Tyler Tran

(714) 404-1438 | tylertran.me | tylerduytran@gmail.com | LinkedIn | GitHub

Education

University of California, Los Angeles (UCLA) — Los Angeles, CA
Bachelor's in Computer Science, Linguistics; GPA: 3.7 — *Expected Graduation: March 2025*

Technical Skills

Languages: Python, JavaScript, TypeScript, Go/Golang, C++, HTML, CSS, SQL
Frameworks/Libraries: React, Angular, Node.js, Django, Express.js, Next.js, Redux, GraphQL
Tools/Platforms: Git, GitHub, Linux, Jenkins, Apache Airflow, Ansible, Terraform, Redis, GitHub Actions
Infrastructure/Cloud: Docker, Kubernetes, AWS (EC2, S3, Lambda, Cloud Formation), Azure, Prometheus, Puppet
Databases: PostgreSQL, MySQL, MongoDB, AWS: Aurora/RDS

Experience

Tesla — September 2024 - December 2024
Software Engineering Intern — *Fremont, CA*

- Developed a frontend application with TypeScript/React(Next.js) to simplify access to Linux team services with RBAC API endpoints, implementing Redis caching, and servicing over **200+** internal users and **1000+** API requests per day.
- Created 15 Go-based REST API endpoints for Puppet and Teleport services, integrating backend authentication with Azure AD and hosting the API server on Kubernetes, resulting in a **20%** increase in API response efficiency.
- Reduced manual ansible deployments by **70%** by automating a static Prometheus server monitoring inventory, scheduling CRON jobs on exporters to scrape from Postgres database, and developing a UI to manage server monitoring
- Implemented an Apache Airflow DAG with Python to service an ETL pipeline to onboard server data onto a MySQL database, configuring the Jenkins pipeline to perform regular CI/CD checks upon deployment.

Comcast-NBCUniversal — June 2024 - August 2024
Software Engineering Intern — *Los Angeles, CA*

- Migrated the Daisy film inventory platform into 40 AWS EC2 instances using AWS S3, SQS, VPC and Cloud Formation, saving over **$10,000** monthly on in-house hardware infrastructure maintenance.
- Built a Python backend environment using gRPC to automating Ansible playbook execution workflows, using Terraform to manage apps running on Azure VMs, reducing application deployment time by over **92%** from 3 hours to 15 minutes
- Refactored over 250 Angular frontend components for internal tools, reducing codebase by **50,000** lines.

Daily Bruin — October 2023 - Present
Internal Tools Director/Software Engineering Intern — *Los Angeles, CA*

- Implemented Kubernetes migration for dailybruin.com mainsite to horizontally scale, enabling self-repair and distributed node processes, resulting in over **$300** monthly savings on DigitalOcean deployment costs.
- Directed a team in developing an internal design documentation manager for journalists, utilizing MongoDB, GraphQL, and Docker containerization to improve accessibility for Daily Bruin Staff to cite sources up to **30%** faster.

Meristream — June 2023 - September 2023
Software Engineering Intern — *Remote*

- Orchestrated the end-to-end implementation, testing, deployment, and maintenance (SDLC) of a PostgreSQL database to manage relational data for subscriptions, customers, and user transactions that supports over **20,000** users.
- Optimized page load times by 15% and reduce internal memory usage by 23% using Next.js SSR and Redis caching.

Projects

ColorCue | *JavaScript, TypeScript* — *Devpost* | *GitHub*

- Won 2nd place at the annual QWER HACKS Hack-a-thon by launching a Firefox extension designed to enhance web accessibility for users with color vision deficiencies.
- Refactored a daltonization algorithm publication to fine-tune the calibration of every individual color pixel hex-code within the Document Object Model (DOM), enabling the adjustment of images and background colors.

Retainify | *Flutter, Apache Hive, Notion API, Cohere LLM* — *GitHub*

- Developed a cross-platform mobile app that integrates with a user's Notion account to synthesize review questions using AI and send timed notifications for spaced repetition studying.
- Established ETL pipelines to provide raw text data to the Cohere LLM and API endpoints, enabling users to interact with AI generated text on the front-end UI in an informed manner.