ALI KEMBER

FREELANCE AI PRODUCT BUILDER

ali@lks.ltd
linkedin.com/in/ali-kember
Remote · United Kingdom
Contract & freelance

My dev team is AI. I architect the system, govern the agents, and decide what ships. Fifteen years agency-side, a decade running a restaurant, and a zoology master's in between. I came back to development in 2023 into a stack from the future that barely existed when I left — creativity unleashed, from a solid foundation, which is most of why this way of working stuck.

SELECTED WORK

VENUE CRM & LEAD CAPTURE

2026 · Live in production

LONDON EXHIBITION VENUE · NEXT.JS · SUPABASE · MAILCHIMP / MANDRILL

- Migrated and consolidated 90,283 historical contacts from five legacy accounts into a single CRM, reconciled against source counts file by file, with an embeddable signup widget served as a self-contained script.
- Live since June 2026 — around 85,000 subscribers, stress-tested to 300,000.
- **Concept to production in eight weeks.**
- Data separation architected around a legally distinct entity within the venue, controlling which subscribers receive which communications.

MULTI-TENANT EMAIL PLATFORM

2026 · Beta

LK STUDIOS PRODUCT · AWS SES · SUPABASE · LISTMONK

- Per-customer sending-reputation isolation using SES tenants — 28 migrations, row-level security across 14 of them, plus an RLS regression test suite.
- Edge function verifying SNS signatures and validating subscription URLs before auto-confirm; cron reconciliation worker keeping two systems in step.
- AWS production access granted at 50,000/day. GDPR double opt-in as a product pillar, not a checkbox.

SEO & ANSWER-ENGINE MONITORING

2026 · Staging

LK STUDIOS PRODUCT · NEXT.JS · SUPABASE · FLY.IO WORKER

- Tracks whether a business appears in ChatGPT, Perplexity and Google AI Overviews — not just classic search — and produces an explainable suggestion queue.
- Vault-backed per-tenant secrets, a custom Supabase Auth hook injecting a tenant claim into the JWT, a pg-boss job queue, live Google Search Console OAuth.
- Scope deliberately argued down to a concierge bring-up: prove the collection pipeline against a real client before investing in UI.

SETTLEMENT DATA EXTRACTION & VERIFICATION

2026 · Delivered

LIVE-EVENTS PROMOTER · PYTHON · PANDAS · CLAUDE API

- Pipeline pulling settlement data out of 16K+ files containing inconsistent Excel workbooks — built as a verification tool first and an extraction tool second.
- Every figure cross-checked against source before it was trusted. The job was accuracy, accountability and reliability, not throughput. Sole developer, 509 commits.

HOW THE WORK GETS DONE

Most of my code is written by AI agent workers. My job is architecture, review, judgement and QA — so I built the layer that governs them.

- **Merge gate** — workers open a pull request and stop. A human merges.
- **Permission floors** — deny-lists blocking force-push, destructive database commands and secret reads at the permission layer, before they can be proposed.
- **Custom MCP server** — separate sessions coordinate as PM and worker, with a three-strike escalation to a human and a pause-and-ask rule that holds even when permission prompts are bypassed.
- **Design before code** — the specification is the source of truth, not the implementation. Every project is scaffolded with a fixed doc set, and design choices are recorded in a decisions log that has to be checked before anything conflicting is proposed. Architecture-heavy work triggers a dedicated design pass whose written proposal is reviewed and locked before a line is implemented.
- **Bug sweep, run to clean** — a mandatory gate before Done: manual review, a simplify pass, then a security pass, iterated against a project-specific red-flag list until a pass finds nothing. Each round is recorded in the pull request. A logic PR that arrives without that record is bounced.
- **Explicit over clever** — code reads like the code around it, every file states its purpose, and nothing fails silently: files that don't parse, queries that come back empty, migrations that no-op. Written down as standing convention and kept honest by a quality pass in the sweep rather than left to taste.
- **Refactor as you go** — leave every file you touch cleaner within the scope of the change, and raise a ticket for anything deeper rather than letting the work sprawl.
- **Negative-hypothesis review** — every review presumes a bug and has to disprove it against live data. One migration was cleared by counting 0 of 1.4 million live rows carrying the field, rather than trusting the diff.

