Ryan Qi

(647) 853-6680 | r28qi@uwaterloo.ca | linkedin.com/in/ryan-qi | github.com/Lushenwar | ryanqiportfolio.vercel.app

Education

University of Waterloo | *Bachelor of Mathematics – Data Science* Expected: Apr 2030
Relevant Coursework: Data Structures, Algorithm Design, Computer Systems, Linear Algebra, Probability

Experience

RBC Capital Markets | *Incoming Software Developer Intern* Jan 2027 – Apr 2027
- Joining the **Quantitative Technology Services (QTS)** team in Toronto.

Mon Sheong Foundation | *Software Engineer Intern* May 2026 – Aug 2026
- **Single-handedly digitized** Mon Sheong's fully paper-based care workflows into an end-to-end **Next.js/Node.js** clinical portal, cutting patient record retrieval time by **40%** across daily nursing operations.
- Engineered **role-based access control** for a care-workflow EHR's **Express APIs** and **PostgreSQL** schemas, scoping every query by caregiver role to restrict patient-record access for **400+ staff across 30 roles.**
- Built real-time **React** dashboards and automated threshold-triggered alerts for clinical metric monitoring.

Waterloo Aerial Robotics Group (WARG) | *Embedded Flight Software Developer* Sep 2025 – Jan 2026
- Refactored **C++** autonomous flight modules on custom **UAV** flight controllers, eliminating dynamic memory allocation from the **real-time** control loop to guarantee deterministic, bounded execution.
- Restructured C++ header dependency trees and build scripts, reducing firmware compilation time by **25%**.
- Optimized sensor fusion filters for IMU processing to eliminate jitter, achieving **95%** signal consistency.

UpNxxt (via Riipen Labs) | *Research Lead* May 2026 – Jun 2026
- Led **UX research** for the AI Sandbox, analyzing competitor engagement loops to cut onboarding friction.
- Designed child-first mechanics adopted into the roadmap, contributing to a **15% DAU** lift over six months.

Projects

Sentinel (AI-Boosted Incident Response) | *FastAPI, Next.js, TypeScript, PostgreSQL* Jul 2026
- Architected a **FastAPI** incident-response engine that cross-references **git** history against error logs, surfacing the faulty commit as the top candidate in ∼**13s** across DB, memory-leak, null-pointer, and other bug classes.
- Engineered a **RAG** pipeline over operational runbooks that generates root-cause diagnoses and remediation steps in ∼**14s**, automatically posting postmortems to **Slack** when error rates cross thresholds.
- Built a **Next.js** dashboard streaming live incident feeds and diffs with an interactive postmortem workspace.

Eco-Pulse (Urban Heat Mitigation Planner, 1st Place @ GenAI Genesis) | *React, FastAPI, scikit-learn* Mar 2026
- Processed **178K** land parcels in **GeoPandas**, filtering out buildings and roads to isolate plantable ground.
- Clustered parcels into risk tiers with **K-Means**, merging **3.8K** high-risk cells into **2,052** ranked planting zones.
- Automated cooling intervention plans with **Gemini** and albedo models, cutting generation speed to <**5s**.

Ferryman (Zero-Downtime Postgres Migrator) | *Go, PostgreSQL, Logical Replication, Docker, CI* Aug 2026
- Built a **Go** CLI orchestrating zero-downtime **Postgres** cutovers, backfilling in parallel ctid page ranges and streaming changes via **logical replication**, gating cutover on replication-lag prechecks to prevent data loss.
- Enforced DDL execution ordering and implemented automated rollback paths to restore source state on failure.
- Wired **CI** workflows testing migrations against seeded Postgres instances to prevent deployment regressions.

LOCATR (Multi-Agent Venue Planner, Winner @ DeerHacks V) | *FastAPI, LangGraph, Snowflake* Feb 2026
- Orchestrated a fault-tolerant 6-agent **LangGraph** pipeline that routes queries by complexity tier and fans out vibe, cost, and risk analysts concurrently via **asyncio**, returning ranked venues in <**8s** cold and <**1s** cached.
- Built a **Snowflake** risk memory that batch-fetches historical venue vetoes for all candidates in one ∼**0.8s** query.

BlackSwan (Quant Model Stress-Tester) | *Python, TypeScript, VS Code Debug Adapter* Apr 2026
- Engineered a **VS Code** extension stress-testing quant models via **Monte Carlo** sampling and adversarial inputs.
- Extended debug adapters to bisect parameter space and isolate breaking points, cutting validation time by **40%**.

Technical Skills

Languages: Python, C/C++, Go, TypeScript, JavaScript, Java, SQL, Bash
Frameworks/Libraries: React, Next.js, Node.js/Express, FastAPI, LangGraph, PyTorch, Pandas, NumPy, Redis
Tools/Cloud: PostgreSQL, MongoDB, Snowflake, Docker, AWS (S3, EC2), GitHub Actions, Git, Linux