Nicholas **Chiang**

Software Engineer · Full-Stack Web Development · Photography

Palo Alto, California · No visa sponsorship required to work in the US

cv@nicholaschiang.com | nicholaschiang.com | nicholaschiang | nicholaschiang

Skills

Programming	JavaScript, Python, Java, C++, TypeScript, SCSS, Make, Bash, LaTeX
Applications	Photoshop, Premier Pro, Inkscape, Ableton, FL Studio
Technology	React, Remix, Next.js, Cypress, PostgreSQL
Tooling	Vim, Git, Linux, AWS, GCP, CI/CD

Experience

Software Intern

Menlo Park, CA

Numbers Station · numbersstation.ai *2022-06–Present*

- Designing and building front-end user interfaces for a state-of-the-art ML platform.

Founding Engineer

San Francisco, CA

Roote Foundation · roote.co *2022-03–2022-08*

- Developed a web app for interacting with articles and the tweets about them.
- Harnessed Hive and Rekt rankings to categorize tweet and article feeds.
- Built an engine to sync Twitter API data to a normalized PostgreSQL schema.

Software Engineer

Palo Alto, CA

Tutorbook · tutorbook.org *2019-02–2022-09*

- Created a web app used by schools and nonprofits to connect students with volunteer tutors and mentors.
- Worked with two schools and three nonprofits that serve over 5000 students and 1000 volunteers.
- Drafted a privacy policy and a terms of use compliant with California's CSDPA v2.
- Wrote and configured continuous integration for Cypress tests (74% code coverage).
- Contributed to open-source libraries such as React, Next.js, RMWC, and the Firebase SDK.

Software Engineer

San Francisco, CA

Hammock · readhammock.com *2021-04–2021-12*

- Developed a web app where you can enjoy reading and learning from newsletters.
- Increased page speed by migrating client-side business logic to serverless API functions.
- Protected against XSS by sanitizing email HTML server-side.
- Worked with Google's OAuth2, People, and Gmail APIs.

Research Intern

Palo Alto, CA

Stanford University · sing.stanford.edu *2018-09–2019-05*

- Designed a methodology for building hardware component knowledge bases using machine-learning.
- Extracted both textual and non-textual information to create relational databases for hardware components.
- Produced application studies that highlight how these databases make hardware component selection easier.

Publications

2020 **Creating Hardware Component Knowledge Bases with Training Data Generation and Multi-task Learning** *ACM TECS*

Luke Hsiao, Sen Wu, **Nicholas Chiang**, Christopher Ré, and Philip Levis

sing.stanford.edu/site/publications/tecs20hack.pdf · *github.com/lukehsiao/tecs-hardware-kbc*

2019 **Automating the Generation of Hardware Component Knowledge Bases** *LCTES*

Luke Hsiao, Sen Wu, **Nicholas Chiang**, Christopher Ré, and Philip Levis

sing.stanford.edu/site/publications/hack-lctes19.pdf · *github.com/lukehsiao/lctes-p27*