

Joey (Yepeng) Zhu

me@yepengzhu.com | linkedin.com/in/yepeng-zhu-joey | github.com/zyppp1 | yepengzhu.com | London, UK

Backend / Full-Stack Engineer — Python, TypeScript & AWS | AWS Certified Solutions Architect (Associate)

SUMMARY

Backend and full-stack engineer specialising in RESTful API design, distributed systems and serverless cloud architecture on AWS — commercial experience building high-volume B2B transaction APIs in Python at Spinnr Tech (London), plus a browser-automation product with 50+ paying users and £2,000+ revenue. AWS Certified Solutions Architect – Associate with an MSc in Advanced Computer Science (University of Nottingham). Combines hands-on problem-solving with clean code — readable, testable, maintainable.

WORK EXPERIENCE

Spinnr Tech Ltd (London, UK) — Backend Engineer

Jul 2025 - Apr 2026

- Designed and implemented RESTful APIs for B2B integrations with gaming providers across multiple countries, supporting high-volume, high-concurrency transaction workflows with idempotency guarantees, PostgreSQL primary relational storage and Redis caching to reduce repeat database queries and improve API response times.
- Owned partner-facing delivery on a B2B integration project: agreed API documentation and feature scope with partners, identified potential collisions with the existing system in advance, and coordinated partner-side setup and IP whitelisting, tracking delivery on a Trello Kanban board.
- Implemented structured logging across microservices and added dedicated error codes for anticipated failure modes, improving observability and reducing mean time to debug.
- Built a Node.js mock server simulating external partner APIs, enabling integration testing in isolation from live third-party systems and supporting stage testing by QA and DevOps teams, releasing through a GitLab stage-to-production workflow with Docker and nginx.

Everbridge, Inc (Beijing, China) — Software Development Engineer in Test (Internship)

Sep 2022 - Mar 2023

- Automated 200+ regression test cases with Python and Selenium, cutting regression testing time by 35% across 31 releases.
- Performed API testing with Postman and diagnosed customer-reported issues, communicating root causes in English.
- Collaborated closely with front-end and back-end developers in a Scrum team with daily morning stand-ups, providing clear bug reports and improvement suggestions.

SKILLS

Languages: Python, TypeScript, JavaScript, Java, C#, Go, C/C++, HTML5, CSS

Backend: Flask, FastAPI, Node.js, Express.js, RESTful API design, object-oriented design, microservices, WebSockets, Server-Sent Events (SSE), event-driven architecture

Frontend: React, Next.js, Tailwind CSS

Databases & Caching: SQL (PostgreSQL, MySQL, SQLite), Redis, MongoDB, DynamoDB, SQLAlchemy

Cloud & DevOps: AWS (Lambda, API Gateway, Amplify, S3), Docker, Kubernetes, containerisation, Continuous Integration/Continuous Deployment (CI/CD), Git, GitLab, nginx, Fly.io, Vercel

AI & LLM Integration: Large Language Models (LLM), LangChain, Retrieval-Augmented Generation (RAG), ChromaDB, prompt engineering, Claude API, OpenAI API, DeepSeek API

Testing & Security: Playwright, pytest, Selenium, Postman, automated & integration testing, JWT, OAuth 2.0 (OAuth2), HMAC, structured logging

Practices: Agile (Scrum, Kanban), Trello

PROJECTS

AI Resume Editor — github.com/zyppp1/AiResumeEditor · ai-resume-editor.fly.dev

2026

- Built a full-stack AI resume platform on Flask: a 7-stage LangChain multi-agent pipeline (job-description analysis → RAG fact retrieval → LaTeX tailoring → cover letter → XeLaTeX compile → ATS quality check → persistence) generating tailored, ATS-scored PDF resumes.
- Implemented Retrieval-Augmented Generation with ChromaDB and locally run sentence-transformers embeddings, retrieving the top-20 most relevant facts per job description and cutting prompt size by ~70% at zero API cost.
- Used a structured JSON profile of atomic facts with signal tags as the single source of truth, enabling fine-grained fact selection without pre-written role-specific bullets, and imported existing .docx and .pdf resumes into that schema using an LLM together with python-docx and pdfplumber.
- Streamed generation progress over Server-Sent Events (SSE) with a live in-browser PDF preview (vendored PDF.js).
- Secured the API with JWT authentication (PyJWT + bcrypt) and optional Google OAuth, served downloads via short-lived HMAC-SHA256-signed URLs, and rate-limited generation with Flask-Limiter (20 generations/day, 5/hour per user).

