AIAppSecAcademy

> RESOURCE GUIDE / LOCAL AI

How to Use AI Without Token Costs

The best open-weight models to run locally, and the agent software to pair with each job

28 jobs mapped to a model and an agent

2 hardware tiers side by side

GPU needs and license for every pick

Dr. Viswanath Chirravuri

November 14, 2026

aiappsecacademy.com
Free AI & AppSec Knowledge Hub

Contents

01 The short answer . . . 3
02 What "free" and "open" really mean . . . 3
03 What your hardware can run . . . 4
04 The table: best open model and agent software for each job . . . 5
Software engineering . . . 5
Everyday work . . . 5
Smaller and larger machines . . . 6
Images, video and voice . . . 7
05 Software engineering: why these picks . . . 7
What changes at 16 GB . . . 8
Choosing the agent . . . 8
Claude Code on a local model . . . 8
A working setup in three steps . . . 9
06 Everyday work: slides from your template, documents, and a general assistant . . . 9
Route 1: Presenton, for a point-and-click workflow . . . 9
Route 2: a coding agent with python-pptx, for exact brand fidelity . . . 10
Claude Desktop on a local model . . . 10
Documents and a general assistant . . . 10
07 Beyond text: documents, images, video, voice and search . . . 10
08 The runtime: what actually loads the model . . . 11
09 Starter stacks by machine . . . 12
10 Caveats: where local still falls short, and how to stay safe . . . 12
Supply-chain safety for downloaded models . . . 13
11 Sources . . . 13

Snapshot as of November 14, 2026. Open-weight models turn over quickly, so check the license and the current release before you build on any pick in this guide. Benchmark scores quoted here are mostly vendor-reported; the independent tests behind the picks are small.

01 The short answer

You can run a coding agent, a slide generator, a document reader, an image generator and a voice stack on hardware you own, with no per-token bill. The models are free to download, the agent software is open source, and the only meter running is your electricity.

This guide gives picks for two machines side by side. On a 24 GB GPU or a 32 GB Mac, the pick is **Qwen3.8-27B**. On a 16 GB GPU or laptop, it is **gpt-oss-20b** for coding and **Gemma 4 12B** for everyday work. On either machine, pair the model with **OpenCode** for software engineering and **Presenton** for PowerPoint decks built from your own template. One model, several agents: the model supplies the reasoning, and the agent software decides what job it does.

If you already use Claude Code or Claude Desktop, you can keep them: both run on the same local model through Ollama, with caveats covered below.

QUICK PICKS BY MACHINE

> **No GPU:** Gemma 4 E4B or Phi-4-mini for chat and summaries, Whisper for dictation, Kokoro for narration. Skip agentic coding at this tier.
>
> **16 GB GPU or laptop:** gpt-oss-20b for coding agents, Gemma 4 12B for chat, slides and documents. Agents work if you keep tasks small.
>
> **24 GB GPU or 32 GB Mac:** Qwen3.8-27B for nearly everything, with Gemma 4 31B for tool-driven automation and reading images. This is where local agents start to feel dependable.
>
> **Team server:** DeepSeek-V4.1-Flash or GLM-5.3-Flash, both MIT licensed, served through vLLM.

One caveat up front: open models still trail the best hosted ones, and the gap is widest at laptop sizes. The Caveats section covers where that matters.

02 What "free" and "open" really mean

"No token costs" is accurate, but "free" is not. You swap a usage bill for three other costs: hardware bought up front, electricity, and your own setup time. The trade pays off when you use AI heavily, handle data that should not leave your network, or want a tool that keeps working offline.

"Open" also covers two different things, and the license column in the tables below tells you which one you are getting.

- **Permissive open weights (Apache 2.0, MIT).** Download, modify, and use commercially with no royalty. Qwen3.8-27B, Gemma 4, gpt-oss and the DeepSeek V4 family sit here.
- **Open weights with a custom license.** You can download and run them, but the terms carry conditions. Examples: Kimi K3 and GLM-5.3 ship under their own licenses, MiniMax M3 is non-commercial by default, FLUX.2 [dev] is non-commercial, and LTX-2.5 is free only for companies under $10M in annual revenue (Morph, LTX).

