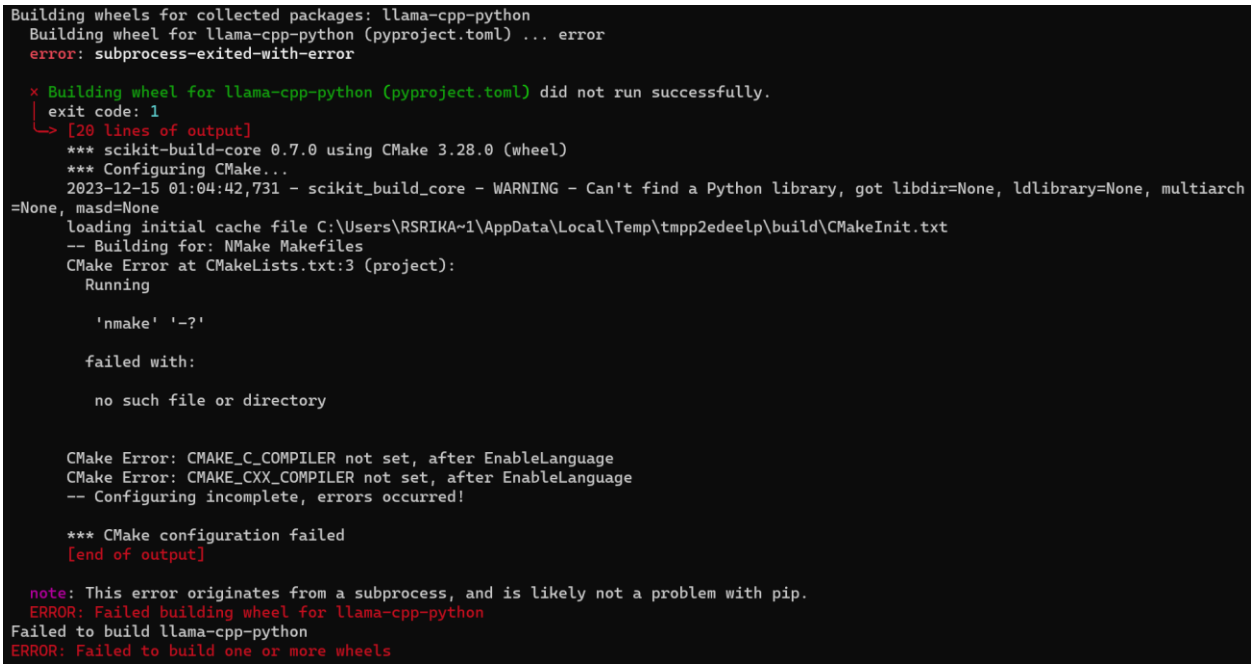
- **PrivateGPT**
  Source: https://www.youtube.com/watch?v=XFiof0V3nhA
  https://docs.privategpt.dev/installation

On Windows:

1. # Clone the repo
   git clone https://github.com/imartinez/privateGPT
2. cd privateGPT
3. Optional:
   conda create --n myenvPrivateGPT python=3.11
   conda activate myenvPrivateGPT
4. https://python-poetry.org/docs/
   pip install poetry
5. # Install dependencies (Run below command from PrivateGPT directory)
   poetry install --with ui,local

   Note: There was issue with Installing llama-cpp-python (0.2.18): Failed
   ChefBuildError

   Solution:
   python.exe -m pip install --upgrade pip
   pip wheel --no-cache-dir --use-pep517 "llama-cpp-python (==0.2.18)"



https://stackoverflow.com/questions/77267346/error-while-installing-python-package-llama-cpp-python

Solution: Install gcc and g++

1. Download Visual Studio
2. Execute VisualStudioSetup.exe
3. In Visual Studio Installer, In Workloads Select following and Install
   Python development
   Node.js development (?)
   Desktop Development with C++
   Linux and embedded development with C++
4. After successful installation, Reboot the PC
5. Check the gcc and g++ versions in cmd

```
C:\Users\rsrikanth\Downloads>gcc --version
gcc (GCC) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.


C:\Users\rsrikanth\Downloads>g++ --version
g++ (GCC) 13.2.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Now install llama-cpp-python using:

pip wheel --no-cache-dir --use-pep517 "llama-cpp-python (==0.2.18)"

