Isaac **Semaya**

San Francisco, CA
401.340.6446
isaac_semaya@alumni.brown.edu

LinkedIn: linkedin.com/in/isemaya   **GitHub:** github.com/isemaya   **Portfolio:** isaacsemaya.com

>LANGUAGES AND TECHNOLOGIES
- **Proficient:** Java, Python, NumPy, SciPy, JavaScript, Bash, LaTeX, Git, Racket, SQL
- **Exposure:** C, Scala, GCC, Go, Maven, D3.js, MapReduce, TensorFlow, Protractor, Selenium, Node.js, AngularJS

>EDUCATION
B.S. Computer Science (GPA: 3.52), *Brown University*   May 2018

>PROFESSIONAL EXPERIENCE
Amazon | Software Development Engineering Intern | Seattle, WA   Jun 2017 - Aug 2017
- Added device information such as device location, App Version to existing remote configuration JSON file in Android Studio to prevent redundant remote configuration files from being created, saving $700.
- Designed criteria evaluator to identify match criteria and characteristics in Java so different types of criteria were systematically evaluated in modular fashion and enabling support for new criteria over 65% faster.
- Engineered device eligibility manager that applied criteria information specified in remote configuration file to produce "eligible" or "ineligible" output for devices in Java.
- Crafted UI alert blocking App usage if device did not meet eligibility criteria and listed all of reasons for device ineligibility in JavaScript, HTML5, and CSS3 saving $2,500 in patch development time.

Onshape | QA Automation Development Engineering Intern | Cambridge, MA   May 2016 - Aug 2016
- Increased coverage of API tests regarding document search filters and account storage capacity by 10% using Mocha testing framework in JavaScript and Node.js.
- Refactored CAD shape generations to use API calls with JavaScript, AngularJS, Protractor, and Selenium, decreasing average test time to be below test timeout of 0.1 seconds.
- Restructured drop-down menus to remove ambiguity using JavaScript, AngularJS, Protractor, and Selenium.
- Investigated feasibility of feature within Protractor framework using JavaScript, AngularJS, and Selenium, concluding that refactor was not worthwhile, saving company more than $750 in development time.

OpenTable | Software Engineering Intern | London, United Kingdom   August 2013
- Searched for restaurants within time defined radius using JavaScript, saving 5 - 6 steps in location process.
- Found restaurants under conditions such as walking only, walking and bus, and walking and train, utilizing iGeolise API, improving time saved for users by 80%.
- Implemented markers to indicate current location and find restaurants within time radius using Google Maps API, saving users from opening X - 1 further pages on Google Maps.

Brown University | Computer Security Teaching Assistant | Providence, RI   Jan 2018 - May 2018
- Led TA hours for 2 hours per week to explain security concepts and guide students on homework and projects.
- Demonstrated how cookies are used in modern websites using Chrome inspect browser tool and JavaScript.

Brown University | UI/UX Teaching Assistant | Providence, RI   Aug 2017 - Dec 2017
- Ran 5 labs for 20 students per session to learn ReactJS, Proto.io, Balsamiq, and Invision; held office hours for A/B testing and eye tracking assignments to explain concepts and debug student code.
- Collaborated in writing JSON to CSV log file converter in Java and TheEyeTribe API, reducing development time for 150+ students (especially non-CS students) in coding eye tracking gaze visualizer.

>PROJECT WORK
mStats | Data Scientist + Software Engineer | youtu.be/4wFHXQOwaXA   Feb 2016 - May 2016
Suite of visualizations that analyze user's Facebook messages, using their own personal "big data."
- Extracted raw messages from HTML files and counted Facebook messages, visualizing frequencies in Python and D3.js to compare frequency of 70,000 messages sent during daytime versus nighttime.
- Developed webpage in Flask to run Python scripts that parsed messages and displayed 8 D3.js visualizations.
- Mapped messages as "bag of words" to train machine learning model (using the Random Forest algorithm) to predict 1 friend out of 500+ most associated with contents of message.