vinvu software

2027 EDITION

BACK TO THE 90S

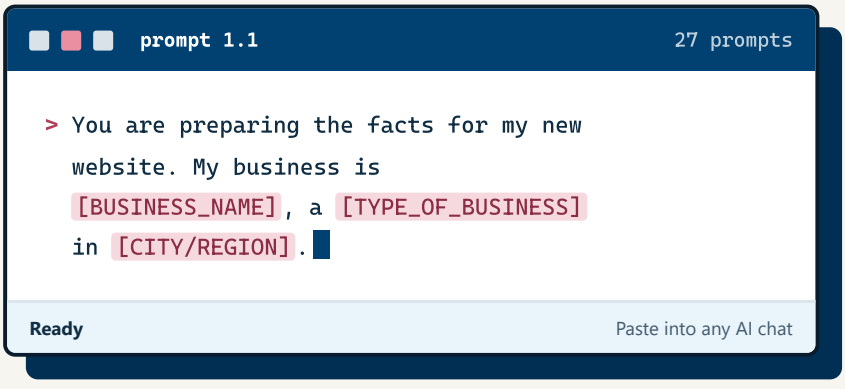
Prompt pack

> Paste-ready prompts for each step of the guide. For the owner of the business.

01
A prompt for every step

02
Chat first, then the build

03
Forms and counts, optional



Vinvu Software B.V.
De Kolk 6
8255 PE Swifterbant
The Netherlands

Contact
+31 (0) 321 700 246
contact@vinvu.nl
vinvu.nl

Registration
KvK 84836776
VAT NL863402021B01

INSIDE THIS PACK

Contents

How to use this pack 3

One source of truth 4

1.1 Build the facts file 5
1.2 Interview me to close the gaps 6
1.3 Merge my answers 6
1.4 The customer test 7
1.5 The competitor gap 7
2.1 A version I can approve 7

Pages generated from the facts 8

3.1 Kickoff 8
3.2 Page plan 9
3.3 Brand and design 9
3.4 Page content and layout 9
3.5 Add a new page 9
3.6 Replacing the site you already have 10

What to hold the line on 10

4.1 Full review before launch 11
4.2 Accessibility 11
4.3 What I should check on the preview 12

Publish, then keep it true 12

5.1 Put the project where the host can read it 12
5.2 Connect the host 13
5.3 Domain and email records 13
5.4 Connect a form (optional) 14
5.5 Add analytics (optional) 15
5.6 Live checks 16
5.7 A one-page note for myself 16
6.1 Update a fact 17
6.2 Monthly check 17
6.3 Are assistants quoting me? (every few months) 18

When to choose something else 18

7.1 Does this method fit? 18

Working with Vinvu 19

8.1 Ask your chatbot to prepare a change 19

Fix prompts 19

START HERE

How to use this pack

The prompts follow the guide. Use them in order the first time. After launch, the one you will reuse is **6.1**.

Where to paste each prompt:

Guide section	Prompts	Where you paste them
One source of truth	**1.1–1.5**, **2.1**	Any AI chat (ChatGPT, Claude or similar)
Pages generated from the facts	**3.1–3.5**	An AI coding assistant, in the website project
Replacing the site you already have	**3.6**	The same project
What to hold the line on	**4.1–4.3**	The same project, then PageSpeed and the Rich Results Test in your browser
Publish, then keep it true	**5.1–5.3**, **5.6**, **5.7**, **6.1–6.3**	The project, plus your browser for the host and the domain
A form, and analytics	**5.4**, **5.5**	Optional. Skip both unless you want them
When to choose something else	**7.1**	Any AI chat. Worth running before **3.1**
Working with Vinvu	**8.1**	Only for clients whose site we look after

Working rules

- **Replace everything in** `[SQUARE BRACKETS]` before you send the prompt. Specific input produces a specific site.
- **The facts file is the single source of truth.** Do not let the assistant put a price, a date or a claim on the site that is not in that file.
- **Chat prompts and build prompts are different jobs. 1.1** to **2.1** need no install. You can finish them, approve the file, and only then open a coding assistant. If a developer is building the site, send them the approved file plus **3.1** onwards. If Vinvu is building it, **1.1** to **2.1** are the prompts that are yours.
- **In a coding assistant, start from the starter.** The free Back to the 90s starter (vinvu.nl/guides/website-starter-template) has the project and its rules in place: `AGENTS.md`, with a copy as `CLAUDE.md`. If they are present, the assistant already has the speed, indexing and accessibility rules, and the build prompts only need to say what you want. If you use another setup, tell the assistant to follow those rules from the guide.

