

Aigberua Nicholas

Lagos, Nigeria (Remote — Open to US / UK Time Zones) | +234 708 072 7247 | aigberuan6@gmail.com | github.com/Ghosteken | linkedin.com/in/aigberua-nicholas

PROFESSIONAL SUMMARY

Backend-focused Software Engineer with 3+ years of experience designing, building, and scaling **REST APIs** and backend systems using **Node.js**, **TypeScript**, **C#**, and **Python**. Proven track record leading API development, integrating third-party services, enforcing API versioning and backward-compatibility strategies, and optimizing database performance across **PostgreSQL**, **MongoDB**, and **Supabase**. Skilled in **CI/CD** automation, cloud infrastructure (**AWS**, **Azure**), and serverless architecture, with hands-on experience across the full engineering lifecycle — from system design and implementation to testing, deployment, and production support. Experienced working in agile, cross-functional teams using **Jira**, comfortable collaborating across distributed/remote teams spanning US, UK, and international time zones.

CORE COMPETENCIES

Languages & Frameworks: Node.js (NestJS, Express.js) • TypeScript • C# (.NET / ASP.NET Core) • Python • REST API Development

Databases & ORMs: PostgreSQL • MySQL • MongoDB • Supabase • Prisma • TypeORM • Mongoose • Redis

Cloud & DevOps: AWS (Lambda) • Azure (App Service) • Docker • Terraform • GitHub Actions • CI/CD Pipelines • NGINX

Testing & Tooling: Jest • Postman (API Testing) • Swagger (API Documentation) • Express Validator • Git / Version Control

Project Management & Collaboration: Jira • Agile / Scrum Methodologies • Confluence

PERSONAL PROJECTS

InsiderSignal Intelligence — MCP Server for Insider Trading Signals | github.com/Ghosteken/mcp-insidersignal

- Built a production-grade **Model Context Protocol (MCP)** server that unifies corporate insider-trading data (**SEC Form 4** filings) with **US Congressional trading disclosures** under the STOCK Act, giving AI agents a single interface onto both datasets.
- Designed an algorithmic "Golden Signal" detector that flags high-conviction trades by identifying when executives and legislators buy the same ticker within a 14-day window, layered with committee-enrichment mapping and legislation-overlap scoring.
- Ingested and indexed 60,000+ historical trading records, exposing five MCP endpoints (unified market signal feed, convergence detection, Congressional trading analysis by ticker, legislator profile deep-dives, and corporate insider buy-signal detection).
- Built with **Node.js**, **TypeScript**, **Express.js**, the **Model Context Protocol SDK**, **PostgreSQL** with optimized indexing, **Redis** caching, and structured logging via Pino for production-grade observability.

AI Work Assessment — Dual-Service Financial & Recruiting Backend | github.com/Ghosteken/AI-Work-Assessment

- Designed a dual-service backend: a **FastAPI** (Python) service that generates professional financial briefing reports, and a **NestJS** (TypeScript) service that automates AI-powered candidate document summarization for recruiter workflows.
- Integrated the **Google Gemini API** for AI-driven document summarization, implementing a queue/worker pattern with fire-and-forget background tasks and a state-machine approach to track summary status.
- Enforced workspace-level access controls for candidate data isolation and used normalized relational schemas over document storage to preserve data integrity across both services.
- Used **PostgreSQL**, **Pydantic** for validation, **Jinja2** for HTML report templating, **Docker**/docker-compose for containerization, and **Pytest**/PowerShell scripts for automated and manual test coverage.

Stock Studio — Multi-Tenant Inventory & Sales Platform (Full-Stack) | github.com/Ghosteken/server-invnt | github.com/Ghosteken/client-invnt

- Built the backend **API** for Stock Studio, a multi-tenant platform managing inventory, sales, purchasing, and reporting across multiple organizations with strict tenant-scoped data isolation.

- Implemented **JWT**-based authentication with role-based access control, comprehensive CRUD for products/invoices/purchases, and CSV/Excel import-export for bulk data operations.
- Added **Redis**-backed caching (Upstash-compatible, with in-memory fallback) for cached reporting dashboards, plus tenant-aware rate limiting via `express-rate-limit` to protect shared infrastructure.
- Delivered real-time client notifications with **Socket.IO**, and hardened the API with Helmet middleware and CORS configuration, built on **Node.js**, **Express.js**, **TypeScript**, and **Prisma/PostgreSQL**.
- Built the companion frontend with **Next.js 16** (App Router) and **React 19**, using **Redux Toolkit** with RTK Query for API calls/caching and Redux Persist for local storage persistence.
- Implemented dashboard analytics, inventory tracking with batch metadata, invoice management with VAT calculations, purchase order handling, and expense workflows with approval systems, styled with **Tailwind CSS** and Material-UI DataGrid, with Recharts for analytics visualizations and Socket.IO for real-time sync.

