

Every fix your site needs, in one place.

<https://www.allbirds.com> · Platform: Shopify

62 / 100 SEO score

NEEDS WORK

COPY · PASTE · SHIP

Ready-to-use replacements.

Paste these finished strings into your site exactly as shown. Each fix lists the platform-specific location beside it. Start here before reading the detailed guide below.

Schema Markup (JSON-LD)

In Shopify Admin, go to Online Store → Themes → Actions → Edit Code, then open `layout/theme.liquid` and paste this block immediately before the closing `</head>` tag.

Adding Organization, WebSite (with Sitelinks SearchBox), and CollectionPage structured data helps search engines understand the brand and current featured collection, improving eligibility for rich results.

CURRENT VALUE

(missing – no current value on page)

PASTE THIS INSTEAD

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "Organization",
      "name": "Allbirds",
      "url": "https://www.allbirds.com",
      "logo": {
        "@type": "ImageObject",
        "url": "https://www.allbirds.com/[[PATH_TO_LOGO_IMAGE]]"
      },
      "sameAs": [
```

```

    "https://www.instagram.com/allbirds",
    "https://www.facebook.com/allbirds",
    "https://twitter.com/allbirds"
  ],
  "contactPoint": {
    "@type": "ContactPoint",
    "telephone": "[[YOUR BUSINESS PHONE]]",
    "contactType": "customer service"
  }
},
{
  "@type": "WebSite",
  "name": "Allbirds",
  "url": "https://www.allbirds.com",
  "potentialAction": {
    "@type": "SearchAction",
    "target": {
      "@type": "EntryPoint",
      "urlTemplate": "https://www.allbirds.com/search?q={search_term_string}"
    },
    "query-input": "required name=search_term_string"
  }
},
{
  "@type": "CollectionPage",
  "name": "The New Canvas Cruiser Collection",
  "description": "Shop the new Canvas Cruiser Collection from Allbirds. Explore sustainable, comfortable shoes and apparel in eight exclusive Pantone-curated shades for men and women.",
  "url": "https://www.allbirds.com",
  "publisher": {
    "@type": "Organization",
    "name": "Allbirds"
  }
}
]
}
</script>

```

SEO Remediation Guide

<https://www.allbirds.com>

Score: 62/100 — Needs work

1. Executive Summary

The scan returned 62/100 (Needs work) across 21 checks: 2 failed, 6 warned, 13 passed. The passing baseline is solid — titles, meta descriptions, canonical tags, Open Graph, hreflang, sitemap, and HTTPS are all clean. What's dragging the score down is almost entirely schema and image hygiene. The two failures (missing alt text on 75 of 108 images, and zero structured data on the page) are both high-impact and fixable without a developer for the alt text issue, and with a single app install or theme edit for schema.

Fix priority:

1. **[FAIL] Schema Markup** — No structured data at all. For an e-commerce homepage, this blocks Product rich results and Google Shopping eligibility simultaneously.
2. **[FAIL] Image Alt Text** — 75 of 108 images untagged. Accessibility liability and a crawl-efficiency drain.
3. **[WARNING] Merchant Center** — Directly downstream of the schema fix; resolves automatically once Product schema is added.
4. **[WARNING] Breadcrumb Schema** — Quick JSON-LD addition; improves SERP appearance for category/product pages.
5. **[WARNING] FAQ Schema** — Add only if FAQ content exists on the page; don't force it.
6. **[WARNING] H1 Heading** — Two H1s found; remove the duplicate.
7. **[WARNING] Heading Hierarchy** — H1 jumps to H3; insert missing H2 levels.
8. **[WARNING] lms.txt Discovery Link** — One `<link>` tag in `theme.liquid`; five-minute fix.

Fixing all 2 fails and all 6 warnings moves all 8 checks into the pass column. Since every check counts equally across 21 total checks, that arithmetic yields a post-remediation score of **100/100** — assuming no regressions in currently passing checks.

2. Platform-Specific Context (Shopify)

Several of these fixes interact directly with how Shopify structures themes, and getting the approach wrong means your changes get overwritten or don't render at all.

Theme files vs. the editor. Schema markup and `<link>` tags belong in `theme.liquid` (or a dedicated snippet) inside the `<head>` block. Changes made through the Shopify page editor (rich text fields) are stripped of raw HTML by design — you cannot inject a `<script type="application/ld+json">` through the content editor. You need **Online Store** → **Themes** → **Edit code**.

Shopify's built-in schema is often incomplete. Many Shopify themes ship with a `product.json` or inline `application/ld+json` block in `product.liquid`, but the homepage (`index.liquid` or `index.json`) typically gets nothing. The scan found *no* structured data on the homepage — this is the norm for Shopify homepages, not a theme bug.

Alt text has two locations in Shopify. Images uploaded to the Files section and images attached to products/pages have separate alt text fields. The scan's first flagged image (`//www.allbirds.com/cdn/shop/files/26Q2_CanvasCruiser_Site_Na...`) is a Files-section image — those require manual editing in **Settings** → **Files** or via the Storefront API. Product images get their alt text in the product editor. Section/block images in a theme customizer get it in the customizer's image panel.

Apps vs. theme edits for schema. The hint in the scan correctly calls out *JSON-LD for SEO* (by Ilana Davis) and *Smart SEO* as legitimate options. If you use an app, disable or remove any schema block you manually add to `theme.liquid` for the same type — duplicate schema isn't penalized outright, but it creates maintenance confusion and occasionally surfaces conflicting data to Google.

Dawn and OS 2.0 themes use `sections/` JSON. If your theme is Online Store 2.0, section-level schema should go into a `{% schema %}` block or a Liquid snippet rendered in `theme.liquid`, not hardcoded into `index.liquid` (which may not exist as an editable file). The code examples below work for both architectures.

3. Per-Issue Remediation

FAIL: Schema Markup (`schema`)

Why this matters first. The scan found *no* structured data of any kind — no JSON-LD, no Microdata. For Allbirds, which sells physical products, this single gap explains two failures simultaneously (this check and the Merchant Center warning below). Google uses Product schema to populate Shopping ads and organic rich results (price, availability, review stars). Without it, you're invisible to that entire surface.

What to add for the homepage. A homepage typically carries `Organization` schema (your brand identity) plus, optionally, a `WebSite` schema block that enables the Sitelinks Search Box. Product schema lives on product pages — the scanner hit the homepage, so product-page schema is a separate task (covered under the Merchant Center warning).

Add the following to `theme.liquid` inside the `<head>` tag. Edit the placeholder values:

