

# HIT'ZERO

AN OPERATING SYSTEM FOR ALL-STAR CHEER

## *Project Dossier*

Everything built, as of April 2026

---

MAGIC CITY ALLSTARS · MINOT, ND

*Zero deductions. Zero drama. Zero excuses.*

1

# What is Hit Zero

**Hit Zero is an operating system for competitive all-star cheerleading gyms. One app. Every role. The coach runs practice. Athletes see their progress in real time. Parents stop asking. Every change is live across all three.**

Built for Magic City Allstars in Minot, ND, with a design bar set against the rest of the all-star cheer software market — an editorial dark-first visual voice that looks nothing like the SaaS platforms it's designed to replace (Jackrabbit, TeamSnap, SportsEngine Motion, QwikCut, Cheer Replay). The positioning is deliberate: premium hardware-product aesthetics, plain-English coach-voice copy, zero corporate SaaS polish.

The category-defining feature is the **AI Routine Judge** — you upload a full-out, and Hit Zero scores it against the USASF rubric, element by element, and writes coach-ready feedback. The rest of the product is what makes a gym actually switch: rosters, skill tracking, messaging, billing, medical, uniforms, leads, a public registration page — every tool a gym already cobbles together, unified.

2

# Where it lives

<b>Public URL</b>	<b>hit-zero.vercel.app</b> — basic-auth locked until public launch (username <a href="#">coach</a> / password rotated per demo)
<b>Marketing page</b>	<b>hit-zero.vercel.app/landing</b> — editorial one-pager for prospects
<b>Backend</b>	Supabase project <code>ldhzkdqznccfgpdvqyfk</code> (US East, Pro tier) — Postgres 17 + Auth + Storage + Realtime + Edge Functions
<b>PWA host</b>	Vercel (nd-ai team), production deployment with Edge Middleware basic-auth gate
<b>Mobile (planned)</b>	Capacitor wrapper ( <code>hit_zero_client/</code> ), ready to ship iOS + Android builds on top of the same codebase

**Engineering stance:** the demo and the real product run off the same React codebase. The PWA operates against an in-browser Supabase-API-compatible mock when it's offline or unauthenticated (so the demo works with zero config); swaps to the real Supabase client the moment keys are present. The AI Judge pipeline behaves the same way — real engine when the Gemini key is set, heuristic assistant engine otherwise.

3

## Brand identity

**HITZERO wordmark.** Fraunces Italic 900,  $-0.055em$  tracking, *opsz 144*. The final *O* takes a  $135^\circ$  teal→pink linear gradient. The gradient is the brand's entire color story.

<b>Ink</b>	#050507 — deep blue-black, full-bleed backgrounds
<b>Teal</b>	#27CFD7 — primary accent, cues confirmation + progress
<b>Pink</b>	#F97FAC — secondary accent, cues the <i>O</i> , celebration, the AI feature
<b>Amber</b>	#FFB454 — warnings, deductions, balance-due flags
<b>Red</b>	#FF5E6C — errors, falls, injury-severity tags
<b>Green</b>	#3FE7A0 — clean states: paid, mastered, delivered, resolved
<b>Display serif</b>	Fraunces — italic, variable, <i>opsz 144</i> for headings and scores
<b>UI sans</b>	Inter — weights 400–800 for every label and body line
<b>Mono</b>	JetBrains Mono — data, IDs, tokens

Copy voice: blue-collar, direct, never corporate. The product talks to a gym owner the way a coach talks to a parent on the drive home. Headlines are editorial serif italic; everything else is tight sans. No em-dashes acting as commas.

4

## What's in the box

Every pillar called out in the 2025–2026 cheer-software competitive analysis (Jackrabbit, Studio Pro, TeamSnap, SportsEngine Motion, QwikCut, SkillShark, Cheerletics). Tier 1 + Tier 2 are fully shipped; Tier 3 differentiation is led by the AI Judge plus several others.

