How to Run a Local LLM with Ollama

A user guide for Windows

Mandy Hathaway · mandyhathaway.com

OVERVIEW

This guide walks you through installing Ollama on Windows, pulling a language model, and running it locally on your own machine. By the time you are done, you will have a working AI assistant that runs entirely offline, with no data sent to any external server.

Ollama is a free, open source application that makes it straightforward to download and run large language models locally. It handles the technical infrastructure behind the scenes so you can focus on working with the models themselves.

This guide is written for professionals who are comfortable with computers but may not have used the Windows command line before. Section 2 covers everything you need to know to get started with it. If you are already comfortable in the command line, you can skip that section.

What You Will Need

- A Windows 10 or Windows 11 machine.
- **Minimum 8GB RAM**. 16GB or more is recommended if you plan to run larger models.
- **At least 10GB of free disk space** for Ollama and at least one model. Models range from roughly 2GB to 20GB depending on size.
- An internet connection for the initial download. Once installed, Ollama runs completely offline.

What You Will Have When You Are Done

- Ollama installed and running on your Windows machine.
- At least one language model downloaded and ready to use.
- The ability to run conversations, send API requests, and integrate models into Python scripts.

> **Important**
>
> Ollama runs AI models directly on your hardware. Larger models require more RAM and processing power. If your machine has less than 16GB RAM, start with a smaller model like Phi-3 or Mistral. The guide covers model selection in detail in Section 4.

SECTION 1: WHAT IS OLLAMA

A large language model (LLM) is a type of AI system trained on vast amounts of text. It can answer questions, summarize information, write code, explain complex topics, and carry on extended conversations. Until recently, using an LLM meant sending your prompts to a remote server run by a company like OpenAI or Anthropic, which raised real questions about data privacy, cost, and internet dependency.

Ollama changes that. It is a runtime that lets you download and run open source language models directly on your own computer. Your prompts never leave your machine. There are no API costs, no rate limits, and no internet connection required once the model is downloaded.

Ollama also exposes a local REST API, which means you can integrate it into your own scripts, applications, and workflows using Python or any other language that can make HTTP requests.

Models Covered in This Guide

Llama 3.2

Llama 3.2 is Meta's latest open source model. It handles a wide range of tasks well, including writing, analysis, summarization, and question answering. The 3B parameter version runs comfortably on most machines with 8GB or more RAM.

Mistral

Mistral 7B is a well-established, efficient model developed by Mistral AI. It is fast, reliable, and runs well on hardware that might struggle with larger models. It is a solid general purpose choice and works well for code-related tasks.

Gemma 3

Gemma 3 is Google’s latest open model family, released in 2025. The 4B version runs comfortably on machines with 8GB RAM and offers a 128k context window, meaning it can handle much longer conversations and documents than most models in its size range. It performs well on general chat, summarization, and instruction following.

DeepSeek-R1

DeepSeek-R1 is a reasoning-focused model that works through problems step by step before producing an answer. This makes it noticeably better than general models at logic, analysis, and tasks that benefit from careful thinking. The 8B version requires 8GB or more RAM. It is a good choice when you need more than a quick response.

> **Tip**
> The number after a model name, like 7B or 3B, refers to the number of parameters in the model. More parameters generally means more capability but also more RAM and disk space required. For most everyday tasks, smaller models perform surprisingly well.

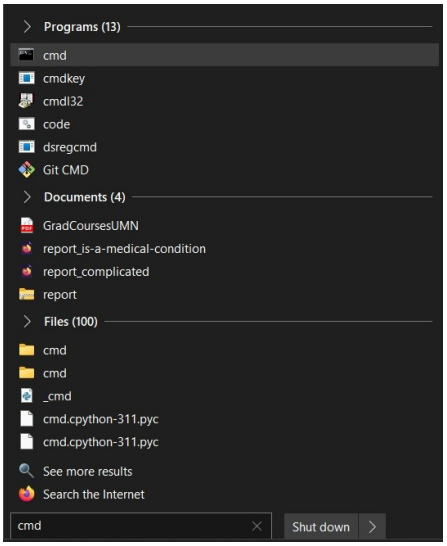
SECTION 2: USING THE WINDOWS COMMAND LINE

Ollama is controlled through the Windows command line. If you have used it before, you can skip ahead to Section 3. If it is new to you, this section covers everything you need to get started.

The command line is a text based interface where you type instructions directly instead of clicking through menus. It sounds more intimidating than it is. For working with Ollama, you only need a handful of commands, all of which are covered in this guide.

