LOCAL AI

Daniel Ge
Period 678

CISCO

CCNP LAB REPORT

Purpose

In the following lab, we will download and set up a locally run AI hosted on our device. In addition to being able to select and use the AI without any network connection on our device, other devices on the network may also access the AI through a web browser. Setting up such a method of AI access ensures security by containing an organization's information within the organization's network, flexibility by allowing for unlimited AI use without the free limits set by public AI websites, and accessibility by allowing AI use without internet access.

Background Information

With the popularization of generative AI in the early 2020s, many people have grown to use AI for personal life and work. According to multiple 2025 studies, 56% of the US population use generative AI to write emails, compose poems, brainstorm ideas, write code, edit writing, or debug. Though AI indeed provides many time-saving conveniences for businesses, the public's increased reliance on AI also poses risks to organizations.

When it comes to AI use in business, the most critical issue is security. If an employee uses an online AI to debug proprietary code, the company hosting the AI will be able to view the code. Indeed, companies such as OpenAI have confirmed that any writing submitted to ChatGPT can be used by OpenAI for purposes such as training. Furthermore, even if the host company does not reuse the proprietary work, the AI can now learn from the prompt and potentially regenerate it for other users. As another example, if a government employee submits an email containing sensitive military information to AI for polishing, this could endanger the security of national intelligence.

Having employees rely on external AI poses other concerns. Administrators may want to track employee usage of external sources, but there is no reliable way to track an employee's use of external AI. This may pose issues in educational settings, for example, where school administrators may allow students to use AI for brainstorming, but restrict students from using AI for plagiarism. Relying on an external AI also comes with other limits such as its reliance on internet connection, rate limits set by AI companies, or server downtime due to maintenance out of our organization's control.

Rather than imposing restrictions on employee use of AI, one solution is to host an AI locally on a company computer or server reserved for employee use. Since the AI is run locally, the prompts, responses, and information exchanges with the AI remain on the company intranet. In particular, though the AI software we will use are created by OpenAI, Google, Microsoft… and downloaded from the internet, once the software is on our devices, they will not be able to communicate with their creators to exchange company information. Thus, all the aforementioned issues with security are solved. Additionally, since we are managing the AI software on company computers, it will be easy for us to track AI use, allow unlimited AI use, and schedule maintenance during times outside of working hours.

To host the AI on company computers, we will use Ollama, Docker, and Open WebUI. Ollama is a free application which allows us to download AI models from the cloud. It is an open-source application with code published publicly on GitHub by a community organization, and it also includes features which improves the efficiency of the various AI that may be downloaded. In this lab, we use Ollama to download an AI from the cloud. After the download, Ollama does not require any internet to run the lab. Therefore, using Ollama alone, we can successfully run a local AI on our device.

To achieve hosting an AI on our network for other users over the network to access, we require OpenWebUI's software. When other devices access the local AI, they will need to connect to the device hosting the local AI through typing its IP addresses and the correct port number. When they open the IP address and port, OpenWebUI will provide them with a web interface for other devices where they can sign in, send their requests, and receive their responses. In particular, OpenWebUI will set up the HTML of the page so that users see an organized and visually appealing webpage on their screen rather than blocks of text. In addition to creating the interface, OpenWebUI will also facilitate the communication of prompts and responses between the device hosting the AI and the client through APIs.

In order to run OpenWebUI's software on our computer, we must use a third application named Docker. Docker is a 2013 open-source engine which downloads images and runs containers on the computer. Since every device is different, when a device hosts an application like OpenWebUI, those applications need to be run in a self-contained "container" which is less affected by differences with the devices. In Docker, we can download "images", which are blueprints for these containers, then run the container. Due to this versatility, Docker can be used to run gaming servers or web proxies. In the context of this lab, we will use Docker to download an image for OpenWebUI's software. After we run the image for docker to create a container to run the software, the computer will become an AI server that can be accessed from other devices connected on the network.

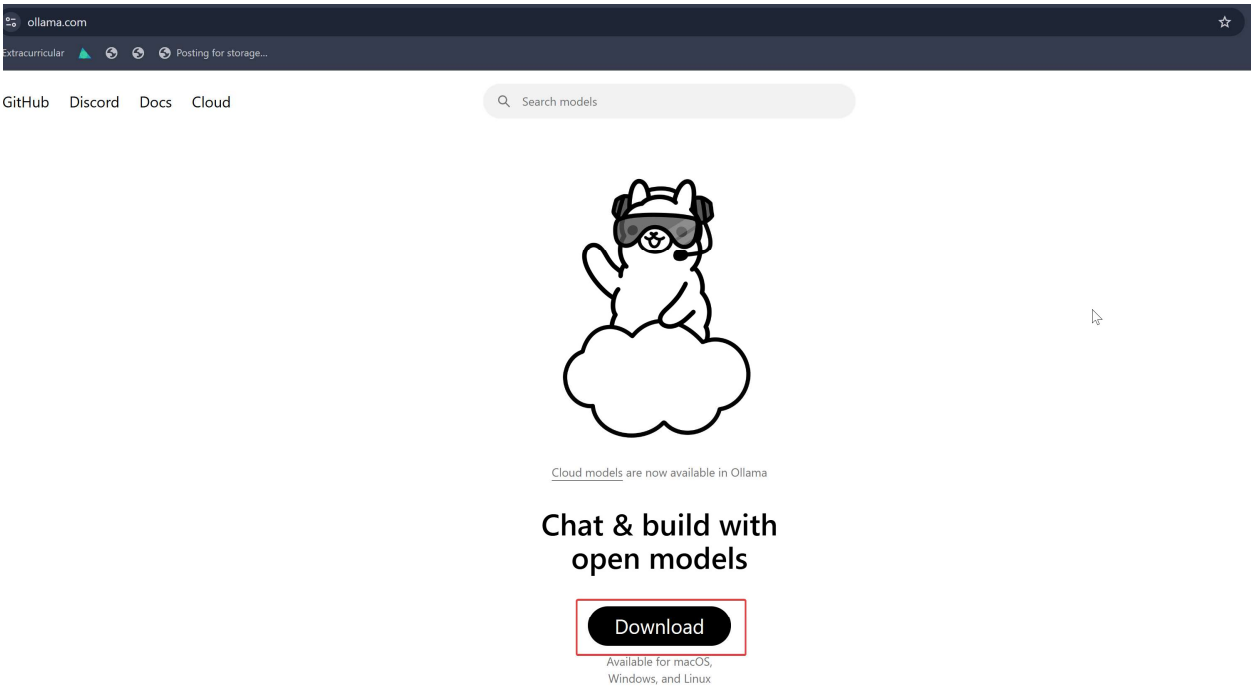
Lab Summary

In Part I of the lab, we will provide a method to run a local AI on the local area network (LAN) connected to the device. We will be downloading, configuring, and running Ollama to install a Gemma AI with four billion parameters. Afterwards, we will download Docker on the computer, which we will use to download and run OpenWebUI's software on port 3000. Using this setup, other devices can type in the IP address of the current devices with port 3000 in their browser to get OpenWebUI's page.

