linkedin.com/in/annieshinn
github.com/annieshinn

Annie Shin

Software Engineer | Los Angeles, CA

annieshin51@gmail.com
(213) 503-2466

TECHNICAL SKILLS

Ruby, Ruby on Rails, GoLang, Terraform, Google Cloud Platform (Spanner, BigQuery, Pub/Sub), Kubernetes, 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, RSpec), Electron, React Router, Docker, Travis CI, GitHub Actions, AWS (EB, EC2)

EXPERIENCE

Handshake — Software Engineer III 2021 – Present

- Constructed a Go microservice from scratch in order to boost notification relevancy through the use of data and Machine Learning models, which has led to incredible results, visible through close tracking & experimentation, such as: 2x as much reach to student users, 2.2x increase in click-through-rates, and no change in unsubscribe rates.
- Orchestrated a GCP Spanner change in which we had to drop and recreate tables yet maintain rows of data, and through meticulous planning and coordination, successfully completed the migrations with zero downtime and zero snags.
- Work closely with GCP Pub/Sub coupled with a custom-built domain events pipeline to ensure transactional guarantees between the monolith app and various miniservices and microservices, particularly those regarding IAM and notifications.
- Establish company-wide blueprints by being a pioneer when it comes to novel projects such as: building a Go microservice, setting up a service with our custom-build domain events pipeline, setting up infrastructure to prepare for ML models.
- Drastically improved upon documentation for processes and projects regarding notifications, notifications relevancy, microservices, IAM, onboarding, on-call, the domain events pipeline, and team processes.
- Provided mentorship through helping teammates onboard, improving upon internal documentation, and being a founding or active member of several community groups and employee resource groups.

Codesmith — Software Engineer 2021

- 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 2020 - 2021

- 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).

RECENT TALKS

- How to set your workspace up for success | Internal talk at Handshake
- Notifications Relevancy Service | Internal talk at Handshake
- 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

INTERESTS

involvement in the A11Y 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