



FREE PREVIEW

The company that started running itself

[Chapter 1 – read the first chapter](#)

MAGIC MONDAY

Chapter 1 — Everyone tells you the what, nobody the how

I've tried it all.

I've sat there hand-wrangling HTML to get a page to line up — a whole afternoon battling a `div` that just wouldn't do what I wanted, until at half past eleven at night I thought: this isn't my trade, this *shouldn't* be my trade. I hired a website builder, and then another when the first one didn't get what I meant. I paid an ad agency that handed me a report I should have been proud of but couldn't do anything with. And I took an SEO course in which an infectiously enthusiastic man spent a whole day telling me exactly *what* I needed to do.

And that's where the sting was. The builder, the agency, the course — they all told me the **what**. What a good website has. What a good ad does. What Google likes to see. But nobody told me the **how**. And the how — which I discovered the hard way, as I went — is exactly where all your time disappears. The what fits on a slide. The how costs you your weekends.

That, if I'm being completely honest, is where Magic Monday began. Not with a vision of a self-running company. Not with a whiteboard full of arrows and the words "AI Operating System" in bold letters. It began with irritation. With the quiet, recurring frustration of an entrepreneur who understands *what* he has to do, and yet ends up alone every single time with the question: yes, but *how exactly*.

I've known that frustration longer than Magic Monday has existed, by the way. It's exactly the same one that, years earlier, in 2014, got me to start PayIBAN: making payments easier, simply with your IBAN. iDEAL existed, PayPal existed, and *still* you had to arrange all sorts of things as an entrepreneur just to get it working. *Why is this so hard?* It's the same question, in a different coat. That question is the primal engine underneath everything I build. And in the summer this part is about, it came back — no longer about paying, but about being found.

What I wanted

My first question was small and concrete. Almost embarrassingly small, in hindsight, for something that would grow into a system that builds entire apps. But that's how it always begins, and that's exactly the point.

I wanted to be found better.

No platform. No agents. No “AI OS” — words I wouldn’t dare say out loud until hundreds of releases later without feeling like a fantasist. I was after one tool. One tool to come up with keywords: short tail, long tail, the terms my customers were *actually* searching for and not the terms I *thought* they were using. Because that was the first lesson the SEO course had genuinely taught me: you talk like the builder of the product, your customer talks like someone with a problem. Between those two languages lies the entire difference between being found and staying invisible.

So I built something. Nothing impressive — a few lines that could pull the data out of Google Search Console: what people typed in, which terms I already showed up for, where I stayed just below the fold. I put that data to Claude, with a question so simple I was almost ashamed to ask it:

Which short and long tail keywords should I also be using?

I expected a tidy little list. Twenty terms, maybe thirty, that I could then calmly get to work on. Something you could take in over an afternoon. Something with a beginning and an end.

And then it started shifting

What rolled out was no little list.

It was **over a thousand keywords**. And behind every term: nothing. Zero content. Over a thousand doors to pages that didn’t exist.

For a moment I just sat there looking at it. Because this is exactly the moment that doesn’t appear in any SEO course. The course stops at “use the right keywords”. The man with the slides closes his laptop, everyone nods, and only at home do you realize that *that’s* where the work didn’t end but *began*. Over a thousand terms isn’t an answer. It’s a mountain. It’s the *what* in its most merciless form: here’s what you should have, good luck with the how.

And then I did the only thing I could think of. I asked the next question.

Claude, turn all those keywords into unique articles.

That worked. Surprisingly well, even — not generic, but targeted, with the right term in the right place. But the articles had barely arrived before the mountain had simply moved. Because

now I had over a thousand articles sitting in a folder, and a website that knew nothing about them. So:

How do I actually get those articles onto my site?

And then when *that* worked — when they were genuinely on the site, neatly published — the mountain was right there again, one step further along:

How do I make each article SEO-proof? Headings, meta, internal links, structure — how do I make sure Google not only finds them, but rewards them?

See what happened there? I didn't solve a problem. I pulled on a thread, and it kept coming. Every answer gave birth to a new question. Today's how flawlessly revealed tomorrow's how. Keywords led to content, content to publication, publication to findability, findability to the inevitable next question — *does it actually work?* — and there a new module was waiting again.

I hadn't built one tool. I'd built the first room of a building whose floor plan I didn't yet know.

