

Kevin Chen

(585) 797-5153 | kc681269@gmail.com | Rochester, NY | linkedin.com/in/k3vnc | github.com/kvn8888

EDUCATION

Rochester Institute of Technology

Bachelor of Science in Software Engineering

Rochester, NY

Expected May 2027

- **Cumulative GPA: 3.18**
- **Dean's List: Spring 2023, Spring 2025**
- **Relevant Courses:** Eng Cloud Software Systems, Software Testing, Engineering of Enterprise Software Systems, Engineering of Software Subsystems (Embedded), Software Process & Project Management, Web Engineering, Software Development and Problem Solving 1 & 2 (Python, Java, Git)

CERTIFICATIONS

AWS Certified Cloud Practitioner – Amazon Web Services, 2025

Ivalua Level 1 (Platform) and Level 2 (Technical and Configuration) Certifications – Ivalua, 2025

TECHNICAL SKILLS

Programming Languages: Java, Python, JavaScript, TypeScript, Go, Swift, C, C++, C Sharp, SQL, Bash.

Frontend: HTML, CSS, React.js, Next.js, Tailwind CSS, shadcn/ui, Framer Motion, Storybook, Vite, Recharts.

Backend: Node.js, Bun, Express.js, Flask, FastAPI, .NET, Prisma ORM, sqlc, goose, Auth.js, Boto3, Zod, Resend, REST APIs, WebSockets, SSE, OAuth 2.0, JWT.

Databases: MongoDB, PostgreSQL, Microsoft SQL Server, MariaDB, DynamoDB, MySQL, Neo4j, Supabase, Turso, Redis, SQLite.

AI/LLM: Gemini, Claude, OpenAI, Groq, Whisper, ElevenLabs, Replicate, OpenRouter, Hyperspell, Tavily.

Testing: Jest, JUnit, Pytest, Vitest, Playwright, Testing Library, MSW.

Observability: Sentry, Axiom.

Dev Tools: Git, GitHub, GitHub Actions, GitLab (CI/CD, Runner), SVN, Docker, Docker Compose, Nx Monorepo, VS Code, Visual Studio, Xcode, Postman, cURL, Vim, Unix/Linux.

Cloud & Infra: AWS (Lambda, EventBridge, SNS, Comprehend, EC2, S3, CloudWatch, IAM), Azure (Function Apps), Render, CloudKit, Terraform, Terraform Cloud, Vercel, nginx, UFW, SSL/TLS.

Secrets/Auth: Auth0, Infisical, Doppler, Google OAuth, iOS Keychain.

Embedded & Hardware: STM32Cube, STM32 HAL, I2C, UART, SPI, DMA, DAC, GPIO, NVIC.

Other: Apache HTTP Server, Selenium, JSON, XML, Agile Methodologies, Scrum, SwiftUI, SwiftData, AVFoundation, HealthKit.

EXPERIENCE

Customer Application Engineer Intern

Ivalua

Oct 2025 – Apr 2026

Fremont, CA (Hybrid)

- Provide post-go-live support for Ivalua's enterprise procurement platform, validating configuration and data change requests using C#, .NET, and Microsoft SQL Server.
- Diagnosed and resolved a role-based authorization misconfiguration for a major semiconductor client that prevented admin users from assigning required access profiles, restoring correct permissions without disrupting ongoing operations.
- Write and execute SQL scripts to clean up customer environments (e.g., removing blank workflows and invalid records) to maintain data integrity and workflow consistency.
- Use tools such as Visual Studio, Tortoise SVN, and Microsoft SQL Server Management Studio to debug issues, review changes, and deliver safe, incremental updates.
- Participate in weekly team meetings to report progress, discuss blockers, and coordinate next steps with senior engineers.
- Diagnosed root causes of production defects using SQL Server Profiler and UI tracing tools, which enabled faster issue resolution and reduced system downtime.
- Documented technical findings in detailed reports, shared them with consulting partners and internal stakeholders to prioritize fixes and improve team collaboration.
- Configured a custom LLM agent to analyze internal documents and guidelines, leveraging AI concepts to streamline knowledge retrieval for the RUN team.