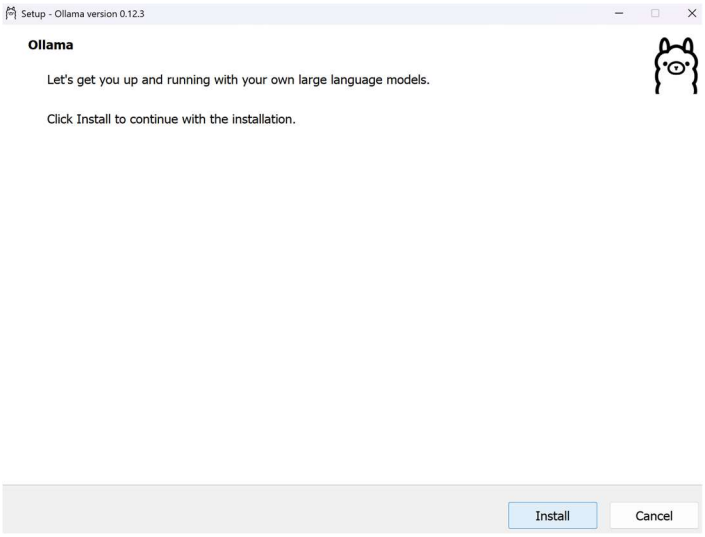
In Part II of the lab, we will provide an alternative method to run a local AI using LM Studio.

Lab Instructions (Part I)

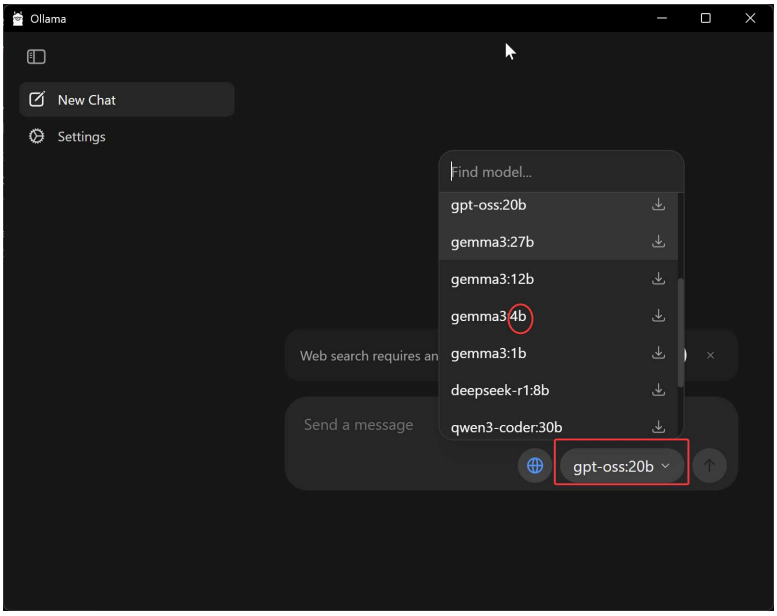
Download Ollama from the official website at ollama.com



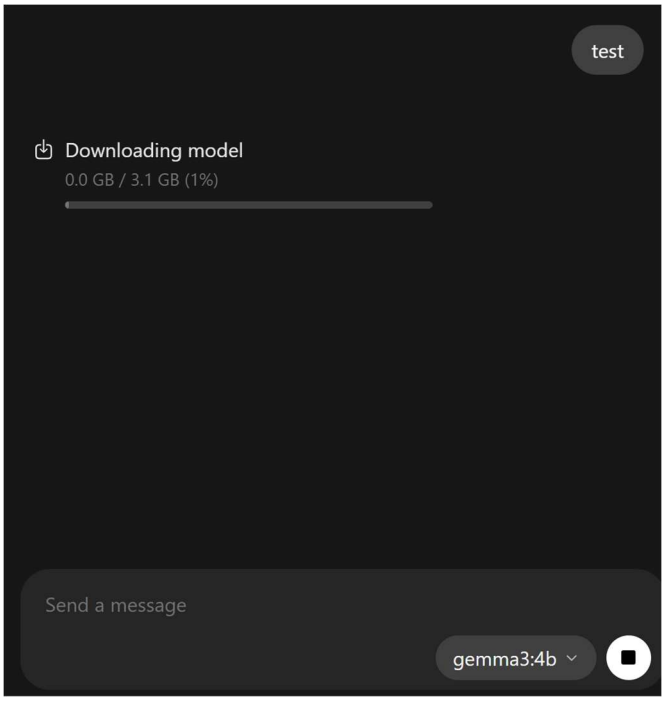
Open the installer and click Install



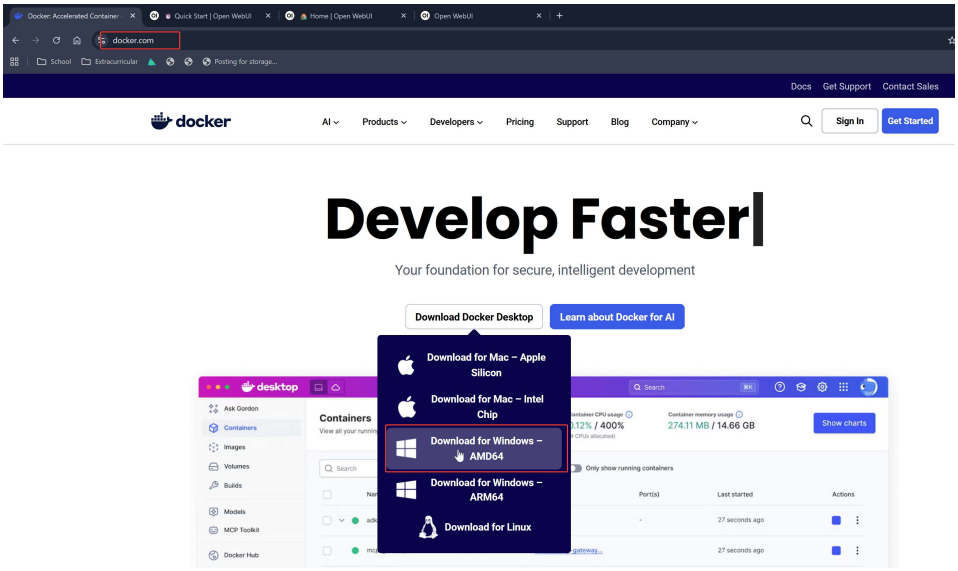
Select and install an appropriate model for your device such as gemma3



