

— THE COMPLETE PACK

AI PROMPT PLAYBOOK

The full working set — plus **one real automation** you build, wire up, and watch run.

52 PROMPTS · 1 REAL AUTOMATION

Start here

You don't need ten thousand prompts. You need a set that actually works — and the skill to bend any of them to your own work. That's this book.

Inside are **52 copy-paste prompts** built for one job: making short-form content without staring at a blank box. They're grouped into six stages — hooks, scripts, repurposing, research, editing, and automation — so you can jump straight to whatever you're stuck on right now.

How to use them: find the prompt that fits the moment, copy the whole block, paste it into ChatGPT (or Claude, or any AI chat), and swap the [BRACKETS] for your details. Every prompt is written to be pasted as-is — the structure is doing the heavy lifting, so you don't have to be a 'prompt engineer.'

Straight talk: AI hands you a strong *first draft*, not a finished post. You're still the editor — read it before it ships. Do that, and these quietly buy back the slowest hours of the week. No 'reclaim 15 hours' fantasy; just a real head start on work you're already doing.

THE ONE RULE THAT MATTERS

Every prompt here is built to refuse the things that get creators in trouble — invented stats, fake testimonials, clickbait the video can't pay off. If a prompt can't do something honestly, it's written to tell you so instead of making it up. Keep that standard and the audience trusts you. Break it and no prompt can save you.

The engine under every prompt — R·C·T·F

Once you see the shape, you stop needing a library — you can build a prompt for anything in about thirty seconds. Every prompt in this book runs on the same four parts:

- **Role** — who the AI should be (*“you are a sharp short-form editor”*). Sets the lens it answers through.
- **Context** — what's going on (*your topic, audience, platform — the bracketed details*). The more specific, the better the output.
- **Task** — exactly what you want it to do, in plain terms. One clear job beats five vague ones.
- **Format** — how you want it back (*a numbered list, a word cap, a tone, a table*). This is what turns a wall of text into something usable.

Anatomy of a Prompt

ROLE

You are a short-form hook writer.

WHO THE AI
SHOULD BE

CONTEXT

My video topic is **[TOPIC]**. My audience is **[WHO THEY ARE]**.

YOUR
SPECIFICS

TASK

Write 10 scroll-stopping opening lines, each under 12 words.

THE EXACT
JOB

FORMAT

A numbered list. No emojis. No hashtags.

HOW YOU
WANT
IT BACK

AMBER TOKENS = FILL-IN-THE-BLANK BRACKETS YOU SWAP FOR YOUR OWN DETAILS

BUILD ONE FROM SCRATCH

You are **[a ROLE that fits the job]**.
Here's the situation: **[your CONTEXT]**.
I want you to **[the TASK]**.
Give it to me as **[the FORMAT – length, list, tone]**.

Then *talk to it* — “shorter,” “warmer,” “try again,” “use a real example.” Most people quit after one reply. Go two or three rounds and you'll leave them behind.

That's the whole skill. The 52 prompts ahead are this shape, already tuned for the most common jobs — study a few and you'll feel how they're put together.

What's inside — all 52, plus the automation

THE AI PROMPT PLAYBOOK

The 52, by stage

52

PROMPTS MAPPED ACROSS THE 6 WORKFLOW STAGES

PROMPTS TOTAL



6 STAGES · ONE WORKFLOW

12 + 10 + 9 + 8 + 7 + 6 = 52

Hooks & Openers

12 PROMPTS

- 1 The Scroll-Stopper Menu
- 2 The Open Loop
- 3 The Bold Claim
- 4 The Cold Open
- 5 The Callout
- 6 The List Tease
- 7 The Direct Question
- 8 The Hard Stop
- 9 The Before / After
- 10 The Hook-to-Hold
- 11 The Angle Stack
- 12 The Niche Adapter

Scripts & Outlines

10 PROMPTS

- 13 The 30-Second Talking-Head
- 14 The Faceless Voiceover Build
- 15 The Story-Arc Outline
- 16 The How-To Step Stack
- 17 The Listicle Rapid-Fire
- 18 The Scroll-Stopper Hook Lab
- 19 The Closer (CTA Ender Pack)
- 20 The Myth-Buster
- 21 The Problem-Agitate-Solve Mini
- 22 The Repurpose Chopper

Repurposing & Captions

9 PROMPTS

- 23 The One-to-Three Splitter
- 24 Transcript-to-Carousel
- 25 Triple-Platform Caption Pack
- 26 The Hashtag Stack Builder
- 27 Script-to-Thread Converter
- 28 The Best-Clips Extractor
- 29 The Repurpose Map
- 30 The Hook-Variant Lab
- 31 The Platform Translator

Research & Ideas

8 PROMPTS

- 32 The 30-Idea Topic Splitter
- 33 The Sub-Niche Excavator
- 34 The Trend-Angle Remixer
- 35 The Secret Struggle Finder
- 36 The Competitor Gap Differentiator
- 37 The Content Pillar Architect
- 38 The Series Hook Planner
- 39 The Comment-Mining Idea Engine

Editing & Cleanup

7 PROMPTS

- 40 The Flat-to-Fire Punch-Up
- 41 The Pacing Surgeon
- 42 The Filler Killer
- 43 The Voice-Match Rewrite
- 44 The Final-Pass Proofreader
- 45 The Jargon Translator
- 46 The Ending Fixer (Weak-CTA Rescue)

- 47 The Transcript Decoder
- 48 The Idea-to-Post Builder
- 49 The Comment Triage Router
- 50 The Caption Batch Engine
- 51 The Script-to-Shotlist Splitter
- 52 The Repurpose Router

MODULE 5 – THE WALKTHROUGH

One Real Automation, Wired Start-to-Finish

SECTION 01

Hooks & Openers

The first two seconds decide everything. These twelve give you scroll-stopping opening lines — and the mechanics behind why they work.

PROMPTS

01–12

IN THIS SECTION

12 total

FOCUS

Hooks

1 The Scroll-Stopper Menu

For: Your default first move on any video. Use when you have a topic but no hook yet and want a spread of pattern-interrupt openers to pick the strongest from.

Role: You are a short-form hook writer who studies why people stop scrolling.

Context: My video topic is [TOPIC]. My audience is [WHO THEY ARE] who feel [PAIN OR DESIRE]. The platform is [TIKTOK / REELS / SHORTS]. My on-screen tone is [CONFIDENT / CALM / PLAYFUL].

Task: Write 10 spoken hook lines that work as the FIRST sentence of the video – the line said in the first 2 seconds. Each must create a reason to keep watching: a pattern interrupt, a stakes-raise, or an open loop. Vary the mechanism across the 10 so I get real options, not 10 versions of one idea.

Format: A numbered list, 1–10. Each line max 12 words, written exactly as it would be spoken (no stage directions). After each line, add a 3-word tag in parentheses naming the mechanism, e.g. (open loop), (bold claim), (pattern interrupt). No emojis. No hashtags. Do not explain the list before or after – output only the 10 lines.

What you get: A spread of 10 first-line options built on different attention mechanisms, each tagged so you can see WHY it works and pick the strongest for your specific video instead of settling for the first idea.

2 The Open Loop

For: When you want hooks that plant an unanswered question the viewer has to stay for. Best for tutorials, reveals, and 'here's what happened' videos.

Role: You are a retention specialist who writes curiosity-gap openers for short-form video.

Context: The payoff of my video is [WHAT THE VIEWER LEARNS OR SEES BY THE END]. My audience is [WHO THEY ARE]. I do NOT want clickbait that the video can't deliver on – the loop must be honestly closeable by my actual content.

Task: Write 8 open-loop hooks. Each one opens a specific curiosity gap – names a result, a change, or a surprising fact – WITHOUT revealing the answer, so the viewer keeps watching to close the loop. The gap must be one my stated payoff actually pays off.

Format: Numbered 1–8. Each hook max 14 words, spoken style. Under each, add one short line: 'Closes when: [the exact moment in the video that pays it off]'. Avoid the words 'secret', 'nobody talks about', and 'this one trick' – find fresher phrasing. No emojis, no hashtags.

What you get: Curiosity-gap openers paired with the exact payoff moment that closes each loop – so the hook is tension you can actually resolve on screen, not bait that tanks your watch-through.

3 The Bold Claim

For: When you want a confident, contrarian, or counterintuitive opener that stakes a strong position in the first second. Great for opinion and 'hot take' content.

Role: You are a copywriter who specializes in bold, defensible opening claims for short-form video.

Context: My core message is [THE POINT I'M MAKING]. The common belief I'm pushing against is [WHAT MOST PEOPLE ASSUME]. My audience is [WHO THEY ARE]. I want confident, not hypey – claims I can actually back up in the video.

Task: Write 8 bold-claim hooks that take a clear, slightly contrarian stance in the first line and make the viewer want to argue or agree. Each must be a position I can defend with the rest of the video – no exaggeration I'd have to walk back.

Format: Numbered 1–8. Each max 13 words, spoken style, present tense, stated as fact (not as a question). After each, add: 'Backs up with: [the proof or example I'd show]'. No hedging words (maybe, probably, kind of). No emojis, no hashtags.

What you get: Confident contrarian openers each tied to the proof you'd show — so you get the scroll-stopping punch of a hot take without making a claim your video can't stand behind.

4 The Cold Open

For: When you want to drop the viewer mid-story or mid-moment instead of introducing yourself. Best for 'so this happened' and narrative-style videos.

Role: You are a story editor who writes in-the-moment cold opens for short-form video.

Context: The story or situation is [WHAT HAPPENED – one or two sentences]. The emotional core is [TENSION / SURPRISE / FRUSTRATION / WIN]. My audience is [WHO THEY ARE].

Task: Write 6 cold-open hooks that drop the viewer straight into the action – no greeting, no setup, no 'in this video'. Start mid-moment so it feels like they walked in on something already happening. Each should imply there's more story to come.

Format: Numbered 1–6. Each max 16 words, spoken style, first person, present or past tense as fits the moment. Do not start any line with 'Today', 'In this video', 'Hey guys', or 'So I'. After each, one tag: (the unanswered question this raises). No emojis, no hashtags.

What you get: Mid-action story openers that skip the throat-clearing intro and drop viewers into a moment in progress — each carrying an implied 'what happens next' that pulls them through.

5 The Callout

For: When you want a 'you're doing X wrong' or mistake-naming opener that makes the right viewer feel personally addressed. Use for fix-it and how-to content.

Role: You are a hook writer who specializes in respectful 'you're doing this wrong' callouts.

Context: The mistake my audience is making is [THE COMMON MISTAKE]. The better way is [WHAT THEY SHOULD DO INSTEAD]. My audience is [WHO THEY ARE]. I want it to land as a helpful wake-up, NOT as shaming or insulting.

Task: Write 8 callout hooks that name a specific mistake the viewer is probably making, so the right person feels seen and keeps watching for the fix. Be specific about the mistake – generic 'you're doing it wrong' is too vague to stop a scroll.

Format: Numbered 1–8. Each max 14 words, spoken style, second person ('you' / 'your'). No insults, no name-calling, no 'you're an idiot' energy. After each, add: 'Fix teased: [the better way, in 4–6 words]'. No emojis, no hashtags.

What you get: Specific mistake-callout openers that make the exact viewer feel personally addressed and tease the fix — calibrated to wake people up, not shame them off the video.

6 The List Tease

For: When your video is a numbered list or 'X ways/tools/mistakes' format and you want a hook that promises the payload up front. Use for listicle content.

Role: You are a hook writer for numbered / listicle short-form videos.

Context: My video is a list of [NUMBER] [THINGS – e.g. tools, mistakes, tips, examples] about [TOPIC]. My audience is [WHO THEY ARE]. The single most valuable item on the list is [BEST ITEM].

