Alvin Hong Nguyen

alvinnguyen@berkeley.edu • alvinnguyen116.github.io/portfolio/

EDUCATION

UNIVERSITY OF CALIFORNIA BERKELEY *2015 - 2019*
Bachelor of Arts in Cognitive Science & Computer Science *3.4 GPA*

TECHNOLOGIES

Frameworks: Angular, React.js, Polymer, Ruby on Rails
Languages: Typescript/Javascript, Python, Java
Tools: CSS/SASS, HTML, Jasmine, GitHub, Bootstrap, jQuery, Regex, SQL

EXPERIENCE

Accenture / Google Mountain View, CA
Front-End Contractor *June 2019 - October 2019*

- Conducted UI experiments in a legacy Polymer app with custom events, behaviors, and global variables.
- Designed a 'Google Product' by following Google Material guidelines and using pre-built internal/public components
- Built a new internal Frontend tool for managers with basic CRUD operations
- Participate in daily meetings, code reviews, and debugging sessions

Cardinal Hire San Francisco, CA
Freelance Front-End Engineer *June 2018 – December 2018*

- Translated static UI/UX designs into responsive, cross-browser compatible, and dynamic markup
- Debugged configuration files, markup issues, and rendering speed while prioritizing customer facing issues

Seva Foundation Berkeley, CA
Web Intern *June 2017 - August 2017*

- Utilized Google Maps API for custom legends, category filtering, info windows, and node clustering
- Converted +100 nodes into JSON with exact geolocations, native HTML, and embedded images

PROJECTS

Upper Division Computer Science Courses *June 2016 – May 2019*

- Coded 5 Large Projects (20-40 hours of work each)
    - RNA-Sequencing: Implement Burrow Wheelers Transform to align RNA reads to genome of 11 million bases. Used a transcriptome to align known reads and TopHat with constraints to align hidden reads with up to 95% accuracy
    - Ghostbusters: Create an AI to locate invisible ghosts using Bayes Net and Joint Particle Filtering. The observations are calculated using Exact and Approximate Inference for initializing and updating the AI's beliefs as time elapses
    - Inside SQL: Design a relational database with B+ trees, Query Optimization, and Multi-Granularity Locking
    - NP-Hard: Generate approximate solutions for geometric TSP by reducing to MST, employing a variety of heuristics, and avoiding local optima with simulated annealing
    - Software Engineering: Collaborate with a client to build a web app which solves common matching problem