Send any message in the chat to begin download

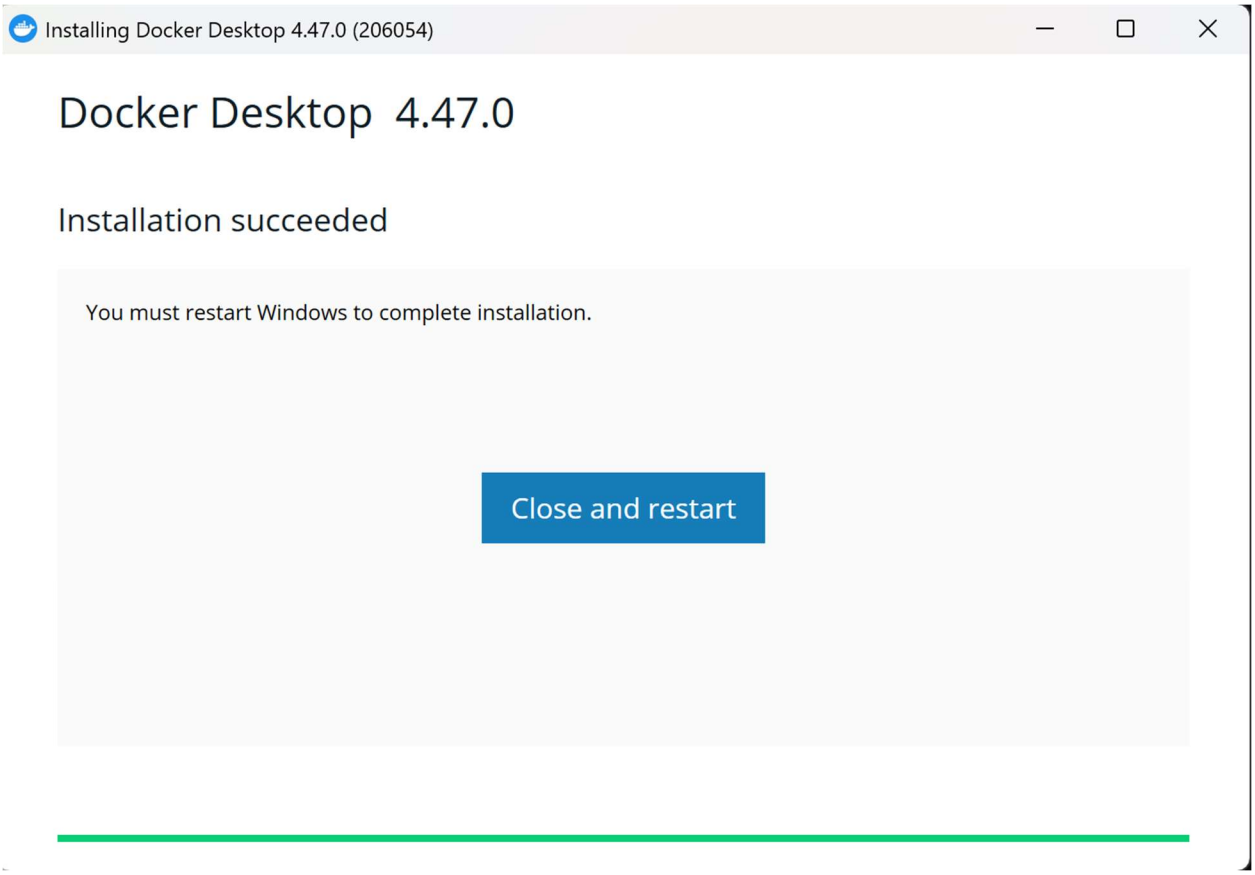


Download docker from the official website at docker.com. Select AMD64 download option



Run the Docker Installer and allow the program to make changes to your computer

Follow default settings, then click “Close and Restart” to complete Docker installation.



Launch Docker after the computer finishes restarting to confirm a successful installation.

Open the Windows Powershell in administrator mode and run the command

```
docker pull ghcr.io/open-webui/open-webui:main
```

