J.R. Maingat

jrmaingat.ca / me@jrmaingat.ca / 647-859-9948

Education

University of Toronto

Honours Bachelor of Science in Computer Science, Co-op — *Degree Expected: May 2021*

- Awarded U of T Scholars Program Award and Renewable Entrance Scholarship
- Cumulative GPA: 3.5/4.0

Skills

Languages: Go, JavaScript, C++, Python, Java, C, C#, TypeScript, HTML, CSS
Libraries & Frameworks: Node.js, React, Angular, Vue.js, Express
Tools & Technologies: Docker, SQL, NoSQL, Elasticsearch, Git, Linux, Android, Bash, Vim

Experience

More at **linkedin.com/jr-maingat**

Cisco Meraki — *San Francisco, CA*

Software Engineer Intern (Expected) — *September 2020 – December 2020*

RBC Royal Bank — *Toronto, ON*

Software Developer Co-op — *January 2019 – August 2019*

- Developed metrics aggregation API with Node.js and Express for graph-based search engine built on Elasticsearch, saving analysts hours per week by automating report generation
- Designed and implemented asynchronous algorithm for graph traversal and document retrieval in Express API, reducing space complexity from *O*(n) to *O*(1)
- Engineered historical data pipeline for Elasticsearch using C#, enabling trend analytics and monitoring
- Upgraded front-end framework from AngularJS to Angular 7, decreasing page load times by 10%
- **Technologies used:** JavaScript, .NET C#, Node.js, Express, Elasticsearch, Angular, TypeScript, RxJS

Ontario Treasury Board Secretariat — *Toronto, ON*

Software Developer Co-op — *January 2018 – April 2018*

- Developed Go API for government-wide social network, decreasing page load times by up to 98%
- Implemented backend error logging for Go API, saving hours of debugging time per week
- Reduced build times by 50% by minifying Docker images, leading to a faster Agile/DevOps workflow
- Scripted static code analysis in continuous integration with Bash, identifying 20+ code quality issues
- **Technologies used:** Go, Docker, Bash, Vue.js, JavaScript, MySQL, Jira, Confluence, BitBucket Pipelines

Projects

More at **github.com/tagniam**

Turn: Hacktober 'n' Slash →

- Led development of open source C++ game with a Hacktoberfest community of 40+ contributors
- Reviewed 70+ pull requests for new features and bug fixes, prioritizing clean object-oriented code
- **Technologies used:** C++, Linux, TravisCI, CMake

Vim for Visual Studio Code →

- Ported 5 features to open source Vim emulator in TypeScript with 1.7+ million users, resolving 10 issues
- Developed and tested cross-platform shell execution Vim command, improving developer productivity
- **Technologies used:** TypeScript, Node.js, Mocha, Gulp

KVRaft →

- Built distributed key-value store in Go, resilient to up to 50% cluster failure and network partitions
- Implemented scalable and containerized Raft-based distributed consensus using Docker and RPCs
- **Technologies used:** Go, Docker, Raft