YI-LEI LI

Newark, DE (open to relocation) | ray1998514@gmail.com
lego514.github.io | linkedin.com/in/legoli | github.com/Lego514

Summary

Software engineer with an **M.S. in Computer Science** (University of Delaware, May 2026) who ships tested, production-style systems: a deployed React + TypeScript app on Node.js / PostgreSQL with its own offline sync engine, a Java 21 / Spring Boot 3 banking service with atomic, idempotent transfers, and a TypeScript pipeline pulling 22,000+ job postings per run. Prior backend internship at Cathay Financial Holdings.

Education

University of Delaware — M.S. in Computer Science, GPA 3.66/4.0 *Aug 2024 – May 2026*
National Kaohsiung University of Science and Technology — B.S. in Electrical Engineering *2020*

Technical Skills

Languages: Java, TypeScript/JavaScript, Python, C++, SQL, HTML/CSS
Backend: Spring Boot, Node.js (Hono), JPA/Hibernate, REST APIs, WebSockets, Cloudflare Workers (Durable Objects)
Frontend: React, PWA (service workers, IndexedDB)
Databases: PostgreSQL, Supabase (Row-Level Security), Oracle; window functions, CTEs
Security: OAuth 2.0 (PKCE), session auth, CSRF protection, rate limiting
Testing & DevOps: JUnit 5, Testcontainers, Vitest, Playwright, Docker/Compose, GitHub Actions CI, Maven, Git
AI: Claude API with structured JSON output (interview-answer feedback, job-posting classification)

Projects

Jun 2026 – Present

LeetCode Coach — Offline-First Interview-Prep Web App — React, TypeScript, Hono, PostgreSQL [Live] [GitHub]

- **Full-stack PWA** that a friend and I use daily for interview prep, deployed on Render + Neon; login secured with scrypt-hashed passwords, hashed session tokens, CSRF checks, and rate limits.
- Built an **offline-first sync engine**: local writes queue in an outbox; the server applies them with last-write-wins and versions each change under a per-user lock, so a device syncing later never skips one.
- **421 unit/API and 123 end-to-end tests** gate every deploy; CI also builds the Docker image, starts it with PostgreSQL via Docker Compose, and runs end-to-end tests against the container.

DSA Whiteboard — Open-Source React Package — React, TypeScript, Cloudflare Workers [Live] [GitHub]

- Data-structure whiteboard with **live co-editing over WebSockets** for up to 8 people: a Cloudflare Worker with one Durable Object per room relays each change, and offline edits replay on reconnect.
- Published to **npm** by GitHub Actions (trusted publishing with provenance) and reused in LeetCode Coach; view-only links are enforced by the server, which rejects any edit they send.

Banking Transactions API — Java 21, Spring Boot 3, PostgreSQL [GitHub]

- Banking REST service with atomic transfers (`@Transactional`); **ordered pessimistic row locks** (`SELECT ... FOR UPDATE`, ascending ids) prevent double-spend and deadlock; an immutable audit ledger records every movement.
- **Idempotency keys**: each key's row commits with its transfer, so a retried request replays the first response.
- **32 tests in CI on real PostgreSQL** (Testcontainers): a 100-transfer concurrency race, 8 simultaneous retries of one key, and SQL reconciliation checks (window-function running totals) proven to catch corrupted rows.

Job-Posting Data Pipeline — Node.js, TypeScript, Supabase [GitHub]

- Scheduled pipeline pulling **22,000+ job postings per run** from 116 employer and job-board sources through 11 platform adapters (Workday, Greenhouse, Lever, Ashby, iCIMS, Oracle Cloud, ...) behind one normalization layer, with bounded-concurrency fetches and per-source failure isolation.
- Idempotent re-runs via a PostgreSQL state table; runs unattended on a cron with Telegram + email alerts. Its Claude API screening stage **fails closed**: on API errors, results are held back instead of sent unscreened. 350 Vitest tests.

Experience

Cathay Financial Holdings Co., Ltd. — Back-End Software Engineering Intern, Taipei *Jul 2023 – Nov 2023*

- Built a full-stack e-book store training project (Java, Oracle SQL, HTML/CSS/JavaScript) on an Agile team: CRUD pages plus a purchase flow that grants the buyer's e-book access in one **database transaction**.
- Replaced loading every book into Java with **SQL pagination**, so each page fetches only the rows it shows.
- Traced the bank's transfer flow end to end: balance checks, per-account concurrency control, and **ACID** transactions.

Graduate Research Assistant, Chang Gung University — **NeRF** 3D scene reconstruction *2021 – 2022*
Undergraduate Research Assistant, NKUST IoT Lab — live sensor dashboard (REST/WebSocket) *2018 – 2020*