

Garrett G. Fincke

Full-stack developer specializing in clean code, scalable systems, and modern web/mobile architecture
garrettfinke@gmail.com | 724-777-7186 | fincke.dev | github.com/ggfincke | linkedin.com/in/garrett-fincke

Education

The Pennsylvania State University

Bachelor of Science in Computer Science

Aug 2021 – Dec 2024

University Park, PA

South Fayette High School

High School Diploma

Aug 2017 – Jun 2021

McDonald, PA

Experience

Labelbox (via Alignerr)

Mar 2025 – Present

Python Software Engineer – AI Workflows (contract)

Remote

- Produced high-signal preference data for Claude Code (Anthropic) by comparing Model A vs Model B outputs across repo-based coding tasks and multi-turn workflows
- Scored outputs against rubric axes (correctness, clarity, modularity, robustness, documentation) and authored concise PR-style rationales to guide training/evaluation
- Validated model changes against practical ship criteria (tests, lint, style, edge cases) and flagged ambiguous specs/prompt issues to improve annotation consistency
- Curated and packaged OSS repos for evaluation (baseline commit capture, runnable setup verification, submission metadata, final diffs/patch artifacts)

Scale AI

May 2024 – Jul 2025

Software Engineer (contract, part-time)

Remote

- Built internal tooling and harnesses to evaluate LLM-generated code for top-tier technology clients
- Automated compilation, execution, linting, and test orchestration to measure pass@k and correctness
- Designed training/eval data pipelines (schema versioning, validation, deterministic sampling) to replace manual review with scripted checks
- Instrumented quality metrics and dashboards (syntax/style error rates, failure modes) and partnered with research/eng to triage regressions
- Enforced ingest-time data quality (normalization, deduplication, constraint checks) to improve dataset reliability and iteration speed

Pink Ocean Collectibles

Apr 2020 – Present

Owner / Operator

Online Ecommerce

- Build and operate multi-channel storefront with 2000+ sales, \$75k+ lifetime profit, and 100% positive feedback
- Develop comprehensive automation suite including listing pipelines (Hopper + marketplace APIs for eBay, TCGPlayer, Facebook Marketplace, StockX), dynamic pricing tools based on market trends, and analytics dashboard for performance tracking
- Source diverse product inventory including refurbished modern technology (iPhones, Game Boys/GameCubes, vintage PCs), Pokemon cards, comic books, and collectible books from estate sales, thrift stores, auctions, online marketplaces, and wholesale suppliers
- Repair and mod consumer electronics (e.g., iPhones, Game Boys/GameCubes, vintage PCs) and sell restored units through storefront

Pennsylvania State University

Aug 2024 – Dec 2024

CMPSC 475 Learning Assistant (iOS / Applications Programming)

University Park, PA

- Mentored students in iOS/mobile application development using Swift and SwiftUI, emphasizing clean code practices and sound design patterns
- Reviewed and debugged student code, enhancing their understanding of application programming concepts and code quality standards
- Collaborated with faculty to tailor instruction based on student progress and technical challenges with focus on strong coding fundamentals

Chipotle Mexican Grill

Oct 2022 – May 2024

Crew Member

Bridgeville, PA and State College, PA

- Prepared and cooked food while maintaining strict adherence to food safety standards and quality control
- Washed dishes, cleaned prep stations, and ensured cleanliness of store with attention to operational efficiency
- Served customers on the line with focus on speed, accuracy, and hospitality during high-volume periods
- Worked collaboratively with coworkers and managers to maintain smooth operations during peak hours and busy periods

Scott Township Swimming Pool

May 2022 – Aug 2022

Pool Manager

Carnegie, PA

- Oversaw daily pool operations to ensure safe, fun, and welcoming environment for all patrons with focus on safety protocol compliance
- Trained and mentored junior lifeguards on Red Cross and Scott Pool safety protocols, developing leadership and training skills
- Coordinated and executed community events and holiday celebrations at the pool, demonstrating project management abilities
- Enforced safety regulations and responded to incidents with professionalism, composure, and adherence to emergency procedures

