

Rebuilding what customers see first

four systems that find restaurants and cafes whose food is fine and whose pictures are not, rebuild the thing letting them down, and send it to the owner. everything here works anywhere in the world and every prompt is written out in full.

2.1 dish to ad 2.2 review gap detector and tap card 2.3 photo set rebuild 2.4 menu rebuild



the only made up picture in this guide. every other image is a real cafe, found with the searches inside and rebuilt from its own real photos.

SECTION 1 · 2 · 3 What this is, and how it works

a cafe with good food and bad pictures loses to a worse cafe with good pictures, every single day, and the owner cannot see it happening.

people decide where to eat by looking at a phone. they see the Google listing, four photos taken by customers, a star rating and a number of reviews. that is the whole shop window now, and for most independents nobody has ever deliberately built it.

so you find the ones where the gap is visible from the outside, rebuild the exact thing letting them down, and send it to them. you are not selling marketing. you are handing over a finished piece of their own shop window, already fixed.

the four systems in this guide all run on the same eight steps, and it is the same shape as every other guide in the series. only steps 2 and 5 change.

the eight steps

- 1 find** cafes and restaurants in one area.
- 2 qualify** against a visible, checkable flaw.
- 3 capture** their own real photos.

4 work out the honest numbers.

5 rebuild the thing that is letting them down.

6 put it on one page next to the original.

7 look at it yourself and approve it.

8 send it.

why food is the easiest industry to start in

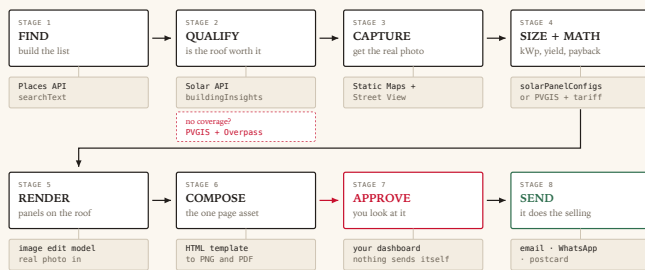
the flaw is **public and countable**. Google will tell you, for free, how many photos a place has, how many reviews, and what its rating is. you do not need to guess who needs help. you can sort a whole town by it.

and the owner already knows. nobody who runs a cafe thinks their phone snap of a breakfast is good. they just have never had ten spare minutes and did not know it could be fixed for the price of a coffee.

one warning before you start. this industry is the most pitched to on earth. every cafe owner gets three messages a week from someone offering social media management. the only thing that separates you is that **you did the work first**. never send a message that describes what you could do. send the finished thing.

The pipeline, stage by stage

left to right, one box per stage. the data source that feeds each stage sits underneath it.



every stage writes its result back to one row in the database. a stage that fails leaves the row where it was, so the next run picks it up again and carries on from there.
stage 7 is a person, not a job. the send stage refuses to read a row that nobody approved.

Fig 1. the eight steps. the grey box under each one is where the data comes from.

the rule that keeps you out of trouble

never change what the food actually is. you can relight it, reframe it and reshoot it. you cannot add a garnish that was not on the plate, make the portion bigger, or swap the crockery. the moment a customer orders it and it does not match, you have done the owner real harm and they will know exactly who to blame. the same rule covers the menu: the items and the prices are theirs, and you may not quietly correct either.

the example business, and it is real

Enclave Coffee and Bakehouse, 18 Wostenholm Road, Nether Edge, Sheffield S7 1LJ. found with the search on the next page. **5.0 stars from 14 reviews**, seven photos on the listing, and its menu is two handwritten chalkboards. three of the four systems in this guide are demonstrated on this one real cafe, using its own real Google photos.

SECTION 6 What you need, and how to find the right cafes

four accounts. two need a card. this is a cheaper stack than the solar guide because there is no expensive measurement api, only photos and text.