A third trap is specific to this guide's promise. Several of the strongest open-weight models are too large for a home machine, so tools list them as cloud-hosted options. A check of Ollama's library in early November found Kimi K3,

GLM-5.3, MiniMax-M3 and DeepSeek-V4.1-Flash available only as cloud tags, while Qwen3.8 pulls as an 18 GB local file (DevToolLab). A cloud tag sends your prompts to someone else's server, so confirm a model is a local download before you trust it with private work.

03 What your hardware can run

Memory decides everything. A model must fit in GPU memory (VRAM) to run fast, and on Apple Silicon the GPU shares the Mac's unified memory, so total RAM is the number that counts. Budget about 0.5 GB per billion parameters for 4-bit weights, and 0.6 to 0.7 GB once the context cache and runtime are included (DevToolLab).

Tier	Typical machine	Models that fit (download size)	Realistic use
CPU only	Any laptop, 8 to 16 GB RAM	Gemma 4 E2B (1.5 GB), Phi-4-mini (2.3 GB)	Chat, summaries, dictation, narration. Roughly 4 to 15 tokens per second.
8 GB GPU	Older gaming card, 16 GB Mac	MiMo-V2.6 9B (6.6 GB), Granite 4.2 8B (5.3 GB), Qwen3.5 9B	Coding help on small repositories, light tool use.
16 GB GPU or Mac	RTX 4060 Ti 16 GB, 16 to 24 GB Mac	Gemma 4 12B (8 GB), gpt-oss-20b (14 GB), Devstral Small 2	Entry-level coding agents and everyday assistant work.
24 GB GPU or 32 GB Mac	RTX 3090 or 4090	Qwen3.8-27B (18 GB), Gemma 4 26B A4B (16 to 19 GB), Gemma 4 31B (19 to 20 GB), Muse Glimmer 30B (18 GB)	The sweet spot: dependable agents, vision, slides.
32 to 64 GB	RTX 5090, two RTX 3090s, 64 GB Mac	Flint 40B (22.9 GB), Qwen3-Coder-Next (about 48 GB)	Long-context agent work on large repositories.
80 to 128 GB	One 80 GB GPU, 128 GB Mac	gpt-oss-120b (65 GB), Qwen3.8-Flash-Next (105 to 120 GB)	Near-server quality on a single box.
Multi-GPU server	Data-center GPUs	GLM-5.3-Flash, DeepSeek-V4.1-Flash, Kimi K3 (about 1.4 TB of weights)	Shared team endpoint.

Sizes come from DevToolLab, Hashnode and Kilo; CPU speeds from PromptQuorum.

Three things the table hides:

- **Fitting is not the same as working.** Flint loads on a 24 GB card but leaves almost no room for context, so a 32 GB GPU or a larger Mac suits it better.
- **Agents need long context.** Use a model with tool calling and at least 64K of context for agent work. Ollama defaults to 4K when the machine has under 24 GiB of VRAM, so raise it yourself (Hashnode).
- **CPU speed depends on memory bandwidth.** Core count matters less than how fast the processor can read weights from RAM, which is why CPU-only stays in small-model territory (InsiderLLM).

04 The table: best open model and agent software for each job

Each row is a job, not a model. The first two tables give a pick for each of two machines: a 24 GB GPU or 32 GB Mac, and a 16 GB GPU or laptop. The same model appears several times because the agent software changes with the task: Qwen3.8-27B writes code inside OpenCode, builds decks inside Presenton, and answers questions over your files inside AnythingLLM.

Picks are made on one criterion: how well the model does that job at a size you can run locally. Where a lab is based plays no part. The winners come from Alibaba, Google, OpenAI, Mistral, Xiaomi, IBM, Baidu, Black Forest Labs, Lightricks and others, and where an independent hands-on test exists it outranks the vendor's own benchmark table.

