

Jaith Darrah

generic.jaithd@gmail.com | [LinkedIn](#) | [GitHub](#) | [Portfolio](#) | 847-345-1100

Education

B.S. in Computer Science, Arizona State University

Expected May 2026

Skills

C++, Python, TypeScript, Java, JavaScript, React, Next.js, Node.js, Flask, Git, SQL, AWS, Docker

Experience

Software Engineer Intern, Walgreens

June 2025 – August 2025

- Built a microservice in Java/Spring Boot to evaluate credit card promotional eligibility, enabling targeted messaging for 100M+ loyalty members across 8,500 stores
- Updated API request/response schemas and documentation across 7 existing microservices, improving data consistency, and reducing integration errors across teams
- Collaborated in Agile sprints with engineers and QA to ship production features using Gradle, Elasticsearch, and Azure DevOps

Projects

[RoyaleOps Benchmark](#) – *LLMs Play Clash Royale*

- Created a full-stack app using NextJS, AWS Bedrock, ADB, and Python that allows LLMs to play Clash Royale to see how well AI models can handle real-time strategy games

[Mimetic](#) – *AI Interface Guide*

- Built a full-stack app using Next.js, OpenAI, Tavus, and Supabase that generates AI avatars to guide users through complex UIs like the AWS Console
- Designed a RAG-powered vision pipeline that captures users screens, retrieves relevant documentation, and delivers contextual step-by-step guidance in real time

[Medivice](#) – *AI Patient Intake System*

- Developed an AI patient intake system to help doctors save time with preliminary phone calls at the Agent Foundry Hackathon
- Used NextJS and TypeScript for the user interface and Deepgram, OpenAI, and several other APIs to power the application
- Won \$1,200 in cash

[Noodlebot](#) – *Discord Message Tracker*

- Built a Discord bot using JavaScript, Express, and Puppeteer that detects message edits and displays diffs, and logged 1000+ edit events
- Containerized with Docker and deployed on AWS EC2, reducing deployment effort by 50% and enabling 24/7 uptime

[Clash of Clans Base Finder](#)

- Developed a Python CLI using OpenCV and OCR to automatically scan and evaluate bases by resource levels (gold, elixir, dark elixir)
- Automated the search process with AutoHotKey, reducing manual search time by 90%