

Sam Richell-Smith

+64 210 285 6358 | samrs6012@gmail.com | github.com/samrichell-smith | linkedin.com/in/sam-richell-smith | samrichellsmith.com

EDUCATION

The University of Auckland

March 2024 – Nov 2027

Bachelor of Engineering (Honours) – Software Engineering, GPA: 8.7/9.0 (A/A+)

- **Dean's Honours List 2024, 2025** – Awarded to the **top 5%** of the cohort

EXPERIENCE

Software Engineer Intern → Software Developer (Part-Time)

Nov 2025 – Present

Figured

Auckland, NZ

- Independently identified a bottleneck and designed a **Redis caching layer** for user exports, reducing latency from **10s to ~3s (3x speedup)**; authored a design doc and drove the solution through engineering leads and DevOps.
- Built an **AI-powered merge conflict detector** using a **multi-stage LLM pipeline** that proactively surfaces high-risk conflicts before code review; containerised with **Docker**, deployed as a **Kubernetes Argo app**, and integrated into the **CI pipeline** with **Slack** alerts – adopted across the full engineering team.
- Led migration of **20+ components** as part of a planned **Vue 2 → Vue 3** codebase-wide upgrade driven by Vue 2 EOL – resolving breaking API changes including **Composition API** rewrites, **v-model** argument changes, and **\$attrs/\$listeners** consolidation; owned post-ship **production triage**, pushing hotfixes to stabilise the release.
- Shipped features across a **Laravel/PHP** and **Vue.js** full-stack codebase; **retained post-internship** as one of the few interns offered a continuing role – selected for demonstrated ownership and production impact from day one.

Software Developer → Tech Executive

April 2025 – Present

Web Development and Consulting Club (WDCC)

Auckland, NZ

- Built a **full-stack investment platform** serving **1500+ members** in **TypeScript/Next.js/MongoDB**, shipping a **live stock tracker**, **Google OAuth**, and **Stripe** payments within a **10-person agile team** following sprints, standups, and retrospectives.
- Elected to the **WDCC executive team** as Tech Executive; responsible for building and maintaining club apps and event sites across the organisation.

Lead Software Developer

Oct 2024 – March 2025

University of Auckland Swedish Club

Auckland, NZ

- Built a **Go REST API** and **full-stack executive dashboard** in **TypeScript/React/Tailwind CSS** to manage club tasks, surface overdue items, and coordinate across the executive team.
- Wrote a comprehensive **Go** test suite covering all API endpoints, achieving **100% test coverage**.

COMPETITIONS & LEADERSHIP

AllUni Competitive Programming 2025 – **1st in New Zealand**, 3rd Internationally

Figured Student Hackathon – Won **Leadership**, **Best UX**, and **Best Design**; led directly to an **internship offer**

Microsoft & Fonterra AI Challenge 2025 – **3rd Place**, Best AI Solution; built a **Power BI** analytics platform with a **multi-persona AI agent** tailored to executives, developers, and stakeholders

Susquehanna Trading Algothon 2025 – Backtested **momentum** and **mean reversion** trading strategies in **Python** and **NumPy**; evaluated on **Sharpe ratio** and **maximum drawdown** across varying lookback windows.

Founder & Tech Executive, Auckland Quant Society – Founded the university's quantitative finance society, building a community for students interested in algorithmic trading and quantitative research

Vice President & Co-Founder, UoA Swedish Club – Scaled from 3 to **50+ members** in under a year

TECHNICAL SKILLS

Languages: Go, TypeScript, JavaScript, C++, Java, Python, PHP, SQL, HTML/CSS, Bash

Frameworks & Libraries: React, Vue.js, Next.js, Node.js, Express.js, Laravel, Tailwind CSS, JUnit

Concepts: Full-stack development, system design, concurrent programming, REST APIs, caching

Databases: PostgreSQL, MySQL, MongoDB, Redis

Tools & Infrastructure: Git, Docker, Kubernetes, Argo Workflows, Linux, CI/CD, Nginx, REST APIs, Stripe, GitHub, GitLab, Power BI

AI & Developer Tools: Claude Code, AI-assisted development, Prompt Engineering

Relay – Real-Time Chat Application | *React, Node.js, Express.js, MongoDB, Socket.io*

[GitHub](#)

- Built and deployed a full-stack messaging app with **JWT/bcrypt authentication**, **real-time messaging** and image sharing via **Socket.io** WebSockets, and live **online presence indicators**.
- Designed a **responsive React** frontend with dynamic theme customisation and profile management; deployed end-to-end on **Render**.

Distributed Job Scheduler | *Go, React, TypeScript, PostgreSQL, Redis*

[GitHub](#)

- Built a fault-tolerant distributed scheduler to reliably execute concurrent jobs across multiple workers without loss or duplication – using a **heap-based priority queue** matching jobs to workers by thread capacity and **per-worker goroutine pools** with **chunked parallel execution**.
- Implemented **mutex-based result aggregation** and **dual-layer persistence** via **Redis** and **PostgreSQL**, ensuring job durability and crash recovery across worker failures.

OHIF Viewer Export Extension | *TypeScript, React, OHIF, JSZip*

[GitHub](#)

- Built an extension for OHIF Viewer – a production medical imaging platform with **7k+ GitHub stars** – adding client-side **DICOM export** that packages images and structured study metadata into a ZIP using **JSZip** and browser APIs, navigating a large-scale open-source codebase independently.

High-Performance Time-Series Database | *C++17, CMake*

[GitHub](#)

- Built an in-memory **columnar storage engine** storing timestamps and values in separate contiguous arrays for **cache-efficient** sequential access, minimising cache misses under sustained ingestion load.
- Achieved **22M points/sec ingestion** and **40µs range queries** via **delta encoding** and **binary search indexing** – a **17–30x speedup** over naive storage, benchmarked end-to-end with CMake test harnesses.

Options Pricing Engine | *C++17, Python, pybind11*

[GitHub](#)

- Built a **Black-Scholes** pricing library in **C++17** with analytical **Greeks** (Δ , Γ , ν , Θ) and a vectorised batch pricer benchmarked at **17.8M contracts/sec**; exposed via **pybind11** Python bindings.
- Applied engine to live **SPY options chain** data to extract **implied volatility** per strike via **Newton-Raphson**; plotted the **volatility smile** and **IV surface**, empirically surfacing where Black-Scholes' constant-volatility assumption breaks down in real markets.