

Lucas Rocchetti

Toronto, ON | lucasrocchetti@outlook.com | [linkedin.com/in/lucasrocchetti](https://www.linkedin.com/in/lucasrocchetti) | github.com/rocchettiucas | lucasrocchetti.com

EDUCATION

University of Toronto

Toronto, ON

Honours Bachelor of Computer Science & Information Technology | GPA: 3.78/4.00 | Dean's List

Expected June 2026

- **Relevant Coursework:** Data Structures, Algorithms, Operating Systems, Machine Learning, Artificial Intelligence

EXPERIENCE

WinLane.GG

Apr 2026 – Present

Co-Founder & Lead Engineer | React, TypeScript, Python, FastAPI, PostgreSQL, Riot Games API

Mississauga, ON

- Grew a live League of Legends analytics platform to **500+ monthly visits** by solely designing and shipping a versioned REST API, normalized PostgreSQL schema, and React 18 SPA across a full cloud deployment stack.
- Reduced API response times to **under 5 seconds** across all endpoints by engineering an automated daily data pipeline that pre-computes and stores thousands of aggregated matchup records into PostgreSQL, eliminating live query overhead entirely.
- Increased recommendation accuracy and data availability by building a multi-signal ranking algorithm with a dynamic confidence model that automatically recalibrates result thresholds on each new data release, requiring zero manual intervention.
- Deployed and maintained a production-grade platform as sole engineer by owning the full system architecture across backend, frontend, database, and cloud infrastructure with CI/CD pipelines via Vercel and Render, implementing IP-based rate limiting at **240 req/min** to ensure reliable uptime under real-world traffic.

PathwayR

Jan 2026 – Apr 2026

Software Engineer | React, Node.js, Supabase, Vercel

Toronto, ON

- Expanded platform authentication to support 3 distinct user roles — students, mentors, and professors — across **4 universities** and **1,500+ indexed professors** by implementing a role-based auth system with institutional email verification enforced at registration.
- Reduced unauthorized account access by engineering a manual approval pipeline for mentor and professor accounts, requiring admin verification before granting platform access, strengthening data integrity across **500+ tracked awards**.

Google Developer Groups — UTM

Sept 2024 – Apr 2025

Software Engineer Member

Mississauga, ON

- Placed **Top 3 at DeerHacks** hackathon, leading a team to build and ship a real-time two-player web game in React with Nash Equilibrium game theory mechanics, presented to a panel of judges under hackathon constraints.
- Attended technical workshops on React, Redux, and NLP and industry panels featuring engineers from Amazon, Shopify, AMD, and Microsoft within a Google-backed developer community.

PROJECTS

RocSpace | Tauri 2, Rust, React, TypeScript, Whisper.cpp, Tokio

May 2026 – Present

- Eliminated context-switching across AI coding agents by building an open-source multi-agent developer workspace in Tauri 2 + Rust that unifies Claude Code, Codex, OpenCode, and standard shells into a single cross-platform desktop application with isolated terminal sessions per agent.
- Architected a two-runtime concurrency model in Rust to guarantee a non-blocking UI across **12 concurrent terminal sessions**, separating blocking I/O operations from async task dispatch using Tokio, ensuring interface responsiveness scales independently of active agent and terminal count.
- Shipped fully offline AI voice dictation with no cloud dependency by engineering an on-device speech recognition pipeline in Rust using Whisper.cpp, with native audio capture, automatic resampling, and crash-safe model downloads, delivering transcription entirely on-device with no data leaving the machine.

NHL Player Dashboard | React, TypeScript, Recharts, NHL API, Vercel

Apr 2026

- Delivered instant player navigation across a **23,000+ player** NHL analytics platform by engineering a session caching layer that stores data on first load, cutting all repeat lookups to zero network requests regardless of how many players are visited in a session.
- Eliminated stale and out-of-order search results by implementing a live search engine with **300ms debouncing** and automatic in-flight request cancellation, ensuring only the most recent query resolves and redundant API calls never reach the server.
- Engineered automated API failure recovery across playoff and pre-season periods by building a date-offset retry system that detects empty responses and walks back through historical snapshots until valid data is found, with a mathematically projected playoff bracket as a live fallback built from current standings using official NHL seeding rules.

TECHNICAL SKILLS

Languages: Python, TypeScript, JavaScript, Rust, Java, C++, C#, SQL, HTML, CSS

Frameworks: React, Node.js, FastAPI, pandas, NumPy, scikit-learn, PyTorch, Tailwind CSS

Developer Tools: PostgreSQL, Supabase, Docker, Git, GitHub, Bash, pytest, CI/CD

Cloud & Systems: AWS, Vercel, Render | Linux, Unix, macOS, Windows