

XAVIER FRANKLINE ODHIAMBO

xavierfrank4@gmail.com | xodhiambo.com | github.com/theypsalmist | +254786409255 | Remote
PROJECTS

Internal Ad-Quality and Domain-Trust Classifier

Code for Africa | Python, LlamaIndex, pgvector

Led development of an internal ad-quality and domain-trust classifier: a gated multi-stage pipeline that screened advertisement URLs through cheap, fast signals first (domain popularity, infrastructure trust checks, link-graph authority), then sent only the URLs that cleared those checks through a more expensive LLM-based content-quality stage. That stage embedded the scraped page content and queried an LLM through templated prompts to detect trust-indicator markers. The cheap-signals-first gating was a deliberate architectural decision: it reserved LLM cost and latency for URLs already worth investigating rather than spending it uniformly across every input.

Elasticsearch Cluster Architecture and Acceptance Benchmarking

Code for Africa | Elasticsearch, Ansible, Elastic Rally

Led the architecture and deployment of multi-node Elasticsearch clusters using Ansible automation, supporting the organization's news indexing and search infrastructure. To validate that a freshly provisioned cluster was actually ready for production traffic, built a Rally-based acceptance benchmark harness that measured indexing throughput, query latency under increasing load, and mixed read/write performance, turning cluster acceptance into a repeatable, measurable process. Implemented observability for the clusters with Grafana, Prometheus, and Alertmanager, and used that instrumentation to investigate and resolve production incidents by correlating exceptions, logs, and metrics.

Self-hosted LLM Platform

Code for Africa | Ollama, OpenWebUI

Led the architecture, deployment, and operation of a self-hosted LLM platform giving non-engineering teams, including editorial, research, and content staff, safe, private access to AI tools for sensitive workloads, complementing the organization's commercial LLM providers. This included access controls, model-selection guidance for non-technical users, and ongoing adoption support, with real, meaningful usage across non-engineering teams.

Mediacloud Story-Indexer

Open Source Contribution | Python, FastAPI, RabbitMQ, Elasticsearch

Contributed to Mediacloud Story-Indexer, an open-source distributed news ingestion and indexing pipeline operated by Mediacloud, a nonprofit research collaboration tracking thousands of news sources globally and indexing millions of stories for researchers. Worked on FastAPI services for orchestration and Elasticsearch query layers for downstream tooling, within a queue-driven worker architecture (fetch, parse, index) using explicit retry, backoff, and dead-letter handling for reliable message delivery at scale. Was part of the team that migrated the crawler off Scrapy to a requests-and-RabbitMQ-based pipeline better suited to the project's distributed queue-and-worker model, a decision made by the project's lead.

LLM Orchestration Layer

Search Atlas | Python, OpenRouter

Improved and extended an existing LLM orchestration layer that multiple internal services depended on for automated analysis and content generation, adding multi-provider routing, automatic failover, and cost-aware model selection, along with structured logging the team could use to reason about model behaviour over time.

Wazimap-NG

Code for Africa | Django, Django REST Framework, GeoDjango

Built and maintained a census and demographic data visualization platform for policy teams and researchers, with nested administrative geographies and demographic indicators served through GeoDjango-backed REST APIs.

Sensors.Africa

Code for Africa | Django, Django REST Framework, Celery

Built backend services for an IoT air-quality network, ingesting and validating sensor readings across multiple African cities.

Algorithmic Trading System

Personal Project | Python, MetaTrader 5 API, Telegram

Built and continue to run an automated trading system for gold (XAU/USD), connecting to a trading platform's Python API to place and manage trades based on rules tested first against historical data through a custom backtester also built for this project. Runs several parallel bot configurations with different settings and sends trade updates through a Telegram integration.