

# Mohamad Salman

Canadian Citizen | Toronto, ON | (437)-602-0684 | mohamad.salman@mail.utoronto.ca | [LinkedIn](#) | [GitHub](#)

## EDUCATION

### University of Toronto

Sept 2024 – Apr 2029 (Expected)

*Bachelor of Applied Science – Computer Engineering (PEY Co-op)*

*Toronto, Canada*

- **Relevant Courses:** Data Structures & Algorithms, Object-Oriented Programming, Digital Systems (Verilog, FPGAs), Computer Organization (Assembly), Software Design (C++, Git), Linear Algebra
- Intended Minors: **Artificial Intelligence Engineering**, Engineering Business

## TECHNICAL SKILLS

**Languages:** C/C++, Python, Ruby, CUDA, TypeScript, SQL, Verilog, Assembly

**Stack:** Node.js, Express, FastAPI, PostgreSQL, Redis, Git, Linux

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

## PROFESSIONAL EXPERIENCE

### Software Developer, Nodalli

Oct 2025 – Present

*UofT Engineering Student Consulting Association*

*Toronto, ON*

- Deployed 3 **TypeScript REST API endpoints** automating CRM workflows for **100+ active campaigns**, reducing manual coordination time by 40% through optimized data processing and real-time tracking.
- Implemented **Redis-based request de-duplication layer** preventing duplicate email sends, eliminating 50+ weekly duplicate communications through distributed caching architecture.
- Integrated **Gmail and Google Calendar APIs with OAuth 2.0** via Node.js task queue processing **100+ daily requests** with retry logic, designed **normalized PostgreSQL schema** with indexed fields optimizing query performance by 60% on high-traffic endpoints, scaling to **50,000+ client interactions**.

### Project Manager – Manufacturing Optimization

Jan 2025 – Apr 2025

*UofT Faculty of Applied Science & Engineering*

*Toronto, ON*

- Coordinated team of 5 engineering students developing **discrete-event simulation using FlexSim and Python** to identify bottlenecks in manufacturing production line, drove configuration changes yielding **26% throughput increase** and **\$211,000 projected annual profit**, ranking in **top 10% of 187 teams**.

## PROJECTS

### Anchor – Productivity Analytics Platform *Demo*

[GitHub](#)

- Built 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 — Guided DIY Automation Platform *Try here*

[GitHub](#)

- Built a no-code automation platform in **Typescript** on **Cloudflare Workers** enabling users to design and schedule workflows executed on edge infrastructure, reducing manual setup and improving automation reliability.
- Implemented workflow state management using **Cloudflare Durable Objects and Workflows API** for deterministic cron execution, handling **100+ scheduled automations** across edge nodes.
- Integrated **Cloudflare Workers AI** to generate step-by-step workflow instructions based on user actions, using **LLaMA model** to suggest automation logic from UI context.

### Offside Rule Simulator – High-Performance Parallel Computing (C++ & CUDA)

[GitHub](#)

- Built a **GPU-accelerated soccer simulation** that processes 5 million position calculations **100 times** faster than CPU-only processing (668ms to 6.6ms), using NVIDIA's parallel computing platform on an RTX 4060 graphics card.
- Optimized data transfer efficiency by organizing memory in 16-byte blocks, reducing processing delays by 85% while ensuring identical accuracy between standard CPU and GPU versions across all rule variations.
- Delivered **50% overall performance improvement** in real-world conditions after accounting for data transfer overhead between system components, demonstrating practical optimization skills beyond theoretical benchmarks.