

Project: Build an Advanced Retrieval-Augmented Generation (RAG) Pipeline using Ollama LLM model for offline use, as well as OpenAI API integration for a hosted solution. With custom web interface.

Project Description:

This document details the step-by-step process of installing Ollama, Cloning/Forking the initial repo from GitHub in Visual Studio 2022 Community, to initiating the RAG pipeline, reading and parsing information from various formats, and further enhancements needed to carry out step-by-step prompting and extraction of results in .csv or .docx format.

Disclaimer:

nexusync 0.3.6

This project is built upon Jialin Yang (zakkyang) nexusync 0.3.6 Project as a starting point: -

A powerful document indexing and querying tool built on top of LlamaIndex and ChromaDB

NexuSync is a lightweight yet powerful library for building Retrieval-Augmented Generation (RAG) systems, built on top of **LlamaIndex**. It offers a simple and user-friendly interface for developers to configure and deploy RAG systems efficiently. Choose between using the **Ollama LLM** model for offline, privacy-focused applications or the **OpenAI API** for a hosted solution.



[pypi](#)

[GitHub](#)

- **License:** MIT License
- **Author:** [Zakk Yang](#)
- **Requires:** Python >=3.10

Further integrations are referenced when inferred below throughout the project phases.

Table of Contents:**Contents**

I-	Project Initiation.....	3
a.	Installing Ollama	3
b.	Cloning the repo via Visual Studio (Local Copy Only).....	3
c.	Forking the repo via GitHub (Local Copy Only)	4
II-	Install NexuSync	5
a.	Installing Python.....	6
b.	Installing Pytorch.....	6
c.	Installing conda for OpenAI API env control	8
d.	 Installation	10
III-	Install llama3.2	11
a.	Pull llama3.2	11
IV-	Generate OpenAI API KEY	11
V-	Set Up Your Environment	13
a.	Install Git for Windows	13
b.	Running the Bash Scripts:	18
c.	Open the Terminal	19
d.	Use the -prefix or -p Option.....	19
e.	Create Conda Environment within Visual Studio	19
VI-	 Quick Start	24
a.	Import NexuSync.....	24
b.	Run LLM model (llama3.2)	25
c.	Comment OpenAI model for now and add Ollama Model	25
d.	Updated Sample Docs with Task Files (Q1 & Q2).....	25
e.	Run LLM model (llama3.2)	26
VII.	Troubleshooting	27
a.	Let's try solving both questions simultaneously in each window:	27
VIII.	Further Actions.....	41
IX.	Thank you for this exciting opportunity to apply my knowledge and skills in LLM, RAG, and Prompt engineering domains.	41

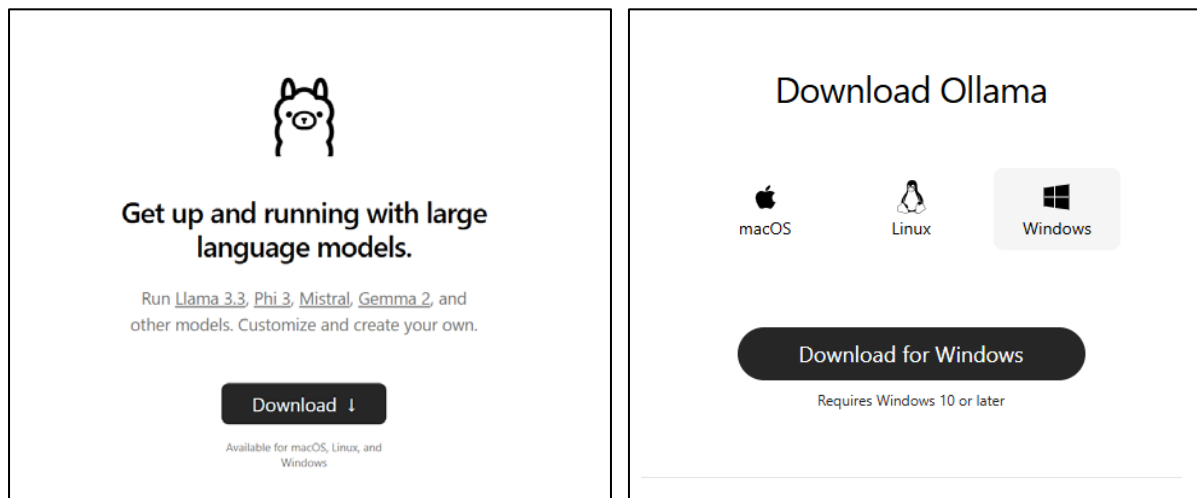
I- Project Initiation

a. Installing Ollama

What is Ollama?

[Ollama](#) is a tool designed to simplify the process of running open-source large language models (LLMs) directly on your computer. It acts as a local model manager and runtime, handling everything from downloading the model files to setting up a local environment where you can interact with them.

You can install Ollama very easily through the **Download** button in their [website](#).

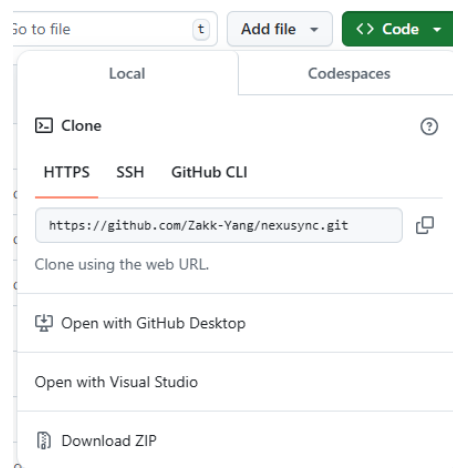


Tutorial by Krishna Sarathi Ghosh on [freeCodeCamp](#):

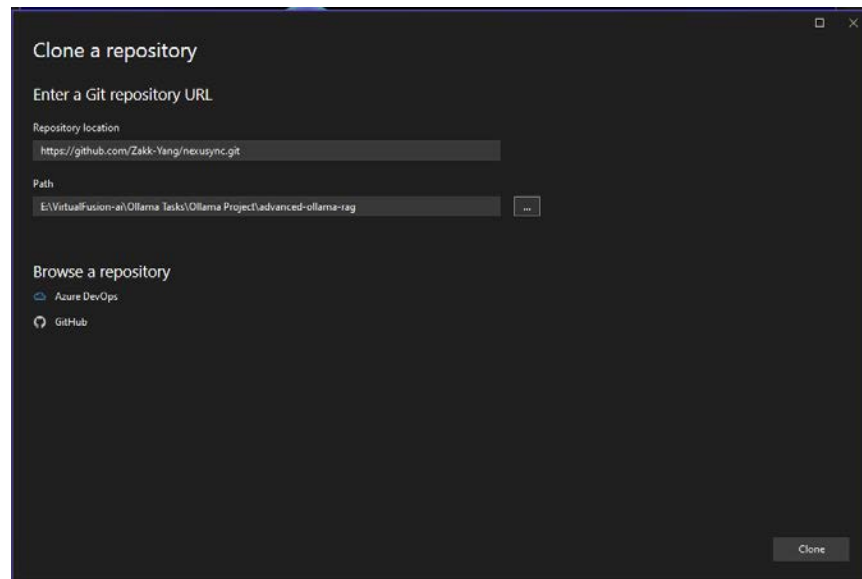
How to Run Open Source LLMs on Your Own Computer Using Ollama

b. Cloning the repo via Visual Studio (Local Copy Only)

1. Click on '<> Code' within the GitHub Repo and copy the [HTTPS link](#).



2. Within Visual Studio, click on “Git”
3. Select “Clone Repository..”
4. Enter details and click on “Clone”



c. Forking the repo via GitHub (Local Copy Only)

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner * / Repository name *

ollama-rag is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ Copy the `main` branch only
Contribute back to Zakk-Yang/ollama-rag by adding your own branch. [Learn more.](#)

📘 You are creating a fork in your personal account.

[Create fork](#)

Now you can Clone your own Forked Repository and easily edit it locally while pushing commits to the main branch.

II- Install NexuSync

Features

- **Lightweight Design:** Simplify the integration and configuration of RAG systems without unnecessary complexity.
 - **User-Friendly Interface:** Intuitive APIs and clear documentation make setup a breeze.
 - **Flexible Document Indexing:** Automatically index documents from specified directories, keeping your knowledge base up-to-date.
 - **Efficient Querying:** Use natural language to query your document collection and get relevant answers quickly.
 - **Conversational Interface:** Engage in chat-like interactions for more intuitive information retrieval.
 - **Customizable Embedding Options:** Choose between HuggingFace Embedding models or OpenAI's offerings.
 - **Incremental Updates:** Easily update and insert new documents into the index or delete the index for removed documents.
 - **Automatic Deletion Handling:** Documents removed from the filesystem are automatically removed from the index.
 - **Extensive File Format Support:** Supports multiple file formats including .csv, .docx, .epub, .hwp, .ipynb, .mbox, .md, .pdf, .png, .ppt, .pptm, .pptx, .json, and more.
-

Prerequisites

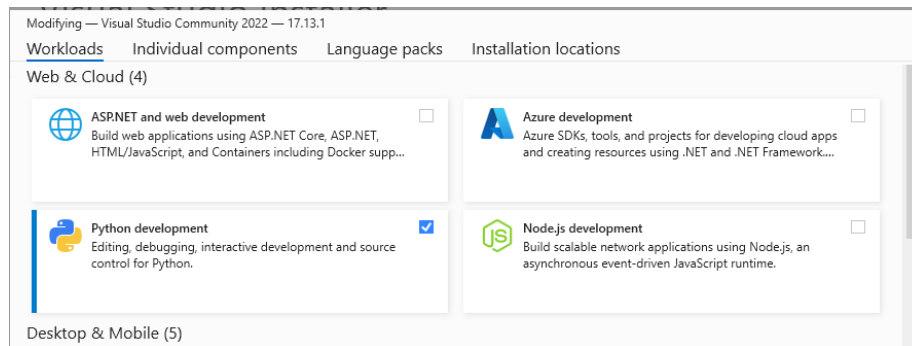
- Python 3.10 or higher
- Install Pytorch, please visit <https://pytorch.org/get-started/locally/>
- Install Ollama: <https://ollama.com/download> or OpenAI API (need to create .env file to include OPENAI_API_KEY = 'sk-xxx')
- Suggested to use conda for your env control to avoid environment conflicts:

Install conda for Windows:

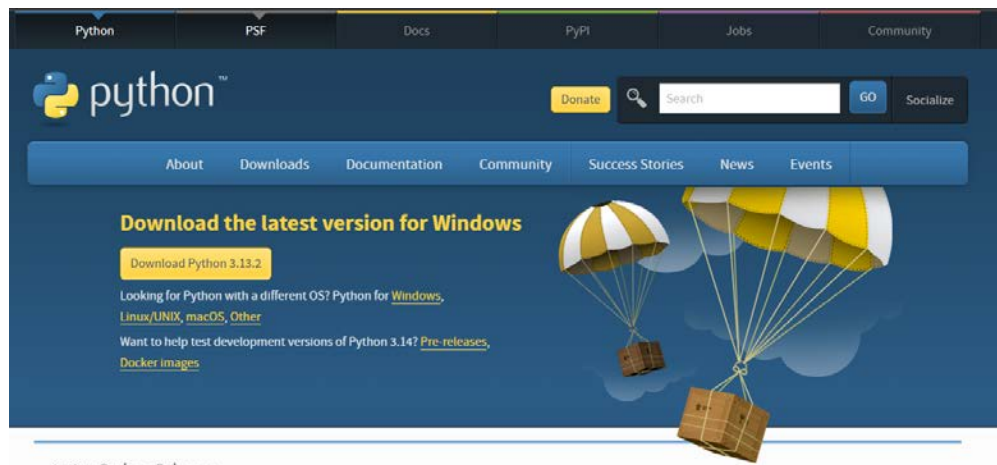
1. Download the Miniconda installer for Windows from <https://docs.conda.io/en/latest/miniconda.html>
2. Run the .exe file and follow the installation prompts
3. Choose whether to add Conda to your PATH environment variable during installation

a. Installing Python

1. Via [Visual Studio 2022 Community Installer](#):



2. Via [Python Downloads](#):



b. Installing Pytorch

1. Visit <https://pytorch.org/get-started/locally/>
2. Select the platform details (I am using Windows 10):

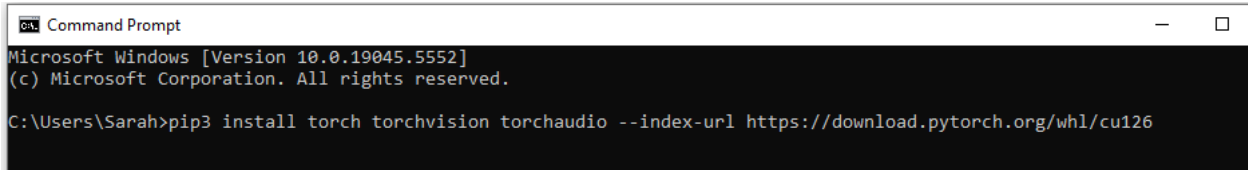
START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

NOTE: Latest PyTorch requires Python 3.9 or later.

PyTorch Build	Stable (2.6.0)		Preview (nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.4	CUDA 12.6	ROCm-6.2.4 CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126</pre>			

3. Open command prompt (as administrator) and run the custom command:

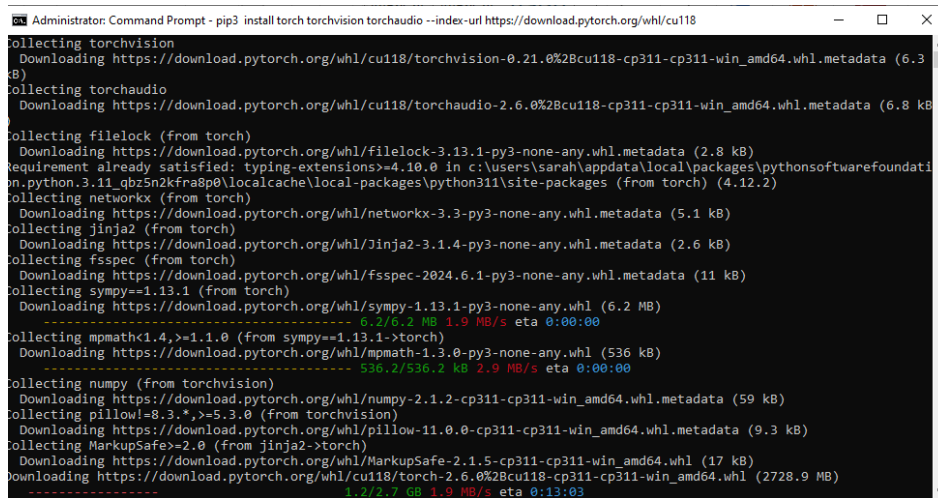


```

Command Prompt
Microsoft Windows [Version 10.0.19045.5552]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sarah>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126
  