Opening the Command Line

1. Press the **Windows key** on your keyboard to open the Start menu.
2. Type `cmd` and press **Enter**. You will see `cmd` listed under Programs at the top of the results. Click it to open Command Prompt.



You will see a black window with white text showing your current location on the file system, followed by a blinking cursor. This is where you type your commands.

Tip

You can also open Command Prompt by pressing the Windows key and R at the same time, typing cmd,

and pressing Enter. On Windows 11, you can right-click the Start button and select Terminal for a more modern version of the same thing.

Basic Command Line Concepts

Running a command

Type the command and press Enter. The command runs and the result appears below it. When it is finished, the prompt returns and you can type another command.

Cancelling a running command

If something is running and you want to stop it, press **Ctrl + C**. This sends an interrupt signal and stops the current process.

Clearing the screen

Type `cls` and press Enter to clear the Command Prompt window if it gets cluttered.

File paths

When you see a file path like C:\Users\YourName, the backslash is the Windows path separator. You will not need to type paths often when working with Ollama, but it helps to recognize them.

Additional Resources for the Command Line

If you want to build more confidence with the command line before continuing, the following resources are reliable and beginner friendly.

- Microsoft's official command line documentation: learn.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands
- SS64 Windows command reference (concise and searchable): ss64.com/nt/
- freeCodeCamp command line crash course for beginners: freecodecamp.org/news/command-line-for-beginners/

Issue: Command Prompt opens and immediately closes.

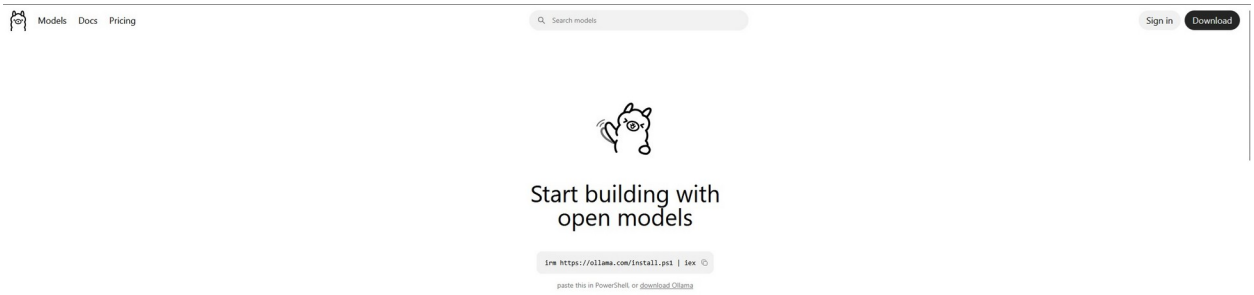
Solution: This usually means Windows is opening and closing a script rather than the interactive Command Prompt. Make sure you are opening Command Prompt by searching for cmd in the Start menu, not by double-clicking a .cmd file.

Issue: You see "Access is denied" when running a command.

Solution: Some commands require administrator privileges. Right-click Command Prompt in the Start menu and select Run as administrator, then try the command again.

SECTION 3: INSTALLING OLLAMA

1. Go to **ollama.com** in your browser.
2. Click the **download Ollama** link below the PowerShell command. This will download the Windows installer directly.



3. Run the installer when the download is complete. Ollama installs automatically with no additional configuration required.
4. Once installation is complete, Ollama runs silently in the background. You will see a small llama icon appear in your Windows system tray in the bottom right corner of your screen.

Verify the Installation

Open a Command Prompt window and run the following command to confirm Ollama installed correctly:

```
ollama --version
```

You should see a version number printed below the command, such as:

```
ollama version 0.17.6
```

Issue: The command returns "ollama is not recognized as an internal or external command."

Solution: Close the Command Prompt window and open a new one. The installer updates your system PATH, but an existing Command Prompt session will not pick up the change until you restart it.

Issue: The Ollama icon does not appear in the system tray after installation.

Solution: Try opening Ollama manually by searching for it in the Start menu. If it still does not appear, restart your computer and try again.

SECTION 4: DOWNLOADING YOUR FIRST MODEL

Ollama does not come with any models pre-installed. You download the ones you want using the pull command. This section covers choosing a model and pulling it to your machine.

Choosing a Model

The right model depends on your hardware and what you plan to use it for. The table below summarizes the four models covered in this guide.

