

Jiemin (Jenny) Zhang

London, UK jenny.tuetoo@gmail.com

Links

[LinkedIn](#) · [GitHub](#) · jieminzhang.me

Daily use

Python · Git · Linux · Docker

Built in projects

FastAPI · PostgreSQL · Redis · Qdrant · LLM agents · RAG · pytest · GitHub Actions

Working knowledge

C · C++ · Bash

Learning next

Production deployment · distributed systems · SQL · Rust

Backend and AI engineer. I moved into tech after **twelve years as a corporate accountant** and an MSc in Finance and Investment.

I completed the 42 London Common Core in **12 months with every project at full marks**. I build FastAPI services, event-driven backends and LLM agents that are measured with benchmarks, and I understand the layer underneath: threads, queues, signals and containers.

I am looking for a **backend or AI engineering role** where correctness matters.

Technical projects

AlphaMind (ft_transcendence) – multi-agent AI research platform

Jun – Oct 2026 | team of three

Python, FastAPI, PostgreSQL, Redis, Qdrant, LangGraph, Docker

- **Product owner and backend lead:** scope, the three services (auth, core, AI), the database layer, tests, CI/CD, cloud deployment and review of all changes. One teammate led the infrastructure, one the frontend.
- Three independent FastAPI services with separate schemas and no cross-service foreign keys, connected by REST and events (key events on Redis Streams, with an outbox). 17 containers start with one command.
- Account security: rotating refresh tokens with replay detection, TOTP two-factor login, OAuth with PKCE, role-based permissions and rate limits.
- AI service: LangGraph pipeline with up to nine specialist agents, a bull and bear debate and a reflection step; hybrid search (BM25 plus vectors, Reciprocal Rank Fusion) and layered agent memory. Developed with AI coding tools; I set the requirements, reviewed the changes, ran the evaluations and decided each change.
- Quality: batch-run evaluations of the AI service; a pytest suite against real PostgreSQL and Redis (written with AI assistance), plus flake8 and mypy --strict gates in CI.

Agent Smith – autonomous coding agent

May – Jun 2026 | team of two

Python, Docker, MCP, pytest

- Built the agent loop (Thought, Code, Observation) with step, token and time limits, an answer parser for several formats, and an LLM client with key rotation and retries.
- Docker lifecycle for SWE-bench (one container per task, result collected as a git patch). My teammate built the code sandbox and MCP tool servers; we integrated and tested together and each studied the other's half.
- Team benchmark of 11 models: best about 90% on all 257 MBPP tasks; best solved 8 of 8 SWE-bench tasks in its best run.

RAG over the vLLM codebase

Mar 2026 | individual

Python, BM25, TF-IDF, Reciprocal Rank Fusion

- Hybrid retrieval with AST-based chunking and multi-query expansion. Recall@5 of about 0.76 on documentation and 0.66 on code on a 100-query benchmark, above the project thresholds.
- Latency work with BM25S and caching; separate modules for chunking, indexing, retrieval, generation and evaluation so each can be tested and swapped alone.
- Measured that multi-query search costs 3 to 4.5 times more latency and helps only on code (100-query benchmark).

treenity – message queue system in C++

Sep – Oct 2026 | team of two

- Built the server and concurrency: a reader and a worker thread per topic, append-only log with offsets, graceful shutdown with ppoll and a signal mask, and a hand-written hash map. A teammate built the client, codec and tests.

- Design choice: one worker thread per topic keeps consumer state free of locks and messages from one producer in order; a full consumer queue is waited on for at most 200 ms, then the message is dropped for that consumer.

Codexion – multi-threaded scheduler in C

Jan 2026 | individual

- FIFO and EDF scheduling over a priority queue; both resources taken in one critical section, so no hold-and-wait and no deadlock. Monitor thread detects burnout within 10 ms. A 200-coder stress test finishes in about a second.
- Design choice: one scheduler lock keeps decisions deterministic and easy to reason about, trading some throughput for correctness.

Call Me Maybe – constrained decoding

Feb 2026 | individual

- Logit masking so a small model (Qwen3-0.6B) can only choose valid function names; argument values come from rules.

Other 42 projects – code on GitHub

2025 – 2026

- **Inception:** multi-service Docker Compose stack from custom Dockerfiles: NGINX with TLS, WordPress with PHP-FPM, MariaDB, Redis, Adminer, Portainer, FTP; Docker secrets and persistent volumes.
- **Fly-in:** drone routing on a graph with capacity limits; my own Dijkstra-based planner and simulator, no graph libraries.
- **The Answer Protocol:** real-time multiplayer game over a line-based TCP protocol; async Rust server (team project).
- **A-Maze-ing, Pac-Man:** maze generation (DFS, Prim, Kruskal) and BFS solving; Pac-Man with ghost modes and a state machine.
- **NetPractice, Born2beroot, libft, push_swap, 11 Python modules:** networking, a hardened Debian VM, C fundamentals, algorithms, OOP and Pydantic.

Education

Software Engineering @ 42 London 42

Oct 2025 – present

- Selected after the September 2025 Piscine. Common Core completed at 100% in October 2026 (usually 1 to 2 years). Every project passed at full marks, including the bonus wherever a project has one. Exams Rank 02 to 06 passed.
- Now in the 42 Advanced part. Peer-reviewed projects in systems programming, concurrency, networking, algorithms and AI.

MSc, Finance and Investment (Merit) @ Durham University

Sep 2023 – Jan 2025

- Thesis mark of 72 (distinction level): “The effects of options trading on the stock market”. Three years of S&P 500 panel data (2020 to 2023), analysed in R and Python with pooled, fixed-effects and random-effects models. Found that options volume significantly raises stock trading volume (calls more than puts) and helps predict stock price moves.

BSc, Financial Management @ University of International Business and Economics

2007 – 2011

Experience in previous career (accounting)

Corporate Accountant @ Gallup Beijing

Aug 2013 – Sep 2023

- Ran end-to-end accounting for Gallup China and Hong Kong (payables, receivables, billing, payroll, tax, cost control), taking on more areas over ten years.
- Built annual budgets, rolling cash-flow forecasts and monthly variance reports that senior management used for planning and decisions.
- Acted as financial business partner: built pricing and contract models with the sales and operations teams.
- Improved payroll, payables and onboarding processes; passed every annual financial, tax and internal audit with zero discrepancies.

Accountant @ Schneider Electric Beijing

Jul 2011 – Jul 2013

- Top performer in Finance Shared Services, consistently above targets. Standardised reconciliation methods that the whole team adopted; wrote process manuals and trained new hires.

Hackathon and qualifications

- **BearingPoint AI Hackathon @ 42 London, 1st place (Dec 2025).** Designed and prototyped a multi-agent sales intelligence copilot with a winning team of six.
- Microsoft Azure AI Fundamentals (2024) · CFA Program Level I (2023) · CICPA Certified Public Accountant (2019)