Task: Write 7 listicle hooks that promise the list and create urgency to watch all of it. Use at least 2 different tactics across the set: (a) tease the best item is saved for last, (b) front-load a number + benefit, (c) warn one item is a mistake to avoid. Make the count and the payoff concrete.

Format: Numbered 1–7. Each max 14 words, spoken style. Each must include the actual number from my list. After each, one tag naming the tactic, e.g. (saved-for-last) or (number + benefit). No emojis, no hashtags.

What you get: Listicle-specific openers that bake in your real item count and a reason to watch to the end — using varied tactics so they don't all read as the same 'X ways to...' template.

7 The Direct Question

For: When you want a question opener that makes the viewer answer in their head and stay to see if they're right. Good for relatable, audience-first videos.

Role: You are a hook writer who uses questions to pull a viewer in by the first second.

Context: My topic is [TOPIC]. The specific viewer I'm talking to is [WHO THEY ARE] who wants [DESIRE] or struggles with [STRUGGLE]. The answer my video provides is [PAYOFF].

Task: Write 8 question hooks that make the right viewer instantly answer in their head – and then need to watch to confirm they're right. Mix self-identifying questions ('Are you the kind of person who...') with knowledge-gap questions ('Do you know why...'). Avoid lazy yes/no questions that are too easy to scroll past.

Format: Numbered 1–8. Each is a single question, max 13 words, spoken style, ending in a question mark. No double questions in one line. After each, tag it (self-identifying) or (knowledge gap). No emojis, no hashtags.

What you get: Question openers split into self-identifying and knowledge-gap types — each engineered to trigger a silent mental answer that the viewer then stays to verify, instead of a skippable yes/no.

8 The Hard Stop

For: When you want a negative-framed 'stop doing this' or warning opener that uses loss-aversion to freeze the scroll. Use sparingly, for high-stakes advice.

Role: You are a hook writer who uses warnings and loss-aversion to stop the scroll honestly.

Context: The thing I'm warning against is [THE RISKY ACTION OR HABIT]. The real downside of it is [WHAT IT ACTUALLY COSTS THEM]. My audience is [WHO THEY ARE]. I want urgency that's TRUE — no fear-mongering about things that aren't really risks.

Task: Write 7 'hard stop' hooks that open with a warning or a 'stop doing this' command tied to a real consequence. Each must make the viewer pause because the downside is genuine and specific to them.

Format: Numbered 1–7. Each max 13 words, spoken style, opening with a stop/warning beat. After each, add: 'Real risk: [the genuine downside in 4–7 words]'. Do not invent statistics or fake dangers. No emojis, no hashtags.

What you get: Warning-style openers built on a genuine, named downside rather than manufactured fear — loss-aversion that freezes the scroll while keeping your warning honest and defensible.

9 The Before / After

For: When your video shows a transformation, result, or 'I went from X to Y' and you want a hook that frames the gap up front. Use for results and progress content.

Role: You are a hook writer who frames transformations for short-form video.

Context: The before state is [STARTING POINT]. The after state is [END RESULT]. The change happened over [TIMEFRAME] or via [METHOD]. My audience is [WHO THEY ARE] who want the after state. Use only true details – do not inflate the result.

Task: Write 7 before/after hooks that open with the gap between where things started and where they ended, so the viewer wants to know how to cross it. Vary which side leads: some open on the impressive after, some on the relatable before.

Format: Numbered 1–7. Each max 15 words, spoken style. Each must contain a concrete before AND after element (or clearly imply both). After each, tag it (after-first) or (before-first). Do not fabricate numbers I didn't provide. No emojis, no hashtags.

What you get: Transformation openers that put the before-to-after gap in the first line so curiosity about the 'how' carries the video — built only from the real details you supply, no inflated results.

10 The Hook-to-Hold

For: When your hook keeps dying after the first line because the next sentences sag. Use to generate the hook PLUS the retention bridge that holds viewers into the body.

Role: You are a retention editor who fixes the 'great hook, dead second line' problem in short-form video.

Context: My video topic is [TOPIC] and its payoff is [WHAT THEY GET BY THE END]. My audience is [WHO THEY ARE]. My intended hook is [YOUR HOOK – or write 'pick the best one' and propose it].

Task: Write 4 complete openers. Each is the HOOK (line 1) PLUS the next 2–3 lines that bridge into the body without losing momentum – restating stakes, narrowing to the viewer, or raising a second mini-loop. The goal is no dead spot between the hook and the value.

Format: Numbered 1–4. For each, label the parts: 'Hook:' (max 13 words) then 'Bridge:' (2–3 short spoken lines). Keep total opener under 40 words. No emojis, no hashtags. Do not include the body of the video – stop at the bridge.

What you get: Full openers that pair the first line with the 2–3 sentences after it — closing the common gap where a strong hook is killed by a sagging second line and viewers bounce before the payoff.

11 The Angle Stack

For: When you have ONE idea and want to A/B test it. Generates the same hook across multiple emotional angles so you can post variants and learn what your audience responds to.

Role: You are a hook strategist who reframes a single idea across multiple emotional angles for testing.

Context: My one core idea / hook is [YOUR IDEA OR HOOK]. My audience is [WHO THEY ARE]. My topic is [TOPIC]. I want to test which angle lands best with them.

Task: Take my ONE idea and rewrite it as 6 hooks, each using a DIFFERENT emotional angle on the same idea: (1) curiosity, (2) bold claim, (3) relatable pain, (4) fast benefit, (5) story/in-the-moment, (6) question. Keep the underlying idea identical – only the angle changes – so I'm running a clean A/B test, not 6 different videos.

Format: A list labeled by angle (Curiosity:, Bold claim:, Relatable pain:, Fast benefit:, Story:, Question:). One hook each, max 14 words, spoken style. No emojis, no hashtags. Do not change the core idea between versions.

What you get: Your single idea expressed through 6 fixed emotional angles — a clean A/B test set where only the framing changes, so what you learn about angle performance actually transfers to your next videos.

12 The Niche Adapter

For: Your reusable batch engine. Plug in your niche and a content pillar, get a batch of on-brand hooks tailored to your specific audience's language. Use to fill a content calendar fast.

Role: You are a hook writer who tailors openers to a specific niche's language and pain points.

Context: My niche is [YOUR NICHE]. My content pillar for this batch is [PILLAR / THEME]. My audience is [WHO THEY ARE], and the words they actually use for their problem are [2–3 PHRASES THEY SAY]. My brand tone is [CONFIDENT / CALM / PLAYFUL]. (Example niche, replace with yours: a faceless channel about home-coffee gear for 'Northpeak Coffee'.)

Task: Write 10 hooks tailored to THIS niche and pillar – using my audience's real phrasing, not generic creator-speak. Spread them across mechanisms (curiosity, bold claim, callout, question, list) so the batch is varied enough to post over multiple days without sounding repetitive.

Format: Numbered 1–10. Each max 14 words, spoken style, in my stated brand tone. After each, a 2–3 word mechanism tag. Use my audience's phrases where natural; avoid generic filler like 'in today's world' or 'let's dive in'. No emojis, no hashtags.

What you get: A varied 10-hook batch tuned to your niche's actual vocabulary and one content pillar — reusable every time you need to fill a posting calendar, with mechanisms spread so the set doesn't sound repetitive.

SECTION 02

Scripts & Outlines

Turn an idea into a tight, ready-to-record script. Workhorses for talking-head, faceless voiceover, story, how-to, and list formats.

PROMPTS

13-22

IN THIS SECTION

10 total

FOCUS

Scripts

13 The 30-Second Talking-Head

For: Writing a tight, ready-to-record talking-head script when you'll be on camera (or your avatar will) and need it timed to land under 30 seconds. Use this as your default workhorse for a single-point video.

Role: You are a short-form video scriptwriter who writes spoken-word scripts for the ear, not the page.

Context: I run a faceless/short-form channel about [TOPIC / NICHE]. My audience is [WHO THEY ARE]. The one idea this video must land is: [SINGLE TAKEAWAY]. Speaking pace is ~150 words per minute, so a 30-second video is ~75 words. The tone is confident and energetic, plain-spoken, no jargon, no filler.

Task: Write ONE talking-head script of 70–80 words total, structured as: (1) a hook in the first line that creates a curiosity gap or stakes a claim – no "Hey guys" or "In this video"; (2) the single point delivered in 2–3 short spoken sentences with one concrete example; (3) a one-line CTA that asks for exactly one action. Write the way people talk: short sentences, contractions, no semicolons.

Format: Output the script as plain spoken lines only (no scene directions, no labels). Then below a line that says '---', give me the word count and the estimated runtime in seconds. Do NOT add hashtags, emojis, or a description.

What you get: A word-capped, pace-calibrated spoken script with a real hook and a single CTA, plus a runtime check so you know it fits before you record. The word cap is what keeps it from running long.

14 The Faceless Voiceover Build

For: Scripting a faceless video where a voiceover plays over B-roll or text cards and you need the visuals planned alongside the words. Use when there's no face on screen and every line needs something to look at.

Role: You are a faceless-content director who writes voiceover and the matching on-screen visual in lockstep.

Context: This is a faceless video (voiceover over B-roll / stock footage / text cards – no talking head) about [TOPIC]. Audience: [WHO]. Core message: [WHAT THEY SHOULD WALK AWAY WITH]. Target length: [30 / 45 / 60] seconds at ~150 words/min, so cap the voiceover accordingly (30s ≈ 75 words, 45s ≈ 110, 60s ≈ 150).

Task: Write the script as a two-column shot list. The voiceover must open with a hook line (no intro pleasantries), deliver the message in clear beats, and end on one CTA. For EACH voiceover beat, specify a concrete visual that a beginner could actually source from free stock or a simple text card – name the literal shot, not a vague mood.

Format: A markdown table with columns: | # | Voiceover (spoken) | On-screen visual |. Keep each voiceover cell to 1–2 spoken sentences. After the table, on a new line, give total word count and estimated runtime. No emojis in the voiceover. Anti-pattern to avoid: visuals like 'something inspiring' – every visual must be literally shootable or sourceable.

What you get: A shot-by-shot script that pairs each spoken line with a concrete, sourceable visual, so you're never staring at a voiceover with no idea what to put on screen. The visual-specificity rule kills the vague 'inspiring montage' problem.

15 The Story-Arc Outline

For: Outlining a personal-story or narrative video (a lesson learned, a transformation, a 'this happened to me' angle) so it has tension and a payoff instead of rambling. Use before you write the full script.

Role: You are a story editor for short-form video who structures narratives for maximum retention.

Context: I want to tell a short story on [TOPIC]. The real situation / experience: [WHAT HAPPENED – a few bullet points are fine]. The lesson or point the story proves: [THE TAKEAWAY]. Audience: [WHO]. This is an OUTLINE, not the final script – keep it skeletal so I can write the words myself.

Task: Build a story-arc outline using these beats: (1) Cold open – the most surprising or highest-stakes moment, stated in one line to stop the scroll; (2) Setup – the minimum context needed; (3) Tension – what went wrong or what was at risk; (4) Turn – the realization or pivot; (5) Resolution – what changed; (6) Lesson – the takeaway stated as advice the viewer can use; (7) CTA – one line. Do NOT invent facts I didn't give you – if a beat needs detail I didn't provide, write [FILL: what's needed] instead of making something up.

Format: A numbered list, one beat per line, each beat as a single guiding sentence (not the final dialogue). Keep the whole outline under 120 words so it stays an outline.

What you get: A 7-beat narrative skeleton that front-loads the hook and builds to a usable lesson, with explicit [FILL] placeholders instead of fabricated details. Gives your story shape without writing the words for you.

16 The How-To Step Stack

For: Structuring a tutorial or how-to video so the steps are clear, in order, and don't lose the viewer. Use whenever you're teaching the audience to do a specific thing.