- Modelled users, profiles and generated documents with the SQLAlchemy ORM — SQLite in development, PostgreSQL in production. Containerised with Docker and deployed to Fly.io with environment-switchable S3 storage via boto3.
- Designed a provider-agnostic LLM layer configurable purely by environment variables (Anthropic, OpenAI, DeepSeek, Groq, local Ollama — production runs DeepSeek).
- Backed the platform with a 224-case pytest suite — 214 fully offline unit/integration tests (auth, API routes, compiler failure paths, RAG retrieval) as the default gate, plus 10 Playwright end-to-end tests covering the core user flows.

Joey OS: Serverless Interactive Portfolio with Live AI Chat — Next.js, TypeScript, AWS · yepengzhu.com 2025

- Designed an event-driven serverless architecture — AWS API Gateway, Lambda, DynamoDB and Amplify — with no always-on servers, running separate stage and production environments for release control.
- Built the front end with Next.js, React, TypeScript and Tailwind CSS, responsive across screen sizes, with the original retro desktop-OS version preserved as a playable exhibit.
- Delivered real-time chat via Pusher WebSockets with Telegram Bot API routing, hand-signing the server's Pusher REST API calls with HMAC-SHA256 via native Python hashlib/hmac — no Pusher server SDK in the zero-dependency Lambda.
- Solved iOS/Android background-tab WebSocket loss with visibilitychange reconnection, a DynamoDB offline message buffer and client-side deduplication. Added VPC-less Redis rate limiting via the Upstash REST API, avoiding ElastiCache VPC cold-start overhead and continuous billing.
- Grounded an LLM assistant in a curated knowledge base with strict prompt guardrails to prevent hallucination, auto-logging unanswerable queries to DynamoDB and surfacing them through a secured review API for iterative knowledge-base updates.

Zoho Mail AI: Email Classification & Routing — github.com/zyppp1/zoho-mail-ai 2026

- Architected a 3-stage hybrid pipeline — rule-based classification (rejections, OTPs, known domains) → dynamic prompt construction → LLM inference — so only ambiguous emails reach the model, cutting LLM API cost by 60–70%.
- Designed a priority-ordered decision-tree classification prompt with few-shot examples and a confidence field, routing uncertain cases to human review.
- Integrated the Zoho Mail REST API with OAuth2 auto-refresh on 401 responses and batched folder operations (20 per request), and added incremental sync via a known-IDs set to skip already-processed emails and avoid redundant API calls.

AI Code Guide: Streaming Coding Mentor — Next.js, TypeScript · code-guide-ten.vercel.app 2025

- Decomposed software development into a 4-phase AI pipeline (Requirements Analysis, Environment Setup, Code Generation, Automated Testing), each phase driven by a dedicated Claude skill agent streaming structured guidance.
- Built a Next.js/TypeScript streaming LLM gateway handling chunked SSE responses from the Claude API for real-time delivery without full-page reloads, with multi-provider support switchable via a single config.
- Implemented JWT signed-cookie access control with per-session request budgeting to prevent token abuse, and designed DynamoDB-backed session persistence with auto-compact summarisation of older context near token limits.

Visa Appointment Automation System — Python, JavaScript, Telegram Bot API 2024

- Built a browser-automation product around a multi-stage pipeline — slot monitoring → booking → Telegram notification → autonomous rescheduling — in Python with a JavaScript/Tampermonkey semi-automated booking layer.
- Delivered the MVP in 2 days, then served 50+ paying users and generated £2,000+ revenue within two weeks of launch, iterating continuously on user pain points.

remote-telegram-Claude: 24/7 AI-Agent Supervision System — Bash, Python, macOS launchd 2026

- Packaged a 24/7 AI-agent supervision system for reuse: a launchd watchdog with state-aware heartbeat, resume-anchored crash recovery, an installer with interactive setup wizard, and bilingual English/Chinese documentation.

EDUCATION

University of Nottingham Sep 2023 - Dec 2024
MSc in Advanced Computer Science Merit Award

Beijing Union University Sep 2019 - Jun 2023
Bachelor of Engineering in Software Engineering Top 5%

CERTIFICATIONS

AWS Certified Solutions Architect – Associate Amazon Web Services

PUBLICATIONS

Using neural networks to explore path planning algorithms for robots Jun 2023

- doi:10.54254/2755-2721/5/20230645 — neural-network approaches to robotic path planning across environments.