

# SAMUEL KIM

☎ (778) 953-2568 ✉ samuelk2@student.ubc.ca 🌐 samuelyongkim.com 🔗 linkedin.com/in/samuelkim019 📀 daedal00

## SUMMARY

Computer Science undergrad with full-stack experience in fintech and retail tech. Delivered scalable web apps at Savi Finance (1.7k+ users) and built POS-eCommerce integrations at ConnectPie with Python, FastAPI, and AWS. Eager to apply agile development skills to create engaging e-commerce experiences.

## EXPERIENCE

### **ConnectPie – POS/E-Commerce Startup**

**July 2025 – Present**

**Software Engineer (Part-time)** | *Python, FastAPI, AWS, PostgreSQL*

Port Coquitlam, BC

- Designed and implemented a multi-tenant order delivery system integrating Ecwid e-commerce with legacy Windows POS systems, ensuring near real-time, fault-tolerant order handoff for local restaurants.
- Built REST APIs in FastAPI for webhook ingestion and POS agent polling, with idempotent delivery state tracking in PostgreSQL to guarantee zero order loss even when offline.
- Provisioned AWS infrastructure (API Gateway, Lambda, RDS) via Pulumi (Infrastructure-as-Code), enabling reproducible deployments and cost-efficient scaling.
- Collaborated with founders to define technical specs and data models, producing Confluence documentation for system design and future scaling.

### **Savi Finance – Fintech startup (1.7k users)**

**Jan. 2024 – Aug. 2024**

**Software Engineer Intern** | *TypeScript, GraphQL, Node.js, Nx*

Remote

- Launched a core income/expense categorization feature in 7 days, which enabled the product team to ship ahead of schedule and meet customer expectations.
- Automated transaction-email ingestion pipeline with AWS (SES,SNS,SQS) and OpenAI 4o-mini, improving data consistency for 400+ users and eliminating manual categorization steps (~35 sec saved per email).
- Led an 8-step, zero-downtime database migration, ensuring uninterrupted service across all financial platforms.
- Verified GraphQL microservice endpoints using Playground, curl & MongoDB Compass, and implemented Jest smoke tests to catch regressions before production, boosting release confidence.
- Authored three Agile-ready technical specifications (API design, workflow automation, categorization logic) alongside PMs, designers, and consultants to accelerate releases.

## PROJECTS

### **Muse** | *Go, GraphQL, React, PostgreSQL, Redis, Docker*

- Designed & implemented a GraphQL API (30+ types/resolvers) in Go using cursor-based pagination and connection patterns, enabling efficient, scalable music discovery.
- Engineered a Redis-based caching layer for Spotify API calls, slashing album search latency by 87% (665→81 ms).
- Integrated Spotify Web API via OAuth2 to ingest real-time song and album metadata and manage user authentication.
- Implemented a GitHub Actions-based CI/CD pipeline with automated unit and integration tests, benchmarking, linting, and security scans—achieving 68% overall code coverage.

## EDUCATION

### **The University of British Columbia**

**December 2025 (Expected)**

*Bachelor of Science, Combined Major in Science (Computer Science & Physics)*

Vancouver, BC

**Relevant Coursework:** Data Structures and Algorithms, Database Systems, Computational Data Science, Intro. Artificial Intelligence, Computer Networking, Reliability and Security, Computer Systems and Hardware

## SKILLS

**Languages:** Python, Go/Golang, Javascript, TypeScript, HTML, CSS, SQL

**Frameworks:** Node.js, Express, gqlgen, Spring Boot, FastAPI, JUnit

**Libraries & APIs:** GraphQL, REST API, MongoDB, Mongoose, React, JSON-RPC, Nx

**Tools:** Git, Linux, Postman, Docker, Jira, Confluence, GitHub, Redis, AWS