WooCommerce store audit: wyposazenie24.pl, September 2026

Prepared: 24 September 2026

Store	wyposazenie24.pl, selling in Poland
Range	catering equipment (4,574 of 7,330 products), workshop and garage, home and garden, power tools, scales, industrial chemicals and more
Sales model	dropshipping: products, prices and stock levels imported from the wholesaler's feed
Platform	WordPress 6.8.1, WooCommerce 9.9.3, PHP 8.3
Theme	Porto 7.5.3 with the Porto Child 1.1 child theme
Plugins	35, of which 32 active
Server	shared hosting, LiteSpeed server
Trading period	September 2024 to 29 July 2026; domain expired on 6 September 2026
Audit	23 September 2026, Basic variant

Sample report of the Basic variant. The store belonged to the auditor. For the audit, it was restored on 23 September 2026 from a copy of its files and database on the same server, at a separate password-protected address, with customer data anonymised (hereafter: the audit copy). The report describes the store as it was when it was shut down; the performance measurements come from the audit copy.

Every departure from a regular audit is described in the last part of the report. Each problem comes with a recommended fix: a pointer to what to change, without step-by-step instructions.

Priorities throughout the report: `critical` costs orders now or opens the store to an attack; `high` holds back sales or raises the cost of upkeep; `medium` is housekeeping and hygiene; `low` has no effect on orders. The priorities are given as if the store were trading. The store is shut down, so today a real risk can come only from the service keys stored in its files and database (point 3.4); the rest of the order applies to a store that would go back to selling.

1. Summary

The checkout worked without the usual obstacles, but from June 2025 the store was selling a catalogue it did not know: the stock import from the wholesaler stopped on 15 June 2025 and was not resumed for as long as the store was running. Phones got a slower store than computers, updates had no way to arrive, and firewall alerts reached nobody.

Status	Area	Finding
✗	Catalogue	the wholesaler does not have 1,095 of the 4,688 products marked as in stock (23%); 580 that it does have are hidden in the store as sold out (point 4.1)
✗	Phones	no page is cached: the server answers in 0.62 s (median) against 0.06 s for a page from the cache; every new visitor loads the first page twice (2.2, 2.3)
✗	Updates	92 published vulnerabilities in 19 components, 45 exploitable without logging in, including a critical one in the WordPress core; automatic fixes blocked by the code repository in the site directory (3.1, 3.2)
✗	Files in the site directory	the `.git` repository and 5,998 file listings left by FTP copying; the server serves hidden files (3.5)
✗	Service keys	stored in the code and in the database; the OpenAI key was active on the day of the audit, and the owner revoked it on 24 September 2026 (3.4)
✗	Firewall and logins	42 alerts with no notification address; no account has two-factor authentication (3.3)
✗	Bank transfer	WooCommerce's sample account details on the confirmation page and in the e-mail to the customer (4.2)
✓	Checkout	guest checkout, one step, the shipping cost visible in the cart; a test order reached the confirmation page in 1.6 s (4.3)
✓	Taxes and prices	VAT 23% by shipping address; the lowest price from the last 30 days recorded for all 7,330 products (4.5)
✓	Accounts	registration closed, three accounts, no application passwords (3.6)

Why sales did not work as well as they could. The checkout itself did not put customers off; sales were held back by things before it. A customer could order one of the 1,095 products the wholesaler did not have and get a cancellation instead of a parcel; they did not see 580 products they could have bought; on a phone, every category and product page waited for the server. Abandoned carts cannot be counted in the Basic variant.

What to do first. The full list with priorities and who does the work: chapter 7, What to fix, in what order.

1. Update WordPress to 6.8.10 and remove the code repository and the file listings left by FTP copying from the site directory.
2. Resume the stock import from the wholesaler, with a notification of every failed run.
3. Restore caching for phones and switch off Guest Mode in the LiteSpeed Cache plugin.
4. Remove the OpenAI key from the plugin's code and revoke the other service keys stored in the store.
5. Plan the update of the plugins and WooCommerce as a separate project, tested on a copy of the store.

What the audit did not cover: customer behaviour (Extended variant; the store had no traffic), content and search rankings, legal compliance, e-mail deliverability, hosting-side backups, and five plugins that are not in public vulnerability databases.