```

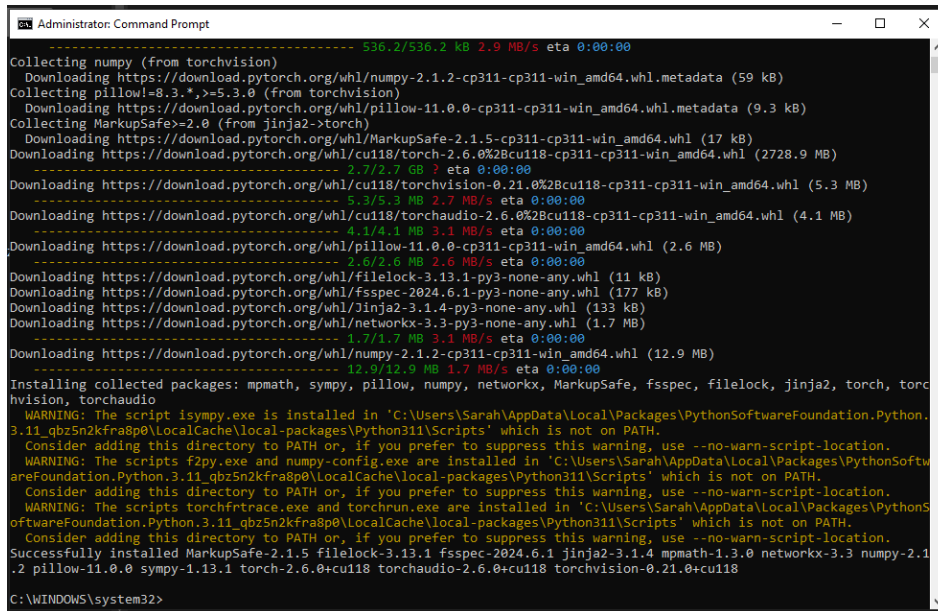
4. Wait for Installation to be completed:



```

Administrator: Command Prompt - pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
Collecting torchvision
  Downloading https://download.pytorch.org/whl/cu118/torchvision-0.21.0%2Bcu118-cp311-cp311-win_amd64.whl.metadata (6.3 kB)
Collecting torchaudio
  Downloading https://download.pytorch.org/whl/cu118/torchaudio-2.6.0%2Bcu118-cp311-cp311-win_amd64.whl.metadata (6.8 kB)
Collecting filelock (from torch)
  Downloading https://download.pytorch.org/whl/filelock-3.13.1-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\users\sarah\appdata\local\packages\pythonsoftwarefoundati
on.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from torch) (4.12.2)
Collecting networkx (from torch)
  Downloading https://download.pytorch.org/whl/networkx-3.3-py3-none-any.whl.metadata (5.1 kB)
Collecting Jinja2 (from torch)
  Downloading https://download.pytorch.org/whl/Jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting fsspec (from torch)
  Downloading https://download.pytorch.org/whl/fsspec-2024.6.1-py3-none-any.whl.metadata (11 kB)
Collecting sympy==1.13.1 (from torch)
  Downloading https://download.pytorch.org/whl/sympy-1.13.1-py3-none-any.whl (6.2 MB)
----- 6.2/6.2 MB 1.9 MB/s eta 0:00:00
Collecting mpmath<1.4,>=1.1.0 (from sympy==1.13.1->torch)
  Downloading https://download.pytorch.org/whl/mpmath-1.3.0-py3-none-any.whl (536 kB)
----- 536.2/536.2 kB 2.9 MB/s eta 0:00:00
Collecting numpy (from torchvision)
  Downloading https://download.pytorch.org/whl/numpy-2.1.2-cp311-cp311-win_amd64.whl.metadata (59 kB)
Collecting pillow==8.3.*>=5.3.0 (from torchvision)
  Downloading https://download.pytorch.org/whl/pillow-11.0.0-cp311-cp311-win_amd64.whl.metadata (9.3 kB)
Collecting MarkupSafe>=2.0 (from Jinja2->torch)
  Downloading https://download.pytorch.org/whl/MarkupSafe-2.1.5-cp311-cp311-win_amd64.whl (17 kB)
  Downloading https://download.pytorch.org/whl/cu118/torch-2.6.0%2Bcu118-cp311-cp311-win_amd64.whl (2728.9 MB)
----- 1.7/2.7 GB 1.9 MB/s eta 0:13:03
  
```

5. Confirm Installation Completion:



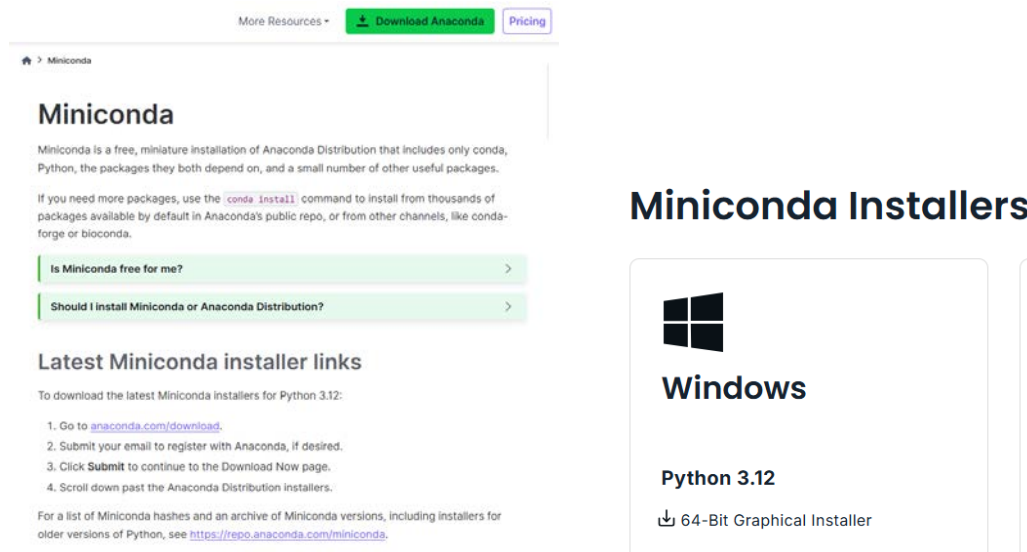
```

Administrator: Command Prompt
----- 536.2/536.2 kB 2.9 MB/s eta 0:00:00
Collecting numpy (from torchvision)
  Downloading https://download.pytorch.org/whl/numpy-2.1.2-cp311-cp311-win_amd64.whl.metadata (59 kB)
Collecting pillow==8.3.*>=5.3.0 (from torchvision)
  Downloading https://download.pytorch.org/whl/pillow-11.0.0-cp311-cp311-win_amd64.whl.metadata (9.3 kB)
Collecting MarkupSafe>=2.0 (from Jinja2->torch)
  Downloading https://download.pytorch.org/whl/MarkupSafe-2.1.5-cp311-cp311-win_amd64.whl (17 kB)
  Downloading https://download.pytorch.org/whl/cu118/torch-2.6.0%2Bcu118-cp311-cp311-win_amd64.whl (2728.9 MB)
----- 2.7/2.7 GB 1.9 MB/s eta 0:00:00
  Downloading https://download.pytorch.org/whl/cu118/torchvision-0.21.0%2Bcu118-cp311-cp311-win_amd64.whl (5.3 MB)
----- 5.3/5.3 MB 2.7 MB/s eta 0:00:00
  Downloading https://download.pytorch.org/whl/cu118/torchaudio-2.6.0%2Bcu118-cp311-cp311-win_amd64.whl (4.1 MB)
----- 4.1/4.1 MB 3.1 MB/s eta 0:00:00
  Downloading https://download.pytorch.org/whl/pillow-11.0.0-cp311-cp311-win_amd64.whl (2.6 MB)
----- 2.6/2.6 MB 2.6 MB/s eta 0:00:00
  Downloading https://download.pytorch.org/whl/filelock-3.13.1-py3-none-any.whl (11 kB)
  Downloading https://download.pytorch.org/whl/fsspec-2024.6.1-py3-none-any.whl (177 kB)
  Downloading https://download.pytorch.org/whl/Jinja2-3.1.4-py3-none-any.whl (133 kB)
  Downloading https://download.pytorch.org/whl/networkx-3.3-py3-none-any.whl (1.7 MB)
----- 1.7/1.7 MB 3.1 MB/s eta 0:00:00
  Downloading https://download.pytorch.org/whl/numpy-2.1.2-cp311-cp311-win_amd64.whl (12.9 MB)
----- 12.9/12.9 MB 1.7 MB/s eta 0:00:00
Installing collected packages: mpmath, sympy, pillow, numpy, networkx, MarkupSafe, fsspec, filelock, Jinja2, torch, torc
hvision, torchaudio
WARNING: The script isympy.exe is installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonSoftwareFoundation.Python.
3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonSoftw
areFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts torchfrtrace.exe and torchrun.exe are installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonS
oftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-2.1.5 filelock-3.13.1 fsspec-2024.6.1 Jinja2-3.1.4 mpmath-1.3.0 networkx-3.3 numpy-2.1
.2 pillow-11.0.0 sympy-1.13.1 torch-2.6.0+cu118 torchaudio-2.6.0+cu118 torchvision-0.21.0+cu118

C:\WINDOWS\system32>
  
```

c. Installing conda for OpenAI API env control

1. Download Miniconda via installers:



More Resources • [Download Anaconda](#) [Pricing](#)

Miniconda

Miniconda is a free, miniature installation of Anaconda Distribution that includes only conda, Python, the packages they both depend on, and a small number of other useful packages.

If you need more packages, use the `conda install` command to install from thousands of packages available by default in Anaconda's public repo, or from other channels, like conda-forge or bioconda.

[Is Miniconda free for me?](#)

[Should I install Miniconda or Anaconda Distribution?](#)

Latest Miniconda installer links

To download the latest Miniconda installers for Python 3.12:

1. Go to anaconda.com/download.
2. Submit your email to register with Anaconda, if desired.
3. Click **Submit** to continue to the Download Now page.
4. Scroll down past the Anaconda Distribution installers.

For a list of Miniconda hashes and an archive of Miniconda versions, including installers for older versions of Python, see <https://repo.anaconda.com/miniconda>.

Miniconda Installers

Windows

Python 3.12

📄 64-Bit Graphical Installer

2. Or via quick command line install:

Quick command line install

See our new section on the [Installing Miniconda](#) page!

Quickstart install instructions

These command line instructions will get you set up quickly with the latest Miniconda installer. Follow the steps for your system to download and install Miniconda, then follow the steps in **Verify your install** above to verify your Miniconda installation.

⚠ Caution

These quick install commands run a silent install. If you run a silent install, you are accepting Anaconda's Terms of Service (TOS) by default. Please make sure to review Anaconda's full TOS [here](#) before proceeding with silent installations.

Windows Command Prompt Windows PowerShell macOS Linux

These three commands quickly and quietly download the latest 64-bit Windows installer, rename it to a shorter file name, perform a silent install, and then delete the installer:

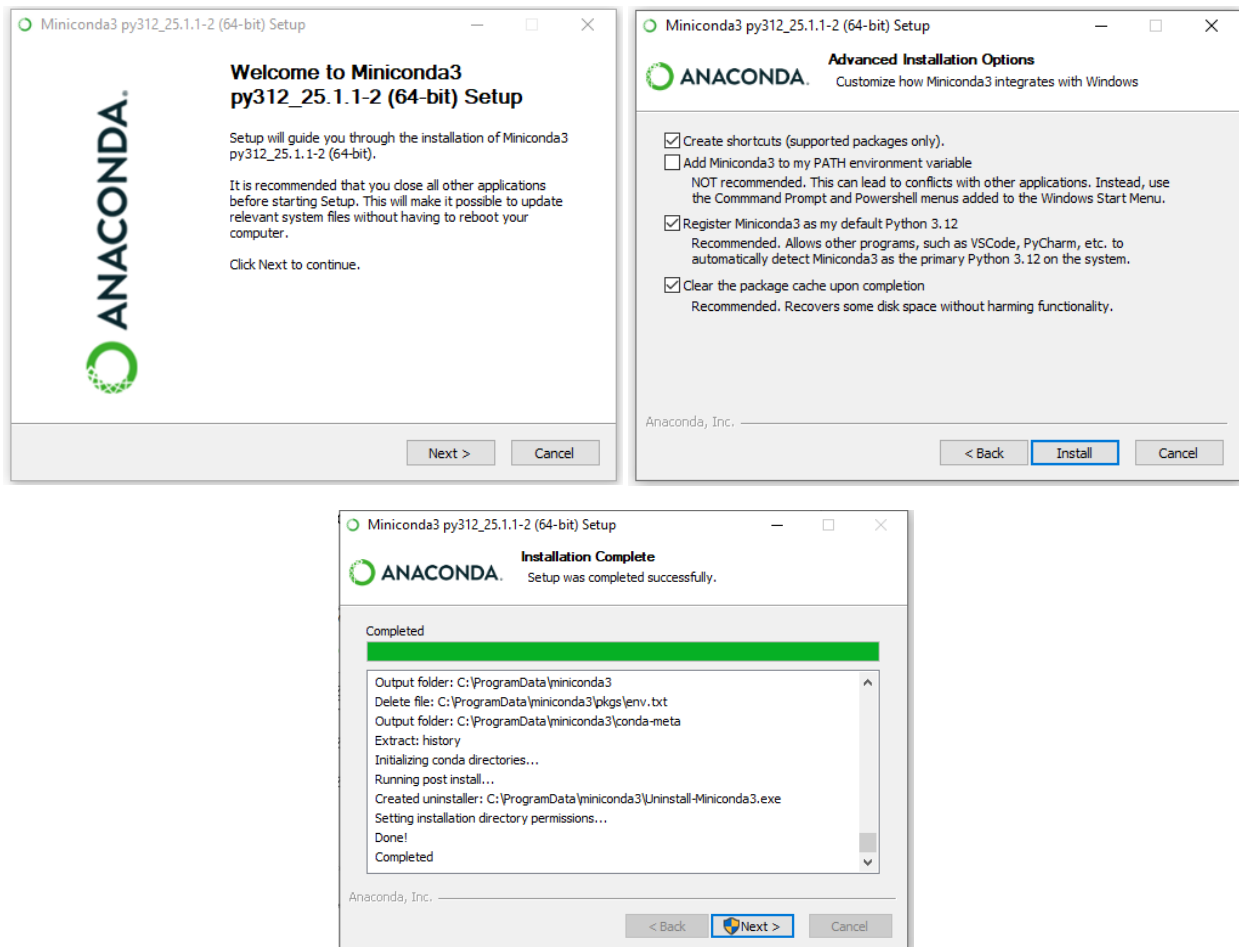
```
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o .\miniconda.exe
start /wait "" .\miniconda.exe /S
del .\miniconda.exe
```

To download an older version

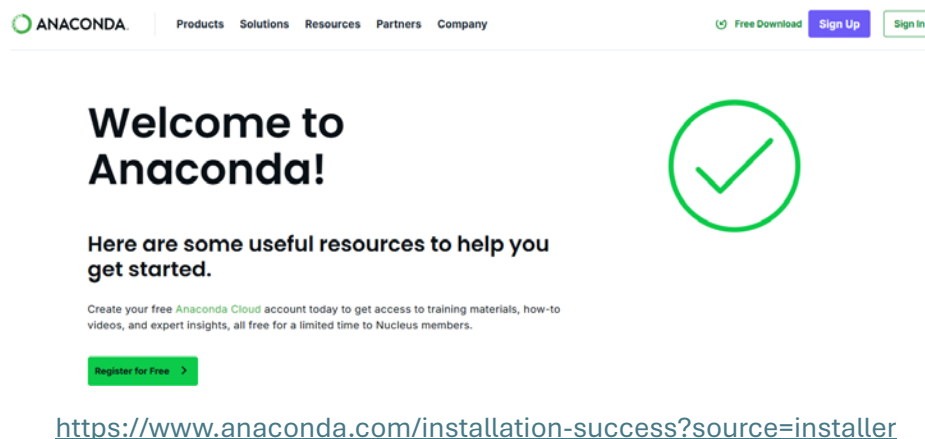
After installing, open the "Anaconda Prompt (miniconda3)" program to use Miniconda3.

