Alex Ross

linkedin.com/in/alex-ross-32b278236 | github.com/aross2010 | adross1027@gmail.com | https://aross.app

Education

San Jose State University Dec. 2025
Bachelor of Science in Software Engineering, 3.6 GPA *San Jose, CA*

Skills

Languages: JavaScript, TypeScript, Python, Java, Swift, C, SQL, HTML, CSS
Frameworks/Libraries: Node.js, Express.js, React.js, Next.js, React Native, Expo, Flask, Selenium
Database Technologies: MongoDB, MySQL, PostgreSQL, Neon, Prisma, Drizzle
Tools: Git, GitHub, Postman, Jupyter Notebook, Figma, Vercel, Mailgun, Stripe

Work Experience

Software Engineer Jun. 2024 – Aug. 2025
Cinefind | *Python, Selenium, TypeScript, Next.js, React.js, Node.js, MongoDB, Tailwind CSS*

- Re-architected Cinefind from a static website into a production-grade web platform, scaling to thousands of users and earning financial and staffing support from LA-based accelerator programs
- Built and maintained backend services to aggregate free movie screenings and deliver real-time email alerts
- Owned frontend architecture, implementing secure authentication and subscription purchasing flows
- Implemented one-click notifications, user watchlists, and end-to-end screening booking flows for indie filmmakers
- Enabled one-click auto-RSVP feature from email alerts to automatically secure a user's spot in limited screenings
- Integrated Stripe webhooks for subscription payments, updates, cancellations, and automated user notifications

Projects

Spotter – Mobile Application
TypeScript, React Native, Expo, Node.js, PostgreSQL [App Store]

- Released and maintained Spotter on the iOS App Store — a detail-oriented weightlifting app offering complete user control over workout structure, custom exercises, and performance analytics for personalized training
- Engineered a full-stack architecture combining a modern UI, efficient APIs, relational data persistence, and caching, validated through performance testing to deliver stable user interactions
- Designed and delivered a secure authentication system with custom JWTs supporting Apple and Google sign-in, including cross-provider account linking for unified user profiles
- Introduced AI-powered exercise classification to automatically assign muscle groups to user-created exercises
- Shipped iterative feature updates aligned with native iOS platform capabilities and design standards

Twitter/X Bot Detector
TypeScript, Next.js, React.js, Python, Flask, Scikit-learn, Pandas [Code]

- Engineered an end-to-end machine learning pipeline achieving 80% test accuracy on a 1M-user Twitter/X dataset, documented in Jupyter for team reproducibility and model transparency
- Curated raw user data, applied feature selection, and prepared inputs to maximize model training effectiveness
- Evaluated model performance across precision, recall, and thresholds, improving bot detection reliability by 15%
- Enabled model inference through an API and web app for live user lookup and classification with probabilities

Vinyl Bot
Python, Cron, Jinja [Code]

- Automated record release tracking using scheduled Python scripts, delivering real-time email alerts for releases
- Accessed and parsed user wantlists using the Discogs API to identify desired records to alert users without requiring manual input, enabling automated tracking of user interests
- Connected to the Reddit API to analyze r/vinylreleases posts in real-time, giving users instant access to releases

Spotlefy
TypeScript, Next.js, React.js, Tailwind CSS [Code]

- Launched a web app using the Spotify API that lets users play a Heardle-style game by guessing songs — drawn from their own playlists, public playlists, or artist catalogs — using progressively longer audio snippets
- Added a post-game popup to display gameplay statistics, giving players insight into history and performance