PILLAR

WHAT IT DOES

STATUS

<b>USASF skill tracking</b>	45 skills × 6 categories × L1–7, one-tap status cycling, celebration ticker.	✓
<b>Roster &amp; profiles</b>	24-athlete seed roster with position, age, readiness, avatar color.	✓
<b>Scheduling + RSVP</b>	Per-session RSVP (going/maybe/no) + iCal subscription endpoint.	✓
<b>Messaging</b>	DMs, team threads, parents-only, coaches-only. Read receipts + unread badges.	✓
<b>Announcements</b>	Broadcast feed with audience targeting (all/coaches/athletes/parents).	✓
<b>Attendance</b>	Session attendance with status + note, per-athlete rollup.	✓
<b>Billing</b>	Per-athlete accounts, season total, paid, autopay, monthly charges.	✓
<b>Uniforms</b>	Catalog + per-athlete orders with fit data, status pipeline, delivery.	✓
<b>Medical + emergency</b>	Blood type / allergies / meds / insurance / physician + tap-to-call parents.	✓
<b>Injury log</b>	Occurred_at, severity, body part, return date, resolved timeline.	✓
<b>Volunteers</b>	Role catalog, per-competition assignments, claim/release by parents.	✓
<b>Practice plans</b>	Drill library + block-by-block session plans with durations.	✓
<b>Evaluations</b>	Custom rubric builder: text/score/rubric/signature/skill_ref fields.	✓
<b>Registration</b>	Public sign-up windows with fees, status pipeline, digital waivers.	✓
<b>Waivers</b>	Versioned templates + signer SVG + audit timestamps.	✓

---

<b>Leads / CRM</b>	Pipeline: new → contacted → tour → trial → converted/lost with touches.	✓
<b>Routine builder</b>	Section-by-section routine editor with count ranges.	✓
<b>Mock Score</b>	Predict competition score from current skill inventory.	✓
<b>Routine Analyses (AI)</b>	Gemini 2.5 Flash watches the full-out, scores it against USASF.	✓
<b>Push notifications</b>	on-skill-mastered edge function (APNs + FCM) on mastery transitions.	✓
<b>Realtime</b>	25+ tables in supabase_realtime — every screen updates live.	✓
<b>Role-based auth</b>	Owner / Coach / Athlete / Parent with RLS on every table.	✓

---

5

# The AI Routine Judge

**The category-defining feature. No other platform in the competitive landscape combines automated full-routine scoring with team-management workflow. This is the moat.**

**How it works end to end:** A coach drops a full-out MP4 in the AI Judge screen. The PWA uploads to Supabase Storage under a program-scoped path, then calls the `analyze-routine-v2` edge function. That function downloads the video, uploads it to the Gemini Files API, and sends a structured prompt that includes the active USASF rubric, the team's division / level / size, and the program's skill catalog. Gemini returns a strict-JSON scorecard: every element it detected, the tier (majority / most / max based on what fraction of athletes hit it), execution metrics, deductions, and judge notes.

The function then persists all of that to `analysis_elements`, `analysis_deductions`, and `analysis_feedback`, generates role-specific feedback blocks (coach-tactical, athlete-motivational, parent-friendly), and writes proposed skill-mastery flips to `analysis_skill_updates` for coach approval. Approving a flip updates `athlete_skills.status`, which fires the `on-skill-mastered` webhook, which sends a push notification and writes a celebration to the live ticker. End to end.

<b>Engine</b>	Gemini 2.5 Flash (native video ingest, no frame extraction)
<b>Fallback</b>	Deterministic heuristic engine when key is missing or video fails — same output shape
<b>Cost</b>	~\$0.01–\$0.05 per analysis on Gemini Flash; free tier covers ~1,500/day
<b>Latency</b>	30–90 seconds end-to-end per routine
<b>Output</b>	Structured scorecard, per-category breakdown, timeline of elements, ranked feedback, proposed skill flips
<b>Engine version</b>	Stamped per analysis in <code>routine_analyses.engine_version</code> — heuristic-assistant-v0 vs gemini-2.5-flash-v1
<b>Positioning</b>	Assistant Mode — coach-in-the-loop, per Grok's 'position as assistant initially' guidance

**Swap path for a future purpose-built CV model:** one function-body swap. The output contract is stable. When a MediaPipe/MMPose + skill classifier pipeline is trained, it plugs into the same rows; the UI and webhooks don't change.

6

# Backend architecture

Supabase project `ldhzkdqzncfcpdvqyfk` (previously `spas-360`, repurposed). Postgres 17 with Row-Level Security on every table.

## Schema — 50+ tables

- **Orgs:** programs, profiles (1:1 with `auth.users`), teams
- **Roster:** athletes (position, age, photo\_color), parent\_links
- **Skills:** skills (USASF catalog, 45 seed rows), athlete\_skills (per-athlete status)
- **Routines:** routines, routine\_sections, skill\_placements
- **Practice ops:** sessions, attendance, practice\_plans, practice\_plan\_blocks, drills
- **Scoring (human):** score\_runs, score\_deductions, celebrations
- **AI Judge:** rubric\_versions, rubric\_categories, routine\_analyses, analysis\_elements, analysis\_deductions, analysis\_feedback, analysis\_skill\_updates
- **Billing:** billing\_accounts, billing\_charges (Stripe-ready)
- **Communications:** announcements, message\_threads, thread\_members, messages, message\_reads
- **Schedule + RSVP:** session\_availability, calendar\_tokens
- **Registration:** registration\_windows, registrations, waiver\_templates, waiver\_signatures
- **Custom forms:** form\_templates, form\_fields, form\_responses, form\_answers
- **Medical:** emergency\_contacts, medical\_records, injuries
- **Uniforms:** uniforms, uniform\_items, uniform\_orders
- **CRM:** leads, lead\_touches
- **Volunteers:** volunteer\_roles, volunteer\_assignments
- **Media:** videos, video\_notes (per-timestamp annotations)
- **Push:** push\_tokens (iOS / Android / web)