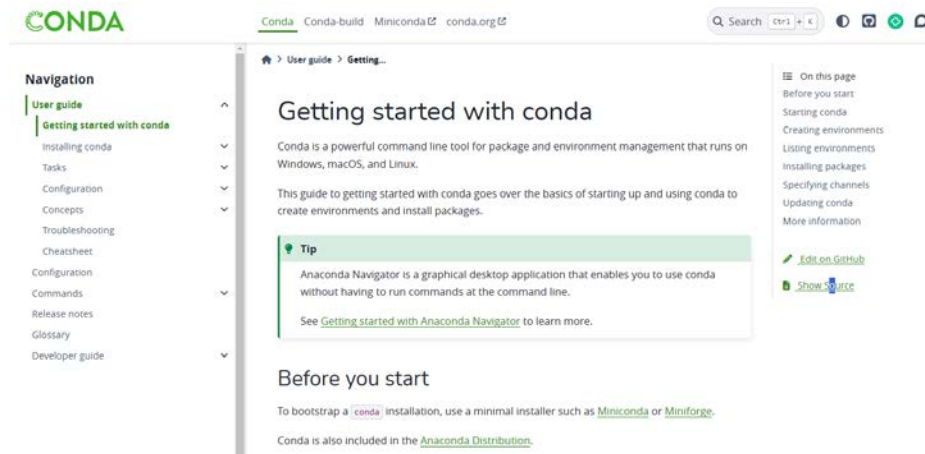
For windows command prompt: `curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe -o .\miniconda.exe`

3. Run Miniconda setup:



4. Click on resources (Optional):





<https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html>

d. Installation

1. Use conda to create env in your project folder:

```
conda create env --name <your_env_name> python=3.10
conda activate <your_env_name>
```

2. Then, install NexuSync under your conda env, run the following command:

```
pip install nexsync
```

Or `git clone https://github.com/Zakk-Yang/nexusync.git` (Covered Above)

3. Install pytorch (<https://pytorch.org/get-started/locally/>): (Covered Above)

- If you are using cuda, make sure your cuda version matches:
- For CUDA 11.8 (example, for windows and wsl2/linux)

```
pip3 install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu118
```

- For CUDA 12.1 (example, for windows and wsl2/linux)

```
pip3 install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cu121
```

- For macOS

```
pip3 install torch torchvision torchaudio
```

III- Install llama3.2

a. Pull llama3.2

1. Pull [llama3.2](#) using Ollama by running the following command prompt:

```
ollama run llama3.2
```

2. Wait for the installation to finish (2.0 GB for 3b version):

```
C:\WINDOWS\system32>ollama run llama3.2
pulling manifest
pulling dde5aa3fc5ff... 13% ▒ 255 MB/2.0 GB 1.3 MB/s 22m14s
```

3. Confirm Installation:

```
hvision, torchaudio
WARNING: The script isympy.exe is installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts torchfrtrace.exe and torchrun.exe are installed in 'C:\Users\Sarah\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-2.1.5 filelock-3.13.1 fsspec-2024.6.1 Jinja2-3.1.4 mpmath-1.3.0 networkx-3.3 numpy-2.1.2 pillow-11.0.0 sympy-1.13.1 torch-2.6.0+cu118 torchaudio-2.6.0+cu118 torchvision-0.21.0+cu118

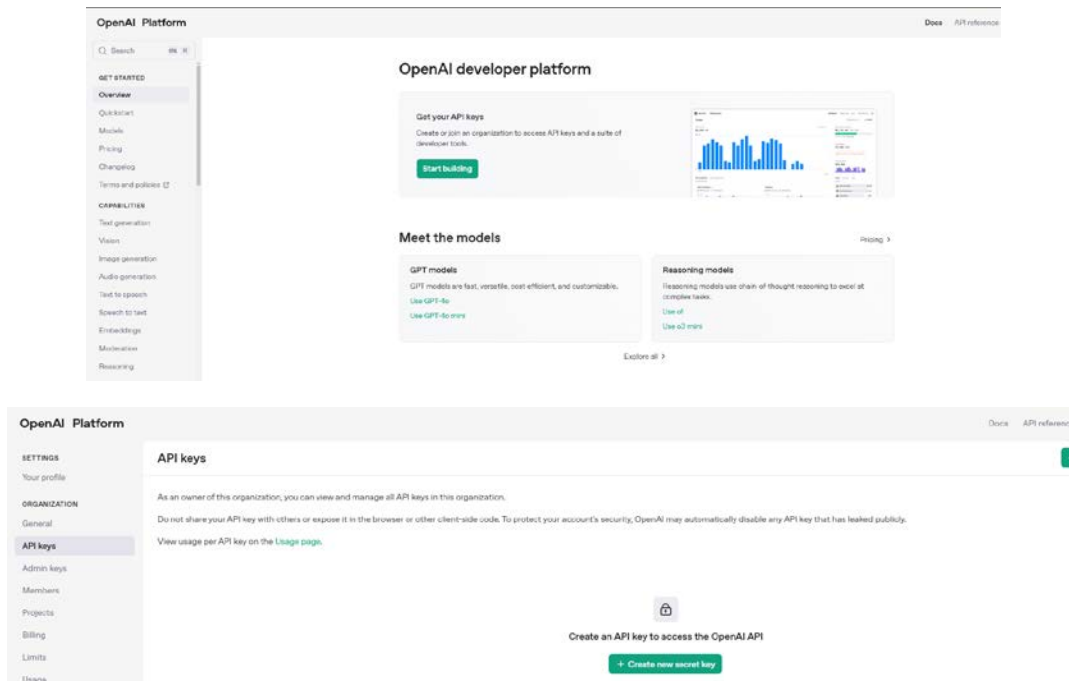
C:\WINDOWS\system32>
C:\WINDOWS\system32>E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag
'E:\VirtualFusion-ai\Ollama' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>from nexsync import NexuSync
'from' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>ollama run llama3.2
pulling manifest
pulling dde5aa3fc5ff... 100% ▒ 2.0 GB
pulling 966de95ca8a6... 100% ▒ 1.4 KB
pulling fcc5a6bec9da... 100% ▒ 7.7 KB
pulling a70ff7e570d9... 100% ▒ 6.0 KB
pulling 56bb8bd477a5... 100% ▒ 96 B
pulling 34bb5ab01051... 100% ▒ 561 B
verifying sha256 digest
writing manifest
success
>>> Send a message (/? for help)
```

IV- Generate OpenAI API KEY

1. Login to [OpenAI Platform](#) and go to [API KEYS](#):



2. Create Secret KEY

Create new secret key

Owned by

☐ You ☐ Service account

This API key is tied to your user and can make requests against the selected project. If you are removed from the organization or project, this key will be disabled.

Name Optional

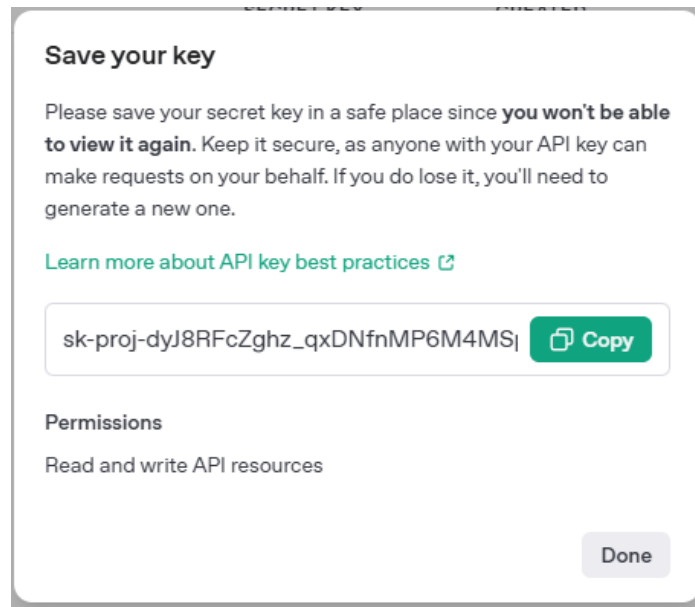
Project

Default project

Permissions

☐ All ☐ Restricted ☐ Read only

3. Save your secret key:



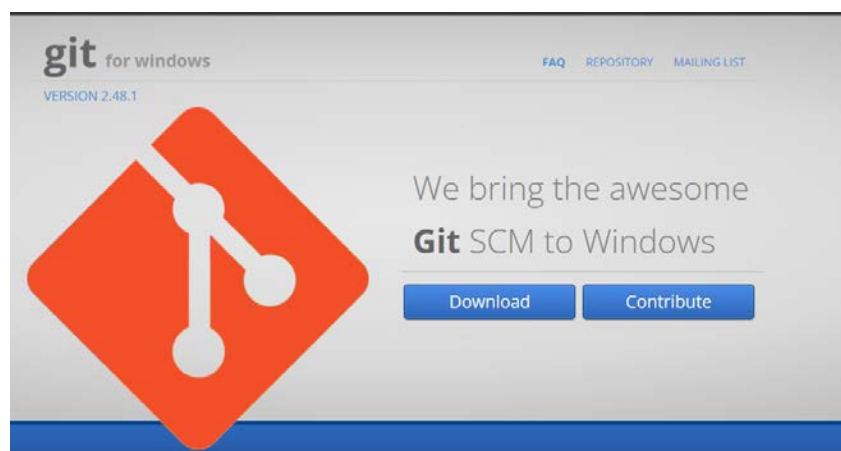
sk-proj-
dyJ8RFcZghz_qxDNfnMP6M4MSp1OgTE14oyJZ_HUj8DkqKVSj8EGIs1bIV9Sd3zxF369tfQuf0
T3BlbkFJGO25QWkGEN4kjRASyMACmREc3rUvdEhaWHjL_dTy3pWhwY4TVwW7HVkQSNig
EKkkQErW5SwPgA

4. D

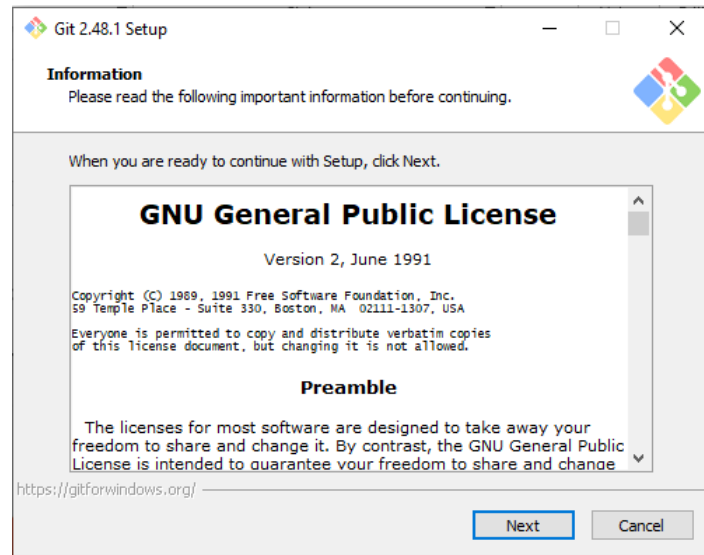
V- Set Up Your Environment

a. Install Git for Windows

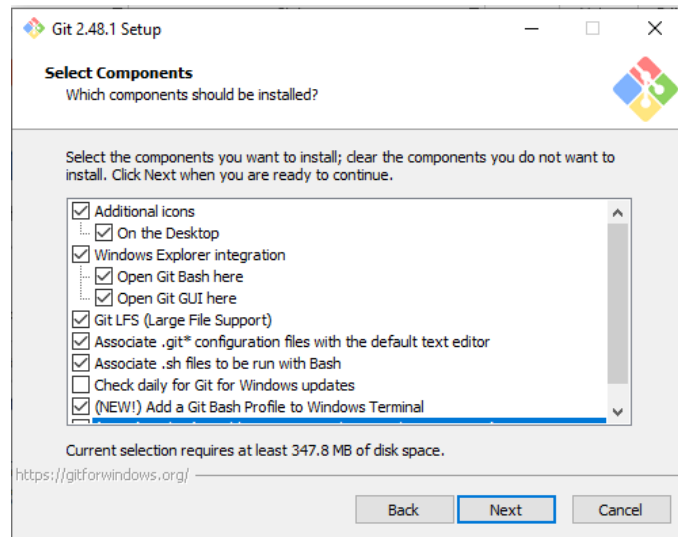
1. GO to <https://gitforwindows.org/> and click “Download”:



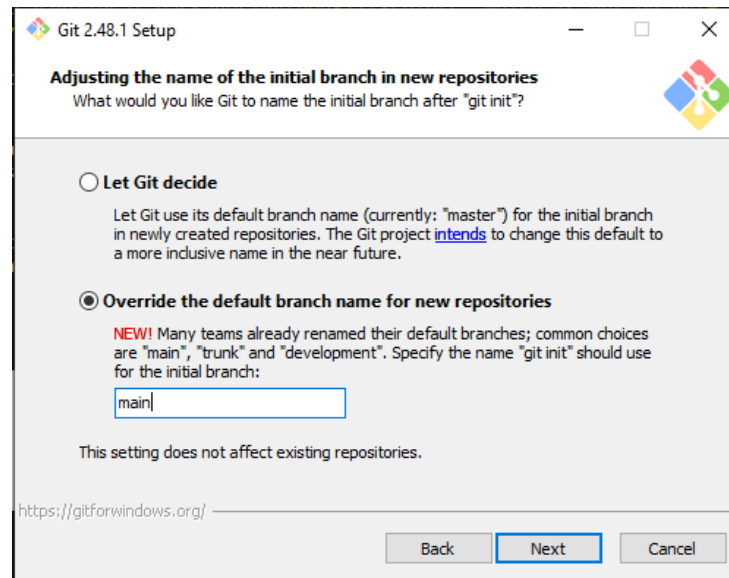
2. Run as Administrator and follow the setup steps:



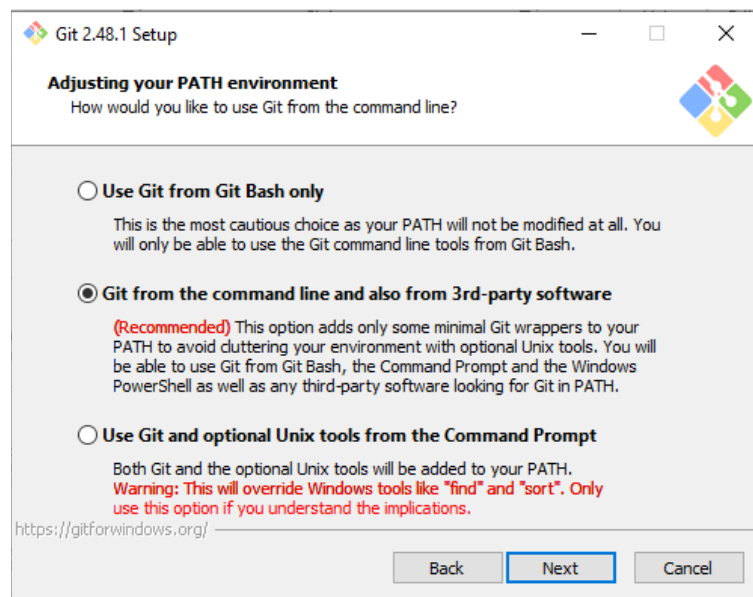
3. Install components:



