

MUHAMMAD QASIM

AI Systems Engineer — Retrieval Architecture & Python Infrastructure

 [qasimio.github.io](https://github.com/qasimio)  [linkedin.com/in/qasimio](https://www.linkedin.com/in/qasimio)  github.com/qasimio  amkassim444@gmail.com

TECHNICAL SUMMARY

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

AI & MLOps: LlamaIndex, LangChain, HuggingFace, PyTorch, PyPI Packaging, CLI Design

Retrieval Systems: RAG Pipelines, Hybrid Search (Sparse/Dense), Cross-Encoders, OCR Ingestion

System Engineering: Linux, Docker, CI/CD (GitHub Actions), File Locking, Memory Management

Algorithms: Inverted Indices, Tries, Vector Quantization, Graph Traversal, O(1) Lookups

SELECTED PROJECTS

MQNotebook — Enterprise-Grade RAG System

Python, LlamaIndex, OCR

Engineered a local RAG engine capable of parsing complex formats (scanned PDFs, PPTX speaker notes, XLSX) using a custom Tesseract + Poppler OCR pipeline.

Implemented a **Hybrid Search** architecture combining dense vector retrieval with a **Cross-Encoder Reranker**, improving context precision by 40% compared to naive cosine similarity.

Solved critical Windows file-locking (WinError 32) issues in persistent vector stores by architecting a dynamic, session-isolated storage handler.

Optimized context injection to reduce token usage by **60%**, deploying the solution to Streamlit Cloud.

DevShelf — Distributed Vertical Search Engine

Java, Systems Architecture

Architected a search engine from first principles (no Lucene), implementing a custom **Positional Inverted Index** and Vector Space Model for O(1) keyword retrieval.

Engineered a specialized "Offline Indexer" to pre-process corpora, reducing runtime query latency to sub-millisecond levels.

Implemented low-level data structures including O(L) Tries for autocomplete and Levenshtein Distance for fuzzy matching.

foldr — Automated Data Management CLI

Python, PyPI, DevOps

Designed and published a production-ready file automation tool to **PyPI** (`pip install foldr`) for cleaning and organizing large-scale datasets.

Engineered a robust heuristic engine that sanitizes file names and restructures directories, essential for preparing raw data for ML training pipelines.

Implemented a safe-guard "**Dry Run**" architecture to preview IO operations before execution, preventing data loss in automated workflows.

Mastered Python packaging standards (setuptools, wheel) and CLI argument parsing to deliver a developer-friendly experience.

EXPERIENCE

Arch Technologies (Remote)

Machine Learning Intern

Present

Fine-tuned BERT models for NLP classification tasks; optimized preprocessing pipelines for improved training throughput.

Collaborated with engineering teams to integrate PyTorch models into internal production prototypes.

EDUCATION

Sukkur IBA University

Bachelor of Computer Science — Distributed Computing & AI Systems

Expected 2028