

FREE GUIDE

Instagram Lead Gen Build It Yourself

**A free, step-by-step guide to an Instagram
lead-finding system built entirely with
Claude Code**

Fabian Markl

FabiMarkl.com

What you're building

A system that finds Instagram leads in any niche every day, finds their email addresses, scores them, writes a personalized email and DM for each one, and shows all of it in a dashboard you open with one click.

You do not need to know how to code. Claude Code writes every file. You paste prompts and answer questions. That's the whole job.

What's inside this guide

- ✓ The exact prompts to paste, in order, to build the whole system
- ✓ The exact email and DM templates, word for word
- ✓ The scoring formula and the rules behind every message
- ✓ A cost breakdown with real numbers, not guesses
- ✓ A list of mistakes already made and fixed, so you don't repeat them

WHY THIS MATTERS (TIME & MONEY)

Realistic time: 30 to 60 minutes including sign ups. There's also a paid installer that does the same thing in about 5 minutes by answering a short interview, that's the time you're paying for, not extra features.

BEFORE YOU START

Get Set Up

Ten minutes of sign ups, then Claude Code takes over.

PREREQUISITES

Do these once, in order

1. Install Claude Code

Download the desktop app from claude.ai/code. You need a paid Claude plan.

2. Make a project folder

Anywhere on your computer, create one empty folder. A simple name with no spaces is safest, for example lead-gen.

3. Get an Apify token (free)

Go to apify.com, sign up, click your avatar, then Settings, then API & Integrations, and copy the Personal API token. It starts with `apify_api_`. This is what reads Instagram.

4. Get a Firecrawl key (free)

Go to firecrawl.dev, sign up, then API Keys in the sidebar, then Create API Key. It starts with `fc-`. This is what visits each lead's website to find an email address.

5. Optional: an Anthropic API key

Go to console.anthropic.com, then API Keys, then Create Key. It starts with `sk-ant-`. Costs roughly \$5 to \$8 a month at 10,000 leads. With it, messages are AI written and reference each lead's real bio. Without it, everything still works, just with simpler templated messages.

6. Open your folder in Claude Code

Open the empty folder from step 2 as your Claude Code project. Everything from here happens by pasting prompts.

TIP

You never write or edit a file yourself anywhere in this guide. You only answer questions in chat, Claude Code creates every file.

PART 1

Build The System

Six prompts, pasted in order.

STEP 1

Your business details and credentials

Paste this one prompt to start. It has Claude Code interview you, then generate your .env and CLAUDE.md files from your answers, no manual editing.

TRY THIS PROMPT

"I want to set up an Instagram lead-gen system. Interview me first, one question at a time, waiting for each answer before asking the next:

- 1. What business/niche am I targeting leads for? (e.g. real estate agents, dentists, wedding photographers)*
- 2. My name*
- 3. My expertise/role (e.g. "AI automation expert")*
- 4. The specific result I get clients (e.g. "a steady stream of new client leads from Instagram, automatically")*
- 5. The benefit of that result (e.g. "close more deals without spending hours manually searching and DMing prospects")*
- 6. My target country*
- 7. My Apify API token, if I say I don't have one, tell me to go to apify.com, sign up free, then Settings > API & Integrations, and copy it*
- 8. My Firecrawl API key, if I don't have one, tell me to go to firecrawl.dev, sign up free, then API Keys > Create API Key*
- 9. Optionally, an Anthropic API key for better AI written messages (explain it's optional, costs roughly \$5-8/month at 10,000 leads, and everything still works without it, console.anthropic.com > API Keys)*

Once you have all my answers, create these two files for me:

- .env with APIFY_TOKEN, FIRECRAWL_API_KEY, and ANTHROPIC_API_KEY (blank if I skipped it), plus a .gitignore containing ".env"

- CLAUDE.md with this exact structure, filled in with my answers:

Instagram Lead Gen, Project Instructions

Business

- Sender name, expertise, specific result, benefit, target audience, target country (my answers)

Scoring formula

- Niche keyword match in bio: 2+ hits = 40 points, 1 hit = 25, none = 0

- Followers 500-20,000 = 20 points; 200-499 or 20,001-50,000 = 10; anything else = 5
- Has a website: +15
- Has a findable email: +25
- Bands: 70+ strong fit, 40-69 decent, under 40 weak

