

THE CLIENT KIT

The field guide for working on someone else's website. How to get access the safe way without ever holding a password, what to ask a new client for, which CMS you are likely looking at, and how to edit a live site without breaking it. Written for a VA on day one.

IN THIS GUIDE

- 01 Why this guide: you work on their sites, not yours

- 02 The access rule: your own login, lowest role, never their password

- 03 The onboarding request: what to ask for, in one message

- 04 Access, platform by platform

- 05 Reading the site: which CMS is this?

- 06 The CMS landscape: what you will actually meet

- 07 Editing safely on a live site

- 08 What to recommend when they ask what to build on

- 09 Free versus paid: the tools that cost money

- 10 The playbook: onboard a client, in order

- 11 The mistakes that cost the most

- 12 The head game: you are a guest in their house

- 13 Offboarding, and what all this buys you · Sources

YOU WORK ON THEIR SITES, NOT YOURS

IN PLAIN WORDS

The websites you work on belong to your clients, not to you, so the first thing to sort out is not the actual work but how to get into their accounts safely. Do that part well, without ever being the person who broke something or leaked a password, and clients start trusting you with more.

Every other guide in this set assumes the website is yours. This one does not. As a VA or contractor, you almost never own the accounts or the site you are hired to improve. The client owns Search Console, the analytics, the domain, and the CMS, and your whole job happens inside their setup. That changes the first question of any engagement from "what do I do?" to "how do I get in, safely, without becoming the person who broke something or leaked a password?"

This guide answers exactly that. It is the practical layer under the SEO, AEO, and GEO work: how to request access the professional way, what to ask a new client for on day one, how to tell what their site is built on, how to edit it without taking it down, and how to hand everything back cleanly when the engagement ends. Get this part right and the client trusts you with more. Get it wrong once and the relationship never recovers.

0

Passwords you should ever hold. Every serious platform lets the client add you as your own user; that is always the right path.

~43%

Share of all websites running WordPress (W3Techs, 2026). If you learn one CMS deeply, learn this one; it is the site you will meet most.

1 click

What it takes the client to remove you when done, if you set access up correctly. That reversibility is what makes them say yes.

Two ideas run through everything here. First, **least privilege**: ask for the lowest level of access that still lets you do the job, and no more. Second, **you are a guest**: it is their house, their keys, their risk. Behave like the professional who is careful with both, and access stops being a hurdle and becomes the thing clients rehire you for.

The words you will see

This guide stays plain, but a handful of terms come up throughout. Here they are once, in one place, so nothing later trips you up. You do not need to memorize them; just know where to look back.

TERM	IN PLAIN WORDS
CMS	The system a site's content lives in and is edited through, like WordPress or Shopify.
Role / permission	How much a given login is allowed to do, from read-only up to full control.
Least privilege	Asking for the lowest role that still lets you do the job, and no more.
2FA (two-factor)	A second login step on top of the password, usually a code sent to a phone.
Slug	The last, readable part of a page's web address, like the about-us in /about-us.
Staging	A private copy of the site where you test changes before they go live.
Cache	A saved copy of a page served for speed; your edit may not show until it is cleared.
Headless	A site whose content and its design are split apart, so a developer controls how the content appears.

02 - THE ACCESS RULE

YOUR OWN LOGIN, LOWEST ROLE, NEVER THEIR PASSWORD

IN PLAIN WORDS

Never take a client's password. Instead, ask them to add you as your own separate user, with only the access you actually need. It is safer for them, and it means you can never be blamed for breaking something you were never able to touch.

There is one rule, and it holds on every platform: **get access under your own login, at the lowest role that does the job, and never take the client's password.** Google, Meta, Shopify, WordPress, Microsoft, and every registrar have a built-in way to add an outside collaborator as a separate user or linked partner. That native path is always the correct one. Asking for the client's password is the amateur move, and security-aware clients notice.

THE RIGHT WAY



- They add YOUR email as a user
- You get a limited role (Editor, not Admin)
- Your own 2FA, on your own phone
- The logs show exactly what you did
- They revoke you in one click, keep ownership

THE WRONG WAY



- You share one password between people
- One login with full access to everything
- Their 2FA breaks, or you text them for codes
- No record of who changed what
- They must reset the password to remove you

Same access, two ways to get it. The left column is what a professional asks for; the right column is the pattern that gets sites hacked and VAs blamed.

