)
- Removing zipp (3.20.2)
- Removing zstandard (0.23.0)
- Installing wrapt (1.16.0)
- Downgrading certifi (2025.1.31 -> 2024.8.30)
- Downgrading charset-normalizer (3.4.1 -> 3.3.2)
- Installing cryptography (3.4.8)
- Installing deprecated (1.2.14)
- Installing frozenlist (1.4.1)
- Installing myuv-extensions (1.0.0)
- Installing myuv-extensions (1.0.0)
     - Installing mypy-extensions (1.0.0)
```

8. Now run LocalChat. (Make sure you have a working Ollama running locally.)

```
PGPT PROFILES=ollama make run
```

```
22:19:85.400 [INFO ] private_gpt.components.llm.llm_component - Initializing the LLM in mode=ollama
22:19:85.486 [INFO ] httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"
22:19:86.488 [INFO ] private_gpt.components.embedding.embedding_component - Initializing the embedding model in mode=ollam 22:19:86.637 [INFO ] private_gpt.components.embedding.embedding_component - Initializing the embedding model in mode=ollam 22:19:86.679 [INFO ] httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"
22:19:86.679 [INFO ] httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"
22:19:86.679 [INFO ] private_gpt.components.ingest.ingest_component - Creating a new vector store index
Parsing nodes: 8it [80:80, 7it/s]
22:19:16.377 [INFO ] private_gpt.ui.ui - Mounting the gradio UI, at path=/
22:19:16.828 [INFO ] uvicorn.error - Started server process [160:80] uvicorn.error - Waiting for 12:19:16.828 [INFO ] uvicorn.error - Waiting for 12:19:16.829 [INFO ]
```

If you get any error related to Module Not found.. Install the module with pip. For example "pip install -qU build" and run the above command again.

```
(LocalChat) info@sangwan ~/labs/local-chat$ PGPT_PROFILES=ollama make run
               poetry run python -m local_chat
               No module named 'build'
               make: *** [run] Error 1
(LocalChat) info@sangwan ~/labs/local-chat$ pip install -qU build (LocalChat) info@sangwan ~/labs/local-chat$ PGPT_PROFILES=ollama make run
 18:28:86.666 [INFO ] local_chat.settings.settings_loader - Starting application with profiles=['default', 'ollama']
None of PyTorch, TensorFlow >= 2.0, or Flax have been found. Models won't be available and only tokenizers, configuration and f ties can be used.

18:28:23.611 [INFO ] local_chat.components.llm.llm component = Table 1.
                                       httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"

httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"

local_chat.components.embedding.embedding_component - Initializing the embedding model in mode=

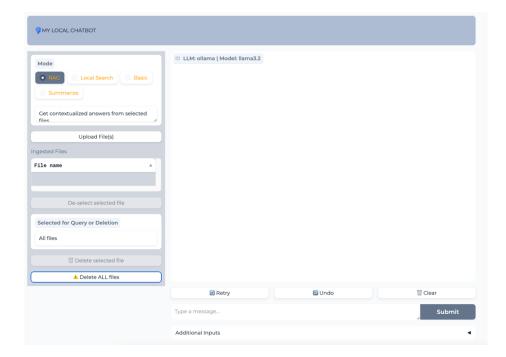
httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"

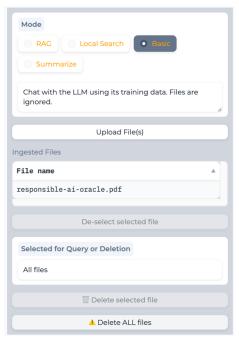
httpx - HTTP Request: GET http://localhost:11434/api/tags "HTTP/1.1 200 OK"

llama_index.core.indices.loading - loading all indices
 18:28:23.653 [INFO
 18:28:23.655 [INFO
 18:28:26.262 [INFO
 18:28:26.265 [INFO
 18:28:26.265 [INFO
18:28:26.265 [INFO
                                       Parsing nodes: 0it [00:00, 7it/s]
Generating embeddings: 0it [00:00, 7it/s]
18:28:53.537 [INFO ] local_cha
                                                           local_chat.ui.ui - Mounting the gradio UI, at path=/
                                                              uvicorn.error - Started server process [3768]
uvicorn.error - Waiting for application startup.
uvicorn.error - Application startup.
uvicorn.error - Uvicorn running on http://e.e.e.e.8801 (Press CTRL+C to quit)
 18:28:54.011 [INFO
 18:28:54.011 [INFO
18:28:54.012 [INFO
18:28:54.012 [INFO
```

9. LocalChat will use settings-ollama.yaml settings file, which is already configured to use Ollama LLM and Embeddings, and Qdrant.

Launch the browser and launch the UI (available at http://localhost:8001)



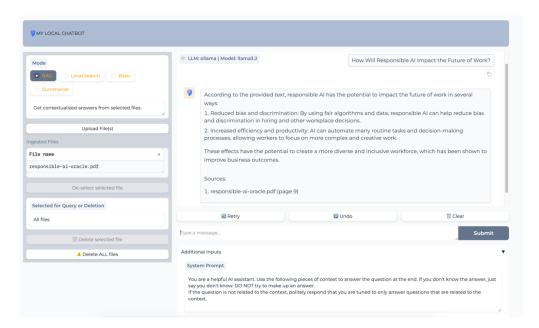


# 10. Select RAG and set the System Prompt to:

"You are a helpful AI assistant. Use the following pieces of context to answer the question at the end. If you don't know the answer, just say you don't know. DO NOT try to make up an answer. If the question is not related to the context, politely respond that you are tuned to only answer questions that are related to the context."



11. Ask a Question "How Will Responsible Al Impact the Future of Work?"



12. Try asking a question not in the context. For example ask about yourself (I asked Tell me something about Ram N Sangwan)

