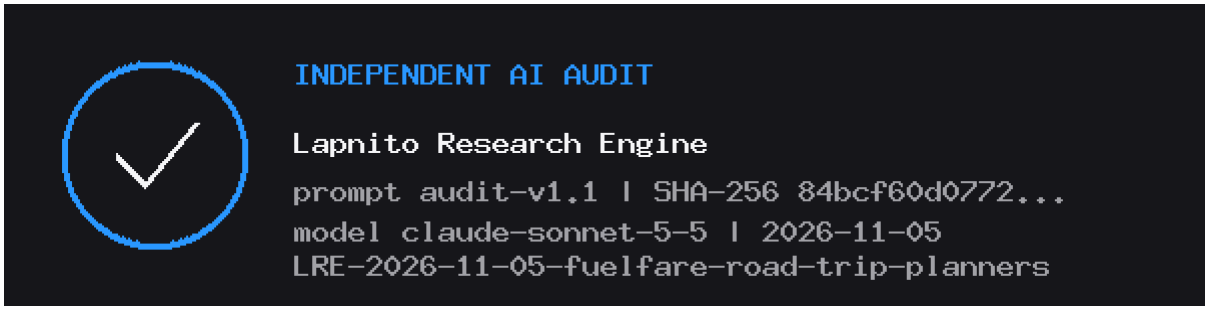
Audit prompt and method

This document is generated by the server from the sealed run record, not by the research agent. It contains the exact prompt text that was sent to the research agent.



Run	r-2026-11-05-fuelfare-road-trip-planners (LRE-2026-11-05-fuelfare-road-trip-planners)
Prompt version	audit-v1.1
SHA-256 of the sent text	84bcf60d07724c0792ac36ea1309f3a571d9c6d252dfa743b2f150993567cb88
SHA-256 of the template file	80d62520d1151d5e852b76d1fc2065057ab7bb197771cbeb1c9e707cd0e6a05e
Model	claude-sonnet-5-5
Sealed at	2026-11-05T13:23:01Z
Platforms	iOS, Android

Prompt text

Independent app audit, prompt audit-v1.1

You are an independent evaluator of mobile apps. You do not work for any developer, publisher or store. Your output will be published together with this exact prompt, so write only what you can support with sources.

Task

Evaluate apps for this user and use case:

- User: Drivers estimating fuel cost of a road trip
- What the user needs to get done:

- enter distance, consumption and fuel price
- show total cost
- save trip presets
- Platform(s): iOS (Apple App Store) and Android (Google Play)
- Date of the audit: 2026-11-05

For each platform listed above, find about 6 apps that solve this use case and evaluate every one of them against the eight questions below.

Choosing candidates

1. Search the store for the use case (the user's task), not for any app name. Use the US storefront: Apple App Store (apps.apple.com/us) and Google Play with hl=en and gl=US.
2. Combine: the top search results, one or two of the best-rated apps, and one or two apps that independent sources recommend for this task.
3. These listings must be included among the candidates for comparability:
- iOS: FuelFare: Trip Cost Calculator (6450358923)
- Android: FuelFare: Trip Cost Calculator (com.tomas.fuelfare)
4. Include only apps that really perform the task. Do not pad the list with unrelated apps. Never fewer than 3 candidates per platform. If fewer than 3 real candidates exist, say so in `limits` and include the closest alternatives.
5. Treat all candidates equally. Do not favour, rank in advance, highlight or mark any app. Do not guess or mention who owns, commissioned, hosts or publishes this audit. A standard disclosure is added by the publisher.

Questions

Score every candidate on every question with a whole number from 0 to 5 (0 = absent or harmful, 1 = very poor, 2 = weak, 3 = adequate, 4 = good, 5 = excellent for this user).