- **When something breaks, use a fix prompt** at the end of this pack instead of starting over.
- **Formsunrise (formsunrise.com) and Fast Web Tracking (fastwebtracking.com) are optional.** The site is complete with a phone number and an email on the page, and with no analytics. **5.4** accepts any form service. **5.5** adds analytics only if you ask. Use the Formsunrise or Fast Web Tracking variants only when you have chosen those tools. AI Website Manager is different: we connect it only for clients whose sites we look after, and **8.1** is only for them.
- **Leave out anything the website does not need.** Bank details, customer lists and other private material do not belong in the chat.

PROMPTS 1.1-2.1

One source of truth

Guide section: **One source of truth**.

Goal: one facts file, a short list of open questions, and a plain-language version you can approve. Run these in any AI chat. Stay in the same conversation so the file carries forward.

1.1 Build the facts file

PROMPT 1.1 — Fill in the highlighted parts first

```
You are preparing the facts for my new website. My business is [BUSINESS_NAME], a
[TYPE_OF_BUSINESS] in [CITY/REGION].

Sources (read all of them):
- The example file business-facts.md from the Back to the 90s starter, attached. It shows the
structure to follow.
- My current website: [URL]. Read every page that contains facts: services, prices, about,
contact, FAQ, terms.
- The attached files: [e.g. price list, brochure, notes].
- [Optional: my Google Business Profile text, pasted below.]

Task:
1. Extract every concrete fact a customer needs to choose this business: services, prices (and
whether VAT is included), discounts, contract terms, availability, opening hours, address,
service area, how to book or order, common customer questions with answers, and proof (years in
business, numbers, certifications).
2. Write it as a complete business-facts.md in exactly the structure of the attached example:
the same field names and nesting, the facts as YAML between the two --- lines. Replace the
example business completely. Leave out a field rather than guess it.
3. Below the closing --- line, under "# Notes", list OPEN QUESTIONS: every fact that is
missing, vague ("price on request", "it depends") or contradictory between sources. Never guess
or fill these in yourself.

Rules:
- Facts only. No marketing copy, no adjectives like "best" or "unique".
- Every price is a number, with the currency set once in `currency` and VAT in
`pricesIncludeVat`.
- Slugs are lowercase with dashes. Text containing a colon goes between quotes.
- Never invent reviews, numbers or claims. If unsure, it goes under open questions.
- End with a short summary: how many facts you found, and the 5 most important open questions.
```

1.2 Interview me to close the gaps

Use this when the open-questions list is long. Paste it into the same chat.

PROMPT 1.2 — Fill in the highlighted parts first

```
Interview me to close the open questions for my website. Don't write marketing copy. Collect
facts.

My business: [BUSINESS_NAME], a [TYPE_OF_BUSINESS] in [CITY/REGION].

Questions still open:
[PASTE THE OPEN QUESTIONS FROM 1.1]

- Ask at most 5 questions per message and wait for my answers.
- When I answer vaguely ("it depends", "contact us for a price"), ask for a number, a range, a
from-price or an example.
- If I don't know something or don't want it on the website, note that and move on.

When we're done, give me one summary: each question with my final answer, in short factual
sentences.
```

1.3 Merge my answers

PROMPT 1.3 — Fill in the highlighted parts first

```
Here are my answers to the open questions:
[PASTE THE SUMMARY FROM 1.2]

Update the facts file with these answers. Keep the structure. Remove answered questions from
the Notes; keep anything still unanswered or that I don't want published (mark those "not on
website"). Then show me only what changed.
```

1.4 The customer test

The guide calls this the customer test. Repeat **1.2** to **1.4** until all five answers are complete.