Model	Size	Min. RAM	Best For
`Llama 3.2 (3B)`	2.0 GB	8 GB	General use, writing, Q&A, fast responses
`Mistral (7B)`	4.1 GB	8 GB	Code, technical tasks, reasoning, efficiency
`Gemma 3 (4B)`	3.3 GB	8 GB	Chat, summarization, long documents, instruction following
`DeepSeek-R1 (8B)`	4.9 GB	8 GB	Logic, analysis, step-by-step reasoning

Tip

The full list of available models is at ollama.com/library. Each model page shows its size, recommended hardware, and what it is optimized for.

Pulling a Model

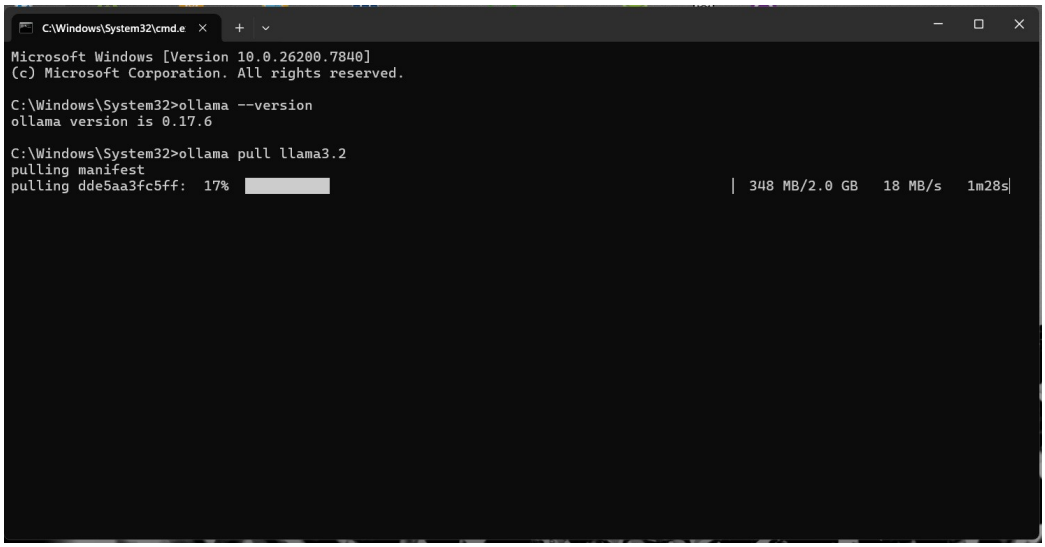
Use the pull command followed by the model name to download it. For example, to download Llama 3.2:

```
ollama pull llama3.2
```

To download Mistral instead:

```
ollama pull mistral
```

Ollama will show a progress bar as it downloads. Depending on your internet connection and the model size, this may take a few minutes.



When the download is complete, you will see a confirmation message and the prompt will return.

Issue: The download starts but stops partway through with a connection error.

Solution: Run the same pull command again. Ollama will resume from where it left off rather than starting over.

Issue: Ollama says the model name is not found.

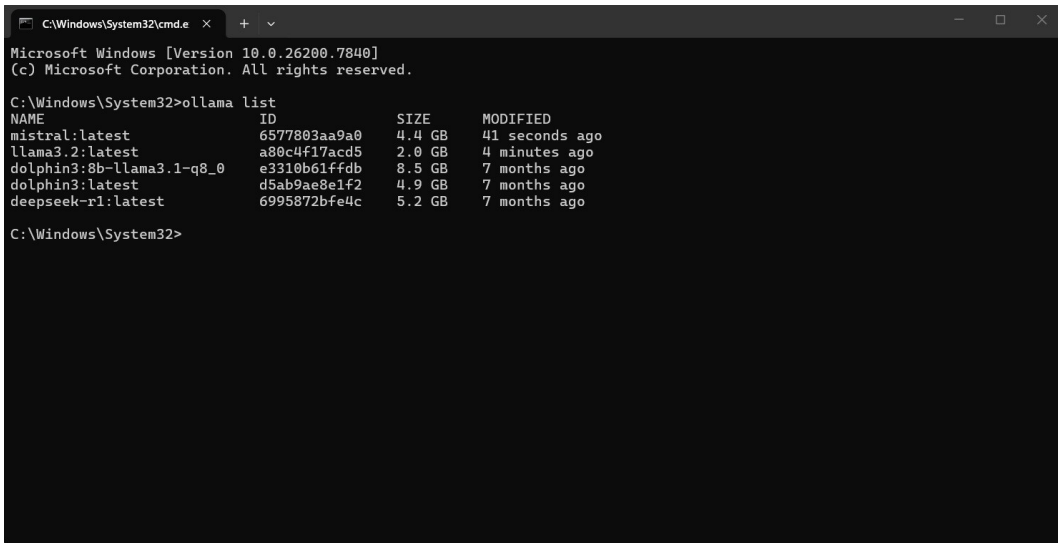
Solution: Check the spelling of the model name. Model names on the Ollama library are all lowercase. You can browse the full list at ollama.com/library to confirm the exact name.

