

KARIM KHALIL

Vancouver, BC | karim.waleid@gmail.com | [LinkedIn](#) | [GitHub](#) | [Portfolio](#)

EDUCATION

University of British Columbia | B.Sc. Computer Science

Sept 2021 – May 2026

GPA: 4.33 · Dean's List 2022–2026

TECHNICAL SKILLS

Languages: Java, C#, Python, SQL, TypeScript, JavaScript

Frameworks & Libraries: Spring Boot, FastAPI, Flask, React, .NET, D3.js

Infrastructure & Data: Docker, GitHub Actions, PostgreSQL, SQLite, CI/CD, pandas

EXPERIENCE

Software Engineer, Intern

May 2025 – Aug 2025

Royal Bank of Canada (RBC) | Global Asset Management

Vancouver, BC

- Helped kick off a migration of a legacy portfolio-management platform from C# .NET to Java Spring Boot REST APIs
- Built an order-validation API in Spring Boot, a middle-tier service called by multiple internal order-entry applications, applying business rules to each order in a transaction and returning per-order status to the UI
- Secured the order-validation API behind the API gateway so only approved internal clients could reach it, and held 90%+ unit test coverage (JUnit, Mockito) on endpoints I owned

QA Automation Engineer, Intern

Sept 2024 – Apr 2025

Royal Bank of Canada (RBC) | Global Asset Management

Vancouver, BC

- Engineered Python and Selenium automation suites for a Canadian institutional compliance platform, extending automated regression beyond the UI into API and database validation
- Architected a parallelized Playwright and Docker framework with Allure reporting and automated nightly Airflow pipelines to replace sequential regression, reducing regression time 90%
- Presented and led a live demo of the compliance platform to senior stakeholders

Software Engineer in Test

May 2026 – Sep 2026

Minga | K-12 Campus Management Software

Kelowna, BC

- Led an optimization of a ~3,600-test Playwright suite, replacing per-test UI login (~8s) with role-scoped session reuse via storageState and IndexedDB, cutting full-run wall-clock time 50% (from 110 min to 55 min) and saving ~5 hrs of CI machine time per run
- Designed and implemented end-to-end Playwright test cases for product features and split monolithic spec files into smaller parallel suites, partnering with feature engineers to expand automated regression coverage

Teaching Assistant | COSC 301, Intro to Data Analytics

Jan 2026 – Apr 2026

University of British Columbia

Kelowna, BC

- Evaluated assignments and midterms for 120+ students across Python (pandas), SQL queries, and data-analysis pipelines

Research Assistant

May 2024 – Aug 2024

University of British Columbia

Kelowna, BC

- Built a full-stack web platform (D3.js, Flask, PostgreSQL) to model and visualize patent lineages
- Web scraped and structured patent data into a queryable relationship graph for tracing technological evolution

PROJECTS

Big Picture | AI Portfolio & Code Analysis Platform

[GitHub](#) | Python, FastAPI, React, TypeScript, SQLite, Gemini

- 1st Place (of all capstone teams), plus Best UI/UX and Most Innovative (Students' Choice)
- Built a multi-language static code-analysis engine using tree-sitter to extract imports, entities, and complexity metrics across many languages, detecting architectural patterns to auto-generate resume-ready summaries of a developer's work
- Engineered a project-deduplication system using file-level signatures with Jaccard similarity and containment scoring to eliminate redundant re-analysis of previously scanned repositories
- Built the portfolio aggregation layer (FastAPI and SQLite) computing skills timelines, language distribution, and a git-history collaboration network, surfaced in a cross-platform Electron and React desktop app

Solid | Friends-Only Task Marketplace (iOS + Web)

[GitHub](#) | Java, Spring Boot, PostgreSQL, Next.js, TypeScript, Capacitor

- Built a full-stack marketplace app solo end to end, from database schema to a shipped iOS build — a Spring Boot and PostgreSQL backend with 72 REST endpoints, a Next.js and TypeScript frontend, packaged as a native iOS app with Capacitor and deployed on Railway and Vercel
- Designed a friends-only trust model enforced at the service layer rather than the UI, plus a task lifecycle state machine moving requests from open to accepted to in progress to completed, with poster approval and cancellation handling
- Integrated PayPal Orders and Payouts with idempotency handling, JWT authentication with BCrypt