2. Performance

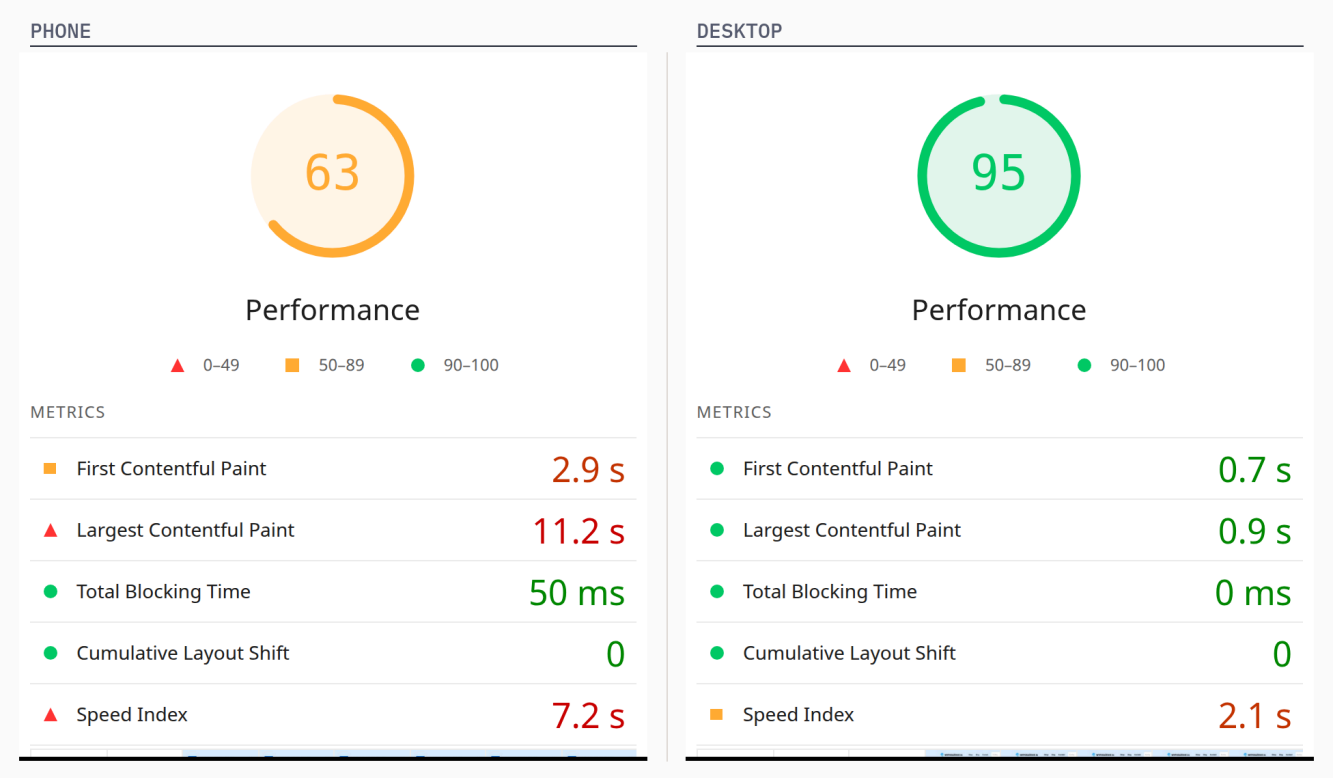
2.1 Measurement

Lighthouse 12.8.2 in Chromium on the auditor's computer, mobile and desktop profiles, simulated network and CPU throttling (default settings), three runs per page, median; 23 September 2026, 13:30-13:54 UTC+2. PageSpeed Insights was not possible, because the audit copy is behind a password; a comparison of this setup with PageSpeed Insights on a public website (mobile: a difference of a few points, within the spread; desktop: the same score) is in “Audit conditions and deviations”.

The page cache is switched off on the copy (it answered before the password check), so every measurement shows a page generated by PHP. For phones this is the store's normal state (point 2.2); for desktop, the state on the first visit after the cache has been cleared. The cart was measured empty.

