Jake Edmonstone

jbedmons@uwaterloo.ca | linkedin.com/in/jake-edmonstone | github.com/jake-edmonstone

Education

University of Waterloo | *GPA: 88.00* — *2022 – Expected 2027*
Bachelor of Mathematics | *Joint Computer Science and Pure Mathematics* — *Waterloo, ON*

Experience

Software Engineer Co-op — *May 2025 – August 2025*
Constant Contact Inc. — *Kitchener, ON (Hybrid)*

- Developed and maintained customer-facing email delivery features using Next.js (frontend) and Java (backend).
- Collaborated on CI/CD pipelines with Jenkins and contributed to automated testing and code quality improvements.
- Participated in agile ceremonies, code reviews, and issue tracking using Git and Jira within a cross-functional team.

Full-stack Developer Co-op — *September 2024 – December 2024*
BCS Automation Ltd. — *Belleville, ON (On-site)*

- Developed and executed KQL queries in Azure Data Explorer to manage and analyze sensor data, establishing a robust data storage and querying solution where none previously existed.
- Designed and trained LSTM autoencoder models using PyTorch for anomaly detection, advancing research into identifying irregularities in time-series data.
- Designed and built a front-end interface for data visualization using Plotly.js, featuring interactive zoom, a data labeling tool, and a menu system, improving data exploration capabilities and usability.

Full-stack Developer Co-op — *January 2024 – April 2024*
BCS Automation Ltd. — *Belleville, ON (On-site)*

- Addressed query slowness by migrating to a NoSQL database (CosmosDB), implementing indexing and partitioning strategies to efficiently handle large-scale data, resulting in improved throughput.
- Implemented authentication using Microsoft Active Directory and C#, enhancing application security where none previously existed.
- Refactored the back-end API to adhere to best practices, ensuring proper use of HTTP methods (e.g. idempotent GET requests) and improving overall code maintainability.

Projects

WLP4 Compiler — *C++, Make*

- Designed and implemented a WLP4 compiler in C++ that converts high-level WLP4 code into MIPS assembly, using a modular architecture with components for scanning, parsing, type checking, and code generation.
- Applied theoretical computer science concepts, including regular expressions and maximal munch scanning for tokenization, LR(1) parsing for syntax tree construction, and context-sensitive analysis for semantic validation.

Chess — *C++, X11, Make*

- Implemented a fully functional chess game with four levels of computer difficulty and a graphical user interface for visual interaction.
- Highlighted proficiency in object-oriented design principles, including inheritance and polymorphism, to support large-scale modular development and code reusability.

Personal Website — *HTML, CSS, JavaScript*

- Created a portfolio website featuring my projects, including images/GIFs and repository links.

Extra Curriculars

- Algebra Tutor at the University of Waterloo Mathematics Tutoring Center
- Volunteer tour guide for the Mathematics Society

Skills

Languages: C++, C#, Python, SQL, KQL, JavaScript/TypeScript, HTML/CSS
Frameworks: React, ASP.NET
Developer Tools: Git, Microsoft Azure, VS Code, Visual Studio, Vim/Neovim
Libraries: PyTorch, Pandas, NumPy, Plotly.js