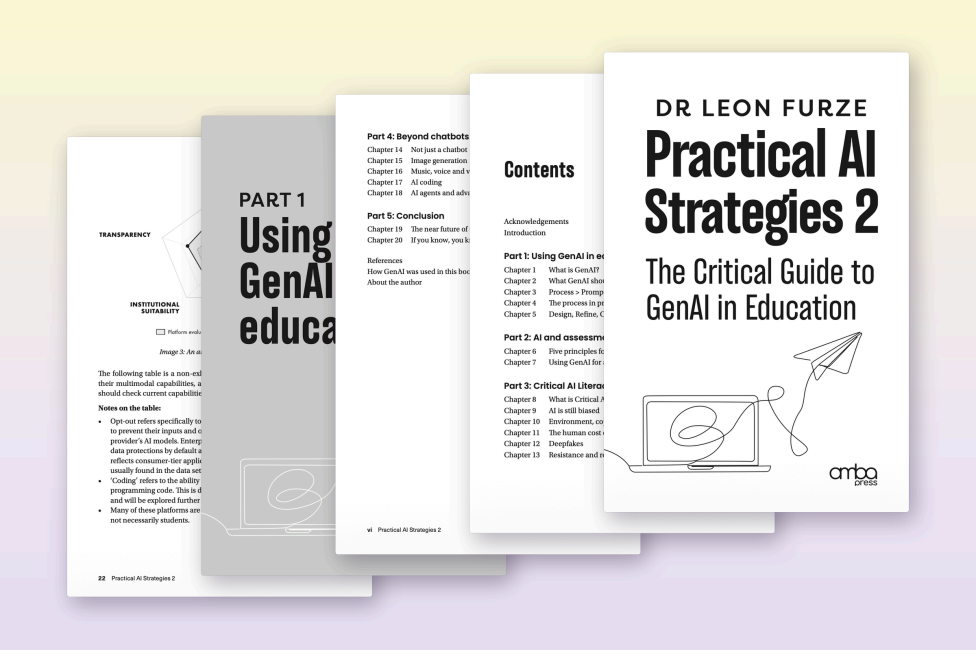
A STEP-BY-STEP GUIDE

How I Use Claude for Book Marketing

Graphics, social scheduling, planning and email automations, in the order you'd do them.

> This PDF was adapted from the blog post at https://leonfurze.com/2026/08/02/how-ive-used-claude-for-book-marketing/. Unlike the blog post, Claude Opus 5 was given free rein to produce this document. May contain AI. The original prompts have been edited for clarity and to make them more easily replicable.



Dr Leon Furze
leonfurze.com

Version 1 · August 2026

Before you start

The blog post describes a weekend of planning, building and scheduling a marketing campaign for a book, using Claude on the web and Claude Code. It runs in the order things actually happened. This is the same process, reordered as a sequence you can follow.

The prompts are the ones used, edited for clarity and to make them easier to reuse. The originals were conversational and full of asides; nothing about the requests themselves has changed.

The rules on text

> I want every bit of written content in the marketing process to be my own, so there are a few hard rules to follow:
>
> - Any text in pull quotes or captions must come verbatim from the book, my blog, or previous emails.
> - Any additional text should be written by me, directly into the prompt, or;
> - Claude can write [placeholder text] which needs to be manually edited out. Placeholder text should describe what text needs to go there, not attempt to write a draft of the actual text.
>
> FROM THE BLOG POST

Which Claude, and when

> **Claude chat for research and ideas:** Use the web or app version of Claude for bouncing ideas and doing quick research, like "how do I set up my old Facebook page?" It's faster and more user friendly, and using Code to do this would be like hammering a nail with a torpedo.
>
> **Claude Code for tools and building:** Use Claude Code for the hard stuff, like creating graphics, analysing data, or making complex use of APIs and MCPs. Code has a larger context window, more tool-use capabilities, and can iterate for hours on a problem if necessary.
>
> FROM THE BLOG POST

The other tools

Buffer schedules to Facebook, Instagram and LinkedIn, and has had an API since May 2026. **Kit** runs the mailing list and has an MCP connector. **Meta Business Suite** links the Facebook Page to the Instagram account. Claude Code installs whatever else it needs as it goes.