Device	Page	Score (3 runs)	FCP	LCP	TBT	Server response	Weight	Requests
Phone	home	63 (56, 65, 63)	2.92 s	11.19 s [1]	54 ms	605 ms	4,268 KB [1]	161
Phone	category	71 (57, 71, 72)	3.11 s	5.04 s [1]	50 ms	677 ms	4,657 KB [1]	164
Phone	product	66 (67, 66, 66)	3.11 s	8.74 s [1]	84 ms	511 ms	4,762 KB [1]	171
Phone	cart	86 (86, 86, 87)	2.96 s	3.33 s	54 ms	422 ms	1,390 KB	134
Desktop	home	95 (95, 94, 95)	0.73 s	0.91 s [1]	0 ms	774 ms	4,991 KB [1]	166
Desktop	category	90 (81, 93, 90)	0.94 s	1.38 s [1]	0 ms	883 ms	4,855 KB [1]	166
Desktop	product	96 (94, 96, 97)	0.75 s	0.84 s [1]	0 ms	735 ms	7,496 KB [1]	189
Desktop	cart	96 (98, 96, 95)	0.82 s	0.95 s	0 ms	433 ms	1,390 KB	136

[1] On the audit copy, images come from the files on the server and from the wholesaler's feed; some return a 404 error, because the bucket with the store's images stopped working together with the domain. The weight and LCP of pages with images do not describe the store. Layout shift (CLS) is 0-0.043 on all pages, which is within the norm.



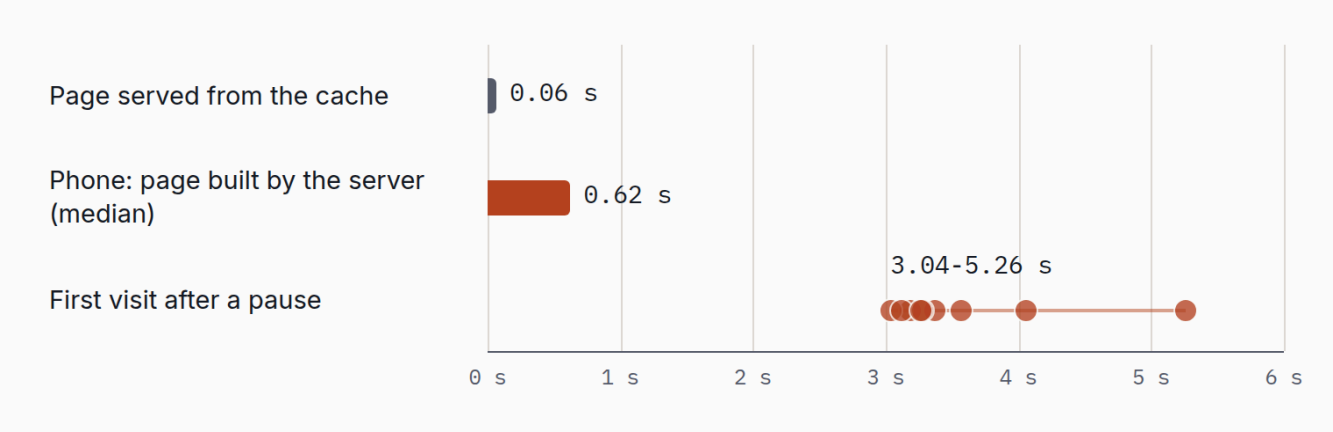
The home page in Lighthouse, the run with the median score: left, phone (63); right, desktop (95). The largest element time on this page depends on the images, so it does not describe the store (footnote [1]). Audit copy, 23 September 2026.

Server response time, a `curl` series from the auditor's computer (home page and product page, mobile and desktop profiles, 10 samples, UTC+2: nine on 23 September 2026, 14:12-22:32, every 53-68 minutes, the tenth on 24 September 2026, 08:39-08:43):

Device	Page	Median	Shortest	Longest	First request after a pause
Phone	home	0.65 s	0.54 s	0.73 s	3.18 s and 3.37 s
Phone	product	0.57 s	0.50 s	0.66 s	4.06 s, 3.04 s and 3.57 s
Desktop	home	0.83 s	0.72 s	1.00 s	3.12 s and 3.28 s
Desktop	product	0.72 s	0.61 s	1.85 s	3.27 s and 5.26 s

Median, shortest and longest come from requests sent 1-2 minutes after the previous one (7-8 per row). The last column is the first request of a sample, after 38-64 minutes without traffic on the copy (in the tenth sample after 10 hours, the first request of that day).

For comparison, a page served from the cache: 0.061 s (home) and 0.062 s (product), single measurements on 23 September 2026, 13:17-13:19 UTC+2, before the cache on the copy was switched off. The first visit after more than ten minutes without traffic took 3.28 s (14:05 UTC+2) and 3.64 s (13:17 UTC+2, the first request after the copy was started).



