STRETCHLEARN · WORKBOOK

Technical SEO — Workbook

This workbook turns the course into a working technical-audit kit you run against a real website. Work each section against a live site, filling the worksheets with that site's URLs, status codes, and metric scores, and using the checklists to verify nothing is silently blocking crawling or indexing. The templates are editable planners for a full crawl-issue triage, a Core Web Vitals tracker, a structured-data map, and a redirect map for migrations.

Crawlability and Indexation Fundamentals

Confirm pages can be discovered, crawled, and indexed, and that your directives say what you intend.

Worksheet: Directive Audit Worksheet

For each important page type, record exactly what your robots.txt, meta robots, and canonical are doing, then check Google agrees in URL Inspection.

Page or template (e.g. product, blog post, category, filter URL)

robots.txt status (Allowed / Disallowed)

Meta robots or X-Robots-Tag value (index/noindex, follow/nofollow)

Declared canonical URL

Google-selected canonical (from URL Inspection)

Mismatch found? (Yes/No and what)

Intended outcome (index / noindex / consolidate) and fix needed

Exercise: Indexing Report Triage Drill

Open the Search Console Pages report and turn the Not indexed reasons into a prioritized fix list, separating alarms from intended exclusions.

• Which Not indexed reasons are alarms (something is broken) versus expected (you intended the exclusion)?

• For Crawled - currently not indexed pages, are they thin, duplicate, or genuinely low value, and what is the fix?

• For Duplicate or Google-chose-different-canonical URLs, do your internal links and sitemap agree with your canonical?

• Is the Indexed trend line stable, or is a growing gap signaling a new technical exclusion?

Checklist: Sitemap Hygiene Checklist

[ ] Sitemap contains only canonical, indexable, HTTP 200 URLs
[ ] No redirects, 404s, noindexed pages, or parameter duplicates included
[ ] Each file is within 50,000 URLs and 50 MB uncompressed (sitemap index used if larger)
[ ] lastmod reflects genuine content changes, not a blanket date
[ ] Sitemap referenced in robots.txt and submitted in Search Console
[ ] Sitemap is generated dynamically from the CMS so it stays in sync

Crawling Your Site and Fixing Status Codes

Crawl the whole site, triage every non-200 URL, and consolidate duplicate and thin pages.

Worksheet: Crawl Issue Triage Worksheet

After a Screaming Frog crawl, log the highest-impact issues with their source so each fix is trackable and re-crawlable.

URL with issue

Issue type (3xx / 4xx / 5xx / duplicate title / bad canonical / orphan)

Source page linking to it

Gets organic traffic? (from Search Console API in the crawl)

Correct fix (edit link / 301 single hop / 410 / add canonical / add internal link)

Priority (High/Med/Low)

Re-crawl confirmed fixed? (Yes/No)

Exercise: Redirect Chain Untangling Drill

Find redirect chains and loops in the crawl and rewrite them to single hops without losing authority.

• Which URLs redirect through more than one hop (A to B to C), and what is the true final destination?

• Are any redirects 302 (temporary) when they should be 301 (permanent)?

• Is any redirect a loop that breaks the page entirely and must be fixed first?

• For deleted pages, should the response be a 301 to a relevant page, or a clean 410?

Checklist: Duplicate and Thin-Content Checklist

[ ] Duplicate sets identified (parameters, http/https, www, facets, print pages)
[ ] A single canonical chosen for each set (cleanest, most-linked, indexable)
[ ] True duplicates 301'd; kept variants rel=canonical to the master
[ ] Faceted-navigation URLs canonicalized or noindexed to stop crawl waste
[ ] Thin pages improved, consolidated, or removed (noindex/410)
[ ] Soft 404s return a real 404/410 instead of a 200 on an empty page
[ ] Fixes applied at the template level so the pattern does not regrow

Core Web Vitals and Page Experience

Measure vitals with the right data source and ship the high-impact LCP, INP, and CLS fixes.

Worksheet: Core Web Vitals Tracker Worksheet

Record field data (the data Google ranks on) per key template, on mobile and desktop, against the Good thresholds.

