

# WESTON P. GREENE

Full Stack Engineer | Raleigh, NC

## CONTACT

-  [linkedin.com/in/wpgreene](https://www.linkedin.com/in/wpgreene)
-  [github.com/RunItBack1127](https://github.com/RunItBack1127)
-  [westonpgreene.com](https://westonpgreene.com)
-  [westonpgreene@gmail.com](mailto:westonpgreene@gmail.com)

## EDUCATION

-  **North Carolina State University**  
Raleigh, NC  
Bachelor of Computer Science  
Aug 2022

## SKILLS

- Java
- Python
- Node.js
- Typescript
- Vue
- React
- HTML5/CSS3

## COMPETENCIES

- REST API Development
- Cloud Infrastructure (AWS)
- Containerization (Docker)
- Version Control (Git)
- Databases (PostgreSQL)

## EXPERIENCE

### Software Engineer Intern, DevEx Platform

*SailPoint Technologies, Inc.*

June 2022 – August 2022

- Collaborated within a large, agile team to develop and showcase the end-to-end functionality of a developer-first microservice dashboard built using React and Typescript
- Streamlined and improved upon the functionality of several large internal microservices through extensive use of Java
- Migrated and expanded the infrastructure of several large apps and services to a cloud-native stack using AWS services such as DynamoDB, RDS, and ECS
- Worked diligently within the Software Development Life Cycle (SDLC) and communicated technical details and documentation across several platforms using Jira and Confluence

### Full Stack Developer

*EcoPRT @ NC State*

May 2021 – current

- Assumed responsibility for implementing the main website redesign for the EcoPRT autonomous vehicle laboratory, built using Vue and Typescript
- Oversaw creation and incentivizing adoption of a 3D model dashboard used for visualizing changes to the vehicle, built using Vue, Flask, and Three.js

## PROJECTS

### EcoPRT Website Redesign

December 2021 – current

- Transitioned from the previous WordPress domain to a fully featured custom web application built using Vue and Node.js
- Lead deployment of the web app using GitHub Actions and Google App Engine, along with container orchestration using Docker