DesignNews ICS

Unlock Limitless LLM Access with Ollama & Open WebUI

By Mark Antonelli - Director of AI/ML & Video Solutions, Integrated Computer Solutions (ICS)
April 15, 2025

Using these tools, you don't need the cloud to run AI models.

Generative AI offers incredible potential, but concerns about privacy, costs, and limitations often push users toward cloud-based models. If you're frustrated with daily limits on ChatGPT, Claude, or other public large language models (LLMs)—or if your company restricts access to external AI models like DeepSeek—there's an appealing alternative: run powerful AI models entirely on your own machine, without relying on the cloud.

With open-source tools Ollama and Open WebUI, you can run models locally, bypassing the constraints of cloud platforms. Whether you need to generate text, answer questions, or perform data analysis, these tools allow you to unlock the full potential of LLMs on your own hardware—no subscription required.

What are Ollama & Open WebUI?

Ollama and Open WebUI are powerful tools that enable software developers to run LLMs on their own machines. Ollama provides a simple interface for managing and running various LLMs locally, while Open WebUI offers a user-friendly, browser-based interface for interacting with these models. This setup offers several advantages, including greater control, improved privacy, the ability to customize models for specific use cases, and reduced costs.

Launched in 2023, Ollama is an independent startup and the brainchild of tech entrepreneurs Michael Chiang and Jeffrey Morgan based on their desire to democratize access to powerful AI technologies. Ollama facilitates running models like Meta AI's Llama but the organizations are not related. In its earliest days, what would become Ollama participated in the W21 (Winter 21) batch of Y Combinator, a noted technology startup accelerator and venture capital firm that has helped launch more than 5,000 companies since its inception in 2005.

Open WebUI, developed by software engineer Tim J. Baek, is often used as a front end for Ollama, providing a more user-friendly way to interact with the LLMs that Ollama runs locally. (Open WebUI relies on a GUI rather than command-line tools.)

By allowing users to run LLMs locally on their own machines, these tools provide users with enhanced control over data privacy, security, and customization.

Take control of your AI experience

Let's explore the process step by step and then take a look at some troubleshooting tips in case something goes awry. Let's get started!

Hardware requirements

You'll need a GPU with at least 4 Gb of dedicated video RAM. When creating this article, I used an older laptop with a i7-9950 CPU, 16 Gb RAM, and an NVIDIA Quadro T1000 with 4Gb of video RAM. In general, the more powerful the system in terms of the video card and video memory, the faster the LLM response will be. A more powerful system will also let you run larger models which will give better, more accurate responses.

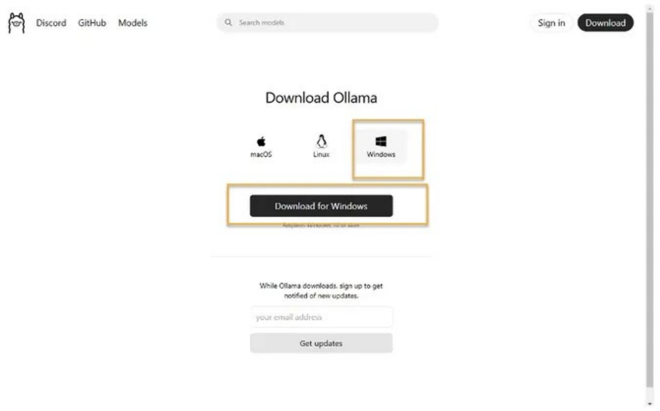
Installation

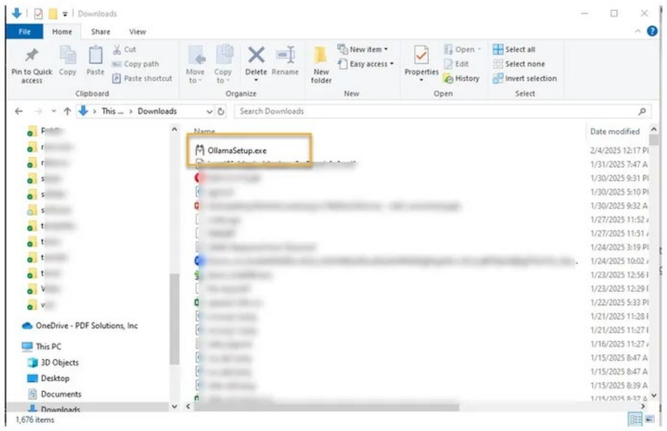
You can handle Open WebUI installation in a number of ways including Docker. However, I chose the more traditional installation using Ollama, Python, and Open WebUI, which I'll explain.

Step 1: Install Ollama
Download and install Ollama using the URL below:

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

Choose the appropriate option, macOS, Linux, or Windows (our choice). Click the 'Download for Windows' button and then locate the installer in your Downloads folder.





You can either double click the command pictured above or run the installer via the Windows Command Prompt. For my installation, I used all the default options. Note that Ollama runs on port 11434.