Checking Your Downloaded Models

To see all models you have downloaded, run:

```
ollama list
```

This shows each model name, its size on disk, and when it was last used.



Issue: ollama list shows no models even though you ran pull successfully.

Solution: Run the pull command again. If the first pull was interrupted before completing, the model may not have been saved. A completed pull always ends with a confirmation message.

SECTION 5: RUNNING A MODEL

With a model downloaded, you can start a conversation with it directly from the command line using the run command.

Starting a Conversation

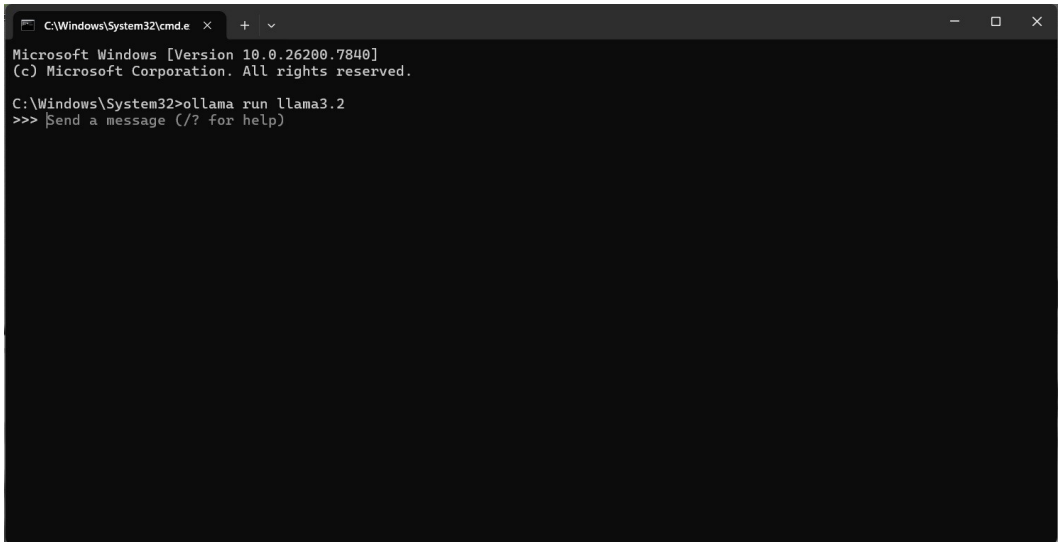
To start a session with Llama 3.2, run:

```
ollama run llama3.2
```

To use Mistral instead:

```
ollama run mistral
```

Ollama will load the model, which may take a few seconds, and then display a prompt where you can type your message.



> **Tip**
> The first time you run a model after downloading it, loading takes slightly longer while Ollama sets up its internal cache. Subsequent runs are faster.

Example Prompts to Try

Type any of the following into the prompt to see how the model responds. Press Enter after each one to send it.

General knowledge

```
Explain how a transformer neural network works in plain language.
```

Writing assistance

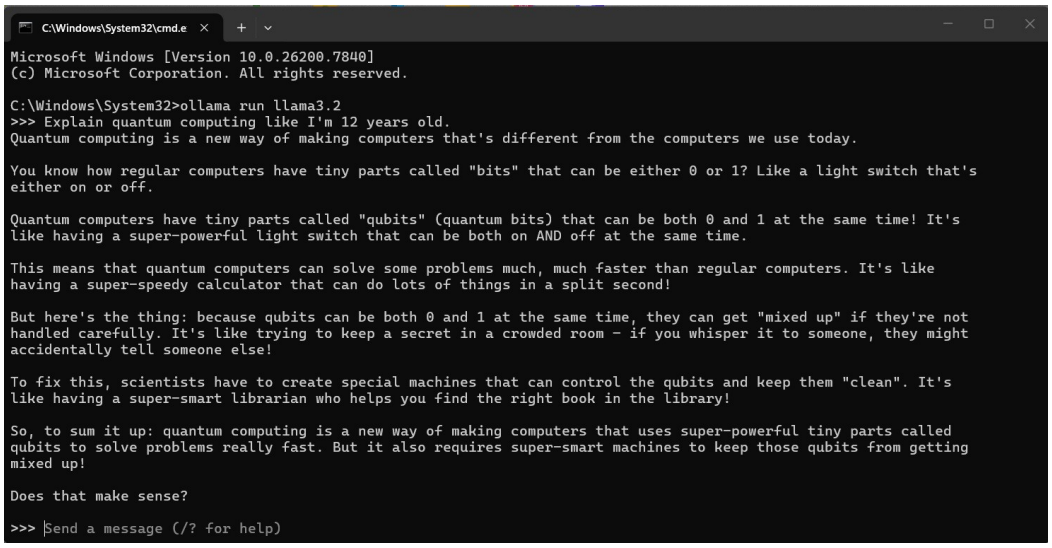
```
Write a professional email declining a meeting request while suggesting an
alternative time.
```

