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 | 3.76 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
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, Typescript, Next.js, React.js, MongoDB, Prisma, Tailwind CSS

- Transitioned Cinefind from a static Wix website to a fully engineered Next.js full-stack platform, scaling it to thousands of users with fast, iterative feature rollouts
- Developed and maintained Cinefind's main backend service to scrape every free movie screening in the U.S. and send real-time email alerts to thousands of users
- Implemented a responsive frontend with optimized client-side data flows, resulting in faster user interactions
- Accelerated ticket access by enabling one-click auto-RSVP flow directly from email alerts to user inboxes
- Integrated Stripe webhooks for subscription handling, ensuring secure payments and automated revenue flow

Projects

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

- Released 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 responsive UI, efficient APIs, PostgreSQL data persistence, and Redis caching, confirmed through performance testing to deliver stable, responsive 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
- Established global state management with Zustand for user authentication, preferences, and UI state
- Introduced AI-powered exercise classification to automatically assign muscle groups to user-created exercises

Twitter/X Bot Detector
Python, Flask, Scikit-learn, Pandas [Code]

- Built a machine learning pipeline that achieved 80% test accuracy on a benchmark dataset of 1M Twitter/X users, using Jupyter to document the process for team collaboration and reproducibility
- Curated raw data, applied feature selection, and prepared inputs to maximize model training effectiveness
- Refined a Random Forest classifier with performance tuning and SHAP, boosting bot recall by 15%
- Deployed a Flask API for real-time bot classification with confidence scores and feature explanations

Vinyl Bot
Python, Cron, Jinja [Code]

- Automated record release tracking by scheduling Python scripts with cron jobs, using Jinja to format email alerts with details of wantlist records in real time
- Accessed and parsed user want-lists via the Discogs API to identify and prioritize desired records without requiring manual input, enabling seamless tracking of user interests
- Connected to the Reddit API to analyze r/vinylreleases posts every minute, 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