Daniel Lee

ds4lee@uwaterloo.ca | https://www.linkedin.com/in/daniel-s-l/ | https://github.com/GitWorkingTime |

Education

University of Waterloo **Sep 2025 - May 2030**
Bachelor of Applied Science, Honours Computer Engineering **Waterloo, ON**

Technical Skills

Languages: C++, HTML, CSS, Javascript, SQL
Developer Tools: Git/GitHub, Linux (WSL), Google Workspace, Onshape
Frameworks: NodeJS, ElectronJS

Student Teams

Lead Programmer **Sep 2022 - Feb 2024**
Bishop O'Byrne FTC Robotics Team | *Java* | *OpenCV* | *Tensorflow* **Calgary, AB**

- Implemented basic odometry through the use of motor encoders and basic PID control, establishing odometry use for subsequent FTC seasons
- Coded April Tag detection and PID control of the robot's IMU to self-correct the robot's heading as proof-of-concept for consistent trajectory following
- Incorporated Tensorflow and OpenCV to detect unique game pieces to score 10 to 20 additional points consistently per autonomous round

Personal Projects

Delirium Detection Monitoring System | *NodeJS* | *SQLite* | *Javascript* | *HTML/CSS*

- Created an HTTP web server using NodeJS to receive sensor data reliably from an ESP32 via an HTTP POST request for patient monitoring
- Designed and implemented the parsing and storing of ESP32 sensor data into a simple SQL database using SQLite for later use with a delirium screening test
- Developed a dashboard using vanilla HTML, CSS, and Javascript that displayed sensor data with timestamps as chronological logs, enabling hospital staff to more easily understand recent patient activity
- Utilized FreeRTOS to code a push button with an ESP32 on the ESP-IDF VSCode extension as a way to test HTTP connectivity and facilitate integration with additional sensors

Basic HTTP Web Server | *C* | *Javascript* | *HTML/CSS*

- Implemented a single-threaded HTTP web server in C to host a custom messaging app, handling metadata for both text messages and images to enable users on a local network to exchange messages
- Designed the user interface using vanilla HTML, CSS, and Javascript for ease of use

HeaderSniffer | *Python*

- Coded a Python CLI tool that verifies the file format given a file path as a precaution against malware attacks
- Utilized magic numbers at the beginning of a file to identify its true format and output a warning when it differs from what appears in file explorer

Snackatron | *C++* | *PlatformIO*

- Built a robot with an ESP32, motors, IR components, and a servo to enable wireless IR control and deploy a treat dispenser tray.