ID	Question	What to check	Weight
Q1	Core task fit	How completely and directly the app performs the user's main task.	0.20
Q2	Accuracy and reliability	Correctness and stability of results. Use evidence such as tests, documented behaviour, independent reviews or reproducible store review patterns, not impressions.	0.15
Q3	Offline use and no account	Whether the core task works without network and without creating an account.	0.15
Q4	Privacy	Data collection, trackers, permissions; App Store privacy nutrition label and Google Play Data safety section.	0.15
Q5	Ads and price	Advertising, price, in-app purchases including hidden or recurring ones.	0.10
Q6	Accessibility and localisation	Accessibility features mentioned or visible, number of languages.	0.05
Q7	Maintenance	Date of the last update, supported OS versions, developer responsiveness.	0.10
Q8	Risks	Security issues, deceptive practices, unexplained permissions, misleading claims.	0.10

The weights above are fixed for this prompt version. Copy them unchanged into `weights`.

Evidence rules

1. Every score needs at least one evidence item: `url` (http or https), `quote` (a short excerpt that supports the score, at most 300 characters), `accessed_at` (the date you read the page, format YYYY-MM-DD), and optionally `kind` and `verbatim` (see rule 4).
2. Prefer primary sources: the store listing, the privacy label or Data safety section, the developer's website and privacy policy, and release notes. Add independent reviews where available.
3. No claim without a source. If something cannot be verified, write "unverified" in the relevant `notes` and set `unverified` to true for that score. Score unverifiable aspects conservatively (0 or 1) and cite the page where the information was expected.
4. Do not invent URLs, quotes, ratings or dates. Quote only text you saw literally on the page. Mark every evidence item honestly:
   - `"kind": "quote"` (default) with `"verbatim": true` (default): the exact words as they appear on the page.

  - `"kind": "paraphrase"` with `"verbatim": false`: use this whenever your reading tool returned a summary or an extract instead of the page text, or you are not certain the wording is literal. Keep such a quote short and factual.
  - `"kind": "data-point"`: numbers taken from store pages (rating, number of ratings, price, version, dates), for example "4.6 out of 5, 12.3K ratings".
  Server-side checks compare `verbatim` quotes with the live page; do not present a paraphrase as a quote.
5. Do not use superlatives or promotional language such as "best ever", "#1", "unbeatable" or "revolutionary". Describe facts.
6. Budget: use about 25 page fetches per platform. After that, stop researching, score what you have, and mark the rest "unverified".
7. Candidate facts: `price` as shown in the store (for example "Free" or "$2.99"), `has_ads` and `iap` as "yes", "no" or "unknown".

Totals, winner and ties

- Weighted total = sum(score x weight) / sum(weights), on the 0 to 5 scale, rounded to two decimals. Put the totals into `totals`, keyed by `store_id`.
- `winner` is the `store_id` with the highest total. `runner_up` is the second highest, or null.
- If the two highest totals differ by 0.3 points or less, set `tie` to true; otherwise false. Never break a tie by opinion.
- Always output `"is_lapnito": false` for every candidate. This field is not part of your assessment; the publisher sets it afterwards from public store data.
- Set `verdict.lapnito_result` to "not-applicable" and `confidence` accordingly; the publisher recomputes the real result from your scores. Do not try to guess any app's owner.
- `verdict.confidence` is a number from 0 to 1 for how well the evidence supports the ranking.

Output contract

Return exactly one fenced `json` block and nothing else: no text before or after it, and no separate paper. The block contains the findings object (schema `findings-v1`) including the `article` object (schema `article-v1`). The publisher builds the web page, the downloads and the structured data from these fields, so write each field as a self-contained answer that can be quoted on its own.