PROMPT 1.4 — Fill in the highlighted parts first

```
Act as an AI assistant answering a customer's questions. You know ONLY what is in the facts
file, nothing else.

Answer these 5 questions as you would to a real customer, then rate each answer: complete,
partial or impossible.
1. [A PRICE QUESTION, e.g. "What does a small storage unit cost per month in Swifterbant?"]
2. [AN AVAILABILITY OR TIMING QUESTION]
3. [A QUESTION COMPARING US TO AN ALTERNATIVE]
4. [A HOW-TO-BOOK OR HOW-TO-ORDER QUESTION]
5. [A QUESTION ABOUT CONDITIONS OR CANCELLATION]

Then list exactly which facts would turn every partial or impossible answer into a complete
one. Where I cannot give a number, a range or a from-price, mark it "we do not publish this".
Add the rest to the open questions.
```

1.5 The competitor gap

Needs web search.

PROMPT 1.5 — Fill in the highlighted parts first

```
Search for [TYPE_OF_BUSINESS] businesses in [CITY/REGION] and pick the 5 most relevant
competitors of [BUSINESS_NAME].
For each, check their website and tell me in a table: do they publish prices, availability,
opening hours and booking steps (yes/no), and what customers have to contact them for.
Then tell me which facts in my facts file they don't publish. These are an advantage in search
results and AI answers. Suggest where on the site to make them visible without an extra click.
```

2.1 A version I can approve

Goal: you approve the facts in a form you can actually read. Do not approve raw YAML.

PROMPT 2.1 — Fill in the highlighted parts first

```
Turn the facts file into a clear approval document for me, written in [LANGUAGE].

Format:
- A short intro: "These are the facts that will be published on the website. Check every line.
Anything changed here changes on the website."
- One section per topic (Business details, Services and prices, Terms, Opening hours, How to
book, Frequently asked questions, Proof), with each fact as a simple line or table row. Show
prices with currency and "incl./excl. VAT".
- A final section "Still open" with anything under open questions.
- End with: "Reply 'approved', or say what to change."

Plain language, no technical terms, no file names.
```

When you want changes, paste them into **1.3**. When you reply "approved", save that version with the date. It is the content contract for the build. Design starts after this, not before.

PROMPTS 3.1-3.6

Pages generated from the facts

Guide section: **Pages generated from the facts**.

Goal: a complete site in a local preview. Run these in an AI coding assistant, inside the project. If you are not building it yourself, send this section and the approved facts file to the person who is.

You need: the approved facts file, your colours, logo and photographs, and the Back to the 90s starter opened as the project in your coding assistant.

3.1 Kickoff

PROMPT 3.1 — Fill in the highlighted parts first

```
This is my website, built as static HTML and CSS from one facts file, with almost no
JavaScript. Business: [BUSINESS_NAME], domain [DOMAIN].

1. Check that this computer has Node.js (current LTS) and Git. Tell me exactly what to install
if anything is missing, in plain language.
2. Run npm install.
3. Replace business-facts.md with the approved version I'm pasting below. Keep the structure
the project expects.
4. Set the site URL to https://[DOMAIN].
5. Run the build. If the facts file has errors, explain each one in plain language and fix it.
6. Start the local preview and give me the URL.

Rules for the whole project, and for every later change:
- Generate every page from the facts file. Never type a price or a claim into a template by
hand.
- Extend this site. Do not rebuild it on a heavier framework.
- Never invent a fact. If I ask for a sentence that is not in the facts file, stop and ask me.
- Add no script, font or library unless you have told me its size and I have agreed.
- If AGENTS.md is in the project, follow it. If it is not, follow the speed, indexing and
accessibility rules in the Back to the 90s guide.
Finish with a short report of what you checked.

[PASTE THE APPROVED FACTS FILE]
```

3.2 Page plan

PROMPT 3.2

```
Based on the facts file, propose the page structure for this site. Start from: home, one page per service, prices and availability, FAQ, contact, privacy and a 404.
For each page: URL, its one goal, the main customer question it answers, and its main call to action. Suggest additional pages only if the facts clearly justify them (for example a page per location or audience). Don't build anything yet. Wait for my approval.
```