It also just went wrong

I don't want to give the impression that this was a smooth advance, because that's a lie entrepreneurs tell each other far too often. During the same stretch I was proud that articles were appearing on the site by themselves, I got a message from Yvette at Yes! Kinderopvang. Not an SEO customer — a PayIBAN user, but the kind of moment that taught me exactly what Part I is about, so it belongs here.

Yvette didn't want anything complicated. She wanted to send a single mandate to a customer. Click, send, done. Instead, this came up on her screen:

```
Send failed (HTTP 400): CSRF_invalid
```

To me, with my head in the code, that message was crystal clear: her session had expired, she just needed to log back in. Technically nothing was broken. The error message was completely *correct*.

But Yvette sat still. She had no idea that “CSFR_invalid” meant “just log back in”. And why would she? The system had told her a truth in a language only I spoke, and stranded her with

it. And when I dug underneath it to fix it, I found something far worse: for customers with multiple companies, the mandates were being booked under the wrong legal entity. The visible error was ugly but harmless. The invisible error beneath it was the real one.

That moment lodged itself in me, because it showed two things at once. First: being technically right is not the same as having a working product. And second: a smooth *happy path* isn't a luxury but the core of the work — because the place where the user gets stuck is exactly the place where all your effort was for nothing.

The principle

Out of those two stories — the cascade of over a thousand keywords and the error message that left Yvette stuck — came the principles this entire part rests on.

The first is about *what* you should build in the first place:

The winning module is the one that does the how, not the one that tells you the what. Advice is free and everywhere. Execution is scarce and expensive. Build — or buy — the piece that takes the execution off your hands, not the piece that explains, once more, what you ought to do.

Anyone can tell you that you need to create content. A module that actually writes, publishes and SEO-proofs the over a thousand articles — *that* is the work that matters. Start at the pain, not at the pretty. Automate first the thing that costs you your weekends.

The second is a warning I learned for good that evening with Yvette:

A technically correct error message is a product fault as long as the human can't act on it. Translate system errors into the user's next step. Not "CSFR_invalid", but "your session has expired, log in again". Make the happy path frictionless, and if it does break, point the way back.

And the third isn't a tactic but the engine under this whole book. It's the reason one SEO tool could grow into a system that runs itself:

Every answer gives birth to a question. *A system doesn't arise from a master plan, but from honestly pushing through to the next question, and the next, and the next. Whoever stops at the first answer is left with a tool. Whoever keeps asking builds a system.*

And because of that...

...I stood at the start of something I could not possibly take in at that moment.

Because that's how it goes. Keywords led to content, content to publication, publication to SEO, SEO to "does it actually work?" — and every time I thought I was done, the next question was already waiting. What was supposed to be one tool to get found better became the first room of a much larger building. And I couldn't stop, because stopping felt like leaving half the work undone. Every answer I didn't think through was a new evening of manual labor I'd lose later anyway.

Three thousand five hundred and forty-seven questions later — or, more neatly put, that many releases — that building was there. Not as a grand design, but as the sum of thousands of times "yes, but how exactly". A fixed set of modules that runs my business processes, and that — and this is the real miracle — have started talking to each other.

Picture a restaurant. Behind the scenes stand master chefs, twenty-four hours a day. Each chef a specialist — one in findability, another in sales, yet another in purchasing. And every morning they share their findings with one Management Agent, who at half past seven lays the ultimate menu of the day in front of you. All you have to do is say yes.

But we're nowhere near that yet. In this part we have only one chef in the kitchen, who has just learned that keywords without content are worth nothing, and that an error message the customer doesn't understand isn't an error message but a wall. Before there can be a menu, those chefs first have to get to know each other. They have to learn to work together, share data, build on each other's work without getting in each other's way.

That's chapter 2.

The keyword and content module from this chapter didn't stay a story. They're the first two modules you can pick up today at magicmonday.nl — built in more iterations than I care to admit, so that you don't have to start at over a thousand empty doors.

✓ Checklist online

De praktische checklist bij dit hoofdstuk — afvinkbaar — staat online op **magicmonday.nl/boek**. Scan en werk hem daar stap voor stap af.



Scan — maak & volg deze checklist op magicmonday.nl/boek

That was chapter 1.

Read on in the full book

Download the complete book for free at
magicmonday.nl/en/boek