ACCOUNT	WHAT IT IS FOR	FREE TIER	CARD
Google Cloud console.cloud.google.com	finding the cafes, and pulling their existing photos. switch on Places API (New) only.	10,000 free calls a month per api tier. the search itself sits in the cheap band.	yes
an image model aistudio.google.com or replicate.com	relighting the food, the shopfront and the menu	none. pay per image. the pictures in this guide were made with openai/gpt-image-2.	yes
Resend · resend.com	sending it	3,000 a month, 100 a day, one verified domain	no
Instagram , a real account	most cafes reply on instagram faster than on email. you need an account that looks like a person.	free	no

STEP 1 · FIND THE CAFES AND READ THEIR FLAWS IN ONE CALL

```
curl -X POST 'https://places.googleapis.com/v1/places:searchText' \
-H 'Content-Type: application/json' \
-H "X-Goog-API-Key: YOUR_KEY" \
-H 'X-Goog-FieldMask: places.id,places.displayName,\
places.formattedAddress,places.rating,\
places.userRatingCount,places.websiteUri,places.photos' \
-d '{"textQuery":"independent cafe in Sheffield England",
  "pageSize":20}'
```

that one call gives you the whole shortlist and the reason. userRatingCount tells you who has a review gap. the length of the photos array tells you whose photo set is thin. websiteUri being absent tells you who is running the business off a Google listing alone.

searches worth running, one town at a time: independent cafe in {{TOWN}} · family run trattoria in {{TOWN}} · brunch in {{TOWN}} · bakery in {{TOWN}} · curry house in {{TOWN}}. avoid chains, they have a head office and you will never reach a decision maker.

what a real result set looks like, and who to pick

this is the actual output of the search above, run once, unedited:

CAFE	RATING	REVIEWS	PHOTOS
Tamper Coffee	4.6	1,972	10
Steam Yard	4.7	1,457	10
Enclave Coffee and Bakehouse	5.0	14	7
CAWA Coffee	4.9	17	5
The Cozy Cauldron	5.0	20	10

you want the bottom three, not the top two. a 5.0 rating from 14 reviews means the food is genuinely good and almost nobody has been told. a 4.6 from 1,972 means they are already winning and do not need you.

the qualifier, and it is one line of code

keep a cafe if **the rating is 4.5 or higher** and **the review count is under about 150**. that combination is the whole business. it says the product is good and the shop window is empty, which is the only situation where your work changes anything.

then add the specific flaw for whichever system you are running:

2.1 dish to ad · at least one food photo on the listing.

2.2 review gap · under 150 reviews, and open more than a year.

2.3 photo set · fewer than 10 photos.

2.4 menu rebuild · no website, or a menu that is only a photo of a board.

SYSTEM 2.1

dish to ad

take the phone snap a customer left on their Google listing and hand it back as a photograph they can actually run an advert with.

SECTIONS 4 · 5 · 7 What it does, who pays, and the steps

who pays you. the owner, directly. an independent with one site, where the person who answers the phone is the person who decides. avoid anything with a head office.

why it lands. a cafe owner has looked at their own Google photos and winced. they know the breakfast looks grey. they have simply never had the ten minutes or the equipment, and they assume fixing it means hiring a photographer for a day. you are showing them it takes one photo they already own.

what they do instead. nothing, usually. or they pay a social media manager a monthly retainer to repost the same grey photos with more hashtags.

the steps that differ

2 qualify · rating 4.5+, under 150 reviews, and at least one photo on the listing that is clearly food.

3 capture · pull their existing photo straight from Places. you need the photo name from the details call, then one more call for the image itself.

```
# the photo name comes back in the photos[] array.
# you cannot cache it and it expires, so fetch it
fresh.
curl -L -o before.jpg
"https://places.googleapis.com/v1/\
places/PLACE_ID/photos/PHOTO_RESOURCE/media\
?maxWidthPx=1200&key=YOUR_KEY"
```

5 rebuild · relight and reframe, never restyle. the prompt is in section 8 and the rule it enforces is the one on page 1: the food on the plate may not change.