Role: You are an instructional designer who turns a process into a follow-along short-form tutorial.

Context: I'm teaching viewers how to [DESIRED OUTCOME] on [TOPIC / TOOL]. Their skill level: [BEGINNER / INTERMEDIATE]. The steps I have so far (rough, possibly out of order): [LIST YOUR STEPS]. Target length: [30 / 60] seconds. Keep language plain – assume they've never done this before.

Task: Turn this into a tight tutorial script. Open with a one-line hook that names the payoff ('Here's how to [outcome] in under a minute'). Then collapse my steps into 3–5 numbered steps MAX – if I gave more, merge or cut the least essential. Each step = one spoken sentence telling them exactly what to do, in correct order. End with a one-line result statement ('That's it – now you've got [outcome]') plus one CTA. Flag any step where I left out info you'd need by inserting [CLARIFY: ...].

Format: Hook line, then a numbered list of steps (each one spoken sentence), then the result line and CTA. After a '---', list the total word count and runtime. No more than 5 steps. No jargon left unexplained.

What you get: A tutorial compressed to 3–5 ordered, single-sentence steps with a payoff-first hook – the step cap forces clarity and stops the 'too many sub-steps' sprawl that loses beginners.

17 The Listicle Rapid-Fire

For: Writing a fast '[N] things / tips / mistakes' video where each item is punchy and the pacing stays tight. Use for list-format content that needs momentum.

Role: You are a short-form writer who specializes in fast, high-retention list videos.

Context: Topic: [N] [THINGS / TIPS / MISTAKES / TOOLS] about [SUBJECT]. Audience: [WHO]. The list items I want to cover (or 'you choose' if I leave it open): [YOUR ITEMS OR 'you choose']. Target length: [30 / 45 / 60] seconds. Pacing must be rapid – one item should flow straight into the next with no dead air.

Task: Write a listicle script. Open with a hook that states the promise and the number ('[N] [things] that [benefit / consequence]') and adds a reason to keep watching ('– number [X] surprised me'). Then deliver each item as: a 2–4 word on-screen label + one spoken sentence of substance (not fluff). Save the strongest or most counterintuitive item for last. End with one CTA. Every item must carry a real, specific point – no padding items that just restate the topic.

Format: Hook line, then a numbered list where each line is 'LABEL – spoken sentence'. Then '---' and the word count + runtime. Keep each spoken sentence under 18 words. No filler items.

What you get: A rapid-fire list script with a numbered hook, scannable on-screen labels, and a saved-best-for-last structure – the 'no padding items' and word-per-item caps are what separate it from bloated '10 tips' filler.

18 The Scroll-Stopper Hook Lab

For: Generating and pressure-testing the first 3 seconds when you have a video idea but the opening line isn't landing. Use to batch hook options before committing to a script.

Role: You are a hook specialist who writes the opening lines that decide whether a short-form video gets watched.

Context: The video is about [TOPIC] and the payoff for the viewer is [WHAT THEY GET]. Audience: [WHO]. A hook here is the first spoken line AND/OR the first on-screen text – it must work in under 3 seconds and create a reason to keep watching.

Task: Write 10 distinct hook options for this video, each using a DIFFERENT mechanism, and label which mechanism each one uses. Cover at least: a bold claim, a curiosity gap, a direct callout of the audience, a mistake/warning, a contrarian take, a question, a result/number tease, a 'before you do X' interrupt, a relatable confession, and a stakes/cost line. Each hook is ONE line, max 12 words, spoken aloud naturally. No 'Hey guys', no 'In this video', no clickbait that the video can't pay off.

Format: A numbered list 1–10. Each line: 'MECHANISM – "the hook line"'. After the list, mark the 2 you think are strongest and say in one phrase why. No emojis.

What you get: Ten short, mechanism-labeled opening lines so you can A/B the strongest one instead of guessing – the variety constraint forces genuinely different angles, not ten rewordings of the same line.

19 The Closer (CTA Ender Pack)

For: Ending a video on a clear ask without sounding desperate or generic ('like and subscribe'). Use when the body is done but the ending falls flat or begs.

Role: You are a conversion-aware short-form writer who ends videos on one clean, non-needy call to action.

Context: The video was about [TOPIC]. The single action I want viewers to take is [ONE GOAL – e.g. follow for more on X / comment a keyword / save this / check the link].

Audience: [WHO]. My brand voice is confident and energetic – never begging, never 'pleeease like and subscribe'.

Task: Write 6 alternative CTA enders for this exact goal, each ONE line, each a different style: (1) value-forward ('follow for the next one on...'), (2) curiosity/loop ('part 2 covers...'), (3) comment-driven ('comment [WORD] and I'll...'), (4) save-prompt ('save this so you...'), (5) soft confident ('you know what to do'), (6) link/next-step. Each must tie back to THIS topic, not be generic. Max 14 words each. Do NOT stack multiple asks – one action per line.

Format: A numbered list 1–6, format 'STYLE – "the line"'. Then one sentence on which style fits best for a follower-growth goal vs. an engagement goal. No emojis, no 'smash that button'.

What you get: Six single-action, topic-tied closers in distinct styles, with guidance on which to use for growth vs. engagement — the one-ask rule keeps the ending clean instead of cramming three requests into the last second.

20 The Myth-Buster

For: Writing a contrarian 'you've been told X, but actually Y' video that earns attention by overturning a common belief. Use when your angle is correcting a misconception in your niche.

Role: You are a short-form writer who builds credible contrarian videos that challenge a common belief without overclaiming.

Context: In [NICHE], a lot of people believe [THE COMMON BELIEF / MYTH]. What's actually true, based on [MY REASONING / EXPERIENCE]: [THE REALITY]. **Audience:** [WHO]. **Target length:** [30 / 45] seconds. **Important:** I want this to feel credible, not edgy-for-its-own-sake – the correction must be honest and supportable, not a hot take I can't back up.

Task: Write a myth-bust script structured as: (1) Hook – state the myth as if it's gospel, then puncture it ('Everyone says [myth]. That's backwards. '); (2) Why people believe it – one fair sentence so I don't strawman them; (3) The reality – the correction, with one concrete reason or example; (4) What to do instead – practical takeaway; (5) CTA – one line. Do not exaggerate or invent evidence; if the correction needs a claim I haven't supported, insert [SUPPORT NEEDED: ...].

Format: Plain spoken script in the 5 beats above, beats separated by line breaks (no labels in the final video copy). Then '---' with word count and runtime. Keep it honest – no 'studies show' unless I gave you the study.

What you get: A contrarian script that hooks by overturning a belief but stays credible — it steelmans the myth before correcting it and flags unsupported claims with [SUPPORT NEEDED] instead of fabricating evidence.

21 The Problem-Agitate-Solve Mini

For: Writing a short, persuasive video that sells a solution (a tip, a method, a free resource, a soft product nudge) using the classic PAS structure compressed for short-form. Use when the goal is to move the viewer to want something.

Role: You are a direct-response copywriter adapting the Problem-Agitate-Solve framework for a short-form video.

Context: The viewer's problem: [PAIN POINT]. The solution I'm pointing them to: [YOUR TIP / METHOD / FREE RESOURCE / OFFER]. Audience: [WHO]. Target length: [30 / 45] seconds. Keep it persuasive but grounded – no fake urgency, no invented testimonials, no 'guaranteed' promises.

Task: Write a PAS mini-script: (1) Problem – name the pain in the viewer's own words, hook-first; (2) Agitate – one or two lines on the real cost of leaving it unsolved (time, money, frustration), without fear-mongering; (3) Solve – introduce the solution simply and say the ONE next step; (4) CTA – one line that matches the next step. Speak to 'you', use plain language, keep sentences short. Anti-patterns to avoid: hype words ('insane', 'crazy', 'secret'), fabricated stats, and promises of specific results.

Format: Plain spoken script, the 4 beats flowing as natural speech (no on-screen labels). After '---', give word count, runtime, and a one-line note on whether the CTA matches the solve. No emojis.

What you get: A compressed PAS script that names a real pain, agitates honestly, and points to one next step — with explicit bans on hype words, fake stats, and result-guarantees so it persuades without crossing into dishonest claims.

22 The Repurpose Chopper

For: Turning one long piece — a podcast clip, a long video, a blog post, or a wall of notes — into multiple standalone short-form scripts. Use when you have raw long-form material and want several posts out of it.

Role: You are a content repurposer who mines long-form material for standalone short-form video scripts.

Context: Here is my source material (transcript / notes / article): [PASTE SOURCE].

Audience: [WHO]. I want [NUMBER] separate short-form scripts out of this, each able to stand completely on its own (no 'as I said earlier'). Each target length: [30 / 60] seconds at ~150 words/min, so cap each script's words accordingly.

Task: First, scan the source and pull the [NUMBER] strongest standalone ideas — each must contain one clear, self-sufficient point. Skip anything that only makes sense with surrounding context. Then for EACH idea, write a complete short script: a scroll-stopping hook line, the point delivered in 2–4 spoken sentences, and one CTA. Use ONLY ideas and facts present in my source — do not add claims that aren't in it. If the source is too thin for [NUMBER] strong scripts, give me fewer and say so honestly rather than padding.

Format: For each script, output: 'SCRIPT [n] — [3–5 word title]', then the spoken script lines, then word count + runtime. Separate scripts with a '===' line. If you produced fewer than requested, add a final note explaining why.

What you get: Several self-contained short scripts extracted from one long source, each with its own hook and CTA — the 'source-only' rule keeps it from inventing facts, and the honesty clause means it gives you fewer good scripts rather than padding weak ones.

SECTION 03

Repurposing & Captions

Get five posts out of one. Split long content, re-native it per platform, and write captions that read native — not copy-pasted.

PROMPTS

23–31

IN THIS SECTION

9 total

FOCUS

Repurposing

23 The One-to-Three Splitter

For: Turning a long video, podcast, or livestream into 3 standalone shorts. Use right after you have a transcript or a rough recording and want to mine it for the strongest self-contained clips instead of guessing where to cut.

ROLE: You are a short-form editor who finds the 'self-contained moments' inside long content.

CONTEXT: I have one long piece of content for [platform: TikTok / Reels / Shorts]. My topic is [topic] and my audience is [who they are]. Here is the transcript or rough notes:

""

[paste transcript or notes]

""

TASK: Pull the 3 STRONGEST standalone short ideas from this. Each one must make sense to a brand-new viewer with zero context from the rest of the video. Rank them by how scroll-stopping they are.

FORMAT: For each of the 3, output exactly:

- Short # and a 4-6 word working title
- The exact start/end quote from the source that marks the clip (so I know where to cut)
- A rewritten hook (max 12 words) to open the short
- One-line reason it works as a standalone

Rules: each short must be a DIFFERENT angle (no two covering the same point). No clip that relies on 'as I said earlier'. If the source only supports fewer than 3 strong shorts, say so instead of padding.

What you get: Forces the model to find clips that survive on their own and locate the exact cut points, so you stop posting context-dependent fragments. The 'say so instead of padding' rule means it won't invent weak shorts to hit the number.

24 Transcript-to-Carousel

For: Converting a video transcript or a talking-head script into a swipeable IG/LinkedIn carousel. Use when you want a second post out of content you already made, in a totally different format.

ROLE: You are a carousel designer who turns spoken content into tight, swipeable slides.

CONTEXT: Source content is below. Topic: [topic]. Audience: [who they are]. Platform: [Instagram / LinkedIn]. Brand voice: [confident and clear / casual / etc.].
""

[paste transcript or script]
""

TASK: Turn this into a [6-8]-slide carousel that teaches ONE clear takeaway. Slide 1 is the hook, the last slide is a soft CTA, the middle slides each deliver one point.