Software engineering

Use case	Pick on a 24 GB GPU or 32 GB Mac	Pick on a 16 GB GPU or laptop	Agent software	GPU needed?	Model license
Agentic coding in the terminal: multi-file edits, running tests	Qwen3.8-27B	gpt-oss-20b	OpenCode	Yes	Apache 2.0
Agentic coding in Claude Code's terminal UI, with the model swapped for a local one	Qwen3.8-27B	gpt-oss-20b	Claude Code, started with `ollama launch claude`	Yes	Apache 2.0
Agentic coding inside VS Code, approving each step	Qwen3.8-27B	Devstral Small 2	Cline	Yes	Apache 2.0
Git-native pair programming with a commit per change	Qwen3.8-27B	Devstral Small 2	Aider	Yes	Apache 2.0
Autonomous issue-to-pull-request runs in a sandbox	Qwen3.8-27B	gpt-oss-20b, on small and well-scoped issues	OpenHands	Yes	Apache 2.0
Inline autocomplete in the editor	Qwen2.5-Coder 7B	Qwen2.5-Coder 7B	Tabby	Yes 8 GB VRAM is enough	Apache 2.0

Everyday work

Use case	Pick on a 24 GB GPU or 32 GB Mac	Pick on a 16 GB GPU or laptop	Agent software	GPU needed?	Model license
PowerPoint deck generated from your own template	Qwen3.8-27B	Gemma 4 12B	Presenton	Yes	Apache 2.0
PowerPoint with exact brand fidelity, scripted against the template file	Qwen3.8-27B	gpt-oss-20b	OpenCode, Claude Code or Goose writing a python-pptx script	Yes	Apache 2.0
Chat, file tasks and coding in the Claude Desktop app, running on a local model	Qwen3.8-27B	Gemma 4 12B	Claude Desktop, connected through Ollama's Claude toggle	Yes	Apache 2.0

Use case	Pick on a 24 GB GPU or 32 GB Mac	Pick on a 16 GB GPU or laptop	Agent software	GPU needed?	Model license
General chat, writing, summarizing	Gemma 4 26B A4B	Gemma 4 12B	Open WebUI or Jan	Yes	Apache 2.0
Desktop automation through declared tools: files, browser, apps	Gemma 4 31B	Gemma 4 12B	Goose with MCP extensions	Yes	Apache 2.0
Chat with your own documents (local RAG)	Qwen3-Embedding 0.6B to retrieve, Qwen3.8-27B to answer	Qwen3-Embedding 0.6B to retrieve, Gemma 4 12B to answer	AnythingLLM	Embedding runs on CPU; the answering model needs the GPU	Apache 2.0
Reading screenshots, charts and photos	Gemma 4 31B	Gemma 4 12B	Open WebUI	Yes	Apache 2.0
OCR and PDF-to-Markdown conversion	PaddleOCR-VL-1.6	PaddleOCR-VL-1.6	PaddleOCR's own pipeline; MinerU or Docling as alternatives	Optional Tesseract with OCRmyPDF covers CPU-only machines	Apache 2.0

Smaller and larger machines

Use case	Machine	Best open model	Agent software	GPU needed?	Model license
Assistant on a laptop with no GPU	8 to 16 GB RAM	Gemma 4 E4B or Phi-4-mini	Jan	No	Apache 2.0 / MIT
Chat-style coding help (too small for dependable agent loops)	8 GB GPU	MiMo-V2.6 9B or Granite 4.2 8B	Aider	Yes	MIT / Apache 2.0
Long-context agent work on large repositories	48 GB VRAM or a 64 GB Mac; Flint from 32 GB	Qwen3-Coder-Next (about 80B) or Flint (40B)	Kilo Code	Yes	Apache 2.0
Shared coding endpoint for a team	Multi-GPU server	DeepSeek-V4.1-Flash or GLM-5.3-Flash	OpenCode or OpenHands, served by vLLM	Yes	MIT
Highest-ranked open model, cost no object	Data-center cluster	Kimi K3	OpenCode or OpenHands, served by vLLM	Yes	Kimi K3 License (custom)