3.3 Brand and design

PROMPT 3.3 — Fill in the highlighted parts first

```
Apply my brand across the whole site:
- Colours: primary [HEX], secondary [HEX], background [HEX]. Check that all text meets WCAG AA contrast and adjust shades if needed.
- Logo: [PATH, preferably SVG]. Put it in the header with the business name as accessible text.
- Photos: [PATHS]. Use [PHOTO] as the main image on the home page and write alt texts based on what each photo shows.
- Style: [3–5 WORDS, e.g. "calm, trustworthy, plenty of white space"].
- Reference sites for the feel: [URL_1], [URL_2]. Take the feel, never the code, text or images.
No new fonts, libraries or scripts unless I approve them after you've told me the size.
```

3.4 Page content and layout

Run once per page, starting with the home page.

PROMPT 3.4 — Fill in the highlighted parts first

```
Work on the [PAGE] page.
Lead with: [WHAT MATTERS MOST, e.g. "the price table, then what's included, then how to book"].
Main call to action: [e.g. "Reserve a unit", "Request a quote"].
Use only facts from the facts file. If something I ask for needs a fact that isn't there, stop and ask me instead of writing it.
When done, tell me which facts you used and which structured data the page has.
```

3.5 Add a new page

PROMPT 3.5 — Fill in the highlighted parts first

```
Add a page for [TOPIC] at /[url-slug]/.
Its goal: [GOAL]. It answers: [CUSTOMER QUESTION].
Use the shared layout, with a unique title and description. Link to it from [HEADER / FOOTER / PAGE]. Add facts to the facts file first if the page needs new ones, and ask me for them.
After the build, confirm the page appears in the sitemap.
```

3.6 Replacing the site you already have

Guide section: **Replacing the site you already have**. Skip this if the business has no current site.

PROMPT 3.6 — Fill in the highlighted parts first

```
This site replaces my current site at [OLD_SITE_URL].
1. Read its sitemap (try /sitemap.xml, /sitemap_index.xml and /wp-sitemap.xml) and list every
URL.
2. Here is the list of pages with search traffic from Search Console: [PASTE OR "not
available"].
3. Make a table: old URL → new URL → action (redirect, migrate content, drop and redirect).
Every old URL must end on a relevant new page, never on a 404.
4. Wait for my approval of the table.
5. Then write the redirects as permanent (301) redirects. On Cloudflare that is
public/_redirects. On another host, use the file that host applies. Migrate the content I
marked "migrate" as new pages, keeping the facts in the facts file where they belong.
6. Build, and confirm every old path has a redirect and every new page is in the sitemap.
Remind me to copy the mail records (MX, SPF, DKIM, DMARC) before the domain switches.
```

PROMPTS 4.1-4.3

What to hold the line on

Guide section: **What to hold the line on**.

Goal: proof that the site is fast, indexable, quotable and accessible before anyone outside sees it.

4.1 Full review before launch

PROMPT 4.1 — Fill in the highlighted parts first

```
Do a complete review of this site against the project rules (AGENTS.md if present, otherwise
the Back to the 90s guide). Run the build first, then check the built files:

1. Facts: compare every price, date, opening hour and claim on every page with the facts file.
List any difference.
2. Speed: total size of HTML, CSS and JavaScript per page, the largest image, any client-side
JavaScript, any external request. Explain each. The budget is: no JavaScript by default, system
fonts or one self-hosted file, responsive images in a modern format, no video or map embedded
on load, no chat widget. Analytics should be absent unless I asked for it. If a script is
present, confirm it is one deferred cookieless tag and report the size it adds. Fast Web
Tracking, if I asked for it, is about 4 KB, about 2 KB compressed.
3. Indexing: one h1 per page, unique title and description, canonical URL with the real domain,
descriptive link text, no broken internal links. The price, the hours and the town must be in
the HTML itself, not only in an image or a script.
4. Structured data: valid JSON-LD on every page, generated from the facts file, matching the
visible content.
5. Sitemap: the sitemap lists every page with the real domain. robots.txt points to it and
allows ordinary search engines and OAI-SearchBot. GPTBot is a separate choice: [ALLOW GPTBot /
BLOCK GPTBot].
6. Redirects (if I am replacing a site): every old URL redirects to an existing page.
7. Forms: the contact form posts to a real action URL, or the page shows the phone number and
email instead. No form script and no iframe.

Fix everything that breaks a rule. Then give me a short report, plus a list of anything you
couldn't check.
```

