

Ryan Tang Kwang Loke

London, UK | +44 7376411777 | contact@ryantang.dev
linkedin.com/in/ryankwangloketang | github.com/RyanT04 | ryantang.dev

PROFILE

Software Engineering graduate (BEng, 2:1) with a focus on low-latency backend systems and concurrency in Go. Dissertation work benchmarked a self-built order-matching engine against an equivalent TypeScript implementation, covering price-time priority matching, leverage and liquidation logic, and live market data ingestion. Prior finance analyst internship handling transaction reconciliation and variance analysis. Seeking a graduate software engineering role in London.

EDUCATION

University of Southampton

Southampton, UK

BEng Software Engineering — 2:1 (Upper Second-Class Honours)

Sep. 2023 – Jun. 2026

- Dissertation: a comparative study of Go and TypeScript architectures for real-time trading systems, implemented as two functionally equivalent trading simulators (TradeGo and TradeX).
- Relevant modules: Advanced Databases (query optimisation, transactions and recovery), Algorithms, Formal Methods (Event-B), Machine Learning.

EXPERIENCE

Assessly Pte Ltd | Growth Engineer (Full-Stack)

Dec. 2025 – Feb. 2026

- Built, tested, and shipped customer-facing full-stack improvements by translating recruiter and candidate feedback into scoped features and bug fixes.
- Introduced real-time candidate test feedback, improved the AI interview experience, and developed profile pages for the top 100 ranked candidates, contributing to a 26% reduction in churn and higher customer satisfaction.
- Implemented PostHog analytics to track funnels, drop-off points, and feature usage patterns, enabling data-informed prioritisation of product improvements.

Trane Aire Sdn Bhd | Finance/Business Analyst Trainee (Full-time Internship)

Jul. 2025 – Sep. 2025

- Analysed and reconciled 8,000+ financial transaction records to identify trends, surface anomalies, and support quarterly strategic planning with data-driven insights.
- Cleaned and integrated datasets from four departments, improving data accuracy by 15% and reducing reporting turnaround time by 20%.
- Supported budget preparation and variance analysis for two major cost centres, identifying opportunities equivalent to 5% in potential annual savings.

PROJECTS

TradeGo — Crypto Trading Simulator | Go, PostgreSQL, AWS, Docker, Nginx, DigitalOcean

Sep. 2025 – Present

- Built a full-stack cryptocurrency trading simulator in Go as one half of a paired architectural comparison, benchmarking a performance-oriented Go stack against an equivalent TypeScript/Node implementation (TradeX).
- Engineered a concurrent order-matching engine using goroutines and price-time priority data structures, with per-order execution latency instrumented and reported in microseconds on every trade.
- Hardened the engine's transactional correctness: closed a time-of-check/time-of-use race allowing balance overdrafts, made buy, sell, and leveraged-position operations atomic under row-level locking, and restructured the data layer around testable interfaces.
- Wrote table-driven and concurrency tests covering order invariants (no fill without sufficient balance, no negative balances, no double-refund on liquidation), including a regression test running 50 concurrent buys against a fixed balance under Go's race detector.
- Built a GitHub Actions CI/CD pipeline gating every push on build, vet, formatting, and race-enabled tests, then deploying the verified image to production over SSH.
- Implemented spot and leveraged trading (up to 50x) with a margin and liquidation engine, backed by a liquidation worker that evaluates open positions against live prices, ingested for 400+ instruments over Bybit WebSocket with REST polling fallback.
- Self-hosted the platform with Docker Compose on a DigitalOcean droplet behind Nginx with Let's Encrypt TLS (`trade-go.tech`), migrating off AWS ECS Fargate to cut running costs and simplify operations.

TradeX — Crypto Trading Simulator | Next.js, TypeScript, React, Supabase, TailwindCSS, Vercel

Sep. 2025 – May 2026

- Built the TypeScript counterpart to TradeGo: a full-stack trading simulator on Next.js with Supabase (Postgres and auth), deployed on Vercel, serving as the architecture and performance baseline for the comparison.
- Implemented the order-book and trading interfaces over REST and WebSocket, then profiled concurrent workloads to surface the latency and CPU bottlenecks that motivated the Go re-implementation.

TECHNICAL SKILLS

Languages: Go, C++, Java, Python, JavaScript, TypeScript, SQL, C#

Frameworks: Gin, Node.js, Express.js, Nest.js, React, Next.js

Databases: PostgreSQL, Supabase

Testing: Go testing (table-driven, `-race`), JUnit

Infrastructure: Docker, Docker Compose, Nginx, Linux, DigitalOcean, AWS, CI/CD (GitHub Actions)

Tools & Libraries: Git, Jira, pandas, NumPy, Matplotlib, TailwindCSS, Rodin (Event-B)