INNOVATION AI LAB

INO AI LAB

[AI EDUCATION • COURSE MATERIAL]

Running LLMs Locally with Ollama

Llama 4, Mistral, private AI on your hardware

LEVEL	DURATION	LESSONS
Intermediate	**4 hours**	**8**

Innovation AI Lab — Tools, Prompts & Performance

// 2026 EDITION

[00]

Table of Contents

[LESSON 01]

Why Run Models Locally in 2026

Three reasons to run local: privacy (sensitive data never leaves your machine), cost (zero per-token fee after hardware), and latency (no network round trip). The 2026 sweet spot is a Mac Studio with M4 Ultra or a PC with an RTX 5090 — both run 70B-class models at usable speeds. Below that, stick to 8B–14B models which now match GPT-3.5 quality for most tasks.

Don't replace frontier APIs entirely. Use local for sensitive workloads, batch processing, and high-volume simple tasks; use APIs for hard reasoning and cutting-edge quality. The right architecture in 2026 is hybrid: a router that sends each request to local or API based on sensitivity, complexity, and budget. This gets you the best of both worlds at 30–60% lower cost than pure API.

// KEY TAKEAWAYS

- Local = privacy + cost + latency.
- M4 Ultra or RTX 5090 for 70B models.
- Hybrid local+API beats pure API on cost.

[LESSON 02]

Ollama: The Standard Local Runtime

Ollama is the easiest path: one-line install, a simple `ollama run llama4` to start, and an OpenAI-compatible API on localhost:11434. Models download as quantized GGUFs and run on CPU or GPU automatically. The library includes most major open models — Llama 4, Mistral, Qwen 3, Gemma 3, DeepSeek V3, Phi-4 — updated within days of release.

Pick quantization carefully. Q4_K_M is the modern default: noticeably smaller and faster than full precision with minimal quality loss. Q8_0 is closer to original quality at 2x size — use when you can afford the RAM. Avoid Q2/Q3 except for tiny models on weak hardware; the quality hit is severe. Test your real workload at each quant level before committing to one.

// KEY TAKEAWAYS

- Ollama = one-line install, OpenAI-compatible API.
- Q4_K_M is the modern default quant.
- Test quant levels on your real workload.

[LESSON 03]

LM Studio, llama.cpp, and the Power-User Stack

For finer control, LM Studio gives you a GUI to download, compare, and serve models with tweakable parameters. llama.cpp underneath powers everything and is what to use directly for max performance and embedded deployments (it runs on Raspberry Pi). For batch inference and serving multiple users, vLLM is the production-grade option — dramatically higher throughput than Ollama for concurrent requests.

Match the runtime to the use case. Personal chat → Ollama or LM Studio. Embedded device → llama.cpp directly. Multi-user internal tool → vLLM behind FastAPI. Production app with autoscaling → vLLM on a GPU pod (RunPod, Modal, Lambda). Each step up adds operational complexity in exchange for throughput; don't over-engineer for a five-user internal tool.

// KEY TAKEAWAYS

- Ollama/LM Studio for personal use.
- vLLM for multi-user serving.
- Match runtime complexity to actual concurrency.

[LESSON 04]

Fine-Tuning, LoRA, and Custom Models

You usually don't need fine-tuning — prompting plus RAG covers 90% of customization needs. When you do need it (consistent format, proprietary jargon, specific tone), use LoRA: train a small adapter on 100–1,000 examples in 1–3 hours on a single GPU. Tools like Unsloth, Axolotl, and MLX make this approachable for individual developers in 2026.

Evaluate honestly. Fine-tunes often overfit to training data and regress on tasks outside the training distribution. Always test the fine-tuned model on a held-out general benchmark to confirm you haven't broken the base capabilities. The pattern of 'fine-tune → great on training data → unusable in production' is depressingly common; rigorous eval is what prevents it.

// KEY TAKEAWAYS

- Prompting + RAG covers 90% of customization.
- LoRA on 100–1,000 examples = 1–3 GPU hours.
- Test fine-tunes on held-out general benchmarks.

[NEXT]

Keep Going

You've completed this course material. The real learning starts when you apply what you've read. Pick one idea from this PDF and run a small experiment this week. Document what worked and what didn't. Share your findings with the community.

Explore more free courses, daily AI tips, and curated tools at:

innovationailab.com

Have feedback or want to suggest a topic? We read every message.

hello@innovationailab.com

— Innovation AI Lab Team —

// part of LumiLife Tech