Page template (home / product / category / article)

Device (mobile / desktop)

LCP field value (Good = 2.5s or less)

INP field value (Good = 200ms or less)

CLS field value (Good = 0.1 or less)

Pass/Fail at the 75th percentile

Primary suspected cause (hero image / heavy JS / unreserved space)

Exercise: Field vs Lab Reconciliation Drill

Resolve a page that scores well in Lighthouse but fails Core Web Vitals in the field by trusting the field and debugging with the lab.

• Does the CrUX field data and the Lighthouse lab score disagree, and in which direction?

• Which real-world factors (mobile devices, third-party scripts, network, cold cache) explain the gap?

• For INP, which Total Blocking Time contributors in the lab will you cut to improve field INP?

• Given the 28-day rolling CrUX window, when will a fix you ship today actually show in the field?

Checklist: Speed and Stability Fix Checklist

[ ] LCP image compressed, resized, and served as WebP/AVIF with responsive srcset
[ ] LCP image preloaded and NOT lazy-loaded; below-fold images lazy-loaded
[ ] Render-blocking CSS/JS reduced; assets served via a CDN
[ ] Long JavaScript tasks broken up; non-critical and third-party scripts deferred
[ ] All images, videos, and iframes have width/height or CSS aspect-ratio set
[ ] Fixed space reserved for ads, embeds, and dynamic widgets
[ ] Web fonts preloaded with font-display: swap; banners overlaid, not injected above content

Schema, JavaScript, and Site Architecture

Add and validate structured data, close JavaScript rendering gaps, and flatten the architecture.

Worksheet: Structured Data Map Worksheet

Plan which Schema.org type each page template gets and confirm it validates and matches visible content.

Page template

Schema.org type (Product / Article / FAQPage / LocalBusiness / BreadcrumbList)

Required properties for that type

Matches visible on-page content? (Yes/No)

Schema Markup Validator result (valid syntax?)

Rich Results Test result (eligible? missing fields?)

Rich result currently available for this type in Google? (check Search Central)

Exercise: JavaScript Rendering Gap Drill

Compare what Googlebot renders against the raw HTML to find content and links that depend on the rendering wave.

• Using URL Inspection rendered HTML vs View Source, what important text or links appear only after rendering?

• Are internal links real anchor tags with href, or buttons/onclick that Googlebot will not follow?

• Are any JavaScript or CSS files blocked in robots.txt, preventing Google from rendering the page?

• Should this template move to SSR/SSG so critical content ships in the initial HTML?

Exercise: Log-File Crawl-Budget Drill

Analyze server logs to see where Googlebot actually spends crawl budget and reclaim it for money pages.

• Which URL patterns get the most Googlebot requests, and are they your important pages or junk (facets, redirects, errors)?

• What share of Googlebot requests hit 3xx, 4xx, or 5xx responses, and which can you eliminate?

• Which high-value pages are crawled rarely or never, and is weak internal linking or crawl depth the cause?

• Did you reverse-DNS verify the Googlebot requests are genuine and not spoofed?

Checklist: Site Architecture Checklist

[ ] Every important page is within about three clicks of the homepage
[ ] Logical hierarchy: homepage to category hub to subcategory to detail
[ ] Topic clusters in place: pillar pages link to cluster pages and back
[ ] Contextual internal links from strong pages to priority targets with descriptive anchor text

[ ] No orphan pages: every key URL has at least one internal link
[ ] Crawl Depth report and log analysis cross-checked to pull deep or uncrawled pages up

Your Action Plan

1. Set up Google Search Console and confirm property verification and sitemap submission
2. Run a full crawl in Screaming Frog with the right user-agent and JS rendering if needed
3. Triage status codes: collapse redirects to single hops, fix broken links at the source, set 410s
4. Audit robots.txt, meta robots, and canonicals, and reconcile each against the Google-selected canonical
5. Work the Search Console Indexing report from highest-impact Not indexed reason down
6. Consolidate duplicate and thin pages at the template level, fixing soft 404s
7. Measure Core Web Vitals from field data per template and ship the LCP, INP, and CLS fixes
8. Add and validate JSON-LD structured data on the page types where a rich result is available
9. Compare rendered vs raw HTML to close JavaScript indexing gaps, moving critical content to SSR/SSG
10. Analyze server logs for crawl waste and flatten architecture so authority reaches priority pages