

# JORDAN DUBRICK

226-339-5539 | [dubrickjordan@gmail.com](mailto:dubrickjordan@gmail.com) | [linkedin.com/in/jordandubrick](https://www.linkedin.com/in/jordandubrick) | [github.com/Jdubrick](https://github.com/Jdubrick)

## TECHNICAL SKILLS

---

**Languages:** Python, JavaScript, TypeScript, C, Java, PHP, SQL, Groovy

**Frameworks & Technologies:** ReactJS, Next.js, Node.js, Express.js, Docker, Jenkins, Git, Chef, AWS, SpringBoot

## EDUCATION

---

### Wilfrid Laurier University

*Honours Bachelor of Science in Computer Science*

Waterloo, ON

GPA: 3.9/4.0 | 2020 - 2023

### Humber Institute of Technology and Advanced Learning

*Honours Bachelor of Commerce in Finance*

Etobicoke, ON

GPA: 3.7/4.0 | Graduate 2020

- CFA Institute Scholarship (2020), Finance Leadership Award (2020)

## PROFESSIONAL EXPERIENCE

---

### Software Engineer Intern

May 2022 - August 2022

*Broadridge Financial Solutions*

Toronto, ON

- Developed a web application using **React**, **Node.js**, **Express.js** and **MySQL** to migrate internal tools to **AWS**.
- Employed the use of **TypeScript** to ensure type safety and scalability for the application.
- Designed **REST API** using **Express.js** to handle API requests from frontend and query **MySQL** database.
- Application was containerized with **Docker** and deployed to an **ECS Cluster** utilizing **AWS Fargate** technology.
- Designed and launched infrastructure on **AWS** to host new internal tool. Infrastructure was created with **Terraform**.
- Launched **EC2** instances acting as **Jenkins** agents for **CI/CD** pipelines for new application.
- Built **Chef** cookbooks for the DevOps team that are used to configure **EC2** instances.
- Migrated tool is used by development teams daily and modern design provides more efficient use.
- Was responsible for deploying the first application to **AWS** for the business unit.

### Instructional Assistant

September 2021 - December 2022

*Wilfrid Laurier University*

Waterloo, ON

- Supervised instructional lab sessions for CP104 - Intro to Programming and CP164 - Data Structures I.
- Assisted students with Python loops, arrays, strings, files, and classes.
- Explained key data structures such as Stacks, Queues, Trees, Graphs and Linked-Lists.

### Software Engineer Intern

May 2021 - August 2021

*Broadridge Financial Solutions*

Toronto, ON

- Developed a web application to facilitate release management for the product team.
- Frontend created with **React** and **JavaScript**, the backend **REST API** was created using **SpringBoot** and **Java**.
- **SpringBoot** backend utilized **SQLite** to store and query data required to populate the web application.
- The application increased productivity of the product management team during yearly release process.

## PROJECTS

---

### Stock Information Tool | *ReactJS, Express.js, JavaScript, CSS/HTML*

- Created with **React** frontend and **Express.js** backend.
- **Express.js** requests data from an external API from Twelvedata and returns formatted information to frontend.
- Containerized with **Docker**.
- User enters ticker symbols into application and it will grab the open price, close price, and volume for the last 30 days.
- Data populates an easy to view table and can be exported to .csv file for financial modelling.

### Sorting Visualizer | *ReactJS, JavaScript, CSS/HTML* | *Jdubrick.github.io/sorting-visualizer*

- Sorts an integer array using any of the Bubble Sort, Insertion Sort, Radix Sort, or Merge Sort algorithms.
- Created using **React** and deployed to **GitHub** pages.
- User has the ability to choose the type of sort that will be used for any randomly generated integer array.

### Portfolio Optimization | *Python, Pandas, NumPy, PyPortfolioOpt*

- Optimizes a portfolio in order to obtain the highest Sharpe ratio within a given volatility range that is set by the user.
- Displays the results of the optimized portfolio versus an equally-weighted portfolio consisting of the same securities.
- **Pandas** dataframes were used to manipulate and store data.
- **PyPortfolioOpt** implemented the efficient frontier algorithm to optimize risk to return.