Kaung Hset Hein

philix.the.explorer@gmail.com | +66 822175247 | Bangkok, Thailand
LinkedIn: linkedin.com/in/kaung-hset-hein/ | GitHub: github.com/PhilixTheExplorer | Portfolio: philix.dev

Education

Bachelor of Science in Computer Science 2023 – 2027
King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, Thailand
– GPAX: 3.86 / 4.0
– Multi-intellectual Scholarship (full undergraduate scholarship, awarded 2023)

Bachelor of Engineering in Civil Engineering (Discontinued) 2017 – 2020
Yangon Technological University, Myanmar

High School Diploma 2006 – 2017
No.2, Basic Education High School, Myeik, Myanmar
– Highest matriculation score in Tanintharyi Region

Experience

NLP & Machine Learning Research Intern Jun 2026 – Jul 2026
Innovative Cognitive Computing (IC2) Lab, KMUTT, Bangkok, Thailand
– Co-authored a peer-reviewed paper on cross-modal emotion discrepancy detection in the ACM proceedings of IAIT '26 (DOI 10.1145/3816713.3818220), by designing and running the multimodal speech-and-text emotion recognition experiments behind it.
– Produced publication-ready results for an international conference submission by evaluating NLP and machine-learning experiment runs against a consistent protocol and documenting every configuration.

Student Mentor Sep 2024 – Present
School of Information Technology, KMUTT, Bangkok, Thailand (Part-time)
– Lifted code quality for annual cohorts of 50+ students by running weekly Java, object-oriented programming, and web development sessions built around clean code and SOLID principles.
– Carried two consecutive cohorts (2024/25 and 2025/26) through practical assessments by teaching systematic debugging and refactoring rather than answer-by-answer fixes.
– Kept mentoring aligned with assessed outcomes by coordinating session content directly with course faculty each term.

Freelance Software Developer 2024 – Present
Self-Employed, Remote
– Delivered client web applications end to end as sole engineer – requirements, implementation, deployment, and post-launch support – with no handoff to a separate ops team.
– Made releases repeatable and reversible by containerising each project with Docker and automating build, test, and deploy through CI/CD pipelines.
– Kept feature additions from forcing rewrites by structuring RESTful backends around clean architecture and explicit separation of concerns.
– Prioritised roadmap work against observed user behaviour by instrumenting analytics and folding client feedback into each release cycle.

Student Researcher Mar 2025 – Feb 2026
Deep Learning Research Lab, KMUTT, Remote
– Improved computer-vision model performance by applying transfer learning and targeted data augmentation, validating each change against held-out evaluation runs rather than training loss.
– Made experiments reproducible for the wider lab by documenting model configurations, evaluation procedure, and findings in publication-ready form.

Natural Language Processing Research Intern Jun 2025 – Aug 2025
Language Understanding Lab, Myanmar, Remote
– Replaced manual page-by-page transcription of Burmese documents with automated OCR and NLP pipelines built on the Google Cloud Vision API, processing scanned material in batches.
– Targeted the slowest pipeline stages by instrumenting throughput and failure-rate metrics and visualising them across runs.
– Contributed to low-resource language processing research, feeding cleaned corpora into downstream NLP work.

Publications

Kaung Hset Hein, Jonathan Chan. **Cross-Modal Emotion Discrepancy Detection: Detecting When Voice Reveals Emotions That Text Conceals**
Proceedings of the 14th International Conference on Advances in Information Technology (IAIT '26), Bangkok, Thailand, 2026, pp. 1–8. DOI: 10.1145/3816713.3818220

Projects

Lexo – *Open-source OCR workstation for Burmese documents, scanned books, and PDFs* Live | Code
Python, PySide6, Typer, PyMuPDF, Google Drive API, Pydantic, uv, GitHub Actions
– Published a reproducible, installable release to PyPI, removing long-lived API tokens from CI entirely, by wiring uv,