Server response time on the audit copy: a page served from the cache 0.06 s, a page built by the server on a phone 0.62 s (median of both pages in the table), the first visit after a pause 3.04-5.26 s (nine measurements from the last column of the table).

2.2 Phones get neither cached pages nor optimised code `priority: high`

The LiteSpeed Cache 7.1 plugin has a separate cache for phones switched on. This works only when the `.htaccess` file holds the plugin's rule that marks requests from phones. Without that rule, the plugin marks every request from a phone as not cacheable (plugin code: `src/control.cls.php`, lines 738-758), and a page that is not cacheable does not go through the CSS and JS optimisation either.

A copy of the `.htaccess` file from 22 June 2025 has this rule; the file in use until the store was shut down (last changed on 4 September 2025, when the Wordfence firewall was configured) no longer has it. On the same server and with the same rules, the audit copy answers like this (23 September 2026, 14:09-14:10 UTC+2):

- desktop profile: the header `x-litespeed-cache-control: public,max-age=604800`, 57 CSS files from the plugin's optimisation directory, and the plugin's note in the page code about optimisation and caching;
- phone profile: `x-litespeed-cache-control: no-cache` and `cache-control: no-cache, no-store`, 57 original CSS files, and the note “Page uncached”.

The effects on a phone: every category and product page is generated anew (times in point 2.1); Lighthouse sees 9-12 files blocking the first render, against 2-8 on desktop, and estimates that taking them out of the way would bring the first contentful paint forward by 1.7-2.15 s; the `no-store` header switches off the browser's page cache, so going back from a product to the category with the “back” button downloads and generates the category again.

> `Recommended fix.`
> Save the LiteSpeed Cache settings again, so that the plugin writes back the rule for phones, and check that a second visit from a phone to the same page is served from the cache.

2.3 Every new visitor loads the first page twice `priority: high`

Guest Mode is switched on in LiteSpeed Cache. A browser without the plugin's cookie gets the page, then the plugin's script asks the server for the cookie and reloads the page (code: `lib/guest.cls.php`, lines 49-81, and `assets/js/guest.min.js`). Browsers that identify themselves as measurement tools (Lighthouse, GTmetrix, Google, Pingdom, bots) are exempt from this, so tests do not show it.

Confirmed in measurement: the first two Lighthouse runs, still without the plugin's cookie, loaded the page document twice, together with the cookie request; once the cookie was set, the following runs loaded the page once. The first visit from an ad or a search engine therefore means two full loads, and on a phone both are generated by the server.

> Recommended fix.
>
> Switch off Guest Mode and Guest Optimization in LiteSpeed Cache, then check with a visit from a fresh browser profile that the page loads once.

2.4 Object cache configured, but not working `priority: high`

Redis is selected as the object cache in LiteSpeed Cache, but the `wp-content` directory has no `object-cache.php` file, and without it WordPress does not use the object cache. On the hosting account, the PHP `redis` extension is loaded and there is a Redis service directory (read from the server on 23 September 2026). Without an object cache, pages outside the page cache (phones, cart, checkout, account) run the full set of database queries every time. Whether the Redis service is running and what memory limit it has, the connection test in the plugin settings will tell.