Before. a real customer photo from a real Sheffield cafe listing. flat overhead light, tilted, half the frame is empty table. nothing wrong with the food.



After. same plate, same items, same crockery. what changed is the light, the height and the crop. **this is the whole system.**



Wrong. steam that was never there, herbs nobody ordered, a studio backdrop, and colours no breakfast has ever had.

why the bad one loses you the client

it is not that it looks fake, though it does. it is that a customer will order the breakfast, get the real one, and feel lied to. the owner then wears a one star review that you caused.

the test: could the kitchen actually plate this, today, exactly as shown? if not, you have gone too far. relight all you like. never add food.

SYSTEM 2.2

review gap detector and tap card

find the cafes whose rating is excellent and whose review count is tiny, then hand them the physical thing that fixes it.

this is the best system in the guide and the one to build first. it needs no image model at all, the qualifier is a single number, and what you sell is a physical object rather than a service, which makes it far easier to charge for.

the problem, precisely. Enclave has **5.0 stars from 14 reviews**. Tamper Coffee down the road has 4.6 from 1,972. on a phone, Tamper wins every time, because 14 reviews reads as unproven no matter how perfect the score. the food is not the issue. the count is.

why it stays broken. asking for reviews is awkward and the moment of maximum goodwill, just after someone has enjoyed the food, is exactly when the owner is busiest. the ask never happens.

what you sell. a small NFC card that sits on the counter. the customer taps their phone against it and their review page opens, no app and no typing. iPhone XS and newer, and any Android with NFC on, open it automatically on a tap. every card also carries a printed QR code so older phones still work.

how to set one up. get the cafe's Google place id from the search you already ran, build their review link as https://search.google.com/local/writerreview?placeid=PLACE_ID, and write that url onto a blank NTAG213 card with any free NFC writer app on your phone. that is the entire technical job. it takes about two minutes per card.



What you actually hand over. a blank NFC card, programmed to their own review page, in a small stand for the counter. no app, no subscription, no monthly anything for the cafe.

the pitch writes itself, because the numbers are public

you are not guessing. you can tell the owner their exact rating, their exact review count, and the exact count of the competitor two streets away who is beating them. all three are on a public listing, and none of them are your opinion.

the line you must not cross

the card opens the review page. that is all it does. it must not filter people by how happy they are, it must not send unhappy customers somewhere private instead, and it must never offer anything in exchange for a review. review gating breaks Google's policy and can get the listing penalised, which would make you the person who cost them their listing. sell the tap, not the outcome.

SYSTEM 2.3 · 2.4

photo set rebuild, and menu rebuild

the other two things on a Google listing that everybody sees and nobody has ever deliberately made.

2.3 · the photo set rebuild

a listing with five photos, all taken by customers, all in flat afternoon light, is doing the cafe active harm. the fix is not new photography. it is **relighting and reframing what is already there**, then handing back a set of four that belong together.

qualify on fewer than ten photos on the listing. that number comes free in the search you already ran.

the four shots a cafe listing needs, in this order, because this is the order people swipe: the front of the shop so they can find it, the room so they know what they are walking into, the counter or the cake case so they know what is on offer, and one dish. you can usually build all four from photos the listing already has.

the same rule applies. relight, straighten, crop, clean up litter. do not remove the neighbouring shop, do not invent outdoor seating, do not change the signage. it has to be the same shop on the same street or the customer will not recognise it when they arrive.



Before. the real photo on Enclave's listing. flat grey light, crooked, and a third of the frame is empty road.



After. same shop, same street, same neighbours, same tables. late light, straightened, cropped in.



Before. Enclave's real menu, and it is genuinely charming in the room. it is also the only menu they have, which means a customer deciding on a phone cannot read it at all.

