Ian Muchesia

Software Engineer

ianmuchesia78@gmail.com +254797159837 Nairobi https://github.com/ianMuchesia

PROFILE

I am a software engineer. I build backend systems in C# / .NET and Python, and AI applications that use large language models. On the backend I work with APIs, Kafka, RabbitMQ, Azure and Docker. On the AI side I build agents, search (RAG) and classifiers, and I test them carefully before I trust the results. I like to measure things, write down what I learn, and say clearly when something did not work.

SKILLS

Languages
C#, Python, TypeScript / JavaScript, SQL

Machine Learning
PyTorch, scikit-learn, LoRA fine-tuning, testing models on held-out data, measuring speed (latency)

AI
LLM agents (Microsoft Agent Framework), MCP tools, prompt engineering, RAG, embeddings, vector search (FAISS, Typesense), OpenAI GPT on Azure, Gemini, Ollama

Backend
.NET, ASP.NET Core, Entity Framework Core, FastAPI, Node.js, REST APIs, gRPC, WebSockets

Messaging
Kafka, RabbitMQ, retries with Polly, dead-letter queues

Data
PostgreSQL, SQL Server, MySQL, MongoDB, Redis, DuckDB, Parquet, pandas

Cloud and Tools
Azure, AWS, Docker, GitHub Actions, Bicep, Linux, Nginx, Git

Frontend
React, Next.js, React Native

PROJECTS

Football Intelligence (Second Ball) Live
Python, Azure OpenAI, Microsoft Agent Framework, MCP, Kafka, PostgreSQL, React

- AI agents watch a football match and explain what is happening, in English and Kiswahili. Match events flow through Kafka, and the numbers are always worked out by code, not by the AI.
- A checker reviews every line the AI writes before it is shown. In my tests it caught all 14 real mistakes in 34 drafts, with no false alarms. Risky lines wait for a person to approve them.
- Each AI call has a spending limit (a full demo run costs about USD 0.03). It runs on Azure and has about 200 automated tests.

Nia — Banking Assistant in English, Kiswahili and Sheng 2026
Python, PyTorch, EmbeddingGemma, scikit-learn, FastAPI, PostgreSQL, React

- Works out what a customer wants from their message, then answers, asks them to confirm, declines, or passes the message to an LLM when it is unsure. Money only moves after the customer confirms.
- Tested on 298 messages it had never seen: 86.6% right on the first guess, and no serious mistakes on money or fraud messages.
- Measured how fast it is (about 104 ms per message, 135 ms for the slowest 1%) and tried fine-tuning the model. A full fine-tune scored higher but made 2 serious mistakes, so I kept the safer version.

Large Data Processing 2026

Python, DuckDB, Parquet, pandas, Docker

- pandas ran out of memory on a 30.5 GB file, so I switched to DuckDB. It processed about 440 million rows in 10.5 minutes using about 1.5 GB of memory, and the row counts matched an independent check exactly.

Event-Driven .NET Services 2025

C#, .NET, Kafka, gRPC, Avro, Polly, Redis, Nginx, Docker

- Order processing: order, product, inventory and payment services that talk through gRPC and Kafka, with event formats stored in a schema registry.
- Notification service: retries failed messages, then moves them to a dead-letter queue. Distributed WebSockets: real-time chat spread across several servers behind a load balancer.

Product Search Comparison 2026

Python, FastAPI, Typesense, EmbeddingGemma, React

- Compared 11 ways of searching a Kenyan electronics shop with 71 test questions. Combining keyword and AI search found the right product first 92% of the time, against 59% for keywords alone.

Kenya Economic Data Collector Live

Python, Azure, GitHub Actions, Docker

- Two scheduled jobs on Azure save data from 19 Kenyan sources (CBK, KNBS, Kenya Law) every day, and send an alert if any source fails.

Also on GitHub: LoRA from the paper (fine-tuning GPT-2) · mini-llm-api (a small GPT-style model trained on Swahili Wikipedia) · FarmCast (weather advice for farmers using Gemini) · Qwen agent playground (five AI agents on a local model)

EDUCATION

Bachelor of Science (Mathematics)

Maseno University