Why this is not just etiquette

These are the reasons to give the client, because they are the reasons a nervous client says yes:

1

Two-factor actually works

If you log in as the client, their 2FA code goes to their phone, so either they disable 2FA (dangerous) or you text them for a code every session (unworkable). Your own login keeps everyone's 2FA intact. This is the most persuasive point in practice.

2

The blast radius stays small

A shared password is one leak away from total compromise. Separate logins isolate the damage.

3

There is an audit trail

Platform logs show who did what. Under a shared login every action reads as "the owner," so nobody can tell your work from theirs, and a dispute has no evidence.

4

Offboarding is clean

When the work ends, the client removes one user. Nothing to rotate, no lingering keys, no "what did they still know?"

5

They stay in control

The owner can downgrade or remove you at any time without locking themselves out. Say this out loud: *you can remove me anytime with one click, and you never give up ownership.*



Privilege climbs left to right. The job almost always lives on the Editor rung: edit any page and the SEO fields, but not the plugins, theme, users, or billing. Climb higher only when a specific task forces it.

WHEN A SHARED LOGIN IS TRULY UNAVOIDABLE

Some legacy tools and hosting panels have no multi-user option. Then, and only then, share the credential through a password manager's shared vault (1Password guest access, or a Bitwarden Organization collection), never over email, chat, or a doc. The vault gives you use of the login without the plaintext living in a message, and the client can revoke it instantly. Put the 2FA seed in the same vault, and rotate the password the day the engagement ends.

03 - THE ONBOARDING REQUEST

WHAT TO ASK FOR, IN ONE MESSAGE

IN PLAIN WORDS

On your first day, send one clear message that lists every account you need, the exact level of access for each, and the email to add you under. Asking for it all at once, instead of a little at a time over weeks, is what gets you working in an hour instead of dragging on for a fortnight.

Do not drip-feed access requests over two weeks. On day one, send one organized message listing every account you need, the exact role for each, and the email address to add. Clarity here is the difference between access in an hour and a fortnight of back-and-forth. Below is the flow, then a checklist you can copy, adapt to the engagement, and paste.



Every access follows the same four beats. The last one, document, is the step most people skip and later wish they had, when it is time to prove or undo access.

SEARCH AND ANALYTICS

- Google Search Console** — Full user
- Google Analytics 4** — Editor (or Analyst for read-only reporting)
- Google Tag Manager** — Publish, on the container
- Bing Webmaster Tools** — Read/Write

CMS AND HOSTING

- WordPress** — Editor (Administrator only if configuring plugins or technical SEO)
- Shopify** — Collaborator account, scoped to Products, Themes, Online Store, Analytics
- Domain registrar / DNS** — delegated DNS-only access, or the client makes changes from your exact instructions

ADS AND SOCIAL

- Google Business Profile** — Manager (the client stays Primary Owner)
- Google Ads** — Standard user, or a manager-account link via their 10-digit Customer ID
- Meta (Facebook Page + Instagram)** — Partner access to the assets, via your Business Portfolio ID

RESEARCH TOOLS

- Semrush / Ahrefs** — you bring your own; only take a client seat if they choose to add and pay for one

ALWAYS INCLUDE

The email address the client should add. Use the same Google account for every Google product, your Microsoft account for Bing, and your partner IDs for Shopify and Meta. The single most common reason a grant fails is the wrong email, or a personal address added where a work one was meant.

04 - PLATFORM BY PLATFORM

ACCESS, PLATFORM BY PLATFORM

IN PLAIN WORDS

Each website tool, like Google Search Console or Shopify, has its own way to add you and its own name for the access level you want. This section is a lookup list: for each one, the exact role to ask for and the exact words to send, so you never have to guess or fumble.

Each row below is the role to ask for and who grants it. After the table, the exact request to send, per platform, written so a non-technical client can follow it.

