

nick young

email: nicholas_young@brown.edu | site: n-young.me | github: [n-young](https://github.com/n-young) | linkedin: [in/ndashyoung](https://www.linkedin.com/in/ndashyoung)

education

- **Brown University** [GPA: 3.94/4.00] - Concurrent ScM. in Computer Science, ScB. in Math and Computer Science May 2023
> Relevant Coursework: Multiparty Computation, Cryptography, Number Theory, Time Series Databases, Database Management Systems, Distributed Systems, Networking, Prescriptive Analytics, Formal Methods, Programming Languages, Compilers, Graphics.

experience

- **Databricks** - Software Engineer (Query Optimization) - Bellevue, WA August 2022 - Present
> Coming up next!
- **Stripe** - Software Engineering Intern (Document Databases Fleet Management) - 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 (Developer Experience) - 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.

leadership

- **Applied Cryptography** - Head Teaching Assistant [[website](#)] Spring 2023
> Founded, designed, and wrote the applied cryptography course at Brown University alongside Professor Peihan Miao.
> Built implementations and handouts from scratch for 6 assignments on cryptographic primitives, secure communication, secure authentication, zero-knowledge proofs, multi-party computation, and fully homomorphic encryption in C++.
- **Database Management Systems** - Head Teaching Assistant [[website](#)] Fall 2021
> Revamped the databases course at Brown University alongside Professor Stan Zdonik. Wrote implementations and handouts for 8 assignments on SQL, paging, hash tables, b+tree indices, query processing, query optimization, concurrency, and recovery.
> Wrote conceptual assignments, coordinated the TA team, created the course website, and led all course logistics.
- **Adversarial Thinking in CS Education** - Research Assistant [[paper](#)] 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 [[website](#)] Winter 2020 - Spring 2022
> 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](https://github.com/n-young/trustdb)] April 2021
> Highly performant time-series database written in Rust. Optimized for writing and querying of high-cardinality data.
> Leveraged finite-state transducer-based index compression to keep log indexes both time- and space-efficient.
- **NetStack** [[n-young/netstack](https://github.com/n-young/netstack)] April 2022
> Minimum viable RFC-compliant implementation of IP, TCP, and DNS written in Go as a proof-of-concept.
> Pioneered boolean logic-based query optimization to accelerate conjunctive normal form queries by up to 1000%.
> Supports traceroute, route aggregation, recursive IP resolution, and route caching on top of regular functionality.
- **Goo** [[n-young/goo](https://github.com/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](https://github.com/n-young/oxysat)] March 2021
> DPLL-powered SAT solver with custom heuristics to optimize satisfiability search, made in Rust.