PART 1

Graphics from the book PDF

The typeset PDF already contains pages, figures and pull quotes. Claude Code opens it, takes its own screenshots, and composites them onto a branded canvas, using a Python script it writes for the job.

1 Put the source files in one folder

The typeset PDF, any existing promotional artwork, and the brand assets. There's no need to organise the folder or say where each file is. Naming the folder is enough; Claude will look through it.

2 Ask for one graphic

Describe the composition, and be specific about it.

PROMPT

Create a 3:2 graphic from the screenshots in this folder. Stack them with a slight offset, each with a drop shadow, arranged along a shallow diagonal. The book cover sits on top, with the remaining pages cascading below it as though laid out on a surface. Leave enough space between them to read some content, but keep the arrangement tight.

> "Prompt engineering" is not a thing. I'm just telling the model (Claude Opus 5) what I want. I'm not being specific about how to do the job — I'll leave that up to the model itself, and then review afterwards.
>
> FROM THE BLOG POST

3 Revise it in plain language

Give it something concrete to work from. An existing cover or hero image beats a colour name, because it can sample the gradient out of the file.

PROMPT · REVISION

Reverse the layout so the pages cascade left to right, still running top to bottom. Change the background to a purple and yellow gradient based on the book cover colours, sampled from the attached image.

PROMPT · REVISION

Revert to the original layout. In the left-to-right version the pages are clipped so that only the right-hand page numbers are visible. Keep the purple gradient, with slightly more yellow in it.

4 Ask for twenty

The scale-up also tells Claude to take its own screenshots from the PDF instead of using the handful taken by hand.

PROMPT

Using the same approach as the previous graphic, take screenshots of a range of pages from the book and build marketing assets in the assets/graphics folder. Include pages with text set alongside them, and a subscribe link to the newsletter. Brand fonts and related files are in the promotional materials folder.

Vary the formats: some as multiple stacked pages at 3:2, some square with a single page, and others at your discretion. Produce twenty. Keep them consistent and on brand, but varied enough that they do not repeat. QA your own work before presenting it.

Claude converts the 196-page PDF to plain text, reads the structure, renders candidate pages to a temporary scratch folder, and then looks at its own renders to pick the interesting ones. *"QA your own work"* triggers a review pass where Claude inspects its output with image recognition and fixes overlapping text and bad crops. *"Others at your discretion"* gives variety you wouldn't have specified.

5 Make the verbatim rule enforceable

PROMPT

Produce twenty more, and propose additional marketing formats of your own choosing, including two LinkedIn carousels. Summarise what you produced at the end.

One constraint: use only text taken directly from the book. Do not write original copy for these assets. Revise the first twenty to the same standard.

In response, Claude Code wrote a verifier: a script that checks every string destined for an image against the source text and fails the build if it isn't found. It was later extended across both books and the blog archive. It caught Claude paraphrasing several times.

6 Change something late

Changing the subscribe URL midway would mean reopening fifty files by hand in a design tool. Here it was one prompt, applied across every asset.

PROMPT

Update every asset to use the new subscribe link. Add launch graphics using the line "Get a Free Copy of Chapter 1" or similar, and include pull quotes selected from Chapter 1.

Page render, figure crop, pull quote, all lifted from the typeset PDF.

Same brand furniture, different layout template. Sixteen minutes from the first prompt to twenty finished graphics; seventy-odd assets by the end.

PART 2

Scheduling to social platforms

Buffer added an API in May 2026, which is what makes this stage possible. There are two human checkpoints, both stated in the first prompt.

1 Choose the assets you want to use

Go through what Part 1 produced and copy the ones you like into a new folder.