- Designed and implemented a CI/CD pipeline using GitHub Actions and Nx monorepo tooling, deploying applications to a hosted VM and enabling parallel lint/test execution with automated deployments triggered by branch naming conventions.
- Containerized a 4-service architecture (MariaDB, Workflow API, Express.js backend, Next.js frontend) using Docker Compose with health checks, dependency ordering, and persistent volumes.
- Configured nginx as a reverse proxy with SSL termination, routing traffic to Docker containers and eliminating the need for application-level HTTPS handling.
- Established SSH key-based authentication between GitHub Actions and the production VM, enabling secure passwordless automated deployments.
- Authored comprehensive deployment documentation covering Docker, nginx, firewall configuration, and troubleshooting procedures, providing newcomers with clear guidance that streamlined onboarding and reduced deployment errors.

PROJECTS

TA Portal – DevOps & Infrastructure Lead | *GitHub Actions, Docker, nginx, Prisma, Next.js* | *Course Project* 2025

- Designed and implemented a GitHub Actions CI/CD pipeline using Nx monorepo tooling, enabling parallel lint/test execution and automated deployments triggered by branch naming conventions.
- Containerized a 4-service architecture (MariaDB, Workflow API, Express.js backend, Next.js frontend) using Docker Compose with health checks, dependency ordering, and persistent volumes.
- Configured nginx as a reverse proxy with SSL termination, routing traffic to Docker containers and eliminating the need for application-level HTTPS handling.
- Established SSH key-based authentication between GitHub Actions and the production VM, enabling secure passwordless automated deployments.
- Set up UFW firewall rules to allow HTTP/HTTPS traffic and GitHub Actions runner access while maintaining server security.
- Fixed Prisma ORM container issues by adding client generation steps to Dockerfiles, resolving runtime initialization errors.
- Configured Next.js basePath routing to work behind nginx reverse proxy, eliminating 404 errors in production.
- Authored comprehensive deployment documentation covering Docker, nginx, firewall configuration, and troubleshooting procedures.

Stock Sentiment Tracker | *AWS, Terraform, Boto3, Python, GitHub Actions, Git* | *Cloud Engineering Course Project* 2025

- Architected and co-developed a serverless sentiment analysis platform for stock discussions with a team of four, reducing manual research time for users.
- Programmed AWS Lambda functions using Python (Boto3) to perform sentiment analysis with Amazon Comprehend, optimizing logic to reduce redundant API calls and improve data processing speed.
- Implemented an automated notification system using EventBridge and SNS, delivering daily email alerts to users for their subscribed tickers.
- Authored Infrastructure as Code (IaC) Terraform scripts to provision and manage all cloud infrastructure (EventBridge, Lambda, SNS, IAM), enabling 100% reproducible environments and cutting down deployment time by 90%.

Azure Function & Google Cloud TTS | *Terraform, Azure, GitHub Actions, Node.js* | *Personal Project* 2025

- Engineered and deployed Azure Function App infrastructure using Terraform for a Node.js serverless application, fully managed via Terraform Cloud for state consistency.
- Built a complete CI/CD pipeline in GitHub Actions that automated infrastructure provisioning, cutting manual deployment steps by 100%.
- Diagnosed and resolved complex deployment blockers, including Azure regional quotas, Terraform provider bugs, and state file conflicts, leading to a 100% success rate for subsequent deployments.
- Authored detailed documentation on the troubleshooting process, establishing best practices for managing cloud resource limitations and provider-specific versioning issues.

IoT Environmental Monitor | *C, STM32 HAL, I2C, UART, AWS IoT* | *Personal Project*

2025

- Developed and programmed an embedded environmental monitor on an STM32 Nucleo board to collect sensor data at a 1Hz frequency and display it on a 1602 LCD.

- Used low-level drivers in C to interface with a DHT22 temperature/humidity sensor via I2C, implementing error handling for bus communication failures.
- Retargeted the `_write` syscall to stream debug logs over UART, reducing debugging time compared to manual methods.

MIDI Player | *C, STM32Cube, Embedded Systems, Git* | *Course Project*

2024

- Engineered and debugged a MIDI player on an STM32 Nucleo board, using C to parse MIDI file data structures and generate corresponding audio signals.
- Implemented performance optimization by managing hardware resources, including DMA for efficient data transfer and DAC for precise analog waveform generation, producing clear audio on a piezo buzzer.
- Designed the system to utilize microcontroller peripherals, including USART for file reception, GPIO for button-based UI, and NVIC for interrupt-driven controls (track selection, play/pause).
- Conducted code reviews and debugging sessions to ensure firmware stability and adherence to embedded C best practices.