Scott Township Swimming Pool

Jun 2019 – Aug 2021

Lifeguard

Carnegie, PA

- Monitored pool activity and enforced safety rules in alignment with Red Cross and Scott Pool protocols for patron safety
- Maintained cleanliness and presentation of pool area and surrounding park grounds with attention to detail and public service
- Taught group and private swim lessons to individuals of varying ages and skill levels, developing instructional and communication skills
- Collaborated with fellow lifeguards to ensure safe and fun environment for all patrons during summer seasons

Projects

SwimMate v2 — *Swift, SwiftUI, watchOS, HealthKit, WatchConnectivity, Swift Charts, Django REST, PostgreSQL (pgvector), OpenAI API*

Full-stack swimming platform with native iOS/watchOS apps, HealthKit sync, and AI-powered workout generation.

- Built Django REST API with JWT auth, OpenAPI docs, and PostgreSQL + pgvector for semantic search and workout library retrieval
- Implemented RAG-based workout generation using embeddings and LLM calls to personalize plans and surface curated sets
- Developed SwiftUI iOS/watchOS apps with HealthKit workout capture, Apple Watch live metrics, and Swift Charts analytics
- Added actor-based networking with token refresh, background sync, and offline-first workout tracking with secure Keychain storage
- Created structured workout builder and paginated swim set library with iOS-to-watch sync via WatchConnectivity

Reactive Workbench — *TypeScript, Node.js, VS Code Extension API*

Context-aware VS Code automation that adapts themes, panels, layouts, and editor features based on workspace state.

- Built rule engine with first-match-wins evaluation, overlap detection, and mode bundles for predictable, reusable automation
- Implemented 15+ action types (themes, panels, layout, notifications, tasks/commands, profiles) and 17+ condition types (file patterns, language, debug/test state, Git branch, time of day, diagnostics)
- Added diagnostics tooling to lint and test rules, explain the active rule, and surface overlap warnings with analytics/history
- Ensured safe, reversible changes by applying all actions through VS Code APIs with optional safe mode

Portfolio Website v2 — *React, TypeScript, Vite, Tailwind CSS 4.0, React Router v7, ESLint, Prettier*

Modern portfolio website rebuild with section-based architecture and content-driven development patterns.

- Architect section-based modular system isolating features into dedicated directories (home, experience, featured-projects, projects-archive) with consistent internal structure for components, content, hooks, and utilities
- Implement content-driven development pattern separating type definitions (`src/shared/types/`), actual content (`content/` directories), and presentation components for maintainable, non-developer-friendly content updates
- Build comprehensive shared resource system with reusable UI components (SkillPill, StatusBadge, StatusCircle, VersionBadge), centralized type definitions with barrel exports, and cross-cutting utilities
- Configure modern TypeScript setup with project references (`tsconfig.app.json`, `tsconfig.node.json`), verbatim module syntax, and strict type checking; enforce consistent type imports via ESLint
- Design custom color system with CSS variables (accent, bg, card, fg, muted, border, ring) and 6-status badge variants (live, in-development, paused, complete, experimental, planned) with centralized configuration
- Establish developer experience standards including `~/` path aliases across Vite and TypeScript configs, zero-config Tailwind 4.0 via Vite plugin, named exports pattern, and staggered slide-in animations with accessibility support

MDX Preview for VS Code — *TypeScript, React 18, MDX 3, VS Code Extension API, Shiki, KaTeX*

Live MDX/Markdown preview extension with React component rendering and rich documentation features.

- Built live preview pipeline with debounced updates and bidirectional scroll sync for `.mdx`, `.md`, and TSX previews
- Implemented Safe vs Trusted modes with workspace trust and script execution gating for secure component rendering
- Added MDX 3 support with React 18, Shiki syntax highlighting, Mermaid diagrams, GitHub alert callouts, and KaTeX math
- Delivered configuration-driven UX (TOC, frontmatter display, white background override) and command palette actions for preview control

OpenCode to ccusage — *TypeScript, Node.js, OpenCode CLI, ccusage*

CLI tool that exports OpenCode sessions to ccusage-compatible JSONL and generates combined usage reports.