Hatchling, and GitHub Actions to Trusted Publishing.
- Kept a PySide6 desktop app and a scriptable Typer CLI behaviour-identical for every user by routing both interfaces through one shared processing engine instead of duplicated extraction logic.
- Cut OCR cost and latency on mixed PDFs by routing per page: pages with usable embedded text are extracted directly, and only scanned pages are rendered and sent to an OCR provider.
- Made Burmese output safe for downstream NLP and LLM pipelines by applying Unicode NFC normalization, zero-width-space-aware cleanup, a default Myanmar language hint, and bundled Myanmar fonts.
- Served human editing and machine pipelines from a single run by exporting three formats: plain text, Markdown with YAML frontmatter, and JSONL.
- Kept long documents cancellable and resilient through bounded concurrency, retry, and cancellation across cropping, spread splitting, rotation, merging, and page-range extraction.

MyanTyper – *Private, guided touch-typing practice for the Windows Myanmar Visual-order keyboard* Live | Code

Next.js 16, React 19, TypeScript 6, Tailwind CSS 4, Better Auth, Drizzle ORM, Neon Postgres, IndexedDB, Vitest, Biome, pnpm, Lefthook, GitHub Actions
- Shipped a six-track typing curriculum to production by deriving the lesson catalogue, static practice routes, navigation, metadata, and sitemap from small reviewed content modules instead of hand-maintained route lists.
- Made typing rules unit-testable in isolation by extracting key matching, Shift handling, visual-order advancement, error tracking, timing, and completion into a deterministic, framework-independent state machine.
- Modelled canonical Myanmar Unicode text, syllable segmentation, and physical KBDMYAN keystroke order as separate immutable views, correctly handling the front-vowel case where the vowel is typed before its consonant.
- Guaranteed practice never blocks on the network with an append-only IndexedDB progress log that commits locally first, then syncs cross-device through optional Google sign-in via Better Auth, Drizzle, and Neon Postgres.
- Built a full learning experience around the state machine: a fluid KBDMYAN keyboard with Shift and finger guidance, live WPM and accuracy feedback, Free Type sessions, four themes, and a bilingual learner guide.
- Locked down production routes with Content Security Policy, HSTS, a restrictive permissions policy, cross-origin safeguards, clickjacking protection, and MIME-sniffing protection.
- Blocked regressions before merge by enforcing curriculum, Unicode, keyboard, and persistence invariants with Vitest, strict TypeScript, Biome, Lefthook pre-commit hooks, and SHA-pinned GitHub Actions.

Bug Gym – *Offline Java exam practice with native desktop testing on Windows, macOS, and Linux* Live | Code

Java 21, JavaFX, RichTextFX, Maven, CommonMark, jpackage, GitHub Actions
- Shipped signed native installers for Windows, macOS, and Linux from a single codebase by automating jpackage builds in GitHub Actions on every tagged release.
- Scored student submissions automatically through an in-memory Java compilation and execution pipeline, capping each test at a five-second timeout so a runaway loop fails fast instead of freezing the application.
- Made failures diagnosable for beginners with character-level output comparison and plain compiler and runtime feedback in place of raw stack traces.
- Decoupled exercise content from application code with Markdown-based problem definitions rendered through CommonMark, so new problems are versionable and reviewable without a code change.
- Preserved fast startup and fully offline operation by keeping all progress and application state local, with no accounts or network calls.

GodView AI – *Real-time clinical anomaly detection via computer vision*

FastAPI, Python, YOLOv8, Supabase, PostgreSQL, Telegram Bot API
- Won 1st place at the International AI Hackathon 2025 (SIT, KMUTT) by delivering a working real-time patient-monitoring platform inside the competition window.
- Detected patient anomalies continuously by running a low-latency YOLOv8 inference path over live camera frames behind a FastAPI service.
- Restricted clinical data to the right staff by enforcing Supabase-backed doctor, nurse, and administrator roles on every authenticated route.
- Escalated detections to on-call staff within seconds through Telegram alerts, with defensive failure handling and operational logging so a dropped notification is visible rather than silent.
- Kept inference, identity, persistence, and notification concerns independently replaceable by structuring the FastAPI backend as separate services.