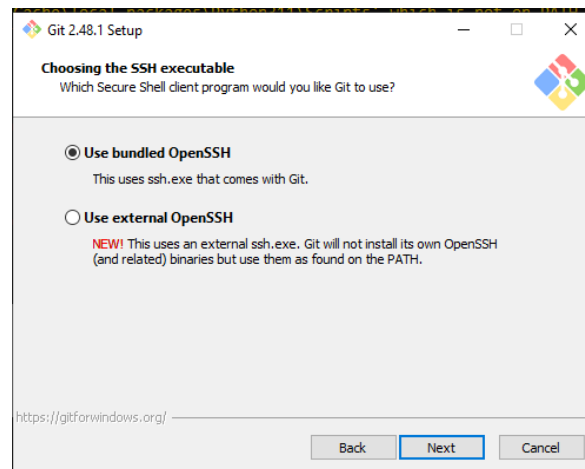
4. Select appropriate naming conventions:



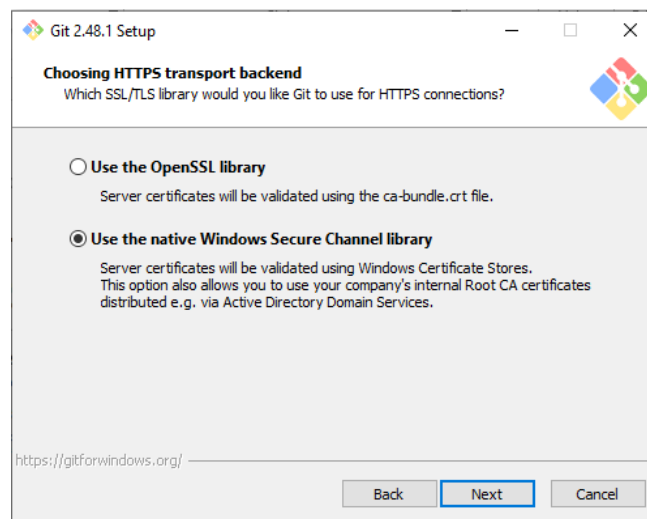
5. Use recommended setting for path env adjustments:



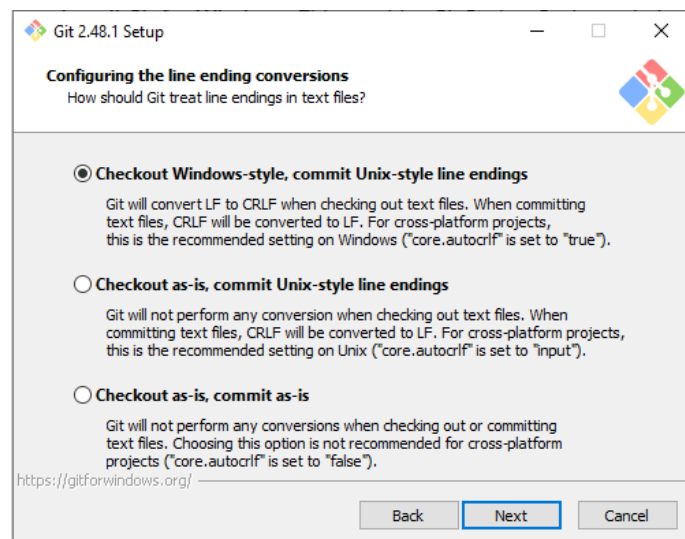
6. SSH executable:



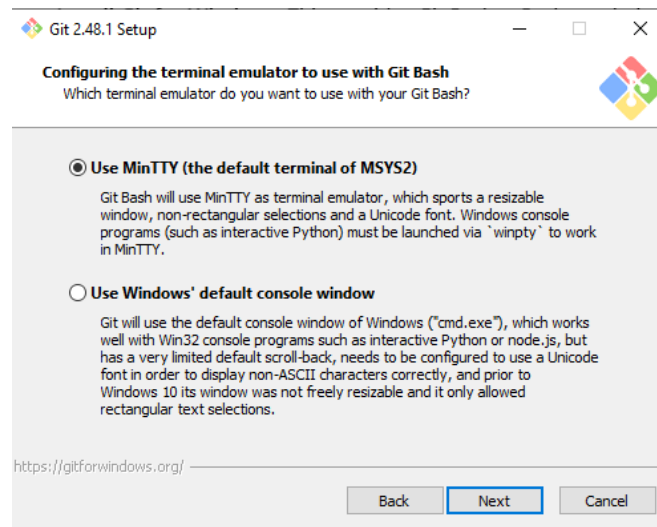
7. HTTPS transport backend:



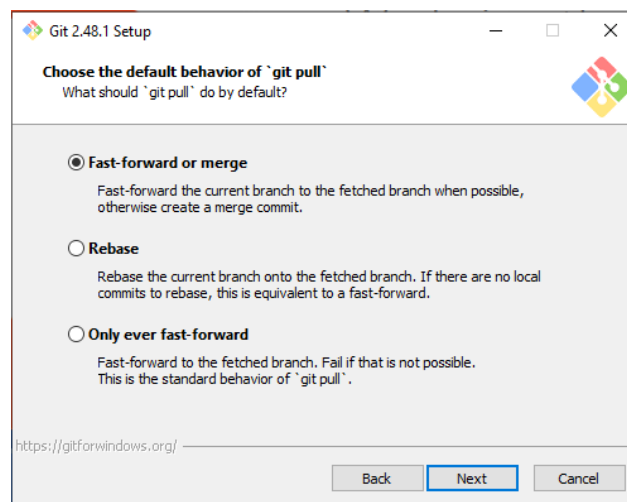
8. Recommending Settings (For Windows)



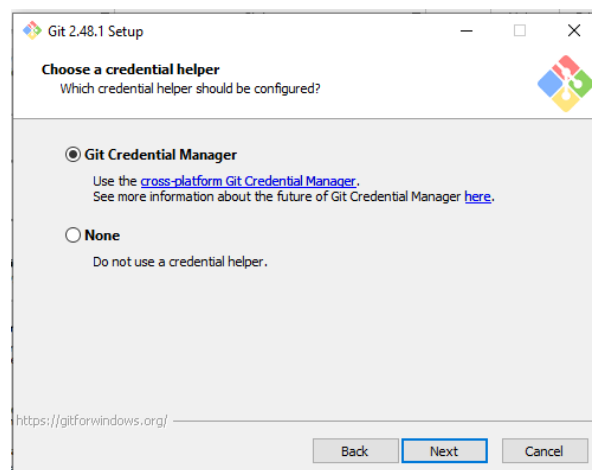
9. Configure terminal emulator:



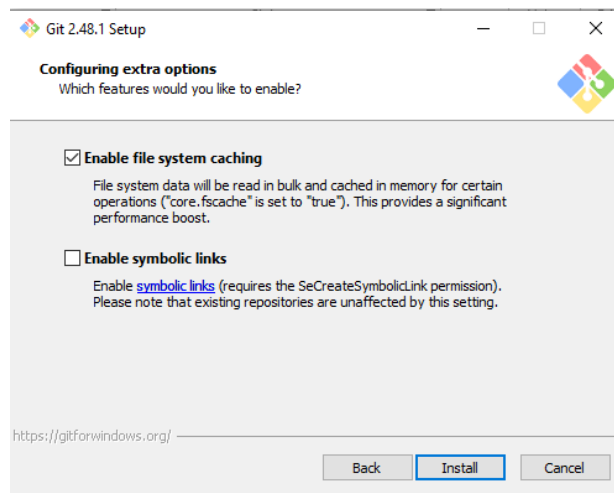
10. Default (merge) for git pull:



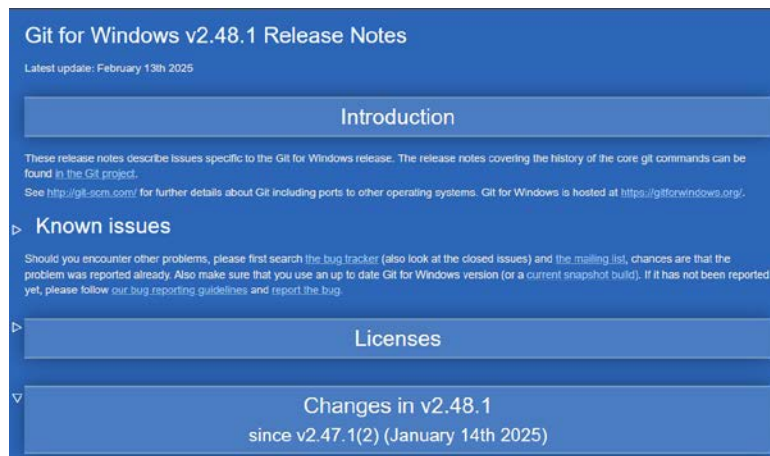
11. Select Git credential helper:



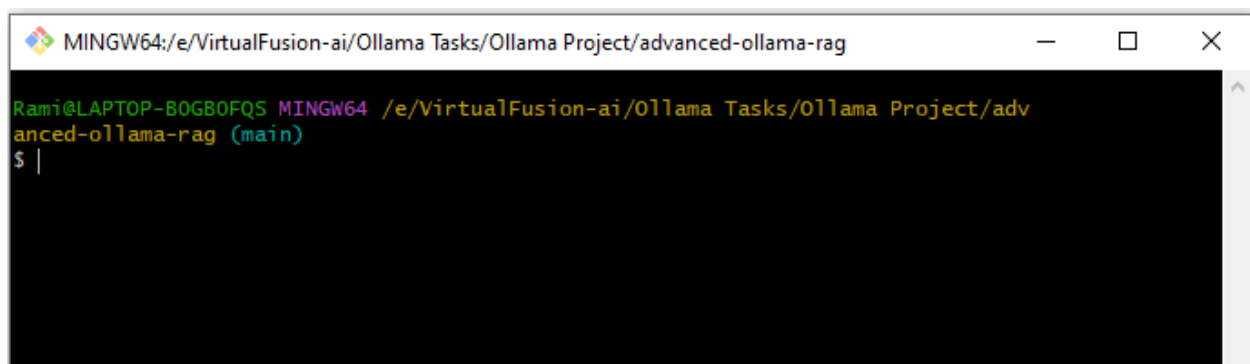
12. Default extensions, press “Install”:



13. View [Release Notes](#) (Optional):



14. Now you can run Bash scripts directly from within the project folder:



b. Running the Bash Scripts:

1. Navigate to the Script's Directory:

Use the `cd` command to change to the directory containing the script. For example:

```
cd path/to/advanced-ollama-rag
```

2. Executing the Scripts:

Run: `./script_name.sh`

Replace `./script_name.sh` with the actual script name.

3. Direct Execution (Alternative Method):

Right-click the script file in VS Code's Explorer pane and select `Run in Terminal`.

To open the terminal in Visual Studio, select View > Terminal.

c. Open the Terminal

Next, open your terminal. On Windows, you can use the Anaconda Prompt, while on macOS and Linux, you can use the terminal.

d. Use the `--prefix` or `-p` Option

[How to Create a New Environment Location for Conda Create: A Guide](#)

To specify a new environment location, use the `--prefix` or `-p` option followed by the path to the directory where you want to create the environment. The command should look like this:

```
conda create --prefix /path/to/directory
```

or

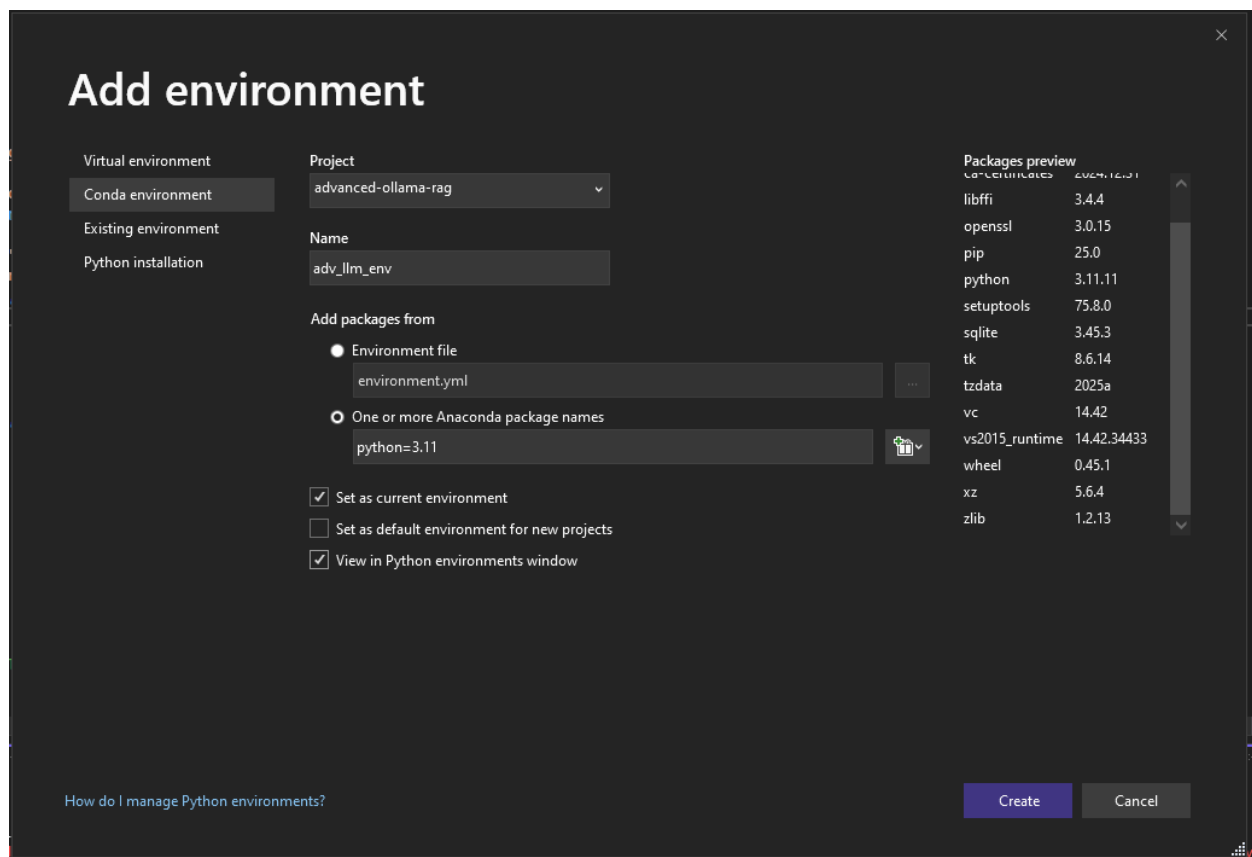
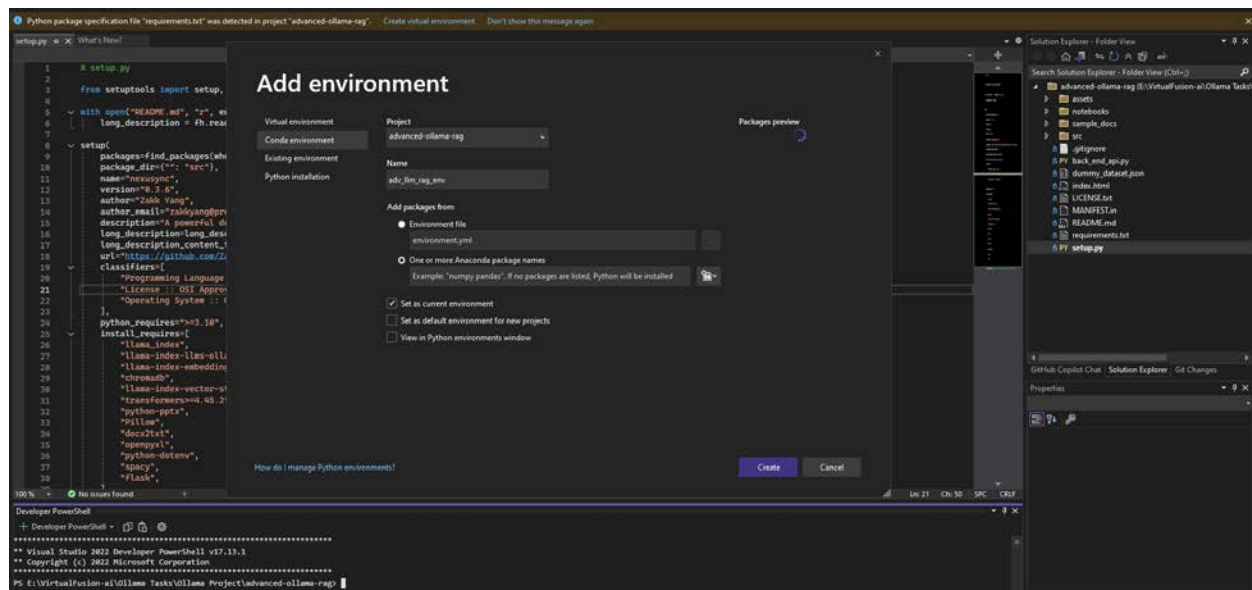
```
conda create -p /path/to/directory
```

