Dohyun Moon

dohyundoor@gmail.com +1 343-777-1702 LinkedIn GitHub Website

Skills

Languages: Python, C/C++, Bash, HTML, CSS, JavaScript, TypeScript, Java, Rust
Technologies: Pandas, NumPy, Beautifulsoup4, PyFPDF2, Flask, Bootstrap, Selenium, PyTest, Firebase, React Native
Tools: Gitlab, Docker, Linux CLI, Jira, Jenkins, Android Studio, Robot Framework, Katalon Studio

Education

University of Waterloo - Computer Engineering (co-op) B.A.Sc. *Waterloo, ON* - Sep 2019 – Jun 2024

- C/C++, Python, Java, MATLAB

Experience

Software Developer – Quantum Bridge Technologies : *Toronto, ON* - Sep 2023 – Dec 2023

- Contributed to the development of Quantum Key Distribution (QKD) Simulation software using **Python** through implementing input/output processing modules and simulation graph generation using **NetworkX**.
- Utilized **PyFPDF2** PDF generation module to implement report generation output module for the program.
- Established and managed a robust CI/CD pipeline for the QKD Simulation program on **GitLab**, integrating **Sphinx** documentation, **PyTest** for unit testing, and enforcing code quality through **PyLint** and **Black**.

Systems Software Developer – Ford Motor Company : *Kanata, ON* - Jan 2023 – Apr 2023

- Developed **Bash** script to facilitate an Adaptive AUTOSAR development environment for multiple teams on Linux.
- Ensured seamless **Bash** script compatibility across multiple version releases of Adaptive AUTOSAR package, resulting in efficient updates for all users.

Network Operations Developer – Rogers Communications : *Brampton, ON* - Jan 2022 – Apr 2022

- Utilized **Python** and **Robot Framework** to develop automated solutions for streamlined resolution of customer ticket flows related to various network device issues.
- Analyzed customer ticket data using **Pandas** to generate prioritized monthly master plans for automated solutions, based on thorough data analysis.

DevOps Software Developer – Imagine Communications : *Waterloo, ON* - May 2021 – Aug 2021

- Developed and provisioned Linux and Windows VM networks using **Terraform** on vSphere client, deploying Docker containers using Ansible within the Terraform scripts.
- Configured foundational automated testing jobs on Jenkins server using **Pester** and **Groovy** pipeline language.

Projects

UFC Matchup Outcome Predictor – Python

- Designed and implemented a **Python** program utilizing custom matchup prediction algorithms to predict the outcome of hypothetical UFC matchups, considering factors such as weight class difference, experience, and age.
- Developed a user-friendly GUI using **Tkinter** and implemented **Beautifulsoup4** to extract fighter images from UFC websites, optimizing the process with user-inputted fighter information.

Emoji URL Shortener – HTML, CSS, JavaScript, Python

- Developed a URL shortener service using **Jinja2** HTML templating, **Flask** backend integrated with **MongoDB** database.
- Incorporated a custom encoding method using emojis for more visually appealing shortened URLs.

Sorting Algorithm Visualizer – HTML, CSS, JavaScript

- Developed an interactive sorting algorithm visualizer using **HTML**, **CSS**, and **JavaScript** that allows users to witness the sorting process of various algorithms (bubble sort, selection sort, insertion sort, and quicksort) in real-time.
- Observing the blocks transition from an unsorted arrangement to a gradient showcases the sorting logic effectively.

City of Waterloo Drivers' App (Fourth Year Design Project) – Python, TypeScript (React Native)

- Designed and built driver's utility app for the city of Waterloo using **Python** and **TypeScript** as a fourth-year design project with three fellow fourth year computer engineering students.
- Developed a mock City of Waterloo government server using **Flask** to simulate real-world functionalities, including processing speeding, red-light, and parking tickets, validating user information, and handling ticket payments.
- Enhanced the user experience by incorporating an interactive Google Map with custom markers for nearby speeding traps, red-light cameras, and parking availability, all built with **React Native**.