## Edge functions

<b>analyze-routine-v2</b>	The AI Judge. Gemini 2.5 Flash + heuristic fallback. Persists elements, deductions, feedback, skill-flip proposals.
<b>calendar-ics</b>	Public .ics feed per <code>calendar_token</code> . Parents subscribe in Google/Apple Calendar, team schedule updates sync automatically.
<b>on-skill-mastered</b>	Fires on <code>athlete_skills.status</code> → 'mastered'. Fans out APNs + FCM push to athlete + linked parents + coaches. Writes celebration row.

## RLS access model

- **Owner / Coach:** full read/write within their program. Locked to `auth_program_id()`.
- **Athlete:** reads self + teammates; write only their own profile + their RSVPs.
- **Parent:** reads linked athletes only; reads own billing + announcements targeted to parents.
- **Public:** registration intake + waiver signatures (INSERT only); lead intake from marketing forms.
- **Helpers:** `auth_program_id`, `auth_role`, `is_coach_or_owner`, `is_linked_parent`, `is_own_athlete`, `is_teammate`, `can_see_athlete`, `is_thread_member` — composable across policies.

7

## *Frontend — the PWA*

React 18 + Babel (in-browser compile) for the prototype. Installable PWA with a service worker, offline-capable demo mode, Supabase-swappable for production. Same codebase feeds the Capacitor native wrapper for App Store + Play Store.

### Screens (role-filtered in the sidebar)

- **Today** — hero dashboard: team readiness, predicted score, needs-work queue, live activity feed.
- **Roster** — sortable team list with readiness bars.
- **Skill Matrix** — tap-to-cycle per-athlete skill grid.
- **Practice Plans** — drill library + block-timed session outlines.
- **Routine Builder** — section-by-section composition with count ranges.
- **Mock Score** — predicted competition scorecard from current skill inventory.
- **AI Judge** — upload, preflight, scorecard, timeline, feedback, proposal queue, trend chart.
- **Evaluations** — custom rubric builder + response gallery per template.
- **Schedule** — upcoming sessions with RSVP chips and iCal subscribe button.
- **Messages** — thread list + chat pane (DMs, team, coaches, parents).
- **Announcements** — one-way feed with audience targeting.
- **Volunteers** — per-competition role claim board.
- **Medical Hub** — owner-wide view of emergency + medical + injury across roster.
- **Uniforms** — catalog + fit sheet + per-athlete order pipeline.
- **Leads** — gym CRM pipeline (owner only).
- **Billing** — per-athlete balance + charges feed.
- **Registration** — public sign-up form, rotates active windows.
- **Athlete Drawer** — tap any athlete from anywhere → six-tab drawer (Overview / Skills / Medical / Uniform / Billing / Timeline).

**Mobile:** responsive breakpoints at 768px; sidebar collapses into a hamburger, drawer goes full-width. PWA installable from Safari / Chrome via Add to Home Screen. Theme color + masked icons already baked for native-feeling installs.

8

## *Status + what's next*

Shipped, in production:

- PWA live at hit-zero.vercel.app (basic-auth gated).
- Public marketing page at /landing.
- Supabase backend live: 50+ tables, full RLS, 3 edge functions deployed, realtime on.
- AI Routine Judge wired end-to-end to Gemini 2.5 Flash. Video upload, scoring, feedback, proposed mastery flips all functional.
- PWA cached and offline-capable via service worker.

**Next unlock (hours of work):**

- First real-video AI Judge run against a Magic City full-out, to validate Gemini output quality.
- Stripe wiring so billing accounts can actually collect tuition.
- Capacitor build — TestFlight iOS beta, Play Store internal track.
- Email triggers for announcements + evaluation auto-email.

**Known gaps (acceptable for v0):**

- AI Judge uses Gemini's Vision understanding; multi-athlete synchronization counting is approximate. Swappable for a pose-estimation pipeline without schema changes.
- No real user auth yet on the PWA (role-picker demo mode). Easy to flip: env keys are wired for the Supabase client.
- Capacitor scaffold is in place but not yet submitted to App Store / Play Store.

---

***Hit Zero · Everything a gym runs on, in one app. The AI judge the others don't have. Built for coaches, not consultants.***