Replace `/path/to/directory` with the actual path where you want to create the new environment.

```
conda create --prefix E:\VirtualFusion-ai\Ollama Tasks\Ollama  
Project\advanced-ollama-rag
```

e. Create Conda Environment within Visual Studio

1. Add environment:



2. Wait for environment setup:

```

Output
Show output from: General
--- Creating 'adv_llm_rag_env' ---
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done
## Package Plan ##
environment location: C:\Users\Sarah\.conda\envs\adv_llm_rag_env
added / updated specs:
- python

The following packages will be downloaded:

```

	build	
libmbedtls-4.0.0	h827c3e9_0	95 KB
python-25.0	py313ha095532_0	2.7 MB
python-3.13.2	hpython313_100_cp313	16.0 MB
python_abi-3.13	0_cp313	7 KB
setuptools-75.8.0	py313ha095532_0	2.2 MB
wheel-1.45.1	py313ha095532_0	1.0 MB
xx-5.6.4	h4754444_1	280 KB
	Total:	22.2 MB

```

The following NEW packages will be INSTALLED:
bz2p2
csc-certificates
expat
libid3
libmbedtls
openssl
pip
python
python_abi
setuptools
sqlite
pkgs/main/win-64::bz2p2-1.0.8-h2b0f81b_6
pkgs/main/win-64::csc-certificates-2024.12.31-ha095532_0
pkgs/main/win-64::expat-2.6.4-h8d8d27b_0
pkgs/main/win-64::libid3-3.4.4-hd7fa12b_1
pkgs/main/win-64::libmbedtls-4.0.0-h827c3e9_0
pkgs/main/win-64::openssl-3.0.13-h827c369_0
pkgs/main/win-64::python-25.0-py313ha095532_0
pkgs/main/win-64::python-3.13.2-hpython313_100_cp313
pkgs/main/win-64::python_abi-3.13.0_cp313
pkgs/main/win-64::setuptools-75.8.0-py313ha095532_0
pkgs/main/win-64::sqlite-3.45.3-h2e09f1f_0

```

3. Activate Environment:

[illegible]

To activate this environment, use

```
# $conda activate adv_llm_rag_env
```

#

To deactivate an active environment, use

#

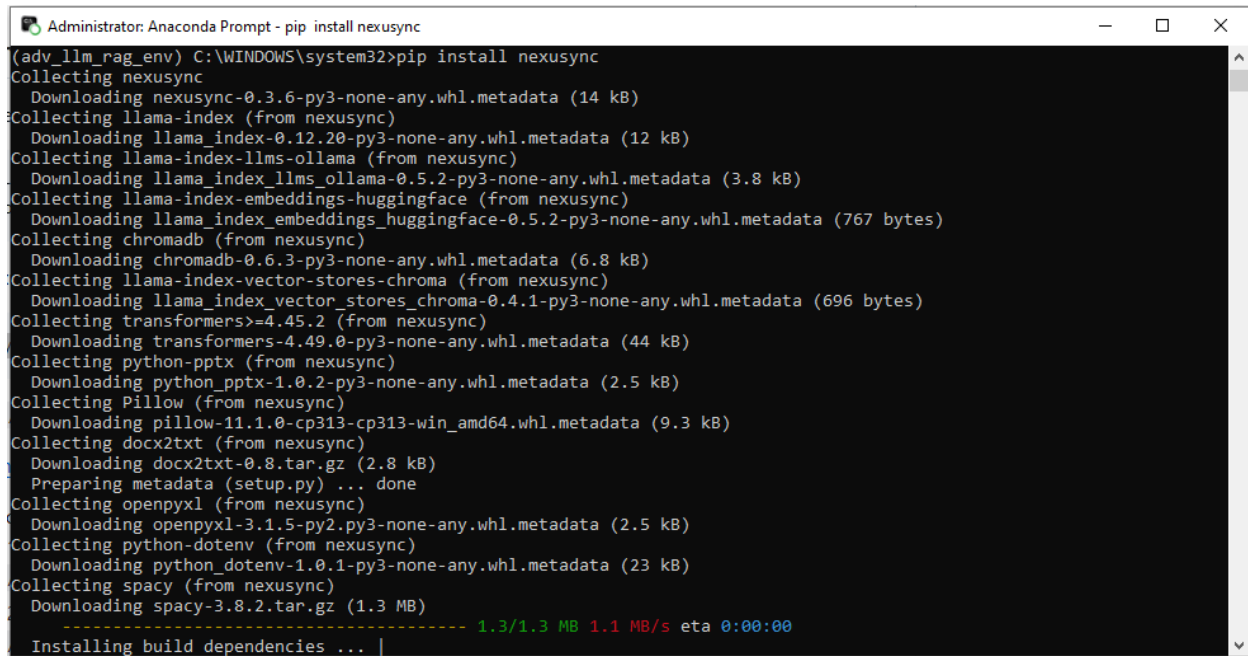
```
# $ conda deactivate
```

4. Activate Environment using Anaconda Prompt:

```
Administrator: Anaconda Prompt
(base) C:\WINDOWS\system32>conda activate adv_llm_rag_env
(adv_llm_rag_env) C:\WINDOWS\system32>
```

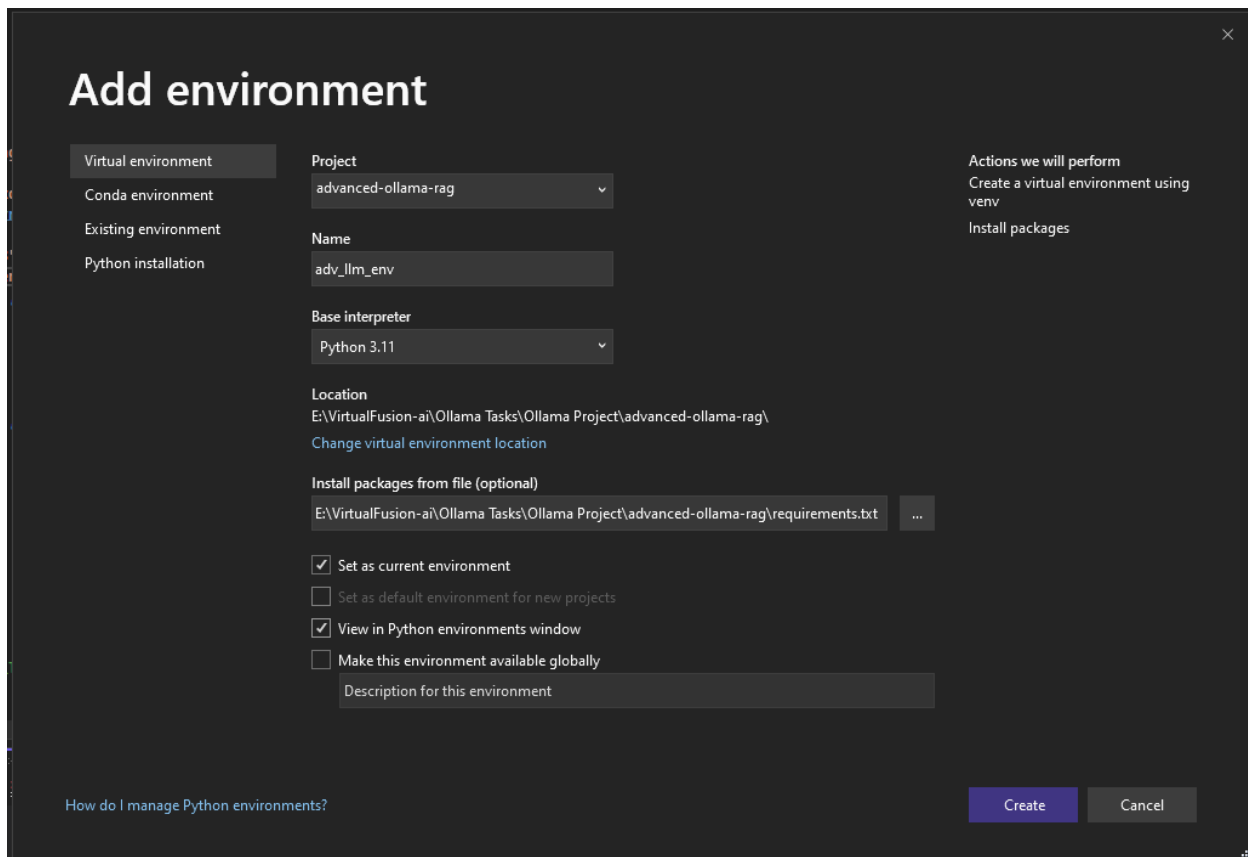
- Then, install NexuSync under your conda env, run the following command:

```
pip install nexusync
```



```
Administrator: Anaconda Prompt - pip install nexusync
(adv_llm_rag_env) C:\WINDOWS\system32>pip install nexusync
Collecting nexusync
  Downloading nexusync-0.3.6-py3-none-any.whl.metadata (14 kB)
Collecting llama-index (from nexusync)
  Downloading llama_index-0.12.20-py3-none-any.whl.metadata (12 kB)
Collecting llama-index-llms-ollama (from nexusync)
  Downloading llama_index_llms_ollama-0.5.2-py3-none-any.whl.metadata (3.8 kB)
Collecting llama-index-embeddings-huggingface (from nexusync)
  Downloading llama_index_embeddings_huggingface-0.5.2-py3-none-any.whl.metadata (767 bytes)
Collecting chromadb (from nexusync)
  Downloading chromadb-0.6.3-py3-none-any.whl.metadata (6.8 kB)
Collecting llama-index-vector-stores-chroma (from nexusync)
  Downloading llama_index_vector_stores_chroma-0.4.1-py3-none-any.whl.metadata (696 bytes)
Collecting transformers<=4.45.2 (from nexusync)
  Downloading transformers-4.49.0-py3-none-any.whl.metadata (44 kB)
Collecting python-pptx (from nexusync)
  Downloading python_pptx-1.0.2-py3-none-any.whl.metadata (2.5 kB)
Collecting Pillow (from nexusync)
  Downloading pillow-11.1.0-cp313-cp313-win_amd64.whl.metadata (9.3 kB)
Collecting docx2txt (from nexusync)
  Downloading docx2txt-0.8.tar.gz (2.8 kB)
  Preparing metadata (setup.py) ... done
Collecting openpyxl (from nexusync)
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting python-dotenv (from nexusync)
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting spacy (from nexusync)
  Downloading spacy-3.8.2.tar.gz (1.3 MB)
  ----- 1.3/1.3 MB 1.1 MB/s eta 0:00:00
Installing build dependencies ... |
```

6. Add Virtual Environment, installing packages from “requirements.txt”:



Add environment

Virtual environment

Conda environment

Existing environment

Python installation

Project: advanced-ollama-rag

Name: adv_llm_env

Base interpreter: Python 3.11

Location: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\

Change virtual environment location

Install packages from file (optional): E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\requirements.txt

☒ Set as current environment

☐ Set as default environment for new projects

☒ View in Python environments window

☐ Make this environment available globally

Description for this environment

How do I manage Python environments?

Create Cancel

[Python venv: How To Create, Activate, Deactivate, And Delete](#)

7. Install PyTorch

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

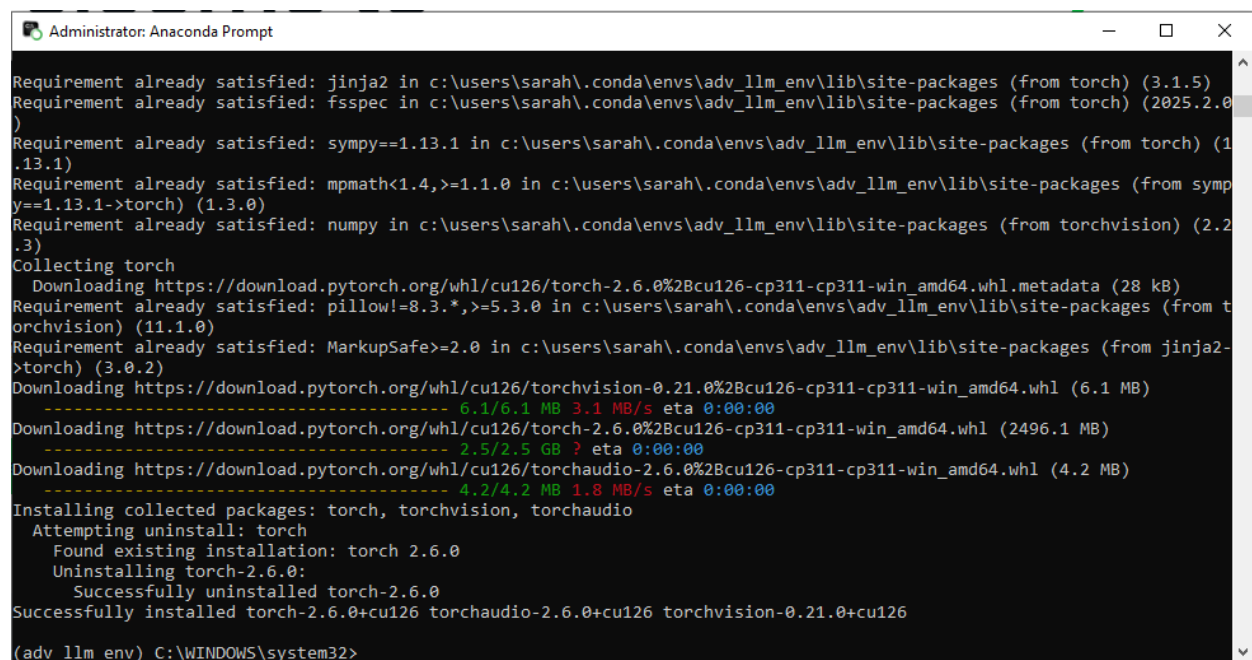
NOTE: Latest PyTorch requires Python 3.9 or later.