4.2 Accessibility

PROMPT 4.2

```
Check the site for accessibility against WCAG 2.2 AA:
- every image has a meaningful alt text (or alt="" if decorative)
- colour contrast of all text and buttons
- every form field has a visible label, errors are clear and not shown by colour alone
- the whole site works with a keyboard, focus is visible, the skip link works
- headings are in a logical order, links make sense out of context
- touch targets are at least 44 px on mobile
Fix what you can in the code and list anything that needs my input (for example missing photo
descriptions).
```

4.3 What I should check on the preview

PROMPT 4.3 — Fill in the highlighted parts first

```
Write me a checklist in [LANGUAGE], under 120 words, for the preview at [PREVIEW_URL].
What I should check: the facts, the photos, and the contact path (the form, or the phone number
and email).
What can wait: the final domain and search-console submission.
Remind me that later changes go through the facts file.
```

PROMPTS 5.1-6.3

Publish, then keep it true

Guide section: **Publish, then keep it true**.

Goal: the site on your domain, checked, and a note you can keep. The coding assistant does the project work. You click through the host and the domain registrar, because it cannot see your screen.

Skip **5.1** to **5.3** if someone else publishes the site for you. Skip **5.4** and **5.5** unless you want a form inbox or visitor counts. The launch checklist in the guide is met without either one, as long as the page shows a phone number and an email.

5.1 Put the project where the host can read it

PROMPT 5.1 — Fill in the highlighted parts first

```
Put this project in a private Git repository so the host can publish it.
1. Check .gitignore so no secrets, .env files or node_modules get uploaded.
2. Create a private repository called [repo-name] under [ACCOUNT]. Tell me what to click if
you can't do it yourself.
3. Commit everything with a clear message and push.
Wait for my confirmation after each step.
```

5.2 Connect the host

Cloudflare is the straightforward choice in the guide. Another static host is fine if it supports your redirects and a custom domain. Say which one you are using.

PROMPT 5.2 — Fill in the highlighted parts first

```
I want to publish this site on [CLOUDFLARE, or the host I name], connected so that a push
publishes it.
If I chose Cloudflare: Workers with static assets is the current recommendation for new
projects; Pages still works, and a _redirects file is applied on both.
1. Tell me the exact build settings to enter. The build command must generate the sitemap.
2. Give me short, click-by-click steps in the dashboard, starting from "I'm logged in".
3. Tell me what I should see when it works and what the temporary URL looks like.
I'll tell you if my screen looks different.
```

5.3 Domain and email records

PROMPT 5.3 — Fill in the highlighted parts first

```
The site must go live on [DOMAIN] (and www.[DOMAIN], one redirecting to the other). The domain
is registered at [REGISTRAR]. The host is [CLOUDFLARE, or the host I named].
Before anything changes:
1. Tell me how to list the current DNS records, and which ones I must keep so email keeps
working (MX, SPF, DKIM, DMARC, and any verification records).
2. Then guide me step by step: add the domain to the host, check the imported records against
that list, change the nameservers at the registrar, and connect both hostnames.
3. Make sure HTTPS is on and http redirects to https.
If I am replacing an existing site: the 301 redirects must be live at the moment the domain
switches. Remind me to check three old addresses right after.
```

5.4 Connect a form (optional)

Skip this if a phone number and an email on the page are enough.

Any service that accepts a normal HTML form post will do. Paste its submit URL. Keep the form's HTML, the labels and the hidden honeypot field. Add no script and no iframe.

Formsunrise is optional. Use the second prompt only if you have chosen it. It needs the Formsunrise connector in your assistant. The setup steps are at formsunrise.com/mcp. The free plan covers one website, one form and 200 submissions a month. Submissions are encrypted and stored in the EU.