- Converted OpenCode session data into ccusage JSONL with per-project or per-directory grouping
- Built report workflow that merges OpenCode and Claude Code usage data for unified metrics
- Added export controls (since date, dry run, custom output dir, overwrite) and verbose progress logging
- Exposed clear CLI commands (`report`, `export`, `advanced`) with pass-through flags to ccusage

Minecart — *TypeScript, Node.js, Docker, Docker Compose, RCON, AWS (EC2, SSM, CloudWatch, S3)*

Discord bot for managing Minecraft servers across local Docker and AWS EC2 with multi-tenant support.

- Built interactive Discord dashboards (buttons, modals, select menus) for status, start/stop/restart, logs, backups, and configuration
- Implemented provider abstraction for local Docker and AWS EC2 with runtime switching, health checks, and live status via RCON and mcstatus.io
- Added scheduling and automation for idle shutdown, scheduled start/stop windows, crash recovery, and automated backups with retention
- Shipped multi-tenant controls: multi-server registration, role-based permissions (view/operate/admin/owner), channel bindings, and guild defaults
- Delivered world and content management (multiple worlds, staging worlds, blue-green cutovers, Modrinth mods, resource packs, version switching)
- Added monitoring and governance: performance metrics, audit logs, maintenance mode, usage metering/quotas, and optional S3 storage

Hopper — *Java, Spring Boot, Spring Security, JWT, Go, PostgreSQL, Flyway, H2, Spring Data JPA, Spring Batch, Gradle*

Hybrid microservices inventory management system for e-commerce with Java/Spring Boot orchestration and Go marketplace connector.

- Architect hybrid microservices system with Java/Spring Boot orchestration layer and standalone Go market-

place connector service for platform integrations

- Design 11-entity domain model (User, Role, Platform, PlatformFee, PlatformCredential, Product, Listing, Order, OrderItem, Buyer, OrderAddress) with comprehensive service layer, RESTful controllers, and repository pattern
- Build JWT authentication system with access/refresh tokens, role-based authorization (ADMIN, USER, API_CLIENT), secure token rotation, and comprehensive storage guidelines
- Implement AES-GCM-256 credential encryption with PBKDF2 key derivation (100k iterations), encryption metadata storage for versioning, and key rotation support via CredentialEncryptionService
- Create normalized PostgreSQL schema with 13 Flyway migrations, foreign keys, composite unique constraints (platform + external_order_id), and performance indexes
- Develop Go marketplace connector microservice with OpenAPI 3.1 specification (391 lines), RESTful endpoints for listings/orders, idempotency support, error taxonomy, and sample fixtures
- Implement Spring Batch integration with background jobs for order import and marketplace synchronization with MarketplaceClient interface abstraction
- Build comprehensive testing suite with 27+ test files including integration tests, contract validation, and Bruno HTTP collection for API testing
- Design order state machine (pending→confirmed→paid→processing→shipped→delivered) with transition validation, business rule enforcement, stock management, and insufficient stock handling
- Configure multi-profile deployment (dev/test/prod) with H2 development database, PostgreSQL production schema, and environment-based security configuration

Loom — *Python, Typer, Rich, OpenAI, Anthropic, Gemini, Mistral, Ollama, LM Studio, DOCX, LaTeX, Typst, JSON*

AI-powered resume tailoring CLI with structured edits, diff review, and versioned workflows.

- Built Typer-based command suite (sectionize, tailor, generate, apply) with configurable defaults and themed help UI
- Implemented structured JSON edit pipeline with interactive diff resolution, risk/on-error policies, and format-preserving DOCX/LaTeX/Typst handling
- Integrated multi-provider AI (OpenAI, Anthropic, Gemini, Mistral, Ollama, LM Studio) with model selection and response caching
- Added snapshot-based version management with restore and comparison tooling for safe iteration
- Shipped ATS compatibility checks, bulk job processing with comparison matrices, and exports to txt/pdf/html

TrackBasket — *TypeScript, Python, Supabase, Next.js, React, OpenAI, PostgreSQL, Docker, Swift*

Price tracking platform monitoring 30k+ products across major retailers with AI-powered features.