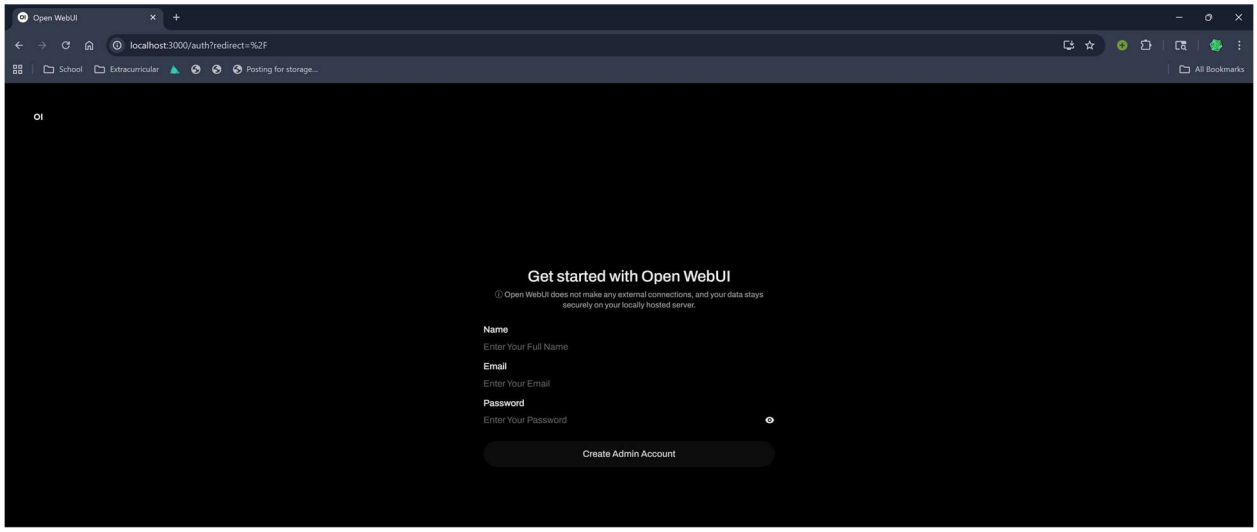
```
PS C:\WINDOWS\system32> docker pull ghcr.io/open-webui/open-webui:main
main: Pulling from open-webui/open-webui
659036234d55: Download complete
9563205556e0: Download complete
ce4392d98604: Download complete
82baee5aa91a: Downloading [>                                                  ]  2.097MB/1.301GB
4f4fb700ef54: Download complete
c8c8cb72c929: Pulling fs layer
d107e437f729: Pulling fs layer
1ee840479262: Pulling fs layer
227ace10ab92: Download complete
555e363b6de4: Pulling fs layer
1ce95d9ae66c: Pulling fs layer
b32275e9f783: Download complete
d5c111aa61fa: Pulling fs layer
a647374e6be3: Download complete
27054ed51888: Download complete
```

Enter the command,

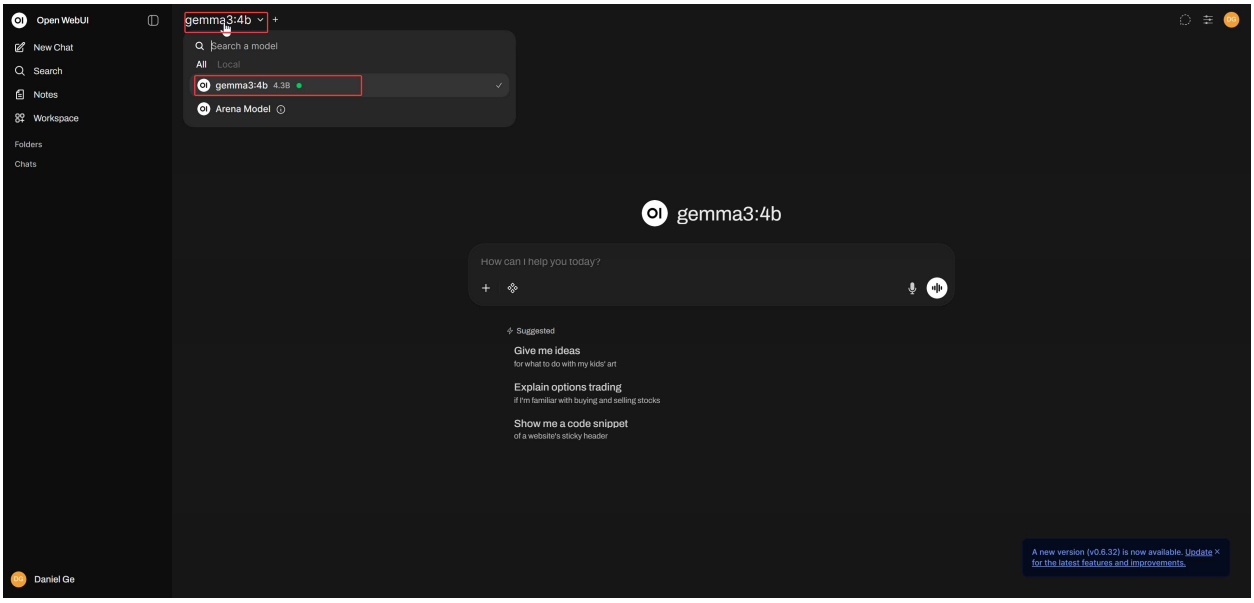
```
docker run -d -p 3000:8080 --gpus all -v open-webui:/app/backend/data --name open-webui ghcr.io/open-webui/open-webui:cuda
```

```
PS C:\WINDOWS\system32> docker run -d -p 3000:8080 --gpus all -v open-webui:/app/backend/data --name open-webui ghcr.io/
open-webui/open-webui:cuda
Unable to find image 'ghcr.io/open-webui/open-webui:cuda' locally
cuda: Pulling from open-webui/open-webui
4f4fb700ef54: Pull complete
f893b1b95fba: Pull complete
0c5c28ae5312: Pull complete
39c7879c2194: Pull complete
97ca80d7d37b: Downloading [=======================>                           ]  2.438GB/5.058GB
89e26dfa6fb4: Pull complete
60dbbb9770c7: Pull complete
c08a0d0f860a: Download complete
e4202b072733: Download complete
a4ce5c928e65: Download complete
e997e6dd4790: Download complete
```

Go to localhost:3000 and create an Admin Account



Select the model from the choices given,

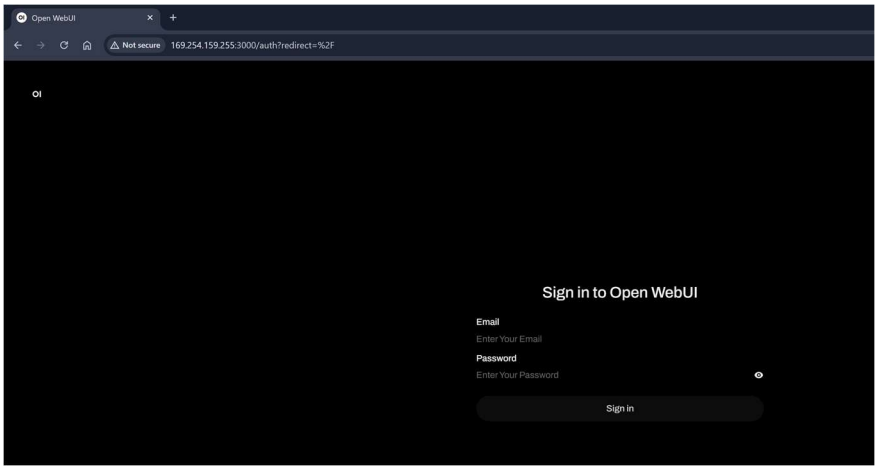


You may now access the model at localhost:3000 from your current device.

