Ryan Fleck

ryanfleck.github.io

ryan.fleck@protonmail.com
+1 (613) 501-4043

Education

- **University of Ottawa** Sep. 2015 – Apr. 2021
  Bachelor of Applied Science in Computer Engineering *Ottawa, ON*

Experience

- **MNP LLP** Jan. 2018 - Dec. 2018
  Co-Op Developer *Kanata, ON*
  - **SiteCore – C#, Razor, JS, XML, MongoDB, SQL:** SiteCore is an Enterprise WCMS. Involved in quality assurance, testing, debugging, extending the backend and building components according to client requirements.
  - **Adobe Experience Manager – JSP, Java, JS, HTL:** Assisted software architects with development of an AEM prototype for a client bid. Responsible for security, user & group management, workflow and 2 components.
  - **TeamSite – XML, XSL, Java, JS, Spring, Bootstrap, CentOS:** Maintained, debugged and extended an OpenText Teamsite system, a Java-based Spring Boot CMS. Utilized CI and CD in an agile team of five.
  - **Drupal – PHP, Twig, JS, MariaDB, Azure:** Investigated, developed and deployed a Drupal CMS to Azure in a team of three. Responsible for CentOS container prototype, components, search page.
  - **Technical Writing – LaTeX, Python:** Produced clean, high-quality documentation for non-technical and system administrator audiences. Responsible for writing release notes, instruction manuals and tutorials.
  - **Quality Assurance – Microsoft TFS, Jira, Python:** Designed and executed multiple successful Quality Assurance Protocols, utilized automated testing, hundreds of bugs captured, tracked and resolved.
- **Xalgorithms Foundation** Apr. 2018 - Present
  Contributing Developer & Technical Writer *Ottawa, ON*
  - **Interpreter Development – Scala, Ruby, Docker:** The IoR Interpreter parses rules in .xalgo format and runs them against an internal collection of tables and provided sample data. Coded some methods and unit tests.
  - **IoR Software Architecture:** Involved in discussions regarding the implementation of the IoR (Internet of Rules) and .xalgo language. For more information, please visit xalgorithms.org.
  - **Quality Assurance:** Responsible for debugging the Scala-based rule interpreter, and ensuring the .xalgo language is designed in an intuitive manner. Bugs and issues are posted and resolved in the Xalgo Github repositories.
- References for the organizations above are available on request.

Personal Projects

- **SuperPull – NodeJS:** Crawls the filesystem for git directories and stores them in a config file. Executing *superpull* brings all configured repositories and remote branches up to date. I use the tool daily. Currently rewriting in **Go**.
- **Games – Three.JS, Pixi.JS, LibGDX, Unreal, Socket.IO:** Numerous 1-4 player titles, usually written and tested for use with four XBox 360 controllers, using Express and Socket.IO for game networking.
- **MorningData – C++, NodeJS:** A seven-segment display sits by my bed and displays the weather, and the arrival time of the next bus at a fixed stop. System composed of an Arduino reading serial input and NodeJS piping formatted OC-Transpo and OpenWeatherMap API Data to a serial port. Next phase: wireless module to connect pi and arduino.
- **NodeJS REST APIs – PostgreSQL, Mongo, NodeJS, Express:** Built multiple NodeJS APIs, using clever async database storage & retrieval methods, for projects and internship applications. Most recent: S19 Shopify Dev Challenge
- **RML: Ryan's Micro LISP – C:** Tiny not-quite-ANSI lisp repl. Tokenization and parsing experiment.
- Projects are unit tested with **Travis CI**, often in a **Heroku** container, and available for code review on Github.

Group Projects

- **CanUX Conference Booth Trivia: NodeJS, Socket.IO, WebPack, Heroku, SASS:** Chat & Trivia app that started as a personal project. After being noticed, the project was given resources and allocated time at MNP LLP.
- **BusBot: NodeJS, MongoDB:** Developed with M.Orchard. Developed a Slack Bot that pulls & processes data from the OC-Transpo API and formats it for display in Slack. Caches bus data in MongoDB. Hosted in Heroku.
- **LightShow: Blockchain Christmas Tree – NodeJS, Express, Solidity, C++:** Hackathon team of three. Utilized a simulated Ethereum blockchain to record transactions from a simple web app. Judges could use the app to change the color of the tree. Wrote the Node-to-Arduino serial interface for the dApp and the Arduino serial interpreter.