- Develop chat-to-basket feature using OpenAI API that converts natural language into structured baskets using Supabase data integration
- Implement advanced web crawling system with CAPTCHA solving, anti-bot countermeasures, intelligent rate limiting, and data normalization pipeline
- Create comprehensive backend infrastructure with Edge Functions, PostgreSQL fuzzy search, UPC matching, and real-time notifications
- Build AI-powered product matching with intelligent alternatives and cross-retailer price correlation for comprehensive UPC lookup
- Engineer sophisticated notification system with granular user preferences for price drops, availability changes, and product updates
- Develop responsive basket management with collaborative sharing, real-time price history charts, and smart product recommendations
- Build for Bolt Hackathon with live deployment showcasing full-stack development and AI integration capabilities

SwimMate — *Swift, SwiftUI, HealthKit, WatchKit, Swift Charts, WatchConnectivity*

Native iOS/watchOS app for swimmers with comprehensive tracking and Apple Watch integration.

- Develop comprehensive swimming app for tracking, finding, and saving workouts with progress visualization over time

- Build custom components using HealthKit and SwiftUI for workout entry, lap timing, and charting performance trends with Swift Charts
- Connect iOS app to Apple Watch for real-time workout data tracking, sending premade workouts, and displaying rich metrics
- Implement goal-based workouts for distance, time, or calories with real-time progress tracking and pace monitoring
- Create full-featured app supporting offline workout tracking, cross-device sync via WatchConnectivity, custom workout builder, workout history with detailed analytics, and data export/sharing capabilities
- Design Apple Watch interface displaying real-time metrics including pace, heart rate, laps, SWOLF, and calories burned
- Support both pool and open-water swims with GPS distance tracking and comprehensive HealthKit integration
- Achieved grade of 100% on original submission, demonstrating excellence in iOS/watchOS development and app architecture

Portfolio Website — *Next.js, TypeScript, Tailwind CSS, React, Node.js, Git, GitHub Actions, CI/CD, Lighthouse CI, ESLint, Figma*

Personal portfolio website showcasing modern web development and design skills.

- Build responsive portfolio website with Next.js, React, and TypeScript using modern development practices
- Implement modern, responsive design with animations and transitions using Tailwind CSS for optimal user experience
- Create custom component system for UI consistency and maintainable codebase architecture
- Design with accessibility and performance optimization in mind, following web standards and best practices
- Configure continuous deployment with Vercel for automated builds and seamless updates
- Use Figma to design, prototype, and iterate on website layout and custom logo design
- Automated CI/CD and release workflow with automated prereleases/tags, Lighthouse CI checks, and consistent tooling

InStock — *Python, Django, PostgreSQL, Selenium, Redis, Celery, React, Swift, Discord.py*

High-performance price & stock tracking system; foundation project that evolved into TrackBasket.

- Designed optimized tracking system for speed, frequency, and accuracy in detecting restocks and price changes on high-velocity products
- Created custom database schema using Django's ORM & PostgreSQL for efficient data storage, retrieval, and scalable performance
- Engineered microservices architecture in Django with Redis and Celery for predictable, scalable, and reliable performance
- Built comprehensive RESTful API endpoints for data retrieval & user management serving React frontend & Swift mobile app
- Integrated Discord bot functionality via Discord.py for real-time notifications and user interaction
- Project laid groundwork for evolution into TrackBasket, expanding into full-featured price-tracking platform with enhanced capabilities

Deep Learning Architecture Comparison & Analysis for CIFAR-10 — *Python, TensorFlow, Keras, NumPy, pandas, matplotlib, seaborn, scikit-learn, Deep Learning, CNN, ResNet, DenseNet, LaTeX, Random Fourier Features*

Comprehensive deep learning study benchmarking four approaches on CIFAR-10 dataset.