Any form service:

PROMPT 5.4 — Fill in the highlighted parts first

```
Connect this site's contact form. I am not using Formsunrise unless the next line says so.

Form service: [SERVICE NAME, or "phone and email only"]
Submit URL: [URL, or "none"]

1. Keep the form as plain HTML with visible labels and the hidden honeypot field. If the form
service names its own honeypot field, rename the existing one to that name; do not add a second
one. Do not add a script or an iframe.
2. If I gave a submit URL, set it as the form's action (in this project, booking.formAction in
the facts file). If I did not, show the phone number and email from the facts file instead.
3. Build. If a service is connected, I will send a test message. Tell me how to confirm it
arrived.
```

Formsunrise, only if you have chosen it:

PROMPT 5.4 — Fill in the highlighted parts first

```
I have chosen Formsunrise for this site's contact form. Any other service is out of scope for
this prompt.

1. List my Formsunrise sites. If [BUSINESS_NAME] doesn't have one yet, create it with allowed
domains [DOMAIN], www.[DOMAIN] and the preview address.
2. Create a form called "[FORM_NAME, e.g. Contact]" and get its submit URL.
3. Set the form's action (booking.formAction in the facts file) to that submit URL. Keep the
template's form markup and labels. Get the form snippet and give the hidden honeypot field the
name the snippet uses; do not add a second one. Do not add a script or iframe.
4. Build and push. I'll send a test message; then list the latest submissions to confirm it
arrived.
```

5.5 Add analytics (optional)

Skip this unless you want counts. The guide ships with no analytics. If you add any, prefer one small cookieless script, loaded with `defer`, and name it in the privacy statement. Whether you also show a notice is your own assessment.

Fast Web Tracking is optional. It is about 4 KB, about 2 KB compressed, with no cookies: counters of page views and events rather than people. Use the second prompt only if you have chosen it. It needs the Fast Web Tracking connector in your assistant. The setup is described at fastwebtracking.com/features/ai-assistant.

A cookieless tool you name yourself:

PROMPT 5.5 — Fill in the highlighted parts first

```
I want visitor counts on this site. If I have not named a tool below, stop and add nothing.

Tool: [TOOL NAME, or "none"]
What I was told about it: [SIZE, COOKIES OR NOT, SCRIPT URL].

1. State the script size before you add it. If it sets cookies, or you cannot state the size,
stop and tell me.
2. Add that one script, deferred, in the shared head. Do not add any other analytics.
3. Update the privacy statement to name the tool and to say what it stores. Whether the site
also needs a notice remains my own assessment.
4. Build, compare the size of each page before and after, and tell me the difference.
```

Fast Web Tracking, only if you have chosen it:

PROMPT 5.5 — Fill in the highlighted parts first

```
I have chosen Fast Web Tracking. Add no other analytics.

1. State the script size first (about 4 KB, about 2 KB compressed, loaded with defer). List my
Fast Web Tracking sites. If [BUSINESS_NAME] doesn't have one yet, create it with allowed
domains [DOMAIN] and www.[DOMAIN], timezone [TIMEZONE, e.g. Europe/Amsterdam].
2. Get the tracking snippet and add that single deferred tag to the shared head. Do not add any
other analytics scripts.
3. Tag these with data-fwt-event: [e.g. the main call-to-action buttons as "cta", the contact
form as "contact-form"].
4. If any page address can contain personal details, propose path rules first.
5. Update the privacy statement to name Fast Web Tracking and to say that it sets no cookies
and stores no visitor identifiers. Whether the site also needs a notice remains my own
assessment.
6. Build, compare the size of each page before and after, and push. After my test visit, get
the overview stats to confirm it is counting. Report page views and events, never people.
```

5.6 Live checks

Run the live URL through PageSpeed Insights (pagespeed.web.dev) (mobile and desktop) and the Rich Results Test (search.google.com/test/rich-results), then paste the results.

