

Mohammed Al-Thobaiti

+966 54 006 2087 | mthobaiti@outlook.com | linkedin.com/in/mthobaiti | github.com/voidwest

SUMMARY

Computer Science graduate focused on systems programming, developer tooling, and ML infrastructure in Rust and Python. Experience building CPU inference engines, benchmarking infrastructure, and local retrieval systems with an emphasis on performance, correctness, and reproducible evaluation. Interested in systems, backend, and ML infrastructure roles.

EDUCATION

Umm Al-Qura University
Bachelor's in Computer Science

Makkah, Saudi Arabia
Sept. 2019 – Feb. 2024

PROJECTS

- Ember (CPU GGUF Inference & Probing Engine)** | *Rust, Python, GGUF* Mar. 2026 – Present
- Built a CPU-first GGUF inference engine in Rust with quantized execution, KV caching, deterministic tracing, and hidden-state extraction for interpretability and systems research.
 - Implemented packed quantized projections, VNNI kernels, interleaved weight layouts, and fused operations, reaching ~69% of llama.cpp multicore decode throughput while adding only 62–72 MiB RSS.
 - Validated hidden-state extraction against llama.cpp across 4,701 Arabic morphology probes with exact token alignment, reproducing the study's main probing conclusions through Ember's independent inference path.
 - Hardened GGUF loading as an untrusted-input boundary using typed validation, resource limits, and fuzz testing across parsing, model construction, and tokenizer paths; converted discovered failures into regression tests.
- InferTrace (Local Inference Benchmarking Platform)** | *Rust, Axum, SQLite, React* July 2026 – Present
- Built a local benchmarking and regression platform used to improve Ember from 16.5× slower to approximately 2× slower than llama.cpp while substantially reducing memory usage.
 - Added process supervision, timeout and cancellation handling, statistical summaries, and comparability checks for repeatable inference benchmarks.
 - Persisted complete run provenance in SQLite and implemented checksum-verified export/import bundles for reproducible experiments.
- DevDocs (Local RAG System)** | *Python, FastAPI, Docker, ChromaDB, Ollama* July 2025 – Nov. 2025
- Built a fully local document retrieval system with incremental indexing, context-aware chunking, and page-level source attribution.
 - Improved embedding throughput by 3× through local model execution and packaged the system for offline deployment with Docker.
- miniOS** | *Rust, x86_64, QEMU* Jan. 2024 – Aug. 2024
- Implemented a minimal `no_std` x86_64 kernel in Rust with paging, heap allocation, interrupt handling, and a basic shell, tested under QEMU.
 - Built low-level memory and interrupt infrastructure without the standard library, including page-table management, dynamic allocation, and interrupt descriptor handling.

SKILLS

Languages: Rust, Python, SQL, TypeScript, JavaScript, Go, Java
Systems & Backend: Linux, Tokio, Axum, FastAPI, REST APIs, SQLite, SQLx, Bash, Process Management
ML Infrastructure: GGUF, Quantized Inference, LLM Benchmarking, llama.cpp, Ollama, RAG
Tools: Git, Docker, React, Vite, ChromaDB, QEMU
Spoken Languages: Arabic (Native), English (IELTS 8.0)