2 Ask for the system before the posts

PROMPT

The assets I have selected are in [folder]. My Buffer account is connected to LinkedIn, Facebook Pages and Instagram. Use the Buffer API to build a full content calendar for all three platforms, choosing the most suitable asset for each.

Adjust the post text to the platform: no links on Instagram, slightly longer copy on LinkedIn. Use only text taken directly from my writing. Direct quotes may be shortened with an ellipsis or adjusted in [square brackets] so they read correctly, and do not need quotation marks. Every post should close with "[Book title] is coming soon. Subscribe to the newsletter for more information."

Cadence: Instagram and Facebook should post more frequently than LinkedIn, which I manage manually and also post to organically. LinkedIn twice a week at 10pm Melbourne time; the other two platforms at 9am and 4pm Melbourne time.

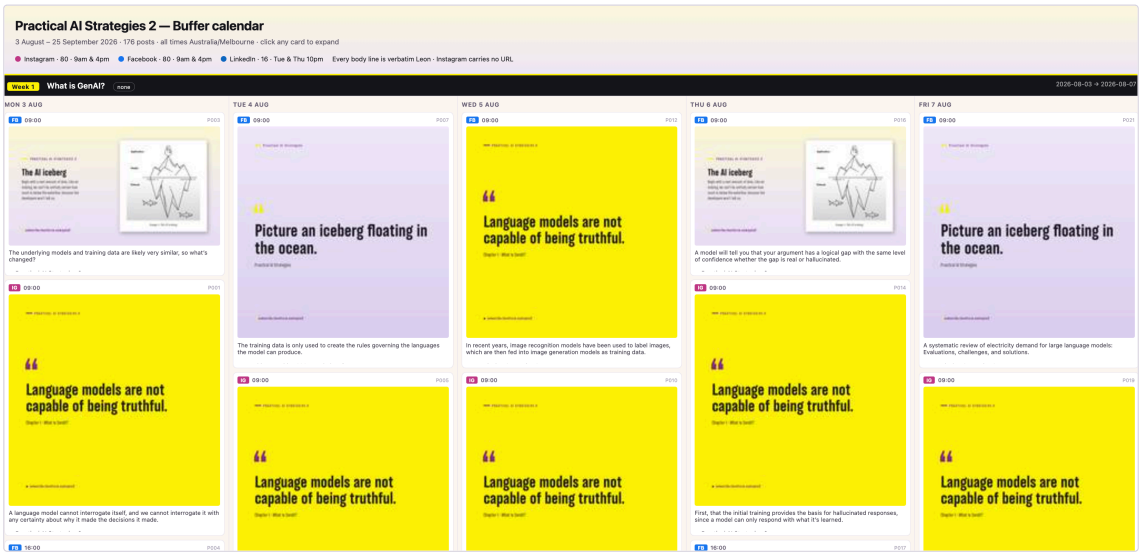
Before building anything, design the system. Produce a strategy document and a database of posts, images and accompanying text. The only link to include, and not on Instagram, is the subscribe link. Return to the books and blog posts for source text where needed, and use workflows or subagents if that helps.

I want to review the content calendar before you have API access. Render it as an HTML mock-up of the Buffer calendar. List any questions before you begin.

> It reinforces the "only use my words" rule in a few places, since that's the most important part: no slop marketing. My instructions about posting cadence are clear, and the request to design the system first exists because, in my experience, Claude Opus 5 will just rush off and start problem solving otherwise. I also asked at the end if it has any questions: these often preempt a few problems.
>
> FROM THE BLOG POST

3 Look at the calendar

> I don't trust a big job like this without several human touch-points. I certainly wouldn't want Claude to just generate and publish dozens of social media posts unattended. So, I asked for the system, and then as an afterthought asked for a visual content calendar. That turned out to be a good idea, because it was immediately obvious that Claude had made mistakes.
>
> FROM THE BLOG POST



The mock calendar, version one.