PLATFORM	ASK FOR	WHO GRANTS	NOTE
Search Console	Full user	Owner, in Settings	covers all day-to-day SEO
Google Analytics 4	Editor / Analyst	Owner, in Admin	Editor to configure, Analyst to report
Tag Manager	Publish (container)	Owner, User Management	Publish so your tags go live
Business Profile	Manager	Owner, People and access	client stays Primary Owner
Google Ads	Standard, or MCC link	Owner, Access and security	manager link is the agency way
WordPress	Editor	Owner, Users	Admin only to configure plugins
Shopify	Collaborator	You request, owner approves	does not use a staff seat
Bing Webmaster	Read/Write	Owner, Settings > Users	can import from Search Console
Meta (Page/IG)	Partner access	Owner, Business Settings	never via a personal profile
Registrar / DNS	DNS-only, delegated	Owner, or does it for you	often a one-off TXT record

Google Search Console

Ask for **Full user**. It shows all data and lets you submit sitemaps, request indexing, and read every report, without the power to manage other users. If the client set up a **Domain property** (verified by a DNS record, it covers every subdomain and both http and https), you inherit everything. A **URL-prefix property** only covers that exact address, so check which exists.

COPY-PASTE REQUEST

Please add me as a **Full user** in Search Console: open Settings > Users and permissions > Add user, enter [my email], set permission to **Full**, and Add.

Google Analytics 4

Ask for **Editor** if you need to configure the property (create key events, custom dimensions, link Search Console or Ads), or **Analyst** if you only build reports and read data. Avoid Administrator. Grant it at the property level, not the account level, unless the client wants you across every property.

COPY-PASTE REQUEST

In GA4, please go to Admin > Property Access Management > +, add [my email], and set the role to **Editor** (or Analyst if you prefer read-only).

Google Tag Manager

Container permissions run Read, Edit, Approve, Publish. Ask for **Publish** if your tracking changes need to go live by your hand, which is normal when you are wiring GA4 tags, events, or schema through GTM. If the client wants to review first, ask for Approve or Edit and let them publish. You rarely need account-level Admin.

COPY-PASTE REQUEST

In Tag Manager, please add me with **Publish** rights on the container: Admin > User Management > +, add [my email], set Account to User and Container to **Publish**, Invite.

Google Business Profile

Ask for **Manager**. It lets you edit the listing, hours, photos, posts, and reviews, while the client stays **Primary Owner**. Losing primary ownership of a Business Profile is a known way businesses get locked out of their own listing, so never ask to be made owner.

COPY-PASTE REQUEST

Please add me as a **Manager** on your Business Profile (you stay Primary Owner): Settings > People and access > Add, enter [my email], choose **Manager**, Invite.

WordPress

Ask for your own **Editor** user. Editor can publish and edit every page and post and the on-page SEO fields, but cannot touch plugins, the theme, settings, or users, which is exactly the boundary you want. Ask for **Administrator** only if the task is installing or configuring an SEO plugin or technical settings, and say why. Never accept the shared `admin` login.

COPY-PASTE REQUEST

Please create my own user: Users > Add New User, email [my email], Role **Editor** (or Administrator only if I will configure plugins or technical SEO). Please do not share the main admin login.

Shopify

Do not ask for a staff account, which uses up one of the client's limited seats. Ask for a **Collaborator account** instead: you request it from your Shopify Partner dashboard, it does not count against staff seats, and you scope it to only the sections you need. Many stores require a 4-digit collaborator code before a request can be sent, so ask for that too.

COPY-PASTE REQUEST

I will send a **Collaborator request** from my Partner account (it will not use a staff seat). If collaborator requests are locked, please send me your **4-digit code** from Settings > Users and permissions, then approve my request and grant Products, Themes, Online Store, and Analytics.

Meta, Google Ads, DNS

For **Meta**, connect through Business Partner access, never a personal profile: the client adds your Business Portfolio ID under Business Settings and assigns the Page, Instagram, and any ad account. For **Google Ads**, either be added as a Standard user or link their account to your manager (MCC) account using their 10-digit Customer ID. For **DNS**, the safest pattern for a one-off (like verifying a Search Console Domain property) is to send the exact record and have the client paste it; for standing access, ask for a scoped DNS-only role (Cloudflare Members, GoDaddy Delegate Access, Namecheap Share Access), never the registrar login.

