What It Is

- llama.cpp is an open source C/C++ project for LLM inference with minimal setup and high performance across CPUs and GPUs.
- It provides a core llama library plus runnable tools such as a CLI and an OpenAI-compatible HTTP server.

Who It’s For

- Primary persona: developers and ML practitioners who want to run GGUF-based models locally or in their own infrastructure.

What It Does

- Runs LLM inference from a plain C/C++ implementation with broad hardware support.
- Supports quantization formats (1.5-bit through 8-bit) to reduce memory and improve speed.
- Offers multiple acceleration backends (e.g., Metal, CUDA, Vulkan, SYCL, and CPU paths).
- Ships user-facing executables including ‘llama-cli‘ and ‘llama-server‘.
- Exposes an OpenAI-compatible REST API through ‘llama-server‘.
- Supports both single-model inference mode and multi-model router mode in server docs.

How It Works (Repo-Evidence Architecture)

- Core: ‘libllama‘ C-style interface in ‘include/llama.h‘; tools are built on top of this library.
- Server path: HTTP requests enter ‘server_http_context‘, route through ‘server_routes‘, become ‘server_task‘s in ‘server_queue‘, run in ‘server_context‘/‘server_slot‘, then return via ‘server_response‘.
- Execution model: ‘server_context‘ runs on a dedicated thread; HTTP workers handle request parsing and JSON formatting.
- Data flow: prompt + model settings -> tokenization/scheduling/batching -> model decode/sampling -> streamed or final response.

How To Run (Minimal)

- 1. Build: ‘cmake -B build‘ then ‘cmake --build build --config Release‘.
- 2. Get a GGUF model locally or use Hugging Face shortcut flags shown in README.
- 3. CLI example: ‘llama-cli -m my_model.gguf‘.
- 4. Server example: ‘llama-server -hf ggml-org/gemma-3-1b-it-GGUF‘.

Not Found In Repo

- Formal product SLA / commercial support policy: Not found in repo.