

# KWANELE NTSHANGASE

0789325315 · kwanelerho69@gmail.com · linkedin.com/in/kwanele-ntshangase-abab7037b · github.com/dev-k99 · Portfolio  
Junior .NET Fullstack Developer — ASP.NET Core 8 · MediatR CQRS · React · Azure · Clean Architecture | Published on  
GitHub Marketplace · 9 live deployed systems · SA regulatory domain expertise: POPIA, SAPS/ITAC

---

## EDUCATION

---

### Nelson Mandela University

— Dec 2025

Diploma in Information Technology

### Khulumezulu High School

— Nov 2021

National Senior Certificate

## SKILLS

---

**.NET Stack:** ASP.NET Core 8, EF Core 8, MediatR 12, CQRS, FluentValidation, SignalR, JWT, xUnit, Moq, Clean Architecture, BCrypt.Net

**Frontend:** React 18/19, TypeScript, JavaScript, Tailwind CSS, Vite, Zustand, TanStack Query v5

**Languages:** C#, TypeScript, JavaScript, Python, SQL

**Databases:** SQL Server, Azure SQL, PostgreSQL, SQLite, Supabase

**Cloud & DevOps:** Azure App Service, GitHub Actions, Docker, Vercel, Render, OIDC Workload Identity Federation

**AI & Tooling:** LangGraph, FastAPI, RAGAS, ChromaDB, Power Automate, Zapier, n8n, Power BI

**Soft Skills:** Critical Thinking, Problem-Solving, Effective Communication, Leadership

**Languages:** English, IsiZulu, IsiXhosa

## PROJECTS

---

### ScrapFlow SA | Full-Stack Scrapyard Management System — React, ASP.NET Core 8, SignalR, PostgreSQL

- Built and deployed a full-stack SAPS/ITAC-compliant scrapyard management platform using ASP.NET Core 8 and Clean Architecture (3-layer: API / Core / Infrastructure), exposing 35+ endpoints across 7 controllers with JWT authentication, refresh token rotation, role-based access control (Owner, Operator, Cashier, Gate, Auditor), and fixed-window rate limiting — hosted on Azure App Service and PostgreSQL v12 (Azure Flexible Server) with automated EF Core migrations on startup.
- Engineered real-time inventory tracking using ASP.NET Core SignalR WebSockets, broadcasting live stock updates to all connected React clients on ticket completion — removing the need for manual page refreshes and enabling accurate multi-site inventory visibility without polling.
- Designed a webhook automation engine using HMAC-SHA256 signed HTTP payloads integrating with Zapier and n8n to trigger supplier payment receipts and manager alerts on business events — with a React/Zustand Owner-only UI for creating, testing, and toggling webhook subscriptions without code changes.

### SupportOS | Internal IT Support Ticket API — ASP.NET Core 8, MediatR, EF Core 8, Azure SQL

- Implemented MediatR 12 CQRS with a 4-behavior typed pipeline (IdempotencyBehavior → LoggingBehavior → ValidationBehavior → PerformanceBehavior) — FluentValidation returns field-keyed errors as Result<T> failures (not exceptions); idempotency behavior short-circuits on cache hit via IMemoryCache, preventing double-submit without any endpoint boilerplate; PerformanceBehavior logs a warning when any handler exceeds 500ms.
- Built a SLA tracking domain service calculating per-priority deadlines (Critical 2h / High 8h / Medium 24h / Low 48h) from ticket CreatedAt — not DateTime.UtcNow, explicitly preventing the silent SLA extension bug in naive implementations; CanTransitionTo() enforces valid status transitions as a domain state machine; EF Core SaveChangesInterceptor captures before/after JSON audit snapshots on every entity write.
- Deployed to Azure App Service (F1) with Azure SQL Database (free tier) via GitHub Actions OIDC workload identity federation (zero stored credentials); EF Core migrations run automatically on startup; validated with xUnit 2.6 + Moq 4.20 unit tests covering handlers, pipeline behaviors, and SLA calculation.