05 - READING THE SITE

WHICH CMS IS THIS?

IN PLAIN WORDS

A CMS, or content management system, is just the program a website is built and edited in, like WordPress or Shopify. Before you promise a client any work, figure out which one their site runs on, because a free browser tool can tell you in under a minute and each system is a completely different job to edit.

Before you quote content work, know what the site is built on, because a WordPress site and a headless Next.js site are two completely different jobs. You can identify almost any site in under a minute with two checks, then confirm with a third.

- **Run Wappalyzer or BuiltWith.** The browser extension loads the site and reads its CMS, framework, e-commerce platform, analytics, and host from public signals. This answers 90% of sites instantly.
- **Check for the tells.** WordPress: a `/wp-admin` login and asset paths under `/wp-content/`. Shopify: `cdn.shopify.com` assets and `/products/` URLs. Wix: `wixstatic.com`. Squarespace: `squarespace.com` assets. Webflow: `data-wf-page` attributes.
- **View source for the generator tag.** Many platforms self-identify in `<meta name="generator"> : "WordPress 6.x", "Squarespace", "Wix", "Ghost", "Hugo"`. WordPress can hide this, so absence is not proof.

CROSS-CHECK BEFORE YOU TELL THE CLIENT

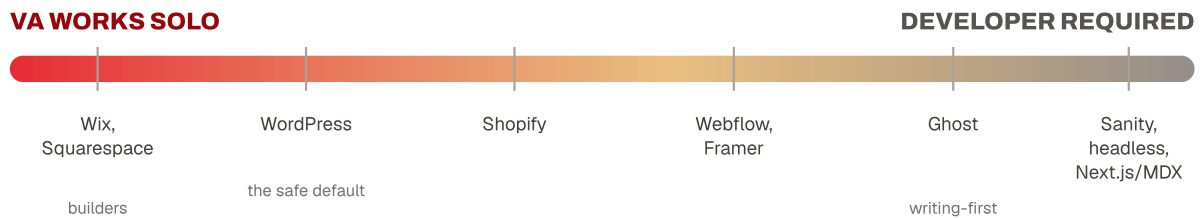
Confirm two signals before you name the platform. A `_next/` asset path with no `/wp-admin` and no CMS generator is the signature of a custom or headless build, which is your cue that a developer is involved and some SEO asks will be dev tickets, not VA tasks. Knowing this before you quote saves an awkward conversation later.

WHAT YOU WILL ACTUALLY MEET

IN PLAIN WORDS

Websites split into two groups: ones you can update on your own, and ones where a developer has to handle the structural parts for you. This section walks through the systems you will actually run into and shows which group each one falls in, so you know before you say yes whether you can finish the job alone.

There are really two families of site. On one side are platforms where a VA can be productive alone with a limited role. On the other are developer-built sites where you can fill in fields but cannot change structure or SEO without the developer. The single most useful thing to know about any CMS is which side of that line it sits on.



The line that matters is not "easy" versus "hard", it is "can I finish the job alone?" WordPress sits in the sweet spot: a limited role, free SEO plugins, and no developer needed for routine work.

CMS	ASK FOR ROLE	SEO CONTROL	THE GOTCHA
WordPress	Editor	strong, via Yoast or Rank Math	ask if a page builder (Elementor) is in use
Shopify	Collaborator	title, meta, slug on every item	changing a handle changes the URL, redirect it
Webflow	Content editor	strong and native	never touch the Designer; legacy Editor is retiring in 2026
Squarespace	Content Editor	adequate, per page	little granularity, SEO settings often need Admin
Wix	scoped role	improved, per page	closed platform, cannot swap templates after publish
Ghost	Editor / Author	native per-post meta	theme changes need a developer
Sanity, Contentful, Strapi, Payload	editor (dev-set)	only the fields a developer exposed	no visual page building, structure is code
Next.js / MDX / static	via an editing layer, if any	whatever is hardcoded	a bad edit can fail the build and down the site