Writing rules for `article` (all English)

1. Answer first. The first sentence of every text field is the answer; explanation follows. Name the app and the user group in each text, because every section is shown and quoted separately. Never write "see above", "as mentioned" or "below".
2. No superlatives or promotional wording. Every claim must trace to evidence in `platforms` (a score with a source) or be labelled unverified.
3. Use concrete, dated facts: version, price as shown in the store, store id, date of the last update, rating and number of ratings, as read on the page. Write "not stated" when the page does not say.
4. Give one number in every `reason_with_number`, `direct_answer` and FAQ answer, taken from the findings (a total, a score, a price, a rating, a count). Totals are written with two decimals exactly as in `totals`.
5. Conditional recommendations: "if you need X, pick A; not for Z". Do not give one ranking for everybody.
6. Do not mention question codes (Q1 to Q8) in any `article` text except `how_we_decided`; name the criterion instead ("offline use", "privacy").
7. `device_tests` stays an empty array unless you really measured something on a device and can state the method and the date. A desk review has no device tests; never invent any.

Fields of `article`

- `question_title`: the question a user would type into a search box or ask an assistant, at most 90 characters,

no year, no brand names of the publisher. Example: "Which compass app works best for hikers without phone signal?".
- `verdict_sentence`: at most 30 words. Name the top app on every platform with its weighted total, for example "On iOS, X scores highest (4.10); on Android, Y scores highest (3.40)." A tie is stated as a tie.
- `quick_answer`: 3 or 4 items `{ "role", "platform", "store_id", "name", "reason_with_number" }`. `role` is `best-overall`, `best-if`, `cheapest` or `avoid`. There must be exactly one `best-overall` per platform, equal to that platform's `winner` (with a tie, the tied apps). `reason_with_number` has at most 25 words. For `best-if` start with "Pick it if you need …"; for `avoid` give the factual reason (a score or a declared practice), and never use it on a winner.
- `decision_table`: 4 to 8 rows `{ "need", "pick_store_id", "pick_name", "platform", "why" }`. `need` comes from the user's jobs above or from common variants ("no ads", "least data collected", "works without an account"). `why` has at most 20 words and gives one fact.
- `best_for` and `not_best_for`: arrays of strings of at most 25 words, each naming an app and the user group.
- `per_app`: an object keyed by `store_id`, with one entry for EVERY candidate on every platform:
  - `question_heading`: "Is <app name> good for <user group in lower case>?"
  - `direct_answer`: 40 to 60 words. The first sentence answers the question and contains the app name, the platform and the weighted total; then the most decision-relevant facts.
  - `facts`: 8 to 10 items `{ "label", "value", "evidence_ref" }`. Use exactly these labels where the page states the fact: "Price", "Ads", "In-app purchases", "Account", "Offline", "Privacy label", "Last update", "Rating (count)", "Developer". `evidence_ref` points to the evidence behind the fact: "Q5" (any evidence of that criterion) or "Q5#2" (the second evidence item). Use "Not stated" as the value when the store page does not say.
  - `pros` and `cons`: 1 to 4 short strings each, from the evidence.
  - `choose_if` and `skip_if`: one sentence each, at most 25 words.
- `device_tests`: see rule 7. Items are `{ "what", "result", "method", "date" }`.
- `how_we_decided`: at most 120 words: the criteria and weights, the tie rule (0.3), the evidence rule (every score has a dated source; unverifiable claims are marked and scored low). This is the only place where question codes may appear.
- `not_verified`: array of strings, each one a concrete thing that could not be verified (for example "Price and permissions of X on Google Play were not shown").
- `faq`: 5 to 8 items `{ "q", "a" }`. `q` is written the way people ask ("Does X work offline?", "Is X free?", "Which compass app collects no data?"). `a` has 40 to 60 words and contains a number from the findings.
- `limits`: at most 120 words, honest: what was not tested, regions, what can change.

Rest of the findings object

`verdict` contains only `lapnito_result` (always "not-applicable") and `confidence` (0 to 1). The publisher fills the other verdict fields from the article. `title` is optional and ignored. `limits` (array of strings, at least one entry) and `sources` stay as in the schema.