SANDWICHES		ALL DAY BREAKFAST	
TUNA MAYO	3.75	• BEANS ON TOAST	4.50
TUNA RED ONION MAYO	3.85	• JAM ON TOAST	3.50
TUNA SWEET CHILLI CHEESE	3.95	• AVO, SCRAMBLED EGG & TOAST	6.50
CHEESE	3.45	• PEANUT BUTTER & BANANA ON TOAST	3.50
CHEESE RED ONION MAYO	3.75	• PORRIDGE-BANANA & GOLDEN SYRUP	3.50
PRAWN MAYO	3.95	• PORRIDGE, BLUEBERRY & HONEY	3.85
PRAWN COCKTAIL	3.95	• SCRAMBLED EGG TOAST	5.50
PLOUGHMAN'S	4.10	• FULL BREAKFAST (incl. hot, cold, drink)	7.00
TURKEY HAM	3.95		
EGG MAYO	3.65		
ADD SALAD	1.00		
		JACKET POTATO	
		2+ TOPPING	5.00
		3+ TOPPING OR SPECIAL	6.00
		PASTA SALAD BOX (incl. hot, cold, drink)	3.95
		1+ TOPPING 4.95	HOT FILLING 5.95

After. every item and every price exactly as written on the board, set so it can be read on a phone and printed on A4. **you must proofread this line by line. see the warning.**

2.4 · the menu rebuild

a huge number of independents have no readable menu anywhere online. the menu exists as a chalkboard on the wall, and the only version on the internet is a customer's photo of that board taken at an angle.

this matters more than it sounds. people decide whether to come *before* they arrive, and a menu they cannot read is a decision they make somewhere else. it is also the single easiest thing to fix, because the content already exists and you are only setting it.

qualify on no website, or a website with no menu page, or a menu that is only an image. all three are visible from the search you already ran.

read this before you send a menu

image models transcribe handwriting **imperfectly**. a price will change. an item will be dropped. a word will be invented. this is not a small risk, it is the normal outcome, and a wrong price on a menu is a real problem in a real shop.

so the rebuild is a draft, always. read every line against the original photo yourself before it goes anywhere near the owner, and say plainly in the message that they must check it. i would rather you sent it with a line saying "check my transcription" than sent it looking finished.

which of the four to build first, if you only build one

2.2, the review gap. no image model, no subscription, a qualifier that is one number, and a physical object at the end that an owner understands instantly. **2.1, dish to ad, is the best demo** and the one most likely to get a reply, so build it second and use it as the thing you attach. 2.3 and 2.4 are what you sell *after* they have said yes once, which is a much easier conversation than the first one.

SECTION 8 The prompts

five prompts, complete, nothing to fill in except the `{{double braces}}`. the right prompts are long because almost all of their work is telling the model what it is *not* allowed to change. that restraint is the entire product.



in · their own photo



out · the dish prompt



in · their shopfront



out · the storefront prompt

STEP 2 · PICK THE CAFES WORTH WORKING ON · ANY CHEAP TEXT MODEL. PASTE THE PLACES RESULT IN AT THE BOTTOM.

You are screening cafes and restaurants for a service that rebuilds the pictures on their Google listing. You will be given one place record from the Google Places API. Decide whether this business is worth building something for.

Apply these rules in order and stop at the first one that fails:

1. REJECT if rating is below 4.5.
Reason: "the food is the problem, and i cannot fix that".
2. REJECT if userRatingCount is above `{{MAX_REVIEWS}}` (start at 150).
Reason: "already established, they do not need me".
3. REJECT if userRatingCount is below 5.
Reason: "too new or too quiet to tell whether it is any good".
4. REJECT if the name contains any sign of a chain: Ltd followed by a number, a branch name in brackets, or a word that also appears in a national brand. When unsure, keep it and flag it.
Reason: "chain, there is a head office and no reachable decision maker".

