



THE COMPLETE TECHNICAL PLAYBOOK

The Ultimate Custom-Coded Website SEO Checklist

How to Rank HTML, PHP & JavaScript
Websites Without Plugins

A Practical Guide for **Developers, Marketers & Business Owners**

SHAMNAS ARAKKAL | DIGITAL MARKETER

INTRODUCTION

Forget the Plugin Myth

"You can't rank a website without WordPress and an SEO plugin." It's one of the most persistent myths in digital marketing — and it simply isn't true.

SEO is not a plugin. SEO is a set of technical and content principles that search engines use to understand, trust, and rank a page. Plugins like Yoast or Rank Math don't create rankings — they automate tasks that, on a custom-coded website, you (or your developer) can implement manually, with far more precision and far less overhead.

In fact, custom-coded websites — built with pure HTML, PHP, or JavaScript frameworks — often have a structural advantage over CMS-based sites:

Superior Page Speed

No bloated themes or unnecessary plugin scripts slowing down every page load. Speed remains one of Google's most reliable ranking signals.

Total Control Over Markup

No plugin decides how your tags are generated. You write exactly what you want, exactly where you want it.

No Plugin Conflicts

Every plugin on a CMS site is a potential point of failure and another script weighing down your Core Web Vitals.

Full Flexibility

Build exactly the schema, sitemap logic, and rendering strategy your business needs — instead of working around plugin limits.

The tradeoff is that none of this happens automatically. On WordPress, a plugin gives you a form field for your meta description. On a custom-coded site, *you* write the meta tag. On WordPress, a plugin auto-generates your sitemap. On a custom site, *you or your developer* build that logic.

This guide exists to close that gap — a practical, step-by-step checklist covering every layer of technical and on-page SEO that matters for HTML, PHP, and JavaScript-based websites, with real code snippets you can hand directly to a developer or implement yourself.

BY THE END OF THIS GUIDE, YOU'LL KNOW EXACTLY

1. What needs to go in your `<head>` section, and why.
2. How to structure your `<body>` content the way Google's crawlers actually read it.
3. How to handle `robots.txt` and sitemaps without a plugin doing it for you.
4. How to add Schema markup for rich results.
5. How to make React, Vue, and Angular sites fully crawlable and indexable.

No plugins. No shortcuts. Just clean code and a clear strategy. Let's get into it.

STEP 01 / ON-PAGE SEO

Optimizing the <head> Section

Before a search engine ever reads a single word of your visible content, it first parses your page's `<head>` section — metadata about your page, rather than the page content itself. It plays a disproportionately large role in how your page is indexed, understood, and displayed in search results.

On a custom-coded site, you write this metadata directly into the HTML of every page. This is a benefit: full control over exactly what each tag says, with no plugin-imposed templates or truncation quirks. Here are the three non-negotiable tags every page needs.

1. The Title Tag

The single most important on-page SEO element. It appears as the clickable headline in search results and as the browser tab title.

```
<title>Custom Web Development Services in Riyadh | YourBrand</title>
```

What it does: Tells users and search engines what the page is about in one concise phrase. Google frequently displays it verbatim as your search snippet headline.

Character limit: 50–60 characters. Front-load your primary keyword, and include your brand name at the end.

2. The Meta Description

Doesn't directly influence rankings, but it's your advertising copy inside the search results — directly impacting click-through rate.

```
<meta name="description" content="Get a custom-coded, lightning-fast website built with clean HTML, PHP, or JavaScript — no bloated CMS, no plugins. Request your free SEO audit today.">
```

Character limit: 150–160 characters. Beyond this, Google truncates with an ellipsis. Include a natural keyword mention and a soft call-to-action.

3. The Canonical Tag

The tag most developers forget — and the one that causes the most silent SEO damage on custom-built sites.

```
<link rel="canonical" href="https://www.yourdomain.com/services/custom-web-development/">
```

What it does: Tells search engines which URL is the "master" version when duplicate or near-duplicate content exists at multiple URLs. Without it, custom-coded sites are highly prone to duplicate content issues that split ranking signals across URL variants.

BEST PRACTICE

Every single page should have a self-referencing canonical tag, even with no current duplicates. It's a safeguard, not just a fix.

QUICK CHECKLIST

- ☐ Every page has a unique `<title>` tag (50–60 characters)
- ☐ Every page has a unique meta description (150–160 characters)
- ☐ Every page has a self-referencing canonical tag
- ☐ No two pages share an identical title or description

STEP 02 / CONTENT & HIERARCHY

Structuring the `<body>` Section

Once a crawler finishes reading your `<head>`, it moves into your `<body>` — where it tries to understand what your page is about and how well-organized it is. A custom-coded site gives you full control over this hierarchy. That's a major advantage, if you use it correctly.