WHAT THAT CATCHES

AI tries hard to please you, but it can't read your mind. Context is everything. It will write code that is locally correct and globally wrong — and it rarely produces nonsense, which is what makes it dangerous. It produces something plausible: code that satisfies the instruction and misses everything around it. Left unchecked, these compound — and the mess is architectural by the time anyone notices. The failures repeat, though, which means you can go looking for them. These are the ones I check for first:

CORRECT ENCODING, WRONG THREAT MODEL

Escaping and validation applied faithfully to the specification, and still unsafe where the data actually lands. The rule was followed. What happens downstream was never asked.

MULTI-STEP WRITES PRESENTED AS ONE OPERATION

Delete-then-insert, read-then-update, check-then-act. Reads as a single action and behaves like one — until something fails between the steps, at which point data is gone and nothing reports it.

SECURITY DEFAULTS LEFT OFF

Access control applied to the things being worked on rather than everything. Whatever wasn't named in the instruction keeps the platform default, and the platform default is usually open.

SILENT FAILURE

Rows dropped, batches truncated, errors caught and discarded. The output looks complete, nothing raises, and the problem surfaces weeks later as a number that won't reconcile.

GUARANTEES CLAIMED IN COMMENTS BUT NOT IN CODE

Documentation and log messages asserting isolation, scoping or validation that was never implemented. Harder to catch than a bug, because every reader after it trusts the claim and stops looking.

MISSING STATE GUARDS

Written for the happy path: runs once, in order, from a valid state. Retries, double submissions and concurrent calls weren't in the instruction, so they aren't in the code.

Further examples exist in client systems — happy to discuss, not to publish.

EXPERIENCE

LK STUDIOS — FOUNDER & DEVELOPER

2025 – present

REMOTE, UK

- Development studio building web products for hospitality, events and small business clients, alongside LK Studios' own products.
- Architecture, review and delivery across the four systems on page one.
- Data migration and consolidation as a recurring thread — 90,283 contacts from five legacy accounts, 16,000+ inconsistent workbooks, three customer databases merged into one in a separate consolidation proposal.

LOREKEEPER — CO-FOUNDER, HEAD OF PRODUCT & QA

2023 – present

- Campaign management platform for tabletop roleplaying games. Led product definition and quality through build and launch — deciding what got built and what shipped.
- Moved into development as AI tooling made it viable, and now build alongside product ownership.

THE OYSTER SHACK — GENERAL MANAGER

2015 – 2025

BIGBURY, DEVON

- Brought in to modernise and digitise the restaurant's internal operations — booking, stock, rotas and reporting — then ran the business day to day for a decade.
- The first version of the job I do now: work out how a business actually operates, then build the systems that make it work better, with people who don't think in software.
- Service, staffing, suppliers and margins in a seasonal business. Freelance web development alongside it.

FREELANCE WEB DEVELOPER

2000 – 2015

WUNDERMAN · PURPLE CREATIVE · SAINT DIGITAL · 22 DESIGN

- Fifteen years contracting agency-side. Flash to begin with, then HTML5, CSS and JavaScript as the industry moved — responsive builds and a large volume of animated advertising work.
- Mostly taking designers' PSDs and Illustrator files to production, so I know where that handover breaks and what won't survive contact with a real browser.
- Lived through one complete replatforming of how this job is done. This is the second.

EDUCATION

- **MSc Zoology** — University of Plymouth. Fieldwork in the forests of Sulawesi
- **BSc Zoology** — Roehampton, University of Surrey

STACK

NEXT.JS | REACT | TYPESCRIPT | SUPABASE | POSTGRESQL | RLS | AWS SES | TAILWIND | DATA MIGRATION | PYTHON | HTML5 / CSS / JS | MCP | CLAUDE CODE