The Case Study Optimization Playbook

A portable methodology + Claude Code skill scaffold for retrofitting your B2B case study pages for SEO and AEO performance.

Companion to: *Inside Our Claude-Powered Case Study Optimization Workflow*, Column Five Media.

Part 1 — The Methodology (Toolchain-Agnostic)

This is the 9-phase workflow we run on every case study. It works with any stack — Claude Code, a different AI assistant, or a senior in-house SEO doing it manually. The phases are the load-bearing part; the tools you swap in are interchangeable.

Phase 1 — Audit (run in parallel)

Pull everything you'll need before making any decisions.

- **Search console data (90-day window):** top queries, impressions, clicks, position, and CTR for the case study URL. If CTR is under 2% on the brand-name query at position 5–15, the slug is fighting the wrong audience — strong signal to pivot.
- **AI visibility data:** is this URL cited by any LLM platform on prompts in your category? Most case studies are absent; that's the baseline.
- **Keyword data:** monthly volume, keyword difficulty, and 12-month median for 6–10 unbranded keyword candidates in the relevant industry and work-type.
- **Live image inventory:** total page weight, image count, format breakdown, alt-text quality, current dimensions.
- **Body word count:** flag anything under 750 words for expansion during retrofit.
- **Last-modified date:** staleness signal.

Output: an audit synthesis table with performance, intent diagnostic, keyword data, image audit, word count, and ranked opportunities.

Phase 2 — Slug + focus keyword (decouple from brand)

The focus keyword must be unbranded. When someone searches the client's brand name, they want the client's own properties — your case study can't win that intent match. Target unbranded research queries where the case study *is* the right answer.

Slug pattern: `/case-studies/{unbranded-keyword}-{intent-modifier}-{brand-slug}/`

- `{unbranded-keyword}` — 2–3 words, industry and/or work-type, highest-aligned phrase from your keyword candidates
- `{intent-modifier}` — `case-study` (strongest research intent), `example`, `campaign`, `examples`
- `{brand-slug}` — at the end, for 301 link-equity recovery

SERP intent check (mandatory): before locking, pull the SERP for your leading candidate. If 8+ of the top 20 results are listicles or definitional content, reject and try the next candidate. Volume × KD ≠ winnability. SERP intent fit is the third axis.

Phase 3 — SEO meta

- **SEO title (≤60 chars):** `{Focus Keyword Capitalized} {Intent Modifier}: {Brand}`
- **Meta description (130–140 chars):** `Inside the {Brand} {focus keyword}: {2–4 specific deliverables, plain prose}.`
- **Focus keyphrase:** the unbranded category phrase from Phase 2

Phase 4 — Body content recommendations

Output as side-by-side current → proposed tables. Even when the body lives in custom fields you can't write to programmatically, generate the copy and ship it as a manifest the operator can paste.

- **H1:** focus keyword as exact bigram or trigram, plus vertical or audience qualifier; no brand name (eyebrow handles that)
- **H2 pass:** tighten existing H2s to add the focus keyword or related work-type term in 2–3 places, preserve original voice
- **Executive summary:** vertical anchor — "*{Brand}, {brand's category descriptor}, needed...*"
- **Visible date stamp:** `Project completed: {year}` — E-E-A-T signal
- **FAQ drafts (parked if no FAQ component):** 5 questions covering brand definition, work delivered, category research, audience-specific research, comparison intent

Phase 5 — Images

The single highest-leverage page-weight lever for image-heavy case studies.

- **Capture format:** WebP at quality 85 (Python imaging library, `cwebp`, Squoosh, or equivalent)
- **Filename pattern:** `{brand-slug}-{primary-keyword-slug}-{descriptor}.webp`
- **Four media-library fields per image (don't skip three of the four):**
- **Title:** human-readable version of filename
- **Caption:** empty (unless visible caption is intended)

- **Alt:** visual-first description, ≤160 chars, focus keyword natural where it fits the visual
- **Description:** 2–3 sentences expanded keyword context, ≤300 chars (this indexes on the auto-generated attachment page — a second indexable surface)
- **Alt-text rule:** describe what's visually in the image. Avoid self-referential framings like "*featured on our list of best agencies in 2026.*"
- **Featured image:** pick the highest-impact WebP for og:image and the post featured image.

For video assets (.mp4 / .mov / .webm), use HandBrake or ffmpeg with target size under 2 MB.

Phase 6 — Internal link sweep

This is the one place where most CMS APIs *will* let you write back. Use it.

For each case study you retrofit, find every cluster page, comparison page, listicle, and the case-studies hub that links to the OLD URL or that should link to the NEW one. Update the inbound link with keyword-loaded anchor text. If you changed the slug, also update inbound links from the old URL to the new canonical to skip the 301 hop (~5% link equity recovered per hop).

Always assert match counts before string-replace — a change in upstream copy can silently zero out your replace and push unchanged content thinking it was fixed.

