

Kelvin Le

College Park, MD | 856-952-1878 | kelvin.le.swe@gmail.com | linkedin.com/in/kelvinxxle | github.com/kelvinxxle

EDUCATION

University of Maryland, College Park

Expected May 2027

Computer and Information Science BS

- **Coursework:** Algorithms, Operating Systems, Databases, Advanced Data Structures, Machine Learning, Data Science

WORK EXPERIENCE

Microsoft Corporation (GitHub)

Jun 2026 - Sep 2026

Software Engineer Intern

Remote

- Fixed a Ruby defect in enterprise team identity sync that could silently revoke repository access for up to 10,000 users on a single team when identity-provider results were truncated; traced it to two compounding causes and added regression tests the existing suite lacked.
- Analyzed production telemetry and SQL for the enterprise deprovisioning path, identifying a job class issuing 3.5M+ primary database reads and zero writes per run; produced routing recommendations for a team initiative raising Enterprise Team member limits 3x for GitHub's 20 largest customers.
- Evaluated two proposed replica-routing changes and showed both could write stale data under replication lag; recommended against rollout and shipped instrumentation measuring the failure rate across a path covering 15M+ user-organization relationships.
- Isolated a regional service's Kafka consumer group in Go, eliminating cross-datacenter offset contention and enabling a 2x increase in replicas for a service streaming enterprise identity events.
- Built and deployed a GitHub App in TypeScript that auto-assigns a reviewer team to qualifying pull requests in a public Terraform provider with 440+ contributors and a 129-PR backlog at a five-month median age, replacing single-maintainer assignment.
- Shipped 26 merged pull requests across GitHub-owned repositories using a test-gated AI-assisted workflow, requiring a regression test to fail without each fix before merging it.

HouseMate

Jun 2024 - Aug 2024

Software Engineer Intern

College Park, MD

- Built a SwiftUI iOS app backed by Firebase Realtime Database that syncs shared household expenses and budgets live across devices.
- Implemented the expense and budget data model in Swift, keeping balances consistent across concurrent updates.

PROJECTS

UFC Fight Outcome Predictor - Pre-fight ML and betting decision pipeline | github.com/kelvinxxle/UFC-ML

- Connected probability quality to betting decisions by evaluating log loss and Brier score alongside accuracy, then converting model probabilities and American odds into expected value and quarter-Kelly stakes, since miscalibration directly produces the wrong stake size.
- Eliminated temporal leakage with 52 strictly pre-fight features, a grouped chronological 70/15/15 event split keeping each card in one partition, and train-only imputation, so no future bout informed training.
- Scraped 500+ bouts across 50+ events with pagination and polite pacing, then implemented a from-scratch Elo system with opponent-average Elo for strength of schedule; compared three models on validation, reaching ~62% on held-out chronological test data.

AlgoViz - Interactive algorithm visualizer | algoviz-six.vercel.app

- Designed a generic TypeScript step-trace engine whose single AlgorithmTopic contract powers all 10 topics through one shared player, narration, pseudocode highlighter, and counter system, adding each visualization with no per-topic engine changes.
- Validated complexity and trace correctness with targeted oracles: built a comparator-based binary min-heap for $O(E \log V)$ Dijkstra and pinned its 16-frame trace to a hand-verified graph; instrumented an LRU list to confirm equal pointer writes at sizes 8 and 800.
- Hardened a step-aware LLM explainer by resolving topic scope server-side, truncating untrusted input, and enforcing a 15-second timeout; injected a swappable provider to test every failure branch, returning a typed 502 or honest 503 instead of fabricated answers.

Swing Analyzer - Explainable geometric golf-swing analysis | swing-analyzer-gray.vercel.app

- Authored five explainable geometric rules over pretrained MediaPipe pose landmarks, normalizing displacement by golfer stature so zoom could not change results; converted coordinates to pixel space and verified portrait and landscape encodings agreed within $1e-6$.
- Prevented camera-motion and handedness artifacts by measuring hip displacement relative to the ankles and inferring a ball-direction sign from shoulder-to-hip position at address; applied that sign across every rule so left- and right-handed swings shared one implementation.
- Bounded a stateless FastAPI video pipeline with pre-analysis validation, a 50 MB upload cap, and a 60-second deadline; mapped oversize files, timeouts, and unexpected faults to explicit 413, 504, and controlled 500 responses, then discarded each processed video.

SKILLS

- **Languages:** Python, Java, TypeScript, JavaScript, Ruby, Go, Swift, C, SQL
- **Frameworks & Libraries:** React, Node.js, Express, FastAPI, Ruby on Rails, SwiftUI, Vitest, Playwright, MediaPipe, OpenCV
- **Data & Infrastructure:** PostgreSQL, MySQL, Redis, Kafka, Firebase, Datadog, Git, Linux, Vercel, OpenAPI