

# Jason Yi

(336) 693-1206 | [jasonyi@unc.edu](mailto:jasonyi@unc.edu) | [LinkedIn](#) | [GitHub](#) | [Website](#)

## EDUCATION

### University of North Carolina at Chapel Hill

Chapel Hill, NC

*Bachelor of Science in Computer Science, Statistics and Analytics*

Aug. 2022 – May 2026

- **Coursework:** Distributed Systems, Operating Systems, Machine Learning, Algorithms, Data Structures, Databases, Models of Languages and Computation, Computer Organization, Stochastic Modeling, Probability
- **TA:** Algorithms (Spring 2025), System Fund. in C (Fall 2024), Data Structures in Java (Fall 2023, Spring 2024)

## TECHNICAL SKILLS

**Languages:** C/C++, Python, Java, TypeScript, JavaScript, HTML/CSS, Assembly, Swift/SwiftUI

**Frameworks/Libraries:** React.js, GraphQL, Angular, Node.js, PostgreSQL, NumPy, Pandas, Matplotlib, JUnit

**Developer Tools:** VSCode, Git, GitHub, Vim, Jira, Jenkins, Splunk, IntelliJ, Linux Kernel, AWS, XCode

## EXPERIENCE

### Amazon

May 2025 - Aug. 2025

*Software Development Engineer Intern*

Seattle, WA

- Incoming Software Development Engineer Intern, Summer 2025 - Team: **AWS Insights and Optimizations**

### NSF RTG Networks | UNC Statistics Department

Sep. 2024 – Dec 2024

*Undergraduate Research Assistant*

Chapel Hill, NC

- Advised by Dr. Chudi Zhong to develop **Interpretable Machine Learning** algorithms/pipelines by refining models such as Decision Trees and Generalized Additive Models to ensure better decisions in high-stakes situations
- Optimized the TreeFARMS algorithm in **C++** and **Python**, reducing runtime by **20%** through parameter tuning and tree depth constraints, enabling faster enumeration of almost-optimal **Decision Trees**

### Fidelity Investments

June 2024 - Aug. 2024

*Software Engineer Intern*

Durham, NC

- Developed Backend services in **GraphQL** via **Experience API** for Account Opening which impacts **50+ million users**, and Frontend services in **Angular and TypeScript** for Crypto IRA
- Implemented customer info, address validation, and risk analysis services to prevent user fraud or illegal activity during account opening using **TypeScript and GraphQL** by matching data from multiple downstream APIs
- Established excellent code quality and performance through unit testing via **Mocha, Splunk, and NestJS**

### CS+Social Good

Jan. 2023 – Dec 2023

*Full Stack Developer*

Chapel Hill, NC

- Collaborated with CATCH (Carolina Adapts Toys for Children) to develop a user-friendly and responsive platform to support their mission of adapting toys for children with special needs using **React, Firebase, UI/UX Design**
- Improved user navigation with a fixed Navbar and essential functions for efficient data management

## PROJECTS

### Distributed Stock Exchange | C++, FIX/FAST, FastDDS, MySQL, TCP/IP, Distributed Systems, Linux

- Engineered **FIX/FAST** protocol based high frequency **C++ stock exchange**, load-balancing over 3 matching engine nodes, supporting **1,000,000+** order insert request and transactions **per second**
- Supports **market/limit/stop order** matching through **Price-Time Priority**; built gateway node for **TCP/IP** request authentication/authorization from **MySQL** database; used **FastDDS** for inter-node communication
- Designed and integrated custom thread-safe optimized data structures (**priority queues, hash maps**) and utilized **GNU** profiling techniques for optimizing insert latency by **100x** from initial speeds

### CC Compiler | C++, LLVM, GNU Bison, Flex, Abstract Syntax Tree (AST)

- Built a custom toy programming language compiler in **C++** using **GNU Bison** for parsing and **Flex** for lexical analysis, generating an **Abstract Syntax Tree (AST)** and transforming it into **LLVM IR** code for execution
- Walked over the **AST** to generate byte/machine code for each node, and integrated **LLVM** to compile and execute the generated code, utilizing **llvm-config** for streamlined builds and testing

### Restaurant-Finder | PostgreSQL, Express.js, React.js, Node.js (PERN Stack)

- Developed a **full-stack website** for Creating, Retrieving, Updating, and Deleting restaurants and reviews
- Maximized data efficiency by **utilizing dual Postgres tables** for managing restaurant and review data
- Improved component data retrieval to automatic and minimized prop drilling by **utilizing Context API**