Code help

```
Write a Python function that reads a CSV file and returns the column names and row count.
```

Summarization

```
Summarize the key arguments for and against algorithmic risk assessment in criminal sentencing in three bullet points each.
```



Issue: The model gives a very short or unhelpful response to a detailed prompt.

Solution: Try rephrasing your prompt with more context. Models respond better to specific, well-formed questions than to vague or very short inputs. Adding a sentence of context before your question often produces significantly better results.

Issue: The model response is cut off mid-sentence.

Solution: The model has hit its context limit. Try asking for a shorter response, or break your request into smaller parts.

Ending a Session

Type `/bye` and press Enter, or press **Ctrl + C**, to exit the session and return to the regular command prompt.

Issue: Pressing Ctrl + C does not exit the session.

Solution: Try typing /bye and pressing Enter instead. If neither works, close the Command Prompt window entirely and open a new one.

Passing a Single Prompt Without Opening a Session

If you just want a quick answer without entering an interactive session, you can pass your prompt directly on the command line:

```
ollama run llama3.2 "What is the difference between supervised and unsupervised learning?"
```

Ollama will print the response and return to the prompt without opening an interactive session.

Issue: The single prompt command returns an error about unexpected arguments.

Solution: Make sure your prompt is wrapped in double quotes. If your prompt contains double quotes itself, escape them with a backslash or use an interactive session instead.

Issue: The model loads but then hangs and produces no output.

Solution: This usually means the model is too large for your available RAM. Press Ctrl + C to cancel, then try a smaller model. Llama 3.2 (3B) and Mistral are both good choices for machines with 8GB RAM.

Issue: Responses are extremely slow, taking more than a minute per response.

Solution: Ollama is likely running on your CPU rather than your GPU. This is normal if your machine does not have a compatible GPU. Response time varies significantly by hardware. For faster local inference, a machine with an NVIDIA GPU and at least 8GB VRAM will perform substantially better.

SECTION 6: USING THE OLLAMA API

Ollama runs a local REST API on your machine at port 11434. This means you can send prompts and receive responses programmatically, which opens up a wide range of possibilities including building your own tools, automating tasks, and integrating Ollama into existing workflows.

The API is available as long as Ollama is running, which it does automatically in the background after installation. You do not need to start anything extra.

Checking That the API Is Running

Open a browser and go to:

```
http://localhost:11434
```

If Ollama is running, the page will display the text “Ollama is running.” If you see a connection error, open the Ollama application from your Start menu to start it. Make sure the URL starts with http and not https — Ollama does not use a secure connection on localhost.

Sending a Request with curl

curl is a command line tool for making HTTP requests. On Windows 10 version 1803 and later, curl is included by default. All curl commands in this section are typed into Command Prompt, not a browser. Open a new Command Prompt window the same way you did in Section 2, and make sure Ollama is still running in the background before you start.

The following command sends a prompt to the Llama 3.2 model and streams the response back to the terminal:

```
curl http://localhost:11434/api/generate -d
"{\"model\": \"llama3.2\", \"prompt\": \"Explain the concept of overfitting in
machine learning.\"}"
```

The response comes back as a series of JSON objects, one per token, streamed in real time. Each object looks something like this:

```
{"model":"llama3.2","created_at":"2024-01-
01T00:00:00Z","response":"Over","done":false}
```

To get a single complete response instead of a stream, add the stream parameter set to false:

```
curl http://localhost:11434/api/generate -d
"{\"model\": \"llama3.2\", \"prompt\": \"What is regularization?\", \"stream\":
false}"
```

Tip

The stream false option is easier to work with when you are exploring the API, because it returns the full response as one JSON object rather than many small ones.

