

# RICKY TANG

437-243-5327 | [rickytangdev@gmail.com](mailto:rickytangdev@gmail.com) | [linkedin.com/in/ricky-tang-dev](https://linkedin.com/in/ricky-tang-dev) | [github.com/rickytang666](https://github.com/rickytang666) | [rickytang.dev](https://rickytang.dev)

## EDUCATION

University of Waterloo

Honours Bachelor of Software Engineering (Co-op)

Cumulative GPA: 4.0

## TECHNICAL SKILLS

**Languages:** Python, TypeScript/JavaScript, C++, SQL, Swift, C, Bash, HTML, CSS

**Technologies:** React, Next.js, Node.js, Express, FastAPI, Flask, RESTful APIs, LangChain, PyTorch, OpenCV, Pandas, React Native

**Tools:** Git, Docker, Linux, CI/CD, MySQL, PostgreSQL, Redis, SQLite, MongoDB, Firebase, Supabase, ChromaDB, GCP, Cloudflare

## EXPERIENCE

### Hamming AI (YC S24)

Software Engineer

May 2026 – Aug. 2026

San Francisco, CA

- Incoming summer 2026

### WAT.ai | Python, FastAPI, LangChain, ChromaDB, TypeScript, React, DataLab

Software Engineer

Jan. 2026 – Present

Waterloo, ON

- Developed self-correcting RAG system for **174-page EPA regulatory PDFs** in collaboration with **Bindwell (YC W25)**
- Achieved **1.00 faithfulness score** (zero hallucinations) via **DeepEval** benchmarking across golden evaluation dataset
- Engineered recursive table restoration with **100% fragment recovery** and **26% token reduction** via uniqueness filtering
- Implemented hybrid **vector + BM25** retrieval with lazy sampling, achieving **20-100x speedup** over full concatenation

### WATonomous | C++, ROS 2, Docker

Software Engineer

Jan. 2026 – Present

Waterloo, ON

- Architected complete **ROS 2 navigation stack** including Costmap, Map Memory, A\* Planner, and Pure Pursuit Controller
- Engineered **adaptive A\* planner** with iterative cost relaxation, achieving **100% path success** in constrained spaces
- Amplified grid resolution by **100% (5cm)** with a **50% larger safety buffer (1.5m)**, eliminating corner-cutting collisions

### Waterloo Aerial Robotics Group (WARG) | Python, React, Flask, OpenCV, MAVLink, Docker

Software Engineer

Oct. 2025 – Present

Waterloo, ON

- Streamlined ground station UI with one-click pause/resume for missions, eliminating manual switching for **50+ operators**
- Reduced mission failure recovery time from 30s+ to 3-5s for command pipeline operations (**85% improvement**)
- Engineered full-stack control pipeline with React frontend, Flask-SocketIO backend, and MAVLink for real-time commands
- Implemented OpenCV object detection in aerial imagery and MAVLink telemetry streaming, achieving **80%+ IoU accuracy**

## PROJECTS

### Quota – VS Code Extension for Cost Optimization | TypeScript, VS Code API, LangChain, Python, FastAPI, Next.js, MongoDB, GCP

- Developed VS Code extension analyzing API/cloud/DB overhead with inline annotations and optimization suggestions
- Achieved <7s initial indexing and **<1s** refresh analysis via file hash caching, outperforming AI IDEs by **47x in speed**
- Implemented hybrid AST + regex parser detecting **2x more cost issues** than AI IDEs while saving **~50k tokens per check**
- Architected RAG-powered web sandbox with **LangChain + FAISS** delivering architectural recommendations in **<2s**

### BrainLattice – PDF to Obsidian Vault Generator | Python, FastAPI, PostgreSQL, Redis, TypeScript, React, AWS, Cloudflare

- Developed React + FastAPI web app and **npm CLI** converting PDFs into Obsidian vaults with NetworkX graph generation
- Reduced token costs by **70-80%** via global concept seeds and batched extraction replacing full-document context caching
- Accelerated processing **70%** via parallelization and embedding deduplication, generating **100+ node graphs** in **<1 minute**
- Built event-driven pipeline using **QStash + Lambda + R2** with **pgvector RAG** for on-demand notes, reaching **300+ installs**

### Tark – Google Earth for Game Devs | TypeScript, Next.js, Python, FastAPI, Redis, Prometheus, Leaflet, SciPy, Numpy, Pytest

- Developed web app turning locations into game-ready 3D meshes in **<15 seconds** (typically weeks of manual modeling)
- Processed Mapbox elevation and satellite imagery to generate terrain meshes at **45K+ triangulated faces per second**
- Extracted **2000+ building footprints** from OpenStreetMap and generated textured .obj files for Unity/Blender workflows
- Productionized with **Redis** job store, reducing latency by **30%** via async workers with **98.2% accuracy** in pytest suites

## AWARDS

- DeltaHacks 2026: **1st Place Overall** | Hack Western 2025: **Best AI Application Built with Cloudflare**
- Hack the North 2025: **Semi-Finalist** (top 32 out of 256 teams, 1000+ hackers)