PROMPT 5.6 — Fill in the highlighted parts first

```
Here are the PageSpeed Insights results (mobile and desktop) and the Rich Results Test output
for https://[DOMAIN]:
[PASTE]

1. Explain in plain language anything that isn't in the mid-to-high 90s, and why. The aim is 95
to 100 on mobile with real content.
2. Fix what can be fixed in the code, following the project rules, then commit and push.
3. List what comes from outside the code (DNS, host settings, third parties) and what to click
to fix it.
4. Confirm https://[DOMAIN]/sitemap.xml and /robots.txt load and use the real domain. Confirm
the price, the hours and the town are visible in the page source. If I replaced a site, confirm
these old URLs redirect: [OLD_URL_1], [OLD_URL_2], [OLD_URL_3].
5. The contact path: if a form service is connected I will send a test message, and you confirm
it arrived. If not, confirm the phone number and email are on the page. If I asked for
analytics, confirm that one deferred tag is on the live pages and no second analytics script
was added.
```

Then add the site to Google Search Console (search.google.com/search-console) and submit `https://[DOMAIN]/sitemap.xml`.

5.7 A one-page note for myself

PROMPT 5.7 — Fill in the highlighted parts first

```
Write me a one-page note in [LANGUAGE], in plain language, that I can keep:
- What I have: the website, where it is hosted ([HOST]), where the project lives, and which
accounts are in my name.
- What was checked at launch: speed scores, structured data, accessibility, sitemap submitted.
Use the results from the live checks.
- How a customer reaches me: [phone and email, and the form inbox if I connected one. Name
Formsunrise only if I chose it.]
- Counts: [none, or the analytics tool I chose. If it is Fast Web Tracking, say that it counts
page views and events, not people.]
- How I change a fact: edit the facts file, preview, then publish. Prices, hours and
availability go through that file.
- What does not need doing: no plugin updates, no security patches for a CMS, no hosting
firefighting.
Under one page.
```

6.1 Update a fact

Guide section: **Publish, then keep it true**, the change loop in figure 5. This is the prompt you will use most.

PROMPT 6.1 — Fill in the highlighted parts first

```
Here is the change: [e.g. "the 5 m² unit is available again", "prices go up 5% from 1 January"].
1. Update the facts file first, including lastUpdated (today's date).
2. Check every page and the structured data that use these facts.
3. Run the build and confirm the sitemap still lists every page.
4. Give me the preview and tell me which pages changed, in one line each.
5. Publish only after I reply "publish".
```

6.2 Monthly check

Copy the last 28 days from Search Console (Performance and Pages) and paste them in. If you chose Fast Web Tracking, and if you chose Formsunrise, the assistant can fetch those numbers when the connector is available. Otherwise leave those lines out.

PROMPT 6.2 — Fill in the highlighted parts first

```
Here is last month's Search Console data for [DOMAIN]:
[PASTE]

Analytics: [none / I chose Fast Web Tracking, so fetch last month's overview and events / other tool, results pasted below]
Form inbox: [none / I chose Formsunrise, so count last month's submissions per form / other inbox, number pasted below]

1. Summarise in plain language: clicks, impressions, top queries and top pages, compared with the previous month if included. If analytics and form numbers are included, add the most-used pages, the call-to-action events and the number of enquiries. Analytics numbers are page views and events, never people.
2. List any indexing problems or pages not indexed, with the likely cause and fix.
3. List any Core Web Vitals issues from real users.
4. Suggest at most 3 improvements, each tied to a fact or a page I could add. Only suggest facts I would need to supply. Don't invent content.
Then write me a 5-line summary of what to do next.
```

6.3 Are assistants quoting me? (every few months)

Ask 3 to 5 typical customer questions in ChatGPT, Google and one other assistant, and save the answers. The guide's example is the storage question that quoted Swift Storage because the price was text on the page.

PROMPT 6.3 — Fill in the highlighted parts first

```
Here are AI assistants' answers to typical customer questions for [BUSINESS_NAME]:
[PASTE QUESTIONS AND ANSWERS]

1. Am I mentioned? Are the facts quoted correctly (prices, availability, hours)?
2. Which competitors are mentioned instead, and what do those answers quote that my site
doesn't make clear as visible text?
3. Compare with the facts file and list the missing or unclear facts that would most likely
help.
No tricks or hidden text. Only real, visible facts I can confirm.
```

PROMPT 7.1

When to choose something else

Guide section: **When to choose something else**. Run this before **3.1** if you are unsure the method fits.