If nothing rejects it, ACCEPT, and then work out which of the four systems this business needs most. Set each flag independently, because a business can need several:

- needs_dish_work: true if any photo on the listing is food.
- needs_review_push: true if userRatingCount is under 150. This is almost always true and it is the easiest thing to sell, so say so.
- needs_photo_set: true if the photos array has fewer than 10 entries.
- needs_menu: true if websiteUri is missing entirely.

Then pick ONE to lead with, in this priority order, and put it in lead_with: review_push first, then dish, then photo_set, then menu. Lead with the review gap because it is the cheapest thing for them to say yes to.

Return ONLY this JSON object and nothing else. No explanation, no code fences.

```
{
  "decision": "accept" | "reject",
  "reason": "one short lowercase sentence",
  "rating": number,
  "review_count": number,
  "photo_count": number,
  "needs_dish_work": boolean,
  "needs_review_push": boolean,
  "needs_photo_set": boolean,
  "needs_menu": boolean,
  "lead_with": "review_push" | "dish" | "photo_set" | "menu",
  "one_line_for_the_owner": "one plain sentence, lowercase, stating the single most useful fact
                             about their listing. no adjectives, no flattery, no pitch."
}
```

Never invent a number. Every figure must come from the record you were given.

Here is the place:

`{{PLACE_JSON}}`

STEP 5 · RELIGHT ONE DISH · GEMINI-3-PRO-IMAGE OR GPT-IMAGE-2. SEND THEIR REAL PHOTO WITH IT.

Relight and reshoot this real photograph of {{DISH_NAME}} so it looks like it was taken by a food photographer, while keeping it recognisably the same plate of the same food.

Keep every item on the plate exactly as it is and in the same arrangement. Do not add, remove, restyle or rearrange any food. Do not change the crockery, the cutlery or the table. Do not add garnish, herbs, sauce, steam, or props of any kind. If something looks slightly untidy on the plate, leave it untidy.

What you may change is only this: the light, the camera position and the crop.

Move to a soft directional light coming from one side with a gentle falloff, so the food has shape and the shadows are soft rather than the flat overhead glare in the original. Bring the camera lower and closer so the plate fills more of the frame. Let the background fall softly out of focus. Straighten the horizon. Keep the colours natural and true to the actual food, and do not increase saturation.

The result must look like an honest photograph of this exact dish on a better day, taken by someone who knew what they were doing. It must not look like a stock image or an advert.

No text, no logos, no borders, no watermarks.

STEP 5 · RELIGHT THE SHOPFRONT OR THE ROOM · SAME MODELS, THEIR REAL PHOTO

Relight and reshoot this real photograph of {{PLACE_NAME}}, a small independent cafe at {{ADDRESS}}, so it looks like it was taken by a professional, while keeping it unmistakably the same place.

Keep the building, the shopfront, the signage, the windows, the door, the furniture, the fittings and the neighbouring shops exactly where they are and exactly as they look. Do not restyle the place, do not change or clean up the sign, do not add plants, furniture, people or lighting that are not already in the photograph, and do not remove the neighbouring buildings.

What you may change is only this: the light, the framing and the tidiness of the ground.

Move it to warm late afternoon light coming along the street, so the front catches the sun and there are long soft shadows, instead of flat overcast grey. Straighten the verticals and level the horizon. Crop tighter so the building fills the frame and empty road or empty ceiling is reduced. Lift the shadows so any dark signage reads clearly. Remove litter from the pavement.

Keep it photographic and honest, like a good local press photograph. Someone who has been to this place must recognise it instantly.

No text changes, no added logos, no watermarks.

STEP 5 · REBUILD THE MENU · SEND THE PHOTO OF THEIR BOARD OR THEIR MENU

Read the menu in this photograph and set the SAME menu as a clean, modern, printed menu page on a warm off-white background, photographed flat from directly above with soft even light.

