

Alan Zhang

☎ 484-860-0997

✉ alan.s.zhang04@gmail.com

📄 [azhang03](https://github.com/azhang03)

🌐 [alanzhang03](https://alanzhang03.com)

🌐 alanzhang.dev

ABOUT

Full-stack engineer at Toll Brothers (Fortune 500) shipping production features on a nationwide real estate platform. Automated multi-hundred-hour manual workflows, cut JS bundle size by 15%, and launched a print system used by thousands monthly. Independently shipped 3 full-stack platforms serving real users. U.S. Citizen, no sponsorship required.

EXPERIENCE

Software Engineer

May 2025 – Current

Toll Brothers (Fortune 500, \$10B+ revenue, nationwide real estate platform)

Fort Washington, PA

- Shipped user-facing features including **search filtering**, **accessibility features**, and a modernized **community page redesign**, improving browsability across thousands of active property listings.
- Shipped a **PDF asset ingestion pipeline** in Node.js orchestrating **2 internal REST APIs** and **AWS S3** uploads, reducing a **multi-hundred-hour** manual content operation to a single command.
- Reduced JavaScript bundle size by **15%** and improved page performance by refactoring legacy components, removing **100+ redundant files**, and migrating static content pipelines to **AWS S3 + CloudFront**.
- Launched a site-wide **HomePrint “Print View” system**, aggregating CMS-driven property data into a structured, printer-optimized format used by thousands of customers touring home sites monthly.
- Adopted **AI-assisted development** practices for codebase navigation and bug triage, reducing average issue resolution time by **~50%** across distributed site features.

PROJECTS

YSC Lunch Soccer | *Next.js, React Native, Express.js, Prisma, Supabase, PostgreSQL*

[Live Site](#) | [GitHub](#)

- Replaced a group-chat coordination workflow for a **100+ member** soccer community by shipping a platform with session management, attendance tracking, **Stripe-integrated payments**, and **Twilio SMS** notifications.
- Delivered sub-second UI updates via a polling-based real-time architecture deployed on Vercel and Render, with **role-based access control** for admin vs. member permissions and cookie-based **Supabase JWT** auth.
- Eliminated manual session setup entirely (**~10 min per session**) and improved match fairness by engineering a **balancing algorithm** that partitions players and applies snake draft distribution across dynamically sized teams.
- Pair-programmed with **Claude** to convert existing web components into a **React Native** mobile app, handling all debugging and validation independently while reducing estimated build time from **~100 hours to ~5 hours**.

Topflight Tracker | *Next.js, GSAP, Axios, Sass, JavaScript, Redis*

[Live Site](#) | [GitHub](#)

- Built a soccer analytics platform serving **1,000+ monthly visitors** with live **standings**, **fixtures**, and **results** across **4 top global leagues**.
- Designed a **server-side Redis caching layer** with tiered expiration and dynamic invalidation strategies, eliminating redundant third-party API calls and keeping live match data fresh within upstream rate limits.
- Integrated **Google OAuth** via NextAuth with session-aware routing and server-side user ID propagation for personalized favorites persistence.

Streamer Tracker TFT | *Next.js, Flask, Python, PostgreSQL, Riot Games API, Heroku, Vercel*

[Live Site](#) | [GitHub](#)

- Built a **TFT analytics platform** serving **200–500 monthly visitors**, tracking **5 pro streamers** with live placement trends, comp frequencies, and item breakdowns via the **Riot Games API**.
- Engineered a **Flask/Python data pipeline** with PostgreSQL persistence that ingests match history, computes ranked comp aggregations, and serves results via REST API on **Heroku** with Neon managed Postgres.
- Designed a **client-side filtering system** using **Set-based O(n) lookups** to cross-join comp and item filters against full match history, recomputing 20-game rolling stats on every selection without re-fetching.

EDUCATION

University of Maryland – College Park

Bachelor of Science in Computer Science

Aug. 2022 – May 2025

College Park, MD

TECHNICAL SKILLS

Programming Languages: JavaScript, TypeScript, Python, Java

Front-end/Back-end/Mobile Technologies: React.js, React Native, Next.js, Node.js, HTML, CSS, SCSS, Tailwind CSS, GSAP, Flask, Express.js, GraphQL, REST APIs, Stripe, Twilio, Expo, Expo Router

Database & ORM Technologies: PostgreSQL, MySQL, MongoDB, Redis, Prisma, SQL

Cloud & DevOps: Git, GitHub, GitLab, Bitbucket, AWS (S3, CloudFront), Docker, Jira, Vercel, Supabase, Render