```
C:\Users\rsrikanth\Desktop\D\Temp\UT_ML\Srikanth\PvtGPT\privateGPT>pip wheel --no-cache-dir --use-pep517 "llama-cpp-python (==0.2.18)
"
Collecting llama-cpp-python==0.2.18
  Downloading llama_cpp_python-0.2.18.tar.gz (7.8 MB)
     ---------------------------------------- 7.8/7.8 MB 2.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'ReadTimeoutError(
"HTTPSConnectionPool(host='pypi.org', port=443): Read timed out. (read timeout=15)")': /simple/typing-extensions/
Collecting typing-extensions>=4.5.0 (from llama-cpp-python==0.2.18)
  File was already downloaded c:\users\rsrikanth\desktop\d\temp\ut_ml\srikanth\pvtgpt\privategpt\typing_extensions-4.9.0-py3-none-any
.whl
Collecting numpy>=1.20.0 (from llama-cpp-python==0.2.18)
  File was already downloaded c:\users\rsrikanth\desktop\d\temp\ut_ml\srikanth\pvtgpt\privategpt\numpy-1.26.2-cp311-cp311-win_amd64.w
hl
Collecting diskcache>=5.6.1 (from llama-cpp-python==0.2.18)
  File was already downloaded c:\users\rsrikanth\desktop\d\temp\ut_ml\srikanth\pvtgpt\privategpt\diskcache-5.6.3-py3-none-any.whl
Building wheels for collected packages: llama-cpp-python
  Building wheel for llama-cpp-python (pyproject.toml) ... done
  Created wheel for llama-cpp-python: filename=llama_cpp_python-0.2.18-cp311-cp311-win_amd64.whl size=1687416 sha256=db7dc84c5cdf6a92
e2e803dd434873ecfa48a4cc7b743b001e2e12c390908f15
  Stored in directory: C:\Users\rsrikanth\AppData\Local\Temp\pip-ephem-wheel-cache-x1ushq4s\wheels\dd\0a\5c\b96120dd6dc069918877040e7
3f1e1bed3c17001804f8e0a21
Successfully built llama-cpp-python

C:\Users\rsrikanth\Desktop\D\Temp\UT_ML\Srikanth\PvtGPT\privateGPT>
```

Just to be sure execute “poetry install --with ui,local” to make sure that all the dependencies are installed.

```
C:\Users\rsrikanth\Desktop\D\Temp\UT_ML\Srikanth\PvtGPT\privateGPT>poetry install --with ui,local
Installing dependencies from lock file

No dependencies to install or update

Installing the current project: private-gpt (0.1.0)
```

6. Run the “scripts/setup” script using poetry.
   The settings to customize privateGPT are in the settings.yaml file. So before executing the setup script once check the settings.yaml file.

   Default: By default Mistral-7B-Instruct-v0.1-GGUF model is used. It can be changed. Even the default embedding model (BAAI/bge-small-en-v1.5) also can be changed.

   local:
   prompt_style: "llama2"
   llm_hf_repo_id: TheBloke/Mistral-7B-Instruct-v0.1-GGUF
   llm_hf_model_file: mistral-7b-instruct-v0.1.Q4_K_M.gguf
   embedding_hf_model_name: BAAI/bge-small-en-v1.5

   can be changed to
   llm_hf_repo_id: TheBloke/Llama-2-7B-Chat-GGUF
   llm_hf_model_file: llama-2-7b-chat.Q4_K_M.gguf

Note: privateGPT is using llama-cpp to run the models, so we have to use GGUF model format.

Be default Llama.cpp uses 256 as max_new_tokens. Change it to 4K,
llm:
mode: local
max_new_tokens: 4096

The setup script will download both the embedding and the LLM model and place them in the correct location (under models folder).
Download Embedding and LLM models
poetry run python scripts/setup

```
Fetching 13 files: 100%|| 13/13 [00:02<00:00,  5.24it/s]
Embedding model downloaded!
Downloading models for local execution...
llama-2-7b-chat.Q4_K_M.gguf: 100%|| 4.08G/4.08G [01:52<00:00, 15.8MB/s]
LLM model downloaded!
Setup done
```

7. (Optional) For Mac with Metal GPU, enable it. Check Installation and Settings section to know how to enable GPU on other platforms

CMAKE_ARGS="-DLLAMA_METAL=on" pip install --force-reinstall --no-cache-dir llama-cpp-python

For Windows NVIDIA GPU support check https://docs.privategpt.dev/installation

CMAKE_ARGS='-DLLAMA_CUBLAS=on'; poetry run pip install --force-reinstall --no-cache-dir llama-cpp-python

8. # Run the local server
   #IMPORTANT FOR WINDOWS USERS: the command, "PGPT_PROFILES=local make run" DOES NOT WORK. Instead, use "set PGPT_PROFILES=local" followed by "make run"
   PGPT_PROFILES=local make run

Note: on Mac with Metal you should see a ggml_metal_add_buffer log, stating GPU is being used

9. # Navigate to the UI and try it out!
   http://localhost:8001/