

# Alvin J. Zhang

[alvinjz2@illinois.edu](mailto:alvinjz2@illinois.edu) | (510)-378-5717

[github](#) | [linkedin](#) | [website](#) | [chess](#) | [goodreads](#)

## Education

University of Illinois Urbana-Champaign

August 2020 - May 2024 (est.)

### *Relevant Coursework*

*B.S in Computer Engineering, Minor in Business*

Algorithms and Models of Computation, Data Structures, Discrete Structures, Computer Systems & Programming, Artificial Intelligence, Analog Signal Processing, Statistics and Probability I & II, Introduction to Computing

*Languages* — Python, C, C++, C#, Go, Javascript, Typescript, SQL (PostgreSQL, SQLite)

*Frameworks* — React.js, Next.js, Flask, Markdoc | *Front End Development* — HTML, CSS, Tailwind CSS, Blazor

*Technologies*— Selenium, Xunit, Git, Azure DevOps, AWS (S3, EC2)

## Work Experience

VertexGraph Inc. — Software Engineer Intern

Summer 2022

### *Full Stack Development*

- Developed an internal API in C# to manage deletion of files shared between users. Worked with Postgres DB and pgAdmin for this. Added async client functions to facilitate API calls from other repositories
- Improved Blazor components to be interactive and intuitive. Created a page to manage a user's two-factor auth
- Lead efforts on creating the product's documentation site, picked up Next.js, Typescript, and Markdoc quickly. Responsible for all of the web dev and styling. Smoothly integrated 6+ interns' work into the Next.js application
- Agile development— completed tasks ahead of schedule consistently, finished documentation site in 2 weeks
- Effective CI practices with Git, streamlined the development process and code reviews for the team

### *Software QA Testing*

- Wrote structured and comprehensive unit tests for individual product features using Xunit
- Authored methodical testing procedures to simulate user flow across different facets of the product
- Utilized Azure DevOps efficiently to document, manage, and integrate 20+ bug fixes to the codebase

Quant Illinois — Quantitative Development

February 2022 - Present

### *Developer for Ceryx*

- Designed and implemented an internal Flask API that allows Quant members to interact with various brokers using a singular interface (Ex. Place buy/sell orders, get account P&L, etc using the Ceryx client).
- Secure authentication scheme by salting and hashing user info, also generating unique secrets to identify users
- Supports administrator/user access controls, as well as detailed logging of API requests by users

### *Developer for Plutus*

- Parallel processing of ticker data using Go in order to make Plutus, the project that backs up our data to AWS timestream databases, more efficient

## Personal Projects

### Personal Portfolio Website

- Built with React.js and Tailwind CSS, hosted on GitHub Pages. Interactive elements and reusable design principles

### TF2 Arbitrage Bot

- Multithreaded website scraping of backpack.tf using Selenium to grab listings data of an in-game item (since no API is available) and store them in a threadsafe priority queue. Arbitraged in order of decreasing profit margin
- Estimated ~7.0s of time taken to load, scrape, and process data before sending trade offers

### FilterTickers

- Utilized TwitterAPI to fetch and parse tweets which alert stocks with unusual options activity
- Made scatter plots with Matplotlib to visualize data and spot correlations between options data and prices