WordPress, in a little more depth

Because you will meet it most, know its shape. Content lives in **Posts** and **Pages**, edited in the **Gutenberg** block editor by default. Many sites layer a page builder like **Elementor** on top, which replaces that editor with a drag-and-drop canvas, so always ask "is there a page builder?" before you quote. SEO comes from a plugin: **Yoast** or **Rank Math**, both of which add a per-page panel for the SEO title, meta description, slug, and schema. **Advanced Custom Fields (ACF)** turns the edit screen into labeled fields rather than a free body, which is safer but means you cannot restructure the page. And if you edit a page and the change does not show, the site is probably **cached** (WP Rocket and similar), so clear the cache and hard-refresh.

The headless reality

On a headless CMS like **Sanity**, the content backend and the front end are separate: you edit fields in a studio, and a developer's code decides how they render. You *can* edit the text and images a developer set up as fields. You *cannot* move a section, change the layout, add a content type, or add an SEO field that was never built. If the client wants the hero moved down and the site is headless, that is a developer request, not a VA task. Contentful and Strapi are friendlier for editors; Payload and raw MDX are the most developer-owned.

07 - EDITING SAFELY

EDITING SAFELY ON A LIVE SITE

IN PLAIN WORDS

The website is live and real visitors are on it while you work. This section is a short list of habits that stop a small edit from taking the whole site down, like making a copy first and never touching the code.

It is their live site, and real visitors are on it while you work. These habits keep a routine edit from becoming an outage or a lost afternoon. None of them are advanced; skipping them is what causes the damage.

- Use staging when the change is non-trivial.** Most managed WordPress hosts, Shopify (a duplicated unpublished theme), and Webflow (preview or branches) let you work on a copy, review, then push live. Never experiment on production.
- Draft, preview, then publish.** Every serious CMS has a draft state and a preview. Look at the rendered page before you publish; do not edit live and hit save blind.
- Duplicate before you edit.** Copy the page, product, or template first. If the edit goes wrong, the original is untouched and you revert in seconds.
- Never touch theme or template code.** WordPress Theme File Editor, Shopify `.liquid` files, and the Webflow Designer can break the entire site with one mistake. Stay in the content and SEO-field layers; code is a developer's job.
- Know that revisions and caching have limits.** WordPress keeps post revisions you can restore, but a cache or CDN can make a bad edit look fine or a good edit look broken. Clear the cache when you check your work.
- Confirm a backup exists before anything bulk.** Before mass edits or plugin changes, check the host or a backup plugin has a recent snapshot you can roll back to.

- Change URLs carefully.** Editing a slug changes the URL. Set a 301 redirect (Shopify auto-offers one; WordPress via a redirect plugin; Webflow in its redirects panel) so you keep the SEO equity and avoid a 404.

THE ONE-LINE SAFETY RULE

If you cannot undo it in under a minute, do not do it on the live site. Duplicate, stage, or ask the developer first. This single test prevents almost every catastrophic edit.

