

Chan Myae Aung

Full-Stack Developer | Next.js, Node.js, Golang, PostgreSQL

- Portfolio: <https://www.chanmyaeaung.com>
 - Bangkok, Thailand
 - ☎ (+66) 955082922
 - ✉ chanaung48944@gmail.com | chanmyaeaung@my.jcu.edu.au
 - LinkedIn: <https://www.linkedin.com/in/chan-myae-aung-4434b9157/>
 - Github: <https://github.com/channmyaeaung>
-

Professional Summary

Full-stack engineer who architects, builds, and scales production systems end-to-end. 4+ years shipping 10+ applications across AI SaaS, real-time platforms, and marketplaces using Next.js, Go, Node.js, and PostgreSQL with a track record of cutting API latency by 35%, storage costs by 60%, and serverless cold starts by 800ms. Owns the full stack from database schema design to AWS deployment, shipping zero-downtime releases with obsessive performance optimization.

Technical Skills

Languages: TypeScript, JavaScript, Go, Python

Frontend: React, Next.js, Tailwind CSS, TanStack Query, Shadcn UI

Backend: Node.js, Express, FastAPI, Go

Database: PostgreSQL, Aurora PostgreSQL, MongoDB, Prisma, Redis

Cloud & DevOps: AWS (VPC, EC2, S3, Cognito, Amplify), Docker, Vercel

Auth & Payments: Clerk, AWS Cognito, JWT/OAuth 2.0, Stripe

Architecture: REST API Design, SSR/SSG, Caching Strategies, Stateless Services

Education

Bachelor of Information Technology, James Cook University, Singapore,
2019

Professional Experience

Full-Stack Engineer | Independent / Remote | 2021 – Present

- Achieved 99.9% uptime across 10+ production applications as measured by zero critical outages by architecting end-to-end solutions across Next.js frontends, Go/Node.js backends, and PostgreSQL databases.
- Slashed API p95 latency by 35% (from 800ms to 520ms) as measured by APM dashboards by deploying TanStack Query client-side caching and eliminating redundant network calls.
- Scaled backend services to support 10,000+ concurrent users with zero downtime as measured by load-testing benchmarks by architecting stateless Go and Node.js services optimized for horizontal scaling.
- Accelerated query performance by 50% as measured by average execution time by engineering normalized PostgreSQL schemas with strategic indexing across 10+ production databases.
- Cut developer environment setup time by 96% (from 2+ hours to <5 minutes) as measured by onboarding duration by containerizing multi-service stacks with Docker Compose.
- Secured 100% of protected resources with zero auth-related incidents as measured by security audit logs by deploying JWT and OAuth 2.0 flows with session rotation and Role-Based Access Control.

Selected Projects

Note: Additional Projects can be found on my Portfolio:

<https://www.chanmyaeaung.com/projects>

1. HomeScout Rental Property Marketplace

- Next.js, Node.js, Shadcn UI, AWS (Aurora PostgreSQL, VPC, EC2, Amplify, S3)

 Live: <https://main.d2sh5zhh8wrjb3.amplifyapp.com/> (AWS Free credits expired)

 Live: <https://home-scout-jet.vercel.app/>

 Demo Video Link:

<https://www.chanmyaeaung.com/projects/home-scout>

- Scaled to 50,000+ property listings with p99 query latency <150ms as measured by CloudWatch metrics by architecting stateless Node.js services on AWS EC2 backed by Aurora PostgreSQL and read replicas.
- Cut deployment time by 80% (from 15 minutes to 3 minutes) and enabled 50+ zero-downtime releases as measured by pipeline duration by engineering automated CI/CD on AWS Amplify.

2. Toshl Clone with MERN Financial Management Features

- React.js, MongoDB, Node.js, Express.js, Tailwind CSS, Clerk, TanStack Query, TanStack Table, Motion, Sanity

 Live: <https://finlytics-red.vercel.app>

 Demo Video Link: <https://www.chanmyaeaung.com/projects/finlytics>

- Built a financial tracking system with **real-time data updates and currency conversion (150+ pairs)**
- Reduced page load time by **~25%** using SSR/SSG strategies in Next.js + Sanity
- Optimized data fetching patterns to minimize client-server round trips

- Designed modular frontend state management for predictable data flow
- Deployed on Koyeb for zero down-time and faster API Performance.

3. AI Meal Plan Generator (LLM-Powered)

- **Next.js, Serverless functions, Tanstack Query, PostgreSQL, Neon, Prisma, Typescript, Tailwind, Clerk, Stripe**

 Live: <https://ai-meal-plan-generator-yjfh.vercel.app>

 Demo Video Link:

<https://www.chanmyaeaung.com/projects/ai-meal-plans-generator>

- Enabled subscription monetization and onboarded 100+ active subscribers within 60 days as measured by Stripe dashboard MRR by architecting a webhook-driven billing system with tiered RBAC.
- Reduced user manual input by 80% and maintained p95 AI response times under 3 seconds as measured by user interaction logs by deploying LLM APIs with streaming response optimization and PostgreSQL connection pooling.

4. Travel Planner - Trip Planning Application

- **Next.js, PostgreSQL, Prisma, OAuth, Google Maps API, TypeScript, Tailwind**

 Live: travel-planner-by-chan.vercel.app

 Demo Video Link:

<https://www.chanmyaeaung.com/projects/travel-planner>

- Implemented itinerary planning with location search and mapping.
- Integrated OAuth authentication (Google, Github) and optimized API calls for smoother rendering.

5. PixVault Full-Stack Media Sharing Application

- **Next.js, FastAPI, PostgreSQL, Python, TypeScript, Tailwind CSS, TanStack Query, Docker, ImageKit**

 Live: <https://pix-vault-eosin.vercel.app>

 Demo Video Link: <https://www.chanmyaeaung.com/projects/pixvault>

- Slashed image storage costs by 60% and accelerated delivery by 2x as measured by CDN bandwidth reports by deploying Pillow-based server-side compression before ImageKit CDN distribution.
- Scaled to 500+ concurrent 24MP uploads with <2% failure rate as measured by upload success logs by architecting async FastAPI endpoints with batched metadata writes and JWT-scoped access control.

6. Events Management Web App

- **Next.js, Golang, TypeScript, Tailwind, PostgreSQL**

 Live: <https://events-app-omega-two.vercel.app>

 Demo Video Link:

<https://www.chanmyaeaung.com/projects/events-app>

- Designed a relational schema for managing events, users, and participation
- Implemented stateless JWT authentication for scalable session handling
- Built APIs supporting dynamic event creation and user interaction

7. Streamly - Movie Streaming Web App

- **Next.js, Golang, TypeScript, Tailwind, MongoDB**

 Live: <https://streamly-one.vercel.app>

 Demo Video Link: <https://www.chanmyaeaung.com/projects/streamly>

- Eliminated 800ms serverless cold-start latency and stabilized p99 response times at 120ms as measured by latency benchmarks by rearchitecting the backend as a persistent-process Go service with connection pooling.
- Reduced manual moderation workload by 80% (from 25 hours/week to 5 hours/week) as measured by ops time tracking by deploying AI-assisted content rating with automated flagging and review queues.