Asked whether content could be reused across three months, Claude's reading was to repeat the same image six or more times a week. The captions had been paired with the images essentially at random.

> **PROMPT · REJECTION**
> Two problems with week one. First, the same image repeats several times within a single week instead of being spread across the campaign. Second, the accompanying text reads as though sentences were selected at random from the book, and much of it does not correspond to the image it appears with. Rebuild week one and address both.

4 Read why it went wrong

The captions were random because the script had scored every sentence in the corpus and taken a high scorer, with no connection to the image. Claude had never looked at the images; it was pairing on filenames and metadata.

> The thing I'd missed is that each image already has a line of yours printed on it. So the caption shouldn't be another quote from somewhere else — it should be the passage that line comes from. Image is the hook, caption is the context, both from the same page.
>
> CLAUDE'S RESPONSE

The same response admitted the verifier had caught it paraphrasing four captions into sentences that appear nowhere in the book. Version two of the calendar was fine.

5 Handle the API key

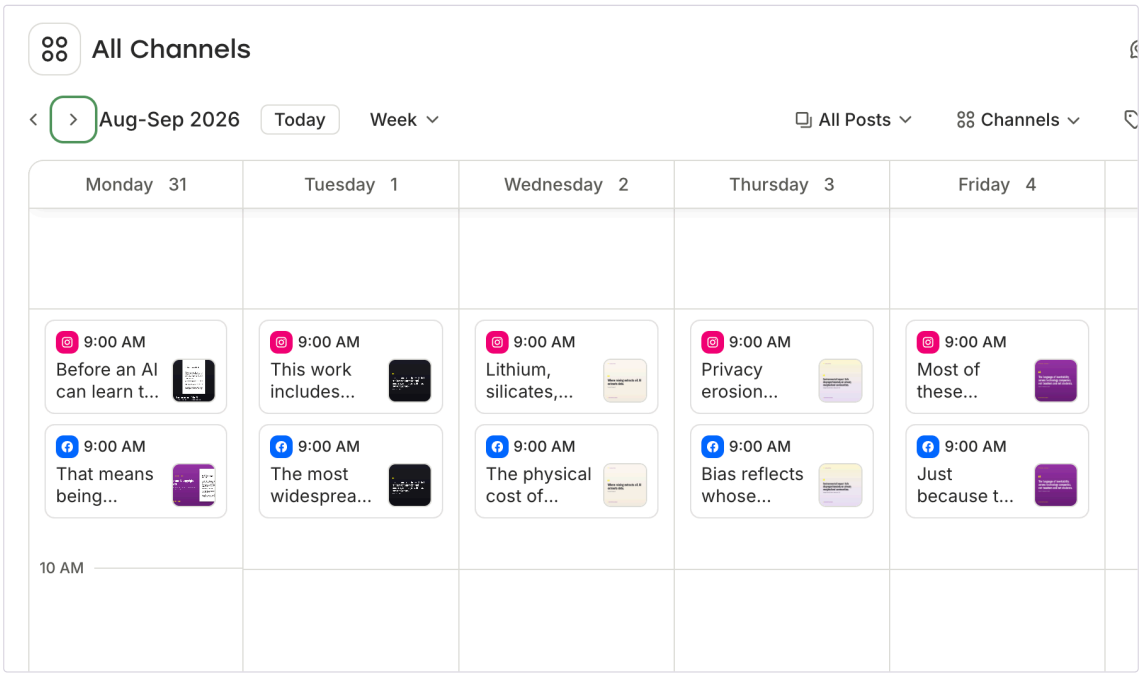
Never paste an API key into a chat. Claude Code researched the options for this setup and came back with three:

1. **Password manager CLI.** The script calls the manager at request time and holds the key in memory only.
2. **System keychain.** No install needed on macOS. The `security add-generic-password` command prompts for the value, so the key never enters shell history.
3. **A file**, outside any cloud-synced folder.

