Anson Chan

(604) 889-2211 | ansonnchan95@gmail.com | linkedin.com/in/ansonnchan | github.com/ansonnchan | ansonnchan.dev

EDUCATION

University of British Columbia Vancouver, BC
Bachelor of Applied Science in Computer Engineering **Expected May 2028**

- Coursework: Data Structures & Algorithms, Operating Systems, Machine Learning, Object-Oriented Programming

EXPERIENCE

Software Engineering Intern **Jun 2026 – Present**
ScalePad *Vancouver, BC*

- Built an internal database operations dashboard for **50+ engineers**, replacing manual SRE requests with audited, dry-run-safe SQL workflows and cutting turnaround time by **94% (4 hours to 15 minutes)**
- Implemented customizable role-based access control across **13 C#/.NET REST endpoints**, replacing coarse permissions with granular authorization for a platform serving **12,000+ MSPs**
- Built an employee offboarding reporting API, reducing PostgreSQL query latency by **99% (700 ms to 4 ms)** across **39M rows** through pre-aggregation and partitioned indexing
- Deployed Prometheus/Grafana monitoring and Terraform-managed alerts across **130,000+ desktop agents** to detect resource regressions during staged releases

Software Engineer **Jan 2026 – May 2026**
Borrow'd *US, Remote*

- Built a Django analytics pipeline processing **4,000+ weekly searches**, identifying popular items for homepage promotions and reducing weekly merchandising analysis by **67% (15 to 5 hours)**
- Implemented server-side authorization and join-request approvals for private item-sharing groups, restricting access to approved members across a **600+ user pilot**

PROJECTS

Pear Programming: Collaborative Browser IDE *Java, Spring Boot, TypeScript, PostgreSQL, Redis, Yjs*

- Built a collaborative browser IDE supporting real-time editing, live code execution, and persistent team workspaces
- Achieved **40 ms p95** edit latency across **5 concurrent users** using Yjs CRDTs and Redis Pub/Sub
- Built a Redis-backed Judge0 execution queue with idempotent submissions, automatic retries, and worker recovery to prevent duplicate execution and recover from failures

nemu: Desktop App Activity Tracker *Go, TypeScript, React, AWS, Terraform*

- Built a desktop activity tracker with a Go agent and React dashboard to visualize daily app usage and active time
- Built a Terraform-managed AWS pipeline (Lambda, API Gateway, S3, DynamoDB) for hourly telemetry ingestion
- Prevented lost or duplicate uploads using atomic local storage, idempotent writes, and automatic retries

Sloth: HTTP Latency Profiler & Regression CLI *Go, Cobra, net/http/httptrace, GitHub Actions, GoReleaser*

- Built a Go CLI profiling **7 HTTP latency phases**, using **6 diagnostic rules** to identify API request bottlenecks
- Benchmarked connection reuse and concurrency across **600 requests** with rate-limited, controlled scheduling
- Built a CI regression gate comparing versioned reports to flag **p50/p95/p99 latency regressions**

SKILLS

Languages: Python, Java, TypeScript, JavaScript, Go, C/C++, C#, SQL
Frameworks: React, Next.js, Spring Boot, FastAPI, Django, .NET, Node.js
Databases & Cloud: PostgreSQL, MySQL, DynamoDB, Redis, AWS, Terraform, Docker, GitHub Actions