

JARRIT HOSKING

Full-Stack Developer & AI-Augmented Product Builder

Cape Town, South Africa | 0661180036 | jarrithosking@gmail.com | linkedin.com/in/jarrithosking-44243a332

This portfolio walks through four live, production systems I've solo-designed and shipped by directing AI development tools — Claude specifically — as a genuine working method. Each one is real: live today, solving a real problem, not a demo or a tutorial project. The sections that follow go deeper into how each was actually built and why.

tripspace.global

Commission-free travel marketplace · tripspaceglobal.com

Airbnb and Booking.com take 15–25% commission on every booking, indefinitely. tripspace.global charges hosts one flat monthly fee instead. It's live, processing real bookings and real revenue today.

What I built

- The full stack, solo: React and TypeScript frontend, 30+ Firebase Cloud Functions, real payment integration through PayFast and PayPal.
- A multi-tier revenue-share system for Affiliates and Partners, including automated commission tracking and payout logic.
- A real-time internal messaging system between hosts, travellers, and support.
- A tiered Admin access system: a founder-level tier with full control, and a separate, restricted "Director Candidate" tier with hand-picked permissions per person, backed by genuine Firestore security rules, not just hidden UI.
- A lead-generation tool using the Google Places API to find real, relevant business prospects, built with hard, checked-before-and-during spend caps to keep real API costs under control.

A real problem I found and fixed

While building the tiered Admin system, I designed a Firestore security rule that — on paper — looked correct: it restricted a sensitive collection to admin-only access. In practice, it accidentally blocked the very access check meant to grant a Director Candidate their own permissions, since the rule never accounted for someone needing to read their own record. I diagnosed this myself, traced it to the exact rule, and corrected it — a genuine, non-obvious security logic bug, found and fixed through careful, systematic investigation, not luck.

Why this matters: tripspace isn't a weekend project. It has real users, real money moving through it, and a genuine security surface I'm personally responsible for getting right.

Sure Drive

Digital loyalty currency system · sure-drive-9a8c9.web.app

Built for my family's automotive repair business. The original idea was a simple car-repair savings fund for customers — but a straightforward savings product would have collided directly with existing insurance and hire-purchase credit regulation in South Africa, creating real legal and compliance risk.

The solution I designed

Instead of a savings product, I designed a digital gift/loyalty currency — "Sure Bucks" (R1 = 1 SB). Customer payments go through PayFast as normal, but the customer-facing balance is a digital token, not a cash-back guarantee. Redeemable for real services (a free wash, a free polish, a repair credit), gifted on signup, earned over time — structurally a gift and rewards system, not a savings or credit product, avoiding the regulatory conflict entirely while still delivering the same real value to customers.

What I built

- Customer-facing site and account dashboard showing real-time SB balance and reward options.
- Admin dashboard for the business: repair intake, progress tracking, and reward redemption.
- Full payment integration, with the actual money routed and held correctly while the customer-facing layer stays entirely in the token system.

Why this matters: This wasn't a coding challenge — it was a business and regulatory problem first. I had to understand what actually made two different real-world systems (insurance law, hire-purchase credit law) risky, then design an entirely different mechanism that delivered the same value without the same exposure.

Aurora Repair

AI-driven vehicle damage assessment · aurora-repair.com

An AI-driven assessment engine that bridges insurers and repairers, built to close a real gap: getting from "car is damaged" to "here's an accurate, trustworthy repair quote" typically takes real manual time and real back-and-forth.

What I built

- OCR license-plate scanning to instantly pull real vehicle details.
- Real-time parts pricing integration.
- Automated, tiered repair quoting (Economy vs. OEM parts pathways), so a customer sees real options, not one number.

Why this matters: This is a genuine AI-integrated production system, not a chatbot wrapper — real OCR, real pricing data, and real business logic working together to produce an output a real insurer or repairer can actually act on.

International Panel Shop — Check-In / Release System

intpanelshop.co.za/checkin & intpanelshop.co.za/release

Built for my family's panel-beating business to replace paper intake and release forms with a proper digital workflow — genuinely resourceful, deliberately reusing a well-established, reliable open-source library (PHPMailer) for email delivery rather than building that piece from scratch.

What I built

- **Check-in:** a digital intake form capturing vehicle and customer details, presenting real, specific terms and conditions (liability scope, repair authorization, parts disposal policy, guarantee terms), with a live, on-screen electronic signature.
- **Release:** a structured, itemized checklist (repair quality, colour match, cleanliness, belongings returned, queries addressed) paired with a customer confirmation and its own signature capture — closing the loop with a clear, written liability handoff.
- On submission, a copy is automatically emailed to both the customer and the business, creating a real, permanent record on both sides.

Why this matters: A small, complete, end-to-end operational tool — the kind of real internal system that quietly removes friction and risk from a business's daily work, exactly the type of resourceful, production-focused building I enjoy most.