Minh Quan Nguyen

Melbourne, VIC | +61 468 356 686 | minhquan.nguyen1202@gmail.com | linkedin.com/in/minh-quan-nguyen-742b9142b | github.com/mingjun6886

Education

Monash University — Melbourne, VIC
Master of Artificial Intelligence — *Feb 2025 – Dec 2026 (expected)*
- Relevant units: Machine Learning, NLP, Cloud Computing, Big Data Processing

Macquarie University — Sydney, NSW
Bachelor of Information Technology, Major in Data Science — *Feb 2023 – Dec 2024*
- Relevant units: Artificial Intelligence for Text and Vision, Graphics, Multivariate Methods and Data Mining, Database Systems

Technical Skills

AWS:	Lambda, EC2, S3, SQS, SNS, DynamoDB, RDS, API Gateway, Cognito, CloudFront, ECR, IAM, CloudWatch, X-Ray
IaC / DevOps:	Terraform, Docker, Linux, GitHub Actions (OIDC), checkov, Git (GitHub)
AI / ML:	LLM APIs (OpenAI, DeepSeek), RAG, LangChain, LangGraph, hybrid retrieval (BM25 + dense, RRF), Qdrant, PyTorch, scikit-learn, OpenCV
Languages:	Python, JavaScript, SQL
Backend / Data:	FastAPI, Pydantic, PostgreSQL, MySQL, Vue 3, pandas
Practices:	Architecture Decision Records, structured logging, observability, least-privilege security, Agile

Experience

CMC Telecom — Vietnam
AI Engineer Intern — *Nov 2025 – Mar 2026*

Python, FastAPI, LangChain, LangGraph, PostgreSQL, Qdrant, DeepSeek, Docker Compose

- Designed and built an LLM-powered career platform (RAG) over 1,081 real IT job postings from Vietnam and Europe: semantic job search, natural-language analytics, and PDF CV-to-job matching with skill-gap reports.
- Architected hybrid retrieval (BM25 and dense embeddings merged with Reciprocal Rank Fusion, then bounded, schema-validated LLM reranking); measured BM25 precision@5 of 0.34 versus 0.88 for dense retrieval on paraphrased queries to tune the fusion.
- Secured the text-to-SQL path with defence in depth: a LangGraph router, AST validation (single `SELECT`, table and column allowlist, enforced `LIMIT`), a read-only database role, and query timeouts.
- Controlled cost and failure modes: batched LLM extraction, rate limiting, retries with recursive batch splitting, decision caching, and per-client and daily caps on paid LLM routes in a password-gated public demo.
- Deployed with Docker Compose behind automatic HTTPS, with tested PostgreSQL and Qdrant backup and restore; verified on real data (12/12 routing, 5/5 SQL answers) with 323 offline tests running in GitHub Actions CI.

Projects

WildLens: Serverless ML Wildlife-Tagging Platform — github.com/mingjun6886/wildlens
AWS (Lambda, S3, SQS, DynamoDB, API Gateway, Cognito, CloudFront, SNS, ECR, CloudWatch, X-Ray), Terraform, PyTorch, GitHub Actions
- Architected an event-driven pipeline (S3 → SQS → container Lambda → DynamoDB) that tags camera-trap images and video with species and counts using a two-stage model (MegaDetector → 46-class PyTorch classifier); all infrastructure as code in 13 Terraform modules.
- Designed cost and failure boundaries: SQS with a dead-letter queue, a concurrency cap of 2 as a hard cost ceiling, a job state machine that retries only transient failures, and a time guard returning partial video results within the 900 s limit.
- Optimised by measurement: cut warm inference latency by 23% and cold-start latency from 55 s to 42 s by caching the loaded detector; reversed an earlier decision to bake weights into the image after benchmarks showed no gain; scored model changes against a hand-checked answer key (F1 0.90).
- Secured end to end: Cognito (PKCE) with an authoriser on every REST route, one least-privilege IAM role per function, hash-checked presigned uploads, and a single CloudFront distribution fronting the site and API.
- Instrumented for operations: correlation-ID logs, EMF metrics, X-Ray tracing, CloudWatch alarms to SNS, credential-free CI with OIDC, `terraform plan` and checkov, and 13 ADRs recording each trade-off.

FIREBREAK: Household Bushfire-Preparedness Web App — Monash FIT5120 • Team of 5
Vue 3, FastAPI, MySQL, Docker, AWS (EC2, RDS), GitHub Actions, OpenAI
- Delivered a deployed app on AWS (EC2, RDS) with automated CI/CD in a 14-week Agile team of 5, shaped by personas, prototypes and user feedback after each of three iterations; owned the AI features, frontend, UX and a layered FastAPI backend.
- Designed a voice assistant (GPT-Live over WebRTC) with guardrails: a closed action list, read-only browser handlers, spoken figures generated by code, and emergency phrases routed to a fixed “call 000” response.
- Built a safety Q&A where the LLM returns only IDs of reviewed CFA answers, so it cannot invent advice.