

## SUMMARY OF QUALIFICATIONS

- Motivated student passionate about autonomous systems, data management, and backend/full-stack development.
- Proven experience in coding languages, agile methodology, and software development life cycle (SDLC).
- Effective teamwork & project management as demonstrated through coursework & previous work experience.

## TECHNICAL SKILLS

- Programming Languages: Java (Proficient), Python (Proficient), SQL (Proficient), C (Beginner), JavaScript (Beginner), HTML (Beginner), CSS (Beginner)
- Frameworks: Spring (Intermediate), React (Beginner), NodeJS (Beginner)
- Technologies: Git, Jira, Jenkins CI/CD, JMeter API testing, Postman, Windows, Linux, Maven, Tomcat, IntelliJ, VSCode, JDBC Connectivity, SQL/NoSQL databases

## EDUCATION

**University of Washington, Seattle, WA**

*Expected June 2023*

**Bachelor of Science, Computer Engineering, 3.84/4.00 GPA (Dean's List)**

- Relevant Coursework: Java Fundamentals, Data Structures and Algorithms, Interaction Programming (Android Development), Data Programming in Python, Software Tools (Git and Terminal), Databases (SQL, NOSQL)

**International School, Bellevue, WA**

*September 2015 – June 2019*

**High School Diploma, 4.0 Unweighted GPA**

- National AP Scholar, National Honor Society, National Science Honor Society, National Math Honor Society, National Technical Honor Society, FRC Robotics Varsity Letter Recipient

## RELEVANT EXPERIENCE

**TIBCO Software, Jaspersoft Product, Palo Alto, CA**

**Software Engineer Intern**

*June 2021 – September 2021*

- Identified problems and potential solutions in windows development environment setup for Jaspersoft to help streamline onboarding process for new windows developers trying to work on the product in the future.
- Researched intricacies of OS/soft PI system time series database and helped integrate it with backend of Jaspersoft business analytics platform via JDBC connection protocol.
- Implemented a throttling and prioritization architecture for certain REST endpoints of the Jaspersoft product to prevent overloading of computing resources and to boost efficiency of server response time especially in high concurrency environments with large clients.
- Created technical documentation, use-case presentations, and demoed these POC projects to management and customer support engineers to show benefits and explain future optimizations to be deployed in production.

**Husky Robotics Team, University of Washington, Seattle, WA**

**Software Engineer**

*September 2020 – June 2021*

- Developed the mission control interface, which allows users to have control of different components of robot while viewing information about mission objectives, camera feed, and telemetry.
- This full stack project has a React-based front-end and a Node-based backend.

## PROJECTS / RESEARCH:

**A\* Algorithm Shortest Path Visualizer, Bellevue, WA**

*August 2020*

- Created a 2-D shortest path visualizer using the Pygame python library. Implemented the program using the priority queue variation of the A\* search algorithm and the Manhattan distance formula as a heuristic based on the grid format.

**Predictive Analysis of Movie Success from Datasets, University of Washington, Seattle, WA**

*March 2020*

- Utilized dataset scraped from IMDb movie database and used Python libraries such as Pandas, Scikit Learn, and Matplotlib to effectively visualize the data and predict the effects of features such as movie budget, production company, country of origin, genre, revenue, release date, run time, etc. on the success of a movie.

**E-Stash, DubHacks, University of Washington, Seattle, WA**

*October 2019*

- Created a receipt stashing app as part of a four-person team competing in the biggest 24-hour hackathon in the Pacific Northwest. This application allows easy storage and provides convenience so users don't have to search through their emails or wallets to find certain receipts.
- Utilized Python and Google Vision APIs for this project. Users input images of their receipts and the program will stash it into an inventory based off of data gathered from text detection.