Guhan Iyer

linkedin.com/in/guhansiyer ↗ | github.com/guhansiyer ↗
Email: guhan.iyer@uwaterloo.ca
Phone: +1 (226) 505-7658

Education

University of Waterloo Expected Graduation: **April 2028**
Bachelor of Applied Science in Computer Engineering *Waterloo, Ontario*

- Relevant Coursework: Real-Time Operating Systems, Algorithms & Data Structures, Digital Computers

Experience

Nokia Sept. 2025 – Dec. 2025
Firmware Engineering Intern *Ottawa, Ontario*

- Built **C++** device initialization service for a new optical transceiver ASIC, optimizing boot sequence control across hardware modules.
- Eliminated packet corruption in a critical **C++** message-passing utility with 128-bit atomic operations, **resolving race conditions** and ensuring data integrity.
- Resolved **10+** major defects in the ASIC SDK by analyzing firmware trace logs, reducing daily crash frequency to **zero**.

Ford Motor Company Jan. 2025 – Apr. 2025
Software Development Intern *Waterloo, Ontario*

- Created a modular library in **Python** to simplify and scale testing for an in-vehicle security daemon, reducing reliance on external tooling.
- Refactored a deprecated generator utility to integrate with new **Python** test infrastructure, enabling and enhancing new testing workflows for **100+ engineers**.
- Reworked **30+** legacy tests to utilize the new library, standardizing test structure for **all downstream developers**.

NCR Voyix May 2024 – Aug. 2024
Software Engineering Intern *Waterloo, Ontario*

- Integrated an internal **Python** query utility into a new patch management system, enabling automatic device data retrieval.
- Instrumented a service to validate per-device network compliance data for use **organization-wide**.
- Developed a cross-platform patch verification tool serving **10,000+** devices across **10+** platforms.

Projects

TT-TPU ↗ | *Verilog, Python*

- Designed a simplified tensor processing unit supporting **2-by-2 integer matrix multiplication**; submitted to **Tiny Tapeout** project for manufacturing.
- Implemented an on-chip scratchpad memory in **Verilog** to stage input and weight matrices, enabling continuous streaming to the compute datapath and sustaining ∼**99.8 MOP/s** of throughput.
- Verified designs with **Python/cocotb** and ran static timing analysis with **OpenSTA**, ensuring functional correctness.

wintop ↗ | *C, MSVC, Windows API*

- Developed a CLI-based **thread and process inspector** in **C** for **Windows** platforms, exposing detailed per-thread scheduling and runtime information.
- Enumerated processes and threads with **Win32 APIs**, retrieving priority, state, and timing metadata with low overhead.
- Designed a terminal interface to provide **real-time** diagnostic information, emulating *top* and *ps*.

osh: The Open Shell ↗ | *C, Linux*

- Created a lightweight **Linux** shell in C supporting built-ins and external programs.
- Implemented pipelines and I/O redirection using **Linux** syscalls and **POSIX** file descriptor semantics.
- Added persistent command history using **readline** to improve interactive usability.

Skills

Languages: C, C++, Python, Java, Bash, Assembly (ARM, RISC-V)
Libraries & Tools: Valgrind, GDB, CMake, Make, Android Tools (adb, Fastboot), Docker
Technologies & Protocols: Linux, QNX, FreeRTOS, CAN, TCP/IP, UART, I2C, gRPC, protobuf