I used the keychain.

6 Check the queue

Twenty minutes to schedule the campaign across three platforms. Then a manual pass over every queued post: a quick eyeball, and a couple of edits to the wording, which was mine from the book anyway.



Scheduled posts in Buffer.

PART 3

Planning and research in Claude chat

This ran in a second window alongside Claude Code for most of the weekend.

1 Open with the plan

Describe the situation and the constraints, including the parts you're unsure about.

> **PROMPT**
> The proofs of the book are back and I want to start marketing it. I have LinkedIn and a mailing list, but I do not use Facebook or Instagram, where a significant part of the K–12 audience is.
>
> The plan is to use Buffer as middleware, managed by Claude Code, connected to Instagram, LinkedIn and Facebook. Claude Code produces the marketing assets, all pointing to a new subscribe link. All body text is taken verbatim from the book so that none of the copy reads as generic AI output.
>
> Claude Code would push posts to Buffer, presumably through an API, and I would approve each week manually before anything publishes. Assess whether this is workable and what it would require.

That prompt is what surfaced the Buffer API, which changed the plan. Chat also answered the small platform questions that otherwise turn up too late: Instagram captions can't hold clickable links, so the URL lives in the bio; new accounts get almost no organic reach at first; Buffer's API publishes but doesn't report.

2 Ask for end-to-end instructions

> **PROMPT**
> What accounts do I need for this? I have an old personal Instagram account and a current personal Facebook account, but I think this needs a dedicated page. Give me end-to-end setup instructions in order, with an indication of how long each step takes.

The answer was a sequenced checklist with time estimates, flagging which step was most likely to stall. It also argued against a book-specific page in favour of an author page, to avoid cold-starting a new audience with every book.

3 Carry context between chats

MCP connectors load their permissions at the start of a conversation. Turning on more Kit permissions meant starting a new chat, so I asked for a handover document first: the plan, the decisions and the state of play, as markdown. The next chat opened with it attached.

> **PROMPT · HANDOVER**
> Continuing from the previous conversation. You should now have broader Kit access and the ability to create email templates for the automations described in the attached markdown file. [Template name] is the base template for all of these emails.

PART 4

Email sequences over MCP

MCP lets Claude read and write inside another platform. Here it's Kit, the mailing list.

1 Bring your own plan

Kit publishes a six-month author launch plan. I edited it down to a shorter runway and handed over both versions, plus the book.

> **PROMPT**
>
> I have adapted Kit's launch plan into a shorter one. Both are attached, along with the book for reference.
>
> Review these and produce a step-by-step plan starting from today. You have a Kit connector, so you can also see the email sent today and the launch team list. That is separate from the social media campaign but may be useful context.
>
> I am open to a different approach if you think one would work better. The requirement is that as much as possible runs through Claude Code, with enough review points that I do not lose track of it, over-commit, or publish something unfinished.