Enterprise Application Project | *MERN Stack, Recharts, Jest* | *Enterprise Engineering Course Project*

2024

- Collaborated with a team of four using Agile Methodologies to develop a full-stack MERN application for a simulated enterprise, meeting specified business requirements.
- Built and tested a secure RESTful API with Node.js, Express.js, and MongoDB, featuring over 12 CRUD endpoints and achieving 95% unit test coverage with Jest.
- Developed a responsive React client with a Recharts analytics dashboard to visualize key metrics, which improved user task completion time and informed strategic decisions.
- Managed the deployment of the full MERN stack to an Ubuntu server, configuring Apache HTTP Server as a reverse proxy to the backend.
- Integrated and debugged an analytics dashboard with Recharts to provide data-driven insights, visualizing key metrics like sales trends and employee performance which informed strategic decisions in project reviews.

WatchOS Voice Chatbot | *SwiftUI, XCode, Groq API, REST* | *Personal Project*

2024

- Programmed a proof-of-concept voice-to-voice chatbot on watchOS using SwiftUI for the user interface and AVFoundation for audio capture and playback.
- Integrated with the Groq API for near-real-time AI inference, leveraging Whisper for speech-to-text and Llama 3.1 for text generation, achieving an average response time of under 2 seconds.
- Engineered the network layer to handle asynchronous API calls and manage data flow between the app and the backend services efficiently.

Rush Hour Puzzle Game | *Java, JavaFX, JUnit, Git* | *Course Project*

2023

- Collaborated in a team of three to build a fully functional Rush Hour puzzle game with a JavaFX GUI, following Agile/Scrum practices.
- Designed core game logic with a backtracking algorithm and wrote JUnit tests covering 90% of code to ensure stability.
- Managed Git workflow (branching, merging, and reviews) to maintain efficient version control and collaboration.

Personal Dashboard (kvn8888.github.io) | *Next.js, Auth.js, Turso, AWS S3, Gemini* | *Personal Project* 2024 – Present

- Built a full-featured personal ops hub behind Google OAuth — public portfolio on the front, authenticated dashboard on the back.
- Engineered an API usage monitor with daily snapshot history and burn-rate projections across Tavily, Vercel, Render, GitHub, and other integrations.
- Built a cover letter workbench with a Turso-backed reusable block and tag library, Gemini-powered block matching and rubric grading, and S3-backed draft and reference resume storage.
- Implemented a runtime secrets manager that overrides environment variables without a redeploy and optionally syncs them to Vercel; added a sign-in manager for approving invited users with per-page access grants.

Polymarket +EV Trading Bot | *TypeScript, Docker, React, Turso* | *Personal Project*

2026 – Present

- Fetches sportsbook odds (NFL, NBA, NHL) from The Odds API and de-vigs to derive true probability, then identifies mispriced Polymarket contracts with edge exceeding a configurable threshold.
- Sizes bets using Kelly Criterion (1/4 Kelly, 5% max position) and supports both paper trading and live execution via the Polymarket CLOB API.
- Implements stale-data rejection, multi-bookmaker agreement checks, and rate limiting to guard against erroneous trades; kill-switch pauses new trades while letting existing positions settle.

- Web dashboard with portfolio overview, live opportunities, position tracking, per-user secrets management, and a pooled Odds API usage monitor.
- Implemented Google OAuth + JWT authentication with a multi-tenant architecture — per-user bot runtime state, config, kill switch, opportunities cache, and CLOB wallet connection — preventing any cross-user interference in the shared scheduler.
- Secrets stored AES-256-GCM encrypted per user in Turso; Odds API key pool resolves user secrets → admin secrets → env vars, with credit-aware key rotation, deduplication, and 5-minute caching. CLOB wallet credentials are now persisted and auto-reconnected after restarts.
- Multi-stage Docker build (Vite → tsc → production with `--omit=dev`, non-root user); Vitest suite covering core business logic (EV calculator, Kelly sizer, fuzzy team matcher); deployed on Render with health check endpoint.