Outreach copy rules

- NEVER use an em dash or en dash anywhere in a message, including inside quoted text
- NEVER paste raw scraped caption text into a message. Strip URLs, emoji and repeated punctuation first. If fewer than 3 words remain, use a generic niche opener instead of a fragment
- Greeting: personal name gets "Hey [Last Name]", business name gets "Hi [Business Name]" with LLC/Ltd/Inc/A.E. trimmed off. Keep "Dr." or "Prof." if present
- Use "an" not "a" before a niche starting with a vowel
- One sentence per line, blank line between each. Max 1-2 emoji
- Only mention follower count if they have more than 500
- This system generates drafts ONLY. It never sends anything automatically

Output

- leads_data.csv, all leads ever found
- Instagram Leads.html, the dashboard, regenerated in place each run
- ig_scrape_log.csv, run history and every username scraped, powers deduplication
- hashtag_yield_log.csv, per-hashtag new vs repeat ratio

Then run: `pip install requests python-dotenv`

STEP 2

The scraper

This is the longest prompt in the guide. It builds the script that finds Instagram accounts by hashtag, removes duplicates before spending any money, enriches each profile, and finds an email address.

TRY THIS PROMPT

"Build `scrape_leads.py`, an Instagram lead scraper. Read `CLAUDE.md` first for my business details. Requirements:

1. **HASHTAGS:** build a rotating list of hashtag batches, 5 hashtags per batch. Include about 5 generic batches for my industry, plus city-specific hashtags built from a list of 40 major US cities combined with two profession words (for real estate that's `f"{city}realtor"` and `f"{city}realestateagent"`, adapt to my industry). This matters: the Apify hashtag actor has no pagination and always returns the same top-post snapshot, so re-scraping one hashtag re-finds the same people. City hashtags reach genuinely different people.
2. Each run uses the **NEXT** batch. Store the last batch index in `scrape_state.json` so runs rotate automatically.
3. Support a `--test` flag that uses 1 hashtag and 15 results and does **NOT** advance the rotation state.

4. Scrape posts with Apify actor `apify/instagram-hashtag-scraper` via REST:
 POST `https://api.apify.com/v2/acts/apify~instagram-hashtag-scraper/run-sync-get-dataset-items?token=<APIFY_TOKEN>`
 body: `{"hashtags": [...], "resultsType": "posts", "resultsLimit": 40}`
 Collect `ownerUsername` and `caption` from each post.
 5. DEDUPLICATE BEFORE SPENDING MONEY: read every username from the `"all_usernames_scraped"` column of `ig_scrape_log.csv` across all past runs, and remove those from this run's list before any enrichment. This is the single most important cost control.
 6. Log per-hashtag yield to `hashtag_yield_log.csv`: `date`, `hashtag`, `usernames found`, `new usernames`, `new percent`. Flag any hashtag under 15 percent new as tapped out.
 7. Enrich remaining usernames with `apify/instagram-profile-scraper` (input is `{"usernames": [...]}` only, that actor has no `resultsLimit` field). Get `biography`, `followersCount`, `externalUrl`, `fullName`.
 8. Find contact info: regex the bio and caption for email and phone. If no email but they have a website, use Firecrawl to try the homepage, `/contact`, and `/about`:
 POST `https://api.firecrawl.dev/v2/scrape` (note: v2, not v1)
 header: `Authorization: Bearer <FIRECRAWL_API_KEY>`
 body: `{"url": "...", "formats": ["markdown"]}`
 Regex the returned markdown for an email.
 If `ANTHROPIC_API_KEY` is set, ALSO send that page content to Claude Haiku and ask it to extract the real person's name (reply NONE if unsure). Save as `website_name`. We already paid for that page, don't throw the rest away.
 9. Score each lead with the formula in `CLAUDE.md`.
 10. Append to `leads_data.csv` with columns: `username`, `full_name`, `website_name`, `followers`, `bio`, `website`, `email`, `phone`, `score`, `date_found`, `email_sent`, `replied`, `follow_up_sent`, `personalization`, `email_draft`, `dm_draft`
 11. Append a run row to `ig_scrape_log.csv`: `date`, `hashtags used`, `total posts`, `new unique profiles`, `leads added`, and ALL usernames scraped comma-separated.
 12. At the end, run `generate_drafts.py` then `regenerate_report.py`, and open the dashboard.
- Use direct REST calls with tokens from `.env`, never MCP, this has to run unattended on a schedule later."

