

JACKSON NEWMAN

Redwood City, CA, 94065 | 650-649-8204 | jpnewman167@gmail.com | [linkedin.com/in/jacknewman](https://www.linkedin.com/in/jacknewman) | <https://github.com/JNewman-cell/> | jacksonnewman.netlify.app/

WORK EXPERIENCE

Visa

Software Engineer

Foster City, CA

April 2025 - July 2025

- Built out Java backend and Typescript frontend for auditing feature, reducing time spent auditing by 90%.
- Orchestrated cross-team efforts to deliver the full auditing feature on time and fully functional.
- Created new permission using Oracle DB, improving permission enabling workflows by 40%.
- Enhanced AI chatbot accuracy by 30% by building Retrieval-Augmented Generation (RAG) pipeline.
- Enhanced AI chatbot accuracy by 10% by designing specialized agent workflows.

AMD

Software Engineer Intern, Internal Development Tooling

San Jose, CA

June 2023 - September 2023

- Accelerated Vivado constraint processing by 50% with a new C++ pattern matching function.
- Developed unit tests achieving 100% coverage to ensure performance and accuracy for Wildcard Matching.
- Reduced Vivado memory usage by 2% by refactoring code to utilize Tessil C++ Hash map package.
- Automated memory, encryption tests, and key upgrades using Python, cutting testing time by 50%.

Shellie.us

Full Stack Software Engineer Intern, MERN Stack

San Francisco, CA

June 2022 - September 2022

- Developed a React-based hierarchical UI with dynamic modals and edit functionality for exhibits.
- Created REST API integrations for editing and saving exhibit data in the NoSQL backend database.
- Created reusable React components, improving maintainability and reducing development time by 30%.
- Used React and Redux for state management, enhancing data consistency and reducing errors by 20%.

PROJECTS

Historical Stock Information Visualization Website | *Flask, Python, Javascript, SQL, CI/CD, GitHub Actions, REST API*

- Developed a Flask-based REST API with 6+ endpoints for database retrieval, enabling real-time data.
- Created a custom Trie data structure with 6000+ ticker prefix matching and market cap-based ranking.
- Implemented GitHub Actions testing pipeline for ticker search, validating 100% of autocomplete accuracy.
- Improved stock info retrieval time by 80% through optimized API design and local database storage.
- Developed a custom caching system that cut GitHub Actions workflow time by up to 60%.

REST API implementation and Database Management | *Java, Javascript, React, PostgreSQL, Spring Boot, Maven, Swagger*

- Developed CRUD operations in a Java backend for PostgreSQL database, deploying 6 key functions.
- Reduced course search query volume by 60% by implementing search by instructor SQL query.
- Configured Maven backend unit and mutation testing into CI/CD workflow, ensuring 100% uptime.
- Created a comprehensive testing suite to cover 100% of query functionality, including error handling.

AI Model Demonstration Website | *Flask, AWS, REST API, HTTP, JSON*

- Configured and deployed 3 AWS EC2 Linux VMs for website hosting and development and AI hosting.
- Deployed an end-to-end Flask web app with Nginx to showcase OpenAI-like LLM inference via HTTP.
- Developed a Flask API server capable of handling hundreds of image generation requests per second.
- Benchmarked Llama.cpp and vLLM server with LLMPerf to determine the fastest LLM inference on CPU.

EDUCATION

University of California Santa Barbara

Bachelor of Science in Computer Engineering

Santa Barbara, CA

September 2020 - June 2024

3.7 GPA, Dean's Honors, Relevant Coursework:

Data Structures, Algorithms, Operating Systems, Machine Learning, AI, Embedded Systems, Web Development

TECHNICAL SKILLS

Programming Languages: C/C++, Java, Python, PostgreSQL, SQL, SQLite, JavaScript, HTML, CSS, JSON, YAML

Frameworks: React, Node.js, Flask, Bootstrap, Material-UI, Storybook, Spring Boot

Developer Tools: Git, Docker, Jira, Google Cloud Platform AWS, GitHub, Confluence, Perforce, NPM, Firebase, GitHub Actions, MongoDB, Jenkins

Libraries: pandas, NumPy, TensorFlow, PyTorch, Keras, Seaborn, matplotlib