



Fixes Kit

Ready-to-implement content for the highest-priority findings

Client	Example LLC (example.com)
Companion to	Full Audit + Internal Full Audit (16 findings)
Includes	Ready-to-use file content, schema snippets, and worked examples for the 10 findings marked WE IMPLEMENT or PATTERN & DELEGATE, plus starting-point examples for the advisory-only findings under “What’s not in this kit” — not all 16 findings as final, ready-to-ship content.
Date	observed 2026-09-30

Handling note

This kit contains actual fix content — code, schema, and config — which the client-facing audit products (Free Audit / Full Audit) deliberately never include. Share it only with whoever is executing the fixes: your own team, or the client’s developer once they’ve engaged the implementation service. It is not a substitute for, or a version of, the client-facing Full Audit.

Confirm before publishing

A few items below need a client fact-check before going live — marked in amber. Don’t publish placeholder values as if they were confirmed data.

What's not in this kit

Every finding below is **ADVISORY ONLY** in the Internal Full Audit — it needs a client decision, account access we don't have by default, or editorial judgment, so there's no ready-to-ship fix to package here. Each one still gets a starting-point example: illustrative, not final content, so the editorial/implementation team has a concrete shape to work from instead of just a name and a reason.

3.9 Site content requires JavaScript to render — AI crawlers can't see it

Why it's not in this kit: Requires the client's development team or hosting/framework access — an infrastructure change, not a content edit.

Starting point (example only — not final content): Illustrative options only — the right one depends on the site's actual framework. To reproduce the problem: fetch the homepage without running JavaScript — it currently returns 0 words and 0 headings.

Option A: Server-side rendering (SSR) if the framework supports it
 Option B: Static pre-rendering / static export at build time
 Option C: A rendering-proxy service (e.g. prerender.io) in front of the existing site

- This is a shape of the fix, not a specific implementation plan — needs the client's developer to confirm which option fits their stack.

3.12 Improve poor mobile page performance (Core Web Vitals)

Why it's not in this kit: Likely tied to the rendering architecture (3.9) and needs developer/hosting access — re-test after 3.9 rather than scoping separately.

Starting point (example only — not final content): Illustrative checklist only — the right mix depends on what a profile of the live page shows. Homepage mobile baseline: largest content appears at about 3.8 s (desktop: 0.9 s).

1. Defer or split non-critical JavaScript so the main content paints first
 2. Pre-render or server-render the above-the-fold content (see 3.9)
 3. Preload the hero image / key font; compress images
 4. Re-measure mobile and desktop after each change

- Generic starting point — needs a real profile of the live site before choosing which steps apply.

3.13 Content doesn't map cleanly to how AI systems interpret buyer intent

Why it's not in this kit: Downstream of the rendering and thin-content findings, plus new educational content that is the client's editorial call — re-test rather than scoping separately.

Starting point (example only — not final content): Example of an informational page outline once the content is crawlable:

H1: What is Service A?
 Intro (1-2 sentences): a direct definition
 H2: How it works | H2: Who it's for | H2: What to expect
 H2: Service A vs Service C (comparison table)

- Illustrative outline only — real copy needs the client's voice and facts.

4.1 Add descriptive alt text to images

Why it's not in this kit: Downstream of the JS-rendering issue above — re-test after that's fixed rather than scoping separately.

Starting point (example only — not final content): Example of descriptive alt text once images are visible to crawlers:

```
alt="Client using Service A at Example LLC"
```

- Illustrative pattern only — real alt text should describe each actual image once accessible.

6.1 Add HSTS / CSP / X-Frame-Options headers

Why it's not in this kit: Requires hosting or server configuration access we don't have by default.

Starting point (example only — not final content): Generic header example — adapt to the actual hosting/server software:

```
Content-Security-Policy: default-src 'self'  
X-Frame-Options: SAMEORIGIN
```

- Generic starting point — a real CSP needs to be tuned to the site's actual scripts/assets, not applied as-is.

6.2 Add reviews/testimonials or verification links

Why it's not in this kit: Client's own testimonial content and editorial call — not ours to invent.

Starting point (example only — not final content): Example testimonial block shape (content is entirely placeholder):

```
<blockquote>  
  "[CONFIRM WITH CLIENT – real testimonial quote]"  
  <cite>[CONFIRM WITH CLIENT – attribution, if permitted]</cite>  
</blockquote>
```

- Entirely placeholder — needs real, client-approved testimonial content and permission to name anyone.

Ready-to-implement fixes

Each fix below is ready to apply as-is, cross-referenced to its Internal Full Audit finding. Gaps in the numbering are the advisory-only findings listed above.