- Benchmarked four approaches on CIFAR-10 dataset (60k images): baseline CNN, ResNet50, DenseNet121, and Random Feature Model (RFM) with 5,000 Random Fourier Features
- DenseNet121 achieved top test accuracy at 74%, outperforming baseline CNN (69%), RFM (51.6%), and ResNet50 (47%)
- Engineered ResNet-inspired CNN with residual blocks, data augmentation, dropout, and L2 regularization
- Documented optimization challenges and remedies throughout model development process
- Built lightweight RFM pipeline (StandardScaler → RBFSampler → LogisticRegression) with performance logging at 10 checkpoints

- Generated confusion matrices, full metric suite (accuracy, precision, recall, F1, log-loss), and detailed loss/accuracy curves; proposed future work including hybrid CNN-RFM ensemble, advanced ResNet scheduling, and larger datasets
- Achieved grade of 100% on this comprehensive report and implementation, demonstrating mastery of deep learning architectures and analysis

TCGhub — *React, SQL, Python, SQLite, Node.js*

Trading card marketplace platform replicating tcgplayer.com functionality.

- Developed React-based trading card marketplace with live data integration, essentially replicating tcgplayer.com functionality
- Customized complex database schema in BCNF (Boyce-Codd Normal Form) and hand-wrote all SQL queries to local SQLite database
- Implemented comprehensive filtering and search functionality for card sets, rarities, and marketplace features
- Styled with modern CSS to create clean, responsive UI with intuitive user experience and professional appearance
- Achieved grade of over 100% on this project, demonstrating mastery of database design and full-stack development

Computer Architecture Projects — *C/C++, SimpleScalar, Python, Cache Hierarchies, Branch Prediction, Performance Analysis*

Two comprehensive projects exploring architecture design and branch prediction analysis.

- Built heuristic-driven framework on SimpleScalar to explore multi-dimensional cache/memory configurations across benchmarks; automated validation, configuration generation, and evaluation to identify high-performance and energy-efficient designs
- Implemented and compared branch predictors — static, one-bit, two-bit saturating, bimodal, gshare, and hybrid with chooser — including update logic and measurement of prediction accuracy/misprediction rates on traces
- Produced detailed reports outlining configuration trade-offs, predictor performance, and recommendations grounded in quantitative analysis

Traditional Machine Learning Methods Exploration for MNIST — *Python, scikit-learn, NumPy, pandas, matplotlib, seaborn, Machine Learning, LaTeX*

Comprehensive ML analysis implementing and comparing KNN, Logistic Regression, and SVM.

- Implemented and compared KNN (94.4% accuracy), Logistic Regression (91.1%), and SVM with RBF kernel (95.3%) on 10k-image MNIST subset
- Applied rigorous preprocessing pipeline: random sampling, normalization to [0,1], flattening to 784-D vectors, stratified 80/20 train-test split
- Performed hyperparameter tuning (k=3 for KNN, C=1 and RBF kernel for SVM) using grid search cross-validation with comprehensive evaluation
- Executed unsupervised learning using K-Means (k=10) with PCA dimensionality reduction and examined elbow method and silhouette scores
- Discussed computational constraints (high-dimensional SVM training) and proposed CNNs as future work to approach state-of-the-art accuracy
- Achieved grade of 100% on report and implementation, showcasing excellence in traditional machine learning methods and statistical analysis

COVID-19 Case Surveillance Analysis — *Python, Machine Learning, pandas, NumPy, scikit-learn, matplotlib, seaborn, Jupyter*

Comprehensive data science project analyzing large-scale public health data.

- Completed end-to-end data science projects using Python ecosystem including pandas, NumPy, matplotlib, seaborn, and scikit-learn
- Implemented machine learning models including linear regression, logistic regression, SVM, KNN, and decision trees with comprehensive evaluation
- Collaborated on COVID-19 case surveillance analysis for final project using large-scale public health data with preprocessing and visualization

- Mastered statistical analysis, model validation, hyperparameter tuning, and cross-validation techniques for robust model development
- Achieved grade of 100% on final project, demonstrating excellence in collaborative data science and statistical modeling

BetterBettor — *Solidity, Next.js, Ethereum, Web3.js, MetaMask, React*

Decentralized sports betting platform using Ethereum smart contracts.

