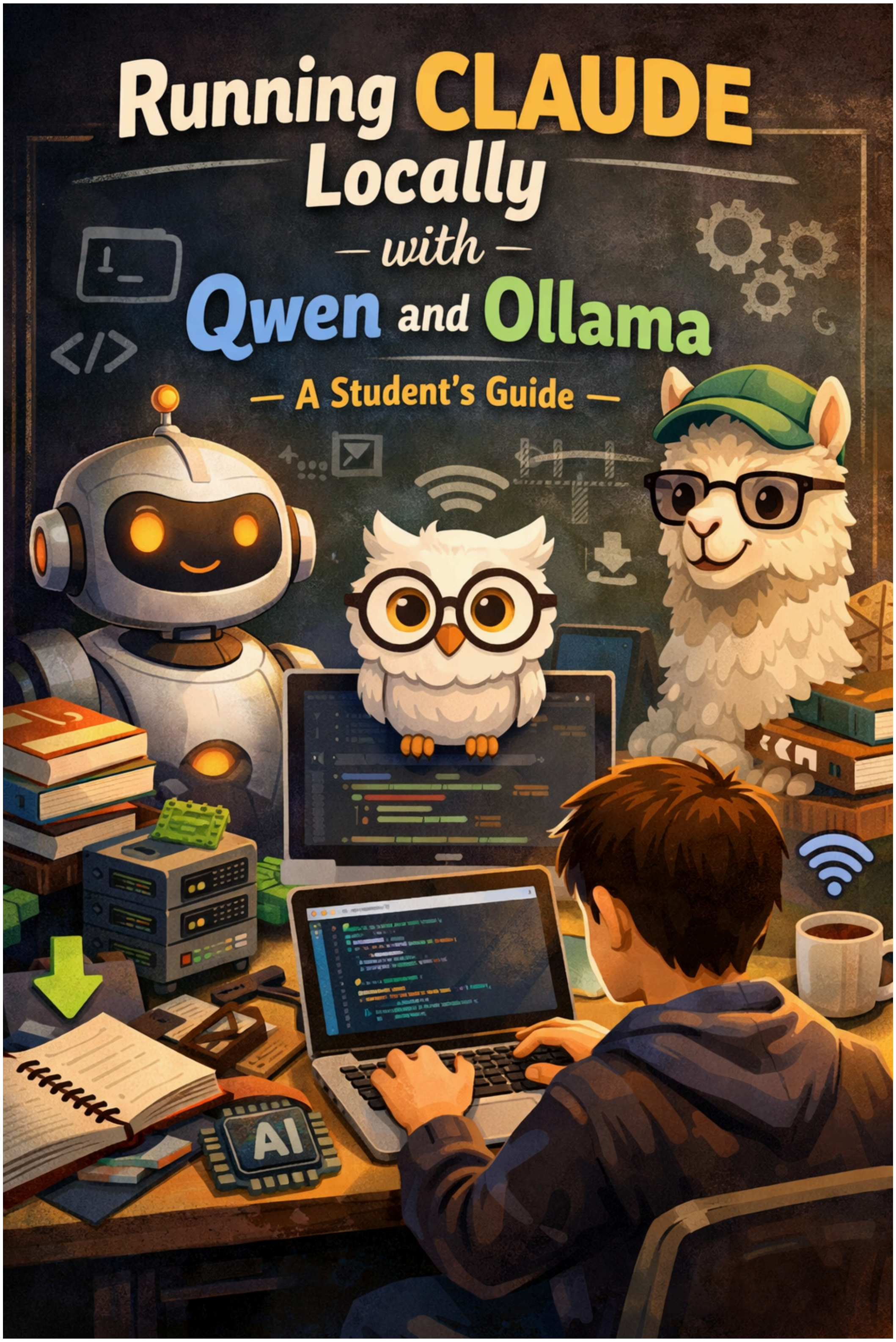

Running CLAUDE Locally with Qwen and Ollama

— A Student's Guide —

Free AI for Coding on Your Laptop

A Student's Guide to Running Qwen Locally in India

No subscription. No message limits. Works offline.

Created by Ajit Jaokar with the help of Claude Sonnet 4.6

Background

I created this guide for some students in India who I am mentoring. Its possible to run Claude for free in India (locally). I am myselves a power user of Claude in the UK but its not easy for students in India - hence this workaround. If you are testing this, please share your comments.

Although my student is in India I think this should work anywhere.

Essentially, this guide will help you set up a free, unlimited AI assistant on your laptop using Ollama and Qwen — an open-source AI model from Alibaba that is excellent for students. Once set up, you will have:

- No message limits — chat as much as you want, any time
- No subscription or payment required
- Works offline — no internet needed after setup
- 100% private — your conversations stay on your device

Who Is This For?

Students in India (or anywhere) who want a capable AI assistant for daily study and work.
People who find Claude Free, ChatGPT Free, or Gemini Free hits limits too quickly.
Anyone who wants a private, offline AI with no ongoing cost.

