Contact

Kawasaki, Kanagawa

090-8004-5935

akiraki.awa@gmail.com

akirakii.github.io

linkedin.com/in/michael-akira-awa-vitug/

Technical Knowledge

- Python, Java & Go
- Design Patterns and Principles
- Microservices
- Docker, Linux, Git
- Authentication with Tokens and Sessions
- Clean Architecture
- SCRUM framework and AGILE philosophy
- Concurrency and Parallelism

Languages

- Spanish (Native)
- English (Native)

Hobbies

- Develop web applications that allow me to be more productive
- Develop videogames
- Competitive programing
- Compose music (jazz, classical, rock, etc.)

Michael Akira **Awa Vitug**

Back End Developer

Visit my website

Introduction

As a Back-End Engineer at Tiendada (a Shopify-like platform), I spent three years specializing in core **logistics, e-commerce,** and **financial** logic. My role focused on bug fixing, designing, and implementing new features to enhance platform functionality and performance.

Work Experience

Full-Time Backend Developer at Tiendada (Python, Java & Golang) | Shopify-like Web Application

2022-01– 2025-01 | Remote work

JAVA

Developed a REST API using Spring WebFlux and JPA Queries for **authentication**, **authorization**, and a **notification** system. Designed with non-blocking logic and reactive streams for high efficiency. Implemented raw SQL migrations with a custom CLI helper and structured the codebase following Clean Architecture.

IMPLEMENTED FEATURES:

- New store user creation
- Authentication and Authorization with Redis session and browser cookies
- Store notifications system (observer design pattern)
- Create store collaborator users with their own permissions and roles
- Customizable subdomain
- Tier plans authorization system

PYTHON

Developed a REST API using Flask and SQLAlchemy ORM for core **e-commerce**, **logistics** tracking, **financial** operations, and UI configuration. Designed and maintained four **microservices** (invoice logs, core e-commerce, UI configurations, and mailing). Managed database migrations with Alembic and structured the codebase following Clean Architecture. Refactored legacy code to improve scalability, maintainability, and performance.

ORDER AND INVOICE FEATURES:

- Core CRUD endpoints using builder design pattern for modular and reusable code. (i.e. shopping cart's order summary or checkout)
- Secure order status tracking logic
- Shipping fee system
- E-mail notification system
- Accumulated data from customers for analysis (and Whatsapp notifications reporting sales data weekly)
- Virtual receipt generation
- Complex invoice edit (always keeps old configuration in a log)

Contact

Kawasaki, Kanagawa

090-8004-5935

akiraki.awa@gmail.com

akirakii.github.io

linkedin.com/in/michael-akira-awa-vitug/

PRODUCT FEATURES:

- Core CRUD endpoints
- Taxes, coupons, sales and wholesale prices (and managed their complex interactions)
- Fast, modular and optimized SQL queries for massive product data fetch
- Fast massive bulk edit
- Stock management
- Customizable variants and categories
- Reviews from customers
- Get similar products
- PDF generation for product catalogue
- Many UI configurations (website color theme, highly customizable layout, product tags, etc.)

GO

Developed a REST API using Gin and Gorm for a real-time concurrent **messaging** system. Integrated WhatsApp, Facebook Pages, and Instagram APIs for seamless communication. Managed database migrations with raw SQL and a custom CLI helper, structuring the codebase following Clean Architecture.

IMPLEMENTED FEATURES:

- Centralized chat management from multiple different sources (decorator design principle)
- Can upload videos, audios, images and documents
- Create new conversations in real time

SQL

We use PostgreSQL for all 6 microservices. 6 database instances in total for each microservice. I use SQL for creating **migration** files, fixing difficult migrations caused by bugs, create or modify tables for new features and optimizing queries for massive data fetchs. Also create **triggers** for custom logic and store **procedures** for utility.

DOCKER AND AWS

We use Docker for both development and deployment. Migrated from **EC2** to **ECS** for improved container orchestration in the cloud. I frequently deploy to development containers via SSH. Additionally, I led the migration from Cloudinary to Amazon **S3** for media file storage.

For personal projects, I prefer working with Docker **containers** for VMs and use Docker **images** with PostgreSQL to manage different migration changes efficiently.

SCRUM AND AGILE

I am **SCRUMstudy certified** (click here) and highly experienced in daily stand-ups, sprint planning, and retrospectives. Worked closely with Product Owners and Scrum Masters to drive efficient Agile development.

Education

Bachelor in Software Engineering

2018-03– 2022-012

UPC: Universidad Peruana de Ciencias Aplicadas – Av. La Marina 2810, San Miguel 15087, Lima, Peru

ACHIEVEMENTS

- Consistently led all my projects in every course.
- Consistently ranked among the top third and top tenth academically in all semesters.