

Kyle Dang

Irvine, CA | kyle.dang.dev@gmail.com | linkedin.com/in/kyle-k-dang | github.com/KyleKDang

EDUCATION

University of California, Irvine

Irvine, CA

Bachelor of Science in Computer Science | Regents Scholar

2027

- Cumulative GPA: 3.92/4.00
- Relevant Coursework: Data Structures & Algorithms, Database Systems, Operating Systems, Computer Networks, Artificial Intelligence, Information Retrieval, Software Testing

TECHNICAL SKILLS

Languages: Java, Python, TypeScript, SQL

Frameworks/Libraries: Spring Boot, Jakarta EE, FastAPI, React, Next.js, Express.js

Developer Tools: Git, PostgreSQL, Docker, AWS, Claude Code

EXPERIENCE

Software Engineer Intern

June 2026 – Sept. 2026

IBM

Durham, NC

- Built microservices for an automated performance-regression triage pipeline in Java and Jakarta EE on Open Liberty, designed to replace ~10 hrs/week of manual review of 100+ daily benchmark results, ~70% false alarms
- Implemented the rerun validation logic that re-executes flagged regressions on real hardware with bounded retries and tracks state in PostgreSQL as a durable, restart-safe work queue, so only confirmed regressions reach engineers
- Built the pipeline's dashboard end to end as sole owner: a JAX-RS REST backend over PostgreSQL and Db2 with server-side aggregation and proxied pipeline controls, and a five-view React/TypeScript UI over 450+ scenarios
- Evaluated Horreum, an open-source benchmark store with change-point detection, as a replacement for in-house detection, wrote the adoption analysis and ingest design, and documented the decision not to adopt it

Undergraduate Research Assistant

Apr. 2026 – June 2026

Apache Texera | UC Irvine

Irvine, CA

- Built a workflow-to-Python translator in Scala for Apache Texera, an open-source workflow engine with 150+ operators, compiling visual dataflow pipelines into standalone pandas/plotly scripts in correct dependency order
- Added code generation for every transformation and data-source operator, from joins and set operations to file, API, and database readers, behind an opt-in interface, so existing workflows ran unchanged
- Built a reflection-based test harness that runs each operator in the live engine and as generated code and diffs the outputs, verifying generated operators end to end with no per-operator test code

PROJECTS

Anchor - Film Taste Engine | *FastAPI, PostgreSQL, React, NumPy* | [GitHub](#) | [Live Site](#) July 2026 – Present

- Designed and shipped a FastAPI and React app that anchors film ratings to the user's reference films instead of a drifting absolute scale, with every rating stored as a half-star band the user orders by hand on a visible wall
- Built a recommendation engine that learns the user's taste from their ratings, scoring unseen films with a NumPy logistic-regression model and reranking the shortlist with Claude in a background worker
- Built a test harness of 600+ backend tests that exercise the real API over a throwaway PostgreSQL database cloned per test from a migrated template, plus Playwright browser journeys, all gating deploys in CI

FinData Analytics API | *Java, Spring Boot, PostgreSQL, Docker, AWS* | [GitHub](#) Jan. 2026 – Mar. 2026

- Built a Spring Boot REST API for aggregating and analyzing financial time-series data, with PostgreSQL table partitioning, composite indexes, and pagination to keep queries fast
- Added a data ingestion pipeline that keeps market data current and consistent, with scheduled jobs, API rate-limit handling, and error tracking
- Designed a portfolio analytics engine that computes risk-adjusted returns, volatility, and weighted performance for user-defined allocations from historical price data

Hold'Em Analytics | *FastAPI, Go, PostgreSQL, Next.js, AWS* | [GitHub](#) | [Live Site](#) July 2025 – Dec. 2025

- Built and deployed a FastAPI and Next.js poker analytics platform over PostgreSQL for session tracking, hand evaluation, and real-time odds calculation
- Wrote a concurrent Go microservice for Monte Carlo odds simulation, ~3× faster than a single-threaded version