```
<!-- Organization schema — add inside <head> in theme.liquid -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Organization",
  "name": "Allbirds",
  "url": "https://www.allbirds.com",
  "logo": "https://www.allbirds.com/cdn/shop/files/allbirds-logo.png",
  "sameAs": [
    "https://www.instagram.com/allbirds",
    "https://www.facebook.com/allbirds",
    "https://twitter.com/allbirds",
    "https://www.linkedin.com/company/allbirds"
  ],
  "contactPoint": {
    "@type": "ContactPoint",
    "contactType": "customer service",
    "url": "https://www.allbirds.com/pages/contact-us"
  }
}
</script>
```

```
<!-- WebSite schema – enables Sitelinks Search Box eligibility -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "WebSite",
  "name": "Allbirds",
  "url": "https://www.allbirds.com",
  "potentialAction": {
    "@type": "SearchAction",
    "target": {
      "@type": "EntryPoint",
      "urlTemplate": "https://www.allbirds.com/search?q={search_term_string}"
    },
    "query-input": "required name=search_term_string"
  }
}
</script>
```

For a downstream LLM adapting this to another stack: `Organization` and `WebSite` schema belong in the `<head>` of the root document. On Next.js use a `<Script id="schema-org" type="application/ld+json">` component in `_document.js` or a layout file. On WordPress, inject via `wp_head` action or a plugin like Yoast. The JSON structure itself is identical across platforms.

Verify. Paste `https://www.allbirds.com` into [Google's Rich Results Test](#) after deploying. You should see "Organization" and "WebSite" detected. Also run the [Schema Markup Validator](#) for a stricter structural check. Allow up to 48 hours for Googlebot to recrawl and reflect the change in Search Console's "Enhancements" tab.

FAIL: Image Alt Text (`img_alt`)

Current state: 75 of 108 images are missing alt text. The first flagged image is:

```
//www.allbirds.com/cdn/shop/files/26Q2_CanvasCruiser_Site_Na...
```

That path (`/cdn/shop/files/`) confirms it was uploaded via **Settings** → **Files** (or a theme section's image picker), not as a product image — which tells you the bulk of the problem is likely in customizer-managed section images, not product catalog entries.

Three places to fix, three different workflows:

① Files-section images (path contains `/cdn/shop/files/`)

These have no native bulk-edit interface in the Shopify admin. Options:

– In the Theme Customizer, click each section that uses an image → select the image block → fill in the "Image alt text" field. This is the most reliable method and doesn't require code.

– Alternatively, use the [Storefront API](#) or Admin REST API (`PUT /admin/api/2024-01/files/{id}.json` with `"alt"` field) for bulk updates if you have developer access.

② Product images

In the Shopify admin: **Products** → **[Product name]** → **click each image thumbnail** → **fill in "Add alt text"**. The format should be descriptive and natural: `"Men's Tree Runner shoe in natural white – side view"`, not `"shoe1"` or keyword-stuffed strings.

③ Page/blog content images

In the rich text editor for any page or blog post: click the image → the image settings panel includes an alt text field.

Writing good alt text — the practical rule. Describe what the image *shows*, not what it *is*. For product images: include the product name, colorway, and angle if relevant. For lifestyle/campaign images: describe the scene briefly. Decorative dividers, spacers, and background textures should get `alt=""` explicitly — that signals to screen readers that the image is presentational, and it's the correct implementation, not a shortcut.

```
<!-- ✅ Correct – descriptive -->


<!-- ✅ Correct – decorative image, intentionally empty -->


<!-- ❌ Wrong – missing entirely (what the scan is flagging) -->


<!-- ❌ Wrong – keyword stuffed -->

```

Prioritize by traffic impact. Start with homepage hero images and any image that appears in a featured product section — those are most likely to appear in Google Image Search. Product images are second priority. Decorative assets last (mark them `alt=""`).

Verify. After saving, run this quick browser check: open DevTools → Console → paste:

```
document.querySelectorAll('img:not([alt])').length
```

A return of `0` means every image has an alt attribute (including empty ones). To find images with *empty* alt text that should be descriptive:

```
[...document.querySelectorAll('img[alt=""]')].map(i => i.src)
```

For a full re-scan, re-run the SEO audit tool that produced this report.

WARNING: Merchant Center (`merchant_center`)

This warning is directly downstream of the Schema Markup failure — the scanner found no Product schema, which is the minimum requirement for Google Merchant Center organic listings and Shopping ads.

The homepage doesn't need Product schema. Homepages don't represent a single purchasable product, so don't add Product schema to `theme.liquid` globally. Product schema belongs on individual product pages (`/products/[handle]`).

Where to add it on Shopify. Open `sections/main-product.liquid` (or `templates/product.liquid` in older themes) and look for an existing `application/ld+json` block. Many themes include one but omit required fields. If there's already a block, augment it. If there's none, add the following inside the product template's `<head>` contribution, or render it as a snippet:

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": {{ product.title | json }},
  "description": {{ product.description | strip_html | strip_newlines | json }},
  "image": [
    {% for image in product.images limit: 5 %}
      "{{ image.src | img_url: 'master' }}" {% unless forloop.last %},{% endunless %}
    {% endfor %}
  ],
  "sku": {{ product.selected_or_first_available_variant.sku | json }},
  "brand": {
    "@type": "Brand",
    "name": "Allbirds"
  },
  "offers": {
    "@type": "Offer",
    "url": "{{ shop.url }}{{ product.url }}",
    "priceCurrency": {{ cart.currency.iso_code | json }},
    "price": {{ product.selected_or_first_available_variant.price | divided_by:
100.0 | json }},
    "availability": "{% if product.available %}https://schema.org/InStock{% else %}https://schema.org/OutOfStock{% endif %}",
    "itemCondition": "https://schema.org/NewCondition",
    "seller": {
      "@type": "Organization",
      "name": "Allbirds"
    }
  }
}
</script>
```

Note on price: Shopify stores prices in cents as integers. The `| divided_by: 100.0` filter converts `9500` to `95.0`. Verify this against your theme's existing price output logic — some themes use a custom `money` filter.