Phase 7 — Deliverable: a single upload manifest

For everything your API can't write programmatically, produce a single markdown file the operator pastes into the CMS admin:

- Header (date, source URL, spec version)
- Before/after summary (page weight, image count, KW changes)
- Slug change + 301 verification step
- SEO meta (title, description, focus keyphrase)
- Body content updates (H1, H2 list, summary, date stamp)
- FAQ drafts (parked)
- Image upload entries — each with title, caption, alt, description, in copy-pasteable form
- Featured image selection
- Inbound link updates (programmatic, recorded for trail)
- Verification checklist

Phase 8 — Verify

After the operator confirms upload, cache-bust and curl-check:

- New slug returns 200

- Old slug returns 301 → new
- SEO title, meta description, H1 visible in head/body
- All new image filenames present in body HTML
- og:image and JSON-LD `thumbnailUrl` point to the new featured image
- Inbound link sweeps still point to canonical

Phase 9 — Calendar entry + memory

Create a calendar entry with provenance: target keyword, new URL, AEO target prompt, what shipped, image stats table, inbound link sweep table, open follow-ups, 30-day re-check schedule. If the run surfaced new patterns or constraints, write them into your team's shared playbook so the next retrofit doesn't re-learn them.

Part 2 — Claude Code Skill Scaffold (Optional Appendix)

If you're running the workflow in Claude Code, here's the skill scaffold we use. Fork and adapt for your stack.

File location

`~/.claude/skills/case-study-retrofit/SKILL.md`

Frontmatter pattern

```
---
name: case-study-retrofit
description: |
  Optimize an existing case study for SEO + AEO via REST-only workflow.
  Triggers when the operator pastes a /case-studies/ URL with optimization
  intent ("optimize this case study," "redo this case study").
  Produces an upload manifest the operator pastes into the CMS admin,
  plus pushes inbound link updates on regular posts (the one programmatic
  write path).
type: workflow
version: 1.0
related-skills: your-seo-content-engine
---
```

Required connectors (swap for your stack)

Our stack	Generic equivalent	What it does
Google Search Console (Rube/MCP)	Any search console with API	90-day query, impression, CTR, position data
Pendium MCP	Any AI visibility platform with API	Cited-sources pull for AEO intelligence
DataForSEO (Rube)	Ahrefs, Semrush, SE Ranking API	Volume, keyword difficulty, SERP intent verification
WordPress REST API	Your CMS REST endpoints	Post fetch + body write where the schema allows
Notion MCP	Airtable, Linear, Coda, your task tool	Calendar entry + provenance trail

Constraint discovery (do this first)

Before encoding the workflow as a skill, map your CMS's write-capability surface. For each output the workflow produces, can your CMS API accept it programmatically?

Output	API-writable?	If not, ship via manifest
Slug	✓ usually	—
Title (SEO + page)	✓ usually	—
Meta description	✓ usually	—
Focus keyphrase	depends on SEO plugin	maybe
Body content	depends on field type (HTML body ✓; ACF/custom fields often ✗)	likely
Image alt text	rarely via API	yes
Image media library fields	rarely via API	yes
JSON-LD schema	rarely via field; sometimes inline body	maybe

Anything in the "not via API" column goes in the manifest. Anything in the programmatic column gets pushed by the skill directly.

Phase encoding

Encode each of the 9 phases above as an explicit step in the skill, with:

- **What it does** (one paragraph)
- **Inputs** (REST endpoints, candidate keyword list, etc.)
- **Outputs** (structured data the next phase consumes)
- **Verification gates** (count assertions, intent verifications, validation checks)
- **Manifest contributions** (sections of the upload manifest this phase adds to)

Hard rules to bake into the skill

These caught us in the first few runs. Bake them in so the skill enforces them:

1. **The focus keyword must be unbranded.** No exceptions.
2. **SERP intent verification gate before locking the focus keyword.** Volume × KD is not enough.
3. **Conceptual framing check** — read the executive summary and first body paragraphs before keyword research. Brand-name confusion is real (e.g., a B2C campaign masquerading as a B2B platform case study because the URL slug is the same).
4. **Image metadata gets all four fields, not just alt.** Title, caption, alt, and description — every image, every time.
5. **Display image metadata inline in chat** for operator copy-paste. Don't rely on the manifest file as the only delivery.
6. **Assert match counts before any string-replace push** to inbound links.
7. **Cache-bust every verification curl** (`?cb=$(date +%s)`).
8. **Live-rendered HTML is not the source of truth** for body manipulation. Always fetch the canonical body via the API.

What to leave out of the skill (operator decisions, not automation)

- The final call on slug pivots (skill recommends; operator approves)
- Conceptual category disambiguation
- Trade-off calls when SERP intent is mixed
- Net-new case study creation (this is a retrofit-only workflow)

What to expect

A clean retrofit on a single case study takes roughly 30 to 60 minutes once the skill is encoded. The biggest leverage points, in order:

1. Image pipeline (page weight, often 50–90% reduction on image-heavy pages)
2. Unbranded slug + focus keyword realignment (CTR recovery on existing impressions)
3. Internal link sweep (passes equity from cluster pages to the new canonical)
4. SEO meta (title + description rewrites pull CTR on the same SERP positions)
5. Image alt + media metadata (the second indexable surface most teams leave blank)

Schema injection, FAQ structuring, and body content rewrites are usually blocked by CMS constraints on portfolio CPTs. Park what you can't ship and revisit when the dev team unblocks the field.

Credits

This playbook is the methodology behind Column Five Media's case study optimization work. Compiled from the c5-case-study-retrofit Claude skill, the UCS SEO Content Engine v3.7.3, and lessons from the first batch of retrofits on the columnfivemedia.com case study portfolio.

Questions, corrections, or your own adaptation to share? Email arumack@columnfivemedia.com.