Images, video and voice

Use case	Best open model	Software	GPU needed?	Model license
Slide art and illustrations	Z-Image Turbo or FLUX.2 [klein] 4B	ComfyUI	Yes About 16 GB VRAM; quantized builds fit 12 GB	Apache 2.0
Images with readable text, such as title slides and posters	Qwen-Image	ComfyUI	Yes 16 to 24 GB VRAM	Apache 2.0
Highest image quality for personal projects	FLUX.2 [dev]	ComfyUI	Yes 24 GB quantized, about 32 GB at FP8	Non-commercial
Short video clips	Wan 2.2	ComfyUI	Yes About 17 to 20 GB VRAM with optimized workflows	Apache 2.0
Video with synchronized audio	LTX-2.5	ComfyUI	Yes 32 GB VRAM with FP8	LTX License (free under $10M revenue)
Transcription and dictation	Whisper large-v3-turbo	whisper.cpp, or OpenWhispr for dictation into any app	No Runs on CPU and Apple GPUs	MIT
Highest-accuracy transcription	Qwen3-ASR-1.7B	vLLM	Yes	Apache 2.0
Narration and text-to-speech	Kokoro-82M	Kokoro's own Python package	No Runs on a laptop CPU	Apache 2.0
Voice cloning	Chatterbox	Chatterbox's own Python package	Yes About 4 to 6 GB VRAM	MIT

Rankings and hardware figures come from the pages listed under Sources, checked on November 14, 2026. Benchmark scores quoted later are mostly vendor-reported.

05 Software engineering: why these picks

Qwen3.8-27B wins the local coding rows on independent evidence, not just its own model card. In a hands-on 24 GB GPU comparison it passed 12 of 12 coding-agent tasks on the first attempt and kept working at context sizes where Gemma 4 31B ran out of memory (Kingy). A separate developer's 30-task agent benchmark had it passing all 30, against 24 for Muse Glimmer (H2S Media). Both tests are small, so treat them as signals and re-test on your own repository.

The model is a dense 27B vision-language model from Alibaba, released in August 2026 under Apache 2.0, with a 262K native context and built-in tool calling (Qwen on GitHub, LM Studio). Its 4-bit build is an 18 GB download.

Three findings shaped the other rows:

- **Vendor scores do not transfer between model cards.** Gemma 4 31B is listed at 52.0 on SWE-bench Verified in Qwen's card and 66.6 in Meta's, same model and same benchmark name (H2S Media). Qwen's 61.7 on SWE-

bench Pro, Meta's 76.0 on SWE-bench Verified for Muse Glimmer and Mistral's 68.0 for Devstral Small 2 are each self-reported under a different harness.

- **Tool calling is a separate skill from coding.** The same Kingy test found Gemma 4 31B perfect on strict, declared tool calls (90 of 90 single-step, 30 of 30 multi-step) but much less reliable inside an iterative coding loop. That is why Gemma takes the desktop-automation row and Qwen takes the coding rows.
- **Small models break agent loops.** One open-source harness found two well-reviewed 9B models failing its easiest agent tier (QuantaMind). Errors also compound: a 95% per-call success rate over eight steps finishes about 66% of the time (PromptQuorum). Use 8 GB models for chat-style help, not autonomy.

What changes at 16 GB

The 16 GB picks are a step down in reliability, not a different kind of tool. gpt-oss-20b is a 14 GB download under Apache 2.0, and OpenAI reports 60.7 on SWE-bench Verified at high reasoning effort (Hashnode). On a 16 GB Mac it leaves only about 2 GB for the system and context, so close other apps or switch to Gemma 4 12B.

Gemma 4 12B is an 8 GB download with native function calling and far more room for context. Kilo's comparison chose it for this tier because its tool-call formatting is reliable, while noting it may trail same-size Qwen models on raw coding (Kilo).

