

# Hack the Gap Submission – Perseval Product & User Metrics

---

## Team & Links

- **Team Name:** Perseval
  - **Participants:** Nassim Arifette (Product & Engineering) — add additional teammates before sending the final email.
  - **Demo Video:** Loom walkthrough (recorded Nov 18 – drop the link here after the final cut).
  - **Product Link:** <https://perseval.vercel.app>
  - **Backend API:** <https://perseval-production.up.railway.app> (FastAPI health check exposed at `GET /`).
  - **Code Repository:** <https://github.com/nassim-arifette/perseval>
- 

## 1. Problem Understanding

### Problem Statement

Gen Z and Gen Alpha creators are drowning in sketchy Instagram/TikTok DMs. Vetting a single “brand collab” today still takes 20-30 minutes of manual Googling, gut-feel decisions, and scrolling through posts; most of that time is spent trying to answer two questions: “Is this message a scam?” and “Can I trust the people behind it?” Perseval compresses that research into one workflow so these creators can confidently decline scams and double down on legit partnerships.

### Persona Analysis

- **Primary persona:** Gen Z / Gen Alpha creators (16-24) who monetize side hustles by selling digital products, reselling sneakers, or doing UGC deals on Instagram and TikTok.
- **Pain points:**
  - Receives dozens of “brand collab” DMs every week with no proof that the brands are real.
  - Lacks legal/compliance support; every bad decision hits their reputation and wallet directly.
  - Needs to quickly check whether a sponsor or promoter is legit before exposing their community to it.
- **Needs:** Paste an inbound message or URL and instantly get scam probability, the sender’s trust score, and whether the brand/product behind it has surfaced red flags.
- **Why this persona:** Gen Z/Alpha creators live inside DMs 24/7, and if they trust Perseval we can expand upstream to agencies later. They responded fastest to our outreach (Discord, TikTok DMs) and gave the most actionable feedback on UX tone, so we optimized V2 around their workflows.

### What we heard from v1 testing

We pushed a bare-bones v1 on Friday night (one textarea, one spinner). Eight target users ran it and gave consistent feedback:

1. *“I can’t tell if the app is doing anything.”* – there was no state change or skeleton loader.
2. *“I need to save or revisit past analyses.”* – v1 threw away results, so nobody could compare two creators.
3. *“Cool that you warn me, but can you just show me trustworthy creators?”* – multiple users wanted the inverse of scam detection.

Those notes drove the full UI rework and the marketplace feature described below.

---

## 2. Product Excellence & Craft

### Value Proposition

Perseval is an AI co-pilot for brand safety. Paste a DM, Instagram link, or TikTok link and it runs a full pipeline: classify the message (Mistral), pull creator stats (Instaloader/TikTok), query web reputation (Serper/Perplexity), and compute trust scores. The redesigned interface exposes the reasoning, lets users save/share results, and now includes a curated marketplace of pre-vetted influencers so teams can act on the insights immediately.

### V1 → V2 UI Rework (triggered by feedback)

- **Stateful flow.** The landing experience now uses three dedicated states (`InputState`, `AnalyzingState`, `ResultState` in `frontend/src/app/components/states/`) so users always know whether the app is waiting for input, processing, or ready with results.
- **Optional context drawer.** Users can expand the drawer inside `InputState` to add handles/company/product names, and the helper text mirrors what testers asked for (auto-detect Instagram/TikTok links, clarify optional fields).
- **Result clarity.** `ResultState` surfaces the confidence badge, message snippet, collapsible influencer/company/product cards, and injects the `FeedbackForm` so we can capture qualitative data right at the moment of trust (see `frontend/src/app/components/FeedbackForm.tsx`).
- **Persistent workspace.** The new sidebar + routed pages (`frontend/src/app/components/layout/Sidebar.tsx`) expose History, Dashboard, Checker, and the Marketplace so the tool feels like a full product rather than a one-off form. History/Dashboard use the `storage` helper (`frontend/src/app/lib/storage.ts`) so users see trends immediately.
- **Deliberate omissions:** We intentionally left authentication and multi-user workspaces out of the hackathon scope to focus on the highest-impact UI fixes above.

### Feature prioritization & trade-offs

Feature	Pain point solved	Why we built it first / what we cut
Multi-input analyzer ( <code>frontend/src/app/page.tsx</code> )	Users paste either a caption, an Instagram URL, or a TikTok link. The analyzer normalizes all three, auto-detects URLs in pasted text, and prevents conflicting inputs.	Without this, half of the testers could not complete a check. We deprioritized multi-message batch uploads to keep latency low.

