

FREE RESOURCE — COPY, PASTE, BUILD

Build your own *content dashboard*

Six prompts that recreate the exact system we use to run our Instagram: competitor intelligence, a carousel generator, a content planner, a hook bank, and a post scheduler. Paste them into Claude Code (or any AI coding agent) one at a time and you will have a working dashboard for your own business account.

00	The foundation	app shell, database, design system
01	Competitor Intelligence	weekly brief, hooks, top posts
02	Carousel Generator	hook in, finished slides out
03	Content Planner	a month of ideas on a calendar
04	Hook Bank	save and reuse what works
05	Post Scheduler	compose, queue, publish

How to use these prompts

Each prompt is self-contained and written for an AI coding agent — Claude Code, Cursor, or similar. You do not need to know how to code. You need to know your brand.

The order matters

Run prompt 00 first — it builds the shell every other view plugs into. After that, add views in any order. One prompt per session works best: let the agent finish, click through the result in your browser, then move to the next.

Personalize before you paste

- Replace anything in [BRACKETS] with your own details — your niche, your competitors, your brand colors.
- Write a short BRAND.md file first: who you are, your content pillars, your voice, what you never say. Every AI feature gets dramatically better when the agent can read it.
- Keep your API keys in a .env file. Never paste them into a prompt.

What you will need

- An Anthropic API key — for the analysis and writing features (Claude does the strategy work).
- A PostgreSQL database — free tiers on Neon or Supabase are enough.
- Optional: an image-generation API for the carousel visuals, and a posting API (or Meta Graph API access) for the scheduler.

The honest part: the dashboard will not make your content good. It removes the friction — the research, the formatting, the scheduling — so your judgment has room to work. That part stays yours.

The *foundation*

A single-page app: fixed sidebar, five views, one warm editorial design system. Everything else builds on this.

COPY FROM HERE

Build a single-page content dashboard web app for my business Instagram account.

STACK

- Python Flask backend, PostgreSQL for storage (connection string from DATABASE_URL in a .env file), vanilla JavaScript frontend – no framework, one HTML template, one app.js, one style.css.
- Organize the backend as Flask blueprints (one file per feature) and a db/package with one module per table. Create tables automatically on startup with CREATE TABLE IF NOT EXISTS.

LAYOUT

- Fixed left sidebar: my logo, a list of competitors I track, and a "run analysis" button.
- Main area with a horizontal nav of 5 views: Intelligence, Carousel Generator, Planner, Hook Bank, Scheduler. Only one view is visible at a time. Use a single resetNav() helper that hides ALL views and deactivates ALL nav buttons before showing the selected one – never toggle views individually (that causes two views showing at once).
- Each view loads its data from JSON endpoints under /api/ and renders client-side.

DESIGN SYSTEM

- Warm off-white background [#fbfbf9], near-black ink text [#111111], one red accent [#d24034] used sparingly. Never pure white or pure black.
- Card-based layout: white cards, 1.5px solid borders in a warm gray, 10–14px rounded corners, generous padding.
- Typography: bold grotesque display font for headings, a clean sans for body, a monospace font for numbers and stats. Editorial, magazine-like, no gradients, no shadows heavier than a whisper.
- My business: [DESCRIBE YOUR BUSINESS IN ONE SENTENCE].

Start the dev server on port 5555 and confirm the empty shell renders with all five views switchable.

Why vanilla JS: no build step means the agent iterates faster and you can read every file. For a tool only you use, a framework is overhead.

Competitor *Intelligence*

The view that earns the dashboard its keep. It watches your competitors, extracts what works, and turns it into a weekly brief you can act on.

COPY FROM HERE

Add a Competitor Intelligence view to my dashboard.

DATA

- I track these competitors: [@HANDLE1, @HANDLE2, @HANDLE3, @HANDLE4].
- Ingest their recent Instagram posts (caption, play count, like count, post URL, thumbnail) via [Apify / an unofficial API / manual JSON import - pick what I have access to]. Store each ingestion as a "run" so history is preserved; the dashboard always shows the latest run.

ANALYSIS (Claude API, model claude-opus-4-7)

For each competitor, generate and store:

- a 3-4 sentence strategy summary
- content patterns (bullet list of what they repeatedly do)
- every hook from their captions, classified by type: Question, Stat, Story, Bold Claim, How-To, Controversy

Then generate one WEEKLY BRIEF across all competitors:

- a TL;DR paragraph, 5 key findings, trending hooks, opportunity gaps (what nobody in my niche is doing that fits MY positioning), and trending tools they mention
- 3-4 carousel ideas for next week, each mapped to one of my content pillars: [LIST YOUR 3-4 CONTENT PILLARS]

Claude must read my BRAND.md and write everything in my voice.

