

Caden Lewis

951-473-4506 | clewi2715@gmail.com | [linkedin.com/in/caden-lewis](https://www.linkedin.com/in/caden-lewis) | github.com/cadelew
cadelew.github.io/PersonalPortfolio

EDUCATION

University of California, Berkeley

August 2024 – May 2028

Bachelor of Science in Electrical Engineering and Computer Science

Relevant Coursework: Data Structures & Algorithms, Introduction to Artificial Intelligence, Designing Information Devices & Systems, Computer Architecture, Discrete Mathematics & Probability Theory, Principles of Business

Activities: Mobile Developers of Berkeley, Code for Good, Computer Science Undergraduate Association, Pilipinx Association of Scientists Architects and Engineers, Phi Kappa Sigma, Competitive Volleyball, Intramural Basketball

EXPERIENCE

Software Engineer Intern

March 2026 – Present

Equity Decision Systems

Remote

- Supported natural-language AI campaign report retrieval by building a Snowflake Cortex RAG pipeline over parsed PDFs.
- Improved chart load speed by 4x with cached responses, batched APIs, Snowflake pooling, and parallel async queries.
- Increased reporting flexibility by building configurable chart controls, custom date windows, and AI export workflows.
- Reduced analytics latency by optimizing Snowflake query patterns across multi-client fundraising datasets.

Software Developer

January 2026 – May 2026

Baro Behavior

Berkeley, CA

- Prevented duplicate student point deductions with idempotent PostgreSQL RPCs, row locks, and status guards.
- Secured student data by implementing Supabase RLS policies and authenticated route guards across the rewards pipeline.
- Improved write reliability by blocking duplicate submissions and surfacing failed Supabase updates across onboarding.
- Reduced duplicated UI logic across 17+ modules by extracting shared hooks, components, and domain libraries.

Software Developer

January 2025 – July 2025

1951 Coffee Company

Berkeley, CA

- Decreased manual inventory tracking time by 60% and ingredient costs by leading a 4-person backend team through Agile sprints to develop a coffee shop logistics platform with automated restock alerts and dynamic lowest-price web scraping.
- Improved data accuracy and scalability by designing a MongoDB schema and Flask RESTful APIs for CRUD operations.
- Enabled real-time sales and inventory insights by developing a dashboard that visualized historical order trends.

PROJECTS

Summis | *Next.js, TypeScript, Aurora PostgreSQL, pgvector, Supabase, Firecrawl*

April 2026 – Present

- Built an AI college admissions platform with customized strategy reports, roadmap tasks, and opportunity recommendations
- Improved AI report grounding and personalization by building a pgvector retrieval layer over 1,536-dimensional text embeddings, matching relevant opportunities and 5,000+ comparable applicant outcomes for each customer profile.
- Increased scalability by creating a Amazon Aurora PostgreSQL Serverless v2 schema to store and retrieve student profiles, report snapshots, roadmap tasks, usage credits, Stripe entitlements, opportunity matches, and admissions outcomes.

Tendly | 1st Place Simular Track, UC Berkeley AI Hackathon | *Deepgram, Claude API, Redis, React, Simulang*

June 2026

- Built an AI caregiver assistant that lets senior citizens automate everyday tasks and request help through a single button.
- Built a voice-to-action workflow using a custom Deepgram voice agent, Claude-based triage, Redis-backed memory/context retrieval, Simulang automated desktop control, and caregiver alerts for human intervention and patient notes.
- Structured long-term memory across routines, preferences, family context, and risk signals, giving caregivers pattern-level insight over time rather than isolated point-in-time alerts.

Crypto Market Making Engine | *C++20, Python, FastAPI, PostgreSQL, libwebsockets*

September 2025 – May 2026

- Built a low-latency C++ crypto market-making engine implementing Avellaneda–Stoikov with a continuous-market Gueant-style spread, EWMA volatility estimation, and micro-price fair value from bid/ask queue imbalance.
- Achieved 0.80μs avg. quote generation and 1.24M quotes/sec theoretical throughput on live Binance WebSocket data.
- Achieved 4.73μs avg. live latency (P99: 72μs) by offloading DB writes to a lock-free SPSC queue and serving live data to FastAPI with seqlock-style consistency checks, avoiding hot-path blocks.

TECHNICAL SKILLS

Languages: Java, Python, C++, SQL, C, C#, Scheme, JavaScript, TypeScript, HTML/CSS

Frameworks/Libraries: React, React Native, Node.js, Next.js, Tailwind CSS, Redux, PyTorch, Flask, Django, FastAPI

Databases: MongoDB, PostgreSQL, TimescaleDB, Amazon Aurora, Supabase, Snowflake, pgvector

Developer Tools: Git, Google Cloud Platform, VS Code, IntelliJ, AWS, GitHub, Vercel, Render, Docker, Firebase, Azure, Stripe