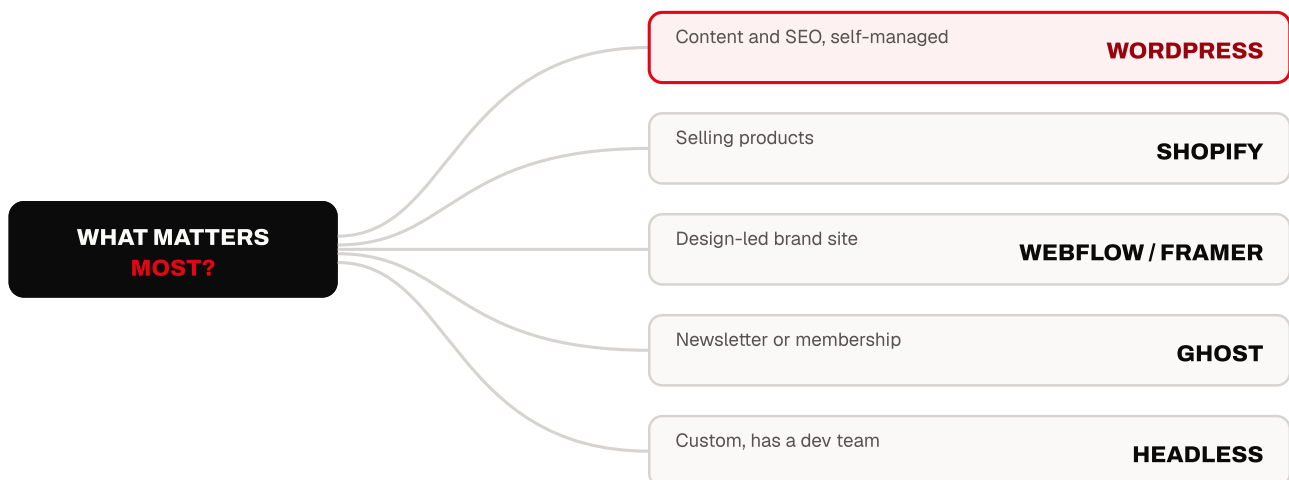
08 - WHAT TO BUILD ON

WHEN THEY ASK WHAT TO BUILD ON

IN PLAIN WORDS

Sooner or later a client will ask which platform they should build their next website on. Start from what matters most to them, whether that is selling, writing, or design, and this section gives you a simple answer for each, with WordPress as the safe pick when they are not sure.

Sooner or later a client asks what they should build their next site on. Start from the priority, not the brand name. There are really five answers, and one safe default.



Five priorities, five answers. WordPress is the safe default whenever content and SEO are the point and the client wants to manage day-to-day and hire help easily.

The line to give a client: *if you want to control content and SEO yourself and hire help easily, WordPress; if you are selling, Shopify; if design is the point and you will not need a plugin ecosystem, Webflow or Framer; if a developer built you something custom, keep that developer on retainer.* Say the last part plainly, because a headless or custom build means some of the SEO work they want will need the developer, and it is better they hear that from you at the start than discover it in an invoice.

THE TOOLS THAT COST MONEY

IN PLAIN WORDS

Almost everything you need is free and good enough, so you do not have to spend a client's money to do solid work. This section lists the few tools that charge a fee and, next to each, tells you whether the free version will do the same job.

Most of what a VA needs is free and genuinely good enough. The free tier of Yoast or Rank Math covers on-page SEO titles, meta descriptions, slugs, sitemaps, and basic schema. Recommend a paid tool only when the client needs a specific paid feature, and say which feature and why. Prices below are approximate 2026 figures and drift, so confirm on the vendor page before you quote them.

TOOL	ROUGH COST	FREE ENOUGH INSTEAD?
Yoast SEO Premium	~\$129/yr per site	yes, free tier covers most on-page work
Rank Math Pro	~\$108/yr, unlimited personal sites	yes, and far cheaper across many sites
WP Rocket (caching)	~\$59/yr, single site	yes, free caching plugins exist
Elementor Pro	~\$59/yr up to agency tiers	free Elementor works for basic builds
Shopify SEO apps	~\$5 to \$40+/mo each, recurring	often yes, native SEO fields are free
Framer editor seats	~\$10/mo content editor, ~\$20/mo editor	needed to edit at all on paid plans

THE AGENCY-PRICING TELL

If you manage many client sites, Rank Math's unlimited-sites pricing is dramatically cheaper than Yoast's per-site model. Recognizing which plugin a client already runs, and what it costs to scale, is itself a piece of advice worth giving.

ONBOARD A CLIENT, IN ORDER

IN PLAIN WORDS

This is the whole start-to-finish routine for taking on a new client, laid out in the order you actually do it: get in safely, learn the site, then settle into a steady monthly rhythm. Follow it top to bottom your first time and you will not miss a step.

Run a new engagement in this order. Follow top to bottom for your first client; jump to your level once the rhythm is second nature. Every step names the exact place to act and how you know it worked.

LEVEL 1 DAY ONE

get in, safely

1 Send one access request.

Paste the onboarding checklist from section 03, trimmed to this engagement, with your email and partner IDs.

Done when: the client has one clear message, not a trickle of asks.

2 Verify each grant as it arrives.

Log in to each platform and confirm you can actually do the thing you need, not just see it.