STEP 3

The message writer

This builds the script that writes an email draft and a DM draft for every lead. It never sends anything, drafts only.

TRY THIS PROMPT

"Build `generate_drafts.py`. It writes an email draft and a DM draft for every lead in

leads_data.csv. It NEVER sends anything.

Follow the outreach copy rules in CLAUDE.md exactly. Specifics:

EMAIL TEMPLATE (exact structure):

[Greeting]

You are building something great as [a/an] [their niche]!

[Only if followers > 500:] Love that you've built [N] followers as [a/an] [their niche]!

Would you like to know how to get [SPECIFIC_RESULT from CLAUDE.md]?

I just made a quick video for you showing exactly how to do this.

I'm [NAME], [EXPERTISE] helping [target audience plural] [BENEFIT].

Just reply "yes" and I'll send you the video (smiley emoji)

[NAME]

DM TEMPLATE (exact structure):

[Greeting without comma]! [Opener]

Quick one, I made a short video on how [their niche]s are getting [SPECIFIC_RESULT]. Want me to send it over? (smiley emoji)

[NAME]

THE OPENER:

- If ANTHROPIC_API_KEY is set: call Claude Haiku (model claude-haiku-4-5-20251001, max_tokens 60) with the lead's bio and recent post topic, asking for ONE casual sentence under 20 words referencing something genuine. Tell it explicitly NOT to start with a greeting, since one is added separately. Then still sanitize the result and strip any leading "Hey/Hi/Hello" it adds anyway, don't trust the output blindly.

- If no key, or the call fails for any reason: fall back to "Came across your page, you're clearly building something great as [a/an] [niche]!"

ALSO BUILD:

- derive_niche(bio): regex patterns labelling each lead's sub-type, PRIMARY identity first not most-specific-string first (a family dental practice that also does pediatric work is a dental clinic, not a pediatric dentist). Plus a fallback label.

- greeting(): business names get "Hi [Name]" with LLC/Ltd/Inc/Corp and international suffixes

like A.E./S.A. trimmed. Personal names get "Hey [Last Name]". Keep Dr./Prof. titles. Single names get "Hey [Name]".

- `article(word)`: returns "an" before a vowel, "a" otherwise.
- `clean_quoted_text()`: strips URLs, emoji, repeated punctuation, and em/en dashes from any scraped text before it goes in a message.

CRITICAL: if a lead's `email_sent` column is filled in, never regenerate its `email_draft`, that's the record of what was actually sent. DM drafts can always be regenerated since nothing is ever auto-sent."

STEP 4

The dashboard

This builds the single HTML file you'll actually look at every day, plus a desktop shortcut so you never go hunting for it in a folder.

TRY THIS PROMPT

"Build `regenerate_report.py`, producing a single self-contained `Instagram Leads.html` in this folder, overwritten in place each run (never a new file, never needing re-import anywhere).

Design:

- Purple gradient hero header with the title and a "last updated" timestamp
- Four stat cards with icons: Total leads, With email, Emails sent, Replied
- A "How the score works" panel showing the exact point breakdown and the 70+/40-69/under 40 bands, so the score is never a mystery number
- A search box (filters by name/niche as you type) and a status dropdown filter
- A table: coloured avatar circle with their initial, @username linking to their Instagram, their full name, a niche chip, follower count, clickable email, score pill (green 70+, amber 40-69, grey under 40), status badge, and a chevron
- Clicking any row expands it to show the full email draft and DM draft side by side, each fully readable, each with a Copy button
- Keep the column count low enough that nothing gets cut off at 1280px width. Use `overflow-x: auto` NOT `overflow: hidden` on the table wrapper, the shorthand silently overrides the scroll behaviour
- When embedding message text into onclick handlers, escape quotes for the HTML attribute context, not just backticks, a literal double quote in the text will otherwise break the whole page's JavaScript

Then create a Windows shortcut to this HTML file on my Desktop. Get the real Desktop path with `[Environment]::GetFolderPath("Desktop")`, don't assume `%USERPROFILE%\Desktop`, it's often redirected to OneDrive."