OSAS Detection System – *Leakage-aware multimodal transformer research on physiological signals* Code

Python, PyTorch, Transformers, NumPy, Pandas, Time-Series Analysis, Healthcare AI
- Converted 961,357 physician-labelled observations from 30 stroke patients into roughly 31,986 training windows by segmenting signals into 60-second windows at 50 percent overlap and discarding any window over 50 percent missing.
- Fused 14 signal streams across three modalities – 5 vital-sign features, 4 ECG/PPG waveform channels, and 5 PSG signal groups – through modality-specific transformer encoders.
- Eliminated optimistic scores caused by patient leakage by enforcing patient-aware train/validation/test splits across the 30-person cohort.
- Kept the minority class learnable under a roughly 87/13 normal-to-anomaly imbalance using weighted and focal losses, stratified sampling, and macro-F1 rather than accuracy for model selection.

– Supported binary anomaly detection, five-class event classification, and joint multitask training from one checkpointed pipeline with reproducible evaluation outputs.

thedevbox – *Privacy-first developer utilities that run in the browser* Live | Code

Next.js 16, React 19, TypeScript, Tailwind CSS, Biome, Vitest, pnpm, Lefthook

– Kept pasted payloads off the network by making local in-browser processing the default execution model, with server-side execution opt-in per tool and stated in the UI.
– Reduced adding a new tool to a single command with a scaffold that generates the component, shared logic, tests, registry entry, and route mappings together.
– Held quality constant across a growing tool set by gating merges on Biome, Vitest, Lefthook hooks, and strict TypeScript in a reproducible pnpm workflow.

SIT Face Recognition System & Photobooth – *Dual-mode recognition and event photography desktop system* Code

Python, PyQt6, OpenCV, InsightFace, FastAPI, pyttsx3, Threading

– Kept a 720p camera preview responsive during live recognition by moving capture, inference, and text-to-speech onto separate threads.
– Sped up identity matching by precomputing InsightFace embeddings ahead of each event and exposing a confidence threshold operators could tune on site.
– Contained changes to one layer at a time through a five-layer architecture – UI, orchestration, matching rules, infrastructure, data – with one-directional dependencies.
– Delivered a three-second photobooth capture flow with four selectable frame overlays, FastAPI upload, and an immediate QR code for attendees to retrieve their photo.

SIT Compendium – *Searchable knowledge system for technical coursework* Live

Next.js, TypeScript, React, Tailwind CSS, LaTeX, KaTeX, Docker

– Kept course, topic, and solution pages structurally consistent as the archive grew by enforcing a schema-driven TypeScript content model instead of free-form pages.
– Made notation-heavy material render reliably by integrating KaTeX and LaTeX-aware components for equations and derivations.
– Delivered fast, indexable navigation for the archive with server-rendered routes and REST interfaces built on the Next.js App Router.
– Made deployment reproducible from any machine with Docker images and automated Vercel builds off the main branch.

CampusLink – *Team-built university operations platform* Code

Node.js, Express, React, PostgreSQL, Tailwind CSS, MVC Architecture

– Delivered a university management platform with a student team by splitting a responsive React client and an MVC-structured Express API into independently maintained repositories behind a shared API contract.
– Implemented session-based authentication and role-aware authorization across the API surface, reviewed through a Git branch-and-review workflow.

WasteWise – *Camera-to-classification waste sorting guidance*

Flutter, FastAPI, Python, YOLO, OpenCV, Railway, REST API, Docker

– Turned a phone photo into a sorting decision end to end by wiring Flutter camera capture to a containerised FastAPI inference service deployed on Railway.
– Kept the inference service horizontally scalable by making every classification request stateless and pushing preprocessing into the request path.

Libro Tracker – *Offline-first reading tracker with optional cloud sync* Code

Flutter, Dart, Hive, Firebase, Firestore, Riverpod, GoRouter, MVVM

– Kept the personal library fully usable offline by defaulting to Hive local storage and making Firebase Authentication and Firestore sync opt-in rather than required.
– Shipped one codebase to mobile, desktop, and web targets by organising the app into feature-level MVVM modules with Riverpod state management and GoRouter navigation.