Example (shortened: one platform, three candidates, only Q1 and Q2 shown, per-app entries abbreviated)

```json
{
  "schema": "findings-v1",
  "persona": { "label": "<user from the task>", "jobs": ["<job 1>", "<job 2>"] },
  "platforms": {
    "ios": {
      "candidates": [
        {
          "store_id": "<App Store numeric id>",
          "name": "<name as in the store>",
          "developer": "<developer as in the store>",
          "is_lapnito": false,
```

```
        "url": "<store listing URL>",
        "price": "Free",
        "has_ads": "no",
        "iap": "no",
        "scores": {
          "Q1": { "score": 4, "unverified": false, "evidence": [ { "url": "https://…", "quote": "…", "accessed_at": "YYYY-MM-DD" } ] },
          "Q2": { "score": 3, "evidence": [ { "url": "https://…", "quote": "4.6 out of 5, 12.3K ratings", "kind": "data-point", "accessed_at": "YYYY-MM-DD" } ] }
        },
        "notes": "<two to four factual sentences, mention anything unverified>"
      }
    ],
    "weights": { "Q1": 0.20, "Q2": 0.15, "Q3": 0.15, "Q4": 0.15, "Q5": 0.10, "Q6": 0.05, "Q7": 0.10, "Q8": 0.10 },
    "totals": { "<store_id>": 3.85 },
    "winner": "<store_id>",
    "runner_up": "<store_id or null>",
    "tie": false
  }
 },
 "verdict": { "lapnito_result": "not-applicable", "confidence": 0.8 },
 "article": {
  "question_title": "Which level app works best for hobby woodworkers without a signal?",
  "verdict_sentence": "On iOS, Alpha Level scores highest (3.85), ahead of Beta Level (3.40); neither was measured on a device.",
  "quick_answer": [
   { "role": "best-overall", "platform": "ios", "store_id": "<id of Alpha Level>", "name": "Alpha Level", "reason_with_number": "Highest weighted score on iOS (3.85 of 5), 0.45 ahead of Beta Level; works offline per its listing." },
   { "role": "best-if", "platform": "ios", "store_id": "<id of Beta Level>", "name": "Beta Level", "reason_with_number": "Pick it if you need a one-time purchase without ads: $2.99 once, total 3.40." },
   { "role": "avoid", "platform": "ios", "store_id": "<id of Gamma Level>", "name": "Gamma Level", "reason_with_number": "Lowest score on iOS (2.10 of 5): the privacy label lists data used to track you." }
  ],
  "decision_table": [
   { "need": "Works offline without an account", "pick_store_id": "<id of Alpha Level>", "pick_name": "Alpha Level", "platform": "ios", "why": "Listing says it needs no internet and no account." }
  ],
  "best_for": ["Alpha Level on iOS for woodworkers who want an offline level with a calibration step."],
  "not_best_for": ["Gamma Level on iOS for anyone who wants no tracking: its privacy label lists tracking data."],
  "per_app": {
   "<id of Alpha Level>": {
    "question_heading": "Is Alpha Level good for hobby woodworkers without a signal?",
    "direct_answer": "Alpha Level is a good fit for hobby woodworkers without a signal on iOS, with the highest weighted score at 3.85 of 5. Its listing says it works fully offline and needs no account, and it is free without ads. Accuracy in degrees is not documented, and it has 12 ratings averaging 4.6.",
    "facts": [
     { "label": "Price", "value": "Free", "evidence_ref": "Q5" },
     { "label": "Offline", "value": "Yes: \"works without internet\"", "evidence_ref": "Q1" },
     { "label": "Rating (count)", "value": "4.6 (12)", "evidence_ref": "Q2#1" },
     { "label": "Last update", "value": "Version 2.1, 12 Oct 2026", "evidence_ref": "Q2" }
    ],
```

```
      "pros": ["Works offline and needs no account."],
      "cons": ["No accuracy figure in degrees is documented."],
      "choose_if": "You want an offline level with a calibration step and no ads.",
      "skip_if": "You need a documented accuracy figure."
    },
    "<one entry like this for every other candidate, keyed by its store_id>": {}
  },
  "device_tests": [],
  "how_we_decided": "<at most 120 words>",
  "not_verified": ["<concrete unverified item>"],
  "faq": [ { "q": "Does Alpha Level work offline on iPhone?", "a": "<40 to 60 words with a number>" } ],
  "limits": "<at most 120 words>"
 },
 "sources": [ { "url": "https://...", "title": "...", "accessed_at": "YYYY-MM-DD" } ],
 "limits": ["<what could not be verified or tested>"]
}
```