OpenFoodJournal | *SwiftUI, SwiftData, Gemini, HealthKit, CloudKit* | *Personal Project* – App Store 2026 – Present

- Built a local-first iOS food journal with AI-powered nutrition scanning: Gemini Flash Lite handles nutrition label scans in under 2 seconds; Gemini Pro with high-reasoning estimates plated meals.
- Tracks 30 FDA-recognized macros and micronutrients with opt-in HealthKit sync for active energy and 10+ micronutrients; includes a configurable daily macro summary bar with progress rings.
- BYOK architecture: users supply their own Gemini API key stored in the iOS Keychain; all API calls go directly from the device with no proxy server required.
- SwiftData persistence with automatic CloudKit Private Database sync across devices; integrated Open Food Facts database (4M+ products) for search and barcode lookup.
- Published on the App Store; resolved an initial Guideline 1.4.1 rejection by adding FDA-cited health disclaimers and inline AI estimation caveats.

CodeGym | *Go, React, Docker, PostgreSQL* | *Personal Project* 2026 – Present

- Building a full-stack coding practice platform with a Go backend, React frontend, and Dockerized per-language code execution sandboxes.
- Designed a filesystem-backed problem pack structure (language/category/problem) where each problem defines skeleton files, a canonical solution, and a test suite via a structured YAML manifest.
- Built the data layer with sqlc and goose for type-safe query generation and repeatable PostgreSQL schema migrations; developing a Storybook component library for the frontend.

FactCheckMe – 2nd Place, c0mpiled-1: Before the Ballot | *React, TypeScript, Supabase, ElevenLabs, Gemini, Hyperspell*

- Real-time audio fact-checking tool for voters: ElevenLabs Scribe transcribes live speech, Gemini 2.0 with Google Search grounding verifies claims and returns verdicts with confidence scores in 2–3 seconds.
- Engineered a Hyperspell semantic caching layer (85% similarity threshold) that recognizes paraphrased claims, cutting repeat verification latency from ~3s to under 100ms and reducing API costs by up to 90%.
- Full-stack pipeline: React/TypeScript frontend, Supabase Edge Functions, PostgreSQL with real-time subscriptions, voice activity detection, and 2-second claim-queuing throttle.

DepScope – 1st Place, Continual Learning Hackathon | *Node.js, Express, React, Gemini, You.com* 2026

- Autonomous 3-agent pipeline for open-source dependency due diligence: a Repo Health Analyzer (GitHub stats), an External Researcher (You.com Search for CVEs and community sentiment), and a Gemini 2.0 Flash Risk Scorer that synthesizes a letter grade (A–F) with severity-ranked findings and opinionated alternatives.
- Scored five risk dimensions – security (30%), maintenance (25%), stability (20%), community (15%), documentation (10%) – with automatic downgrades for unpatched critical CVEs, archived repos, and single-maintainer staleness.
- Streamed agent progress live via SSE to a React dashboard with radar chart scoring, pattern aggregation, and alternative package comparison.

GuardianEye – 2nd Place, Autonomous Agents Hackathon | *FastAPI, React, Neo4j, Gemini, Tavily, Yutori* 2026

- Multi-agent scam detection pipeline analyzing screenshots, voice recordings, and text messages to return a RED / YELLOW / GREEN verdict.
- Six parallel agents: Reka Vision (image text), GLiNER (entity extraction), Tavily (scam DB search, ~2s), Yutori Research (deep investigation, ~60s), Gemini 3 Flash (verdict synthesis), Neo4j (scam network graph mapping).
- Voice deepfake detection via Modulate; all services degrade gracefully to rule-based fallbacks without API keys; identical submissions return from SQLite/Turso cache instantly.

Cerberus – 3rd Place, HackWithBay 2.0 | *Neo4j, FastAPI, React, RocketRide, Claude, STIX* 2026

- Cross-domain threat intelligence agent: paste a single entity (npm package, IP, domain, CVE, or fraud signal) and Cerberus autonomously traces the full attack chain across software supply chain, network infrastructure, and financial fraud surfaces simultaneously.

- Built on Neo4j for native cross-domain graph traversal and RocketRide's wave-planning agent for parallel tool orchestration; streams a Claude-generated narrative with threat score, blast radius, IOC table, force-directed graph, geo IP map, and MITRE ATT&CK heatmap.
- Self-improvement loop writes confirmed threat patterns back to Neo4j so repeat investigations skip the LLM and return in ~ 2 seconds; exports STIX 2.1 bundles with TLP markings.