Transcribe every single item and every single price exactly as they appear. Do not invent items, do not drop items, do not round or alter any price, do not reword anything, and do not correct spelling or capitalisation. If a word is genuinely unreadable in the photograph, write it as XXXX rather than guessing, so that a human can find it and fix it.

Keep the same sections in the same order with the same headings, and keep each item under the heading it appears under in the original.

Set the section headings in a clean contemporary serif and the items in a simple sans serif, with generous line spacing and prices right aligned in a neat column. Plenty of white space, and a narrow rule under each section heading. No decoration, no icons, no photographs, no logo, no border.

It should look like a well made independent cafe menu: calm, readable and plain. Nothing else in the frame.

the XXXX instruction is the important line

without it the model fills gaps with something plausible, and a plausible wrong price is far more dangerous than an obvious blank. with it, the errors announce themselves and you can fix them in thirty seconds. keep that line in even when the handwriting looks clear.

and proofread anyway

read the rebuilt menu against the original photograph line by line, out loud if you have to, before it leaves your machine. this is thirty seconds of work that prevents the one mistake in this guide that could actually cost a business money.

STEP 8 · WRITE THE MESSAGE · ANY TEXT MODEL

Write the message that goes with a rebuilt picture being sent to the owner of a specific cafe. The picture does the selling. Your job is the few lines around it.

Facts you may use, and you may not use anything else:

- cafe name: {{PLACE_NAME}}
- what you rebuilt: {{WHAT}}
- their rating: {{RATING}} from {{REVIEW_COUNT}} reviews
- number of photos on their listing: {{PHOTO_COUNT}}
- the channel: {{CHANNEL}}, either instagram dm or email

Rules:

- Under 55 words.
- All lowercase is fine and preferred. Everyday words, the way one person messages another.
- No em dashes. Use full stops and commas.
- Open by saying what the attached picture is and that it is their own photo. Do not open with a greeting formula, a compliment about the food, or "I came across your page".
- State one real number about their listing, once. Never more than one.
- Say plainly that it is free and that they can use it whether or not they reply. Mean it.
- Do not offer a call, a meeting, a strategy session, an audit, or a package.
- End with one easy question that can be answered in one word.
- Never use: unlock, elevate, transform, boost, level up, game changer, passionate, curated, hidden gem, I hope this finds you well, quick question, just following up.
- Do not claim you have eaten there, do not claim to be local unless you are, and do not imply you already know them.

Return only the message text.

what that gives you for Enclave, on instagram:

this is your own breakfast photo off your google listing, relit. same plate, nothing added. you are on 5.0 from 14 reviews, which is the only thing holding the listing back. yours to use either way. want the other three?

When you outgrow the spreadsheet

for this guide you can go a very long way on a spreadsheet, because the qualifier is one number and you are only ever working on a handful of cafes in one town at a time. build the app when you have a client paying monthly, and not before.

three screens. /prospects is the list sorted by review count ascending, so the biggest gaps are at the top. /prospects/[id] puts their original photo and your rebuild side by side with the message in an editable box. /sends is one row per send, never overwritten.

three tables. places is one row per cafe with its rating, review count and photo count. assets is the before, the after and the message, with a status that starts at pending. sends is one row every time something goes out. you do not have to write the sql yourself.



2.2 · the tap card

SANDWICHES		ALL DAY BREAKFAST	
TUNA MAYO	3.75	• BRANZO ON TOAST	4.50
TUNA RED ONION MAYO	3.85	• JAM ON TOAST	3.50
TUNA SWEET CHILLI CHEESE	3.95	• AYO, SCRAMBLED EGG & TOAST	6.50
CHEESE	3.65	• PEANUT BUTTER & BANANA ON TOAST	3.50
CHEESE RED ONION MAYO	3.75	• PORRIDGE - BANANA & GOLDEN SYRUP	3.50
PRAWN MAYO	3.95	• PORRIDGE - BLUEBERRY & HONEY	3.85
PRAWN COCKTAIL	3.95	• SCRAMBLED EGG TOAST	5.50
PLOUGHMANS	4.10	• FULL BREAKO SUN ON TART: HEN	7.00
TURKEY HAM	3.95	JACKET POTATO	
EGG MAYO	3.65	2+ TOPPINGS	5.00
ADD SALAD	1.00	3+ TOPPING OR SPECIAL	6.00
SAUSAGE	3.85	PASTA SALAD BOX PLAIN	3.95
SAUSAGE & EGG	3.95	1+ FILLING 4.95	HOT FILLING 5.95
FRIED EGG	3.50		