Issue: curl is not recognized as a command.

Solution: Your Windows version may be older than 1803. Download curl for Windows from curl.se/windows and add it to your system PATH, or upgrade to a newer version of Windows.

Issue: The curl command returns a 404 or model not found error.

Solution: Check that the model name in your request exactly matches a model you have downloaded. Run

ollama list to confirm the correct name.

Using the API with Python

The Ollama Python library is the cleanest way to interact with the API from Python. Install it with pip:

```
pip install ollama
```

Basic prompt and response

The following script sends a single prompt and prints the response:

```
import ollama

response = ollama.chat(
    model="llama3.2",
    messages=[
        {"role": "user", "content": "What is the difference between precision and
recall?"}
    ]
)

print(response["message"]["content"])
```

Issue: The script runs but prints nothing.

Solution: Check that the key path matches. Try printing the full response object first with print(response) to see its structure, then adjust your key access accordingly.

Multi-turn conversation

To maintain context across multiple messages, pass the full conversation history in the messages list:

```
import ollama

messages = [
    {"role": "user", "content": "Explain what a confusion matrix is."},
    {"role": "assistant", "content": "A confusion matrix is a table used to..."},
    {"role": "user", "content": "Can you show me an example in Python?"}
]

response = ollama.chat(model="llama3.2", messages=messages)
print(response["message"]["content"])
```

Issue: The model seems to forget earlier parts of the conversation.

Solution: The model only knows what you pass in the messages list. If you are running multiple separate script calls, you need to carry the full conversation history forward yourself by appending each exchange to the messages list before the next call.

Streaming responses

For longer responses, streaming lets you display output as it is generated rather than waiting for the full response:

```
import ollama

stream = ollama.chat(
    model="mistral",
    messages=[{"role": "user", "content": "Write a function that parses JSON safely
in Python."}],
    stream=True
)

for chunk in stream:
    print(chunk["message"]["content"], end="", flush=True)
```

Issue: The streaming output appears all at once instead of word by word.

Solution: Make sure flush=True is included in your print call. Without it, Python buffers the output and releases it all at once rather than printing each token as it arrives.

Using a system prompt

A system prompt sets the context or persona for the model before the conversation begins. This is useful for giving the model a specific role or set of instructions:

```
import ollama

response = ollama.chat(
    model="llama3.2",
    messages=[
        {
            "role": "system",
            "content": "You are a technical writing assistant. Respond clearly and
concisely."
        },
        {
            "role": "user",
            "content": "Review this sentence for clarity: The system was utilized to
facilitate the processing of requests."
        }
    ]
)

print(response["message"]["content"])
```

Issue: The model ignores the system prompt and responds as if it were not there.

Solution: Some smaller models have limited ability to follow system prompts reliably. Try making the instruction more explicit, or switch to a larger model. Mistral and Llama 3.2 both handle system prompts well.

Issue: pip install ollama fails with a permissions error.

Solution: Try running the command with the user flag: pip install ollama --user. If you are using a virtual environment, make sure it is activated before running pip.

Issue: The API returns a connection refused error.

Solution: Ollama is not running. Look for the llama icon in your system tray. If it is not there, open Ollama from the Start menu to start it, then try again.

SECTION 7: MANAGING YOUR MODELS

As you experiment with different models, you will accumulate files on your disk. These commands help you keep things organized.

Useful Commands

List downloaded models

```
ollama list
```

Remove a model

```
ollama rm mistral
```

Replace mistral with the name of the model you want to remove. This frees up the disk space the model was using.

View model details

```
ollama show llama3.2
```

This shows the model's parameter count, quantization, context length, and other technical details.

Pull a model update

```
ollama pull llama3.2
```

Running pull on a model you already have will check for updates and download a newer version if one is available.

> **Tip**
> Model files are stored in C:\Users\YourUsername\.ollama\models on Windows. If you are running low on disk space on your C drive, you can move this folder to another drive by setting the OLLAMA_MODELS environment variable to a new path before launching Ollama.

> **Issue:** ollama rm reports that the model does not exist.
> **Solution:** Run ollama list to confirm the exact model name. The name must match exactly, including any version tags.

> **Issue:** ollama show returns no output or an error.
> **Solution:** Make sure you have the model downloaded locally. ollama show only works on models that are already on your machine, not models that exist on the library but have not been pulled yet.

> **Issue:** Disk space does not appear to free up after removing a model.
> **Solution:** Windows may not immediately reflect the freed space in File Explorer. Try restarting Ollama or waiting a few minutes. You can also run ollama list to confirm the model is gone.

