Molly Braswell

419 E. 36th St. Apt. 208 | Charlotte, NC, 28215 | mollybraswell232@gmail.com | (804)898-8553

EDUCATION

University of South Carolina Columbia, South Carolina August 2021

Bachelor of Science, Computer Science

EXPERIENCE

Spectrum Reach Charlotte, North Carolina February 2023 - Current

Full Stack Developer

- Used Angular and Python to help create and maintain full stack applications
- Utilized AWS CloudFormation to create and scale infrastructure
- Helped rewrite multiple APIs utilizing uvicorn
- Stood up an application to work with our mapbox components

InfraSight Software Charlotte, North Carolina

Associate Software Engineer September 2021 - October 2022

- Utilized Micronaut and Java to produce a REST API
- Used Mockito and Micronaut Test Resources to write unit tests
- Helped update and maintain a metrics collection service written in C++
- Used C++ to create a License Manager for the metrics collection service
- Led training/onboarding of new engineers

University of South Carolina Columbia, South Carolina

Researcher September 2019 - August 2020

- Utilized AWS Lambda to ingest data from Withings API
- Used Android Studio to create interfaces/applications

Teacher's Assistant August 2017 - May 2019

- Assisted students with debugging of Java assignments
- Mentored students on more efficient and advanced Java practices

Boeing St. Louis, Missouri May 2019 - August 2019

Full Stack Intern

- Used Agile and Lean practices to improve problem solving and efficiency for consulted teams
- Helped build the front end, back end, and pipeline on a new project

SKILLS

Languages and Frameworks

- Java, C, C++, Python, Typescript, Angular, Micronaut, Spring

Technologies

- Linux, Git, Docker, AWS, MongoDB, SQL, Postgres, Flyway

PROJECTS

Online CV

- braswell.dev
- Written using expressJS for the frontend and Rust's rocket for the api
- Features a small collection of personal projects

Infrasight License Manager

- Used VCPKG to manage and add custom packages
    - Created a Node.js application to easily share and add custom packages to VCPKG
- Utilized Licensepp and crypto++ to create secure RSA-based licenses
- Employed CMake to help build and package the application

Keyboard firmware

- Used C and qmk to write firmware for pi40 and Piggy60 keyboards