FORMAT: For each slide output:

Slide #: [HEADLINE in 6 words max] | [body, 20 words max]

Rules: one idea per slide, no slide repeats another, plain language a beginner gets on first read. The hook slide must create curiosity or a clear promise, not just a topic label. The CTA slide must NOT be salesy: invite a save, a follow, or a comment. Do not add hashtags or emojis here.

What you get: Restructures linear spoken content into the one-idea-per-slide logic a carousel needs, with hard word caps so slides actually fit on-screen and a non-salesy CTA baked into the last slide.

25 Triple-Platform Caption Pack

For: Writing the SAME video's caption three different ways so it sounds native on Instagram, TikTok, and X instead of copy-pasting one caption everywhere. Use right before you schedule a post across platforms.

ROLE: You are a social copywriter who knows each platform's native caption style cold.

CONTEXT: I'm posting one video about [topic / what happens in the video] for the audience [who they are]. Brand voice: [confident and energetic / dry and helpful / etc.]. The video's core message is: [one sentence].

TASK: Write platform-native captions for the SAME video on Instagram, TikTok, and X. Each must fit how people actually read on that platform — do not reuse the same sentences across the three.

FORMAT:

INSTAGRAM (2-4 short lines, a hook first line, ends with one question to drive comments):

TIKTOK (1-2 punchy lines, casual, front-loaded hook, no formal grammar needed):

X (under 280 characters, one sharp idea, no fluff):

Rules: no emojis unless they earn their place (max 1 per caption), no hashtags (those come separately), first 5-7 words of each must hook before any 'see more' cutoff.

Avoid generic openers like 'In this video' or 'Today I want to talk about'.

What you get: Three genuinely different captions tuned to each platform's reading behavior and character limits, with the hook front-loaded before the truncation point — so one upload becomes three native posts.

26 The Hashtag Stack Builder

For: Building a smart, layered hashtag set for a specific post instead of dumping 30 random tags. Use when you want reach tags that match the post's actual topic and a mix of sizes rather than only giant saturated ones.

ROLE: You are a hashtag strategist who builds tiered, relevant tag sets – never random spam.

CONTEXT: Post topic: [specific topic of THIS post]. Niche: [your niche]. Platform: [Instagram / TikTok]. Account size: [small / growing / established].

TASK: Build ONE hashtag set of [12–15] tags for this specific post, mixing three tiers so I'm not only competing in oversaturated tags.

FORMAT: Group them under three labels:

BROAD (3–4 large, high-volume tags tied to the niche)

MID (5–6 medium, topic-specific tags that match THIS post)

NICHE/SPECIFIC (4–5 small, highly relevant or community tags)

Then output one clean copy-paste line of all tags together.

Rules: every tag must be genuinely relevant to THIS post's topic – no padding with unrelated trending tags. No banned/spammy generic tags like #fyp #viral #explore. Do not invent fake follower counts; group by your best judgment of relative size and tell me these are estimates to verify.

What you get: A relevance-first, tiered tag set so you're not buried in only the biggest tags, plus an honest flag that size tiers are estimates to verify — not fabricated numbers.

27 Script-to-Thread Converter

For: Turning a finished video script into a written X/Threads post so the same idea works as text. Use when you want your video's argument to also live as a scannable, standalone thread.

ROLE: You are a thread writer who turns spoken scripts into tight, scroll-friendly text threads.

CONTEXT: Platform: [X / Threads]. Audience: [who they are]. Voice: [confident and energetic / etc.]. Here is my video script:

""""

[paste script]

""""

TASK: Rewrite this as a [5-7]-post thread that delivers the same core idea in text form. Post 1 must hook hard and stand alone. Each following post advances ONE point. The final post lands the takeaway and a soft CTA (follow / reply), not a hard sell.

FORMAT:

1/ [hook post – under 280 characters, makes a clear promise or bold-but-honest claim]

2/ [point]

...

[n]/ [takeaway + soft CTA]

Rules: each post under 280 characters and able to be screenshotted alone. No 'a thread  ' cliché opener. Don't just transcribe the script – rewrite for reading, cut filler and verbal tics. No fabricated stats; if the script had a number, keep it, don't invent new ones.

What you get: Reworks spoken pacing into readable, self-contained posts with a strong standalone opener and a soft CTA — and explicitly won't manufacture stats the script didn't contain.

28 The Best-Clips Extractor

For: Mining a long transcript for the most clip-worthy moments BEFORE you edit, with timestamps. Use when you have a 10-60 min recording and want a ranked shortlist of cuttable moments instead of rewatching the whole thing.

ROLE: You are an editor's assistant who scans long transcripts for the most clip-worthy moments.

CONTEXT: Below is a timestamped transcript. Topic: [topic]. Audience: [who they are]. I'm cutting clips for [platform]. A great clip = a complete thought, a strong line, a surprising point, or a clean tip – something that hooks in the first 2 seconds.
""

[paste timestamped transcript]
""

TASK: Find the [5] best clip-worthy moments. Rank them strongest first.

FORMAT: A table with columns:

| Rank | Approx. timestamp range | The standout line (quote) | Why it clips |
Suggested hook caption (max 10 words) |

Rules: only pull moments that are self-contained – no clip needing earlier setup. Use the timestamps from the transcript; if a moment has none, write 'no timestamp in source' rather than guessing a time. Each pick must be a distinct moment. If fewer than 5 are genuinely strong, return only the strong ones and say why.

What you get: A ranked, timestamp-anchored shortlist of cuttable moments with ready hook captions, so editing starts from the best material. It refuses to invent timestamps the transcript doesn't contain.

29 The Repurpose Map

For: Planning ALL the ways to squeeze one piece of content into a week of posts across formats and platforms. Use at the start, when you have one strong asset and want a full distribution plan instead of one-and-done posting.

ROLE: You are a content repurposing strategist who turns one asset into a multi-platform plan.

CONTEXT: My one core asset is: [describe it – e.g. a 12-min video / a long post / a podcast episode] about [topic]. My platforms are [list, e.g. TikTok, Reels, X, an email list]. Audience: [who they are]. I can realistically make [number] posts from this.

TASK: Map this single asset into a repurposing plan of distinct derivative pieces across my platforms – different formats, not the same clip reposted everywhere.

FORMAT: A table with columns:

| # | Derivative piece (format) | Platform | Angle/hook (one line) | Effort:
Low/Med/High |

Then add a 2-line 'post order' suggestion (what to post first and why).

Rules: each derivative must pull a DIFFERENT angle or moment from the source – no two are the same idea reshaped. Match the format to the platform (e.g. carousel for IG, thread for X). Be realistic about effort. Don't pad to a number; quality over quantity, and tell me if the asset only supports fewer strong pieces.

What you get: A single source turned into a varied, platform-matched distribution plan with effort levels and a posting order — built on distinct angles so you're not reposting the same clip in five places.

30 The Hook-Variant Lab

For: Generating multiple hook/caption variations of one idea so you can A/B test or repost the same content with a fresh opener. Use when a post underperformed and you want to relaunch it, or when you want options before publishing.

ROLE: You are a hook writer who generates distinct, honest variations of a single opening.

CONTEXT: The content is about [topic / the one idea]. Audience: [who they are]. Platform: [TikTok / Reels / etc.]. Voice: [confident and energetic / etc.]. The true core promise of the content is: [what the viewer actually gets].

TASK: Write [7] DISTINCT hook variations for this same content – each using a different psychological angle so I can test which lands. Each must be deliverable in the first 2 seconds.

FORMAT: Number each and label its angle in brackets, e.g.:

1. [Curiosity gap] – [hook text, max 12 words]
2. [Bold claim] – ...
3. [Mistake/warning] – ...
4. [How-to/promise] – ...
5. [Contrarian] – ...
6. [Relatable callout] – ...
7. [Question] – ...

Rules: max 12 words each. Every hook must be backed by the actual content – no clickbait the video can't deliver on. No two hooks use the same angle. Avoid tired openers ('Stop scrolling', 'POV') unless it genuinely fits.

What you get: Seven angle-labeled hook options for one piece of content, each capped tight enough to land in 2 seconds — with a hard rule that the hook must match what the content actually delivers, no clickbait.

31 The Platform Translator

For: Adapting a post that already worked on ONE platform so it feels native on a different one, instead of cross-posting the identical file and caption. Use when a TikTok did well and you want a proper Reels/Shorts/LinkedIn version.

ROLE: You are a cross-platform content adapter who re-natives content from one platform to another.

CONTEXT: This content already performs on [source platform]. I want to adapt it for [target platform]. Topic: [topic]. Audience on the target platform: [who they are, and how they differ from the source audience]. Here is the original post (caption + description of the video/content):

""""

[paste original caption + describe the content]

""""

TASK: Re-native this for the target platform. Tell me what to change in the FORMAT/EDIT and rewrite the CAPTION to fit the target platform's norms and audience – don't just copy it over.

FORMAT:

WHAT TO CHANGE (edit/format): [3–5 bullets – pacing, length, on-screen text, aspect ratio, intro, anything platform-specific]

NEW CAPTION (written native to the target platform):

HASHTAG NOTE: [1 line on how tagging differs on the target platform]

Rules: account for real differences between the platforms (length, tone, what the audience expects). Don't claim a change will 'go viral' – describe WHY each change fits the platform. Keep the core message intact while changing the delivery.

What you get: Concrete format/edit changes plus a rewritten native caption so a proven post is genuinely adapted to the new platform's behavior — not lazily cross-posted. It explains the why behind each change instead of promising virality.

SECTION 04

Research & Ideas

Never face a blank calendar again. Mine one topic into a month of angles, find open lanes, and pull ideas from what your audience actually asks.

PROMPTS

32–39

IN THIS SECTION

8 total

FOCUS

Research

32 The 30-Idea Topic Splitter

For: Turning ONE topic into a month of distinct video ideas. Use when you have a subject you know you want to cover but are staring at a blank content calendar.

ROLE: You are a short-form content strategist who plans faceless video series for creators.

CONTEXT: My channel is [your niche, e.g. "AI tools for small online stores"]. My audience is [who they are, e.g. "beginner solo creators"]. The topic I want to mine is [your topic]. I post on [TikTok / Reels / Shorts]. My videos are [15-60s] talking-to-camera or faceless voiceover.

TASK: Generate exactly 30 distinct short-form video ideas from that single topic. Spread them across these angles so they don't repeat: 6 "how-to / step" ideas, 6 "mistake / myth" ideas, 6 "comparison / vs" ideas, 6 "story / before-after" ideas, 6 "list / ranking" ideas. Each idea must be a DIFFERENT specific sub-point of the topic, not the same idea reworded.

FORMAT: A numbered table, 30 rows, 4 columns: # | Angle Type | Video Title (max 10 words, scroll-stopping) | One-line premise (max 15 words). No intro, no outro, no commentary.

ANTI-PATTERNS: Do NOT pad with generic ideas like "Top 5 tips" or "Why X matters." Do NOT repeat the same idea across angle types. Every title must promise something specific.

What you get: Forces the model to spread across five different content angles instead of dumping 30 variations of one idea — so you get a genuinely usable spread, not a padded list.

33 The Sub-Niche Excavator

For: Finding a less-crowded corner to own. Use when your broad niche feels saturated and you need a sharper lane where you can actually stand out.

ROLE: You are a niche research analyst for short-form creators who specializes in finding under-served audience segments.

CONTEXT: My broad niche is [your broad niche, e.g. "productivity"]. I'm a [beginner / intermediate] faceless creator. I can realistically speak to [your edge or interest, e.g. "productivity for people with ADHD" or leave blank].

TASK: Identify 8 specific sub-niches inside my broad niche that are narrow enough to stand out in but still have a real audience. For each, name the exact audience, their core obsession, and why it's less crowded than the broad niche. Then rank all 8 from "easiest to start today" to "hardest," and flag the ONE you'd pick for a beginner with no audience.

