

# Charles Asiama

casiama@alumni.cmu.edu | (312) 617-1166 | charasi.com | github.com/charasi | linkedin.com/in/charles-asiama

## Personal Statement

Software Engineer proficient in Go, React, and TypeScript. Experienced in scalable backend systems, interactive frontends, and cloud-native infrastructure. Focused on clean architecture, resilient design, and continuous learning.

## Education

**Carnegie Mellon University** – MS in Mobile and IoT Engineering

September 2021 - May 2024

**University of Wisconsin-Madison** – Capstone Certificate in Computer Science

June 2019 - August 2021

**Western Illinois University** – BA in Economics

January 2006 - August 2010

## Skills

**Languages:** Go, TypeScript, Java, Python, C, SQL

**Frontend:** React, Tailwind CSS, HTML, PixiJS, GSAP

**Backend & APIs:** Node.js, Express.js, gRPC, REST

**Cloud & Infrastructure:** AWS (Lambda, EC2, EKS), GCP (GCE, Load Balancing, Storage), Docker, Kubernetes

**DevOps & Automation:** Terraform, Ansible, Jenkins, GitHub Actions, CI/CD

**Testing & QA:** Unit testing, Integration testing, Regression testing, SQL-based test scripting

**Databases & Storage:** MySQL, MongoDB, Timestream

**Certifications:** Google Cloud Associate Cloud Engineer

## Experience

**Practicum**, SLAC National Accelerator Laboratory – Pittsburgh, PA

September 2023 - December 2023

- Architected event-driven AWS infrastructure for energy forecasting using Lambda and Timestream, ensuring 99.9% uptime and real-time efficiency in grid analytics
- Designed a Python-based hot-reload system for Lambda, enabling instant code deployment with automated CloudWatch logging, cutting debug cycles by 40%

**Cert Workforce Development Intern**, Software Engineering Institute – Pittsburgh, PA

May 2022 - December 2022

- Built a React-Node web app with Express.js and MySQL, applying modular design and RESTful principles to automate cybersecurity workflows, accelerating training delivery and reducing manual processing time by 60%
- Implemented Infrastructure as Code using Terraform on Proxmox, streamlining deployment and improving system reliability and scalability

**Software Quality Engineer I-II**, CDW – Chicago, IL

September 2011 - May 2019

- Automated over 1,200 backend tests via SQL and UFT, accelerating regression cycles by 30% and increasing defect detection by 25%
- Integrated Application Lifecycle Management (ALM) tools into continuous testing pipelines, automating test execution and reducing manual intervention by 40%, while improving platform resilience
- Generated bi-weekly regression reports and dashboards with Excel and ALM reporting tools, improving platform health visibility and accelerating risk response

## Projects

**Raft** | Golang, Linux/Unix, Multithreading, Reflection, RPCs, Synchronization, TCP/IP

- Built a Go-based Raft protocol with custom RPC library for leader election, log replication, and fault recovery in distributed systems
- Enhanced fault-tolerance via multithreaded node communication and state synchronization

**AlgoWB** | React, Typescript, Pixi.js, GSAP

- Developed AlgoWB, an interactive algorithm visualization platform using TypeScript, PixiJS, and Monaco Editor, leveraging the TypeScript Compiler API to instrument code and replay debug logs for step-by-step algorithm execution
- Implemented AlgoWB features with a PixiJS visualization pane for arrays, stacks, queues, trees, and graphs, and a debugger pane with call stacks, watch variables, and execution logs, integrated with Monaco Editor for interactive TypeScript coding

**E-commerce Web-Service** | Golang, Docker, AWS (CloudFormation, EC2, EKS, MongoDB, MySQL)

- Developed a high-performance, scalable e-commerce backend in Go with RESTful APIs, JWT authentication, and microservices using MySQL and MongoDB, optimizing large-scale transaction handling and user experience
- Created fault-tolerant backend services with circuit breakers, bulkheads, and Kubernetes on AWS EKS, enhancing system reliability, resilience, and automating failover processes

**Santorini Game Engine** | React, Typescript, Pixi.js, Spine 2D, GSAP

- Developed an isometric board game using PixiJS, Spine 2D, and GSAP, with decoupled animation and game logic for smooth UI/UX transitions
- Seamlessly synced PixiJS canvas with React UI, enabling scalable scene management and animation sequencing

**Cloud Infrastructure & K8 Automation** | Ansible, Jenkins, GCP, Kubernetes, Python, Terraform

- Orchestrated Kubernetes clusters on GCP with automated VPC provisioning and load balancing, enabling scalable and highly available deployments
- Streamlined a CI/CD pipeline with Python automation and SSH key management to enable secure cluster provisioning and automated backups