

# Calvin Tran

calvintran1027@gmail.com | [linkedin.com/in/calvintran-tech](https://www.linkedin.com/in/calvintran-tech) | [github.com/CTran10](https://github.com/CTran10)

Software engineer with 2+ years shipping production backend systems (Django, FastAPI, PostgreSQL). Strengths in backend performance, distributed and event-driven design, and end-to-end ownership of customer-facing products.

## SKILLS

---

**Languages:** Go, Python, TypeScript, JavaScript, SQL

**Backend:** FastAPI, Django, REST, PostgreSQL, Redis, Kafka / Redpanda

**Infra & Tooling:** Docker, CI/CD, Prometheus, Alembic

**Frontend:** React, Next.js, Svelte, TanStack

**AI:** OpenAI API, LLM evaluation workflows

## EXPERIENCE

---

**Software Engineer** — Breef

2024 – Present

- Cut database query counts up to 93% and p90 latency 56% across critical Django API endpoints, lowering endpoint DB workload ~70%.
- Caught and fixed LLM token-accounting logic that overstated operational AI cost ~6x, correcting framework-level cached-token handling.
- Built PostgreSQL-backed data pipelines and tooling for LLM production sampling, review tracking, and evaluation workflows.
- Automated internal evaluation workflows via OpenAI API integrations, cutting manual operational overhead.

**Full Stack Web Development Intern** — IBM

2022 – 2024

- Built internal React/TypeScript applications for UX research teams, streamlining workflow visibility and data collection.
- Integrated REST APIs and JWT-based authentication to strengthen platform security across multiple services.

## PROJECTS

---

**Clearance** [Go](#) · [Kafka/Redpanda](#) · [PostgreSQL](#) · [Redis](#) · [Docker](#)

- Built an event-driven payment-authorization platform across four Go services over Redpanda/Kafka, using a transactional outbox so no event is lost after a database commit.
- Engineered idempotent, effectively-once processing with idempotency keys and at-least-once consumers with retry and dead-letter handling, keeping the pipeline correct across consumer crashes and message redelivery.
- Prevented double-spend under redelivery with concurrency-safe funds authorization: Postgres advisory locks over a double-entry immutable ledger.
- Added proxy-aware rate limiting (atomic Redis Lua), gated Prometheus metrics, and CI running build, vet, and tests

**Apex — Setup Vault** [Next.js](#) · [FastAPI](#) · [PostgreSQL/RLS](#) · [Supabase](#) · [TypeScript](#) · [Docker](#)

- Built a full-stack setup-logbook and community platform for track riders: Supabase Auth, onboarding, garage/setup editing, session logging from current setup or saved presets, maintenance history, shared feed/search, follows, kudos, comments, and opt-in public rider profiles.
- Modeled the rider iteration loop in Postgres: mutable bike setup state, reusable setup presets, immutable per-session setup snapshots, adjustment apply/result linking, per-bike timelines, shop handoffs, coach feedback, rider goals/drills, and revocable access grants.
- Enforced privacy with FastAPI JWT validation, request-scoped SET LOCAL ROLE/JWT claims, least-privileged runtime DB config, Postgres RLS policies, revoked anon table privileges, and IDOR/RLS tests covering owner, shared, social, shop, and coach paths.
- Added production-readiness plumbing: Pydantic schemas, keyset pagination, exact-filter shared-session search with privacy-safe telemetry, configurable rate limits, security headers, request IDs, 25 Alembic migrations, and CI/dependency audits across 300+ backend/frontend tests.

## EDUCATION

---

**Colorado School of Mines** — B.S. Computer Science, 2026