

Example. This store does not exist. The findings are real failure modes from the Ottom stall library, arranged on a shop we made up. No number in this document is a measurement of anything.

Example Store

example-store.test · Shopify · scanned 29 August 2026

58 of 100

STALLED at select_variant

An agent walked this store the way an AI shopping assistant would: find a product, read its details, compare it, buy it. This is how far it got.

Discover		78	PASS
Evaluate		71	PASS
Compare		54	SLOW
Buy		20	STALLED

How far the agent got

The buy attempt, step by step. It stops at the payment step. No card details are entered and no order is placed.

#	Stage	Action	Result
1	Buy	Opened the product chosen for this scan	ok
2	Buy	Looked for a size control to set	stopped

What it found

Worst first. Each one is what an agent hit, and what to change.

HIGH hover_only_variant_picker Buy

Sizes can only be chosen by hovering

The size control on the product page reveals its options on mouse hover and exposes no form field an agent can set. A person moves a mouse over it without thinking. An agent has no mouse, finds no selectable value, and cannot add the item to a basket.

The fix. Render the size options as a real form control, a select or a group of radio inputs, so the value can be set without a pointer. Keep the hover behaviour if you want it; the point is that it stops being the only way.

HIGH popup_blocks_flow Buy

A newsletter popup covers the basket

A modal appears eight seconds after the first page load and overlays the page until it is dismissed. The agent reached the basket underneath it and could not act on anything the modal covered.

The fix. Delay the popup until a scroll or an exit intent rather than a timer, or suppress it on the basket and checkout paths. Either keeps it for shoppers and stops it ending a purchase.

HIGH missing_availability Evaluate

Stock is shown as a colour, not as text

The product page signals availability with a green dot and no words. There is no availability field in the structured data either, so nothing on the page states in a readable form whether the item can be bought.

The fix. State availability in text next to the dot, and set `availability` in the product markup. The dot can stay: it is the only signal that is the problem.

HIGH missing_shipping_terms Compare

Delivery cost only appears at checkout

Nothing on the product page or in the policies states a delivery cost or a threshold, so an agent comparing this store against another has no number to compare on and tends to leave it out rather than guess.

The fix. Publish delivery costs and any free-delivery threshold on a page of their own, in text, and link it from the product page.

LOW no_llms_txt Discover

No llms.txt

The store publishes no llms.txt, so an assistant describing it works from whichever page it crawled first rather than from a description the merchant wrote.

The fix. Publish an llms.txt at the root of the site naming what you sell and your key pages.

HIGH policy_in_image Evaluate

The returns policy is a picture

The returns page is a single image of a designed document. A person reads it without noticing. An agent asked for something with free returns gets no text to read, so it cannot confirm the policy and will not assert it to the shopper.

The fix. Put the policy on the page as text. Keep the designed version as a download beside it if it matters to you, but the terms themselves have to be readable without the picture.

MEDIUM missing_gtin Evaluate

Products carry no GTIN

No product exposes a GTIN, EAN or MPN. Nothing here fails on its own, but it is what an assistant matches your item against the same item elsewhere with, so this shop is harder to recognise as carrying a product somebody already named.

The fix. Publish the barcode you already hold in the product data as a gtin field in your product markup and in your feed.

MEDIUM no_category_structure Compare

Categories exist in the menu and nowhere else

The navigation is built in the theme rather than from a category structure the pages declare, so an agent narrowing to "40 litre duffels" has nothing to narrow with and has to read every product page to find out what is in the range.

The fix. Expose the categories as real collection pages with breadcrumb markup, so the structure you already sell by is visible outside the menu.

Run example scripted

Forward this to your developer. Our agents identify themselves at ottom.io/agent.