SECTION 8: ADDING A BROWSER INTERFACE WITH OPEN WEBUI

The command line interface is practical, but if you prefer working in a browser, Open WebUI gives you a full chat interface for your local models that looks and feels similar to ChatGPT. It connects directly to your running Ollama instance.

What You Will Need

- Docker Desktop for Windows, available at docker.com/products/docker-desktop.
- Ollama already installed and running, as covered in this guide.

Installing Docker Desktop

Open WebUI runs inside a Docker container, so you need to install Docker Desktop before you can use it. Docker Desktop is free for personal use.

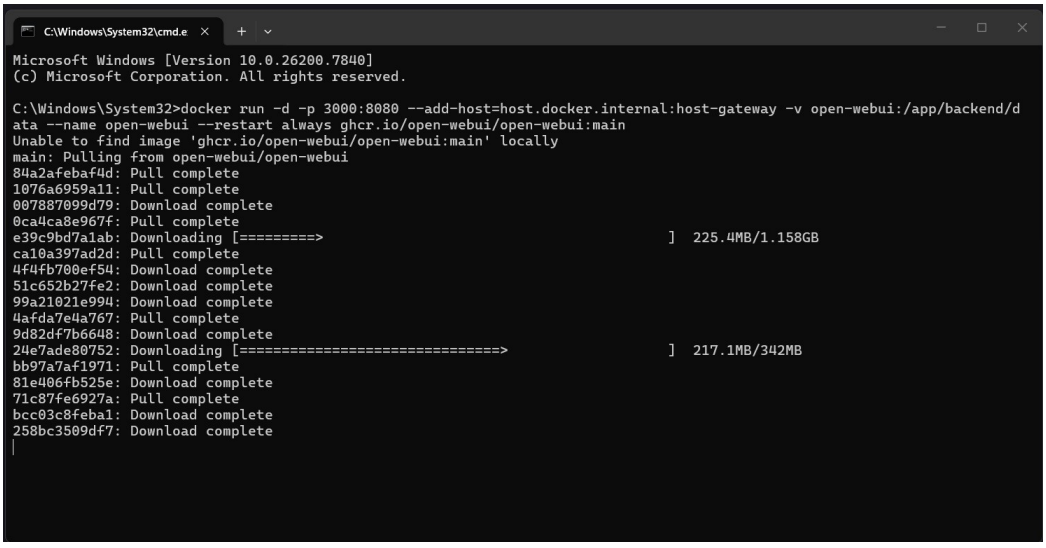
5. Go to `docker.com/products/docker-desktop` in your browser and click **Download for Windows**.
6. Run the installer and follow the on-screen instructions. When the Configuration dialog appears, leave the default settings as they are: **Use WSL 2 instead of Hyper-V** should be checked, **Allow Windows Containers** should be unchecked. Click **OK** to continue.
7. Restart your computer when the installer finishes.
8. After restarting, open **Docker Desktop** from the Start menu and wait for it to finish starting up. You will see a whale icon in your system tray when it is ready. Do not run the `docker run` command until Docker Desktop is fully running.

Installing Open WebUI

With Docker Desktop installed and running, open a Command Prompt and run the following command:

```
docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main
```

Docker will download the Open WebUI image and start the container. This may take a few minutes the first time.