If using an app instead. Install [JSON-LD for SEO](#) — it handles Product, BreadcrumbList, Organization, and more with correct Liquid variable bindings and stays updated as Google's requirements change. If you go

this route, do *not* also add the manual block above — you'll get duplicate schema.

Google Merchant Center feed vs. schema. Schema markup on product pages enables *organic* Shopping results (free listings). Paid Shopping ads require a separate Merchant Center product feed. Both are valuable; schema is the free, lower-effort starting point.

Verify. Use the [Rich Results Test](#) on any product URL (e.g., `https://www.allbirds.com/products/mens-tree-runner`). It should show "Product" as a detected type with price and availability populated.

WARNING: Breadcrumb Schema (`breadcrumb_schema`)

Breadcrumbs in SERP snippets replace the raw URL with a readable path like `Allbirds > Men's > Shoes > Tree Runner`. For an e-commerce site with a multi-level category hierarchy, this is a meaningful click-through-rate lever.

Add to your product and collection templates. Breadcrumb schema should reflect the actual navigation path to the current page — don't hardcode it on the homepage where there's no hierarchy to express.

For `sections/main-product.liquid` or a `snippets/breadcrumbs.liquid` file:

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "BreadcrumbList",
  "itemListElement": [
    {
      "@type": "ListItem",
      "position": 1,
      "name": "Home",
      "item": "{ shop.url }"
    },
    {
      "@type": "ListItem",
      "position": 2,
      "name": "{ collection.title | json }",
      "item": "{ shop.url } { collection.url }"
    },
    {
      "@type": "ListItem",
      "position": 3,
      "name": "{ product.title | json }",
      "item": "{ shop.url } { product.url }"
    }
  ]
}
</script>
```

For collection pages, use a two-level version (Home → Collection). Wrap the middle `ListItem` in a Liquid conditional so it only renders when `collection` is defined:

```
{% if collection %}
  {
    "@type": "ListItem",
    "position": 2,
    "name": {{ collection.title | json }},
    "item": "{{ shop.url }}{{ collection.url }}"
  },
{% endif %}
```

For a downstream LLM on another stack: BreadcrumbList `item` values must be absolute URLs. The `position` values must be sequential integers starting at 1. Each level in the path gets its own `ListItem`. The schema should mirror the visible breadcrumb navigation — if there's no visual breadcrumb, add one before adding the schema.

Verify. Rich Results Test on a product URL; look for "BreadcrumbList" in the detected structured data panel.

WARNING: FAQ Schema (`faq_schema`)

Does Allbirds' homepage have FAQ content? The scan warned that no FAQPage schema was found. Before acting: scroll the current homepage and check whether there's an accordion, an expandable Q&A section, or any content formatted as questions and answers. If there isn't, skip this fix entirely — fabricating FAQ structure to capture rich results is against Google's spam policies and risks a manual action.

If FAQ content *does* exist on the page (or on a dedicated FAQ/help page):

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "Are Allbirds shoes machine washable?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Yes. Most Allbirds styles can be machine washed on a gentle cold cycle. Remove the insoles and laces before washing."
      }
    },
    {
      "@type": "Question",
      "name": "What materials are Allbirds shoes made from?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Allbirds uses natural materials including ZQ-certified merino wool, FSC-certified eucalyptus tree fiber, and sugarcane-based SweetFoam® soles."
      }
    }
  ]
}
```

```
}  
]  
}  
</script>
```

Replace the example Q&A pairs with the actual questions and answers from your page. The `name` value in each `Question` must match the visible question text on the page — Google cross-references schema content against the rendered HTML.