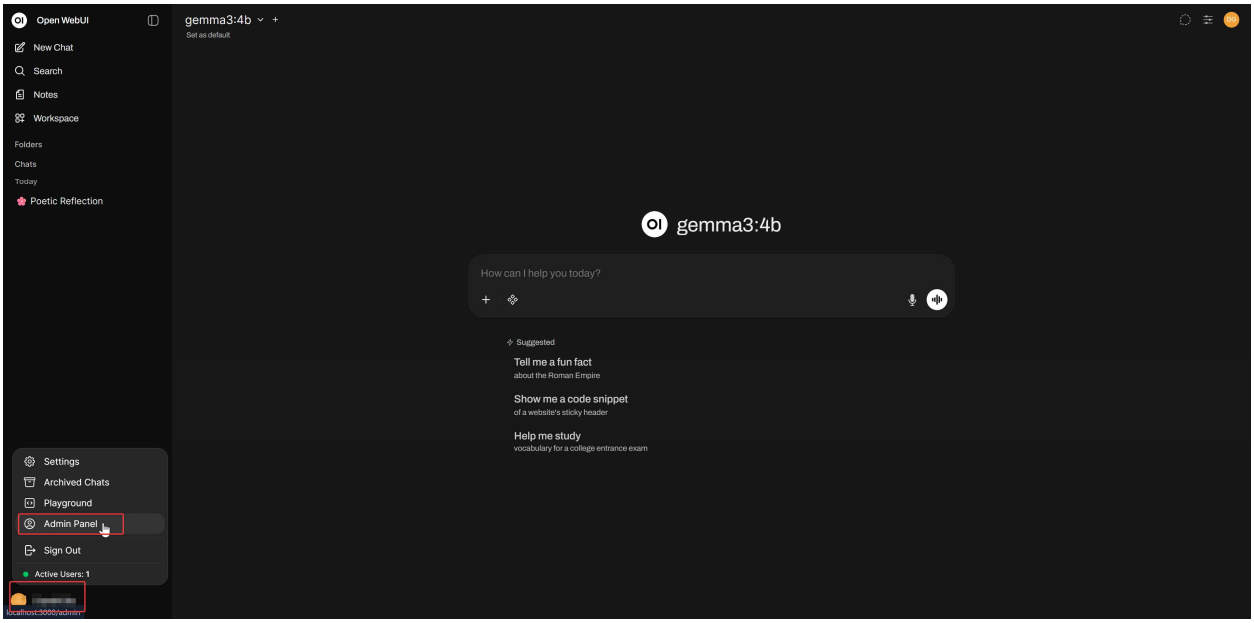
To access the model from other devices on the LAN, check your IP address using the command prompt's ipconfig command

```
C:\Users\Daniel>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter vEthernet (WSL (Hyper-V firewall)):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e490:d5d7:f82e:dda4%57
   IPv4 Address. . . . . . . . . . . : 172.19.0.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Autoconfiguration IPv4 Address. . : 169.254.159.255
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
```

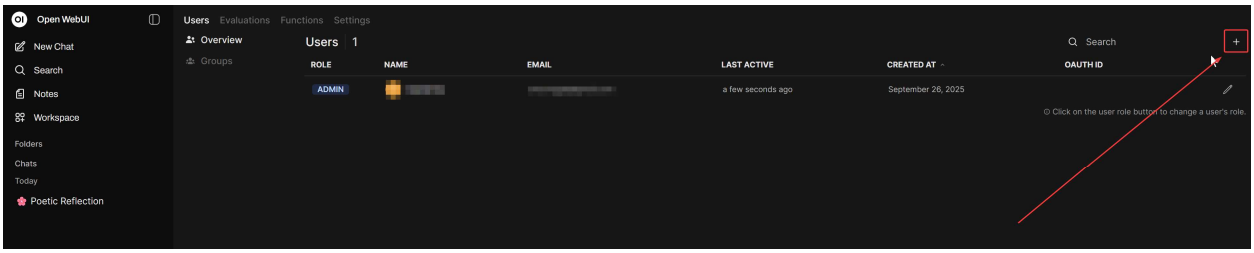
From another device connected on the Local Area Network, go to [ip address]:3000



If they do not already have an account, return to the device hosting Ollama's AI, click your profile name at the bottom-left of the Open WebUI screen, then select Admin Panel



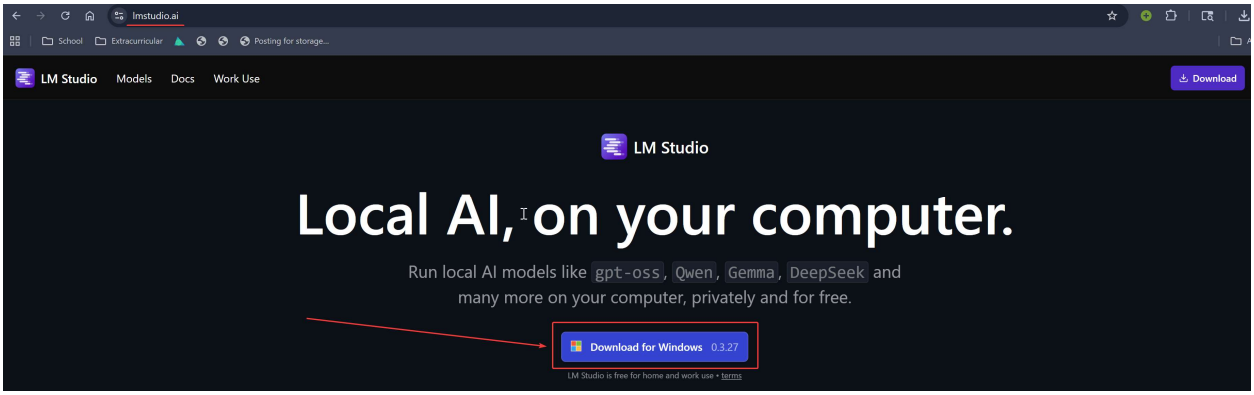
Click the Plus at the top right to register a new user for the other device



The new user can now sign in to use the AI with their configured password and username.

