

Mohamad Salman

Canadian Citizen | Toronto, ON | (437)-602-0684 | mohamad.salman@mail.utoronto.ca
linkedin.com/in/msalman06 | github.com/mohamadmsalman82

EDUCATION

University of Toronto

Bachelor of Applied Science in Computer Engineering

Sept 2024 – Apr 2029
Toronto, Canada

- Relevant Coursework: Algorithms & Data Structures, Software Design (C++, Git), Computer Organization, Digital Systems

TECHNICAL SKILLS

Languages: Python, C/C++, Java, TypeScript/JavaScript, SQL, Verilog

Backend & Infrastructure: Node.js, Express, FastAPI, PostgreSQL, Redis, MySQL, REST APIs, JWT Auth

Tools & Platforms: Git, Linux, Chrome Extensions API, Cloudflare Workers, CUDA, GPU Computing

Professional skills: Collaborative, detail-oriented problem solver with strong ownership in fast-paced environments

PROFESSIONAL EXPERIENCE

Software Developer – Backend Systems

Nodalli (University of Toronto Engineering Consulting Association)

Oct 2025 – Present
Toronto, ON

- Designed a **unified action adapter layer** that reduced NLP-to-execution integration time by **70%**, routing structured commands to 4 platform APIs via a single async entry point using **Python and SQLAlchemy**.
- Consolidated **email infrastructure and authentication management** to eliminate code duplication, utilizing **Redis-backed OAuth token storage** and automated credential refresh workflows.
- Secured **database operations** by implementing **field-level access control and whitelist validation**, preventing unauthorized cross-user data access via the **SQLAlchemy Object-Relational Mapper**.
- Standardized **cross-team development** by establishing a **JSON request/response contract with unified error handling**, enabling independent parallel development across backend and frontend teams.

Machine Learning Researcher & Developer

Bigger Bird (University of Toronto Machine Intelligence Student Team)

Jan 2026 – Present
Toronto, ON

- Developing **sparse attention patch for BART transformer** to reduce sequence processing from **O(n²) to O(n)** complexity, enabling longer context windows while maintaining model performance on summarization tasks.
- Implemented **custom GPU kernels in CUDA** achieving **2.8x faster attention computation** compared to standard PyTorch implementation, improving memory bandwidth utilization for training on sequences exceeding baseline implementation limits.

TECHNICAL PROJECTS

Anchor | Focus Tracking Chrome Extension [GitHub](#)

TypeScript, Node.js, PostgreSQL, Next.js

- Developed a **Chrome extension in TypeScript** that tracks active tab URLs and system idle time quantifying productive versus distracted browsing behavior across user sessions.
- Built a **REST API using Node.js/Express and PostgreSQL** to store session telemetry, and developed a **Next.js dashboard with Recharts** to visualize daily focus patterns and 30-day productivity trends across **100+ sessions**.
- Implemented **background service workers using the Chrome Extensions Manifest V3 API** to capture tab events and idle state changes, processing events in **under 5 ms** while consuming **less than 10 MB of memory**.
- Integrated the **Claude AI API** to generate personalized productivity insights from focus data; adopted by **25+ students** in beta with an **85% user satisfaction score** based on weekly feedback surveys.

NexFlow AI | Serverless Automation Platform [Demo](#)

TypeScript, Cloudflare Workers, Edge Computing

- Built **workflow orchestration engine on Cloudflare Workers** using **Durable Objects for persistent state management** and distributed cron scheduling across edge infrastructure.
- Integrated **Cloudflare Workers AI (LLaMA)** to translate natural language workflow descriptions into executable automation steps, providing AI-generated configuration suggestions to reduce manual setup complexity.

Accelerated Offside Simulation Engine [GitHub](#)

C++, CUDA, GPU Computing

- Implemented parallel GPU kernel to process **5 million independent geometric calculations**, achieving **100x speedup in computation time (668ms → 6.6ms)** on RTX 4060 through SIMT execution model.
- Optimized memory access with **16-byte aligned data structures for coalesced global memory transactions**, validated correctness through bit-exact comparison of CPU and GPU outputs across all test scenarios.