Add to the correct template. If FAQ content lives on `/pages/faq`, add this block to that page's template or as a metafield-driven snippet, not to `theme.liquid` globally.

Verify. [Rich Results Test](#) → should show FAQPage with expanded Q&A pairs.

WARNING: H1 Heading (`h1`)

Current state: The scanner found 2 H1 headings on the page. Best practice — and Google's documented guidance — is exactly one H1 per page.

Where the duplicate likely comes from on Shopify. Shopify themes typically render the page/shop title as an H1 automatically (often hidden visually with CSS but present in the DOM). A common mistake is then adding a second H1 in a homepage section via the rich text editor or a theme block with a hardcoded heading level.

How to find both H1s. In Chrome DevTools:

```
document.querySelectorAll('h1')
```

This returns all H1 elements. Note their text content to identify which is the "real" one and which is the duplicate.

Fixing it. If the duplicate is in a theme section or customizer block: open the Theme Customizer → find the section containing the duplicate heading → change the heading element from H1 to H2 (or whatever level fits the hierarchy). If the duplicate is hardcoded in a Liquid template, change `<h1>` to `<h2>` in the file.

Do not delete the heading content — just change its level. The goal is one H1 that identifies the page's primary topic (for a homepage, something like the brand name or primary value proposition), with all other section headings at H2 or below.

WARNING: Heading Hierarchy (`heading_hierarchy`)

Current state: The page's heading structure jumps from H1 directly to H3, skipping H2 entirely.

This matters because headings create the document outline that search engines use to understand content structure. A gap from H1 → H3 signals either missing content sections or a theme/template issue where H2 was never added.

Fixing skipped levels. The resolution depends on what's creating the H3s:

Source of H3	Fix
Rich text editor content block	Select the H3 text → change to Heading 2 in the dropdown
Theme section with hardcoded <code><h3></code>	Edit the Liquid file, change <code><h3></code> to <code><h2></code>
Theme block setting (e.g., "heading size")	Check if the block has a heading level option in the customizer
CSS class doing visual sizing (the tag is actually H2 already)	Inspect in DevTools — if the tag is already H2, this warning may have been triggered by a different section

The correct hierarchy for a Shopify homepage typically looks like:

```
H1: Brand name / page title (one instance)
  H2: Section heading (e.g., "Featured Products")
    H3: Subsection or product name within that section
  H2: Another section (e.g., "Our Materials")
  H2: "Frequently Asked Questions" (if applicable)
```

You should not need to add content to fix this — only adjust the heading *level* of existing content. Verify with:

```
[...document.querySelectorAll('h1,h2,h3,h4,h5,h6')].map(h => h.tagName + ': ' +
h.textContent.trim().substring(0,60))
```

That returns the full heading outline in DOM order, making the skip immediately visible.

WARNING: llms.txt Discovery Link (`llms_link`)

Current state: `llms.txt` exists at `https://www.allbirds.com/llms.txt` (a passing check), but the page's `<head>` doesn't include a `<link>` tag pointing to it. AI crawlers and tools that parse `<head>` rather than probing root paths won't discover it.

This is a five-minute fix. Add one line to the `<head>` section of `theme.liquid`:

```
<link rel="alternate" type="text/plain" href="{{ shop.url }}/llms.txt"
title="LLMs.txt">
```

Some implementations use a custom `rel` value instead:

```
<link rel="llms-txt" href="{{ shop.url }}/llms.txt">
```

Either form satisfies the convention. The `rel="alternate"` version has broader parser support since `alternate` is a registered link relation; `rel="llms-txt"` is the emerging standard for this specific use case. Including both is harmless:

```
<link rel="alternate" type="text/plain" href="{{ shop.url }}/llms.txt"
title="LLMs.txt">
<link rel="llms-txt" href="{{ shop.url }}/llms.txt">
```

Where in `theme.liquid`: Place it near your other `<link rel="canonical">` and `<link rel="alternate" hreflang>` tags for maintainability — all discovery links together.