Ollama has a command line interface. While you can interact with it directly this way, without Open WebUI, you'll only use it to install LLM models. The following command will list what models have already been installed. For a new installation, there will be none.

```
ollama list
```

Next we can install a model. We will pull (download) the latest DeepSeek model that is customized for limited compute power.

```
ollama pull deepseek-r1:1.5b
```

That is all you need to do with Ollama, though you can experiment with the command line interface — just type the word ollama to see the options.

Step 2: Install Python
The next step is to install Python. Python 3.11 is recommended. If you do not already have it installed you can get it here and run the installation executable.

```
https://www.python.org/downloads/release/python-3119/
```

I used the Windows Installer (64-bit) with all the default options:

```
https://www.python.org/ftp/python/3.11.9/python-3.11.9-amd64.exe
```

It is a good idea to use virtual environments with Python so that your many projects do not interfere with each other. I recommend using the following commands to create and activate a virtual environment named venv. Note that you should first create a directory that will contain the required environment files. I named my directory open-webui, but you can use any name you'd like.

```
cd %USERPROFILE%
```

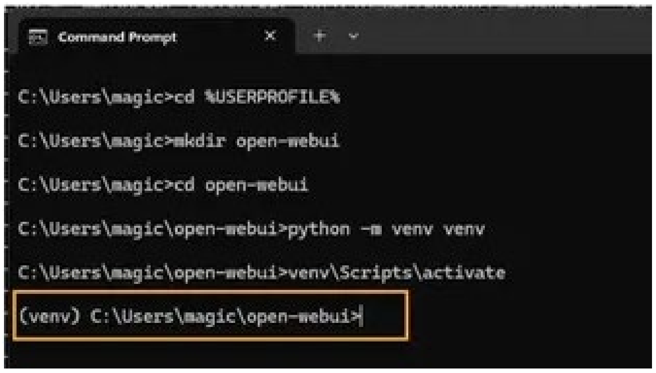
```
mkdir open-webui
```

```
cd open-webui
```

```
python -m venv venv
```

```
venv\Scripts\Activate
```

Your command prompt should look something like this. Note the (venv) indicating a virtual environment is active:



```
C:\Users\magic>cd %USERPROFILE%

C:\Users\magic>mkdir open-webui

C:\Users\magic>cd open-webui

C:\Users\magic\open-webui>python -m venv venv

C:\Users\magic\open-webui>venv\Scripts\activate

(venv) C:\Users\magic\open-webui>
```

Step 3: Install Open WebUI

The last step is to install Open WebUI. In the same Windows command prompt in which you installed Python, use the following command to install Open WebUI.

```
pip install open-webui
```

You can get full details on Open WebUI, including alternate installation methods at the following URL.

```
https://github.com/open-webui/open-webui
```

Running Open WebUI

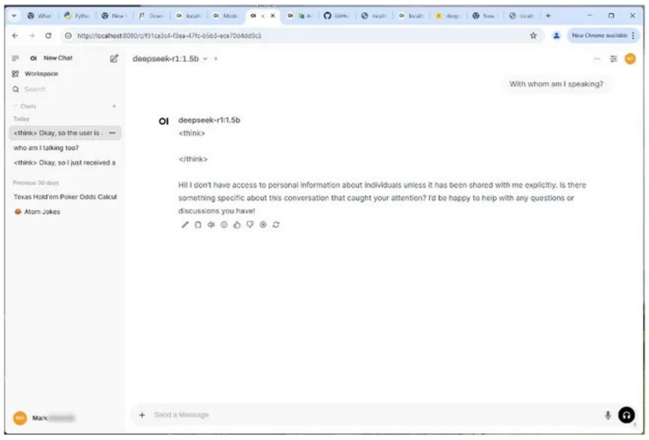
After installation, you can start Open WebUI by executing the following command in the same Windows command prompt:

```
open-webui serve
```

After some time for configuration, this will start the Open WebUI server on port 8080. In your favorite web browser, enter the following UI and then you can start your chat session.

```
http://localhost:8080
```

You should see something like the following:


http://localhost:8080/c/f91ca3c4-f3ea-47fc-b5b3-ece70d4dd9c3
New Chrome available
New Chat
Workspace
Search
Chats
Today
<think> Okay, so the user is
who am I talking too?
<think> Okay, so I just received a
Previous 30 days
Texas Hold'em Poker Odds Calcul
Atom Jokes
Mark [redacted]
deepseek-r1:1.5b
Send a Message

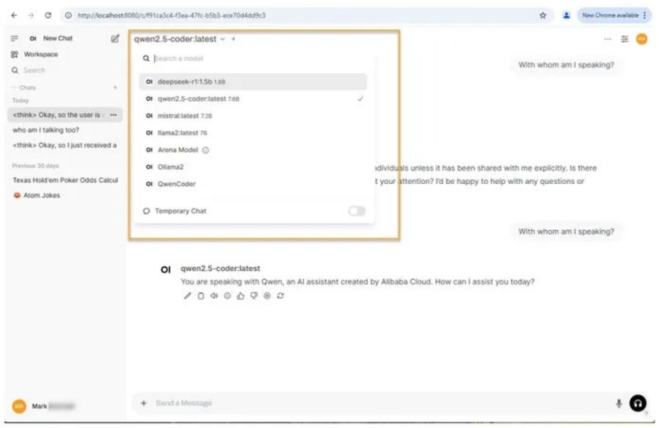
With whom am I speaking?

deepseek-r1:1.5b
<think>

</think>

Hi! I don't have access to personal information about individuals unless it has been shared with me explicitly. Is there something specific about this conversation that caught your attention? I'd be happy to help with any questions or discussions you have!