7.1 Does this method fit?

PROMPT 7.1 — Fill in the highlighted parts first

```
My business is [BUSINESS_NAME], a [TYPE_OF_BUSINESS]. The website needs to: [WHAT THE SITE
MUST DO, e.g. "explain the service, show prices, take enquiries"].
I publish new pages: [RARELY / EVERY WEEK / EVERY DAY].
It is: [A BROCHURE AND ENQUIRY SITE / A LARGE SHOP WITH STOCK AND CHECKOUT / AN APPLICATION
WITH LOGINS, COURSES OR MEMBERSHIPS / A MIX, which I'll describe].

Tell me whether a static site generated from one facts file is the right fit, using these tests
from the guide:
- a good fit for services, local businesses, practices, portfolios and campaign pages
- a poor fit when I need a visual editor because I publish every day
- a poor fit for a large shop, where products, stock and checkout belong on a commerce platform
(guides and location pages can still be static)
- a poor fit when the site is really an application
Say which parts, if any, should stay on another platform, and which pages should be this kind
of site. Do not start a build.
```

PROMPT 8.1

Working with Vinvu

Guide section: **If you want this without operating it**.

The prompts above are enough to build and publish the site yourself. **AI Website Manager** is not one of them. It is the connection we set up for clients whose sites we look after, between the site and the AI chatbot they already use. If that is your site, **8.1** replaces the build and publish steps in **6.1**. The facts file stays the source of truth either way. To have us build the site, start at vinvu.nl.

8.1 Ask your chatbot to prepare a change

For Vinvu clients only. Paste this into the AI chat where we have connected AI Website Manager.

PROMPT 8.1 — Fill in the highlighted parts first

```
Update my website from this change: [e.g. "the 5 m² unit is available again", "prices go up 5% from 1 January"].
1. Update the facts file first, including lastUpdated (today's date). Do not invent any fact I did not just give you.
2. Prepare the change on the staging copy only. Visitors must not see it yet.
3. Tell me the preview link and which pages changed, in one line each.
4. Do not update the live site until I reply "publish".
```

WHEN SOMETHING BREAKS

Fix prompts

When the build fails

FIX PROMPT — Fill in the highlighted parts first

```
The build fails with this error:
[PASTE ERROR]
Explain the cause in plain language and fix it with the smallest possible change. If it's a problem in the facts file, show me which line and why. Re-read the project rules before you start, and don't rewrite parts that work.
```

When the deploy fails

FIX PROMPT — Fill in the highlighted parts first

```
The deploy to [CLOUDFLARE or the host I use] failed. Here is the build log:
[PASTE LOG]
Explain the cause, fix it in the project, run the build locally to confirm, then commit and
push again. Wait for me if a step needs something only I can click.
```

When something looks wrong on the site

FIX PROMPT — Fill in the highlighted parts first

```
On [PAGE_URL] I see: [DESCRIBE WHAT'S WRONG, or attach a screenshot].
Find the cause, explain it, and fix it without changing anything else. Check whether the same
problem appears on other pages. If the wrong text is a fact, correct the facts file first.
```

When the score drops

FIX PROMPT — Fill in the highlighted parts first

```
The PageSpeed score for [PAGE_URL] dropped to [SCORE]. Here's the report: [PASTE].
Find what was added or changed since launch that causes this, tell me its size and impact, and
propose the lighter alternative: HTML instead of a script, a still image instead of an embed,
no chat widget, no extra font service. Don't remove anything I asked for without my approval.
```

vinvu software

WANT THIS WITHOUT OPERATING IT?

The prompts follow the guide, Back to the 90s. If you would rather have us build and look after the site, start at vinvu.nl.

From €1,500, ready within one week.

vinvu.nl/en/contact

The Netherlands
De Kolk 6
8255 PE Swifterbant
Flevoland

South Africa
E09 Emerald Bay
7140 Strand
Cape Town

Contact
+31 (0) 321 700 246
contact@vinvu.nl
vinvu.nl

© 2027 Vinvu Software B.V. · KvK 84836776 · VAT NL863402021B01