Verify. View source on the homepage (`Ctrl+U` / `Cmd+U`) and search for `llms`. You should see the tag in the `<head>` block.

Note: This check is an AI-readiness convention, not a Google ranking factor. It ensures LLM-powered tools and AI search agents that parse `<head>` metadata can locate your `llms.txt` without a direct path probe. Worth doing given Allbirds already has the file; the effort is trivial.

4. Technical Reference

Full 21-Check Audit Table

Check	ID	Category	Status
Page Title	<code>title</code>	meta	✔ Pass
Meta Description	<code>meta_description</code>	meta	✔ Pass
Canonical Tag	<code>canonical</code>	meta	✔ Pass
Open Graph Tags	<code>og_tags</code>	social	✔ Pass
Twitter Card	<code>twitter_card</code>	social	✔ Pass
robots.txt	<code>robots_txt</code>	crawlability	✔ Pass
XML Sitemap	<code>sitemap</code>	crawlability	✔ Pass
Robots Meta Tag	<code>robots_meta</code>	crawlability	✔ Pass
Viewport Meta	<code>viewport</code>	technical	✔ Pass
HTML Lang Attribute	<code>lang</code>	technical	✔ Pass
HTTPS	<code>https</code>	technical	✔ Pass
Hreflang Tags	<code>hreflang</code>	international	✔ Pass
llms.txt	<code>llms_txt</code>	ai_readiness	✔ Pass
H1 Heading	<code>h1</code>	content	⚠ Warning

Check	ID	Category	Status
Heading Hierarchy	<code>heading_hierarchy</code>	content	⚠ Warning
FAQ Schema	<code>faq_schema</code>	schema	⚠ Warning
Breadcrumb Schema	<code>breadcrumb_schema</code>	schema	⚠ Warning
llms.txt Discovery Link	<code>llms_link</code>	ai_readiness	⚠ Warning
Merchant Center	<code>merchant_center</code>	ecommerce	⚠ Warning
Image Alt Text	<code>img_alt</code>	content	❌ Fail
Schema Markup	<code>schema</code>	schema	❌ Fail

Score arithmetic: 13 passing checks out of 21 = 62/100. Fixing all 8 failing/warning checks moves the score to 21/21 = **100/100**.

Authoritative Documentation

Schema.org types used in this guide

- Organization: <https://schema.org/Organization>
- WebSite + SearchAction: <https://schema.org/WebSite>
- Product: <https://schema.org/Product>
- BreadcrumbList: <https://schema.org/BreadcrumbList>
- FAQPage: <https://schema.org/FAQPage>

Google Search Central

- Understand how structured data works:
<https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data>
- Product rich results requirements: <https://developers.google.com/search/docs/appearance/structured-data/product>
- Breadcrumb structured data: <https://developers.google.com/search/docs/appearance/structured-data/breadcrumb>
- FAQ structured data: <https://developers.google.com/search/docs/appearance/structured-data/faqpage>
- Image SEO best practices (includes alt text):
<https://developers.google.com/search/docs/appearance/google-images>
- Sitelinks Searchbox: <https://developers.google.com/search/docs/appearance/structured-data/sitelinks-searchbox>

Validation tools

- Rich Results Test: <https://search.google.com/test/rich-results>
- Schema Markup Validator: <https://validator.schema.org/>
- Google Search Console (Enhancements tab): <https://search.google.com/search-console>

Shopify-specific

- Shopify theme code editor: <https://help.shopify.com/en/manual/online-store/themes/theme-structure/extend/edit-theme-code>

- Adding alt text to product images: <https://help.shopify.com/en/manual/online-store/images/theme-images#add-alt-text>
- Shopify Storefront API (Files): <https://shopify.dev/docs/api/storefront>

MDN

- `` alt attribute: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img#alt>
 - `<link>` element: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link>
 - Heading elements (H1–H6): https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements
-

Generated by **Abby SEO** — abbyseo.com — Good girl, good metrics.

This guide contains platform-specific remediation instructions tailored to your Shopify website.