

Chibuzor Okwusiuno

647-551-0922 | okwusiunosamuel@gmail.com | [linkedin.com/in/chibuzor-okwusiuno-24041b271/](https://www.linkedin.com/in/chibuzor-okwusiuno-24041b271/) | samokw.vercel.app

EDUCATION

University of Guelph

Bachelor of Computing: Software Engineering

Guelph, ON

Expected April 2026

EXPERIENCE

Coding Instructor

June 2023 – Present

Zebra Robotics

Brampton, ON

- Instructed students from grades 1–12 in fundamental concepts of engineering, computer science, coding, and robotics, enhancing their understanding and interest in STEM fields.
- Enhanced understanding of Object-Oriented Programming and Memory Allocation for students by developing age-appropriate analogies and interactive coding challenges
- Instructed C, Python, Java, JavaScript, HTML/CSS courses

PROJECTS

FixtureSphere | *Spring Boot, ReactJS, SQL, AWS EC2, S3* | <https://github.com/samokw/Premier-League-Dashboard>

- Engineered a **full-stack solution** using **Spring Boot** (adhering to **SOLID principles** and **MVC pattern**) and **ReactJS**, enabling users to efficiently access and analyze statistics for 20,000+ fixtures through **custom REST API endpoints** and **interactive data visualizations**.
- Optimized **content delivery speed** by 20% through deployment on **AWS EC2** (backend) and **S3** (frontend), leveraging **CloudFront caching** as a **Content Delivery Network (CDN)** to ensure faster load times and reduced latency for global users.
- Designed with scalability in mind, incorporating **horizontal scaling** strategies for seamless expansion as user demand grows.
- Developed a **responsive, data-driven dashboard** using **React** and **Material UI**, incorporating **accessibility features** such as high-contrast themes to provide an inclusive user experience.
- Enhanced security by integrating **HTTPS** and DNS routing via **AWS Route 53**, ensuring safe and reliable content delivery.

Application Load Balancer | *Go* | <https://github.com/samokw/loadbalancer>

- Designed and implemented a load balancer, handling up to three backend servers with **dynamic health checks (5-second intervals)** and **automatic recovery mechanisms (60-second revival checks)**, ensuring high availability and minimal downtime.
- Developed a least-connections algorithm to distribute requests fairly, reducing server load disparity by **30% compared to the round-robin approach**.
- Demonstrated system reliability and scalability by ensuring continuous service availability with **99% uptime**, achieving **100% traffic rerouting** to healthy servers during failures, and recovering failed servers within a maximum of **60 seconds**.

Chess Web Application | *Python, C (SWIG), JavaScript, HTML, CSS, SQLite* | <https://github.com/samokw/chess>

- Developed a multiplayer chess variant application combining **C-based game logic wrapped with SWIG** and a Python **HTTPServer** backend, enabling real-time gameplay and player matchmaking.
- Engineered an interactive frontend using **Chessboard.js** and Bootstrap, delivering real-time board updates, move replays, and intuitive game analysis.
- Implemented robust **C functions** for move validation, rule enforcement, and time tracking, ensuring precise gameplay and optimal performance.
- Integrated **AJAX polling** and SQLite to achieve seamless data synchronization, reducing latency and enhancing the multiplayer experience.

TECHNICAL SKILLS

Languages: Java, Python, C, SQL, JavaScript, TypeScript, R, Go

Frameworks/Libraries: ReactJS, Node.js, Next.js, Pandas, Matplotlib, Seaborn, Supabase, PostgreSQL, MySQL, Material UI, TailwindCSS, Chi

Tools: Git, Docker, Spring Boot, AWS (Lambda, EC2, RDS, CloudWatch, S3, CloudFront, Route 53), Vercel, Linux, Agile Methodologies