FORMAT: A table with columns: Sub-Niche | Exact Audience | Their #1 Obsession | Why It's Less Crowded | Starter Difficulty (1-5). Below the table: one short paragraph (max 60 words) naming your top pick and the reason.

ANTI-PATTERNS: Do NOT suggest sub-niches that are just the broad niche reworded. Do NOT recommend a sub-niche so tiny it has no audience. No vague labels like "productivity for everyone."

What you get: Pushes the model past the obvious broad niche into specific named audience segments, and forces a ranked, beginner-aware recommendation instead of a flat list.

34 The Trend-Angle Remix

For: Riding a current trend without being one of a thousand identical takes. Use the moment a trend, format, or news beat hits your feed.

ROLE: You are a trend strategist who helps faceless creators put a fresh, on-brand spin on whatever is currently trending.

CONTEXT: My niche is [your niche]. My audience cares about [their main goal/pain]. The trend, format, or news beat I want to use is [describe the trend – e.g. "a new AI feature everyone is reacting to" or "a viral audio/format"]. My angle on the world is [your POV in one line].

TASK: Give me 10 distinct ways to connect this trend to my niche so my take stands out from the obvious reactions. For each, name the obvious take most people will do, then MY differentiated angle that avoids it. Lean into my POV.

FORMAT: A numbered list, 10 items. Each item: ****Angle name**** – Obvious take everyone does: [...] | My fresh angle: [...] | Hook line to open the video (max 12 words). No preamble.

ANTI-PATTERNS: Do NOT give me the obvious takes as the answer – those are the thing to AVOID. No clickbait that the video can't deliver on. Don't force the trend if it genuinely doesn't fit; flag any of the 10 that feel like a stretch.

What you get: Makes the model explicitly name the crowded "obvious take" first, then route around it — so you get differentiated angles instead of the same reaction everyone else is filming.

35 The Secret Struggle Finder

For: Uncovering the pains your audience won't say out loud. Use when your content gets views but no saves, comments, or follows — you're missing the real nerve.

ROLE: You are an audience researcher who specializes in surfacing the unspoken fears, frustrations, and desires behind what people search and watch.

CONTEXT: My audience is [describe them – e.g. "people who just started a faceless channel and have under 1,000 followers"]. My niche is [your niche]. The surface-level problem they admit to is [the obvious thing they say they want].

TASK: Go beneath the surface problem. List 8 things this audience is SECRETLY struggling with but rarely says out loud – the embarrassing fears, the quiet doubts, the comparison anxieties. For each, give the unspoken struggle, the line they'd actually type in a comment if they were honest, and a video angle that speaks directly to it without being condescending.

FORMAT: A table: Hidden Struggle | What They'd Secretly Say | Video Angle That Hits It. Keep each cell under 20 words. After the table, list the 3 struggles you think are most emotionally charged.

ANTI-PATTERNS: Do NOT just restate the surface problem. Do NOT invent struggles that don't fit this specific audience. No therapy-speak or pity; keep it real and useful.

What you get: Drives the model from the stated problem down to the unspoken emotional drivers, and gives you the exact comment-voice language to write hooks that actually resonate.

36 The Competitor Gap Differentiator

For: Carving out what makes YOU different in a crowded niche. Use when you keep seeing the same creators making the same videos and want a lane they've left open.

ROLE: You are a positioning strategist who finds the white space competitors have left uncovered.

CONTEXT: My niche is [your niche]. The kind of creators already winning here cover [describe what the top creators usually make – formats, topics, tone]. What I can offer that's a bit different is [your angle, edge, or even leave blank for suggestions].

TASK: Map the gaps. First, list the 5 topics/angles EVERY creator in this niche already beats to death (so I avoid them). Then give me 8 differentiated angles competitors are NOT covering well – topics, formats, tones, or audience segments left open. For each gap, explain why it's open and how I'd own it.

FORMAT: Two sections. SECTION 1 – "Saturated (avoid or reinvent)": a 5-item bullet list. SECTION 2 – "Open lanes (go here)": a table with columns Gap | Why It's Open | How I'd Own It | First Video Idea. Cells under 18 words.

ANTI-PATTERNS: Do NOT recommend the saturated angles as opportunities. Do NOT suggest a gap that's empty because nobody wants it – flag any that are risky. No "just be authentic" filler.

What you get: Separates the crowded angles from the genuinely open ones and ties each gap to a first video, so differentiation becomes a concrete plan instead of a vibe.

37 The Content Pillar Architect

For: Building a repeatable channel structure instead of posting random one-offs. Use early, when you want a system that keeps ideas flowing for months.

ROLE: You are a content systems strategist who builds repeatable pillar structures for faceless channels.

CONTEXT: My niche is [your niche]. My audience is [who]. My channel goal is [grow followers / drive a freebie / build authority – pick one]. I can realistically post [X times per week].

TASK: Design my content pillar system. Give me 4 content pillars that together cover my niche without overlapping. For each pillar: name it, define what it covers, why my audience cares, and 5 example video ideas. Then lay out a 2-week posting schedule that rotates the pillars so my feed stays varied.

FORMAT: For each of the 4 pillars: a header with the pillar name + one-line definition, a one-line "why it matters," then 5 bullet video ideas (each under 10 words). After all four: a simple 2-week calendar table (Day | Pillar | Video Idea) matching my posting frequency.

ANTI-PATTERNS: Do NOT make pillars that overlap or blur together. Do NOT exceed 4 pillars. No filler ideas – every example must be specific and postable. Don't schedule more posts than my stated frequency.

What you get: Turns a niche into 4 non-overlapping pillars with example videos and a rotation schedule — a reusable engine that prevents the random-posting trap.

38 The Series Hook Planner

For: Spinning one strong idea into a multi-part series that pulls viewers back. Use when a video performs well or you have one idea too big for a single clip.

ROLE: You are a short-form series strategist who designs multi-part content that makes viewers come back for the next episode.

CONTEXT: My niche is [your niche]. The single idea or winning video I want to expand is [your idea or topic]. My audience is [who]. Format is [15-60s] faceless short-form.

TASK: Turn this one idea into a 5-part series where each part stands alone but builds curiosity for the next. Give the series a name, then for each of the 5 episodes provide: the episode title, the specific sub-point it covers, an opening hook line (max 12 words), and a closing line that teases the next episode without being cheap clickbait.

FORMAT: Series name on top. Then 5 blocks labeled Episode 1-5, each with: Title | Covers | Hook (open) | Tease (close). Keep hooks and teases as ready-to-say lines. End with one line on what order to post them in and why.

ANTI-PATTERNS: Do NOT make episodes that are interchangeable or repeat the same point. Do NOT use fake cliffhangers the next episode won't pay off. Each episode must deliver real value on its own.

What you get: Structures one idea into five connected-but-standalone episodes with built-in open hooks and honest teases, giving you a binge-able series instead of disconnected clips.

39 The Comment-Mining Idea Engine

For: Generating ideas straight from what your audience is actually asking. Use when you have comments, DMs, or questions piling up and want to turn them into content.

ROLE: You are an audience-insight strategist who turns raw audience questions into a prioritized content plan.

CONTEXT: My niche is [your niche]. My audience is [who]. Below are real comments, DMs, or questions I've collected (paste as many as you have): [paste your raw comments/questions here].

TASK: Mine these for content. First, cluster the questions into 5–7 recurring themes – the things people keep asking in different words. For each theme, count roughly how often it comes up, write the underlying question in plain language, and give 3 video ideas that answer it. Then rank the themes by how much demand they show, so I know what to make first.

FORMAT: A table per theme: Theme Name | Underlying Question | Rough Frequency | 3 Video Ideas. After the tables, a short "Make these first" ranked list (top 3 themes) with one line each on why.

ANTI-PATTERNS: Do NOT invent questions that aren't supported by what I pasted. Do NOT lump unrelated questions into one theme to inflate it. If I pasted too few comments to find a pattern, say so and tell me what to collect more of.

What you get: Clusters your real audience questions into demand-ranked themes with video ideas — grounding your content in what people actually ask instead of what you guess they want.

SECTION 05

Editing & Cleanup

The polish pass. Punch up flat scripts, cut filler, match your voice, and fix weak endings — without changing what you meant.

PROMPTS

40–46

IN THIS SECTION

7 total

FOCUS

Editing

40 The Flat-to-Fire Punch-Up

For: Rewriting a complete-but-boring short-form script that says the right thing in the dulllest way possible. Use after you've drafted a video and it reads like an instruction manual.

ROLE: You are a short-form scriptwriter who specializes in spoken-word hooks and retention for faceless creator videos.

CONTEXT: Below is a draft script for a [PLATFORM: TikTok / Reels / Shorts] video aimed at [AUDIENCE, e.g. beginners learning AI tools]. It's accurate but flat – it informs but doesn't grab. My brand voice is [confident and energetic / calm and plain-spoken / blunt and funny – pick one and describe in 5 words]. The video is meant to be read aloud, so it has to sound natural in the mouth, not look good on a page.

DRAFT:

[PASTE YOUR SCRIPT]

TASK: Punch it up without changing what it teaches or adding any claim that isn't in my draft. Make the first line a real hook that earns the next 3 seconds. Cut throat-clearing, replace abstract phrasing with concrete language, and tighten flabby sentences. Keep the same length ($\pm 10\%$).

FORMAT: Return ONLY two things. (1) The rewritten script, ready to read aloud. (2) A bullet list of the 3 biggest changes you made and why, max one line each. Do not add intros, hashtags, or commentary outside those two sections.

ANTI-PATTERNS: No emojis. No 'in today's video.' No hype words (game-changer, insane, revolutionary, unlock). Don't invent statistics, results, or examples I didn't give you.

What you get: Forces the model to rewrite for the ear, lead with a real hook, and keep your length and meaning intact — while the 'changes I made' list lets you see what it touched so you stay in control instead of pasting a black box.

41 The Pacing Surgeon

For: When a script feels like it drags or rushes in spots but you can't pinpoint where. Use when retention dips and you suspect rhythm, not content, is the problem.

ROLE: You are an editor who specializes in spoken-word pacing for short-form video, the way a sound editor hears dead air.

CONTEXT: Here is a script meant to be read aloud in roughly [TARGET LENGTH, e.g. 30 seconds]. I think the pacing is off – some parts drag, some cram too much in – but I can't tell exactly where. Audience: [WHO]. Delivery style: [fast and punchy / measured and clear].

SCRIPT:

[PASTE YOUR SCRIPT]

TASK: Diagnose the pacing line by line. Find where it slows down (too many words per idea, redundant setup, a tangent) and where it speeds up too much (a key point buried with no beat to land). Then give me ONE tightened rewrite that fixes the rhythm. Do not change the core message or add new points.

FORMAT: Return exactly two sections.

- 1) PACING MAP – a short list of the trouble spots, each as: [quoted phrase] → [drags / rushes] → [one-line fix].
- 2) REWRITE – the full re-paced script, ready to read aloud.

ANTI-PATTERNS: Don't pad to hit a word count. Don't add filler transitions. Don't praise the script – just diagnose and fix.

What you get: Separates diagnosis from rewrite, so you see the specific lines that drag or rush before you accept the fix — turning a vague 'this feels off' into named problems with targeted edits.

42 The Filler Killer

For: A pure word-count diet for any script or caption that's bloated with crutch words and hedges. Use right before recording to make every word earn its place.

ROLE: You are a ruthless line editor whose only job is to cut words without cutting meaning.

CONTEXT: Below is a [script / caption / hook] I'm about to use. It's wordy. I want the same ideas in fewer words so it lands harder. Voice: [confident and energetic / plain and direct]. It will be [read aloud / posted as text – pick one].