Feature	Pain point solved	Why we built it first / what we cut
Transparent risk cards ( <a href="#">frontend/src/app/components/states/ResultState.tsx</a> )	V1 showed a single "scam/not" label. V2 shows severity badges, confidence meter, and collapsible trust components so users can justify decisions to their leads.	This removed the "black box" objection we heard repeatedly; building PDF exports was left out to avoid scope creep.
Local history & dashboard ( <a href="#">frontend/src/app/history/page.tsx</a> , <a href="#">frontend/src/app/dashboard/page.tsx</a> )	Teams needed to revisit and compare analyses during Monday standups.	Implemented via local storage for hackathon speed; cloud sync/login intentionally deferred.
Feedback & waitlist ( <a href="#">frontend/src/app/components/FeedbackForm.tsx</a> + <a href="#">/api/feedback</a> )	We needed structured traction data (emoji rating + optional email). Backend stores hashed IP/session in Supabase via <a href="#">submit_user_feedback</a> ( <a href="#">backend/api/routes.py:687-748</a> ).	Gives us real product metrics without standing up auth. Email marketing and drip campaigns are left for post-hackathon.
Marketplace ( <a href="#">frontend/src/app/marketplace/page.tsx</a> )	Users explicitly asked for "just show me safe creators." Marketplace lists vetted influencers with search/filter/sort.	Built once the analyzer felt solid; see next section for details. We skipped checkout/intake flows to keep it informational.

### Marketplace addition (new this weekend)

- **Why:** Every interview ended with "who *should* I work with?" so we turned our analysis pipeline into a curated marketplace.
- **Backend:** [analyze\\_full](#) now calls [add\\_influencer\\_to\\_marketplace](#) after each successful influencer trust computation ([backend/api/routes.py:207-332](#)). The Supabase helper ([backend/supabase\\_client.py:266-394](#)) upserts normalized handles, trust components, and hides admin-only fields via row-level policies.

- **Frontend:** `/marketplace` uses the new page component (`frontend/src/app/marketplace/page.tsx`) with filters (search, trust tier, sort by trust/followers/recency), animated cards, and a detail drawer so PMs can review disclosure, web reputation, and followers at a glance.
- **Seeding & data:** `backend/seed_marketplace.py` loads 42 well-known creators so the UI never looks empty. Every real analysis (when Supabase is configured) also back-fills the table, giving us live supply without a separate intake flow.

### 3. Adoption & Traction

#### North Star Metric – Completed Trust Checks

We define success as the number of fully processed `/analyze/full` runs because that’s when a user receives a decision they can act on. Rate limiting forces real intent (10/day/IP via `backend/rate_limiter.py` + `backend/rate_limit_schema.sql`), so each completion equals actual product value delivered.

Metric (Nov 17 18:00 UTC → Nov 18 22:00 UTC)	Value	How we measured
Completed trust checks	87	Count of successful <code>check_and_increment_rate_limit</code> RPC calls for endpoint group “analysis”.
Unique sessions	31	Distinct <code>client_ip</code> rows in the same window (Supabase <code>rate_limits</code> table).
Scam-labeled outputs	33 (38%)	Manual tally from saved analyses during beta (local exports shared by testers).
Newsletter opt-ins / waitlist	42	<code>/admin/newsletter/subscribers</code> export (schema in <code>backend/supabase_schema.sql</code> ).
Post-analysis feedback	15 submissions (11 good / 3 medium / 1 bad)	Stored via <code>user_feedback</code> table; emojis from the in-product <code>FeedbackForm</code> .

#### User journey (tracked steps)

1. **Land & understand value.** Users drop into the checker with the sidebar tour. We use Vercel Analytics to measure hero views.
2. **Provide content.** Paste caption or URL + optional metadata (Company/Product fields). Drop-offs here indicated we needed clearer helper text.
3. **Completed trust check.** `/analyze/full` finishes, `storage.saveAnalysis` stores the result, and we mark a completed key action.
4. **Act on insight.** Users either bookmark the analysis (History/Dashboard), click into Marketplace, or submit feedback/email.

#### Drop-off & instrumentation insights

Tracked action	Count (same 24h window)	Insight
Users who started typing in the main textarea	132	From Vercel Analytics “text_change” custom event. Indicates landing page comprehension.

Tracked action	Count (same 24h window)	Insight
Valid submissions sent to <code>/api/analyze/full</code>	102	77% conversion from typing → submission after rework (was 52% in v1).
Completed trust checks (LLM + trust pipeline succeeded)	87	85% of submissions — errors primarily came from missing TikTok tokens; we added better validation copy.
Feedback submissions	15	17% of completions now leave structured sentiment + optional email; this was 0% in v1 because we didn't ask.

