PHAM DUC THANG

Email: phamthang3564@gmail.com **Location:** Ho Chi Minh City, Vietnam GitHub LinkedIn

EDUCATION

University of Greenwich *Expected: 2027*
BSc Information Technology
TOEIC English Proficiency — Score: 700

SKILLS

Technical: Java, Spring Boot, PostgreSQL, Redis, RabbitMQ, Kafka, WebSocket, OAuth2/JWT, Docker, Git, React, TypeScript, React Native/Expo. AI workflow: ChatGPT, Claude, Codex, Markdown project context and decision notes.

PROJECTS

Internal Room Booking System (RoomHub) *Mar 2026 – May 2026*

github.com/phamthang44/internal-room-booking-system

Role: Full-Stack Developer - Personal classroom-booking project with Docker-based deployment.

Key Contributions:

- **Implemented concurrency protection** using JPA optimistic locking (@Version) and PostgreSQL time-range exclusion constraints to help reject overlapping room reservations and stale updates. A k6 script covers concurrent booking writes and room reads.
- **OAuth2 + JWT authentication** with Google login integration, refresh-token rotation, and Redis-backed token invalidation; implemented password recovery and OTP workflows.
- **Async notifications with RabbitMQ** - separated booking side effects from the request flow using Spring events and message consumers for background delivery.
- **Real-time booking updates** through Spring WebSocket; maintained schema changes with Flyway migrations and delivery configuration with Docker and GitHub Actions.

Technologies used: Java, Spring Boot, PostgreSQL, Redis, RabbitMQ, WebSocket, OAuth2, JWT, Flyway, Docker, GitHub Actions

ChargeOps — EV Charging Booking Platform *Jun 2026 – Present*

Role: Full-Stack Developer - University of Greenwich final-year project; backend workflows and driver/operator interfaces.

Key Contributions:

- **Implemented booking lifecycle rules** across reservation, payment, check-in, cancellation, and refund states. Updated refund eligibility around the payment grace period and separate review for verified station incidents.
- **Designed concurrency-aware reservations** with database locking and version checks on payment-critical state transitions; separated booking commits from downstream events with a transactional outbox and Kafka.
- **Implemented QR check-in workflows** that validate booking ownership, connector match, and the eligible check-in window; support booking history and session monitoring with simulated charging telemetry.
- **Built driver and operator interfaces** with React Native/Expo and React, typed API access, and bilingual support. Organized requirements and decision notes in a Markdown knowledge base to provide context for AI-assisted implementation and review.

Technologies used: Java, Spring Boot, PostgreSQL/PostGIS, Redis, Kafka, Keycloak, React, TypeScript, React Native/Expo, Docker