Dave Grewal

289-771-1502 | davee.grewal@mail.utoronto.ca | linkedin.com/in/dave-grewal | woo-dg.github.io/Portfolio

Summer 2026 availability: 12 weeks (May–Aug)

Education

University of Toronto, St. George — Toronto, ON

B.Sc. in Computer Science — *Sep. 2025 – Apr. 2029*

Experience

Software Engineer Intern, Hypernym AI (Meta Llama startup) — May 2025 – Aug. 2025

Toronto, ON

- Replaced a CLI-based LLM benchmarking script with a React and TypeScript web application so the engineering team could configure, launch, and review experiments through a browser UI.
- Designed a metrics dashboard with charts for compression ratio, semantic similarity, ROUGE, and tokens per second, used in weekly reviews to compare models and datasets side by side.
- Aligned front-end metric calculations with existing Python reference scripts within approximately 2% and added CSV/HTML export to reuse results in reports and follow-up runs.
- Dockerized the application, introduced a one-command development environment, and integrated Playwright end-to-end tests into GitHub Actions as a required check for the core upload → configure → run → export flow.

Projects

qurious.xyz | Next.js, React (TypeScript), Python, CLIP, react-force-graph, Vercel — 2025

- Built Qurious in Next.js (TypeScript) with a Python backend, indexing 1,000+ papers from 5 Ontario universities so students can explore research by theme instead of navigating separate departmental sites.
- Computed CLIP embeddings and cosine similarities in Python and used react-force-graph on the front end to cluster papers into an interactive graph, making related work visible in a single view rather than a flat results list.
- Used lightweight NLP in Python to extract limitations and future work sections and surfaced them with one-click DOI and author email links, shortening the path from identifying an open problem to contacting the researcher.

Hully, World Robot Olympiad biofouling robot | Arduino C++, Python (YOLOv8, OpenCV) — 2023–2024

- Built a magnetically adhered hull-cleaning robot for steel vessels and competed at the World Robot Olympiad finals in Panama, placing **2nd globally** out of **450+** teams and earning a **$1,000** innovation grant.
- Trained a YOLOv8 model in Python on hull imagery and used OpenCV to detect algae regions so the robot could prioritize contaminated surfaces instead of sweeping the entire hull uniformly.
- Implemented Arduino C++ firmware for a four-wheel drive and brush system, including PWM speed control, operating modes, emergency stop handling, and serial telemetry for tuning cleaning passes.

Twinly, AI communication assistant | Electron, React (TypeScript), Node.js, AWS — 2024

- Built Twinly as an Electron desktop application in TypeScript and React that aggregates multiple messaging platforms into a single unified inbox.
- Developed a Node.js service that calls LLM APIs to read conversation history and draft replies in the user's tone, reducing manual typing while preserving context across channels.
- Deployed a Dockerized Node.js backend on AWS (EC2) using **$5,000** in awarded credits to handle message ingestion, storage, and LLM workloads behind a simple API layer.

Technical Skills

Languages: JavaScript, TypeScript, Python, C++, Bash, HTML/CSS
Frontend: React, Next.js, React Query, Tailwind CSS, Vite/Webpack, Electron
Backend and ML: Node.js and Express, REST and WebSocket APIs, CLIP embeddings, OpenCV, YOLOv8
Embedded and hardware: Arduino, Raspberry Pi, basic power electronics, multimeters, wiring harness assembly, through-hole and basic SMT soldering
Tools and cloud: Git, Docker, GitHub Actions, Vercel, Jest, React Testing Library, Playwright, Figma, AWS (EC2 and storage); comfortable on macOS, Windows, and Linux