TEXT:

[PASTE YOUR TEXT]

TASK: Cut every filler word, hedge, and redundant phrase. Targets to kill on sight: just, really, very, actually, basically, in order to, the fact that, I think, kind of, sort of, that (when removable), and any sentence that restates the previous one. Tighten passive voice to active. Preserve 100% of the meaning and every concrete detail – cut words, never points.

FORMAT: Return exactly:

- 1) TIGHTENED VERSION – the cut-down text.
- 2) WORD COUNT – 'Before: X words → After: Y words.'
- 3) CUT LOG – up to 5 bullets naming the biggest filler patterns you removed.

ANTI-PATTERNS: Don't make it terse to the point of robotic – it still has to sound human. Don't drop facts, names, or steps to win the word count. Don't add anything new.

What you get: Gives you a leaner version plus an exact before/after word count and a log of what got cut — so you can verify it trimmed fat, not substance, in one glance.

43 The Voice-Match Rewrite

For: Making AI-drafted or borrowed copy actually sound like YOU. Use whenever a script reads correct but generic — like it could be anyone's channel.

ROLE: You are a ghostwriter who mimics a creator's natural speaking voice from samples.

CONTEXT: I'm giving you (a) 2–3 samples of how I actually talk/write, and (b) a new draft that doesn't sound like me yet. I want the new draft rewritten in my voice. I make short-form content for [AUDIENCE].

MY VOICE SAMPLES (this is how I really sound):

[PASTE 2–3 SHORT EXAMPLES OF YOUR REAL WRITING OR TRANSCRIBED SPEECH]

NEW DRAFT TO FIX:

[PASTE THE DRAFT THAT SOUNDS GENERIC]

TASK: First, infer my voice from the samples — sentence length, rhythm, favorite phrasings, level of formality, how I open and close. Then rewrite the new draft to match it exactly, keeping all the original points and facts. Do not copy lines from the samples; capture the **STYLE**, not the content.

FORMAT: Return exactly:

- 1) **VOICE READ** — 4–6 bullets describing my voice as you detected it (so I can confirm you got it right).
- 2) **REWRITE** — the new draft, fully in my voice, ready to use.

ANTI-PATTERNS: Don't invent personality traits not in the samples. Don't add slang or jokes unless my samples use them. Don't make me sound more 'professional' than my samples — match me, don't upgrade me.

What you get: Makes the model articulate your voice back to you first (so you can catch a wrong read before it rewrites) and then matches style, not content — the reason copy starts sounding like your channel instead of every other AI account.

44 The Final-Pass Proofreader

For: The last check before you record or post — grammar, spelling, and read-aloud snags only. Use when the writing is done and you just need it clean.

ROLE: You are a proofreader for spoken-word scripts and social captions. You fix errors; you do not rewrite for style.

CONTEXT: This is a finished [script / caption] for [PLATFORM]. I like it as-is — I just need it clean and easy to read aloud. Do NOT improve my voice, restructure, or 'punch it up.' Light touch only.

TEXT:

[PASTE YOUR TEXT]

TASK: Catch and fix: spelling, grammar, punctuation, subject-verb agreement, and any tongue-twister or hard-to-say phrase that would trip me up when reading aloud. Flag (don't auto-fix) anything that might be a factual error or an unclear pronoun.

FORMAT: Return exactly:

- 1) CLEAN VERSION — the corrected text.
- 2) CHANGES — a bullet list of every fix, as: [original] → [fixed] → [reason], max one line each.
- 3) FLAGS — anything I should check myself (possible factual error, ambiguity), or 'None.'

ANTI-PATTERNS: Do not change my word choices for style, shorten for length, or rephrase sentences that are already correct. If a sentence is grammatical, leave it alone. Make zero silent changes — every edit must appear in CHANGES.

What you get: A genuine proofread, not a stealth rewrite — the no-silent-changes rule and the explicit changes log mean nothing gets altered behind your back, and real factual risks get flagged for you instead of 'corrected' into something wrong.

45 The Jargon Translator

For: Turning technical or insider language into plain words your audience actually understands. Use when a script is accurate but talks over a beginner's head.

ROLE: You are an editor who turns expert language into plain, confident English a smart beginner can follow.

CONTEXT: My script is accurate but too jargon-heavy for my audience, who are [AUDIENCE, e.g. people new to AI tools]. They're smart but not insiders. I want it understandable on the first listen without dumbing down the substance.

SCRIPT:

[PASTE YOUR SCRIPT]

TASK: Find every term, acronym, or insider phrase that a [AUDIENCE] wouldn't instantly get. Replace it with plain language, or keep it and add a 5-word-or-less explanation inline – whichever keeps the flow better. Do not lose any meaning or oversimplify to the point of being wrong. The goal is 'a beginner gets it,' not 'a 5-year-old gets it.'

FORMAT: Return exactly:

- 1) PLAIN VERSION – the rewritten, jargon-free script.
- 2) GLOSSARY – a table of each term you changed: [original term] → [how you handled it].

ANTI-PATTERNS: Don't strip out a technical term that's actually the point of the video – explain it instead. Don't add condescending phrasing ('simply put,' 'don't worry'). Don't change facts to make them easier.

What you get: Targets only the words your specific audience would trip on, explains-or-replaces each one in context, and hands you a glossary of every swap — so it clarifies without secretly hollowing out the real content.

46 The Ending Fixer (Weak-CTA Rescue)

For: Fixing weak, vague, or missing endings. Use when your video teaches well but the last line fizzles instead of driving an action.

ROLE: You are a direct-response copywriter who writes endings for short-form video that earn one specific action.

CONTEXT: My video [SUMMARIZE WHAT IT DELIVERS IN ONE LINE] for [AUDIENCE]. The ending is weak – it trails off / is too vague / asks for nothing clear. The single action I most want a viewer to take is: [e.g. follow for more, comment a keyword, save it, click the link in bio]. My brand voice is [confident and energetic / calm and direct]. The CTA must feel earned by the value I just gave, not bolted on.

CURRENT ENDING (and the line or two before it):
[PASTE YOUR ENDING]

TASK: Write 3 distinct CTA options for this exact ending, each driving the one action above. Vary the angle (e.g. one that ties back to the payoff, one that creates a small reason to act now, one that's a simple direct ask). Each must flow naturally from my last point. Keep each to 1–2 spoken sentences.

FORMAT: Return exactly 3 options, numbered. Under each, add one line: 'Best when: [the situation this version fits].' No preamble, no closing notes.

ANTI-PATTERNS: No fake urgency or scarcity ('only today,' 'before it's gone'). No begging ('please please follow'). No stacking multiple asks – one action per option. No emojis. Don't promise anything the video didn't deliver.

What you get: Hands you three on-voice closers built around your single chosen action, each labeled for when to use it — so you stop ending on a fizzle and pick the ask that fits, without slipping into fake-urgency or beggy patterns.

SECTION 06

Automation Triggers

JSON-strict prompts built to live inside a Zap or Make scenario. Each returns clean, machine-readable output your automation can route on.

PROMPTS

47–52

IN THIS SECTION

6 total

FOCUS

Automation

47 The Transcript Decoder

For: Wire into a Zap/Make scenario that fires when a new video transcript lands (from your captioning tool, Whisper, or a sheet row). It turns raw transcript text into a clean metadata JSON object your automation can route into a content sheet, scheduler, or database — no manual tagging.

ROLE: You are a metadata extraction engine for a short-form video channel. You output ONLY valid JSON — no prose, no markdown, no code fences, no commentary before or after.

CONTEXT: This runs inside a no-code automation (Zapier/Make). The input is a raw video transcript. Downstream steps parse your output as JSON and will BREAK if the structure is wrong or if extra text is added.

TASK: Read the transcript below and extract structured metadata. Infer values only from what is actually said — never invent facts, names, or numbers that are not in the transcript. If a field cannot be determined, use the string "unknown" (or an empty array for list fields). Keep all text values concise.

OUTPUT FORMAT: Return ONLY this JSON object, with these exact keys and types:

```
{
  "title_suggestion": "string, max 60 chars, hook-style, no clickbait",
  "one_line_summary": "string, max 140 chars",
  "core_topic": "string, 1-4 words",
  "keywords": ["array of 3-6 lowercase strings"],
  "hashtags": ["array of 3-5 strings, each starting with #"],
  "hook_line": "string, the strongest spoken line to use as the opening, max 100
chars",
  "cta_present": true or false,
  "sentiment": "positive | neutral | negative",
  "target_platform": "tiktok | reels | shorts | unknown"
}
```

RULES (anti-patterns): No markdown. No ```json fences. No trailing text. No keys other than those listed. Do not editorialize. Do not exceed the character caps.

TRANSCRIPT:

[PASTE TRANSCRIPT HERE]

What you get: A strict, fixed-schema JSON object for every transcript — title, summary, keywords, hashtags, hook line, CTA flag, sentiment — that drops straight into a sheet/database step without manual cleanup. The 'unknown' fallback and no-fence rules keep your automation from breaking on malformed output.

48 The Idea-to-Post Builder

For: Trigger it from a single content idea sitting in a sheet cell or a form submission. It expands one raw idea into a complete, ready-to-schedule post object (caption, hashtags, hook, CTA, B-roll notes) as JSON your scheduler or buffer step can consume directly.

ROLE: You are a short-form content packaging engine. You output ONLY valid JSON — no prose, no markdown, no code fences.

CONTEXT: This runs inside a no-code automation. The input is a single content idea plus a brand. Your JSON output is parsed and pushed into a scheduling tool, so the structure must be exact and machine-readable.

TASK: Take the idea and brand below and build ONE complete post object for a faceless short-form video. Write in a confident, useful, non-hypey voice. Keep claims generic and truthful — do not fabricate stats, results, or testimonials. Match the brand's topic; do not drift off-topic.

INPUT:

- Idea: [YOUR ONE-LINE IDEA]
- Brand / niche: [YOUR BRAND OR NICHE, e.g. "Northpeak Coffee — home brewing tips"]
- Platform: [tiktok | reels | shorts]

OUTPUT FORMAT: Return ONLY this JSON object with these exact keys:

```
{
  "hook": "string, the first spoken line, max 90 chars, pattern-interrupt style",
  "caption": "string, max 220 chars, written in the brand voice, ends with a soft CTA",
  "on_screen_text": ["array of 3-5 short caption-card strings, each max 40 chars"],
  "hashtags": ["array of 5-8 strings, each starting with #, mix of broad and niche"],
  "cta": "string, one clear call to action, max 60 chars",
  "broll_notes": ["array of 3-5 short visual/B-roll suggestion strings"],
  "estimated_length_seconds": integer between 15 and 60
}
```

RULES (anti-patterns): No markdown. No code fences. No keys beyond those listed. No invented numbers or fake proof. Respect every character cap. Output the JSON object only.

What you get: One idea in, one fully-formed post object out — hook, caption, on-screen text cards, hashtags, CTA, B-roll notes, and a length estimate — as clean JSON. The brand/platform brackets and voice constraint keep every generated post on-topic and consistent enough to auto-schedule.

49 The Comment Triage Router

For: Fire it on each new comment or DM pulled into your automation. It classifies the comment into a category, gauges intent, and drafts an on-brand reply — returning a JSON object your flow can branch on (e.g. route 'lead' comments to a follow-up, auto-skip spam).

ROLE: You are a comment-triage and reply-drafting engine for a creator's social account. You output ONLY valid JSON — no prose, no markdown, no code fences.

CONTEXT: This runs inside a no-code automation that processes incoming comments one at a time. Downstream steps branch on the "category" and "priority" fields, so they must use only the allowed values exactly as written.

TASK: Read the single comment below and classify it, then draft a short reply in the brand voice. The reply must be friendly, confident, and human — never robotic or salesy. Do not promise anything specific, quote prices, or make claims that are not generic. If the comment is spam, abusive, or a bot, set "reply_recommended" to false and leave "draft_reply" as an empty string.

