

The Repurpose System Playbook

How one source video becomes a short, an SEO blog post, and a lead magnet, and how to run the pipeline yourself.

Why I built this

I used to spend most of a day turning one video into everything else. Watch it back, transcribe it, rewrite the script for a reel, rewrite it again for a blog, record a voiceover, cut a vertical edit, write ten captions, then build a freebie to go with it. The filming took twenty minutes. The reformatting took the afternoon.

So I wired the reformatting into a pipeline. You give it one source, a YouTube video, an Instagram reel, a TikTok, or an article page, and it produces three finished assets: a 45 to 55 second vertical video posted across your platforms, an SEO blog post, and a gated PDF lead magnet. You approve the script, you review the finished video, and the machine does the parts in between.

The basic way people try this is a re-captioning app: same clip, new subtitles, maybe an auto-crop. That gets you one output in one format. It does not rebuild the script in your voice, it does not give you a blog post that can rank, and it does not build the lead magnet. This playbook covers the full pipeline and the order the stages have to run in.

01

How it actually works

1. The source goes in and the transcript comes out. A YouTube link reads the caption track, a reel or TikTok link uses a scraper, an open article gets its page text pulled. That raw text is the only input the rest of the pipeline needs.
2. Claude rewrites the transcript into a 45 to 55 second script in your brand voice, checks the word count, and retries if it is long or short. The run then pauses so you can read the script and approve it. Nothing is generated until you do.
3. Once approved, the system writes a caption for every platform and saves an Airtable row, then generates the ElevenLabs voiceover, builds the HeyGen avatar against that audio, renders the video in Remotion with captions and B-roll, and waits for you to review it before it schedules to your accounts. An optional last step reuses the same script to publish the blog post and the freebie.

There is no database behind this. The run state lives in the browser storage and two small files on disk, which is why the service worker can pick a run back up if Chrome restarts it mid-pipeline. It also generates a caption for Substack, but that one is saved to the Airtable row rather than posted, so you can paste it into a Note yourself.

The one thing almost everyone gets wrong

The stages are not interchangeable. The voiceover has to be generated before the avatar, because HeyGen lip-syncs the talking head to the actual audio file you give it. If you build the avatar from text first, the mouth never matches your cloned voice and it drifts against the captions as the clip runs.

WATCH OUT

Generate the ElevenLabs MP3 first, then pass that file to HeyGen. If you are wiring your own version, this is the ordering bug that shows up first. The second thing to keep is the script-approval pause: it costs ten seconds and it is the cheapest place to catch a bad script before you spend credits on voice, avatar, and render.

The prompts that do the rewriting

Two model calls carry the pipeline: the script rewrite and the platform captions. Here is the shape of each. Fill the brackets with your own voice notes and details.

SCRIPT REWRITE (TRANSCRIPT TO SHORT-FORM)

You are rewriting a transcript into a short-form video script.

SOURCE TRANSCRIPT:
[PASTE TRANSCRIPT]

BRAND VOICE:
[3 to 5 lines on how you sound: sentence length, words you use, words you avoid, how you open and close]

RULES:

- 110 to 135 words, which lands around 45 to 55 seconds spoken
- First line is a hook that states the payoff, no throat-clearing
- One idea per sentence, plain nouns, no hype adjectives
- No em or en dashes, use a comma or a full stop
- End on a single call to action: comment [KEYWORD] to get [WHAT THEY GET]

Return only the script text.

PLATFORM CAPTIONS (ONE SCRIPT TO MANY POSTS)

Write posting captions for this short-form video.

SCRIPT:

[PASTE FINAL SCRIPT]

CTA KEYWORD: [KEYWORD]

LINK FOR PLATFORMS THAT ALLOW ONE: [LANDING URL]

For each platform below, write a caption in its native format:

- instagram reels, tiktok, youtube shorts: hook first line, CTA to comment [KEYWORD], up to 5 hashtags
- linkedin: professional, arrow bullets, like and comment CTA, no hashtags in the body
- twitter: under 280 characters, hook then CTA
- threads, bluesky: one tight paragraph
- facebook, pinterest: short description plus the link

No em or en dashes anywhere. Return a JSON object keyed by platform.

WHERE THE SYSTEM LIVES

- AISocialHub Content Engine
<https://builds.digicuratoragency.com/content-engine/>

Set it up

1. Install the backend. In the backend folder run npm install, then copy .env.example to .env. The backend is a small Node.js server that runs on your own machine at localhost:3000.
 2. Add your keys to .env. You need a Claude key (routed through OpenRouter), an ElevenLabs API key and voice ID for your cloned voice, a HeyGen API key and avatar ID, your Airtable API key, base ID, table name and table ID, and a Blotato API key plus one account ID per platform you post to. Add an Apify token if you want to pull from Instagram or TikTok links.
 3. Ask Claude to put the keys in place for you. Open the folder in Claude Code or Codex, paste each key, and tell it to write them into .env in the right format. Then start the server with npm run dev.
 4. Check the server. Open localhost:3000/health in the browser. It returns a true or false for every key, so you can see at a glance which ones are missing before you run anything.
 5. Load the Chrome extension. Go to chrome://extensions, turn on Developer mode, click Load unpacked, and select the extension folder. Pin it to your toolbar. Activate your license in the popup the first time you open it.
 6. Do one test run. Open a YouTube video, click the extension, pick your format and B-roll style, and start. Approve the script when it pauses, then review the video when it finishes. A good first run ends with the MP4 attached to a new Airtable row and the post scheduled.
-

The commands I actually run

START THE BACKEND

```
cd backend && npm run dev
```

CONFIRM EVERY KEY LOADED

```
curl -s http://localhost:3000/health | python3 -m json.tool
```

ASK CLAUDE TO ADD A KEY THE RIGHT WAY

Add this to backend/.env exactly as [KEY_NAME]=[VALUE], keep the existing lines, do not reformat the file, then tell me to restart the server.

PULL YOUR BLOTATO ACCOUNT IDS

```
curl -H "blotato-api-key: [YOUR_BLOTATO_KEY]"  
https://backend.blotato.com/v2/users/me/accounts
```

RE-RUN ONLY THE BLOG STEP FOR A PIECE

In the extension popup, open the finished piece and turn on the blog option, then start. It reuses the saved script and publishes the post and freebie without regenerating the video.

When it breaks

ERROR	FIX
/health shows a key as false	The value is missing or has a stray quote or space in .env. Fix the line, restart with npm run dev, reload /health.
Transcript step returns nothing for a YouTube link	The video has no caption track. Pick a different source or use a reel or TikTok link with an Apify token set.
Instagram or TikTok link fails at the transcript step	APIFY_API_TOKEN is not set. Add it to .env and restart.
Avatar mouth does not match the voice	The avatar was built before the voiceover, or from text. Regenerate so the ElevenLabs MP3 is created first and handed to HeyGen.
Captions drift out of sync on a web B-roll style	The whisper model path is not set on this machine, so timing fell back to even spread. Set the model path and re-render.
Script came out off-topic	Reject it at the approval pause and tighten the brand voice lines, then re-run the script step only.
Post shows as failed in the schedule step for one platform	That platform's Blotato account ID is wrong or the connection dropped. Re-pull your account IDs and retry that platform.
Blog step never finishes	The job runs a headless Claude session and its state is only held in memory, so a backend restart loses it. Start the blog step again from the popup.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>