

# Nicholas Chiang

Software Engineer · Web Development

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

✉ [cv@nicholaschiang.com](mailto:cv@nicholaschiang.com) | 🏠 [nicholaschiang.com](https://nicholaschiang.com) | 🌐 [nicholaschiang](https://nicholaschiang.github.io) | 📄 [nicholaschiang](#)

## Skills

**Programming** TypeScript, Python, Java, C, CSS

**Technology** React, Vite, Tailwind, Remix, Next.js, Cypress, SQL

## Experience

### Software Engineer

Menlo Park, CA

Numbers Station · [numbersstation.ai](https://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](https://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 data between Twitter and a Postgres database.

### Software Engineer

Palo Alto, CA

Tutorbook · [tutorbook.org](https://tutorbook.org)

2019-02–2022-07

- 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.
- Configured continuous integration for and wrote 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](https://readhammock.com)

2021-04–2021-12

- Developed a web app where you can enjoy reading and learning from newsletters.
- Decreased LCP by migrating client-side business logic to serverless API functions.
- Worked with Google's OAuth2, People, and Gmail APIs.

### Research Intern

Palo Alto, CA

Stanford University · [sing.stanford.edu](https://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 | <b>Creating Hardware Component Knowledge Bases with Training Data Generation and Multi-task Learning</b>  | ACM TECS |
|      | Luke Hsiao, Sen Wu, <b>Nicholas Chiang</b> , Christopher Ré, and Philip Levis   |          |
|      | 📄 <a href="https://sing.stanford.edu/site/publications/tecs20hack.pdf">sing.stanford.edu/site/publications/tecs20hack.pdf</a> · 🌐 <a href="https://github.com/lukehsiao/tecs-hardware-kbc">github.com/lukehsiao/tecs-hardware-kbc</a> |          |
| 2019 | <b>Automating the Generation of Hardware Component Knowledge Bases</b>  | LCTES    |
|      | Luke Hsiao, Sen Wu, <b>Nicholas Chiang</b> , Christopher Ré, and Philip Levis   |          |
|      | 📄 <a href="https://sing.stanford.edu/site/publications/hack-lctes19.pdf">sing.stanford.edu/site/publications/hack-lctes19.pdf</a> · 🌐 <a href="https://github.com/lukehsiao/lctes-p27">github.com/lukehsiao/lctes-p27</a>             |          |