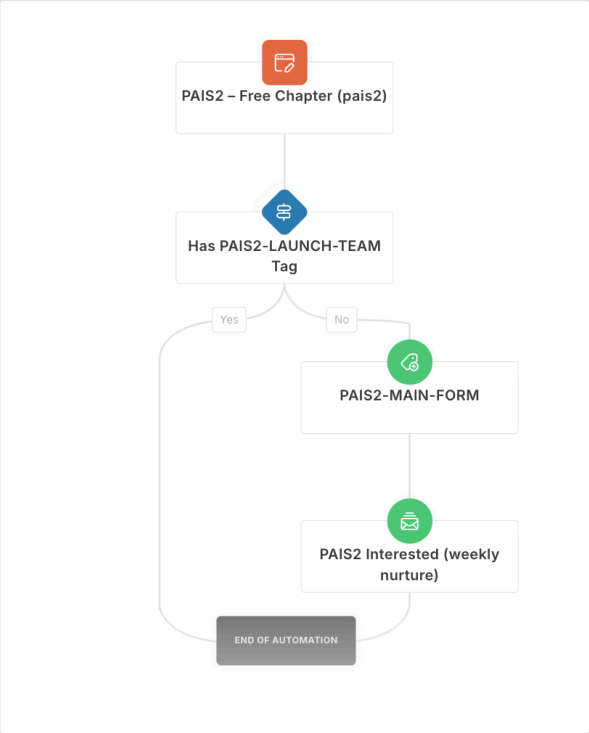
2 Drafts and placeholders

> Importantly, Claude creates the email sequences as drafts, and with placeholder text. This is really an administrative task, not a creative one, since Claude can easily handle things like timed email sequences with messages that publish after a certain number of days, or only to specific audience segments.
>
> FROM THE BLOG POST

A placeholder should say what needs writing rather than attempt it: *"[2–3 sentences on why this chapter is the one you'd most want stress-tested before print]"*.

3 Do the parts the MCP can't reach

> The Kit MCP can't handle visual automations, so I then had to go into the Kit platform, and set up the sequences that automatically tag subscribers and trigger the email sequences to send. My automations aren't complicated.
>
> FROM THE BLOG POST



The automation, built by hand in Kit.

Ask early which parts need the dashboard: the form trigger and entry conditions, the link-click rules that apply tags, file attachments, and any gate on a final email.

4 Use the connector to check your work

Asked to check the sequences from the API side, Claude found a merge tag with no fallback that would have greeted much of the list with “Hi ,”, an image hotlinked from a CDN with an expiring URL that would have broken before the email sent, and a bulk operation that had applied to only the visible page of results rather than everything matching the filter.

Principles

These are from the end of the blog post.

Start and end with your own writing

[…] Even if you think AI sounds more polished than you, you sound better than AI. Write first, and let AI draw on your writing.

Claude chat for research and ideas

Use the web or app version of Claude for bouncing ideas and doing quick research, like "how do I set up my old Facebook page?" It's faster and more user friendly, and using Code to do this would be like hammering a nail with a torpedo.

Claude Code for tools and building

Use Claude Code for the hard stuff, like creating graphics, analysing data, or making complex use of APIs and MCPs. Code has a larger context window, more tool-use capabilities, and can iterate for hours on a problem if necessary.

Plan and work in systems

Don't treat everything like an isolated task. As often as possible, try to connect the dots between different chats so that you're building systems. Systems can be repeated later — now I have built one set of graphics, I know how to do it next time.

Try it once, then scale until it breaks

Similar to the above, once you have something that works, push at it from a few directions. Do it bigger, or broader, or more specific. Take something that works and put it into a new context until it doesn't, then figure out why it broke.

Expertise is the only real failsafe

I know my own book well enough to spot errors. I know enough about APIs to understand why I shouldn't copy/paste keys into a chat. I can do enough coding to ask the right questions in Claude Code. A little bit of expertise goes a long way, and saves you time and risk.

The steps in order

1. Source files in one folder. The typeset PDF, artwork, brand assets.
2. Claude chat: describe the plan, ask what's possible, ask for setup instructions.
3. Claude Code: one graphic. Describe the composition and be specific.
4. Revise two or three times, then scale to twenty. Ask it to QA its own work.
5. State the verbatim rule, then ask for a verifier that enforces it at build time.
6. Choose the assets you like and put them in their own folder.
7. Ask for the system and a viewable calendar before handing over API access.
8. Reject version one on a single week. Read the explanation.
9. Keys through a password manager or the keychain, never in a chat.
10. Push the schedule, then check every queued post.
11. Email: bring your own plan; sequences as drafts with bracketed placeholders.
12. Build the visual automations by hand, then use the connector to check them.

The full account, including everything that went wrong, is at leonfurze.com. The Claude Code transcript is linked from there.

Dr Leon Furze · leonfurze.com · Version 1, August 2026
Set in Futura and Palatino. Typeset with Typst.