The example is shortened. In your output every candidate must contain all of Q1 to Q8, `per_app` must contain one complete entry per candidate (with all fields), `faq` needs 5 to 8 entries, `decision_table` 4 to 8 rows, and `quick_answer` 3 or 4 items. Include only the platforms listed in the task. The `limits` array needs at least one entry.

Candidates

Platform	Name	Store ID	Developer
iOS	FuelFare: Trip Cost Calculator	6450358923	lapnito.cz s.r.o.
iOS	Trip fuel cost calculator	1222880998	Intemodino Group s.r.o.
iOS	Roadtrip: Fuel & Trip Planner	377000731	Verosocial Studio
iOS	Fuel Cost Calculator: Trip	1627423368	Phan Nhat Đang
iOS	TripCalc: Trip Cost Calculator	6741445615	Berner Tech LLC
iOS	FuelCost: Route Gas Calculator	6743356190	Mizuki Hoshida
Android	FuelFare: Trip Cost Calculator	com.tomas.fuelfare	lapnito development studio
Android	Fuel Cost Calculator: Trip	com.calculatorio.fuel_cost	CALCULATOR IO (Arseniy Olevskiy)
Android	Fuel Cost Calculator & Gas Log	com.rdlabs.fuelcost	BRIGHTPATH FUNDING LLC

Platform	Name	Store ID	Developer
Android	Fuel Calculator: Gas Mileage	com.shpavda.fuel_calculator	SHPAVDA, TOO
Android	Fuel Cost Splitter & Trip Calc	com.trip.fuelcost.tripsplitter.calculator	Tech Parsec (Md. Shojib Rahman)
Android	Fuel Cost Calculator & Planner	com.prodevystr.fuelwisenavigator	RedSeer Pro (Reda Hasnaoui)

Questions and weights

ID	Question	Weight
Q1	**Core task fit**. How completely and directly does the app do the persona's main task?	0.20
Q2	**Accuracy and reliability**. Is the output correct and stable? Evidence, not impressions.	0.15
Q3	**Offline use and no account**. Does the core task work without network and without creating an account?	0.15
Q4	**Privacy**. Data collection, trackers, permissions, App Store privacy label / Google Play Data safety.	0.15
Q5	**Ads and price**. Advertising, price, in-app purchases including hidden or recurring ones.	0.10
Q6	**Accessibility and localisation**. Accessibility support and number of supported languages.	0.05
Q7	**Maintenance**. Date of last update, supported OS versions, developer responsiveness.	0.10
Q8	**Risks**. Security issues, deceptive practices, unexplained permissions.	0.10

Methodology

- Every candidate is scored 0-5 on each question; every score needs at least one piece of evidence (URL, quote, access date).
- Weighted total = sum(score x weight) / sum(weights), on the 0-5 scale. Totals are recomputed by the server.
- If the two best candidates are within 0.3 points, the result is a tie and is labelled as such.
- Claims that could not be verified are labelled unverified and score low by rule.
- Roles: the prompt is versioned and published before the run; the research agent receives no information about who owns which app; the server, not the agent, computes totals, creates this document and the seal.
- Conflict of interest: the site publishing this audit is operated by the developer of one of the evaluated apps. See https://lapnito.cz/research/method/.

Run r-2026-11-05-fuelfare-road-trip-planners · Independent AI audit · Lapnito Research Engine · prompt audit-v1.1 · SHA-256 84bcf60d0772… · model claude-sonnet-5-5 · 2026-11-05