STEP 5

Test it for real

TRY THIS PROMPT

"Run: `python scrape_leads.py --test`
Show me the real output and open the dashboard."

⚠️ WATCH OUT

This costs about \$0.05. Look at the actual messages before going further. If a niche label is wrong or a message reads oddly, tell Claude Code what looks wrong and have it fix the patterns. Don't scale up until the small run looks right.

STEP 6

Run it daily by itself

📝 TRY THIS PROMPT

"Register this in Windows Task Scheduler to run daily at 8am:

- Use the FULL absolute path to python.exe (Task Scheduler doesn't inherit my PATH, "python" alone will silently fail)*
- -StartWhenAvailable so a missed run still fires when I turn my PC on*
- -MultipleInstances IgnoreNew*
- Then read back the task state to confirm it actually registered"*

Claude Code does not need to be open for this to run. Only your PC needs to be on.

PART 2

What It Costs

Real numbers from a real run, not estimates.

COST BREAKDOWN

Per run of 5 hashtags x 40 results, about 200 posts scraped

Schedule	Profiles / month	Cost / month
One run a day	~4,900	~\$34
Two runs a day	~9,700	~\$67
Optional Anthropic key	n/a	+ \$5 to \$8

WHY THIS MATTERS (TIME & MONEY)

A single run of about 160 unique profiles costs roughly \$1.10 in Apify and Firecrawl combined.

WATCH OUT

About volume claims: roughly 25 percent of scraped profiles have a findable email, so 10,000 scraped profiles is about 2,500 contactable leads. Anyone quoting huge lead numbers is usually counting scraped profiles, not contactable people. Plan with the real number.

About sustaining volume: the 105 hashtags in this build take 21 days to cycle through at one run a day. After that, dedup means fewer new people per run. Watch `hashtag_yield_log.csv`, when hashtags drop below 15 percent new, add more cities to the geographic list. Nothing scrapes infinitely.

About country targeting: hashtags can't guarantee a country. These actors return no reliable location field, so you'll get some international leads regardless of your target setting. City hashtags help a lot.

PART 3

Mistakes Already Fixed

Real bugs from actually building this. Each one cost real debugging time.

READ THIS BEFORE YOU CUSTOMIZE ANYTHING

So you don't repeat the same eight mistakes

- **Verify an Apify actor's input schema before writing the call:** field names differ per actor. instagram-profile-scraper has no resultsLimit, and using search plus searchType on the general scraper returns hashtag analytics, not posts.
- **MCP connectors don't work in scheduled runs:** they only exist in a live Claude session. Scheduled scripts must use direct REST calls with your own API keys.
- **Firecrawl is on v2:** v1 is deprecated.
- **Keep every script's CSV column list in sync:** add a column in one script and not another, and the next run corrupts the file.
- **Don't build nine stage files:** one script that works beats an elegant architecture that doesn't.
- **Deduplicate before the paid calls, not after:** otherwise you pay to re-enrich people you already have.
- **Never quote raw scraped captions into messages:** they contain URLs, emoji pileups, and truncate mid-word. Sanitize, or use a generic opener.
- **Test on a small batch and actually read the output:** before scaling up.

WANT IT TO SEND THE EMAILS TOO?

The next level, already built

Everything above gets you a complete system: it finds leads, scores them, and writes a ready email and DM for each one. You review the dashboard and send what you want yourself. That's genuinely enough to run on its own.

If you want it to go further, sending emails automatically (capped and spread through the day so it doesn't look like a spam blast), checking your inbox for replies, and sending exactly one follow-up to anyone who doesn't respond, that's already built. You could keep extending this guide's approach to add it yourself, but it's a fair bit more work to get the safety details right (throttling, reply-matching, never double-following-up).



WHY THIS MATTERS (TIME & MONEY)

That version is ready to install, no extra building required.

Want this running in 5 minutes, with sending and follow-ups included?

[Get it here -> fabimarkl.com/library](https://fabimarkl.com/library)