Eugene Chao

epchao.github.io | linkedin.com/in/epchao | github.com/epchao | epchao@berkeley.edu

Education

University of California, Berkeley — Berkeley, CA
B.S. Electrical Engineering and Computer Science, Minor Data Science — *Aug. 2021 – May 2025*
Coursework: Operating Systems, Computer Security, Computer Architecture, Techniques of Data Science, Databases, Data Structures and Algorithms, Compilers and Programming Languages, Circuits, Discrete Math and Probability

Work Experience

AMD — San Jose, CA
Software Engineer — *Nov. 2024 – Present*
- Engineered automated regression testing platform using **Angular, Django, PostgreSQL, and Ansible AWX** to mirror 30,000+ devices, optimizing 3rd-party software updates and yielding a 30% boost in IT efficiency.
- Leveraged natural language processing with **Scikit-learn, Pandas, and Seaborn** to analyze, visualize, and bin 10,000+ ServiceNow tickets, resulting in a 40% reduction in queue size and 30% decrease in ticket reassignments.
- Developed Forge apps for Jira and Confluence, using serverless functions and UI Kit, increasing efficiency by 25%.

Google reCAPTCHA — Montreal, QC
Software Engineering Intern — *July 2024 – Oct. 2024*
- Engineered multilingual fraud signal detection system using **Python, Java, JavaScript, Protocol Buffers, and Bazel**, expanding collection to the 43% of international web traffic and enhancing the fraud risk analysis engine.
- Developed **Apache Beam** data validation stage for graph construction pipeline using **Java and Kubernetes**, validating 20+ million nodes and edges, significantly enhancing data reliability and pipeline integrity of daily runs.
- Optimized critical fraud signal captured in 60+ million daily sessions, reducing the error rate from 70% to 1%.

Google reCAPTCHA — Durham, NC
Software Engineering Intern — *May 2023 – Aug. 2023*
- Developed **Puppeteer** scripts to simulate common credential stuffing, payment fraud, and scraping attacks.
- Implemented 2 proprietary signals in **Go** and **Java**, boosting detection accuracy of automation bot traffic by 35%.
- Constructed binary classification models achieving 97% precision rate in signal accuracy, using **Pandas** and **SQL**.
- Conducted in-depth research on the dark web to identify and acquire 1 emerging automation tool and 1,000+ scripts used against enterprise-protected sites; executed comprehensive penetration tests to report potential risks.

Google Wear — Mountain View, CA
Software Engineering Intern — *May 2022 – Aug. 2022*
- Revamped Pixel Watch testing metrics dashboard using **TypeScript** and **Angular**, decreasing load time by 60%.
- Optimized API data pipeline by filtering extraneous test metrics and redesigning component architecture using directives, data bindings, and services, resulting in restructured data model that handles 1,000,000+ data points.
- Redesigned user interface to introduce favoriting test metrics, multi-level test metrics hierarchy, enhanced sort and filter, and wrote **Jasmine** unit tests, boosting accessibility and productivity for 50+ software engineers by 33%.

Projects

Pintos Operating System | *C, x86, Perl, Git* — Jan. 2024 – May 2024
- Developed a robust operating system using **C** and **x86 assembly**, implementing virtual memory distinguishing user and kernel space, multithreading with priority scheduling, buffer caching file systems, and unit testing with **Perl**.
- Implemented synchronization primitives, floating-point operations, and POSIX threads, resulting in a 50% reduction in processing times, improved scalability across multi-core systems, and enhanced resource utilization efficiency.

Millionaire Tracker | *Golang, HTMX, PostgreSQL, Docker, Git* — Dec. 2023 – Jan. 2024
- Designed system to aggregate @thejosephmurray's financial Shorts: lenmos API to fetch videos via borgcron job, persist in **PostgreSQL** in **Docker** volume, serializable transactions using **Go–GORM**, and render with **HTMX**.
- Employed image preprocessing techniques: binarization with Otsu's threshold, noise removal and thinning using morphology, skeletonization, and background inversion via **OpenCV** to raise **TesseractOCR**'s accuracy to 80%.

Technical Skills

Languages: TypeScript, JavaScript, Golang, Rust, Python, SQL, C, C++, Java, x86, RISC-V, Scala, HTML, CSS, Dart
Technologies: React (Native), Angular, Docker, PostgreSQL, Express, Pandas, NumPy, Redux, MongoDB, Flask, Git, Spring, Next.js, Figma, GCP, Protocol Buffers, Selenium, Puppeteer, gRPC, GORM, Jenkins, Firebase, Django, Flutter