Devstral Small 2, Mistral's 24B coding model, reports 68.0 on SWE-bench Verified but fits 16 GB only with aggressive quantization (ZimaSpace). On any of the three, keep tasks smaller and sessions shorter than you would on a 24 GB card.

Choosing the agent

Agent	Where it runs	Pick it when	License
OpenCode	Terminal and desktop	You want one agent that works with any local or hosted model	MIT
Claude Code	Terminal	You already know its workflow and want to keep it while swapping the model	Proprietary, free to download
Cline	VS Code	You want to approve each file edit and command	Open source
Aider	Terminal	You want every change as a clean git commit	Apache 2.0
OpenHands	Web UI with a sandbox	You want unattended issue-to-pull-request runs, isolated from your machine	Open source
Goose	Terminal and desktop	You want MCP extensions for tools beyond code	Apache 2.0
Kilo Code	VS Code	You run Qwen3-Coder-Next, which was trained to work inside it	Open source
Tabby	Editor plugin plus local server	You want autocomplete, not an agent	Open source

Agent details are from Morph, ZimaSpace and Kilo.

Claude Code on a local model

Claude Code's terminal interface works with open models, because Ollama serves an Anthropic-compatible API on your own machine. One command wires the two together: `ollama launch claude` (Ollama docs). The manual route

is three environment variables: `ANTHROPIC_BASE_URL=http://localhost:11434`, `ANTHROPIC_AUTH_TOKEN=ollama`, and an empty `ANTHROPIC_API_KEY`. LM Studio, Jan and llama.cpp's server offer the same kind of endpoint (Yunosuke Naito).

This pairing has an unusual point in its favor. Qwen measured its 61.7 on SWE-bench Pro with Claude Code as the harness, so Qwen3.8-27B inside Claude Code is the combination the vendor itself tested (H2S Media).

Four things to know before you depend on it:

- **It is not an Anthropic-supported setup.** Anthropic's documentation says it does not support routing Claude Code to non-Claude models through any gateway (Claude Code docs). Support comes from Ollama and the community, and a Claude Code update can break it.
- **Set the context yourself.** Ollama asks for a context window of 64K or more for this integration, and Claude Code assumes a much larger window than a local model usually has.
- **Some features are missing.** Ollama's compatibility layer covers only part of the API. Hosted web search and advanced tool controls are not fully supported.
- **Watch the model picker.** The launcher lists Ollama's cloud models alongside your local ones. Cloud models leave your machine and can be metered, so choose a local tag.

Claude Code is free to download but is not open source. If an open-source agent matters to you, OpenCode gives a similar terminal workflow under the MIT license.

A working setup in three steps

1. Install Ollama and download the model (use gpt-oss:20b on a 16 GB machine): `ollama pull qwen3.8:27b`.
2. Raise the context window to at least 64K. Ollama's default of 4K on machines under 24 GiB of VRAM will make any agent look broken.
3. Add Ollama to OpenCode as an OpenAI-compatible provider with the base URL `http://localhost:11434/v1` in `opencode.json` (Hashnode).

Alternatives worth a test on your own code: Muse Glimmer 30B from Meta, Devstral Small 2 from Mistral, and North Mini Code from Cohere, a 30B model with 3B active parameters. All three are Apache 2.0.

---

06 Everyday work: slides from your template, documents, and a general assistant

Generating a PowerPoint deck from a company template is the task where local setups most often disappoint, because the model has to respect a layout it did not design. Two routes work, and they suit different people.

Route 1: Presenton, for a point-and-click workflow

Presenton is an Apache 2.0 presentation generator that runs in Docker or as a desktop app. It connects to Ollama, LM Studio or any OpenAI-compatible endpoint, and it exports editable PPTX and PDF files.

1. Upload your existing PPTX. Presenton generates a reusable template from it.
2. Give it a prompt or a source document and choose that template.