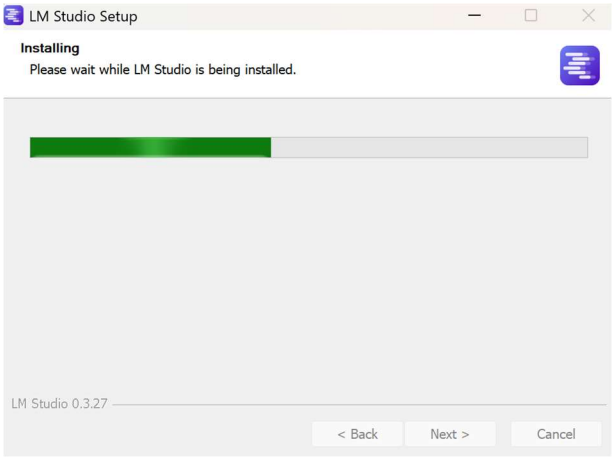
Lab Instructions (Part II)

Download lmstudio from the lmstudio.ai website

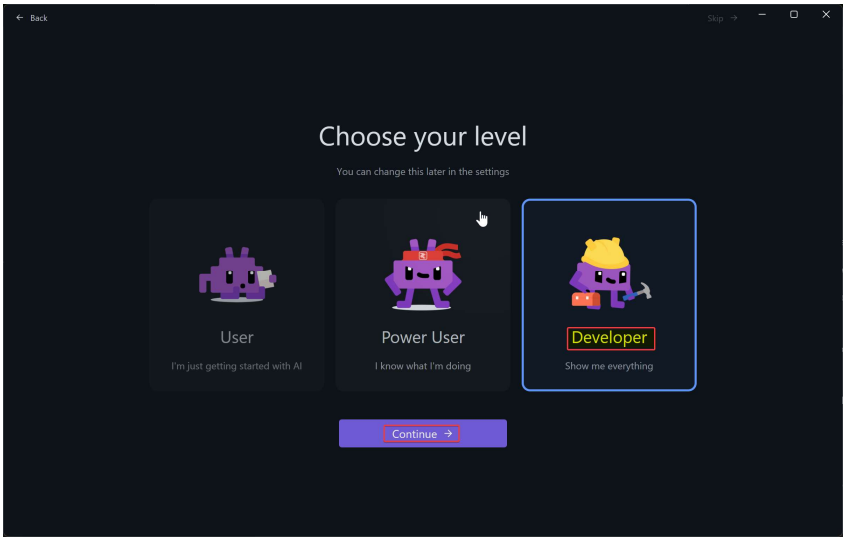


Run the download executable file to begin the installation

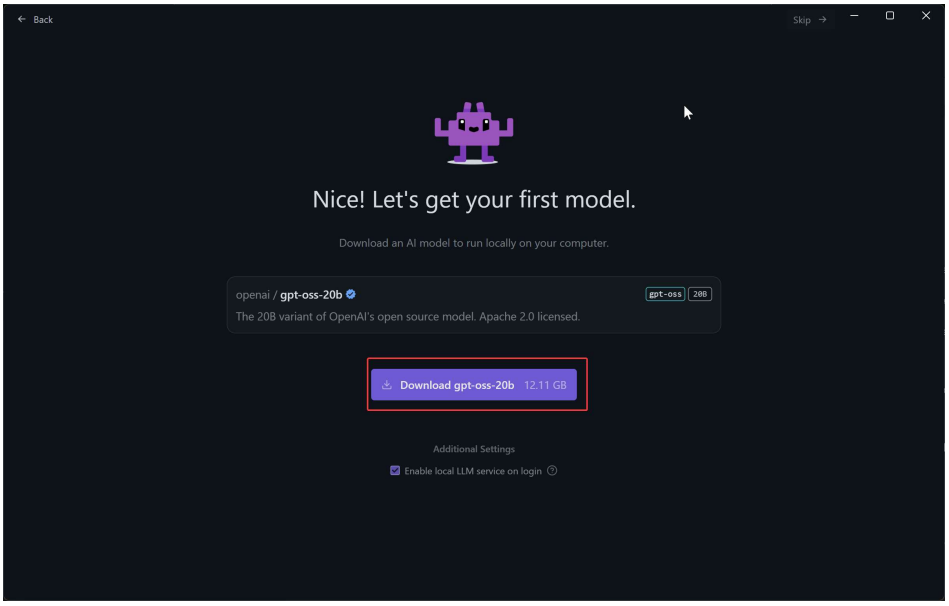
Click “Next”, “Install”, and “Finish” to apply default installation settings



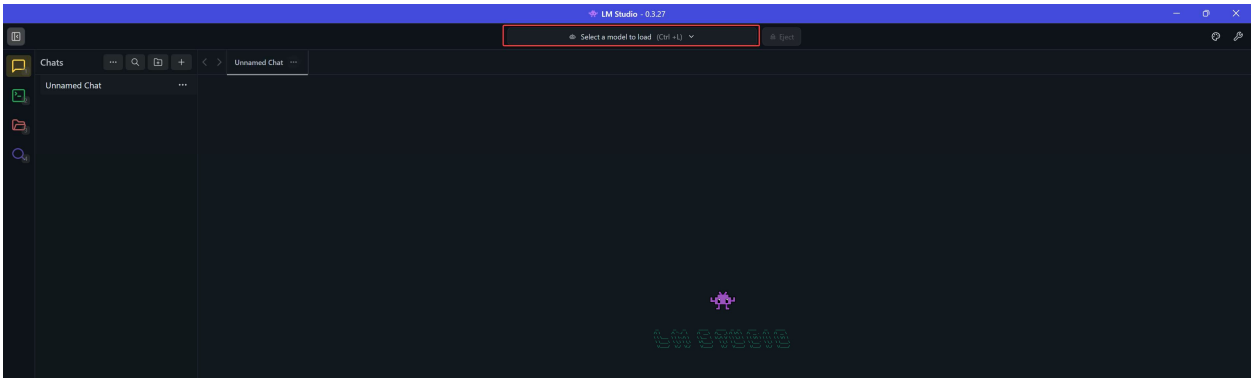
Select “Developer” mode and “Continue”



