Hojung Park

LinkedIn | 778-751-5843 | hojungpark.devops@gmail.com | GitHub

EDUCATION

Simon Fraser University **Burnaby, BC**
Bachelor of Applied Science in Computer Science 2019-Present

SKILLS

- **Development**: Java, Python, Golang, HTML/CSS/JavaScript, React, Spring Boot
- **Testing**: Appium, Selenium, TestNG, Postman, Charles Proxy
- **Operation**: Linux, macOS, Git, Maven, Jenkins, Docker, Kubernetes, AWS

WORK EXPERIENCE

Atimi Software **Vancouver, BC**
Test Automation Engineer (Co-op) *01/2023 - present*

- Automated mobile and web application testing using **Java**, **TestNG**, **Appium**, and **Selenium**, resulting in a **58%** reduction in testing runtime through implementation of the **Page Object Model** design pattern in the testing framework.
- Developed **75+** comprehensive test cases to verify software functionality and performance, ensuring software quality and reliability.
- Streamlined the building and testing process with **Jenkins** for continuous integration, resulting in faster build times and increased efficiency.
- Verified backend API responses using the **REST** assured library and **Postman**, ensuring accurate data transmission and response handling.
- Leveraged Atimi's internal resources including **Bitbucket**, **Confluence**, and **Jira** improving project management efficiency and enhanced team collaboration.

PROJECTS

CrewCentral | *React, Spring Boot, MySQL, Jenkins, Docker*

- Constructed a full-stack CRUD web application for managing employee information, utilizing **REST API** for data exchange.
- Implemented **Axios Library** for efficient front-end to back-end data transfer.
- Integrated **Spring Boot** and **MySQL** using **Spring Data JPA**, allowing for seamless access and management of database information.
- Containerized the application with **Docker** to facilitate deployment and scalability.
- Automated the building, testing, and deployment processes with **Jenkins**.

LinkVerify | *Go, Docker, Kubernetes*

- Implemented a **Go** program to perform **HTTP** GET requests on a list of websites.
- Utilized **goroutines** to concurrently process and verify multiple URLs, enhancing the efficiency and speed of the program.
- Employed **Docker** containerization to streamline deployment and enable scalability.
- Utilized **Kubernetes** for container orchestration and management, enabling automatic scaling and self-healing capabilities.

CLI-Talk | *C*

- Constructed a **socket-based** program that allows for terminal communication between users.
- Implemented **mutex locks** for efficient thread synchronization and avoiding deadlock.
- Employed **Valgrind** to automatically detect memory management and threading bugs.
- Utilized the **GDB** to analyze program execution for debugging purposes.