If you have multiple models installed, you may have to initially select the model. In addition, you can set a default model.



With whom am I speaking?

qwen2.5-coder:latest
You are speaking with Qwen, an AI assistant created by Alibaba Cloud. How can I assist you today?

If all went well, you are up and running and can interact with your local LLMs free of charge!

Troubleshooting

Something go wrong? Here are some troubleshooting tips.

Port usage

By default, Ollama uses port 11434, and Open WebUI uses 8080. Port 8080, in particular, is used by a lot of applications including Apache Tomcat, Jenkins, and proxy servers. You can easily change the port used in the start-up command. For example, try:

```
open-webui serve --port 8081
```

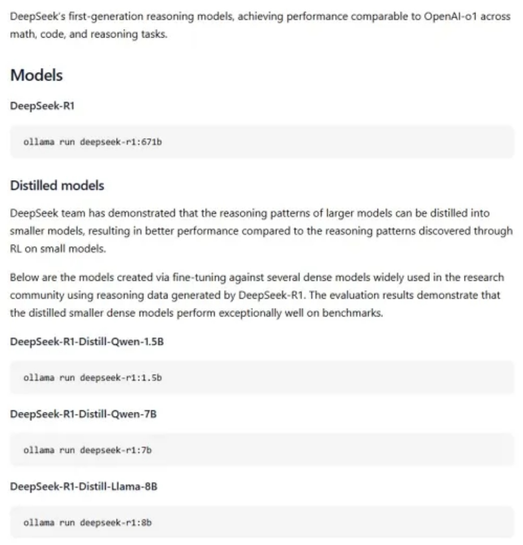
Install additional models

Installing additional models is quite simple. First, go to the following Ollama URL to see all available models:

```
https://ollama.com/library
```

Click on the model name link to understand parameters and get the exact name of the model you wish to download. Here is an example for DeepSeek. Note that you will want a distilled model that is a variation of the base model that is less compute intensive.

You may see the command ollama run . That will both pull and run the model. For purposes of Open WebUI, always use ollama pull and then it will run automatically in Open WebUI.



DeepSeek's first-generation reasoning models, achieving performance comparable to OpenAI-o1 across math, code, and reasoning tasks.

Models

DeepSeek-R1

```
ollama run deepseek-r1:671b
```

Distilled models

DeepSeek team has demonstrated that the reasoning patterns of larger models can be distilled into smaller models, resulting in better performance compared to the reasoning patterns discovered through RL on small models.

Below are the models created via fine-tuning against several dense models widely used in the research community using reasoning data generated by DeepSeek-R1. The evaluation results demonstrate that the distilled smaller dense models perform exceptionally well on benchmarks.

DeepSeek-R1-Distill-Qwen-1.5B

```
ollama run deepseek-r1:1.5b
```

DeepSeek-R1-Distill-Qwen-7B

```
ollama run deepseek-r1:7b
```

DeepSeek-R1-Distill-Llama-8B

```
ollama run deepseek-r1:8b
```

Since you have already installed DeepSeek, here's another example where Llama 3.2 is pulled from the Windows command prompt:

```
ollama pull llama3.2:1b
```

To have this model show up in Open WebUI, just refresh your web page and it should show up as a new option.

Conflicting versions of Python

If you already have Python installed, you may have concerns about the installation of Python 3.11 causing version conflicts. Check to see whether you already have the Python launcher py installed. If not, you can install it as part of Python 3.11. With the launcher, you can create your virtual environment as follows:

```
py -3.11 -m venv venv
```

Whenever the virtual environment is activated, it will use the 3.11 of Python.

Takeaways: Increase security, decrease costs

If you're looking to harness the power of generative AI without compromising privacy, dealing with daily usage limits, or facing recurring cloud costs, run powerful AI models locally using open-source tools like Ollama and Open WebUI. With this approach, you can enjoy more secure and limitless access as well as greater control and zero extra expenses.

About the Author

Mark Antonelli

Director of AI/ML & Video Solutions, Integrated Computer Solutions (ICS)

Mark Antonelli, Director of AI/ML & Video Solutions at Integrated Computer Solutions (ICS), is a technology specialist with particular expertise in the realm of audio/video systems development and machine learning. For more than a decade, he headed the development team creating a robust, market-leading, multi-channel video surveillance system and multiple follow-on products.

ICS

DesignNews

April 15, 2025