Agent Harness — Production-Grade Engineering Skills Framework for AI Coding Agents | github.com/Ghosteken/agent-harness

- Authored a framework of 173 curated engineering skills (24 core SDLC workflows) that package senior-engineer practices — including principles from *Software Engineering at Google* like Hyrum's Law and trunk-based development — into structured guidance AI coding agents can follow consistently.
- Designed a three-layer architecture: a command layer (`/spec`, `/plan`, `/build`, `/test`, `/review`, `/code-simplify`, `/ship`) mapped to SDLC phases, a persona layer of 12 specialist engineering roles (code-reviewer, api-developer, db-architect, etc.), and a skills layer of step-by-step workflows with built-in checkpoints and verification criteria.
- Built each skill to auto-activate based on detected intent/context, with reference checklists covering testing, security, performance, and accessibility.
- Made the framework platform-agnostic using plain Markdown skill files, supporting Claude Code, Cursor, Gemini CLI, Windsurf, GitHub Copilot, and other AI coding agents; released under MIT license.

Peppermint API Key Management System — Secure Backend Take-Home Assessment | github.com/Ghosteken/peppermint-BE-Assesment

- Built a secure API key management backend enabling users to generate (with a default 30-day expiration), list, revoke, and rotate API keys, with backward-compatible handling for legacy keys lacking expiration dates.
- Implemented **JWT**-based authentication for management endpoints, per-user authorization so users can only manage their own keys, and a configurable cap of 3 active keys per user to prevent credential sprawl.
- Enforced rate limiting (10 requests/minute per key) and request validation with `class-validator`, alongside comprehensive audit logging covering both key usage and management actions.
- Built with **NestJS** and **MongoDB** (using `migrate-mongo` for schema migrations), reaching 80%+ test coverage with **Jest**, documented via a Postman collection, containerized with **Docker**, and deployed to **AWS Elastic Beanstalk**.

PROFESSIONAL EXPERIENCE

Lead Software Engineer, **Appnable**

Nov 2025 – Present | Remote

- Lead backend development for Appnable's core platform, architecting and maintaining complex **Node.js/TypeScript** services, driving **API** logic design, and optimizing performance across high-traffic endpoints.
- Own database operations and authentication flows built on **Supabase**, implementing secure session handling, access control, and data validation to protect sensitive user data.
- Design and maintain **CI/CD** pipelines with **GitHub Actions**, automating linting, testing, and deployment steps to ship reliable releases with minimal manual intervention.
- Manage **Azure**-hosted infrastructure end-to-end, including App Service deployments, resource provisioning, scaling rules, and monitoring/alerting to keep services available and performant under load.
- Drive testing and debugging initiatives across services, triaging production incidents, identifying root causes, and implementing fixes that improved overall system stability.
- Contributed to backend services built with **C# (.NET / ASP.NET Core)**, working with controller-based APIs and integrating them with existing **Node.js** services in a multi-service environment.

- Built and maintained **REST APIs** in **Node.js** with **TypeScript**, integrating third-party services while keeping endpoints stable for client applications and internal teams; automated **GitHub Actions** workflows for testing, linting, and deployment, including a staging pipeline that eliminated hours of manual redeployments per week.
- Worked across **PostgreSQL** and **MongoDB** for event logging, selecting ORMs (**Prisma**, **TypeORM**, **Mongoose**) based on project needs, and managed **Jest**-based unit/integration test suites wired into CI/CD so every commit was validated before merge.
- Designed and enforced **API versioning** and backward-compatibility strategies, introducing `/api/v1` namespaces and legacy fallbacks to protect existing clients during incremental refactors and security hardening.
- Built serverless **AWS Lambda** functions integrated with **PostgreSQL** and **MongoDB**, handling connection reuse, cold-start mitigation, and query optimization to avoid performance degradation at scale.
- Tracked and managed engineering work in **Jira**, maintaining sprint boards, backlog grooming, and cross-team visibility into feature delivery.