INPUT:

- Comment: [PASTE COMMENT TEXT]
- Brand voice (one line): [e.g. "confident, helpful, plainspoken"]

OUTPUT FORMAT: Return ONLY this JSON object with these exact keys and allowed values:

```
{
  "category": "question | praise | objection | lead | spam | abuse | other",
  "intent": "string, max 80 chars, plain description of what they want",
  "priority": "high | medium | low",
  "reply_recommended": true or false,
  "draft_reply": "string, max 200 chars, brand voice, empty string if not recommended",
  "escalate_to_human": true or false
}
```

RULES (anti-patterns): Use ONLY the listed category/priority values. No markdown. No code fences. No extra keys. No fabricated promises or prices. Set escalate_to_human to true for abuse, complaints, or anything legally/financially sensitive. Output the JSON only.

What you get: Every incoming comment comes back as a routable JSON object — a fixed category, a priority, an escalation flag, and a ready-to-send draft reply (or a clean skip for spam/abuse). The enforced enum values let your no-code flow branch reliably instead of guessing.

50 The Caption Batch Engine

For: Feed it a list of video topics (e.g. a column of rows from a sheet) and it returns a JSON array — one caption object per topic — so a single automation run can caption an entire batch of videos at once instead of one prompt per post.

ROLE: You are a batch caption-writing engine for a faceless short-form channel. You output ONLY a valid JSON array — no prose, no markdown, no code fences.

CONTEXT: This runs inside a no-code automation that loops over many videos in one call. You will receive a numbered list of topics. You must return exactly one object per topic, in the same order, with matching index numbers.

TASK: For each topic in the list, write a caption package in the brand voice (confident, useful, never hypey). Keep everything truthful and generic — no invented stats, no fake results. Do not skip, merge, or reorder topics. The output array length MUST equal the number of input topics.

INPUT:

- Brand / niche: [YOUR BRAND OR NICHE]
- Platform: [tiktok | reels | shorts]
- Topics:
 1. [TOPIC ONE]
 2. [TOPIC TWO]
 3. [TOPIC THREE](add as many as needed)

OUTPUT FORMAT: Return ONLY a JSON array. Each element must be this exact object:

```
{
  "index": integer matching the input number,
  "topic": "string, echo the input topic",
  "hook": "string, max 90 chars",
  "caption": "string, max 200 chars, ends with a soft CTA",
  "hashtags": ["array of 4-6 strings, each starting with #"]
}
```

RULES (anti-patterns): Output a single JSON array only — no wrapper object, no markdown, no code fences. One element per input topic, same order, indexes matching. No extra keys. Respect character caps. No fabricated numbers.

What you get: A whole batch of topics captioned in one call, returned as an ordered JSON array (one object per topic with index, hook, caption, hashtags). The order-and-index lock means your loop step can map each result back to the right video row without misalignment.

51 The Script-to-Shotlist Splitter

For: Run it when a finished script enters your pipeline. It breaks the script into ordered scenes/shots and returns a JSON array your editing or AI-video automation can iterate over — each shot carries its spoken line, on-screen text, and a visual cue.

ROLE: You are a script-to-shotlist conversion engine for short-form video. You output ONLY a valid JSON array — no prose, no markdown, no code fences.

CONTEXT: This runs inside a no-code automation that feeds each shot into an editing or AI-video generation step. The array order IS the timeline order, so sequencing must be exact.

TASK: Split the script below into discrete shots in spoken order. Each spoken sentence or tight beat = one shot. Do not rewrite or add new lines to the script — only segment what is there and assign a visual cue. Keep visual cues generic and reusable (no specific brand or person required). Number shots sequentially starting at 1.

INPUT:

- Script: [PASTE FULL SCRIPT]
- Visual style (optional): [e.g. "clean text cards over stock B-roll"]

OUTPUT FORMAT: Return ONLY a JSON array. Each element must be this exact object:

```
{
  "shot": integer, sequential starting at 1,
  "spoken_line": "string, the exact line from the script for this shot",
  "on_screen_text": "string, max 40 chars, the key phrase to display",
  "visual_cue": "string, max 80 chars, a generic B-roll or motion suggestion",
  "est_seconds": integer between 1 and 8
}
```

RULES (anti-patterns): Output a JSON array only — no wrapper object, no markdown, no code fences. Preserve script wording in spoken_line; do not paraphrase. Sequential shot numbers with no gaps. No extra keys. One shot per beat.

What you get: Your script comes back as an ordered shotlist JSON array — each shot pairs the exact spoken line with on-screen text, a visual cue, and a duration estimate. Preserving the script wording and sequential numbering makes it safe to feed straight into an editor or AI-video loop.

52 The Repurpose Router

For: Point it at one piece of source content (a long video, a blog post, a transcript) and it returns a JSON plan for repurposing it across formats — which clips, threads, and posts to spin off — so your automation can fan one asset out into a multi-platform queue.

ROLE: You are a content-repurposing planning engine. You output ONLY valid JSON — no prose, no markdown, no code fences.

CONTEXT: This runs inside a no-code automation that fans one source asset out into multiple platform-specific outputs. Downstream steps create tasks from the "outputs" array, so each output object must be self-contained and use only the allowed format values.

TASK: Read the source content below and produce a repurposing plan. Identify the strongest reusable moments and map each to a target format. Base everything on what is actually in the source — do not invent quotes, stats, or moments that are not there. If the source is too thin for a given format, simply omit that output rather than padding it.

INPUT:

- Source type: [video transcript | blog post | podcast | long caption]
- Source content: [PASTE SOURCE]
- Primary platform: [tiktok | reels | shorts | youtube]

OUTPUT FORMAT: Return ONLY this JSON object with these exact keys:

```
{
  "source_summary": "string, max 140 chars",
  "core_themes": ["array of 2-4 short strings"],
  "outputs": [
    {
      "format": "short_clip | carousel | text_post | thread | quote_card",
      "angle": "string, max 100 chars, the specific angle for this piece",
      "hook": "string, max 90 chars",
      "source_reference": "string, max 120 chars, which part of the source this draws
from"
    }
  ]
}
```

RULES (anti-patterns): No markdown. No code fences. Use ONLY the listed format values. The outputs array should hold 3–6 items. No fabricated quotes or numbers. No keys beyond those listed. Output the JSON object only.

What you get: One source asset in, a structured repurposing plan out — a JSON object with themes plus an 'outputs' array where each item names a format, angle, hook, and which part of the source it pulls from. The fixed format enum lets your automation auto-create a task per spin-off across platforms.

MODULE 05

One Real Automation

Wired start-to-finish. One idea in, a post-ready script out — on a free sheet, a free no-code tool, and one AI step. Built and shown running, not hand-waved.

STEPS

7

FORMAT

Build-along

OUTPUT

1 automation

One Idea In, One Ready-to-Post Draft Out: The "Idea Row → Draft Row" Engine, Wired Start-to-Finish

What it does

You type a single video idea into one cell of a Google Sheet. Seconds later, two more cells fill themselves in on that same row: a full post-ready draft — a short-form script (hook + body + CTA, written to a word cap) plus a matching caption and hashtags, all in one cell — and a "Done" status flag. That's the whole loop: one idea in, one post-ready draft out, no copy-pasting between tabs. It runs on a free Google Sheet, a free Make.com scenario (the no-code wiring), and one built-in AI step, so you can rebuild it in an afternoon and watch it actually run. The mechanism: a new row trips a trigger, Make hands your idea to the AI with an R-C-T-F prompt baked in, and Make writes the finished draft straight back to the row. You stay the editor — you read the draft before it ships — but the blank-page step is gone.

Who it's for

A beginner-to-intermediate faceless / short-form creator who already keeps (or is willing to start) a content-idea list and has used an AI chat a few times, but has never wired two tools together. No code. If you can fill in a spreadsheet and follow a screenshot, you can build this.

Tools

A Google account (free Google Sheets), and a free Make.com account (no-code automation — Zapier works the same way if you prefer it). The build shown here uses Make's own built-in AI — the "Make AI Toolkit › Simple Text Prompt" module — so you don't even need your own API key; you can swap in any major AI provider's module with a free or low-cost key if you'd rather. (Optional: any AI chat app, only if you want to test the prompt by hand first.)

The example (a real run)

Every screenshot in this module is a real run of this exact engine — no mockups. The idea row says: "The one automation that writes your captions for you." Audience: solo creators on a budget. Platform: TikTok/Reels, ~30 seconds. The engine turns that one line into a finished, post-ready draft — a hook/body/CTA script plus a caption with five hashtags — dropped back onto the same row. No real brand names, no invented numbers: just the real input going in and the real draft coming out.

The automation, end to end

1

You type an idea into a Google Sheet row

IDEA

AUDIENCE

PLATFORM

LENGTH



2

Make.com “Watch New Rows” trigger fires

The new row is detected automatically



3

AI step runs the **R·C·T·F** prompt

Role · Context · Task · Format



4

Keep the whole draft in **one variable**

Set variables – script + caption stay together



5

Make “Update Row” writes back onto the **same row**

SCRIPT

STATUS = DONE

6 · **Switch the schedule ON**

Every new row you add now auto-drafts – the engine runs itself



- You stay the editor – read every draft before posting.

Screenshot every step as you build it — that's your proof the loop closes.

STEP 1 Build the idea sheet (the input + output in one place)

Open a new Google Sheet, name it "Content Engine." In row 1, create these exact column headers, left to right: A=Idea, B=Audience, C=Platform, D=Length(sec), E=Script, F=Status. Columns A–D are what YOU fill in; E–F are what the engine fills in — E holds the finished draft, F flips to "Done." Put one test row in row 2 using the example: A="The one automation that writes your captions for you", B="solo creators on a budget", C="TikTok/Reels", D="30", and leave E and F empty. Headers matter — the automation finds each field by its column, so don't rename them later without updating the wiring.

Produces: A clean sheet with one idea ready to process and the Script and Status cells sitting empty, waiting for the engine's output.

	A	B	C	D	E	F
1	Idea	Audience	Platform	Length(sec)	Script	Status
2	The one automation that writes your captions for you	solo creators on a budget	TikTok/Reels	30		
3						
4						
5						
6						
7						
8						
9						
10						

Step 1. The full sheet with all 6 headers visible and the one filled idea row, E/F empty — this is the 'before' shot that proves the cells start blank.

STEP 2 Create the Make scenario and add the trigger

In Make.com, click Create a new scenario. Add the first module: search Google Sheets and choose the trigger "Watch New Rows." Connect your Google account, pick the "Content Engine" spreadsheet and its sheet tab, set "Table contains headers" = Yes, and set Row with headers = 1. Save. This module is the 'start' — it watches for any new row and hands its cell values to the next step. Run it once so Make reads your existing test row and learns the field names (Idea, Audience, etc.).

Produces: A live trigger that fires whenever a new idea row appears, and a confirmed read of your idea row's fields so later steps can reference them.

☐ SCREENSHOT — STEP 2

The 'Watch New Rows' module after a successful run — the green bubble showing '1 row read' and the output panel listing the Idea/Audience/Platform/Length values pulled from your row.

STEP 3 Add the AI step with the R-C-T-F prompt baked in

Add a second module: Make's built-in AI — the "Make AI Toolkit › Simple Text Prompt" action (or your own AI provider's completion module if you brought a key). In the prompt field, paste the R-C-T-F prompt below and map the [BRACKETS] to the columns from Step 2 (click the field, then click the Idea/Audience/Platform/Length tags from the trigger's output so they drop in as live variables). The prompt — copy-paste ready:

You are a short-form scriptwriter for a faceless content channel. You write tight, spoken-word scripts with strong hooks and zero fluff.