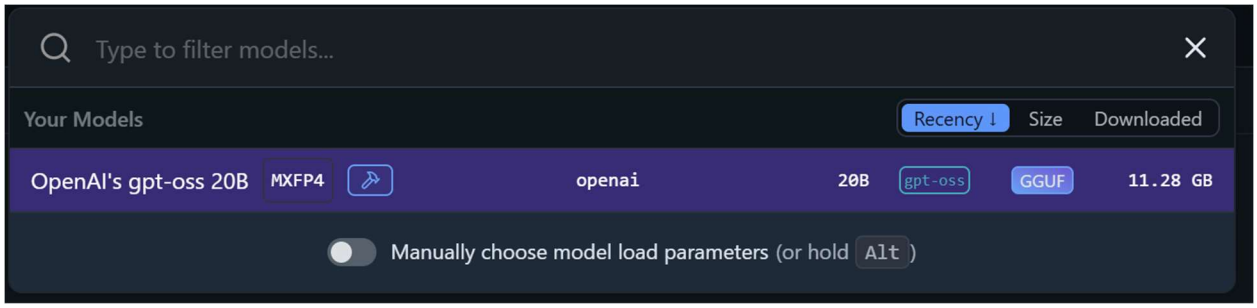
Click “Download gpt-oss-20b” to download the local AI



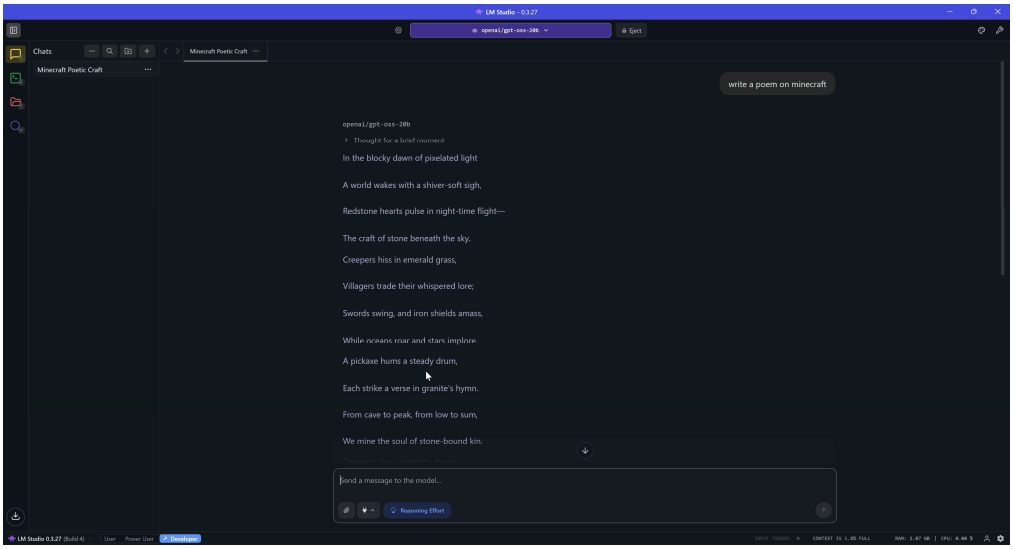
Once the download finishes, click “Select a model to load”



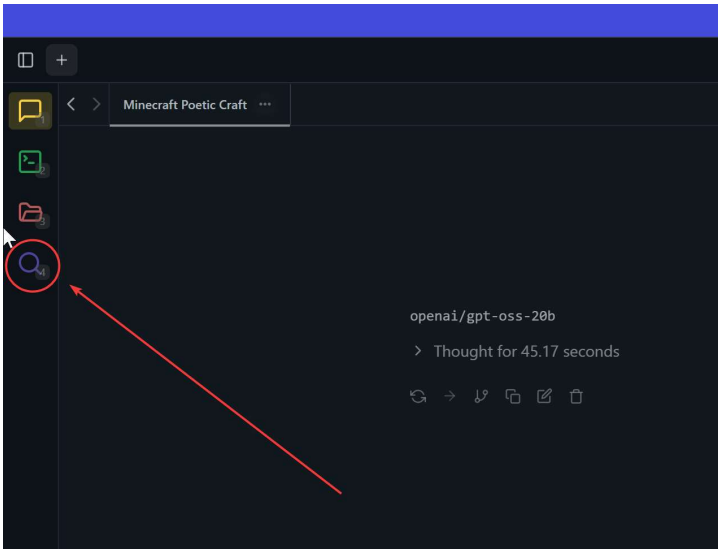
Select the newly downloaded gpt-oss model,



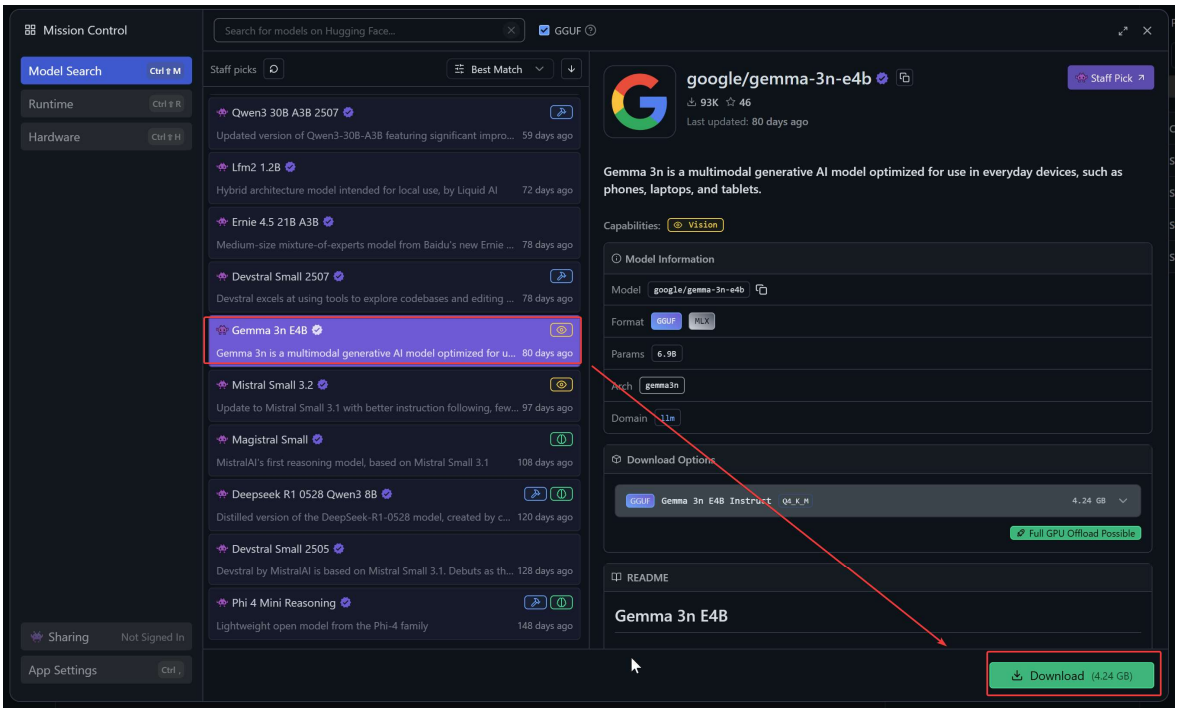
The model is now ready for local use, even without internet.



