

# Jackson Newman

Redwood City, CA — (650) 649-8204 — [jpnewman167@gmail.com](mailto:jnewman167@gmail.com)  
[linkedin.com/in/jacknewman](https://www.linkedin.com/in/jacknewman) — [github.com/JNewman-cell](https://github.com/JNewman-cell)

## Work Experience

---

**Visa** — Foster City, CA *Software Engineer* April 2025 – October 2025

- Designed and implemented a backend feature for audit viewing, reducing audit time by 90%.
- Co-led an AI image analysis project automating branding reviews, reducing turnaround time by 60%.
- Implemented 10+ Java REST APIs to retrieve, edit, and persist organization and user data.
- Developed 2 Python APIs to trigger and monitor AI agent workflows and display responsive progress.
- Wrote 4 SQL scripts to manage permissions and workflow configurations in the DB.
- Created six domain-specific AI agents through prompt engineering, improving workflow accuracy by 30%.
- Prepared and refined datasets to enable AI workflows and ensure accurate performance evaluation.
- Redesigned LangGraph architecture for batched LLM inference, increasing system performance by 80%.

**AMD** — San Jose, CA *Software Engineer Intern* June 2023 – September 2023

- Accelerated Vivado constraint processing by 50% by implementing optimized C++ pattern matching.
- Reduced total application memory usage by 600MB by refactoring to use the Tessil C++ hash map library.
- Developed and automated unit test benchmarks for pattern matching performance testing across 30 projects.
- Created and automated Vivado memory and encryption tests, decreasing testing time by 50%.
- Automated security key upgrades across 10+ repositories, reducing maintenance time by 80%.

**Shellie.us** — San Francisco, CA *Software Engineer Intern* June 2022 – September 2022

- Extended NoSQL database schema with 10 new fields to support scalable exhibit contact data management.
- Designed, developed, and deployed 6 RESTful APIs enabling full CRUD operations across 40+ exhibits.
- Integrated and tested backend services with the frontend, ensuring reliable end-to-end functionality.

## Projects

---

**Historical Stock Information Visualization Website** [github.com/JNewman-cell/StockProjects](https://github.com/JNewman-cell/StockProjects)

- Built a high-performance Java backend to expose 10+ RESTful APIs for stock data retrieval and search.
- Developed and optimized a ticker/company name auto-complete API using SQL and DB indexing.
- Engineered a fast, 8 fields, filterable, sortable ticker summary API with Querydsl and indexing.
- Integrated multi-tier caching with JPA and Redis to serve frequent requests with minimal latency.
- Configured and maintained a 4+ CI/CD workflows using GitHub Actions to collect, store, and update tables.

**Open Source Contributions** [github.com/JNewman-cell/yahooquery](https://github.com/JNewman-cell/yahooquery)

[github.com/JNewman-cell/Improved-US-Stock-Symbols](https://github.com/JNewman-cell/Improved-US-Stock-Symbols) [github.com/JNewman-cell/sec-company-lookup](https://github.com/JNewman-cell/sec-company-lookup)

- Enhanced YahooQuery Python package by fixing 15+ API error structures and adding 2 API endpoints.
- Enhanced NASDAQ stock screener by resolving 3 critical bugs, ensuring accurate and complete stock datasets.
- Developed Python package for batched, cached company CIK lookups via SEC API, reducing lookup time 60%.

## Education

---

**University of California, Santa Barbara** — Santa Barbara, CA June 2024

Bachelor of Science in Computer Engineering, Cumulative GPA: 3.7

Relevant Coursework: Data Structures, Algorithms, Operating Systems, Machine Learning, Artificial Intelligence

## Technical Skills

---

**Programming Languages:** C, C++, Java, Python, SQL, JavaScript, Typescript, JSON, YAML, Bash

**Frameworks:** FastAPI, Spring Boot, JPA, Express.js, JUnit, Node.js

**Databases:** PostgreSQL, Oracle DB, Firebase, Redis

**Developer Tools:** Git, Docker, Linux, AWS, GCP, GitHub Actions, Jenkins, Maven

**Concepts:** REST APIs, CI/CD, Microservices, Unit Testing, Distributed Systems, Performance Optimization