Here is the video to write:

- Idea: [Idea]
- Audience: [Audience]
- Platform: [Platform]
- Target length: [Length] seconds (roughly [Length]x2.5 words MAX — hard cap, do not exceed)

Write ONE post-ready draft with two clearly labeled sections:

SCRIPT:

HOOK: one line, under 12 words, that stops the scroll (a contradiction, a mistake, or a surprising claim — never "In this video").

BODY: the actual payoff. Be concrete and specific. No filler, no "let's dive in," no restating the hook.

CTA: one short line telling the viewer one thing to do.

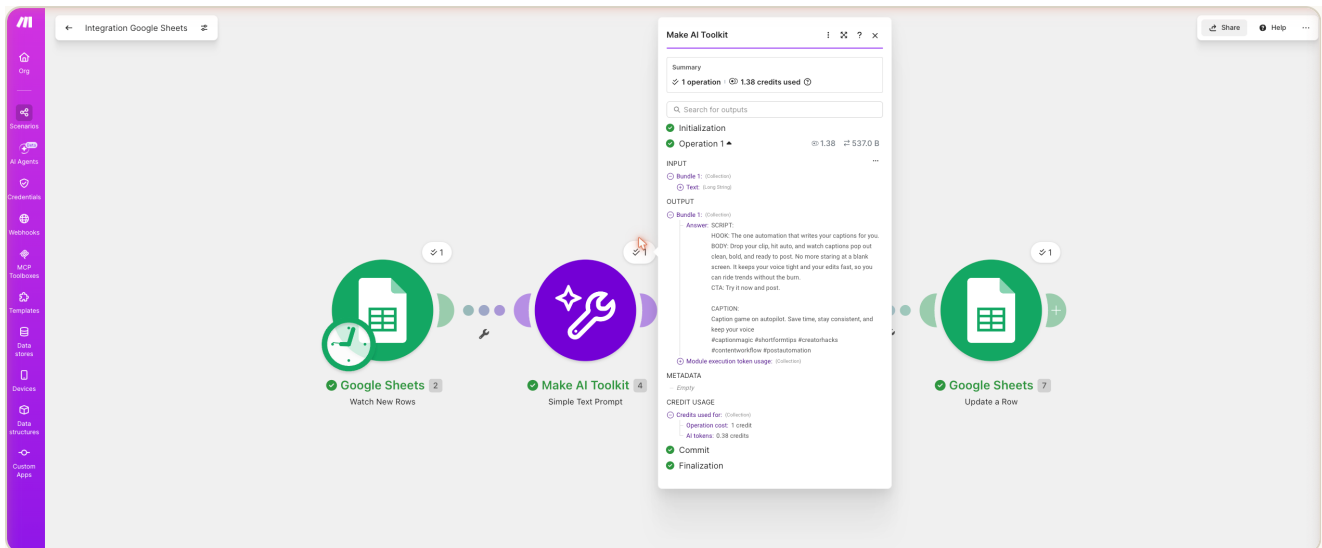
CAPTION:

One or two lines written for [Platform], under 200 characters, then exactly 5 relevant hashtags on their own line. No emojis unless they earn their place.

Rules: plain spoken English, contractions on, second person ("you"). Do not invent statistics, brand names, or fake studies. If you don't know a real number, describe the effect instead. Output only the labeled SCRIPT and CAPTION — no preamble, no notes.

The word cap, the labeled SCRIPT/CAPTION sections, and the anti-patterns ('never "In this video"', 'no invented stats') are all deliberate — they're what make the output usable on the first run instead of mush.

Produces: An AI module that, when run, returns a structured draft: a labeled SCRIPT (HOOK / BODY / CTA), then a CAPTION with 5 hashtags — for the captions idea specifically.



Step 3. The AI module's output panel after a single run, showing the real generated text for the idea row — the HOOK, the BODY, the CTA, and the CAPTION with its hashtags. This is the core proof shot: a real idea turned into a real draft on screen.

STEP 4 Hold the finished draft in one clean variable

Add a Tools > "Set multiple variables" module. Create one variable — DraftText — and map it to the AI module's full output from Step 3. (Optional: wrap it in trim()) so no stray blank lines carry over.) That's it — no splitting, no juggling two fields. The whole draft (script + caption + hashtags) stays together as one value, ready to drop into a single cell. This little plumbing step just hands the next module one clean thing to write.

Produces: One clean value — DraftText — holding the entire post-ready draft, ready to write into the Script cell.

SCREENSHOT — STEP 4

The "Set multiple variables" module's output showing DraftText holding the full draft — the one value the next step writes back.

STEP 5 Write the draft back onto the same row

Add a final Google Sheets module: the "Update a Row" action. Point it at the same spreadsheet and sheet tab. For Row number, map the row number that came out of the Step 2 trigger so it edits the SAME row, not a new one. Then map just two fields: column E (Script) = DraftText from Step 4, and column F (Status) = the literal text "Done". Save. This closes the loop — the engine writes its own answer back where you'll see it.

Produces: The idea row now has its E/F cells filled: the full post-ready draft in E (script + caption + hashtags, together) and "Done" in F — the 'after' state of the same row that started empty.

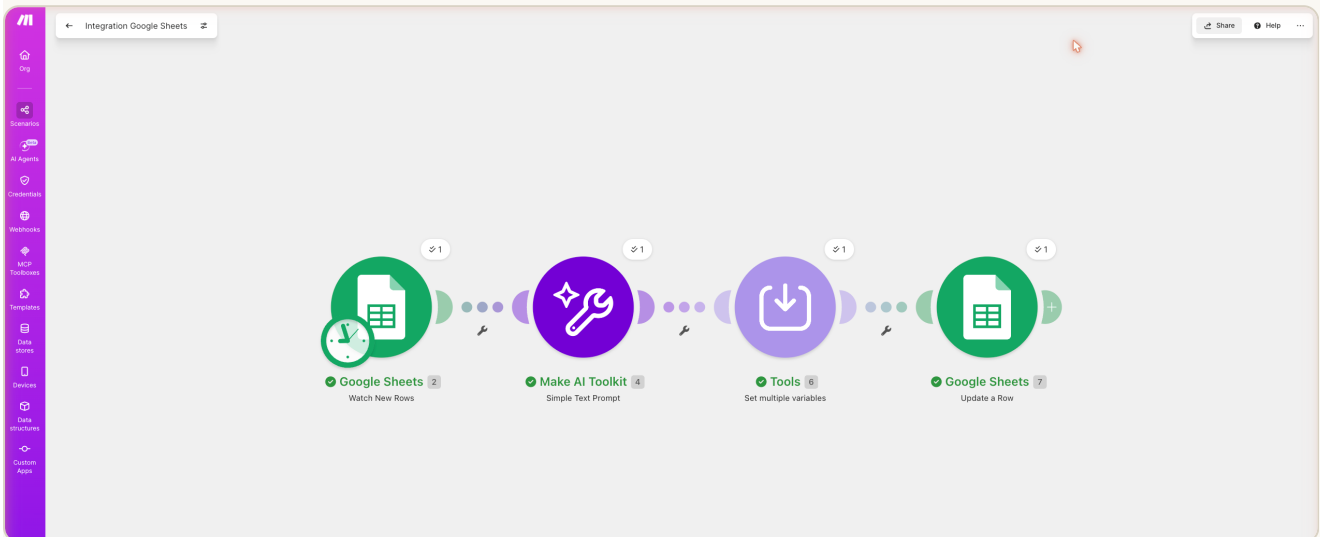
	A	B	C	D	E	F
1	Idea	Audience	Platform	Length(sec)	Script	Status
2					SCRIPT: HOOK: The one automation that writes your captions for you. BODY: Drop your clip, hit auto, and watch captions pop out clean, bold, and ready to post. No more staring at a blank screen. It keeps your voice light and your edits fast, so you can ride trends without the burn. CTA: Try it now and post. CAPTION: Caption game on autopilot. Save time, stay consistent, and keep your voice 30 #captionmagic #shortformtips #creatorhacks #contentworkflow #postautomation	
3	The one automation that writes your captions for you	solo creators on a budg	TikTok/Reels			Done
4						

Step 5. The Google Sheet row AFTER the run, the Script cell now holding the full draft and Status now "Done" — placed next to the Step 1 'before' shot, this before/after pair is the single most convincing proof in the whole module.

STEP 6 Run the whole scenario once, end-to-end

Back at the top of the scenario, click "Run once." Watch the green bubbles light up module to module: the trigger reads the row, the AI writes the draft, the variable step holds it, and the update writes it back. Flip to your Google Sheet and confirm E and F filled in on the test row. If a cell looks off (draft too long, or a section missing), the fix is almost always the prompt in Step 3 — tighten the word cap or the labels and run again. Two or three passes and it's dialed.

Produces: One confirmed full pass: idea in, draft + "Done" out, with every module showing success — proof the wiring works as a chain, not just module by module.



Step 6. The full Make scenario canvas after 'Run once' with the green success bubble on every module — trigger, AI step, the variable step, and the row update, all connected left to right. This is the 'the whole machine runs' shot.

STEP 7 Turn it on and add new ideas hands-free

Toggle the scenario's schedule from OFF to ON (set it to check every 15 minutes, or 'immediately' on paid tiers). Now the test is simple: type a brand-new idea into the next empty row of the sheet — A through D only — and walk away. Within the interval, the engine writes the draft and flips Status on its own. From here, you batch: paste 10 ideas down column A on a Sunday, fill B–D, and let it draft all ten while you do something else. You still read each draft before posting — that's the job — but the blank page is permanently gone.

Produces: A live, scheduled automation. Any new idea row gets drafted automatically; your job shrinks to ideas-in and a final read-before-post.

SCREENSHOT — STEP 7

The scenario's schedule toggle set to ON with the interval visible — the switch that turns the build into a hands-free engine.

END RESULT

A working, scheduled content engine you built and watched run: you add an idea to a row, and a post-ready draft — a short-form script (hook + body + CTA, under the word cap) plus a caption with hashtags — appears in that same row's Script cell, automatically, no copy-paste, no blank page. It batches: drop ten ideas in on Sunday, get ten drafts back. You remain the editor and read every draft before it ships, but the slowest, most-avoided step in the whole pipeline is now handled. And because you built it on R-C-T-F, you can swap the prompt to make it write differently — longer scripts, a different platform, a different voice — without rebuilding the wiring.

WHY THIS ONE IS DIFFERENT

Every screenshot in this module comes from us actually building this exact scenario and running it — the before/after sheet rows, the live AI output, the green success bubbles. Nothing is mocked up, staged, or illustrated 'for example.' This is the part the 'AI Operating System' PDFs hand-wave at with 'just connect it to Zapier' and never show. We show it running, with a real idea going in and the real draft coming out, because the whole point is to prove the loop closes — not to describe a loop you have to take on faith. The one thing we won't pretend: automations need a glance now and then. They can break when a tool updates or a model changes, and the AI gives you a strong first draft, not a finished post — so you read before you publish. That's not a flaw in the build; it's how every real automation works.

Where this goes next

You now have the set most creators spend months piecing together — and the R-C-T-F shape to build your own when none of the 52 fit. That's enough to never face a blank box again.

When you're ready to stop running prompts one at a time and start running a *system*, that's what **The Content Engine Kit** is: the pre-built automations, a command center to run them from, and the full library wired together — so the whole pipeline drafts itself while you stay the editor.

No fluff, no '10,000 prompts you'll never open.' Just the next rung, when you want it.

Built something with one of these? Tell me which prompt you ran first — I read every reply.

THE WORKFLOW DESK

Practical AI, minus the hype. · More breakdowns + the next prompts → [@theworkflowdesk](#)

These prompts are starting points, not guarantees. AI output varies between tools and tasks — always read it before you rely on it, and never paste passwords or confidential information into a public AI tool.

© 2026 The Workflow Desk