

Ryan Shabaneh

ryan.shabaneh@mail.utoronto.ca | [+1 647-391-1001](tel:+16473911001) | linkedin.com/in/ryan-shabaneh/ | github.com/ryanshabaneh

EDUCATION

University of Toronto

Expected May 2029

Honours Bachelor of Science (HBS), Computer Science

- Relevant Coursework: Software Design, Data Structures and Analysis, Software Tools and Systems Programming, Linear Algebra II, Theory of Computation, Computer Organization, Probability

EXPERIENCE

CI Financial

Jan 2026 – Apr 2026

Software Engineering Intern

Toronto, ON

- Cut quarterly reporting time by **90%**, consolidating 250+ manually maintained reports into 60 config-driven templates using pandas and python-pptx
- Shipped an internal HR RAG assistant using Qwen3 + Ollama for 100+ employees, cutting policy lookup time by **85%**
- Reduced HR assistant format failures from **28% to 5%** by fine-tuning Qwen3 with LoRA for policy QA

Dalhousie University

Oct 2023 – Jan 2024

Machine Learning Research Assistant

Halifax, NS

- Built the property-comparison dashboard for REALM, an IEEE-published ML real-estate appraisal system, using Django and Chart.js
- Validated a Random Forest model at $R^2 > 0.90$ on a held-out test set by building its training and evaluation pipeline

PROJECTS

Fovea | *TypeScript, WebGPU, WGS, Vite* | [Demo](#) | [GitHub](#)

Jun 2026

- Built Fovea, a browser-based GPT-2 (124M) inference engine with a forward pass written from scratch in WGS, running fully client-side on WebGPU
- Engineered WebGPU kernels for attention-head ablation across all **144** GPT-2 heads, validating GPU outputs against an FP32 reference
- Reduced model memory by **50%** (496 MB → 248 MB) by storing weights in fp16 with fp32 accumulation

Atlas Queue | *Go, PostgreSQL, Redis, gRPC, Docker* | [GitHub](#)

Jan 2026 – Mar 2026

- Built a broker-free distributed job queue in Go and gRPC for running background jobs across workers, using PostgreSQL leases and Redis for coordination
- Cut crashed worker's job recovery from **55s to <20s** at p90 by shortening lease expiry and the reclaim scan interval
- Stress-tested 10,000 jobs through repeated worker crashes, observing **zero lost jobs** or **duplicate completions**

Shipyard | *Next.js, React Three Fiber, TypeScript, Socket.IO, Claude API* | [Devpost](#) | [GitHub](#)

Mar 2026

- Won **3 tracks** at GenAI Genesis for an AI platform that scores GitHub repositories for production-readiness and visualizes results as a live 3D city
- Reduced Claude API calls by **88%** per scan by batching analyses into one request, surfacing initial scores in **<5s**
- Automatically retried failed AI deployment fixes up to **3x** through sandboxed build and runtime checks before pushing verified GitHub commits

TECHNICAL SKILLS

Languages: C++, C, Python, Go, WGS, TypeScript/JavaScript, SQL

Frameworks & Libraries: Django, FastAPI, gRPC, Next.js, pandas, PyTorch, pytest, React, SQLAlchemy

Databases & Data: pgvector, PostgreSQL, Redis, Snowflake, SQLite

Systems & Tools: Docker, Git, Linux, Ollama, WebGPU