1. Semantic HTML & Heading Tags

Search engines use heading tags to understand the structure and topical hierarchy of your content, much like a table of contents.

The Golden Rule: Use exactly one `<h1>` per page, containing your primary target keyword. Follow it with `<h2>` for major sections and `<h3>` for subsections.

```
<h1>Custom Web Development Services in Riyadh</h1>

<h2>Why Choose Custom-Coded Websites Over WordPress</h2>
<h3>Faster Load Times</h3>
<h3>Better Security</h3>

<h2>Our Development Process</h2>
<h3>Discovery & Planning</h3>
<h3>Design & Build</h3>
```

Avoid skipping levels, and never use multiple `<h1>` tags on one page. Use semantic HTML5 elements — `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<footer>` — instead of generic `<div>` tags for everything.

2. Image Alt Tags

Search engines cannot "see" images. The `alt` attribute describes an image's content and purpose — a ranking factor for Google Image Search and a key accessibility requirement.

```

```

Describe images accurately and naturally, include your keyword only if genuinely relevant, and keep descriptions under 125 characters.

3. Internal Linking Structure

Internal links help search engines discover every page on your site and distribute ranking authority from your strongest pages to others that need a boost.

```
<p>Learn more about our <a href="/services/seo-audit/">professional SEO audit service</a> to identify technical gaps on your custom-built website.</p>
```

Use descriptive, keyword-relevant anchor text instead of "click here." Keep your architecture shallow — important pages should be reachable within 3 clicks from the homepage, with no orphan pages left unlinked.

QUICK CHECKLIST

- ☐ Exactly one `<h1>` per page, containing the primary keyword
- ☐ Logical, non-skipping heading hierarchy
- ☐ Semantic HTML5 tags used instead of generic `<div>` soup
- ☐ Every meaningful image has a descriptive alt attribute
- ☐ Internal links use descriptive, keyword-relevant anchor text
- ☐ No orphan pages — everything is reachable through navigation

STEP 03 / TECHNICAL SEO

Sitemap & Robots.txt

On a CMS, plugins generate your `robots.txt` and `sitemap.xml` automatically and update them silently as content changes. On a custom-coded site, these files need to be created manually — or, far better, generated dynamically by your developer. This is where most custom-built websites fall short.

Robots.txt — The Crawler Instruction File

A plain text file at your domain root (`yourdomain.com/robots.txt`) that tells crawlers which parts of your site they may or may not access.

```
User-agent: *
Allow: /
Disallow: /admin/
Disallow: /private/
Disallow: /cgi-bin/
Disallow: /tmp/
Disallow: /*?*sessionid=
Disallow: /search-results/

Sitemap: https://www.yourdomain.com/sitemap.xml
```

`User-agent: *` applies the rules to all crawlers. Each `Disallow:` line blocks a folder or URL pattern that would create duplicate or low-value indexed content. The `Sitemap:` line points crawlers directly to your sitemap.

CRITICAL WARNING

Never accidentally disallow your entire site with a bare `Disallow: /` while testing — this single line has taken down the rankings of more custom-built websites than almost any other technical mistake.

The Dynamic Sitemap.xml

The mistake most business owners make is treating the sitemap as a one-time, static file. It should never be static — every new page or blog post needs to be reflected automatically. Ask your developer for a **dynamically generated sitemap.xml** that queries your content

source in real time, so new pages appear and removed pages disappear without manual edits.

```
<url>
  <loc>https://www.yourdomain.com/services/custom-web-development/</loc>
  <lastmod>2026-06-20</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.8</priority>
</url>
```

Submitting to Google Search Console

- 1 Log into Google Search Console and select your verified property.
- 2 Navigate to the Sitemaps section in the left-hand menu.
- 3 Enter your sitemap path (e.g. `sitemap.xml`) and click Submit.
- 4 Monitor discovered vs. indexed URLs over the following days and fix flagged errors.

QUICK CHECKLIST

- ☐ robots.txt is live at the domain root and correctly configured
- ☐ No accidental Disallow: / blocking the entire site
- ☐ Sitemap.xml is dynamically generated, not a static file
- ☐ Sitemap includes <lastmod> dates for all URLs
- ☐ Sitemap is submitted and verified in Google Search Console

STEP 04 / STRUCTURED DATA

Schema Markup

Schema markup (structured data) is a standardized vocabulary, maintained by Schema.org, that you add to your HTML to explicitly tell search engines what your content *means* — not just what it says. This is how Google generates "rich results": star ratings, business hours, FAQ dropdowns, and knowledge panel details shown directly in search results.

On a custom-coded site, you write the structured data directly — giving you more accuracy and flexibility than a plugin's predefined schema types. The most widely supported format is **JSON-LD**, placed inside a `<script>` tag in your `<head>` or before the closing `</body>` tag.

