

Annie Shin | Software Engineer | Los Angeles

(213) 986-6714 | annieshin51@gmail.com | [linkedin.com/in/annieshinn](https://www.linkedin.com/in/annieshinn) | github.com/annieshinn

TECHNICAL SKILLS

Strong: Javascript (ES6+), Typescript, React (Hooks, Context), Redux, Node, Express, Webpack, Git workflow, Relational/Non-relational databases, PostgreSQL, Mongoose, HTML/CSS/SASS, Agile/Scrum methodology

Experienced: Test-Driven Development (Jest, Enzyme, Puppeteer), React Router, React Native, AWS, Docker, Travis CI, RESTful API, Svelte, Gatsby, Material UI, OAuth 2.0, Passport

EXPERIENCE

Spearmint | *open-source test-generating developer tool*

2021-Present

- Built with a Node runtime (in Electron) to create a native desktop application that can utilize NPM packages, while avoiding globals by only exposing node_modules that are relevant to specified files, and modularizing code.
- Utilized Typescript to refactor legacy Javascript codebase in order to ensure continued scalability and reliability.
- 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.
- Employed Jest to generate unit tests to deliver to the user using Jest's built-in libraries and compatibility with React.
- Generated Puppeteer tests for users to export and utilize, delivering performant tests tailored for JavaScript developers.
- Employed React Context and React Hooks to build state management that follows a predictable and unidirectional data flow in order to reduce prop-drilling and modularize code without the overhead of a third-party state manager.
- Incorporated Sass's stylesheet variable and nesting capabilities in order to adhere to DRY principle and modularize codebase, allowing for speedy yet efficient changes to the UI/UX, such as focus borders for non-mouse user accessibility.
- Utilized react-beautiful-dnd API to create draggable components that were both accessible to non-mouse users as well as lightweight and straightforward regarding configuration for usage with nested list components.
- 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.
- Product developed under tech accelerator OS Labs (opensource.labs.io).

forEach | *web application for scholarship fund management*

2021

- Incorporated Bcrypt to encrypt sensitive client information in SQL database by leveraging Bcrypt's one-way hashing algorithm, ensuring safe storage of data and protection from dictionary attacks.
- 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 creating a sense of trust with the user by associating with a familiar name.
- Utilized a PostgreSQL database to securely store relational user and donation data, enabling ACID compliant behavior.
- Established Express framework to efficiently handle requests to a myriad of endpoints by engaging the middleware design pattern, writing routers and controllers to modularize backend requests, and enhancing code readability.

OS Cards | *multi-sensory flash card app*

2020-2021

- Configured Webpack to bundle app files for future scale, maintainability and faster load times through HMR, ES6+ transpilation for developer packages, and using minification and uglification features to decrease bundled file size.
- Utilized React Router to boost loading performance by establishing static routes to minimize server calls within single page applications and handle progressive resolution of views.

Proj-Org | *productivity tool to track project progress*

2020-2021

- Implemented React with Redux state management in order to modularize and render components to the front-end while maintaining a singular store and harnessing the power of Redux DevTools.
- Utilized MongoDB to store nonrelational data regarding projects and each project's tasks in order to leverage the noSQL database's potential to create nested data and store a wider variety of data types, such as image files

RECENT PUBLICATIONS

- SingleSprout Speaker Series Tech Talk: "React Hooks vs. Redux. How to Architect State?"

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 AIYY 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