2.4 · the menu

The three screens you need

anything beyond these three is you assuming the part where you send it.

1 - /prospects

the list

all - queued - reviewed - waiting - sent - dead

Selco Builders Warehouse, Bristol

5,875 sq roof - 2,382 panels

view info

Rua De Antonio Martins, Enxal

288 sq roof - 61 panels max

view

Longacre Garden Centre, Ashford

6,115 sq roof - 2,128 panels

view

2 - /prospects/[id]

preview and approve

BEFORE

the real photo

AFTER

the rebuild

6.6 sqp - 24 panels - 16,250 kWh a year

summary Dec 2024 - Google Scholar 10% - 100%

the message, actually, right here

so you can be one word without a message

approve and send

reject

redo sender

3 - /sends

the log

date	who	channel	status
01-04	Selco Bristol	email	delivered
01-04	Longacre Ashford	email	bounced
01-05	Rua's villa	whatsapp	read
01-06	Rua's Gap Bristol	postcard	in transit
01-06	Cassini Birmingham	email	SMTP 250

This screen is the only panel the system works. A bounce is a dead address, not a dead prospect. So the address and send it right.

one row per send, one: never overwritten one, or you have the only record of what you actually said.

The three screens, and nothing else. the only real difference from the other guides is that the list is sorted by *review count ascending*, because in this industry the smallest number is the best prospect.



2.1 · the dish



2.3 · the frontage

Set up Supabase for this project using the Supabase CLI, and write the SQL migration for me.

First, if it is not already set up: install the CLI, run `supabase init`, then `supabase link` to my project, and tell me exactly what to paste when it asks for my project ref and database password.

Then create a migration with three tables. Use uuid primary keys with gen_random_uuid() as the default, and give every table a created_at that defaults to now().

TABLE 1, places. One row per cafe or restaurant I am considering.

- owner_id, text, not null. This is my Clerk user id.
- place_id, text, unique, not null. The Google Places id.
- name, text, not null. address, text, not null.
- website, text, nullable. Null means they have no site, which is a qualifier in itself.
- rating, numeric. review_count, int. photo_count, int.
- needs_dish_work, needs_review_push, needs_photo_set, needs_menu, all boolean.
- lead_with, text. One of 'review_push', 'dish', 'photo_set', 'menu'.
- raw, jsonb. The entire Places response, kept as is.
- status, text, not null, default 'new'.

TABLE 2, assets. One row per thing I rebuild for a place.

- place_id_fk, references places, and delete it if the place is deleted.
- kind, text. One of 'dish', 'storefront', 'room', 'menu', 'tapcard'.
- before_url and after_url, text. Where the two pictures live in storage.
- body, text. The message that goes with it.
- model, text, and prompt, text. Which image model made it and the exact prompt used.
- proofread, boolean, not null, default false. For menus this must be true before it can be approved, because menu transcription errors are the one mistake that costs a real business money.
- status, text, not null, default 'pending'. Only 'pending', 'approved' or 'rejected'.

TABLE 3, sends. One row every single time something goes out. Never updated, only inserted.

- asset_id, references assets, and delete it if the asset is deleted.
- channel, text. One of 'instagram', 'email', 'walk_in'.
- to_handle, text, not null. status, text, not null, default 'queued'.
- sent_at, timestampz, nullable.