You need to understand Qwen from Alibababa and ollama

I recommend Path A ie an 8 GB laptop with Qwen2.5 7B to start with but also this enhanced option will be even better

- For **general coding help** — explaining concepts, debugging, writing functions, understanding error messages — Qwen2.5 7B is genuinely good and will serve a student well day-to-day.
- However, if the student is doing more **complex coding tasks** (multi-file reasoning, architectural decisions, harder algorithms), there's actually a better option within the same hardware constraint: **Qwen2.5-Coder 7B**. Alibaba released a coding-specific variant that is fine-tuned specifically on code, and it outperforms the base 7B model on coding tasks while requiring the same RAM.
- It supports 92 programming languages, has much stronger code completion, better at generating working code on the first try, and is better at reading and explaining existing code.
- There's also a **14B version** if they ever upgrade to a 16 GB laptop — noticeable jump in quality for harder problems.
- So the recommendation would be: same hardware plan, just swap the model from `qwen2.5:7b` to `qwen2.5-coder:7b`. Everything else in the guide (Ollama setup, Open WebUI, etc.) stays exactly the same. Worth updating that one line in the guide if you're sharing it with the student.

Step 0: Check Your Hardware First

Before installing anything, figure out how much RAM your laptop has. This determines which model to use.

How to Check Your RAM

Windows: Press Windows + R, type `msinfo32`, press Enter. Look for “Installed Physical Memory.”

Mac: Click the Apple icon → About This Mac → look for “Memory.”

Linux: Open terminal, type `free -h`, look at the “Mem:” total.

Which Model Should You Use?

RAM	Recommended Model	What to Expect
4 GB	Qwen2.5 1.5B or 3B	Basic help, simple questions. Slower.
8 GB	Qwen2.5 7B (recommended)	Good for essays, coding help, summaries. Sweet spot for students.
16 GB	Qwen2.5 14B	Strong reasoning. Handles complex topics well.

16 GB + GPU	Qwen2.5 32B or 72B	Excellent, close to Claude Sonnet quality.

💡 Recommendation for Most Students

If you have an 8 GB laptop (very common), use Qwen2.5 7B. It's the sweet spot: good quality, manageable download size (~4.7 GB), and runs smoothly without a dedicated GPU.

Path A: Local Setup with Ollama (Recommended)

Ollama is the easiest way to run AI models locally. It handles all the complexity for you — you just install it and pull the model you want.

Step 1: Install Ollama

Go to https://ollama.com and download the installer for your operating system.

Windows / Mac: Run the downloaded installer. It installs like any normal app.

Linux: Open your terminal and paste this command:

```
curl -fsSL https://ollama.com/install.sh | sh
```

Once installed, Ollama runs quietly in the background. You do not need to open a separate app — it is always ready.

Step 2: Download Your Model

Open a terminal (or Command Prompt on Windows) and type one of these depending on your RAM:

4 GB RAM:

```
ollama pull qwen2.5:3b
```

8 GB RAM (most students):

```
ollama pull qwen2.5:7b
```

16 GB RAM:

```
ollama pull qwen2.5:14b
```

> ⚠️ **Important: Download Size**
> The 7B model is about 4.7 GB to download. Make sure you are on Wi-Fi, not mobile data.
> The download only happens once. After that, the model runs entirely offline.
> If you are on a slow connection, start the download before bed and let it run overnight.

Step 3: Chat in the Terminal

Once downloaded, you can start chatting immediately from the terminal:

```
ollama run qwen2.5:7b
```

Type your question and press Enter. Type /bye to exit. That's it!

Step 4: Get a Better Interface (Optional but Recommended)

The terminal works, but a proper chat interface is much nicer. Open WebUI gives you a browser-based chat that looks and feels like Claude or ChatGPT.

Installing Open WebUI

You will need Docker installed for this. If you do not have Docker:

- Go to docker.com/products/docker-desktop and install Docker Desktop
- Restart your computer after installing

Then run this single command in your terminal:

```
docker run -d -p 3000:80 \
  --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
```

After it starts (takes 1-2 minutes), open your browser and go to:

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

Sign up with any email (this is local — no real account), select your Qwen model from the dropdown, and start chatting!

What Open WebUI Adds

✓ A clean chat interface in your browser

✓ Upload and chat with PDF files and documents

✓ Save your chat history

✓ Multiple conversation threads

✓ Supports multiple models (switch between them easily)

Path B: Groq Free API (For Weak Laptops or Fast Setup)

If your laptop has less than 4 GB RAM, or if you want something that works immediately without any setup, Groq is the best option. Groq hosts open-source models (including Qwen and Llama) on their own servers and gives you a very generous free tier.

Groq vs Local Ollama: Key Differences

Groq is cloud-based: your messages are sent to Groq's servers (not Anthropic's), so it's not private.

Groq is extremely fast: responses are near-instant, much faster than running locally.

Groq is free with generous daily limits: much more than Claude Free or ChatGPT Free.

No local GPU or RAM requirements: works on any laptop with internet.