3. Export to PPTX and check the result in PowerPoint.

Presenton also exposes an MCP server, so an agent such as Goose can request a deck as one step in a longer task. The project page does not say how closely a generated template tracks complex slide masters, so test it with your real template before you rely on it.

Route 2: a coding agent with python-pptx, for exact brand fidelity

Ask OpenCode or Goose to write a short Python script using the python-pptx library. The script opens your actual template file, picks the named layouts, and fills the placeholders with content the model drafts. Because the output is built from the template itself, masters, fonts and colors stay exactly as your brand team made them.

This route takes an hour to set up and then becomes a reusable command. It works best when you keep the two jobs apart: the model writes an outline as structured data, and the script handles placement.

Claude Desktop on a local model

The Claude Desktop app can also run on open models. Ollama added official support on August 25, 2026: open Ollama, select Claude, and switch the toggle on. Ollama then configures Claude Desktop's third-party gateway, and you choose which local model backs it (Ollama blog). Switching the toggle off restores your previous setup.

A hands-on report found chat, Cowork file operations and coding all working on local models only, with no paid Claude plan required (Yasu Shiina). That makes it the easiest route for a non-developer who wants a polished desktop assistant with no token bill. The same cautions apply as for Claude Code: quality is that of the local model, not of Claude, and the cloud models in Ollama's list are not local.

Documents and a general assistant

- **Chat and writing.** Open WebUI gives a team a browser-based chat front end over local models. Jan is the simpler choice for one person: an Apache 2.0 desktop app that works fully offline (MyClaw).
- **Questions over your own files.** AnythingLLM ships with an embedder and a vector database built in, so a laptop can index documents with no extra services, and answers come with citations (MyClaw). In the Kingy test, Qwen3.8-27B answered 23 of 24 private-document questions correctly and Gemma 4 31B answered 22.
- **Desktop automation.** Goose is an Apache 2.0 agent with a terminal and a desktop app. It gains tools through MCP extensions and now includes built-in local inference, so it can run without a separate model server (Agentic AI Foundation).

For work in languages that English-centric models handle poorly, regional models can beat the general picks. Sarvam 30B from India is Apache 2.0 and was preferred about 90% of the time on Indian-language text in its makers' evaluation (Sebastian Raschka). Japan's llm-jp-4.1 is the top Japanese-made open model on an independent Japanese benchmark, though Qwen3.8-27B still leads that board's single-GPU class (Qualiteg).

---

07 Beyond text: documents, images, video, voice and search

The best models for these jobs are small specialists, and several need no GPU at all.

- **OCR and document parsing.** Dedicated document models now beat large general vision models. PaddleOCR-VL-1.6 leads the OmniDocBench v1.6 benchmark at 96.34%, ahead of MinerU2.5-Pro at 95.75% and GLM-OCR

at 95.22%, all at roughly 1B parameters (Roboflow). On a PC with no graphics card, Tesseract with OCRmyPDF remains the practical choice for scanned PDFs (Pasquale Pillitteri).

- **Image generation.** Z-Image Turbo, a 6B model, fits within 16 GB of VRAM and renders an image in seconds. FLUX.2 [klein] 4B is the smallest strong option. Both are Apache 2.0 and run in ComfyUI (VDF). Qwen-Image is the pick when the picture must contain readable text. FLUX.2 [dev] sets the quality bar but is a 32B model under a non-commercial license (Thunder Compute).
- **Video generation.** Wan 2.2 is the strongest fully permissive option, under Apache 2.0. Later Wan versions were announced without open weights (CrePal). LTX-2.5 is the only open model that generates synchronized audio and video in one pass (LTX). Expect minutes per clip on a 24 GB card.
- **Speech to text.** Whisper large-v3-turbo covers 99 languages under the MIT license, and its quantized build is a 574 MB file that runs on a CPU through whisper.cpp. Qwen3-ASR-1.7B tops the Open ASR Leaderboard at 5.40% average word error rate, against 7.64% for Whisper turbo, but wants a GPU server (Pinggy).
- **Text to speech.** Kokoro-82M runs faster than real time on a laptop CPU under Apache 2.0, with a fixed set of voices. Chatterbox, under MIT, adds voice cloning from a few seconds of audio and needs about 4 to 6 GB of VRAM (Local AI Master).
- **Embeddings for search.** Qwen3-Embedding leads the open field on the multilingual MTEB benchmark, and its 0.6B size is a 639 MB download that accepts 32K-token inputs. BGE-M3 is the long-standing multilingual alternative under MIT (Local AI Master, Morph).

