Tavonga Obey Chitambira

Full Stack Engineer · ML / MLOps

obichitas03@gmail.com

+91 70577 87903

Pune, Maharashtra, India

tavongachitambira.vercel.app

github.com/tavongachitambira

linkedin.com/in/tavonga-chitambira

Available upon graduation (June 2028)

Zimbabwean

Summary

Full-stack engineer and Computer Science student at Vishwakarma University, Pune, moving into ML and MLOps. I build real-time distributed systems and am currently architecting SyncBoard, a horizontally-scaled multiplayer incident management platform, with ML-driven incident classification and runbook retrieval planned next. Core stack: Next.js, TypeScript, Apollo GraphQL, Prisma and PostgreSQL, with a growing focus on GenAI and LLM engineering.

Skills

Technical Proficiency

Languages: TypeScript, JavaScript, Python, SQL

ML & Data: NumPy, scikit-learn, XGBoost, ARIMA forecasting

Frontend: Next.js App Router, Tailwind CSS, shadcn/ui, Zustand, Apollo Client

Backend: Node.js, Express, Apollo GraphQL, Prisma, Socket.io, BullMQ

Databases & Infrastructure: PostgreSQL, Redis, Docker, Kubernetes (HPA)

Tools & Platforms: Cloudinary, Stripe, Git, Vercel

Languages

English — Fluent

Shona — Native

Education

BSc (Hons) in Computer Science,
Vishwakarma University, Pune
08/2024 – Present | India

Relevant coursework: Discrete Mathematics, Data Structures & Algorithms, Object-Oriented Programming and Database Systems

Projects

SyncBoard

07/2026 – Present

Real-Time Incident Management Platform · Next.js · Socket.io · Redis · PostgreSQL · Yjs · Zustand · BullMQ · Stripe · Kubernetes

- Architected a horizontally-scaled WebSocket cluster using @socket.io/redis-adapter for Redis Pub/Sub fan-out across pods, with reconnect delta sync to recover missed operations
- Implemented Yjs CRDT for conflict-free collaborative text editing and optimistic locking with a version column for structured field conflict resolution (409 rollback on stale writes)
- Built a Redis Sorted Set presence system with 30s heartbeat, 45s TTL window, and background cleanup, surfacing live collaborators per incident room
- Designed a debounced PostgreSQL flush worker (500ms interval) to protect the database from raw socket event volume
- Wired Stripe seat licensing and per-incident usage metering via BullMQ async jobs, fully decoupled from the socket hot path
- Implemented a 15-minute operation replay window and per-connection Redis sliding-window rate limiting (100 events/60s)

TCS Data Science Industry Project

Retail sales performance analysis · AdventureWorks dataset

- Forecast sales with ARIMA and segmented customers with K-Means clustering
- Built Random Forest and XGBoost prediction models and estimated customer lifetime value (CLV)

Self-Study

GenAI & LLM Engineering

07/2026 – Present

Self-directed, 8-phase structured roadmap. Currently: Python and ML math foundations (NumPy, linear algebra; Deisenroth, *Mathematics for Machine Learning*), building an ML repository alongside the reading.