nick young

email: nicholas_young@brown.edu | site: n-young.me | github: n-young | linkedin: in/ndashyoung

education

- **Brown University** [GPA: 3.96/4.00] - Concurrent ScM. in Computer Science, ScB. in Math and Computer Science — May 2023
  > Relevant Coursework: Time Series Databases, Database Management Systems, Distributed Systems, Networking, Programming Languages, Compilers, Graphics, Formal Methods, Probabilistic Methods, Cryptography, Number Theory.

experience

- **Stripe** - Software Engineering Intern (DocDB Fleet Management Team) - Seattle, WA — June - August 2022
  > Designed and implemented a distributed Mongo load simulation framework in Go, supporting up to 100,000 writes/sec.
  > Wrote a custom load generation engine and supporting user interface in React to evaluate database performance.
  > Spearheaded cross-team database host management migration in Go, deprecating hundreds of lines of legacy code.
- **Bloomberg LP** - Software Engineering Intern (DevX Team) - New York, NY (Remote) — June - August 2021
  > Deployed an adaptive connection pool in Python, reducing new connections by 98% and lowering latency by 50%.
  > Used internal telemetry tools to track connection pool performance; exposed data through 4 unique Grafana dashboards.
  > Delivered a React UI for users to interact with the DevX Policy Engine and improved SonarQube static analysis tooling.
- **Stochastic AI** - Software Engineering Intern - Cambridge, MA (Remote) — June - August 2020
  > Spearheaded the web team and set up CI/CD for the frontend and backend with Github Actions, GCP, and Netlify.
  > Engineered the backend microservices architecture from scratch using GraphQL, Docker, Rust, Python, and Javascript.
- **Stylindex** - Full Stack Engineering Intern - Los Angeles, CA (Remote) — June - August 2020
  > Developed an automated GraphQL emailing microservice using Rust, eradicating the need to manually send email alerts.
  > Revamped the internal analytics system by tracking API activity with Rust and Python and building new charts in React.

leadership

- **Database Management Systems** - Head Teaching Assistant — Fall 2021
  > Revamped course to operate in Go. Wrote reference implementations, autograders, and handouts for 8 assignments on the following: SQL, paging, hash tables, b+tree indices, query processing, query optimization, concurrency, and recovery.
  > Drafted conceptual assignments, coordinated the TA team, created the course website, and led all course logistics.
- **Adversarial Thinking in CS Education** - Research Assistant — Spring 2021
  > Published a paper on adversarial thinking in 100+ beginner computer science students in the ACM ICER 2021 conference.
  > Analyzed student responses to 300+ ethics assignments using intercoder reliability with Professor Shriram Krishnamurthi.
- **Full Stack at Brown** - President — Winter 2020 - Present
  > Led a web development club of 250+ members and 50+ client-facing projects. Oversaw operation of all project groups.
  > Organized and conducted a web bootcamp to teach HTML, CSS, JS, React, and SQL to new members every semester.

projects

- **TRustDB** [n-young/trustdb] — April 2021
  > Highly performant time-series database written in Rust. Optimized for writing and querying of high-cardinality data.
  > Pioneered boolean logic-based query optimization to accelerate conjunctive normal form queries by up to 1000%.
  > Leveraged finite-state transducer-based index compression to keep log indexes both time- and space-efficient.
- **NetStack** [n-young/netstack] — April 2022
  > Minimum viable RFC-compliant implementation of IP, TCP, and DNS written in Go as a proof-of-concept.
  > Supports traceroute, route aggregation, recursive IP resolution, and route caching on top of regular functionality.
- **Goo** [n-young/goo] — May 2021
  > Performant and minimal YAML- and Markdown-based static site generator built in Go as a replacement to Hugo.
  > Authored support for template partials, data injection, Markdown-to-HTML compilation, inline LaTeX, and :joy:-like emojis.
- **OxySAT** [n-young/oxysat] — March 2021
  > DPLL-powered SAT solver with custom heuristics to optimize satisfiability search, made in Rust.
  > Tested using an automated oracle built in Python. Compatible as a backend for the Forge model-checking language.