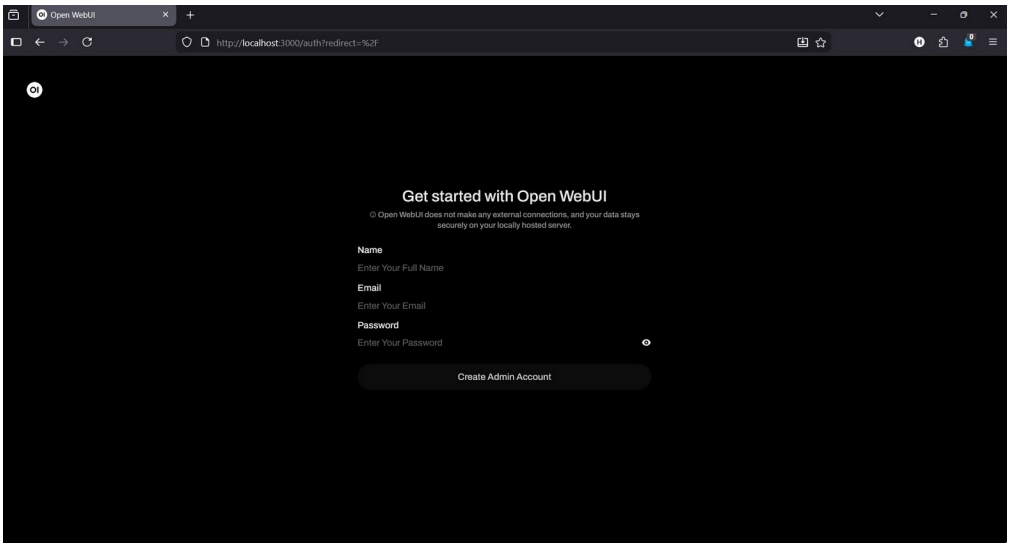


Accessing Open WebUI

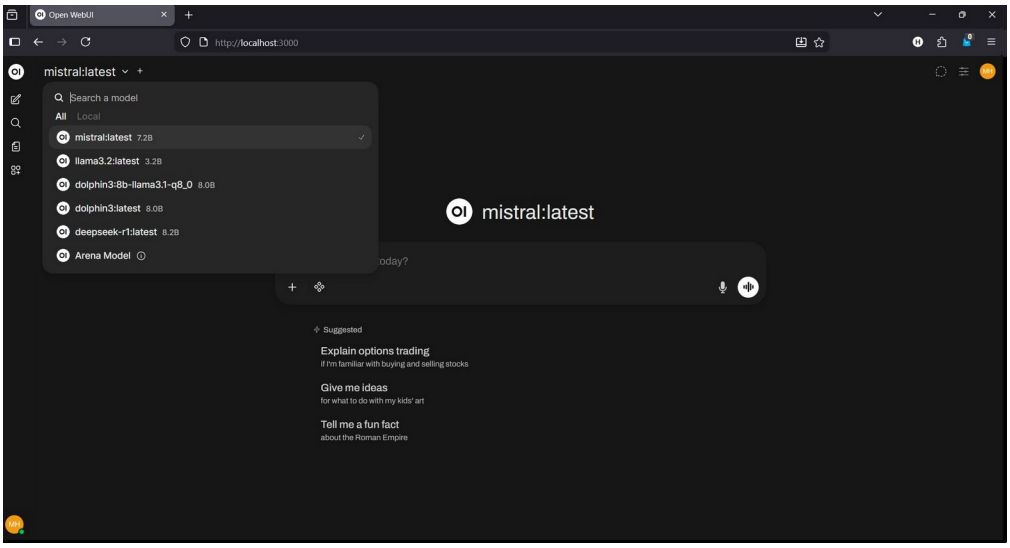
Once the container is running, open a browser and go to:

```
http://localhost:3000
```

You will see a “Get started with Open WebUI” screen on your first visit. Fill in your name, email, and a password, then click **Create Admin Account**. This account is stored entirely on your machine and is not connected to any external service.



After logging in, you will see a chat interface. Click the model selector at the top of the page to choose between your downloaded Ollama models and start a conversation.



Tip

Open WebUI persists your conversation history between sessions, which the command line interface does not. If you want to keep a record of your interactions with a model, Open WebUI is the better option.

Issue: The browser shows a connection refused error at localhost:3000.

Solution: The Docker container may not have started correctly. Run docker ps in Command Prompt to check whether the open-webui container is listed as running. If it is not, run docker start open-webui to start it.

Issue: Open WebUI loads but shows no models in the selector.

Solution: Make sure Ollama is running. Open WebUI connects to Ollama at localhost:11434 by default. If Ollama is not running, start it from the system tray or Start menu and then refresh the Open WebUI page.

QUICK REFERENCE

Essential Commands

Pull a model

```
ollama pull llama3.2
```

Run a model interactively

```
ollama run llama3.2
```

Run a model with a single prompt

```
ollama run llama3.2 "Your prompt here"
```

List downloaded models

```
ollama list
```

Remove a model

```
ollama rm modelname
```

Check Ollama version

```
ollama --version
```

API Endpoints

Generate a response

```
POST http://localhost:11434/api/generate
```

Chat with message history

```
POST http://localhost:11434/api/chat
```

List available models

```
GET http://localhost:11434/api/tags
```

Useful Resources

- Ollama documentation and model library: ollama.com
- Ollama Python library on PyPI: pypi.org/project/ollama
- Open WebUI documentation: docs.openwebui.com
- Ollama GitHub repository and issue tracker: github.com/ollama/ollama
- Windows command line reference: ss64.com/nt/