# Oliver Kwun-Morfitt

Toronto, ON | 647-323-8175 | oliverkwunmorfitt@gmail.com | LinkedIn | GitHub | olliekm.com

## Education

**University of Toronto – St. George Campus** **Expected graduation: 2028**
*Honours BSc, Computer Science & Statistics, Math (Co-op)* *Toronto, ON*
***(GPA 3.77)*** *(Dean's List Scholar)*

## Experience

**Software Engineer – ML Infrastructure** Sep 2025 – Present
*University of Toronto Machine Intelligence Student Team* *Toronto, ON*

- **Enabled** scalable GPU research workflows by developing **open-source Go CLI** tools that let students submit, queue, monitor, and debug compute jobs on **Tenstorrent** and **AMD** servers.
- **Improved reliability** in multi-user GPU environments by implementing **token-based authentication**, structured logging, and **robust error handling** using Go and Kong-style CLI patterns.

**Full-Stack Developer** May 2025 – Present
*University of Toronto Climbing Club* *Toronto, ON*

- **Enhanced maintainability** and user experience by implementing a clean, **component-driven UI** using **Next.js** and **Tailwind**.
- **Improved responsiveness** and layout consistency by building reusable, **mobile-first** navigation and page-structure components.
- **Increased** codebase stability by conducting regular **PR reviews** that standardized styling and enforced component patterns across the app.

## Projects

**Parsec – LLM Orchestration Toolkit** | *Python, Pydantic, asyncio, OpenAI API, PyPI* Nov 2025

- Published **open-source library** to PyPI for enforcing structured JSON output from **LLMs** with **validation**, **auto-repair**, and streaming capabilities.
- **Designed** extensible **adapter architecture** supporting multiple providers (OpenAI, Anthropic) with async streaming and partial JSON parsing.
- Built **validation engine** with schema-based repair heuristics and **LRU caching**, reducing redundant API calls and **improving reliability**.

**Real-time Ledger API** | *Go, Gorilla/Mux, UUID, Docker, REST, PostgreSQL, Git* Nov 2025

- **Ensured** strict **financial correctness** by building a **production-grade double-entry ledger** that enforces per-currency invariants, validates postings, and prevents negative balances.
- **Improved** operational **reliability** by implementing **token authentication**, structured logging, and health probes using **Go**, **Gorilla/Mux**, and **Docker**.
- **Achieved 3.6k req/s** throughput **(p95 = 14ms)** under full **ACID** guarantees by optimizing transaction handling and enforcing strict posting invariants in **PostgreSQL**.

**E-commerce REST API** | *Go, MySQL, Docker, Redis, Prometheus, Git* Jun 2025 – Jul 2025

- **Increased** system performance from **2.1k to 4.4k** req/s by adding **Redis caching** and building a **concurrent**, connection-reuse–optimized load balancer in Go.
- **Improved** scalability and latency by engineering a modular **REST API** with **JWT** auth, **Dockerized** services, and clean routing.
- **Raised** service availability to **99.6%** by implementing a custom **round-robin load balancer** with health checks, **rate limiting**, and automated failover.

## Technical Skills

**Languages**: Python, Go, Java, Typescript, SQL, Javascript, HTML/CSS
**Frameworks**: React, Node.js, Next.js, gRPC, FastAPI
**Developer Tools**: AWS (S3, SageMaker), GCP, Docker, PostgreSQL, MySQL, MongoDB, Redis, Kafka, Git, CI/CD
**Libraries**: Pandas, NumPy, Matplotlib, Poetry, PyTorch, TailwindCSS, pytest, SQLAlchemy, bcrypt