






Cyrus Singer

 [github.com](#)  [linkedin](#)  [cyrus.singer.dev](#)  cysinger@seas.upenn.edu  (561) 403-8133

EDUCATION

University of Pennsylvania

Bachelor of Science And Engineering in Computer Science

Aug 2022 - May 2025

Current GPA: 3.57/4.0

SKILLS

Languages: JavaScript/TypeScript, Java, Kotlin, Python, C/C++/C#, Haskell, Ocaml, SQL, HTML, CSS

Software Tools: MATLAB, GCP, AWS, Firebase, Docker, Git, GDB, Selenium

Software Frameworks/Libraries: React, Node.js, Express.js, Pytorch, Tensorflow, pandas, NumPy, tRPC, Prisma

General Skills: embedded-programming, full-stack development, system-level programming, customer engagement, source control, CI/CD, product management, version control, linux, powershell, bash, REST, HTTP

WORK EXPERIENCE

Software Engineering Lead on Bizzybots Platform | Wharton Behavioral Lab

2022 - Present

- **Led a team of five** to develop a web platform for creating custom controllable chatbots, used by over 500 students and professors across 5 universities
- Designed the platform using **React.js** for the frontend and **Express** for the backend, with **Firestore** as the database and **Firebase** authentication
- Orchestrated deployment on **GCP**, using **Docker** and **Cloud Run**, ensuring **99.9% uptime**
- Designed a system of independent **co-existing chatbot versions**, allowing for rapid deployment of new features
- Implemented a comprehensive **Jest** and **Selenium**-based test suite, reducing production bugs by 80%
- Conducted infrastructure load testing to verify the architecture can handle **10,000 messages per second**
- Published an Excel-style formulas package [link](#) and a number parsing package [link](#) on **npm**

Reference available upon request

Software Developer Intern | CFO4ALL at Penn Venture Lab

May 2024 - Sep 2024

- Worked on a web application to provide CPG companies with cash flow forecasting and insights:
- Set up **CI/CD** pipeline using **GitHub Actions** & **Docker** to deploy to **AWS App Runner** & **CloudFront**; reducing deployment time 90% and allowing production credentials to be removed from developers' computers
- Designed a scalable time series-tree data structure for financial data; integrated this data structure with backend analysis and frontend visualizations; **speeding up queries by 70%**
- Implemented a data ingest pipeline from Quickbooks to a **PostgreSQL** database on **Heroku** for customers with **250,000+ data points**; **automating the customer onboarding process**

Teaching Assistant for Internet and Web Systems (graduate level) | University of Pennsylvania

2024 - Present

- Through Office Hours and an online forum, I help students debug their code and understand the course material
- Mentor two groups through the final project, a distributed search engine with **1,000,000** indexed pages

PERSONAL PROJECTS

Webapp to play chess with an LLM | TypeScript, Next.js, LLMs, Minimax, | [source code](#)

2024

- A web application that allows users to play chess against a chess engine powered by LLMs
- Utilizes a compressed, budgeted, form of the **Minimax algorithm**, with LLMs generating moves and evaluating game states, running in parallel for scalable performance
- Included logging and tuning capabilities, allowing for easy monitoring and adjustment of the algorithm's parameters

Brittle Object Simulation | Python, GPU optimization, Physics Simulation | [source code](#),

2022

- Created program that simulated the internal stresses of rigid lattices under arbitrary forces and collisions using **Newtonian mechanics**
- Estimated internal bond stresses using **gradient descent** to fit internal stresses to the lattice's Newtonian motion
- Optimized the **bond estimation algorithm** to run with fewer gradient descent iterations
- Reimplemented the algorithm to run on a GPU, **improving performance over 10x**

RL Experiment | Java, Python, Tensorflow, Deep Q Learning | [source code](#)

2020

- Developed a 2D physics environment in **Java** to train agents to navigate an obstacle course
- Implemented various ML techniques, predominantly using **Double Deep Q Learning**, to drive the agents
- Implemented the neural network execution from scratch in Java for closer integration with the simulated environment
- Achieved success in training agents to execute precise jumps and navigate the environment effectively