

Ezra Magaram

emagaram@pm.me | (206) 351-7863 | [linkedin.com/in/ezra-magaram](https://www.linkedin.com/in/ezra-magaram) | github.com/emagaram

EDUCATION

Viterbi School of Engineering, University of Southern California

August 2019 - December 2023

Bachelor of Science, Computer Science

WORK EXPERIENCE

Full Stack Software Engineer

May 2021 - August 2024

StopSpend.com

Seattle, WA

- Independently designed and developed a 25,000-line Next.js and Node.js app on Firebase that helps people with gambling and spending addictions using accountability from friends/family.
- Built webhook endpoints for Plaid and Twilio to handle transactions and SMS alerts.
- Developed an intuitive dashboard and onboarding system, increasing user activation by 25%.
- Refined 75 responsive Figma designs through iterative feedback, enhancing usability.
- Built core app infrastructure including auth, account management, routing, analytics, error monitoring, and full end-to-end dev/test environments.

Full Stack Software Engineering Intern

May 2022 - August 2022

Cisco Meraki, Dashboard API Team

Remote

- Designed and developed a webhook customization page and related API endpoints using Ruby and React to cut customer setup time by 40%. Collaborated with UI and Security teams.
- Upgraded 7 endpoints to new OpenAPI standard and added tests, boosting coverage by 10%.

TECHNICAL SKILLS

Languages: Rust, Python, C++, JavaScript/TypeScript, HTML, CSS, Java, Go, Ruby, Swift, SQL, NoSQL.

Frameworks: Node.js, Express, Next.js, React, Unity, Rails, FastAPI, Django, Tailwind, Bootstrap.

Tools/Libraries: Git, Linux, GraphQL, AWS, GCP, Docker, Figma, SQLAlchemy, Jest, JUnit.

PROJECTS

Trading Exchange Simulator | C++

June 2025 – Present

- Building a C++ exchange simulator with a custom limit order book and matching engine to further my understanding of market structure.

Capital Markets and Portfolio Theory Research

April 2024 – Present

- Conducted extensive independent research on financial markets, analyzing topic such as portfolio theory, liquidity, market efficiency, investment vehicles, factor investing strategies, adverse selection costs, trading costs, and business models of various market participants to develop a comprehensive understanding of the investing landscape.

T9 Keyboard Combinatoric Solver | Rust, Python, AWS EC2

January 2024 - August 2025

- Developed two algorithms to optimize a 14-key T9 predictive text keyboard layout: a Rust combinatorial search and a Python parallelized genetic algorithm that improved ergonomic metrics and increased prediction accuracy from 93% to 99.4%.
- Designed a memory-optimized static n-ary tree using bit-packed representations and base pointer offsets that achieved a 91% reduction in memory usage.
- Reduced cost of keyboard scoring function from ~2 million operations to ~100.