Technologies Used: *Node.js • TypeScript • NestJS • C# (.NET / ASP.NET Core) • Supabase • PostgreSQL • MongoDB • Prisma • AWS Lambda • Azure • GitHub Actions • Jest • Jira • REST APIs*

Lead Backend Developer, **Peppermint**

Mar 2026 – Jul 2026

- Led backend development on **AWS**-hosted infrastructure, building and maintaining **NestJS** services backed by **MongoDB** for core product features.
- Designed and implemented **REST APIs** to support client-facing and internal integrations, with **Swagger** documentation kept current for all new endpoints.
- Built and scheduled **cron jobs** to automate recurring data processing and background tasks, improving reliability of time-sensitive workflows.
- Implemented **Google** and **Microsoft** integrations, including **calendar** sync and call-recall/meeting-recording integration, enabling seamless third-party account connectivity within the platform.
- Developed **calendar integration** features to support scheduling and availability sync across connected Google and Microsoft accounts.
- Planned and tracked sprint work in **Jira**, coordinating backend delivery with product and design stakeholders.

Technologies Used: *AWS • NestJS • MongoDB • REST APIs • Swagger • Cron Jobs • Google Calendar API • Microsoft Graph API • Jira*

Senior Lead Developer, **Sceptre**

Apr 2024 – Nov 2025

- Led product development and set technical direction for core platform features, translating business requirements into scalable architecture decisions.
- Coordinated project tasks across development, design, and product teams, keeping cross-functional delivery on schedule.
- Integrated **APIs** and backend services to enhance system performance and feature reliability.
- Reviewed code and development workflows, introducing standards that improved engineering efficiency and maintained project quality.
- Facilitated communication between stakeholders and team members to keep priorities aligned and execution on time.
- Used **Jira** for sprint planning, backlog prioritization, and tracking feature delivery across the engineering team.

Technologies Used: *Node.js • REST APIs • Agile Methodologies • Jira • Git*

Software Engineer, **Workbrook**

Apr 2025 – May 2025

- Participated in codebase reviews, assessing application structure and identifying opportunities to improve efficiency.
- Provided feedback on application functionality and development workflow to the wider engineering team.

Technologies Used: *Git • Code Review Tooling*

Backend Developer, **Timest Web**

Aug 2024 – Sep 2024

- Developed and secured **API** endpoints using **NestJS**, aligning implementations with industry and security standards.

- Identified and resolved backend issues, reducing system errors and improving overall application performance.
- Conducted thorough testing to confirm APIs were reliable, safe, and compliant with best practices.

Technologies Used: NestJS • Node.js • REST APIs

Network Security Intern, **Tiger-Logic Africa**

Mar 2024 – Sep 2024

- Handled testing and configuration of firewalls to meet organizational security standards.
- Applied technical troubleshooting skills to help reduce recurring system errors.
- Tested firewall configurations to confirm they were secure and compliant.

Technologies Used: Firewalls • Network Security Tools

Lead Backend Developer, **Wish to Wear**

Jan 2024 – Jul 2024

- Designed and developed **RESTful APIs** to facilitate efficient client-server communication.
- Validated user input against the database using **Express Validator** to maintain data integrity and security.
- Led API documentation efforts using **Swagger**, producing clear references for other developers.
- Conducted thorough API testing with **Postman** to verify functionality, reliability, and performance.
- Tracked development tasks and bug fixes in **Jira** to maintain clear visibility into sprint progress.

Technologies Used: Node.js • Express.js • Express Validator • Swagger • Postman • Jira

Full Stack Developer, **Edge Tech**

Jul 2023 – Jan 2024

- Designed client-side and server-side architecture and implemented cloud-based backend solutions.
- Integrated third-party **APIs** from external applications into web platforms.
- Built **RESTful APIs** with proper routing, authentication, and **Swagger** documentation.
- Optimized database queries and indexing strategies to improve overall database performance.

Technologies Used: Node.js • REST APIs • Swagger • SQL

EDUCATION

Anchor University, Lagos

Mar 2023 – Sep 2025

Bachelor's Degree — Second Class Upper Division (equivalent to a 3.3–3.7 / 4.0 GPA, or UK 2:1 Honours)

ADDITIONAL INFORMATION

Languages: English (Fluent)

Certifications: Network+ (Cybrary) • CCNA 200-301 Network Fundamentals (Simplilearn)

Work Authorization: Open to remote roles and relocation; available to work across US and UK time zones.