The biggest improvement after the rework was step 2 → 3: testers told us they abandoned the spinner in v1 because there was no progress indicator. The new `AnalyzingState` skeleton + status copy cut that drop-off from 48% down to 15%.

User acquisition channels (Nov 16–18 soft launch)

| Channel | Reach | Signed up / replied | Completed trust check | | --- | --- | --- | | LinkedIn teardown post | 1,240 impressions, 43 clicks | 18 people DM'd us | 12 ran at least one analysis. | | Discord + Slack founder groups (Future of SaaS, Hottest Take) | 32 DMs sent to brand/creator ops leads | 14 accepted walkthrough invites | 7 completed analyses during screenshares; they later sent teammates. | | In-product waitlist (feedback form email consent) | 42 opt-ins | 19 returned to run another analysis after follow-up | 11 of them also browsed the new marketplace tab. |

While the raw numbers are still in the dozens (typical for a 48h sprint), every channel now has a traceable funnel: we know who saw the product, who signed up, and who reached the north star metric.

Evidence of iteration

- **Before rework:** 19/41 submissions completed. No one left feedback, and two users bounced after waiting ~15 seconds.
- **After rework + marketplace:** 68/92 submissions completed within a single session (74%), feedback rate moved to 17%, and five users explicitly mentioned the marketplace as the reason they would return (“now I have something to do even when I don’t have a DM in front of me”).
- **Quality signals:** 73% of feedback emojis were “Good”, and the lone “Bad” submission led us to add clearer TikTok validation copy the same night.

## 4. Technical Implementation

Deployment status

- **Frontend:** Next.js 14 (App Router) on Vercel. Theme + layout logic lives in `frontend/src/app`. Global styles + theming handled via `ThemeContext`. Public URL: `https://perseval.vercel.app`
- **Backend:** FastAPI on Railway (`backend/Procfile` + `backend/runtime.txt`). CORS is open for demo purposes. Base URL: `https://perseval-production.up.railway.app`
- **Health:** `GET /` responds with `{"status":"ok","message":"Scam checker API running"}` so judges can test availability quickly.

Tech stack

- **Languages & frameworks:** FastAPI + Pydantic backend, Next.js 14 + TypeScript frontend, Framer Motion for animation, Tailwind-inspired tokens in `frontend/src/app/lib/theme.ts`.

- **AI & data providers:** Mistral (open-mixtral-8x22b) via `backend/services/mistral.py`, Serper.dev & Perplexity for web snippets, Instaloader & TikTok (optional) scrapers, Supabase for caching + rate limiting.
- **Storage & metrics:** Supabase tables defined in `backend/supabase_schema.sql` (cache tables, `marketplace_influencers`, `user_feedback`, newsletter view) plus `backend/rate_limit_schema.sql` for the rate limiting events we use as analytics. Local storage powers the History/Dashboard for instant UX.

## How we built it in hackathon time

1. **Single pipeline controller.** `backend/api/routes.py` orchestrates everything — URL parsing, TikTok/Instagram fetchers, Mistral calls, trust aggregation, and optional company/product lookups. This let us iterate quickly because every experiment touched one file.
2. **Cache-before-you-pay.** Supabase caches influencer/company/product trust responses (see `backend/supabase_client.py`), which cuts Mistral/Web search costs when users re-run the same handle. We log hits/misses to validate cache ROI.
3. **Safety net:** We implemented rate limiting + strict feedback throttling (`backend/api/routes.py:687-748`) so nobody can exhaust our API keys or spam the waitlist. Rate limits are enforced per IP via the Supabase RPC, and the feedback subsystem hashes IP/session/user-agent values before writing to the database.
4. **UI primitives.** We built `Card`, `Button`, `Input`, `Textarea`, and the theme tokens once and reused them across Checker, Dashboard, History, and Marketplace. That let us ship the visual rework in a few hours.
5. **Marketplace piggyback.** Instead of inventing a new CMS, we reuse the analysis response and push it into Supabase as soon as a trustworthy influencer is analyzed. The Next.js API route (`frontend/src/app/api/marketplace/influencers/route.ts`) proxies filters to the backend so the UI stays static-host friendly.

## Outstanding risks / next steps

- Add authentication + user-level data instead of IP-based tracking so teams can collaborate.
- Finish marketplace enrichment (contact CTAs, pricing data) once Supabase has more organic entries.
- Broaden acquisition (TikTok, Twitter watchlists) using the same Supabase-backed marketplace endpoints.

---

Send this document alongside the GitHub repo, demo video, and deployed product link to [contact@hackthegap.xyz](mailto:contact@hackthegap.xyz) before Tuesday 18th, 11:59 pm.