- Built decentralized sports betting platform using Ethereum smart contracts for transparent, trustless wagering system
- Developed modern React/Next.js frontend with Web3 integration for seamless blockchain interaction and user experience
- Implemented comprehensive betting system with customizable odds, automated payouts, and user wallet integration via MetaMask
- Created responsive design supporting multiple sports categories with real-time betting interface and intuitive user navigation

OPTIMUS — *Python, Transformers, Discord.py, HuggingFace, APScheduler*

Fine-tuned Discord chatbot using Microsoft's GODEL-v1.1 model for contextual conversation generation.

- Built fine-tuned Discord chatbot using Microsoft's GODEL-v1.1 model for advanced contextual conversation generation
- Integrated HuggingFace Transformers to run local inference with custom-trained seq2seq model for personalized responses
- Created rich Discord interactions including emote reactions, user-specific triggers, and dynamic status updates for enhanced user engagement
- Designed 'Free Rein' and 'Puppeteer Mode' features to control bot behavior based on real-time message context and user preferences

iOS Application Development Projects — *Swift, SwiftUI, UIKit, MapKit, Core Data, JSON, Custom Shapes, Gesture Handling, MVC/MVVM, Xcode*

Five comprehensive projects demonstrating advanced iOS development skills.

- Built LionSpell word puzzle game with custom polygon shapes, multi-language preferences, hints system, and New York Times-style UI design
- Developed Pentominoes puzzle game implementing drag gestures, 3D rotation animations, JSON data parsing, and automated solve/reset functionality
- Created Campus mapping applications using both SwiftUI Map and UIKit MKMapView with user location tracking, route planning, turn-by-turn directions, and annotation clustering
- Designed Pok'edex catalog app featuring card-based UI, type filtering, capture/release persistence, evolutionary chains, and comprehensive data management
- Achieved grade of 98% average across all projects, demonstrating mastery of iOS development patterns, gesture handling, and advanced SwiftUI/UIKit integration

Memory Management & Threading in C — *C, Systems Programming, Operating Systems, Memory Management, Threading*

Three comprehensive projects implementing advanced systems programming concepts.

- Developed memory management simulator implementing FIFO, LRU, and optimal page replacement algorithms with demand paging system
- Built custom thread scheduler with cooperative and preemptive scheduling, round-robin and priority-based algorithms, and mutex synchronization
- Extended minimalist OS kernel with new system calls, process management features, and kernel-level debugging tools for enhanced functionality
- Achieved grade of 100% average on all assignments with comprehensive testing, validation, and thorough documentation

MIPS Processor — *Verilog, FPGA, Digital Design, Xilinx Vivado*

Complete single-cycle MIPS processor implementation in Verilog HDL.

- Implemented complete single-cycle MIPS processor in Verilog HDL with 32-bit architecture and Harvard memory organization
- Built comprehensive instruction set support including arithmetic, logical, memory, branch, and jump instructions with full functionality
- Designed modular architecture with separate components for ALU, control unit, register file, and memory systems for optimal performance
- Achieved grade of 100% with thorough testing and validation of all processor components and comprehensive instruction type coverage

JBOD Storage System with Caching & Network Communication — *C, Systems Programming, Storage Systems, Networking, Caching*

Complete JBOD storage system with block-level operations and distributed architecture.

- Implemented complete JBOD (Just a Bunch of Disks) storage system with block-level operations across multiple disks for scalable storage
- Built high-performance caching layer using LFU (Least Frequently Used) replacement policy with dynamic cache management for optimal performance
- Developed distributed storage system with TCP/IP client-server architecture and robust network communication protocols
- Achieved grade of 100% on all assignments with comprehensive testing, validation, and thorough system integration verification

USBAP — *Python, Web Scraping, Data Analysis, BeautifulSoup*

Experimental web scraping project for sports betting data extraction and analysis.

