*Last updated in Sep 2025*

# Ahmed Shahriar

✉ ahmed.shahriar343@gmail.com ☎ (226) 988-9814 🔗 ahmedshahriar.ca in ahmed-shahriar-343-as

## Education

**University of Waterloo**, Bachelor of Computer Science (BCS) Sep 2021 – Apr 2027

- Relevant Coursework: Distributed Systems, Concurrent/Parallel Systems, Machine Learning, Networking

## Skills

**Programming:** C, C++, Go, Python, Java, JavaScript, Kotlin

**Tools & Frameworks:** Git, Linux, PyTorch, React, gRPC, Redis

**Professional:** Adaptability, Technical & Business Communication, Time Management

## Experience

**Research and Development Engineer** Jun 2023 – Aug 2024
— *University of Guelph* Guelph, ON, Canada

- Prototyped a low-cost IoT telemetry framework, enabling real-time monitoring for thermal research projects.
- Engineered a distributed MQTT-based pipeline in Go with concurrent endpoints, achieving sub-2s end-to-end latency across a distributed deployment (4-8 sensors per site/node).
- Led and mentored a team of 6 engineers, delivering scalable components with peer-reviewed Git workflows.

**Database Analyst** Jan 2024 – Aug 2024
— *Ontario Provincial Police (OPP)* Orillia, ON, Canada

- Integrated a Python + Azure AI pipeline into frontline surveys, cutting processing time by 60% via automation.
- Optimized existing database workflows for forwarding 5000+ responses to analytics more reliably and efficiently.
- Automated sentiment analysis dashboards for Commissioner’s office, delivering actionable insights from trends.

## Projects

**Concurrent File Transfer Protocol (CFTP) — GitHub ↗** Jan 2024 – Feb 2024

- Created a layer-5 file transfer protocol using Go, supporting concurrent uploads/downloads over TCP/IP and UDP.
- Modularized core operations atop Go’s `net.Conn` interface for seamless, protocol-agnostic transport integrations.
- Leveraged goroutines for parallel client–server nodes, sustaining multi-file transfers with minimal overhead.
- Benchmarked performance under 4 concurrent clients x 10 transfers each to evaluate throughput and reliability.

**WLP4 Compiler — GitHub ↗** Feb 2023 – Apr 2023

- Built a full WLP4 compiler in C++, implementing custom NFA, CFG, and AST modules from first principles.
- Generated MIPS assembly through an end-to-end toolchain (lexer → parser → generator → linker → assembler).
- Devised optimization engine reducing assembly output by 38%, from 144K down to 89K on benchmark suite.

**Predictive Text Suggestions — GitHub ↗** Apr 2023 – Apr 2023

- Developed a C++ predictive text engine using compressed tries to suggest word auto-completions to users.
- Incorporated frequency weighting to prioritize commonly typed words, enhancing user experience over time.
- Achieved constant time predictions per keystroke, averaging under 50ms for real-time auto-completion.

**CFA Tools — VS Code ↗ | GitHub ↗** Sep 2025 – Present

- Maintain my published VS Code extension providing the **only** available C∀ ↗ programming language tooling.
- Develop a real-time syntax engine to improve code readability and enable future AI + IntelliSense integration.
- Use this tool daily in CS 343 (Parallel Programming) to build concurrent programs with highlighted C∀ primitives.