Nathan Daniel

(647) 563-9459 | nathan.m.daniel@queensu.ca | linkedin.com/in/nathan-daniel-nd

nathanddaniel.github.io/My-Personal-Website/

Education

Queen's University | *Candidate for B.A.Sc. Computer Engineering* Sep 2021 – exp. Apr 2026

- Recipient of Dean's Scholar 2024-2025: Combined Fall and Winter GPA of 3.5 or higher
- Recipient of Betty Harrison Awards for Black Canadian Students (2021-2025)
- Relevant coursework: Algorithms, Computer Networks, Data Structures, Database Management Systems, Object Oriented Programming, Data Science, Operating Systems, Software Specifications.

Technical Skills

- **Programming Languages:** C, C++, Java, Python, JavaScript, PHP, SQL, HTML/CSS
- **Frameworks & Libraries:** React, Next.js, Node.js, Express, Spring Boot, CUDA
- **Databases:** MongoDB, PostgreSQL, MySQL, SQLite

Experience

Educational Technology Intern May 2024 – Aug 2024

Queen's University Department of Health Sciences - OPDES

- Worked with **Articulate 360** to create interactive educational modules for 370+ health sciences students, enhancing their learning experience through assessment tools.
- Partnered with faculty, instructional designers, and content experts to ensure educational materials met academic standards and addressed students' learning needs effectively.
- Integrated coding elements to improve learning experience, utilizing **HTML, CSS, and JavaScript**.

Projects

qRate | *CSS, HTML, JavaScript, React, NextJS, NodeJS, Express, MongoDB* March 2025 – Present

- Designed and developed a full-stack web application for Queen's ECE students to rate and review courses
- Built a responsive front-end with **React** and **NextJS** to allow users to view and review courses and professors
- Implemented **RESTful APIs** with **Node.js and Express** to handle form submissions and retrieve course data from **MongoDB**.

32-bit RISC CPU Design | *Verilog, Quartus Prime, ModelSim, FPGA* Jan – April 2025

- Designed and implemented a custom 32-bit RISC CPU using **Verilog**, featuring arithmetic/logical instructions, memory access, branching, and instruction decoding.
- Simulated the datapath and control unit using ModelSim and deployed the full CPU design onto a Cyclone V FPGA using Quartus Prime.
- Built a modular system with register files, ALU, control FSM, instruction decoding, and memory-mapped I/O, using **JTAG UART** for interactive testing.
- Debugged and validated each instruction through waveform analysis and hardware test programs, achieving correct execution of a full testbench program.

Self-Driving Taxi Rover | *Python, OpenCV, Flask, Raspberry Pi, Google Colab* January – Apr 2025

- Engineered an autonomous taxi prototype using a **Raspberry Pi**, integrating GPS, sensors, and camera-based object detection to navigate a simulated urban environment
- Implemented real-time computer vision with **OpenCV** and a custom-trained CNN to detect road signs and pedestrians, enabling intelligent stop, yield, and lane-following behavior.
- Developed a **RESTful API** for vehicle positioning and fare assignment, synchronizing location tracking with server-based fare dispatch and reputation scoring system
- Designed and tested A* pathfinding algorithms to optimize route efficiency while avoiding high-risk and pedestrian-heavy zones, balancing cash flow and safety metrics.