Nicholas Chiang

Software Engineer · Web Development
Provo, Utah · No visa sponsorship required to work in the US
cv@nicholaschiang.com | nicholaschiang.com | nicholaschiang | nicholaschiang

Skills

Programming TypeScript, Python, Java, C, CSS
Technology React, Svelte, Tailwind, Remix, Next.js, SQL

Experience

Software Engineer — *Provo, UT*

Alation · alation.com — *2025-05–Present*

- Building the user interface for Alation's AI platform.

Software Engineer — *Menlo Park, CA*

Numbers Station · numbersstation.ai — *2022-06–2025-05*

- Helped build the company from day one through acquisition by Alation.
- Led frontend: design, testing, CI/CD, hiring, tooling, framework decisions.
- Built the user interface for a state-of-the-art machine learning 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 data between Twitter and a Postgres database.

Software Engineer — *Palo Alto, CA*

Tutorbook · 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 — *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 — *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*