Getting Started with Groq

1. Go to console.groq.com and sign up for a free account
2. Click 'Create API Key' and copy the key
3. Use it in any AI app that supports custom API endpoints

The easiest way to use Groq is through a simple chat interface. Recommended options:

- TypingMind (typingmind.com) — paste your Groq API key and chat
- Open WebUI (see Path A) — also supports Groq as a backend
- Write a simple Python script (see below)

Simple Python Script to Chat with Groq

If you know basic Python, this gets you started in under 5 minutes:

```
pip install groq

# Then create a file called chat.py:
from groq import Groq

client = Groq(api_key='your_api_key_here')

while True:
    user_input = input('You: ')
    if user_input.lower() == 'quit':
        break
    response = client.chat.completions.create(
        model='llama-3.3-70b-versatile',  # or 'qwen-qwq-32b'
        messages=[{'role': 'user', 'content': user_input}]
    )
    print('AI:', response.choices[0].message.content)
```

Path C: Google Colab (Free Cloud GPU)

If you have a Google account, you can run Qwen on Google's servers for free using Colab. This gives you access to a GPU without owning one.

4. Go to colab.research.google.com
5. Create a new notebook
6. In a code cell, paste and run:

```
# Install Ollama inside Colab
!curl -fsSL https://ollama.com/install.sh | sh
import subprocess, time
subprocess.Popen(['ollama', 'serve'])
time.sleep(5)
!ollama pull qwen2.5:7b

# Now chat
import requests
response = requests.post('http://localhost:11434/api/generate',
    json={'model': 'qwen2.5:7b', 'prompt': 'Explain photosynthesis simply.',
'stream': False})
print(response.json()['response'])
```

⚠️ Colab Limitations

Free Colab sessions time out after ~12 hours and you lose your work.
You must re-download the model each session (use Colab Pro to persist storage).
Best for experimenting or one-off tasks, not daily use.

Quick Comparison: Which Option is Right for You?

Feature	Claude Free	Local Qwen (Ollama)	Groq Free API
Cost	Free	Free	Free
Message limit	~30-100/day	Unlimited	Very generous
Internet needed?	Yes	No (after setup)	Yes
Model quality	Claude Sonnet (excellent)	Good to very good	Very good (Llama/Qwen)
Privacy	Data may train model	100% private, local	Sent to Groq servers
Setup difficulty	None	Easy (15 min)	Very easy
Best for	Occasional use	Heavy daily use, privacy	Weak laptops, speed

🏆 Recommendation Summary

8 GB laptop, want daily use with no limits → Path A (Ollama + Qwen 7B)
Weak laptop or want instant setup → Path B (Groq Free API)
Curious about cloud GPU or experimenting → Path C (Google Colab)
Just want occasional help → Use Claude Free at claude.ai

Tips for Getting the Most Out of Your AI Assistant

Writing Better Prompts

The quality of AI responses depends a lot on how you ask. Here are patterns that work well:

- Be specific about what you need
  - Instead of: “Explain photosynthesis”
  - Try: “Explain photosynthesis in simple terms, as if explaining to a 15-year-old. Include an analogy.”

- Tell it your level
  - Add “I am a second-year engineering student” or “I know basic Python” so it calibrates its response.

- Ask it to check your work
  - “Here is my essay introduction. Give me feedback on clarity and argument structure.”

- Ask for step-by-step explanations
  - “Explain this concept step by step and give me an example after each step.”

What It's Good At (for Students)

- Explaining difficult concepts from textbooks
- Helping debug code and explaining why bugs occur
- Drafting and improving essays, reports, and emails
- Summarising long research papers
- Creating study notes and flashcards from your notes
- Solving and explaining math problems step by step
- Translating content or explaining English idioms

Where to Be Careful

- Do not rely on it for very recent facts (its training data has a cutoff date)
- Always verify important medical, legal, or financial information with real sources
- For academic work, use it to understand concepts — not to write assignments you submit as your own

---

Troubleshooting Common Issues

Problem: “Ollama: command not found”

Solution: Restart your terminal after installing Ollama. On Windows, close and reopen Command Prompt.

Problem: Model runs very slowly

Solution: Try a smaller model. Switch from 7B to 3B. Also close other apps (browser tabs, etc.) to free up RAM.

Problem: Download stuck or very slow

Solution: Ollama download resumes automatically if interrupted. Just run the same pull command again.

Problem: Open WebUI won’t start

Solution: Make sure Docker Desktop is running first. Check that Ollama is also running (open a terminal and type: ollama serve).

Problem: Model gives wrong or confusing answers

Solution: Rephrase your question more specifically. Add context. For factual questions, verify with a textbook or trusted source.

A Final Note

The gap between free local models and paid AI services is closing fast. Qwen2.5 7B running on your own laptop is genuinely useful for the vast majority of student tasks. You will notice a difference in quality on very complex reasoning tasks compared to Claude Pro or GPT-4, but for everyday study help, writing, and coding, a local setup works very well.

As your needs grow, you can always upgrade to larger models or explore the free tiers of cloud services. The important thing is that you have a solid, unlimited, private AI assistant available right now — at zero cost.

Quick Links to Get Started

Ollama: https://ollama.com

Open WebUI: https://openwebui.com

Groq Console: https://console.groq.com

Google Colab: https://colab.research.google.com