To change to a different model, click the search icon

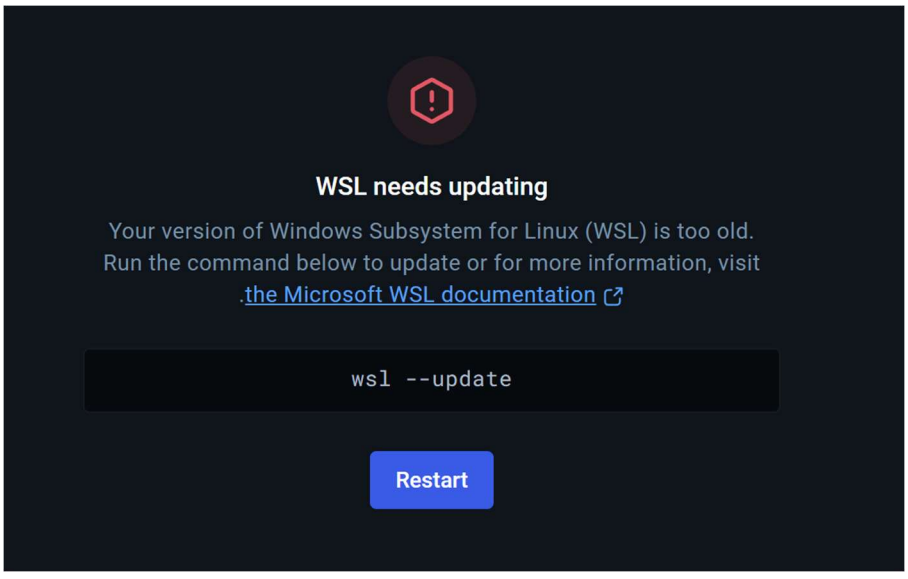


For an RTX A4000, it is optimal to select a model with roughly 8 billion parameters, such as Google's Gemma



Problems

Upon launching Docker, an error appeared on WSL needing updating.

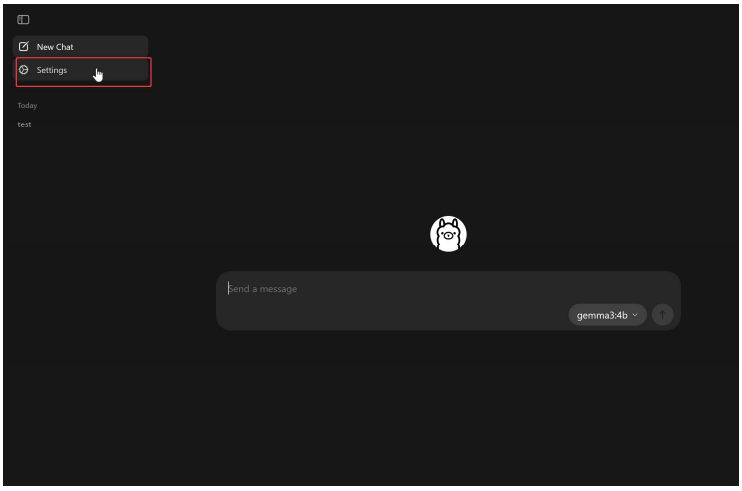


To fix the error, I opened the Windows Powershell in administrative mode and typed the command

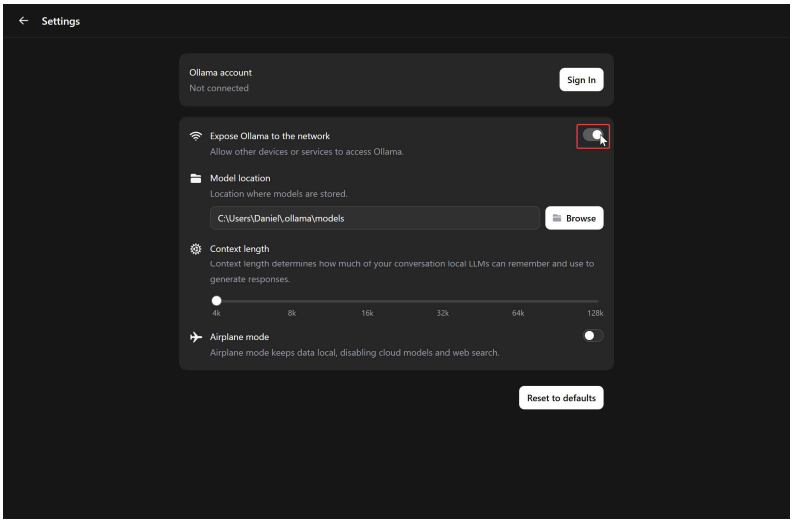
```
PS C:\WINDOWS\system32> wsl --update
Downloading: Windows Subsystem for Linux 2.6.1
[======                      11.6%                          ]
```

Once the download was completed, I clicked Restart on the Docker app and everything functioned as normal.

After the docker setup finished, I was able to reach localhost:3000 on the device and use the AI. However, the AI did not return any response when other devices on the network try to use it, even though the OpenWebUI interface showed up. Though I initially thought this was a Docker issue solved by restarting the container, it turned out to be a configuration issue with Ollama. In Ollama, I clicked the settings menu.



After clicking “Expose Ollama to the network”, the AI worked and returned responses to other devices on the network.



Conclusion

Through this lab, we practiced how to protect the security of company information and ensure the reliability of employee access to AI. With the continued advent of generative AI becoming more integrated in daily life and work, such improvements are indispensable to ensuring that our organization fully leverages that AI can offer. Rather than offer. Rather than restricting AI due to its risks, we now encourage employees to embrace the technology in good faith, improving the overall productivity of our organization.

Signoff Page