08 The runtime: what actually loads the model

Every setup above has three layers: the model file, a runtime that loads it and exposes a local API, and the agent that calls that API. Pick the runtime by how you work, since most of them share the same engine underneath.

Runtime	What it is	Pick it when	License
Ollama	Command-line tool and background service; one command downloads and serves a model	You want the shortest path, and launchers for agents such as Claude Code	MIT
LM Studio	Desktop app with a model browser; checks whether a model fits your memory before downloading	You prefer a graphical app, or want the fastest path on Apple Silicon	Proprietary, free to use
llama.cpp	The engine underneath Ollama and LM Studio	You want full control and the newest optimizations	MIT
MLX	Apple's machine-learning framework for Apple Silicon	You are on a Mac and want maximum speed	Open source
vLLM	A serving system built for many concurrent users on GPUs	You are hosting a shared endpoint for a team	Open source
Jan	Offline desktop chat app with a local server	You want a simple private assistant with no command line	Apache 2.0

Runtime details are from DevToolLab, Inventive HQ and Digital Applied.

Ollama, LM Studio, llama.cpp's server and vLLM all expose OpenAI-style endpoints, so most agents connect by changing one base URL. For a single user the runtimes perform similarly, because they share an engine. The gap

opens under load: in one test vLLM delivered 793 tokens per second on a single A100 where Ollama delivered 41, which is why vLLM is the choice for teams (Digital Applied).

09 Starter stacks by machine

If you want a shopping list instead of a comparison, start here and adjust after a week of real use.

Machine	Runtime	Models to download	Agents and apps
No GPU, 8 to 16 GB RAM	Jan	Gemma 4 E4B, Whisper large-v3-turbo, Kokoro-82M	Jan for chat, whisper.cpp for dictation
16 GB GPU or Mac	Ollama	gpt-oss-20b, Gemma 4 12B, Qwen3-Embedding 0.6B	OpenCode, Open WebUI, AnythingLLM, Presenton
24 GB GPU or 32 GB Mac	Ollama, or LM Studio on a Mac	Qwen3.8-27B, Gemma 4 31B, Z-Image Turbo, PaddleOCR-VL-1.6	OpenCode or Claude Code, Claude Desktop or Open WebUI, Goose, Presenton, ComfyUI
64 GB Mac or two GPUs	Ollama or llama.cpp	The 24 GB list plus Qwen3-Coder-Next or Flint	Kilo Code for long-context work
Team server	vLLM	DeepSeek-V4.1-Flash or GLM-5.3-Flash	OpenCode and OpenHands for every developer

A 24 GB card holds one large model at a time. The runtime swaps models when you change tasks, which takes time, so plan on one resident model for interactive work.

10 Caveats: where local still falls short, and how to stay safe

Local models are good enough for a large share of daily work, but they are not a like-for-like replacement for the best hosted models. On the Artificial Analysis Intelligence Index, the best open-weight model scores 46 against 60 for the top proprietary one, and Qwen3.8-27B, the strongest model that fits a single consumer GPU, scores 20 (Refusal Rate). A sensible pattern is local by default, with a hosted model kept for the hardest problems.

