

ALEXANDER KING PEROCHO

647-701-8779 | perocho.xela@gmail.com | [LinkedIn](#) | [GitHub](#) | [Website](#)

EDUCATION

Western University.

Sep 2020 — Expected Apr 2026

B.Sc Computer Science — Software Engineering.

London, Ontario

Coursework: [Data Structures & Algorithms](#), [Object-Oriented Design & Analysis](#), [Databases I](#), [Software Project Management](#), [Intro to Software Engineering](#), [Software Tools & Systems Programming](#), [Operating Systems](#), [Computer Architecture & Organization](#), [IBM Z Xplore \(Master the Mainframe\)](#).

SKILLS

Languages: C++, Java, Python, Bash, SQL, NoSQL, JSON, JavaScript, PHP, DOM, HTML, CSS.

Tools & Technologies: Git, GCC, Make CMake, Vim, Linux, Unix, Relational Databases, UML, ERD, Jira, Confluence, [AWS](#).

Concepts & Frameworks: OOP, SOLID, ACID, Agile, Scrum, Waterfall, SDLC, MVC.

EXPERIENCE

Front End Developer Intern

May 2020 — Aug 2020

Long Branch Neighborhood Association

Toronto, Ontario

- Designed and implemented user-friendly navigation pages with an interactive map in Flutter/Dart; enhanced trail discovery and improved user engagement in a mobile app used for local nature tours by **80%** for a community with over **10,000+** residents.
- Collaborated with an Agile team of 4 to deliver UI/UX updates in a structured git workflow; ensured clean commit histories and reduced merge conflicts by **50%**, improving code quality and commit tractability.
- Conducted functional and cross-platform testing of UI components across **15+** device configurations to ensure responsive behavior; resolved over **20+** visual bugs and layout issues on both Android and iOS devices.
- **Utilized: Flutter, Dart, Figma, Github, Git.**

PROJECTS

Chess Engine

- Created a UCI-compliant chess engine in modern C++ using Object Oriented Programming; emphasized adherence to SOLID design principles; applied Singleton, Strategy and State design patterns; wrote clean C++ built to last through industry standards.
- Designed a polymorphic 64-bit bitboard system with custom bitwise operators; enabled constant-time move generation and caching of evaluations; improved search, position-evaluation, and memory efficiency.
- Applied a heuristic evaluation function with compact data structures and inline functions; reduced position analysis compute times.
- **Utilized: C++, SOLID, OOP, Design Patterns, Polymorphism, CMake, Git.**

Custom Memory Allocator

- Built a simple dynamic memory allocator using a first-fit allocation policy; gained an understanding of memory management.
- Managed memory allocation with constant-time bitmap updates and linear-time search for free space; optimized space utilization and ensured constant-time allocation/deallocation.
- **Utilized: C.**

Backup & Cleanup Automation Script

- Created a Bash script to automate daily backups of specified directories into timestamped .tar.gz archives; eliminated manual backup risks, ensured historical data tracking with timestamped archives, and provided a simple disaster recovery option.
- Implemented automated log rotation and old file cleanup based on retention policies using find, xargs, and rm; prevented disk space overuse and simplified long-term system maintenance.
- Scheduled the script using cron for hands-off daily execution and maintained logs for audit tracking; ensured accountability, traceability, and simplified troubleshooting.
- **Utilized: Bash, Unix Tools.**

Hospital RDBMS

- Designed a normalized relational database in MySQL to model relationships between patients, doctors, nurses, and departments.
- Developed form-driven web pages using HTML and CSS; enabled user-friendly management of database through a UI.
- Created server-side logic with PHP; enabled CRUD operations with prepared SQL statements and input validation.
- Applied RDBMS best practices, including referential integrity, ACID-compliant foreign-key constraints, and index/query optimizations; improved performance and ensured data consistency.
- **Utilized: MySQL, HTML, CSS, PHP, ACID, Apache Web Server, Git.**