Now the security. Turn on row level security on all three tables, then write policies so I can only ever see and change my own rows. places has owner_id on it directly, so compare that against the Clerk user id, which Supabase exposes as auth.jwt() ->> 'sub'. assets and sends have no owner column of their own, so their policies have to join back up to places to work out who the row belongs to.

Add one more thing: a check constraint or a trigger so that an asset of kind 'menu' cannot be set to status 'approved' while proofread is still false. I want the database to stop me sending an unchecked menu, not my own memory.

Also create a private storage bucket called 'assets' for the pictures, with a policy that only lets me read files inside a folder named after my own user id.

Finally, add the indexes that matter: I will be listing places by owner and status, sorting them by review_count ascending, and looking up assets that are still pending.

Explain in plain English what each policy does before you write it, run the migration with `supabase db push`, then show me one test query proving a second user cannot read my rows.



What you actually hand over. a physical object, no subscription, and the easiest yes in this guide.

how to charge, and this industry is different

cafes have thin margins and are pitched to constantly, so a monthly retainer is a hard sell and a fast churn. **sell things, not services.**

the tap card is a product. one price, they own it, no monthly. it is the easiest yes in this guide and it gets you in the door as someone who delivered a physical object that worked.

the photo set is a job. a fixed price for a set of four, done once. offer the first one free, always, because that is what makes the message land at all.

the menu is a job with a warning attached. price it as a rebuild, hand it over as an editable file as well as a picture, and make them sign off the transcription.

the monthly, if you ever get one, is a fresh dish photo every week for their social. that is worth paying for continuously because it runs out. photo sets and menus do not.

run it yourself, always

handing this over does not work. someone has to look at every rebuild and decide whether the food still looks like the food, and a cafe owner will not do that about their own dish. keep the system, sell the output.

country notes

this guide degrades less than any other in the series, because Google Places covers restaurants almost everywhere on earth and a photo is a photo in every language.

what changes is the channel and the platform. in most of latin america, india, indonesia, nigeria and kenya, **whatsapp is how a cafe is reached**, and a business number on the listing is normal. in japan and korea, use email and expect a slow, formal reply. across east and southeast asia the review platform that matters may not be Google at all, so check which listing the locals actually use before you build a qualifier on review counts.

before you send the first one

- [] the Google key is **restricted** to the Places API only, with a budget alert on the project
- [] you have rebuilt **at least ten dishes** and looked at every one, so you know when relighting has tipped into restyling
- [] every menu you have made has been proofread line by line against the original photo
- [] your instagram account looks like a person, not a service. a blank account with a logo gets ignored
- [] you can name the cafe getting the first one, and say what its exact review count is

what breaks

the photo name expires. Google says plainly that you cannot cache a photo name and that it can stop working. fetch it fresh from a details call every time, or your pipeline will start failing on prospects that were fine last week.

the model restyles instead of relighting. it will add a garnish, tidy a plate, or swap a chipped mug. this is the failure that matters and you catch it by holding the two images side by side, not by reading the prompt again.

nobody answers email. in food, instagram dm beats email by a wide margin, and walking in at 3pm on a Tuesday with a printed photo beats both. this is one of the few industries where the offline version of the send is genuinely the strongest.

what i have not tested

the Places search, the photo pull and every rebuild in here were run against the live api on the real cafes named. i have **not** sent any of it to those businesses, programmed a physical tap card, or measured a reply rate. Enclave, CAWA and The First Sip are real businesses used as public examples and have no idea this guide exists.

the one legal line worth knowing, everywhere

you may relight a photograph that a customer took and posted to a public listing, and hand it to the business it depicts, as a demonstration. you should **not** republish it yourself, sell it to anyone else, or strip the attribution and claim it. the safe version, and the one i would stick to, is: rebuild it, give it to the owner, tell them where it came from, and let *them* decide what to do with it. that also happens to be the version that makes them like you.