2.1 Add FAQPage schema with site FAQs

Add this FAQPage JSON-LD inline in the <head> of the homepage, as a separate script block next to the existing structured data. Answers are placeholders where we don't have confirmed specifics — see amber flag.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "What should I expect during my first visit to Example LLC?",
      "acceptedAnswer": {"@type": "Answer",
        "text": "[CONFIRM WITH CLIENT – describe the typical first-visit flow: intake,
consultation, which service is recommended first]"}
    },
    {
      "@type": "Question",
      "name": "How often should I use Service A or Service B for recovery?",
      "acceptedAnswer": {"@type": "Answer",
        "text": "[CONFIRM WITH CLIENT – typical recommended frequency for regular vs.
occasional clients]"}
    },
    {
      "@type": "Question",
      "name": "Do you accept walk-ins or is an appointment required?",
      "acceptedAnswer": {"@type": "Answer",
        "text": "[CONFIRM WITH CLIENT – walk-in policy]"}
    }
  ]
}
</script>
```

- All three answers are placeholders — confirm actual visit flow, recommended frequency, and walk-in policy with the client before publishing.

3.1 Add H1 to the pages missing one

Worked example for the homepage, inline in the page template — apply the same pattern (one clear H1 stating the page's topic) to the other 6 pages. Crawl evidence: the homepage returns 0 headings in its raw HTML.

```
<!-- Homepage hero section, inline in the page template -->
<h1>Example City's Premier Recovery Center – Service A, Service B & More</h1>

<!-- Pattern for the rest of the site: -->
<!-- /services -> <h1>Recovery Services & Options</h1> -->
<!-- /resources -> <h1>Learn About Recovery</h1> -->
<!-- /method -> <h1>Your Recovery Plan</h1> -->
<!-- /about -> <h1>About Example LLC</h1> -->
<!-- /contact -> <h1>Contact Example LLC</h1> -->
```

3.4 Fix meta descriptions outside the 160-character limit

Shortened homepage example (from 209 to ~140 characters), edited in the page's <head> or its CMS meta-description field — apply the same tightening pattern per page.

```
<meta name="description" content="Example City's premier recovery center – Service A, Service B, Service C & more. Trusted by local clients.">
```

3.10 Fix title tags outside the recommended length

Shortened homepage example (from 72 to ~48 characters), edited in the page's <title> tag or its CMS title field — apply the same tightening pattern per page.

```
<title>Example LLC | Service A, Service B & Recovery</title>
```

3.11 Fix canonical tags that point away from the page itself

Generic pattern — review before use. Each page's canonical tag should point to itself (non-www). Inline <link> tag in each page's <head>, plus a server-level redirect for the www host. The redirect below is a generic example — adapt to the actual hosting/server. Crawl evidence: 6 of 7 pages currently point their canonical elsewhere, and the www host answers 200 instead of redirecting.

```
<!-- On each page, canonical should be the page's own non-www URL: -->
<link rel="canonical" href="https://example.com/services">

<!-- Generic nginx example for the www -> non-www redirect (adapt to actual server): -
->
server {
    server_name www.example.com;
    return 301 https://example.com$request_uri;
}
```

- The server snippet is a generic example — confirm actual hosting/server software before applying.

5.6 Add a description and page links to /llms.txt

Replace the existing standalone file at /llms.txt (site root) with this version, which adds the missing description line and page links.

```
# Example LLC

> [CONFIRM WITH CLIENT – one-sentence description of the business]

## Main pages

- [Services](https://example.com/services): what we offer
- [Resources](https://example.com/resources): educational content
- [Method](https://example.com/method): how a recovery plan works

## About

- [About](https://example.com/about)

## Optional

- [Contact](https://example.com/contact)
```

- The description line is a placeholder — write it with the client; the page list assumes the five main pages currently in the sitemap.

5.3 Create /ai/summary.json

Publish as a standalone static file at /ai/summary.json (site root).

```
{
  "name": "Example LLC | Premium Recovery Center",
  "description": "[CONFIRM WITH CLIENT – one or two sentences describing the business and its services]",
  "url": "https://example.com",
  "lastModified": "[CONFIRM WITH CLIENT – set to actual publish date]"
}
```

- Description and lastModified are placeholders — fill in with the client before shipping.

5.4 Create /ai/service.json

Publish as a standalone static file at /ai/service.json (site root), listing services drawn from the site's existing structured data.

```
{
  "services": [
    "Service A", "Service B", "Service C", "Service D", "Service E"
  ],
  "areaServed": "Example County",
  "url": "https://example.com"
}
```

- The service list is illustrative — replace with the client's actual service names.

5.5 Create /ai/faq.json

Publish as a standalone static file at /ai/faq.json (site root) — same content as the FAQPage schema above, in this format too.

```
{
  "faqs": [
    {
      "question": "What should I expect during my first visit to Example LLC?",
      "answer": "[CONFIRM WITH CLIENT – same answer as the FAQ schema above]"
    },
    {
      "question": "How often should I use Service A or Service B for recovery?",
      "answer": "[CONFIRM WITH CLIENT – same answer as the FAQ schema above]"
    },
    {
      "question": "Do you accept walk-ins or is an appointment required?",
      "answer": "[CONFIRM WITH CLIENT – same answer as the FAQ schema above]"
    }
  ]
}
```

■ Same placeholders as the FAQ schema fix above — confirm once and reuse for both.

6.3 Add a visible About page link

Add a visible link in the main navigation and/or footer template (a CMS menu edit). If an About page doesn't exist yet, this is a one-page content addition first.

```
<a href="/about">About Us</a>  <!-- add to main nav and/or footer -->
```