FarmLink – *Mobile agricultural marketplace*

Node.js, Express, PostgreSQL, Flutter, Cloudinary, Riverpod, REST API

– Validated the full marketplace path from role-aware onboarding to transaction-ready data by designing a normalised PostgreSQL schema for products, users, and transactions behind RESTful Node.js services.
– Kept product media fast and off the application server by delegating image storage and delivery to Cloudinary.

Thachinbook – *Type-safe publishing model for Burmese lyrics and guitar chords*

Next.js, TypeScript, React, App Router, CSS Modules

– Replaced unversioned lyric documents and social posts with a structured, searchable archive by designing an extensible song format for Burmese lyrics, chords, artists, and future multi-chord extensions.
– Delivered card-based browsing and dynamic song routes with the Next.js App Router, backed by a type-safe TypeScript content model for maintainable archive entries.
– Kept lyric and chord presentation consistent across songs through reusable frontend components built with CSS Modules.

Achievements & Awards

International AI Hackathon 2025 – Winner, School of Information Technology, KMUTT 2025
Academic Excellence Award (1st–3rd year), King Mongkut's University of Technology Thonburi 2023 – 2026
CSC 302 Talkathon – Most Creative Topic Award, School of Information Technology, KMUTT 2025
The Hackathon 2019 – 1st Runner-up, Ooredoo and UNDP 2019
National Mathematics Skills Competition – First Prize, Department of Basic Education, Ministry of Education, Myanmar 2010
8 further national-level academic prizes, including Mathematical Olympiad national Top 7, Various national competitions, Myanmar 2010 – 2017
Outstanding Award, 2B-KMUTT #12 – one of two awards given to international students, King Mongkut's University of Technology Thonburi 2015

Technical Skills

Languages:	TypeScript, Python, Java, JavaScript, Dart, SQL
Frontend & Apps:	Next.js, React, Tailwind CSS, Flutter, JavaFX, PySide6/PyQt6, MDX
Backend & APIs:	FastAPI, Node.js, Express, NestJS, REST APIs, GraphQL, Socket.IO, Typer
Data Access & ORMs:	Prisma, Drizzle ORM, Pydantic, Zod
ML & Data:	PyTorch, Transformers, YOLOv8, InsightFace, OpenCV, scikit-learn, NumPy, Pandas
Data & Cloud:	PostgreSQL, Neon, SQLite, MongoDB, Redis, Firestore, Supabase, Google Cloud, Vercel, Railway
Delivery & Security:	Docker, GitHub Actions, CI/CD, Trusted Publishing, CSP/HSTS, Vitest, Biome, Lefthook, Git, uv, pnpm, Maven

Certifications

Software Development

- Meta Full Stack Developer: Front-End & Back-End from Scratch – Meta
- Python for Everybody Specialization – University of Michigan
- Python 3 Programming Specialization – University of Michigan
- Java Programming and Software Engineering Fundamentals Specialization – Duke University
- Become a Software Developer – LinkedIn Learning
- Career Essentials in Software Development – Microsoft & LinkedIn

Machine Learning & AI

- Machine Learning Specialization – DeepLearning.AI & Stanford University
- Natural Language Processing Specialization – DeepLearning.AI
- Fundamentals of Deep Learning – NVIDIA Deep Learning Institute
- Developing Explainable AI (XAI) – Duke University
- Introduction to Machine Learning – Duke University
- Intro to Machine Learning – Kaggle

Security & Infrastructure

- Cloud Native, Microservices, Containers, DevOps and Agile – IBM
- AWS Fundamentals: Going Cloud-Native – Amazon Web Services

Communication & Delivery

- Effective Communication: Writing, Design, and Presentation Specialization – University of Colorado Boulder
- Excel Skills for Business Specialization – Macquarie University
- Project Management & Other Tools for Career Development Specialization – University of California, Irvine
- Career Success Specialization – University of California, Irvine