Done when: you have acted once in every account, under your own login.

3 Identify the CMS.

Run Wappalyzer, confirm with a second signal, and note whether a developer is involved.

Done when: you can name the platform and whether SEO asks are VA or dev tasks.

LEVEL 2 WEEK ONE

get oriented

4 Baseline the site.

Pull the starting numbers from Search Console and GA4 so you can prove your impact later.

Done when: you have a dated snapshot of impressions, clicks, and key events.

5 Confirm the editing surface.

Find where content and SEO fields live in the CMS, whether a page builder is in use, and whether there is staging.

Done when: you have made one safe test edit and seen it render.

6 Start a change log.

Open a shared doc and record every change: what, where, when. This is your record if anything breaks.

Done when: the client could see exactly what you have touched.

7 **Work the engagement loop.**

Cycle access, audit, plan, edit, measure, report on a monthly cadence.

Done when: the client gets a monthly report tied to the baseline.

8 **Keep access lean.**

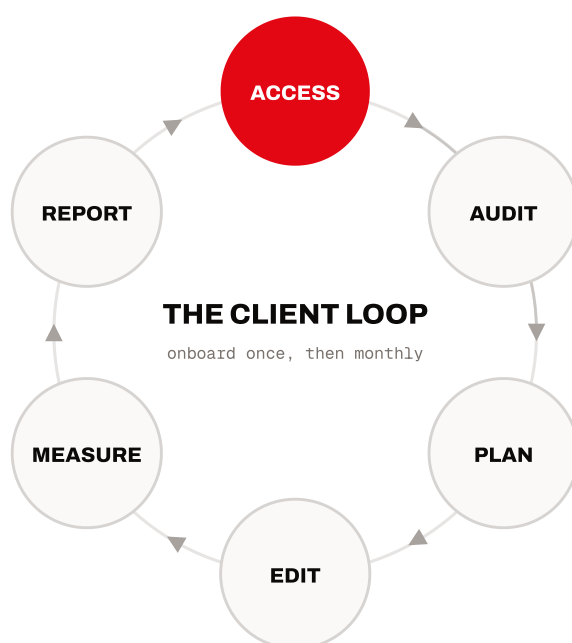
Drop any role you no longer need, and never escalate past Editor without a task that requires it.

Done when: your access still matches the work, months in.

9 **Prepare the clean exit in advance.**

Keep the change log and access list current so offboarding is a five-minute job, not a scramble.

Done when: you could hand back everything today without loose ends.



After onboarding, the engagement is a loop: audit what is there, plan the changes, edit safely, measure against the baseline, and report. Then round again.

11 - ANTI-PATTERNS

THE MISTAKES THAT COST THE MOST

IN PLAIN WORDS

This is a plain list of the errors that do the real damage, like taking a password or editing code on a live site. Read it once so you can see each one coming, because avoiding these matters far more than any clever trick.

- Taking the client's password to "go faster." It breaks their 2FA, erases the audit trail, and marks you as the person who does not know better. Decline even when they offer.
- Asking for Administrator or Owner when Editor would do. Over-asking makes a careful client nervous and puts you one slip away from breaking something you never needed to touch.
- Editing theme or template code on a live site. One wrong character in a theme file or a `.liquid` template can white-screen the whole site. That is a developer's risk to take, not yours.
- Quoting content work before checking the CMS. A headless or static build can turn a "quick edit" into a developer ticket you cannot fulfill.
- Changing a slug without a redirect, then losing the page's rankings to a 404.
- Forgetting the cache, so you tell the client a change is live when it is not, or think your edit failed when it worked.
- Keeping no change log, so when something breaks a week later nobody, including you, can say what changed.
- Leaving access in place after the engagement ends, so the client is left wondering who still has the keys.

12 - THE HEAD GAME

YOU ARE A GUEST IN THEIR HOUSE

IN PLAIN WORDS

The real job here is being someone a client feels safe handing their website to. They are nervous about letting anyone in, so being careful, clear, and never a mystery is what gets you rehired and recommended, far more than any single clever change.