## **Eskhawini LibraryOS** | *Full-Stack Library Management System — ASP.NET Core 8, React 19, Azure*

- Architected and deployed a production-grade REST API using ASP.NET Core 8 and Clean Architecture (3-layer: API / Core / Infrastructure), exposing 35+ endpoints across 7 controllers with JWT authentication, refresh token rotation, role-based access control (Admin, Librarian, Patron), and fixed-window rate limiting — hosted on Azure App Service and PostgreSQL v12 (Azure Flexible Server) with automated EF Core migrations on startup.
- Built a real-time notification system using SignalR WebSockets with group-based routing and a daily background service that probes active loans for due-date alerts; resolved a JWT refresh race condition on the frontend using a subscriber queue pattern, eliminating duplicate refresh calls across parallel Axios requests.
- Eliminated N+1 database queries by refactoring repository interfaces to use eager-loaded GetAllAsync() with .Include(); implemented a GlobalExceptionHandler returning structured ProblemDetails JSON; added PostgreSQL health check endpoints (/health/ready, /health/live) used as Azure platform liveness and readiness probes.

## **WardCare+** | *ASP.NET Core 8 MVC — Team Project, Final Year*

- Solely owned the cloud deployment pipeline for a 4-person team project: provisioned Azure App Service and Azure SQL Database, configured GitHub Actions CI/CD with OIDC workload identity federation (zero stored credentials), and automated EF Core migrations on startup — reducing every release to a single push to main.
- Implemented a POPIA-compliant audit trail using an EF Core SaveChangesInterceptor capturing before/after JSON snapshots on every INSERT, UPDATE, and DELETE across 25+ entities with a soft-delete pattern on all tables — validated with an xUnit + Moq test suite covering the interceptor at service-layer logic.

## **RaptorReview AI** | *GitHub Actions — Published on GitHub Marketplace*

- Published a GitHub Action on the GitHub Marketplace that automatically reviews every pull request using Groq LLM inference, posting line-specific security, performance, and readability comments directly in the PR diff interface — installable in one workflow YAML step with zero hosted infrastructure or paid API key required.
- Engineered diff-hunk-only context segmentation using tiktoken to keep token count proportional to change size (not file size), preventing context window overflows on real codebases regardless of file length.

## **ADDITIONAL PROJECTS**

---

**DocLens** (Python, FastAPI, ChromaDB, RAGAS) — PDF RAG system with hybrid BM25 + dense vector retrieval and cross-encoder reranking; +25–31% improvement across all four RAGAS metrics vs cosine-similarity baseline. [github.com/dev-k99/RAG-Assistant](https://github.com/dev-k99/RAG-Assistant)

**MemOS** (Python, LangGraph, PostgreSQL, LangSmith) — Agentic AI assistant with custom LangGraph StateGraph, persistent cross-session memory, and full LangSmith observability. [github.com/dev-k99/MemOS](https://github.com/dev-k99/MemOS)

**Kova** (React 18, TypeScript, Zustand, Paystack) — Production e-commerce storefront with ZAR checkout, admin dashboard, TanStack Query v5, URL-synced filters, and aria-live accessibility. [github.com/dev-k99/Kova](https://github.com/dev-k99/Kova)

## **CERTIFICATIONS & AWARDS**

---

**Angular & ASP.NET Core Web API** — *Udemy, Oct 2025*

**Introduction to Cybersecurity** — *Cisco Networking Academy, Oct 2023*

**AI Fluency Framework & Foundations** — *Anthropic, Mar 2026*

**Data Analyst Bootcamp: Python, Excel, Power BI & SQL** — *Udemy, Oct 2025*

**C# Programming Award** — *Nelson Mandela University, Apr 2024*

**Student Merit Awards (2024 & 2025)** — *Nelson Mandela University*