Muhammad Qasim

AI Systems Engineer · Autonomous Agents · Retrieval Infrastructure · Applied ML

qasimio.github.io | linkedin.com/in/qasimio | github.com/qasimio | amkassim444@gmail.com

Professional Summary

AI Systems Engineer with hands-on experience architecting autonomous agent loops, retrieval-augmented generation pipelines, and developer tooling. Built **Operon** — a terminal-native autonomous coding agent with a persistent cross-file symbol graph, AST-based refactoring engine, and a deterministic verification layer that eliminates the most common failure modes of local LLMs. Strong foundation in classical information retrieval applied directly to practical RAG systems. Designs software where safety and correctness are first-class engineering constraints.

Technical Skills

Languages: Python (4 yrs), Java (2 yrs), C++ (1.5 yrs), SQL, Bash

Agentic AI: ReAct Architecture, Tool Calling, Multi-step Planning, Self-correction, Human-in-the-Loop Oversight

LLM & ML: LlamaIndex, LangChain, HuggingFace Transformers, PyTorch, BERT Fine-tuning, Cross-Encoders

Retrieval: RAG Pipelines, Hybrid Search (Sparse+Dense), LanceDB, ChromaDB, FastEmbed, Inverted Indices, Reranking

Parsing / AST: Python `ast`, `tokenize`, Tesseract OCR, Poppler, Document Intelligence

Systems: Linux, Docker, GitHub Actions, Multithreading, File Locking, Memory Management, PyPI Packaging

Tooling: Textual (TUI), argparse, Click, `setuptools`, `wheel`, Git

Engineering Projects

Operon — Autonomous Code Intelligence Agent

Python · Textual · LLM APIs · AST · ReAct

https://github.com/qasimio/Operon

- Architected a terminal-native autonomous agent that builds a persistent, hash-gated symbol graph across an entire repository — enabling cross-file refactoring, documentation generation, and execution flow analysis without loading full files into context.
- Engineered a **deterministic-first REVIEWER** that verifies file changes by comparing disk content against diff memory snapshots before any LLM call, eliminating hallucinated change confirmation — the root cause of most agent loop failures.
- Implemented a **CRUD fast-path** using Python `ast` and `tokenize` for structured operations (import insertion, variable renaming, comment placement), removing LLM dependency for predictable edits and cutting noop failure rate to zero for these cases.
- Designed a **5-tier surgical diff engine** with cascading matching from exact string to whitespace-normalized to fuzzy multi-line, enabling reliable SEARCH/REPLACE patching against files modified by both LLM and deterministic paths.
- Built a universal **LLM router supporting 9 providers** (local, OpenAI, Anthropic, OpenRouter, Deepseek, Groq, Together, Azure, custom) with hot-reload config — model switching takes effect on the next call, no process restart required.
- Implemented mandatory **approval gate architecture**: no filesystem write occurs without explicit human confirmation; 300s timeout auto-rejects to prevent thread hang in the multithreaded Textual TUI.

MQNotebook — Enterprise RAG System

Python · LlamaIndex · Tesseract · Poppler · Streamlit

https://github.com/qasimio/MQNotebook

- Engineered a local-first RAG pipeline targeting enterprise document formats excluded by most RAG implementations: scanned PDFs (image-only pages), PPTX speaker notes, and multi-sheet XLSX — using a custom Tesseract + Poppler OCR ingestion layer.
- Implemented **hybrid retrieval** combining dense vector search with a Cross-Encoder reranker, achieving **~40% precision improvement** over naive cosine similarity in context selection accuracy.
- Solved persistent WinError 32 file-locking failures in ChromaDB on Windows by designing a **session-isolated storage handler** that dynamically routes each session to a separate persistent store, eliminating cross-session lock contention.
- Optimized context injection strategy, **reducing token consumption by ~60%** without degrading retrieval quality — enabling deployment within Streamlit Cloud resource constraints while maintaining full document coverage.

DevShelf — Vertical Search Engine (from First Principles)

Java · Information Retrieval · Data Structures

https://github.com/qasimio/DevShelf

- Architected a full vertical search engine for CS literature **without Lucene or ElasticSearch** — implementing a custom Positional Inverted Index with **O(1)** keyword retrieval via direct-address hashing.
- Engineered an Offline Indexer that decouples corpus processing from query time, achieving **sub-millisecond** online query latency. Implemented O(L) Trie for autocomplete; Levenshtein Distance for fuzzy correction.

- Classical IR architecture from DevShelf directly informed the hybrid search and reranking design in MQNotebook — demonstrating intentional, theory-grounded engineering progression.

foldr — File Automation CLI pip install foldr *Python · PyPI · CLI Design*

https://github.com/qasimio/foldr

- Designed and published a production-ready CLI tool to PyPI for automated file organization by extension — targeting data pipeline preparation and ML dataset cleaning workflows.
- Implemented **dry-run architecture** that previews all planned I/O operations before execution, with automatic conflict resolution; directories are explicitly protected from modification.
- Mastered Python packaging standards (`setuptools`, `wheel`, `entry_points`) and CI/CD release pipeline to PyPI.

Experience

Machine Learning Intern · Arch Technologies (Remote) Present

- Fine-tuned BERT models for NLP text classification tasks; profiled and optimized preprocessing pipelines, materially improving batch throughput for downstream training runs.
- Integrated trained PyTorch models into internal production prototypes in collaboration with senior engineering teams, including model serialization, versioning, and serving infrastructure.

Open Source & Published Tools

- **foldr** · PyPI — Published CLI tool available via `pip install foldr` pypi.org/project/foldr
- **Operon** · GitHub — Open-source autonomous code intelligence agent github.com/qasimio/Operon

Additional Technical Work

BabyGPT — Character-level LSTM language model from scratch (TensorFlow); demonstrates sequence modeling understanding prior to transformer usage. **MQ Banking Core** — Transactional banking system in C++ with file-level I/O and balance integrity guarantees. **Digital Eye** — CNN-based image classification pipeline (Keras/TensorFlow).

Education

B.Sc. Computer Science — Distributed Computing & AI Systems · Sukkur IBA University, Pakistan Expected 2028