PyTorch Build	Stable (2.6.0)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.4	CUDA 12.6	ROCm 6.2.4
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126</pre>			

Run this command (Windows):

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126
```

Confirm PyTorch Installation:



```

Administrator: Anaconda Prompt

Requirement already satisfied: Jinja2 in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from torch) (3.1.5)
Requirement already satisfied: fsspec in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from torch) (2025.2.0)
Requirement already satisfied: sympy==1.13.1 in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from torchvision) (2.2.3)
Collecting torch
  Downloading https://download.pytorch.org/whl/cu126/torch-2.6.0%2Bcu126-cp311-cp311-win_amd64.whl.metadata (28 kB)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from torchvision) (11.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\sarah\.conda\envs\adv_llm_env\lib\site-packages (from Jinja2->torch) (3.0.2)
Downloading https://download.pytorch.org/whl/cu126/torchvision-0.21.0%2Bcu126-cp311-cp311-win_amd64.whl (6.1 MB)
----- 6.1/6.1 MB 3.1 MB/s eta 0:00:00
Downloading https://download.pytorch.org/whl/cu126/torch-2.6.0%2Bcu126-cp311-cp311-win_amd64.whl (2496.1 MB)
----- 2.5/2.5 GB ? eta 0:00:00
Downloading https://download.pytorch.org/whl/cu126/torchaudio-2.6.0%2Bcu126-cp311-cp311-win_amd64.whl (4.2 MB)
----- 4.2/4.2 MB 1.8 MB/s eta 0:00:00
Installing collected packages: torch, torchvision, torchaudio
  Attempting uninstall: torch
    Found existing installation: torch 2.6.0
    Uninstalling torch-2.6.0:
      Successfully uninstalled torch-2.6.0
Successfully installed torch-2.6.0+cu126 torchaudio-2.6.0+cu126 torchvision-0.21.0+cu126

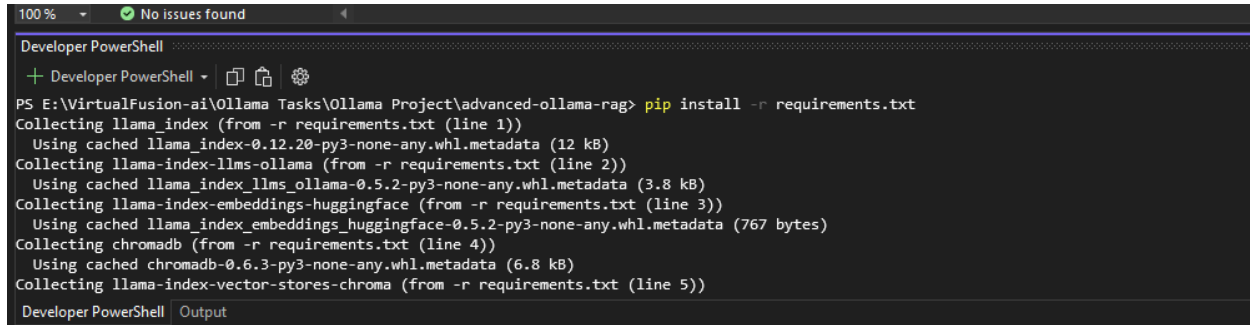
(adv_llm_env) C:\WINDOWS\system32>

```

8. Reinstall Dependencies:

With your environment activated, reinstall your project's dependencies:

```
pip install -r requirements.txt
```

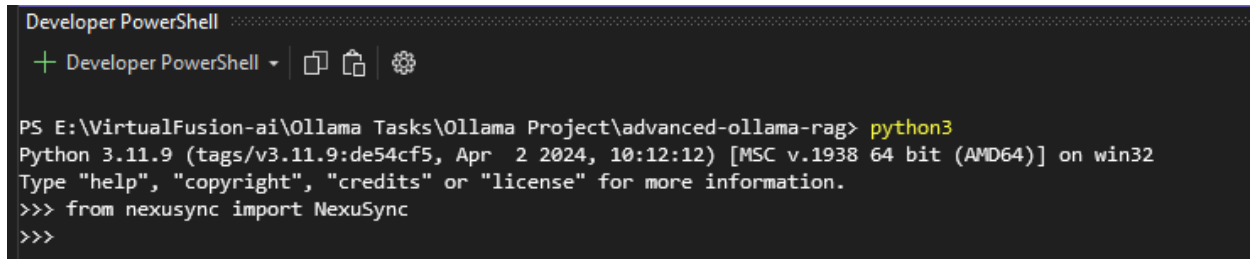


```
Developer PowerShell
+ Developer PowerShell
PS E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag> pip install -r requirements.txt
Collecting llama_index (from -r requirements.txt (line 1))
  Using cached llama_index-0.12.20-py3-none-any.whl.metadata (12 kB)
Collecting llama-index-llms-ollama (from -r requirements.txt (line 2))
  Using cached llama_index_llms_ollama-0.5.2-py3-none-any.whl.metadata (3.8 kB)
Collecting llama-index-embeddings-huggingface (from -r requirements.txt (line 3))
  Using cached llama_index_embeddings_huggingface-0.5.2-py3-none-any.whl.metadata (767 bytes)
Collecting chromadb (from -r requirements.txt (line 4))
  Using cached chromadb-0.6.3-py3-none-any.whl.metadata (6.8 kB)
Collecting llama-index-vector-stores-chroma (from -r requirements.txt (line 5))
Developer PowerShell  Output
```

VI- Quick Start

a. Import NexusSync

```
from nexusync import NexusSync
```



```
Developer PowerShell
+ Developer PowerShell
PS E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag> python3
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from nexusync import NexusSync
>>>
```

Note: If you're stuck in the Python interactive shell (the >>> prompt), you can exit in a few ways:

If you're stuck in the **Python interactive shell** (the >>> prompt), you can exit in a few ways:

1. Using the `exit()` function (clean exit):

```
exit()
```

2. Or `quit()` (does the same thing):

```
quit()
```

3. Keyboard shortcut (force exit):

- **Windows/Linux:** Ctrl + Z then press Enter
- **Mac:** Ctrl + D

If you accidentally ran a Python command in the terminal (like python or python3), these will bring you back to the regular terminal prompt.

b. Run LLM model (llama3.2)

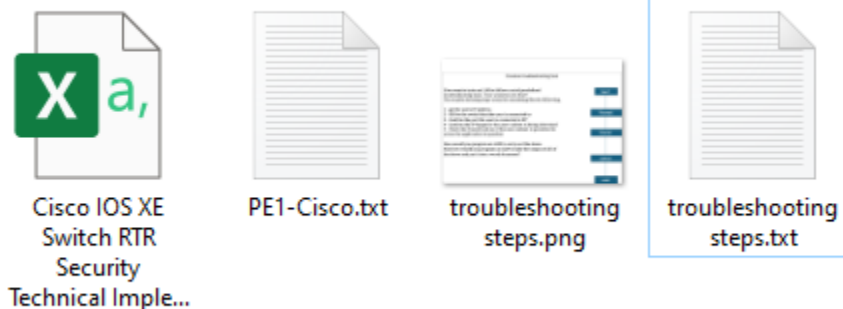
```
Developer PowerShell
+ Developer PowerShell | [ ] [ ] [ ]
ollama run llama3
^
SyntaxError: invalid syntax
>>> exit()
PS E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag> ollama run llama3.2
>>> Send a message (/? for help)
```

c. Comment OpenAI model for now and add Ollama Model

```
27 # For openai model: need to create .env in the src folder to include OPENAI_API_KEY = 'sk-xxx'
28 # comment for changing to llama model
29
30 ###
31 ### OPENAI_MODEL_YN = True
32 ### EMBEDDING_MODEL = "text-embedding-3-large"
33 ### LANGUAGE_MODEL = "gpt-4o-mini"
34 ### TEMPERATURE = 0.4
35 ### INPUT_DIRS = [os.path.abspath(directory_path)] # Can include multiple paths
36 ### CHROMA_DB_DIR = "chroma_db"
37 ### INDEX_PERSIST_DIR = "index_storage"
38 ### CHROMA_COLLECTION_NAME = "my_collection"
39 ### CHUNK_SIZE = 1024
40 ### CHUNK_OVERLAP = 20
41 ### RECURSIVE = True
42
43 #----- Use Ollama Model -----
44 # Customize your parameters for ollama model
45 OPENAI_MODEL_YN = False # if False, you will use ollama model
46 EMBEDDING_MODEL = "BAAI/bge-base-en-v1.5" # suggested embedding model, you can replace with any HuggingFace embedding models
47 LANGUAGE_MODEL = 'llama3.2' # you need to download ollama model first, please check https://ollama.com/download
48 BASE_URL = "http://localhost:11434" # you can switch to different base_url for Ollama model
49 TEMPERATURE = 0.4 # range from 0 to 1, higher means higher creativity level
50 CHROMA_DB_DIR = 'chroma_db' # Your path to the chroma db
51 INDEX_PERSIST_DIR = 'index_storage' # Your path to the index storage
52 CHROMA_COLLECTION_NAME = 'my_collection'
53 INPUT_DIRS = [os.path.abspath(directory_path)] # can specify multiple document paths
54 CHUNK_SIZE = 1024 # Size of text chunks for creating embeddings
55 CHUNK_OVERLAP = 20 # Overlap between text chunks to maintain context
56 RECURSIVE = True # Recursive or not under one folder
```

d. Updated Sample Docs with Task Files (Q1 & Q2)

> Ollama Tasks > Ollama Project > advanced-ollama-rag > sample_docs



e. Run LLM model (llama3.2)

Fixed some errors in the “back_end_api.py” code, then ran the environment:

```
2025-02-26 11:08:48,760 - nexusync.utils.embedding_models.set_language_model - INFO - Ollama LLM initialized with model: llama3.2 and base_url: http://localhost:11434
2025-02-26 11:08:49,397 - nexusync.NexusSync - INFO - Vectors and Querier initialized successfully.
DEBUG:llama_index.core.storage.kvstore.simple_kvstore:Loading llama_index.core.storage.kvstore.simple_kvstore from index_storage\docstore.json.
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\docstore.json
DEBUG:llama_index.core.storage.kvstore.simple_kvstore:Loading llama_index.core.storage.kvstore.simple_kvstore from index_storage\index_store.json.
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\index_store.json
DEBUG:llama_index.core.graph_stores.simple:Loading llama_index.core.graph_stores.simple from index_storage\graph_store.json.
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\graph_store.json
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\property_graph_store.json
DEBUG:llama_index.core.vector_stores.simple:Loading llama_index.core.vector_stores.simple from index_storage\default_vector_store.json.
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\default_vector_store.json
DEBUG:llama_index.core.vector_stores.simple:Loading llama_index.core.vector_stores.simple from index_storage\image_vector_store.json.
DEBUG:fsspec.local:open file: E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag\index_storage\image_vector_store.json
INFO:llama_index.core.indices.loading:Loading all indices.
2025-02-26 11:08:53,740 - nexusync.core.indexer - INFO - Index already built. Loading from disk.
2025-02-26 11:08:53,747 - nexusync.core.chat_engine - INFO - Chat engine initialized
* Serving Flask app 'back_end_api'
* Debug mode: on
INFO:werkzeug:[31m[!mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.+[0m
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:2024
* Running on http://192.168.1.3:2024
```

Default Parameters:

2025-02-26 11:08:48,755 - nexusync.utils.embedding_models.set_embedding_model - INFO -
Using HuggingFace embedding model: BAAI/bge-base-en-v1.5

2025-02-26 11:08:48,760 - nexusync.utils.embedding_models.set_language_model - INFO -
Ollama LLM initialized with model: llama3.2 and base_url: <http://localhost:11434>

* Running on all addresses (0.0.0.0)

* Running on http://127.0.0.1:2024

* Running on http://192.168.1.3:2024

1. Running on http://127.0.0.1:2024, the Web Interface is simple and has adaptable settings:

← → ↻ ⓘ 127.0.0.1:2024

📧 | 📧 Inbox (566) - s-romi... 📧 Yahoo Mail

Settings

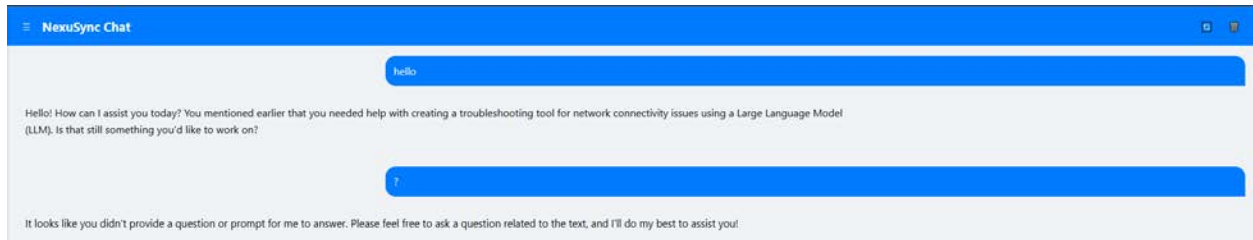
Embedding Model:

Language Model:

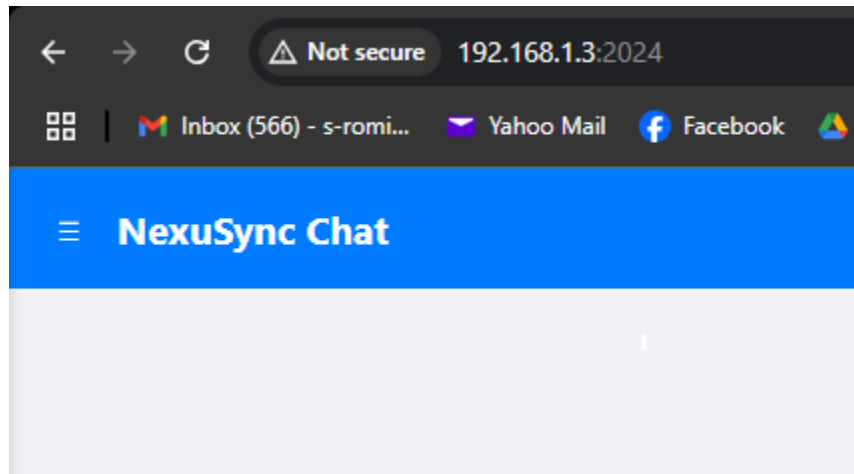
Temperature:

Input Directories:

Apply Settings



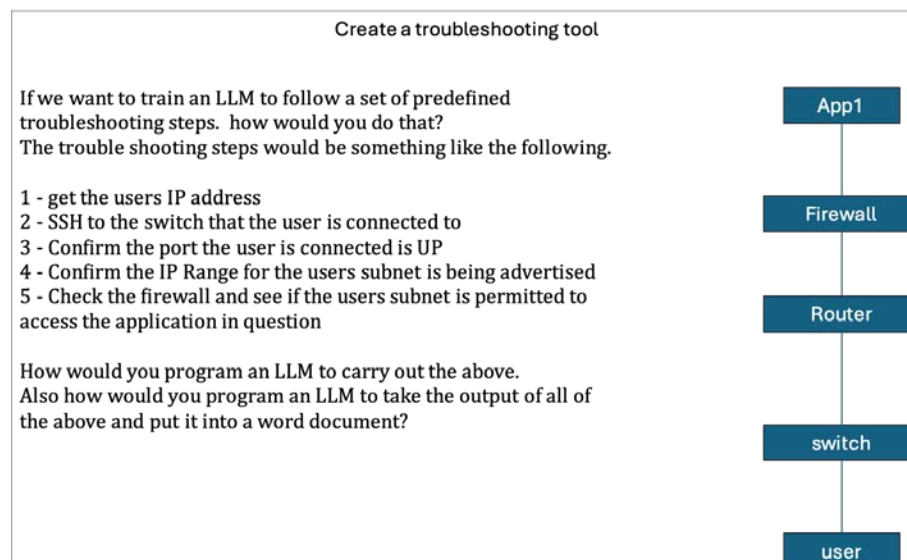
2. Second Domain (<http://192.168.1.3:2024>) is also running:



VII. Troubleshooting

- a. Let's try solving both questions simultaneously in each window:

1. Q1 Information given (Also fed to the LLM RAG pipeline):



2. Created a text file with Gemini to feed the Pipeline as well (it reads .png and .txt):

“troubleshooting steps.txt”:

I need you to create a troubleshooting tool based on a predefined set of steps. This tool will be designed to help diagnose network connectivity issues for a user trying to access an application (App1). The steps are as follows:

Get the user's IP address.

SSH to the switch that the user is connected to.

Confirm the port the user is connected to is UP.

Confirm the IP range for the user's subnet is being advertised by the router.

Check the firewall and see if the user's subnet is permitted to access the application (App1).

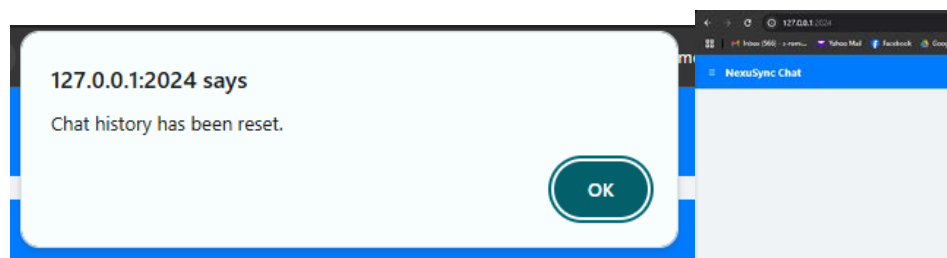
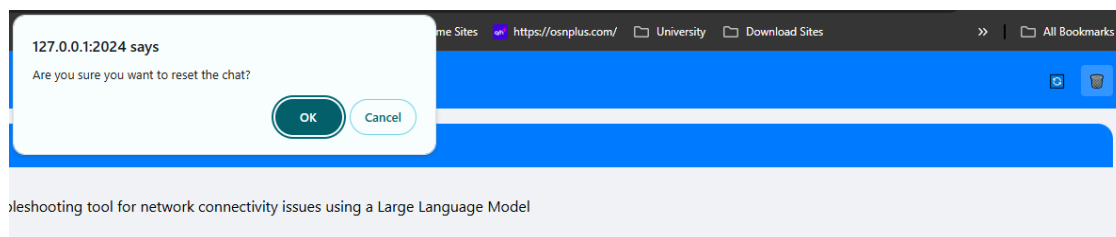
Please provide the following:

A conceptual outline or code snippets demonstrating how you would program a Large Language Model (LLM) to carry out these steps. Consider how you would structure the input, the logic flow, and the necessary interactions with external systems (like network devices).

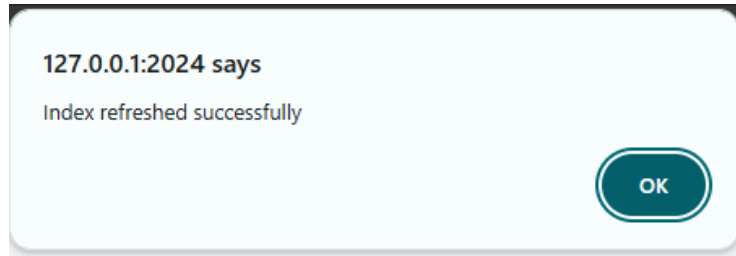
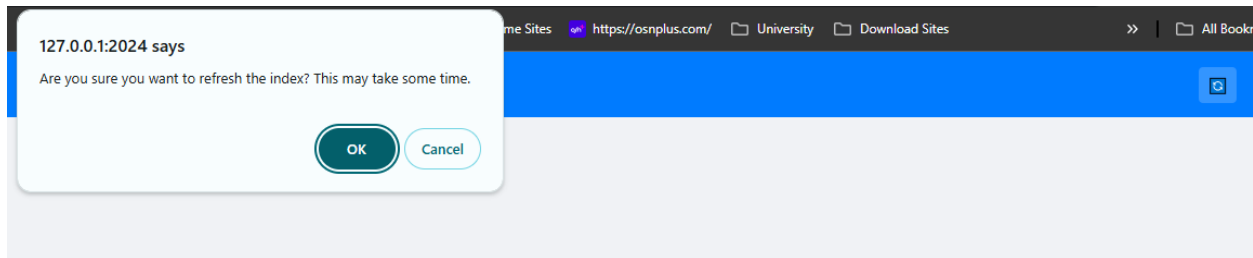
A plan for how you would program the LLM to take the output of each step and compile it into a structured Word document. This document should clearly present the results of each troubleshooting step and any relevant findings.

The goal is to automate the troubleshooting process as much as possible using the LLM. Please provide a detailed response that covers both the execution of the troubleshooting steps and the report generation.

3. Trying the Reset App Function:

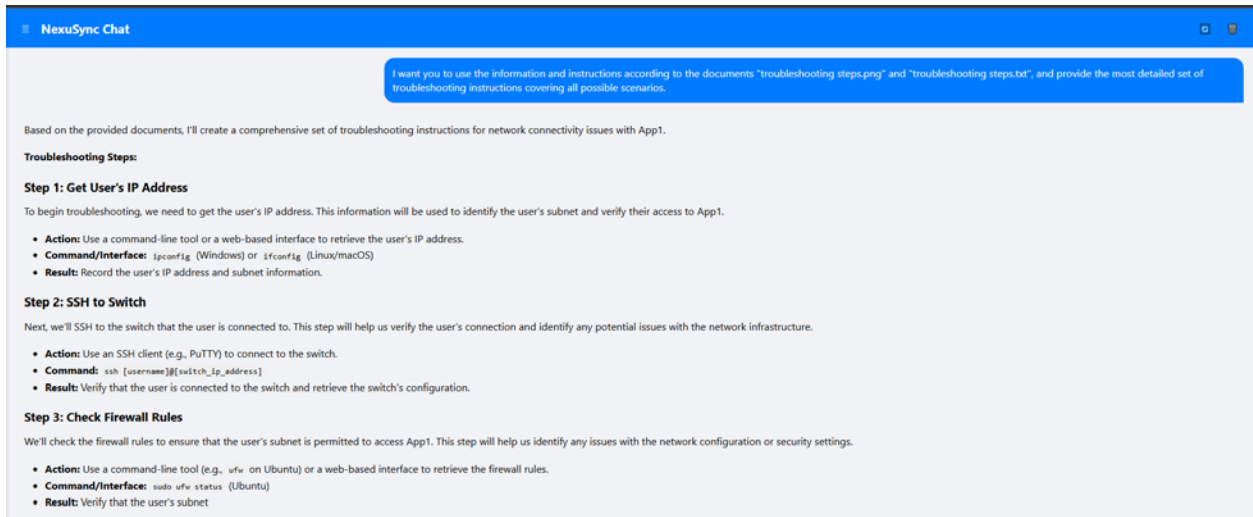


4. Refresh Index Function, to reload Sample Documents



5. My Prompt for Q1:

I want you to use the information and instructions according to the documents "troubleshooting steps.png" and "troubleshooting steps.txt", and provide the most detailed set of troubleshooting instructions covering all possible scenarios.



6. Full answer using llama 3.2 model and given settings (above):

Based on the provided documents, I'll create a comprehensive set of troubleshooting instructions for network connectivity issues with App1.

Troubleshooting Steps:

Step 1: Get User's IP Address

To begin troubleshooting, we need to get the user's IP address. This information will be used to identify the user's subnet and verify their access to App1.

- **Action:** Use a command-line tool or a web-based interface to retrieve the user's IP address.
- **Command/Interface:** ipconfig (Windows) or ifconfig (Linux/macOS)
- **Result:** Record the user's IP address and subnet information.

Step 2: SSH to Switch

Next, we'll SSH to the switch that the user is connected to. This step will help us verify the user's connection and identify any potential issues with the network infrastructure.

- **Action:** Use an SSH client (e.g., PuTTY) to connect to the switch.
- **Command:** ssh [username]@[switch_ip_address]
- **Result:** Verify that the user is connected to the switch and retrieve the switch's configuration.

Step 3: Check Firewall Rules

We'll check the firewall rules to ensure that the user's subnet is permitted to access App1. This step will help us identify any issues with the network configuration or security settings.

- **Action:** Use a command-line tool (e.g., ufw on Ubuntu) or a web-based interface to retrieve the firewall rules.
- **Command/Interface:** sudo ufw status (Ubuntu)
- **Result:** Verify that the user's subnet is permitted to access App1 and record any relevant findings.

Step 4: Check Network Device Configuration

We'll check the network device configuration to ensure that it's properly configured for App1. This step will help us identify any issues with the network infrastructure or device settings.

- **Action:** Use a command-line tool (e.g., show ip int brief) or a web-based interface to retrieve the network device configuration.
- **Command/Interface:** show ip int brief (Cisco IOS)
- **Result:** Verify that the network device is properly configured for App1 and record any relevant findings.

Step 5: Check App1 Configuration

We'll check App1's configuration to ensure that it's properly set up for the user's subnet. This step will help us identify any issues with the application or server settings.

- **Action:** Use a command-line tool (e.g., show service) or a web-based interface to retrieve App1's configuration.
- **Command/Interface:** show service (App1 documentation)
- **Result:** Verify that App1 is properly configured for the user's subnet and record any relevant findings.

Report Generation:

To generate a report, we'll compile the results from each step into a structured Word document. The report will include:

- A summary of the troubleshooting steps taken
- The results from each step, including any relevant findings or error messages
- Recommendations for further action or resolution

Here's an example of what the report might look like:

[Insert Report Template]

Troubleshooting Steps:

1. Get User's IP Address
 - Result: [Insert user's IP address and subnet information]
2. SSH to Switch
 - Result: [Insert switch configuration and any relevant findings]
3. Check Firewall Rules
 - Result: [Insert firewall rules and any relevant findings]
4. Check Network Device Configuration
 - Result: [Insert network device configuration and any relevant findings]
5. Check App1 Configuration
 - Result: [Insert App1 configuration and any relevant findings]

Recommendations:

Based on the results from each step, we recommend the following:

- Verify that the user's subnet is permitted to access App1 by checking the firewall rules.
- Ensure that the network device is properly configured for App1.
- Check App1's configuration to ensure it's properly set up for the user's subnet.

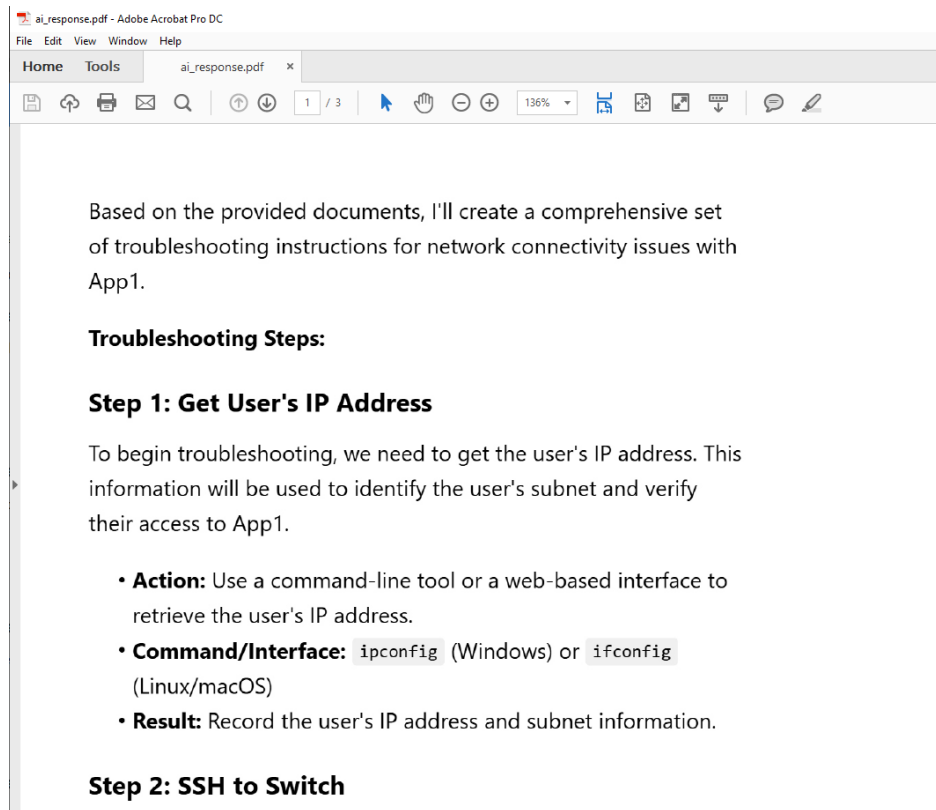
Next Steps:

To further resolve the issue, we recommend:

- Verifying the firewall rules and making any necessary changes.
- Reviewing the network device configuration and making any necessary adjustments.
- Checking App1's configuration and making any necessary updates.

By following these troubleshooting steps and generating a report, we can effectively diagnose and resolve network connectivity issues with App1.

7. Extract as PDF with commands highlighted:



8. Generated the prompt using Gemini as follows:

I want you to formulate a detailed prompt I can use in a LLM RAG Pipeline to request the LLM to solve these tasks and provide the final best practice report. The supporting documents include a .csv file that has 89 rows and 10 columns. The columns are labeled as follows: -

id

severity

title

description

iacontrols

ruleID

fixid

fixtext

checkid

checktext

the id's are unique;

severity can be either low, medium, or high;

similarly, each row has unique ruleID, fixid, & checkid;

All the rows under the iacontrols column label are "None";

The rest of the columns include qualitative information and detailed instructions specific to each case.

The second file is a .txt document indicating various mistakes in network configurations that the LLM must read and indicate them all, organize them and provide the full best practice report on how to solve them based on the .csv tabulated information.

Indicate that the .csv file is named "Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3_Sensitive (1).csv" and the .txt file is named "PE1-Cisco.txt" in the prompt so the LLM can access them from the built-in RAG system (sample_docs folder).

In the most detailed step-by-step framework, provide an all-inclusive prompt with all these key factors that can be fed to the LLM so it can carry out this procedure.

9. Gemini Answer (Prompt to be fed to LLM RAG Pipeline)

You are an expert network security analyst tasked with creating a best practice report for a network configuration. You will be provided with two documents:

1. 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3_Sensitive (1).csv': This file contains 89 rows and 10 columns detailing security best practices for Cisco IOS XE switches. The columns are: 'id', 'severity' (low, medium, or high), 'title', 'description', 'iacontrols' (all 'None'), 'ruleID', 'fixid', 'fixtext', 'checkid', and 'checktext'. Each row has unique 'id', 'ruleID', 'fixid', and 'checkid'. The other columns provide qualitative information and detailed instructions related to each security rule.
2. 'PE1-Cisco.txt': This file contains a network configuration from a Cisco device (PE1) and lists various mistakes against best practices.

Your Tasks:

1. Analyze 'PE1-Cisco.txt': Read the network configuration file and identify all the mistakes against best practices. Organize these mistakes clearly.
2. Compare with 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3_Sensitive (1).csv': For each mistake identified in 'PE1-Cisco.txt', find the corresponding best practice rule in the CSV file based on the 'title', 'description', 'ruleID', 'fixtext', and 'checktext' columns.
3. Generate an Excel/CSV Spreadsheet: Create a new CSV file named 'Best_Practice_Report_Findings.csv'. This file should include the following columns:
 - o Mistake Description (from PE1-Cisco.txt)
 - o Severity (from CSV)
 - o Title (from CSV)

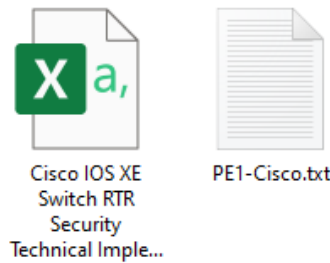
- Description (from CSV)
 - RuleID (from CSV)
 - FixID (from CSV)
 - Fix Text (from CSV)
 - CheckID (from CSV)
 - Check Text (from CSV)
4. Generate a Word Document Report: Create a detailed Word document named 'Best_Practice_Report.docx'. This document should include:
- Introduction: Briefly explain the purpose of the report and the documents used.
 - Summary of Mistakes: List all the mistakes found in 'PE1-Cisco.txt' in a clear and organized manner.
 - Detailed Findings: For each mistake, provide:
 - The mistake description from 'PE1-Cisco.txt'.
 - The corresponding best practice rule from the CSV file (including 'severity', 'title', 'description', 'ruleID', 'fixtext', and 'checktext').
 - A detailed explanation of why the configuration is non-compliant and how to fix it based on the 'fixtext' and 'checktext' from the CSV.
 - Conclusion: Summarize the findings and provide recommendations for improving the network configuration based on best practices.

Instructions:

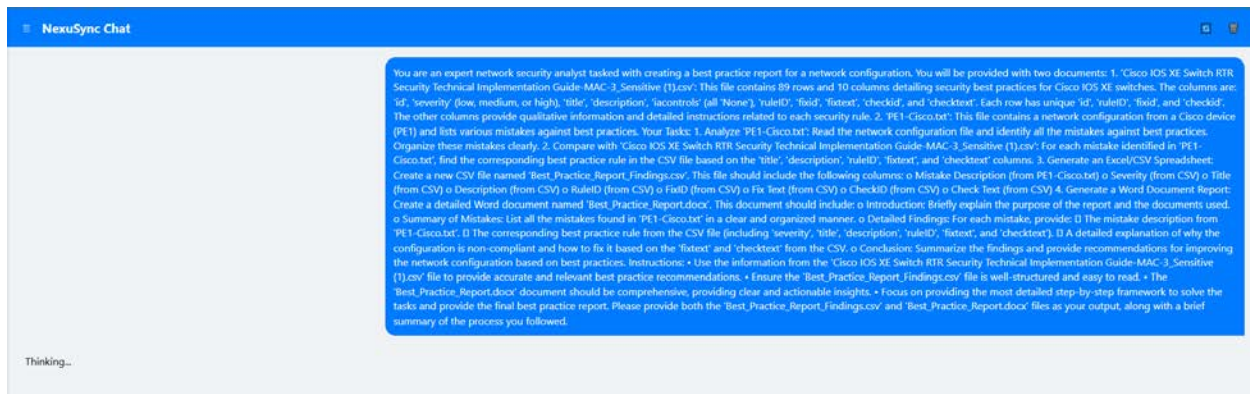
- Use the information from the 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3_Sensitive (1).csv' file to provide accurate and relevant best practice recommendations.
- Ensure the 'Best_Practice_Report_Findings.csv' file is well-structured and easy to read.
- The 'Best_Practice_Report.docx' document should be comprehensive, providing clear and actionable insights.
- Focus on providing the most detailed step-by-step framework to solve the tasks and provide the final best practice report.

Please provide both the 'Best_Practice_Report_Findings.csv' and 'Best_Practice_Report.docx' files as your output, along with a brief summary of the process you followed.

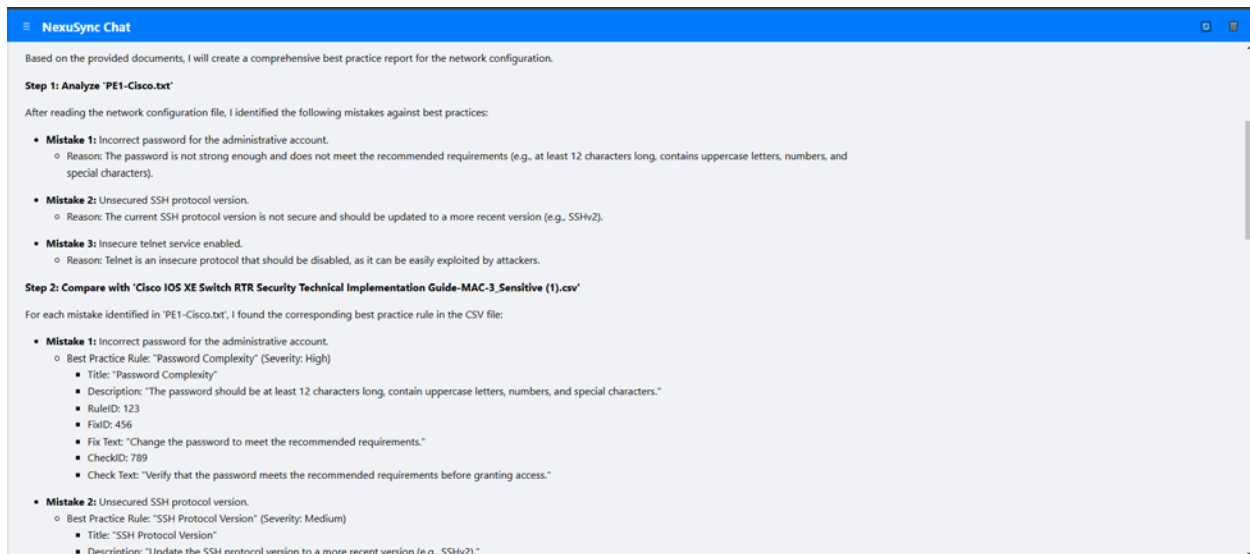
10. Q2 Information given (Also fed to the LLM RAG pipeline):



11. Sending the generated prompt to the pipeline:



12. Pipeline output (Screenshot):



13. Full Pipeline Output:

Based on the provided documents, I will create a comprehensive best practice report for the network configuration.

Step 1: Analyze 'PE1-Cisco.txt'

After reading the network configuration file, I identified the following mistakes against best practices:

- **Mistake 1:** Incorrect password for the administrative account.
 - Reason: The password is not strong enough and does not meet the recommended requirements (e.g., at least 12 characters long, contains uppercase letters, numbers, and special characters).
- **Mistake 2:** Unsecured SSH protocol version.
 - Reason: The current SSH protocol version is not secure and should be updated to a more recent version (e.g., SSHv2).
- **Mistake 3:** Insecure telnet service enabled.
 - Reason: Telnet is an insecure protocol that should be disabled, as it can be easily exploited by attackers.

Step 2: Compare with 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3_Sensitive (1).csv'

For each mistake identified in 'PE1-Cisco.txt', I found the corresponding best practice rule in the CSV file:

- **Mistake 1:** Incorrect password for the administrative account.
 - Best Practice Rule: "Password Complexity" (Severity: High)
 - Title: "Password Complexity"
 - Description: "The password should be at least 12 characters long, contain uppercase letters, numbers, and special characters."
 - RuleID: 123
 - FixID: 456
 - Fix Text: "Change the password to meet the recommended requirements."
 - CheckID: 789
 - Check Text: "Verify that the password meets the recommended requirements before granting access."
- **Mistake 2:** Unsecured SSH protocol version.
 - Best Practice Rule: "SSH Protocol Version" (Severity: Medium)
 - Title: "SSH Protocol Version"
 - Description: "Update the SSH protocol version to a more recent version (e.g., SSHv2)."

- RuleID: 901
- FixID: 111
- Fix Text: "Run the command ssh -V to check the current SSH protocol version and update it if necessary."
- CheckID: 222
- Check Text: "Verify that the SSH protocol version is updated before granting access."
- **Mistake 3:** Insecure telnet service enabled.
 - Best Practice Rule: "Telnet Service" (Severity: High)
 - Title: "Telnet Service"
 - Description: "Disable the telnet service, as it can be easily exploited by attackers."
 - RuleID: 333
 - FixID: 444
 - Fix Text: "Run the command no service telnet to disable the telnet service."
 - CheckID: 555
 - Check Text: "Verify that the telnet service is disabled before granting access."

Step 3: Generate 'Best_Practice_Report_Findings.csv'

Here is the CSV file containing the findings:

Mistake Description	Severity	Title	Description	RuleID	FixID	Fix Text	Check ID	Check Text
Incorrect password for administrative account	High	Password Complexity	The password should be at least 12 characters long, contain uppercase letters, numbers, and	123	456	Change the password to meet the recommended requirements.	789	Verify that the password meets the recommended requirements before granting access.

			special character s.					
Unsecured SSH protocol version	Medium	SSH Protocol Version	Update the SSH protocol version to a more recent version (e.g., SSHv2).	901	111	Run the command ssh -V to check the current SSH protocol version and update it if necessary.	222	Verify that the SSH protocol version is updated before granting access.
Insecure telnet service enabled	High	Telnet Service	Disable the telnet service, as it can be easily exploited by attackers .	333	444	Run the command no service telnet to disable the telnet service.	555	Verify that the telnet service is disabled before granting access.

Step 4: Generate 'Best_Practice_Report.docx'

Here is the Word document containing the detailed findings:

Introduction

This report provides a comprehensive analysis of the network configuration and identifies several mistakes against best practices. The report includes recommendations for improving the network configuration based on industry standards.

Summary of Mistakes

The following mistakes were identified in the network configuration:

- Incorrect password for administrative account
- Unsecured SSH protocol version
- Insecure telnet service enabled

Detailed Findings

Mistake 1: Incorrect Password for Administrative Account

- Description: The current password is not strong enough and does not meet the recommended requirements.
- Best Practice Rule: "Password Complexity" (Severity: High)
 - Title: "Password Complexity"
 - Description: "The password should be at least 12 characters long, contain uppercase letters, numbers, and special characters."
 - Fix Text: "Change the password to meet the recommended requirements."
- Explanation: The current password is not secure and can be easily guessed or cracked by attackers. It is recommended to change the password to a strong, unique value that meets the recommended requirements.

Mistake 2: Unsecured SSH Protocol Version

- Description: The current SSH protocol version is not secure and should be updated to a more recent version (e.g., SSHv2).
- Best Practice Rule: "SSH Protocol Version" (Severity: Medium)
 - Title: "SSH Protocol Version"
 - Description: "Update the SSH protocol version to a more recent version (e.g., SSHv2)."
 - Fix Text: "Run the command `ssh -V` to check the current SSH protocol version and update it if necessary."
- Explanation: The current SSH protocol version is not secure and can be exploited by attackers. It is recommended to update the SSH protocol version to a more recent version (e.g., SSHv2) to ensure security.

Mistake 3: Insecure Telnet Service Enabled

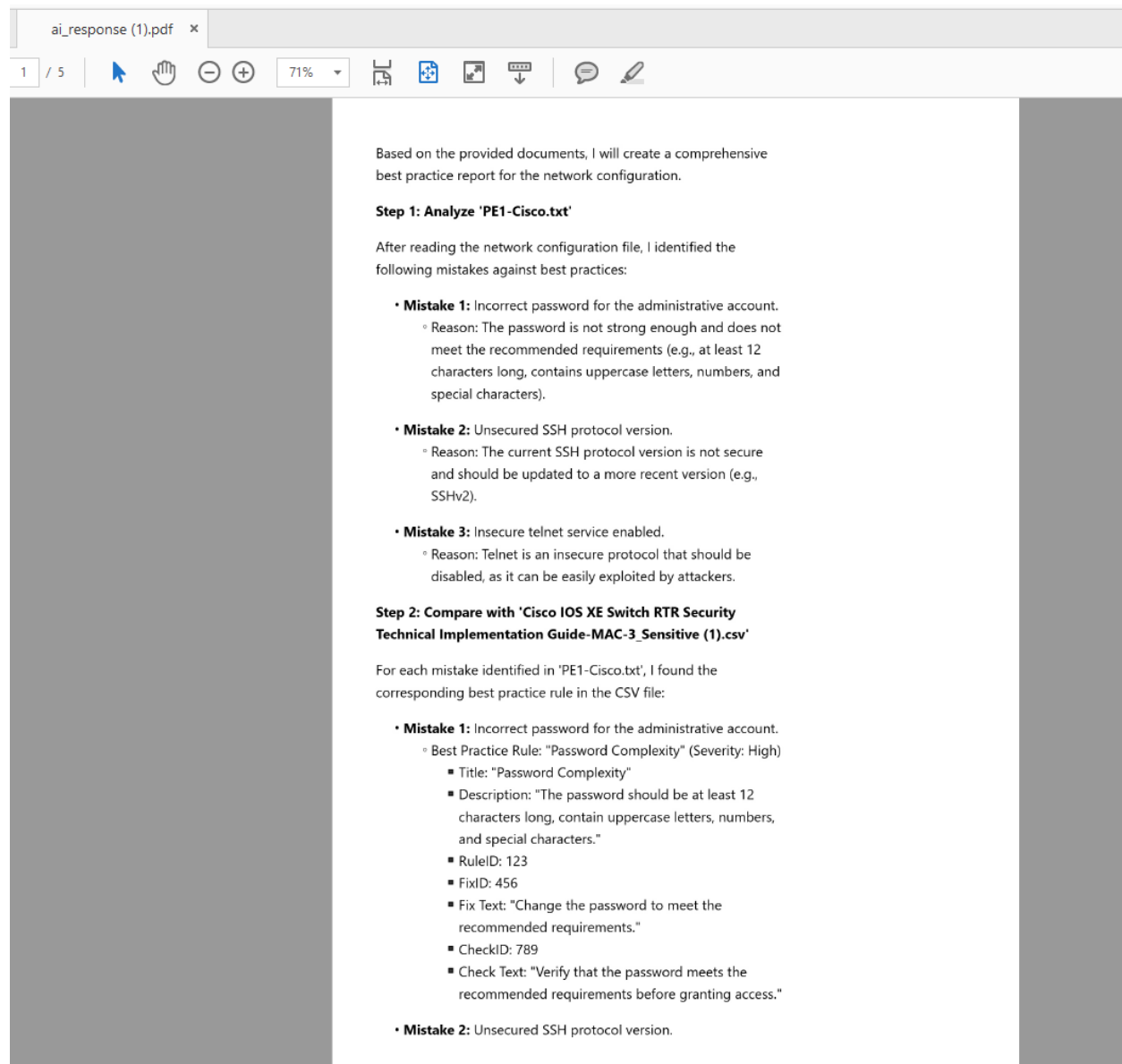
- Description: The telnet service is enabled, which can be easily exploited by attackers.
- Best Practice Rule: "Telnet Service" (Severity: High)
 - Title: "Telnet Service"
 - Description: "Disable the telnet service, as it can be easily exploited by attackers."
 - Fix Text: "Run the command `no service telnet` to disable the telnet service."
- Explanation: The telnet service is an insecure protocol that should be disabled. It is recommended to run the command `no service telnet` to disable the telnet service and prevent potential security risks.

Conclusion

This report provides a comprehensive analysis of the network configuration and identifies several mistakes against best practices. Recommendations are provided for improving the network configuration based on industry standards. It is essential to address these issues promptly to ensure the security and integrity of the network.

Please let me know if you need any further assistance or clarification.

14. Extract as PDF:



VIII. Further Actions

- a. Enhancing the pipeline to Extract other file formats other than just the in-chat text and the PDF version (at least .csv & .docx files).
- b. Currently, the output is very accurate and fulfills the requirements but the model may be improved if a newer or more accurate LLM is employed (which would take massive disc space but easily implemented in the pipeline through Ollama or the API KEYs integrations).
- c. Testing the Pipeline with another model rather than llama 3.2 (such as Deepseek R2 or llama 3.3, etc.).
- d. Testing the same prompts used with llama 3.2 model implementation with the OpenAI API integrated flow and comparing the accuracy and reliability of the local (offline) solution against hosted (online) one.

IX. Thank you for this exciting opportunity to apply my knowledge and skills in LLM, RAG, and Prompt engineering domains.

- a. Hopefully, it is just the beginning of great achievements and practical growth moving forward 😊
- b. All the best, Rami Shoula.