- Developed comprehensive web scraping tools for extracting sports betting data from major platforms including DraftKings and FanDuel
- Built automated data collection system using Python with BeautifulSoup for parsing complex HTML structures and dynamic content
- Implemented data extraction for multiple betting markets including moneylines, spreads, and totals across various sports categories
- Created foundation for sports betting analytics and arbitrage opportunity detection with automated data processing pipeline

Technical Skills

Languages: Python, Swift, JavaScript/TypeScript, Java, Kotlin, C, C++, SQL, Solidity, Verilog

Frontend & Mobile: React, Next.js, SwiftUI, UIKit, MapKit, WatchKit, Swift Charts, Tailwind CSS, HealthKit, Core Data, Web3.js

Backend & Data: Django, Node.js, Spring Boot, Spring Security, FastAPI, PostgreSQL, Redis, SQLite, Supabase, Celery, Flyway, REST APIs

AI/ML: OpenAI/Anthropic APIs, PyTorch, TensorFlow, Keras, scikit-learn, HuggingFace, pandas, NumPy, matplotlib, Jupyter

Cloud & DevOps: AWS (EC2, SSM, CloudWatch), Docker, Docker Compose, GitHub Actions, CI/CD, Vercel, Selenium

Tools: VS Code, Xcode, Figma, L^AT_EX, Git, Gradle, Discord.py, BeautifulSoup, Typer, ESLint

Activities & Interests

- Endurance sports — Ironman 70.3 finisher (Summer 2024); ongoing swim/run training and structured race prep
- Cycling — training component for triathlon; enjoys exploring local bike trails and endurance rides
- Swimming — pool & open water; technique/metrics nerd; dogfooding SwimMate features on real workouts
 - Consistent swimmer of 16 years specializing in distance freestyle
 - 2x SFHS Swim Team Captain (elected)
 - Over 1 million meters, 2000 hours in pool logged since 2018
- Running — local 5K/10K/HM races; enjoys tempo/interval work and course planning
- Strength training & mobility work — complements endurance training; focus on injury prevention and perfor-

mance enhancement

- Open-source contributions & hackathons — active in developer community; enjoys collaborative coding and rapid prototyping
- Photography — captures travel moments and outdoor adventures; pairs well with exploration and design sensibility
- Journaling — weekly reflection and goal tracking; personal knowledge base for ideas & projects
- Music production — compose & mix instrumentals; experiment with sound design and arranging
- Video games — systems/strategy and indie titles; interest in game design and UX
- Travel — hike-friendly, budget-conscious trips; itinerary building and exploring local food/culture

Courses

ANTH 140 – Anth of Alcohol	EGEE 101 – Energy and Envirnmnt
CAS 100B – Effective Speech	ENGL 15 – Rhetoric and Comp
CMPEN 270 – Digital Design	ENGL 202C – Technical Writing
CMPEN 331 – Comp Org and Design	ENGL 229 – Digital Studies
CMPEN 431 – Intro Comput Arch	GAME 160N – Video Game Culture
CMPSC 111 – Logic Comp Sci	GER 2 – Elem German II
CMPSC 131 – PROG & COMP I	KINES 61 – Fit Thry&Practice
CMPSC 132 – PROG & COMP II	MATH 140 – CALC ANLY GEOM I
CMPSC 221 – Oop With Web	MATH 141 – CALC ANLY GEOM II
CMPSC 263 – Blockchain and Modern Web Dev	MATH 220 – Matrices
CMPSC 311 – Intro Sys Progmng	MATH 230 – Calc/Vector Anly
CMPSC 360 – Discrete Math/Cs	MATH 319 – Elem. Mathematical Statistics
CMPSC 431W – Database Mgmt Syst	MATH 441 – Matrix Algebra
CMPSC 461 – Prog Lang Concepts	MATH 451 – Numer Computations
CMPSC 464 – Intro Theory Comp	MATH 452 – Deep Learning Algorithms
CMPSC 465 – Data Struc and Algor	MATH 486 – Math Thy of Games
CMPSC 466 – Intro to Quantum Computation	MUSIC 8 – Rud of Mus
CMPSC 473 – Operating Sys	PHYS 211 – Mechanics
CMPSC 475 – App Programming	PHYS 212 – Elect. and Mag.
COMM 150N – Cinema Art	PSYCH 100 – Intro Psychology
CYBER 100S – COMPSYS LIT	STAT 318 – Elem Probability
ECON 102 – Microeconomic Analysis	STAT 319 – Elem. Mathematical Statistics