Organization / LocalBusiness Schema

A clean, ready-to-use template — replace the placeholder values with your actual business details.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "LocalBusiness",
  "name": "YourBrand Web Development",
  "image": "https://www.yourdomain.com/images/logo.jpg",
  "url": "https://www.yourdomain.com",
  "telephone": "+966-XX-XXX-XXXX",
  "priceRange": "$$",
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "123 King Fahd Road",
    "addressLocality": "Riyadh",
    "addressRegion": "Riyadh Province",
    "postalCode": "12345",
    "addressCountry": "SA"
  },
  "geo": {
    "@type": "GeoCoordinates",
    "latitude": 24.7136,
    "longitude": 46.6753
  },
  "openingHoursSpecification": {
    "@type": "OpeningHoursSpecification",
    "dayOfWeek": ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday"],
    "opens": "09:00",
    "closes": "18:00"
  },
  "sameAs": [
```

```
"https://www.instagram.com/yourbrand",  
"https://www.linkedin.com/company/yourbrand"  
]  
}  
</script>
```

What it does: `@type` defines the entity kind — use `Organization` for a non-physical business or `LocalBusiness` for a regional one. `address` and `geo` help Google associate you with a location, crucial for local search and Maps visibility. `openingHoursSpecification` can trigger an "Open Now" status directly in results, and `sameAs` links your data to official social profiles for stronger entity recognition.

ALWAYS VALIDATE

After implementation, run your page through Google's Rich Results Test to confirm your structured data is valid and which rich result types it's eligible for. Fix any errors before considering it complete.

QUICK CHECKLIST

- ☐ JSON-LD schema added to every key page (home, contact, services)
- ☐ Schema type matches the actual business model
- ☐ All address, contact, and hours details are accurate
- ☐ Schema passes Google's Rich Results Test with zero errors

PRO TIP / MODERN FRAMEWORKS

React, Vue & Angular Sites

If your custom-coded site runs on React, Vue, Angular, or similar, there's a technical issue that catches even experienced developers off guard: **client-side rendering can make your content invisible to search engines.**

Why This Happens

A traditional HTML or PHP page sends fully-formed content the instant the server responds. A typical client-side-rendered (CSR) JavaScript app sends back a nearly empty HTML shell, and the actual content is only built and injected after JavaScript executes in the browser.

Googlebot can execute JavaScript, but this happens in a separate, delayed rendering queue — not instantly during the first crawl pass:

- Indexing can be delayed by days or weeks while pages wait in the rendering queue.
- If your JavaScript fails or times out, your page may be indexed as a blank page with no title, no content, and no ranking potential.
- Other crawlers — Bing, social preview bots, some AI-search crawlers — have far weaker JS rendering and may see nothing at all.

The Solution: Server-Side Rendering

The fix is to ensure your content is fully rendered into HTML before it reaches the crawler — known as Server-Side Rendering (SSR), or its close relative, Static Site Generation (SSG). The complete HTML is sent directly to users and crawlers, and JavaScript "hydrates" the page afterward for interactivity.

If your site uses...	Use this SSR-capable framework
React	Next.js
Vue	Nuxt.js
Angular	Angular Universal

What to Tell Your Developer

"Our site needs to be crawlable and indexable by search engines from the very first server response. Please implement this using Server-Side Rendering or Static Site Generation – for example, migrating to Next.js, Nuxt.js, or Angular Universal – rather than relying on pure client-side rendering."

How to Verify It's Working

- 1 Right-click → "View Page Source" and confirm real content appears in the raw HTML, not an empty container div.
- 2 Use Search Console's URL Inspection tool and click "View Crawled Page" to see exactly what Googlebot rendered.
- 3 Run the page through Google's Rich Results Test, which also shows the rendered HTML Google sees.

QUICK CHECKLIST

- ☐ Confirmed whether the site uses client-side or server-side rendering
- ☐ Migration to Next.js / Nuxt.js / Angular Universal planned or implemented
- ☐ "View Page Source" shows real content, not an empty shell
- ☐ URL Inspection confirms full content is being indexed



FINAL THOUGHTS

Code It Right. Rank It Right.

Custom-coded websites aren't a disadvantage in the world of SEO — they're an opportunity. Every tag, every file, and every line of markup is something you control directly, with no plugin layer standing between your strategy and your code.

The businesses that win in search aren't the ones with the most plugins installed — they're the ones who understand why each technical element matters and implement it with precision. Treat this checklist as a living document: revisit it every time you launch a new page, migrate a framework, or notice a dip in visibility.

BROUGHT TO YOU BY

Shamnas Arakkal

Digital Marketer

Instagram: @bydigi_shamnas

Website: www.shamnasarakkal.com

Let's scale your business visibility. For custom digital marketing strategies and professional SEO audits, feel free to connect!