Other limits to plan around:

- **Speed.** A 27B model on a 24 GB card is responsive for one person. Long agent runs still take noticeably longer than a hosted service.
- **Benchmarks.** Most scores in circulation are vendor-reported under each vendor's own harness. Run your own five or ten representative tasks before committing.
- **License drift.** Terms change between releases in the same family. Mistral Small 4 is Apache 2.0, while Mistral Large 4 ships under a license that requires a separate agreement for some commercial uses (GeoToolbox). Read the license file at download time.
- **Shelf life.** The single-GPU pick changed from Qwen3.6-27B in April to Qwen3.8-27B in August. Treat any table like this one as a snapshot.

Supply-chain safety for downloaded models

A model file is third-party software running on your machine, and an agent built on it can run commands. Apply the same discipline you would to any dependency.

- Download weights from the publisher's own repository, not from a re-upload with a similar name.
- Prefer the safetensors and GGUF formats. Older pickle-based checkpoint files can execute code when loaded.
- Scan model files before first use and verify published checksums.
- Keep the runtime's API bound to localhost. Security researchers have documented risks from Ollama servers left exposed to the public internet (Wikipedia).
- Give agents the least access that gets the job done: approval prompts in Cline or Claude Code, a sandbox in OpenHands, and a review of every MCP extension you install.
- Check that the model you selected is a local download and not a cloud-hosted tag.

---

11 Sources

MODEL RANKINGS, SIZES AND LICENSES

- Morph: The Best Open Source LLMs, November 2026
- DevToolLab: Best Offline LLM Models and Tools 2026
- Hashnode: The best local LLMs for coding in 2026
- Kilo: The Best Local Coding Models for Any Setup
- GeoToolbox: The 11 Best Open-Source LLMs in November 2026
- Refusal Rate: Which open-source LLM leaderboards can you trust
- Qwen3.8 on GitHub and LM Studio's Qwen3.8 page
- Sebastian Raschka: open-weight releases of early 2026
- Qualiteg: Japanese LLM Rankings 2026

INDEPENDENT TESTS

- Kingy: Qwen3.8-27B vs Qwen3.6-27B vs Gemma 4 31B on a 24 GB GPU
- H2S Media: Best Local Coding Models in 2026
- QuantaMind agent benchmark notes
- PromptQuorum: Best Local Models for Tool Calling and Best CPU-Only Local LLM
- InsiderLLM: CPU-Only LLMs

AGENTS AND APPS

- Ollama docs: Claude Code integration and Ollama blog: Claude Desktop support
- Claude Code docs: LLM gateways
- Yunosuke Naito: Running Claude Code with Local LLMs and Yasu Shiina: Claude Desktop with a local model
- Presenton on GitHub
- Morph: Open Source AI Coding Assistants and ZimaSpace: Top 10 Open-Source AI Coding Assistants
- Agentic AI Foundation: hands-on with Goose and local models
- MyClaw: AnythingLLM alternatives

DOCUMENTS, MEDIA AND VOICE

- Roboflow: Best Open-Source OCR Models and Pasquale Pillitteri: best local OCR in 2026
- VDF: Best Local Image Generation Models and Thunder Compute: Best Open-Source Image Generation Models
- LTX: Open Source Video Generation Models, InsiderLLM: Best Local AI Video and CrePal: LTX 2.3 vs Wan 2.2
- Pinggy: Best Open Source Speech-to-Text Models and Local AI Master: Best Local TTS Models
- Local AI Master: Best Ollama Embedding Models and Morph: Ollama Embedding Models

RUNTIMES

- Inventive HQ: Ollama vs LM Studio vs llama.cpp
- Digital Applied: Run Local LLMs in 2026

AIAppSecAcademy

AI AppSec Academy is a free knowledge resource covering AI for AppSec, securing AI systems, and securely using vendor AI services. No paywalls, no accounts.

aiappsecacademy.com

© 2026 Dr. Viswanath Srinivasan Chirravuri. All rights reserved. This guide reflects publicly available information as of November 14, 2026 and is provided for education; it is not a guarantee of any product's behavior, licensing or fitness for a purpose. Product and model names are the property of their respective owners. Claude Code and Claude Desktop are products of Anthropic; running them on non-Claude models is not supported by Anthropic.