The psychology of this work is not patience, it is trust. You are handed the keys to something a client has spent years and real money on, and they are nervous about it, whether or not they say so. Everything that makes you easy to trust makes you easy to rehire and to refer. That is the actual career, more than any single ranking.

What to expect

Early on, expect friction on access; it is not distrust of you specifically, it is a client who has been burned or simply does not know how. The VA who sends clear steps and asks for the lowest role is the one who gets access fastest, precisely because they look like they have done this before. Expect, too, that you will be judged less on any one clever change than on whether you are careful, communicative, and never a mystery.

Traps to avoid

1

The shortcut that erodes trust

Taking a password or full admin to save ten minutes trades a small convenience for the exact thing that makes you rehirable. Never make that trade.

2

Touching outside your scope

"While I was in there I also changed..." is how a small engagement becomes a blame conversation. Stay in the lane you were hired for; suggest the rest, do not just do it.

3

Going quiet

Silence reads as either nothing happening or something going wrong. A short, regular update, even "no changes this week, here is what is queued," is worth more than it costs.

4

Scope creep by osmosis

Access to everything is not permission to touch everything. More access than the job needs is a liability for both of you.

THE ONE HABIT

Log every change you make: what, where, and when. It is five seconds of typing that turns you from a potential mystery into the person with the clearest record in the room. When something breaks and you can show it was not your last edit, or that it was and here is the fix, you become the professional the client keeps.

13 - THE CLEAN EXIT

OFFBOARDING, AND WHAT ALL THIS BUYS YOU

IN PLAIN WORDS

When the work ends, you neatly hand back every bit of access you were given and send a note confirming it. Because you never took a password and only used your own logins, this is a quick, clean goodbye, and that is exactly what makes a client call you first next time.

When the engagement ends, the whole least-privilege model pays off at once. Because you took your own login everywhere, offboarding is a short, clean list rather than a password-rotation scramble. Walk it deliberately and confirm each item with the client.

- Remove your user from **Search Console, GA4, Tag Manager, Google Ads, and the Business Profile**, or ask the owner to.
- Delete or downgrade your **WordPress** user (reassign your content on deletion), and revoke any application passwords you created.
- Remove the **Shopify** collaborator link and the **Meta** partner link.
- Give up any **registrar or DNS** delegated access.
- Rotate any credential that was ever genuinely shared, and remove any authenticator you registered on a shared account. This is the one case where removal alone is not enough, because you knew the secret.
- Send a short offboarding note listing every access relinquished. It is professional, and it protects you from a later "they still have access" claim.

Do this well and the last thing the client remembers is how easy you were to end with, which is exactly what makes them call you first next time. Careful access, safe edits, a clear record, and a clean exit are not the boring part of the work. For a VA, they are the reputation.

For the tools named throughout, see **The Stack** companion; for the SEO, AEO, and GEO work you do once you are safely inside, see the rest of the Foundations set.

Sources

- Google Search Console: managing owners, users, and permissions support.google.com/webmasters/answer/7687615
- Google Analytics 4: access and data-restriction management support.google.com/analytics/answer/9305587
- Google Tag Manager: managing users and permissions support.google.com/tagmanager/answer/6107011
- Google Business Profile: manage owners and managers support.google.com/business/answer/3403100
- Google Ads: manager accounts (MCC) and user access support.google.com/google-ads/answer/7459700
- WordPress: roles and capabilities wordpress.org/documentation/article/roles-and-capabilities/
- Shopify: collaborator accounts help.shopify.com/en/manual/your-account/users/security/collaborator-accounts
- Bing Webmaster Tools: how to add users bing.com/webmasters/help/how-to-add-users-d5d00364
- Meta: add partners to your Business Manager facebook.com/business/help/708679622611131
- Cloudflare: manage account members and roles developers.cloudflare.com/fundamentals/manage-members/
- Webflow: legacy Editor deprecation and Content editor role webflow.com/updates/legacy-editor-deprecation
- W3Techs: CMS usage and market share, 2026 w3techs.com/technologies/overview/content_management
- Wappalyzer: identify the technology a site uses wappalyzer.com
- Companion to the SEO, AEO, and GEO Foundations guides selwynuy.dev/guides