

# Dinh Dai Vu

(+84) 818916621 | dinhdaivu1109@gmail.com | LinkedIn | GitHub

## EDUCATION

---

**VNUHCM – University of Science**  
*Bachelor of Information Technology — GPA: 3.52/4.0*

Ho Chi Minh City, Vietnam  
*Sep. 2023 – Aug. 2027*

## CERTIFICATIONS & AWARDS

---

**ICPC Vietnam Provincial Programming Contest** | *Provincial Qualifier*

Oct. 2024

**IELTS: 6.0**

Mar. 2023

## EXPERIENCE

---

**Software Engineer Intern**

Jan. 2026 – Present

*i-stech*

*Ho Chi Minh City, Vietnam*

**Social Analytics Platform**

Apr. 2026 – Present

- Diagnosed and resolved **15x** ETL bottleneck on a **28M+**-row pipeline by replacing SQLAlchemy `execute_many` with `psycopg2.execute_values` and 100K-row chunked reads, cutting full-load time by over **80%** (from **3h30m** → **40min**) while capping peak RAM under **0.3 GB**.
- Cut CSV export download time by **70%** (13s → 4s) on a **42M-row** PostgreSQL table via LRU caching and proactive cache warming, eliminating live query execution on repeated export requests

**Project Execution Platform (PEP)**

Jan. 2026 – May. 2026

- Built refund workflows for **Stripe** and **bank transfer** payment methods with **idempotency keys** and **optimistic locking** to prevent duplicate processing under concurrent requests, correctly crediting customer balances and adjusting pending invoice amounts
- Automated the full billing lifecycle via background jobs — contract expiration checks, **Stripe** event ingestion, and database reconciliation — eliminating **10+ hours/week** of manual tracking and data entry
- Eliminated **N+1** sequential query pattern on buyer chat dashboard by consolidating room membership, unread counts, and read status into a **single indexed SQL query**, reducing DB calls and cutting response time by **~90%** (5s → 500ms) ,

## PROJECTS

---

**Authlyn**

Apr. 2026 – Present

*Personal Project*

**GitHub**

- **Technologies:** *Spring Boot 4.0, Spring Security, React, PostgreSQL, Redis*
- Fixed concurrent refresh token race condition via **pessimistic locking**, preventing reuse-detection bypass under parallel requests
- Replaced unbounded `ConcurrentHashMap` in rate limiter with **Caffeine LRU cache** (TTL-evicted), eliminating heap exhaustion under IP-spoofed flood attacks

**HomeStay Dorm**

Mar. 2026 – May.2026

*Team Leader (Team of 4)*

**GitHub** | **Live Page**

- **Technologies:** *Spring Boot 4.0, Angular, PostgreSQL*
- Designed and implemented a REST backend using **clean architecture**, enforcing strict separation between domain, application, and infrastructure layers.
- Eliminated an **N+1** query bottleneck and transactional email I/O in a `@Scheduled` processor, reducing scheduler-tick p99 latency by **45%** (**1,650ms** → **899ms**) under load testing with **20 concurrent users** across **50 runs**.

## TECHNICAL SKILLS

---

**Languages:** Java, TypeScript, Python

**Frameworks:** Spring Boot, Spring Security, Angular, Next.js, FastAPI

**Databases:** PostgreSQL, Redis

**Integrations:** Stripe, Supabase

**Cloud & Tools:** Docker, GitHub Actions, Railway (Cloud provider)