

Hassan Tsuma

Backend & Full-Stack Engineer · Java, Spring Boot, PostgreSQL, TypeScript

Homabay, Kenya · open to remote worldwide, on-site in Kenya, or relocation

+254 714 931 575 · karungwahassan03@gmail.com · hassan200503.github.io · github.com/hassan200503 ·

linkedin.com/in/hassan-tsuma

PROFILE

Backend-focused full-stack engineer and Computer Science student. I designed and built **RentManager** end to end and deployed it: a multi-tenant rental-management platform for the Kenyan market that collects rent over M-Pesa, keeps an append-only ledger of every transaction, and computes rental-income tax against Kenyan law. It runs on three surfaces — a web app, a native mobile app, and the API behind both. I work on systems where correctness matters more than speed: money, records, and anything a business cannot afford to get wrong.

SELECTED PROJECT — RENTMANAGER

Multi-tenant property management SaaS · sole designer and engineer · deployed and live

Java 21 · Spring Boot 3 · PostgreSQL · Flyway · Docker · Next.js 16 · React 19 · TypeScript · React Native (Expo) · Clerk · M-Pesa Daraja

- Built a modular monolith of **21 backend modules** on a ports-and-adapters shape, with **1,588 tests** across 261 test classes and integration tests running against real PostgreSQL via Testcontainers.
- Designed the schema and wrote **99 sequential Flyway migrations**, none edited after running. The rent ledger is append-only at the database level: a correction is a new linked reversal, never an edit.
- Integrated M-Pesa Daraja in both directions — STK push to collect rent, B2C to pay landlords out — with idempotency windows and callback handling that stays correct when callbacks arrive late, twice or out of order.
- Enforced tenant isolation from the verified JWT claim rather than a client-supplied header, at both the authorization and repository layers, so a missed filter cannot leak one landlord's data to another.
- Built a rental-income tax module against the Finance Act that fails closed: every gate stays off until a human confirms it applies.
- Delivered the web client (**62 routes, 258 components**) with types generated from the backend's OpenAPI contract, so a breaking response change fails the build rather than a screen.
- Built the companion **React Native (Expo) app, 44 screens**, sharing that same generated API client, with push notifications, offline detection and tokens held in the device keystore.
- Deployed and operate the stack on free infrastructure — API on Render, web on Netlify, PostgreSQL on Neon — with nightly encrypted backups retained 90 days, and wrote the runbook for it.

ALSO BUILT

Kenya Keys — sponsorship management system

Multi-module Java 21 / Spring Boot system matching sponsors to students, covering institutions, support requests, analytics and a financial ledger whose operations are modelled as explicit command objects. Roughly 390 source files, on the same layering I use everywhere. Not tested or deployed, and I would not ask to be judged on it.

TECHNICAL SKILLS

Languages	Java 21, TypeScript, SQL
Backend	Spring Boot 3, Spring Security, Spring Data JPA, REST, OpenAPI, JWT/JWKS
Architecture	Modular monolith, ports and adapters, domain-driven design, aggregates, domain events, transactional outbox, multi-tenancy
Data	PostgreSQL, Flyway, schema and constraint design, append-only ledgers
Frontend	React 19, Next.js App Router, TanStack Query, Tailwind CSS
Mobile	React Native, Expo, Expo Router, push notifications, secure storage
Payments	M-Pesa Daraja (STK push, B2C), idempotent transactions, AES-GCM credential encryption
Testing	JUnit 5, Mockito, Testcontainers, ArchUnit, Vitest
Delivery	Docker, Git, GitHub Actions, Render, Netlify, Neon

EDUCATION

Tom Mboya University — BSc Computer Science, Homabay, Kenya. Expected 2028.

CURRENTLY BUILDING DEPTH IN

Running a system under real production load: metrics, alerting and incident response. Working inside a large codebase I did not design. Code review in both directions. I have deliberately avoided distributed systems so far — a modular monolith with a database-backed outbox was the right call at this scale.