VIEW

- Overview: stat tiles (posts analyzed, hooks collected, last run), the weekly brief in a large card, opportunity gaps, carousel idea cards tagged by pillar, trending tools, two charts (posting activity per competitor, hook-type distribution) using Chart.js.
- A "Recent hooks" table: thumbnail, hook text, competitor, type badge - paginated, 10 per page with prev/next.
- Clicking a competitor in the sidebar opens their report: summary, chips for posting frequency and top format, content patterns, top posts (show ONLY the hook per post - clicking a post opens a modal with the full caption), and their best hooks.
- Proxy competitor thumbnails through my backend so they render despite CORS, and cache them.

Carousel *Generator*

You write one hook. The system plans the narrative, generates the imagery, and renders finished slides in your brand style — then lets you fix anything by hand.

COPY FROM HERE

Add a Carousel Generator view to my dashboard.

INPUT

- A textarea for my hook — it goes on slide 1 word for word, never rewritten.
- A Style select (my visual styles: [e.g. "Product", "Fashion"]) and a slide-template multi-select dropdown: I pick which section templates to include and in what order; Cover and CTA slides are always locked in. If I pick nothing, Claude decides (5-8 slides).

PIPELINE (run server-side as a background job with progress polling)

1. Claude (claude-opus-4-7) reads my BRAND.md and plans the carousel: one brand_concept (a single product/story carried across all slides), per-slide headline + sub-line + an emphasis array (words to render in an italic serif accent color), and an image prompt per slide.
2. Image prompts must request TEXT-FREE photographs only — no screens, signs, paper, or packaging with words. All text is overlaid later. Every prompt names 3-5 palette hexes from my brand style.
3. Generate the images with [YOUR IMAGE API — e.g. Higgsfield, Flux, gpt-image-1].
4. A Python compositor (Pillow) renders 1080x1350 slides: lays out the images per template, then draws the headline/sub-line on top using my brand fonts, with emphasis words in [ACCENT HEX] italic serif.
5. The last slide is always a CTA: [YOUR CTA, e.g. "Comment PROMPTS"].

REVIEW

- Show the plan first (slides, concept, image prompts) for approval before spending image credits.
- Render finished slides in a horizontal strip. Each slide opens in a browser editor where I can click any text to edit it inline, drag decorations, recolor elements, and re-export the PNG at full size.
- A download button zips all slides, ready to upload.

The rule that saves the whole feature: never let the image model write words. It will mangle them.

Generate clean photographs and draw the text in code — that is what makes slides look designed instead of generated.

Content *Planner*

A month at a glance. Every idea has a date and a pillar, so "what do I post today" stops being a question.

COPY FROM HERE

Add a Content Planner view to my dashboard.

- A monthly calendar grid (Monday-first), with prev/next month navigation and a "today" highlight.
- Each day cell holds zero or more idea chips. An idea has: date, text, a type from my content pillars [LIST YOUR PILLARS, e.g. pain-led, educational, project-showcase, tool-tip], and optional notes.
- Color-code the chips by type – muted editorial tints, not neon.
- Click an empty day to add an idea (inline form: text, type select, notes). Click an existing chip to edit or delete it. Ideas can be dragged to another day.
- Store ideas in a `planner_ideas` table with full CRUD endpoints under `/api/planner/`.
- Add a "send to scheduler" action on each idea that pre-fills the Scheduler compose form with the idea text.
- Below the calendar, show a small legend of my pillars and a count of planned vs empty days this month.

Why pillars and not just notes

Typing ideas into a calendar is easy. The pillar tag is what shows you, at a glance, that you have planned twelve product posts and zero educational ones — the imbalance you cannot see in a notes app.

Hook *Bank*

Every scroll-stopping first line you meet, saved with enough context to reuse it. Your future captions start here instead of from a blank page.

COPY FROM HERE

Add a Hook Bank view to my dashboard.

