





NIZAR JANDAL ALRIFAI

 [saber74.github.io](https://github.com/saber74)  njandala@uwaterloo.ca  linkedin.com/in/nizar-alrifai/  github.com/saber74

Education

University of Waterloo

Sep. 2019 – April 2024 (expected)

Bachelor of Applied Sciences in Computer Engineering

Waterloo, ON

- Awarded Dean's Honor list twice with an average of **96.5%** in 2nd year (Awarded to top 10/160 students)

Technical Skills

Languages: Python, TypeScript, Java, Groovy, C, C++, JavaScript, SQL, R

Web-Dev: Node.js, React.js, Knex.js, HTML5, CSS3, Django, Koa.js, REST API, Bootstrap

Other: Git, Jenkins, Docker, Redis, PostgreSQL, Jest.js, AWS, CI/CD, Maven, Azure, Gradle, Linux, Unix, Jfrog, PCF, Pytest

Experience

Wealthsimple

August 2021 – December 2021

Software Engineer Co-op

Toronto, ON

- Used **TypeScript, Redis, AWS, Datadog, and Periscope** to reduce latency across the codebase and ensure backend systems that support trading are robust, and can scale to massive user traffic and order volume
- Increased security by 100%** and reduced the risk of harmful transactions by designing and developing an internal data transformation tool used across various repositories, utilizing **PostgreSQL, and Typescript**
- Designed and implemented the backend, schemas and API of critical trading features leveraging **Koa.js, Knex.js, and Jest.js**
- Reduced **type related errors by 90%** in the trade repository, eliminating the chances of type-related critical incidents, through planning and executing a major codebase redesign
- Spearheaded stress and regressional crypto and trade load testing using custom automated **k6 scripts**

Manulife Financial

January 2021 – April 2021

DevOps Engineer Co-op

Waterloo, ON

- Provided continuous development and functional testing on **40+** Jenkins Pipelines used by **500+** employees using **Groovy**
- Used **Node.js** and **React.js** to create various internal web-based solutions to facilitate other teams in the organization
- Automated various tasks leveraging REST APIs resulting in **100x faster** setup/cleanup time to teach employees new skills
- Developed utilities and scripts utilizing **Python** and **Shell** scripts to assist in multiple major migrations of tools
- Created new pipelines using **CI/CD** principles, integrating code quality tools like **SonarQube**, and pushing Artifactory to **Jfrog**

Sony Interactive Entertainment (PlayStation)

May 2020 – August 2020

Software Engineer Co-op

Waterloo, ON

- Worked on the PlayStation 4 Console Store using **Python** to migrate to new API dependencies to unify logic across platforms
- Supported world-wide releases of the **\$20+ billion** yearly console store through integration and regression development and testing
- Optimized regression test suites to be **19.83% faster**, through code and logic recreation and optimization
- Designed and developed automated tests with maximum requirement coverage through **Pytest**
- Maintained multiple environments using **Docker**, and performed code execution through custom-made **Jenkins** pipelines

Canada South Science City

July 2019 – September 2019

Full Stack Software Developer

Windsor, ON

- Built the organization's main website from scratch boosting, the website's monthly visitors by **317.6%**

Projects

BusyMaps

Hack the 6ix

- Created a real-time platform to visually represent busyness of areas using data from **Geotab** and **Google Places API**
- Relayed user input to a **Flask** based back-end using custom made API endpoints and **Microsoft Azure**

Mini-Games Android Mania

Personal Project

- Developed a series of entertaining multiplayer visual **Java** games using **LibGDX**
- Created an app to utilize phones as controllers, and established a connection through sockets and a remote server

SeizeControl

Hack the North

- Utilized **OpenCV, Firebase, Pytube, HTML** and **CSS** to create a Chrome extension to filter seizure-inducing YouTube videos
- Processed and analyzed frames using self-designed mathematical algorithms in **Python**, and modeled the data with **Javascript**