

Rashid Mahmood

linkedin.com/in/rashid-mahmood-952152311 | github.com/rashidm2006

EDUCATION

University of Toronto

Sep. 2024 – Apr. 2028

Bachelor of Applied Science in Electrical and Computer Engineering (PEY Co-op)

Toronto, ON

- **Relevant Coursework:** Operating Systems, Data Structures & Algorithms, Control Systems (in progress); Computer Organization, Digital Logic, Signals and Systems, Object-Oriented Programming
- **Activities:** Undergraduate Class Representative, Frosh Week Leader

EXPERIENCE

Engineering Intern (Geospatial Machine Learning)

May 2026 – Aug. 2026

LGND AI

Remote

- Shipped 10+ projects over the summer to map what LGND's satellite-imagery embedding platform could and couldn't do: national-scale analyses, few-shot detectors, 4 FastAPI/Leaflet web apps and 2 benchmarks.
- Cut a continental land-cover map's storage by 98% (24.6 GB → 456 MB) with PCA-64 + int8 quantization after showing the loss was below existing label noise; the 6.0M-cell map hit AUC up to 0.957 vs. USDA, NLCD and ESA.
- Caught a silent bug where the embedding API returned integer-quantized vectors, breaking cosine similarity in the browser without any error; normalized them server-side and verified the client math cell-for-cell against a NumPy reference.
- Built a national similarity-search app that recolours all 55,793 U.S. cells per click with zero API calls, by running a resumable ~100K-call ETL and shipping every embedding to the browser as one 14.7 MB payload.

Volunteer Software Engineer (IT Department)

Jun. 2026 – Present

National Non-Profit Organization

Toronto, ON

- Built and launched a Spring Boot + SQL event-management app for a 30,000+ member national organization, now used by 200+ members to propose, approve and schedule events, with role-based access control and automatic email confirmations.

Embedded Software Developer (Combat Robotics)

Sep. 2024 – Jan. 2025

University of Toronto Robotics Association

Toronto, ON

- Wrote C++ firmware for the team's combat robots (sensor integration, motor control) and fixed control-loop failures found in testing to stabilize motor response.

PROJECTS

Real-Time Fractal Visualizer | *Bare-metal C, RISC-V, FPGA* | [Demo](#)

Apr. 2026

- Rendered Mandelbrot and Sierpiński fractals to VGA in real time, in bare-metal C on a Nios V RISC-V soft-core (DE1-SoC).
- Replaced floating point (too slow for real-time frames) with hand-written Q8.24 fixed-point math, plus 4×4 block rendering, zoom-adaptive iteration counts and double-buffered vsync.
- Wrote a PS/2 mouse driver from scratch for click-to-zoom, and added an int8-quantized CNN gesture recognizer for touchless zoom and pan.

FPGA Digital Drawing Engine | *Verilog, ModelSim, Quartus*

Nov. 2025

- Designed a Verilog drawing system on a DE1-SoC: a custom 640×480 VGA controller, a from-scratch PS/2 receiver for cursor tracking, and FSMs that sequence memory reads/writes to save and reload drawings.

OpenStreetMap Mapping & Routing App | *C++, EZGL/GTK* | [Demo](#)

Apr. 2026

- Parsed city-scale OpenStreetMap data into custom C++ data structures, picking containers and indexes so every query (name search, street length, one-way connectivity) met strict complexity limits.
- Implemented Dijkstra and an optimized A* for turn-by-turn routing; the A* optimizations let our 3-person team pass every routing performance test under hard time limits.
- Wrote a multithreaded solver for an NP-hard travelling-courier problem: greedy construction (~5 s), then simulated annealing (~45 s), timed so the whole budget goes to improving routes.

Human Rights Incident Database & Mapping Platform | *Python, OpenCV, Gemini/OpenAI APIs*

May 2026

- Created a searchable, map-based platform documenting persecution incidents since 1947, extracted from 40,000+ scanned newspaper pages (35,000+ in Urdu); every record links to its source page and was hand-checked.
- Layouts varied too much for templates, so wrote an OpenCV pipeline that splits pages into articles via rule-line detection, feeding a recall-tuned multi-model LLM pipeline (Gemini, OpenAI) that finds and translates incidents.
- Capped each LLM context at 2 pages (~2,000 tokens) to curb hallucination, with per-page checkpointing and batch-API fallback so failed runs resume instead of restarting.

TECHNICAL SKILLS

Languages: C, C++, Verilog, Python, Java, SQL

Embedded: Bare-metal C on Nios V RISC-V, DE1-SoC (Intel Cyclone V), PS/2 and VGA drivers, fixed-point math, Arduino

Tools: Git, Linux, Docker, ModelSim, Quartus, NumPy, FastAPI