

TECHNICAL SKILLS

JavaScript (ES5+), TypeScript, Node, Express, RESTful architecture, React (Hooks, Router), Redux, Webpack, GitHub + Git workflow, Relational/Non-relational databases, PostgreSQL, Mongoose, HTML5, CSS3, SASS, Agile (Scrum) methodology, Test-Driven Development (Jest, Enzyme, Puppeteer), Electron, React Router, Docker, Travis CI, GitHub Actions, AWS (EB, EC2)

EXPERIENCE**Codesmith — Software Engineer**

2021 – Present

- Employed Docker to enable containerized development environments, operating system virtualization, and automated configuration and installation of runtime dependencies to ensure a consistent development experience across platforms.
- Converted CI/CD infrastructure to GitHub Actions (from Travis CI) in order to run exponentially more jobs concurrently, streamline processes by cutting third-party integrations, and lower costs by utilizing features on existing GH account.
- Maintained custom YAML content parser in lieu of a traditional CMS to map JSON data derived from static YAML to dynamic React components, enabling a seamless and efficient workflow between technical and non-technical team members.
- Provided mentorship and technical training to junior developers across the stack via one-on-one code reviews, internal lectures, SCRUM methods, and pair programming in order to ensure consistency and best practices across the codebase.
- Harnessed Jest, Supertest, React Testing Library, and Enzyme to expand testing suite coverage to ensure added features do not impact dependability of our CI/CD pipeline for the Codesmith public site, which attracts over 165,000 annual visitors.

Spearmint — Software Engineer | open-source test-generating developer tool

2021 – Present

- Generated React Testing Library, Puppeteer and Jest tests for users to export and run, delivering performant tests to JavaScript developers in order to optimize their development process by abstracting away the responsibility of writing tests.
- Leveraged Axe-Core's accessibility testing engine to create reliable accessibility tests to be exported to the end user, all while guaranteeing zero false positives and complying with WCAG, W3C, and Section 508 accessibility guidelines.
- Utilized TypeScript to refactor JavaScript codebase to include a type system that catches type-errors at compile time rather than runtime, with the goal of ensuring continued scalability and reliability as the project expands.
- Built with a Node runtime (in Electron) to create a native desktop application that leverages the CommonJS module pattern to organize code and control scoping — creating space for future scalability of both the team and the application.
- Leveraged Spectron in conjunction with Google's Accessibility Developer Tools to perform a test-driven refactoring of legacy code and UI to launch an application that delivers a vision and motion accessibility-first desktop experience.
- Used SASS to adhere to DRY principle by creating variables, leveraging nesting, and utilizing modules — allowing for speedy yet impactful improvements to the UI/UX, such as pronounced focus borders for non-mouse user accessibility.
- Product developed under tech accelerator OS Labs (opensourcelabs.io).

OPEN SOURCE**forEach | scholarship donation platform**

- Utilized a PostgreSQL database to securely store relational user and donation data, enabling ACID compliant behavior.
- Used Express framework to efficiently build RESTful architecture by engaging the middleware design pattern and writing routers and controllers to modularize backend requests, thus enhancing code readability and scalability.
- Implemented OAuth 2.0 alongside Passport middleware to allow users to log in with pre-existing Google credentials, minimizing the barriers of entry as well as delegating the responsibility of authentication to a trusted, secure API.
- Incorporated Bcrypt to encrypt sensitive client information in SQL database by leveraging Bcrypt's SALT rounds and one-way hashing algorithm, ensuring safe storage of data and limiting potential damage in case of breach.

OS Cards | multi-sensory flash card app

- Configured Webpack to bundle app files for future scalability, maintainability and faster load times through HMR, ES6+ transpilation for developer packages, and minification + uglification features that decrease bundled file size.
- Utilized React Router to boost loading performance by establishing static routes to minimize server calls, handle progressive resolution of views, and allow backward/forward history navigation, all within a single-page application.
- Acted as scrum master to make user stories come alive through delegating tasks and enforcing strict sprints and code review practices, in order to maximize the productivity of the team and ensure that the user-experience is always prioritized.

RECENT TALKS

- React Hooks vs. Redux: How to Architect State? | SingleSprout Speaker Series
- Women's Intro to JavaScript: Variables & Looping | BuiltWithCode LA

EDUCATION**University of California, Santa Cruz | B.A. in Business Management Economics**

2014 – 2018

- Relevant coursework: Introduction to Programming Java, Introduction to Programming C++

INTERESTS

involvement in the ALLY community • raising monarchs from egg to butterfly • overwatering my cacti • creating plant propagation stations • acro-yoga & hot yoga (not simultaneously) • visiting new parks with my schnauzer-impersonating pup