Maxwell Qin

m26qin@uwaterloo.com | github.com/odozdee | linkedin.com/in/maxwell-qin

Education

University of Waterloo Waterloo, ON

Bachelor of Applied Science, Computer Engineering *Sept. 2026 - Apr. 2031*

- Awarded the President's Scholarship of Distinction

Technical Skills

Languages: Python, C/C++, Java, JavaScript, SQL, HTML/CSS

Frameworks: React, Next.js, Node.js, FastAPI, Flask, ROS 2 Jazzy, Nav2

Infrastructure: Git/GitHub, Linux (Ubuntu/WSL), Supabase, Steel Browser

Experience

Software Engineer Waterloo, ON

UW MedTech Robotics *Sept. 2026 - Present*

- Building autonomous software in Python for a drowning prevention robot using ROS2, RViz, and Nav2
- Developed a LiDAR-based obstacle avoidance system with GNSS/IMU localization, enabling the simulated boat to detect obstacles and navigate around them
- Developed a click-to-navigate feature in Gazebo Sim using the Smac Hybrid-A* Planner, enabling autonomous navigation to user-selected destinations with Nav2 path planning

Firmware Developer Waterloo, ON

University of Waterloo Formula Electric *Oct. 2026 - Present*

- Developed STM32 HIL firmware in C for UW Formula Electric's race car to support simulation-based testing of a Battery Management Unit fan control, regulating accumulator cell temperature
- Used combined timer input capture and polling to measure PWM frequency and duty cycle, with a serial CLI to report measurements and verify fan speed commands

Projects

fiddle | *Python, FastAPI, Next.js, PostgreSQL, Supabase, Librosa, Audiveris*

- Built a full-stack violin practice studio that provides specific practice tips from a music score and performance feedback from user recordings using Anthropic Claude API and Librosa
- Integrated Computer Vision-based Optical Music Recognition with Audiveris to read the score PDF, exporting the details as an MusicXML file
- Used music21 to read the created file and turn it into a structured note data for Claude to generate practice tips
- Built the frontend with Next.js and the backend with Python and FastAPI. Used Supabase for authentication and PostgreSQL for saving practice sessions
- Validated the score analysis and recording comparison on Paganini's Caprice No.24 that successfully identified up to 3 challenging passages per page and suggested videos on intonation and shifting

Syla | *Python, Flask, JavaScript, Claude API, Steel.dev*

- Built an AI-powered study planner for the 2026 Battle of the Schools Hackathon, with the purpose of improving time management and organization for students
- Used Claude API and Steel.dev web agent to extract course information from an uploaded syllabus, live-research course content on the web, and rank assignments based on priority and time
- Developed a Python scraper and integrated Supabase for data storage along with D2L API to access Waterloo LEARN course information
- Built the frontend using HTML, CSS, and JavaScript and the back-end using Python and Flask