- A hook entry has: the hook text, a reusable template version of it (e.g. "I tested [X] so you don't have to"), a type (Question, Stat, Story, Bold Claim, How-To, Controversy), a niche tag, view count of the post I found it in, the creator handle, and the post URL.
- "Save a hook" form at the top – collapsed by default behind a toggle header with a chevron, so the list stays the hero of the page.
- The list: one card per hook showing the text, type badge (color-coded per type), niche, formatted view count, creator, and a link to the source post. Newest first.
- Filters: a search box (matches text, template, and creator) plus facet chips for type and niche, built from whatever values exist in the data. Filters combine.
- Full CRUD under /api/hook-bank/ with a hook_bank table.
- Expose a "pick from bank" modal that other views can open (the Scheduler compose form will use it): searchable list, click to select, returns the hook text to the caller.

The template field is the point

A saved hook is trivia. A saved template is an asset. "This founder rebuilt her funnel in 4 days" becomes "This [WHO] [did the hard thing] in [timeframe]" — and now it works for your niche, not theirs.

Post *Scheduler*

Compose once, queue it, and let the dashboard remember what went out and when. The queue is your publishing memory.

COPY FROM HERE

Add a Post Scheduler view to my dashboard.

COMPOSE

- A "Compose post" card – collapsed by default behind a toggle header.
- Fields: caption textarea with live character count, a "pick from Hook Bank" button that opens the bank's picker modal and inserts the chosen hook as the first line, optional fields for angle and CTA, media upload (image or video, previewed inline), and a datetime picker.
- Two actions: "Schedule" and "Publish now".

PUBLISHING

- Publish through [YOUR POSTING API – e.g. Meta Graph API, or a service like Zernio/Buffer]. Store every post locally in a `scheduled_posts` table regardless: caption, media path, `scheduled_for`, account, status, external post id.
- Status flow: `scheduled` → `posted` (flip it lazily whenever the queue is fetched and `scheduled_for` is in the past) / `published` (via `publish now`) / `failed`.

QUEUE

- "Upcoming queue" list: one card per post – thumbnail, first line of the caption, scheduled date, and a status badge (green `scheduled`, indigo `posted`, red `failed`).
- Clicking a card opens a detail modal with everything: full caption (line breaks preserved), status, scheduled and posted times, account, hook, angle, CTA, and the media preview. Delete button with a confirm step, `stopPropagation` so it doesn't open the modal.
- Sort: soonest first, posted items sink below upcoming ones.

Make it *yours*

Iterate with screenshots

The fastest way to fix UI you do not like: screenshot it, give it to the agent, and say what feels wrong ("the labels don't align with the buttons", "too much text on these cards"). Concrete beats abstract every time.

Feed it your voice, not ours

Every prompt above leans on a BRAND.md file. Spend an hour on it. Who you serve, your 3-4 content pillars, your tone rules, the words you never use. It is the difference between AI output you post and AI output you rewrite.

Seed it, then trust it

Ask the agent to write a small seed script with realistic demo data first — you will find ten UX problems in a filled dashboard that an empty one hides. Tag the demo rows so one command deletes them later.

What we learned building ours

- One view per prompt session. Agents finish small scopes well and large scopes halfway.
- Keep the AI calls server-side and the key in .env — never in frontend code.
- Store every analysis run instead of overwriting. Your past briefs become training data for better future ones.
- The dashboard organizes the work. The judgment — what to post, what to skip, what sounds like you — never left your side of the table.

IF YOU WANT TO SKIP THE BUILD

Three ways to *get there*

This guide is the free tier. If you would rather start from a working system than a blank folder, there are two more.

Free

THIS GUIDE

The prompts

The six prompts in this PDF, the build order, and the lessons from building ours. You bring an AI coding agent and an afternoon.

You are holding it.

\$20

CODE + SKILL

My complete codebase, plus a Claude skill that sets it up for you

You get the full source of my dashboard — every view, the database schema, the AI pipelines — together with a ready-made Claude Code skill. Open the project, run the skill, and it walks Claude through setting everything up on your machine and rebranding it to your business. Zero to a running dashboard in one session.

DM "CODE" to @frame.gen.studio

\$300

DONE FOR YOU

I set the dashboard up for you

I configure the whole system around your account: your brand tokens and fonts, your content pillars, your competitors, and prompts tuned to your voice — so the generations come out consistent from day one and the intelligence data is tailored to your niche, not a template. You start with a dashboard that already knows your business.

DM "SETUP" to @frame.gen.studio

Built by *@frame.gen.studio*

We make AI video and content systems for ecommerce brands — with a human in the loop, on purpose. If you build your dashboard with these prompts, tag us. We read everything.