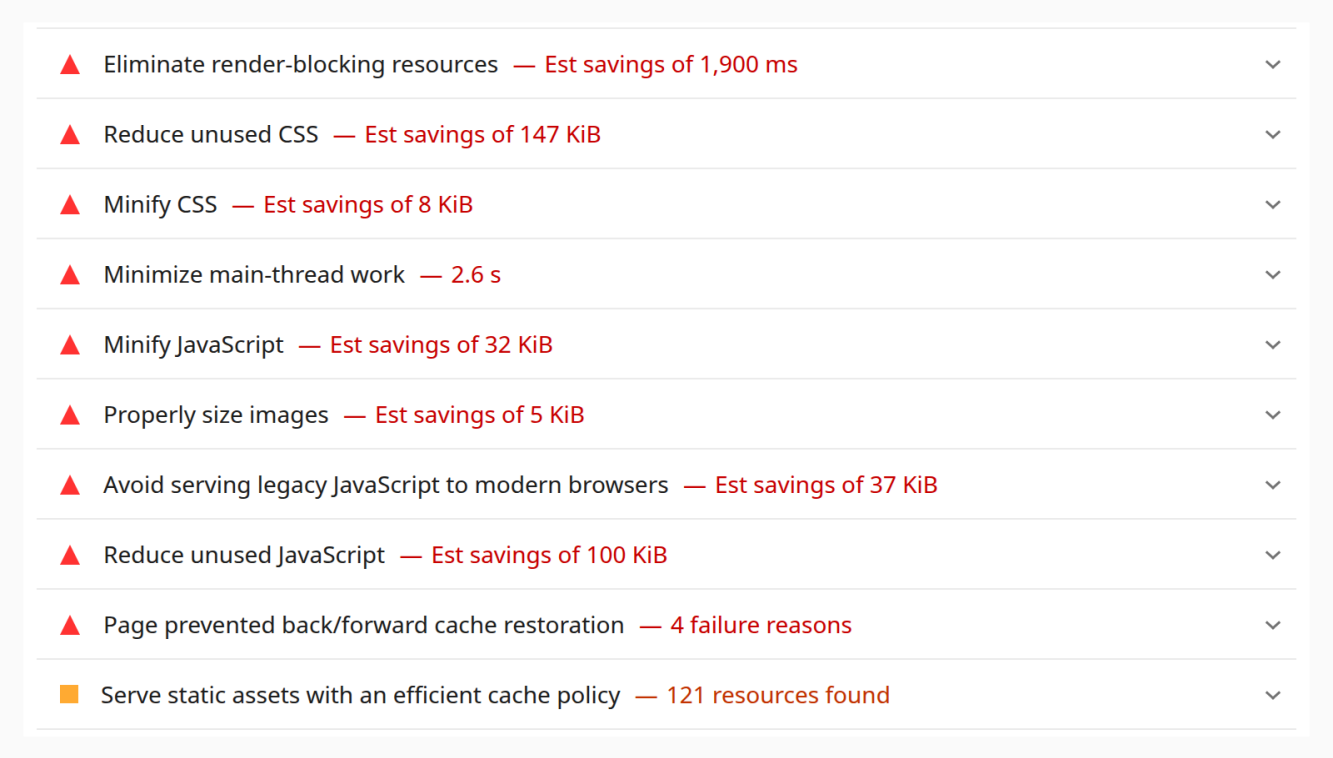
> Recommended fix.
>
> Turn on the object cache with the Redis address given by the hosting provider, and compare the server response time before and after the change.

2.5 Weight of the page code `priority: high, project`

A product page on a phone, without images, is 148 files and about 1.26 MB: scripts 575 KB (59 files), fonts 393 KB (24), styles 246 KB (61), HTML 47 KB. Every page loads 56-61 CSS files and 50-60 JS files; Lighthouse points to 134-158 KB of unused CSS and 74-168 KB of unused JS per page. Two Font Awesome icon files weigh 254 KB together and load everywhere.

Fonts come from Google's servers (8-9 files) and also from Elementor's local copies (Roboto, Poppins, Roboto Slab); on the audit copy, another 9-11 font files point to the old domain and do not load, so the store had even more fonts. The largest single script is Google Tag Manager (172 KB).

The consent banner script (CookieYes, 34 KB) blocks the rendering of every page: about 1.07 s in the phone simulation and about 0.26 s on desktop. The plugin inserts it as the first script on purpose, so that it can stop other scripts until consent is given, so it should not be deferred; only a review of the consent tool will bring a gain. Pages have 1,266-2,927 elements, and processing them keeps a phone's CPU busy for 1.7-3.1 s.



Lighthouse, a product page on a phone, the run with the median score (66): opportunities with estimated savings. At the top, the files blocking the first render, then, among others, unused CSS and JavaScript and 121 files without a long cache lifetime (point 2.6). Audit copy, 23 September 2026.

> Recommended fix.
>
> Slim down the page code as a separate project: load theme and plugin assets only where they are used, keep one icon library and serve fonts from your own server (chapter 7).

2.6 Other

- No browser cache rules for static files `priority: medium`: Lighthouse finds 106-121 files per page without a long cache lifetime. The block with these rules disappeared from `.htaccess` together with the block from point 2.2.
- The image that is the largest element of the home page on a phone has the `loading="lazy"` attribute `priority: medium`, so the browser downloads it later than it could.
- Cart requests on every page `priority: medium`: the side cart plugin (Side Cart WooCommerce 2.6.6) asks the server for the cart contents after every page load, even when the cart is empty; on the home page, in the cart and on the confirmation page, WooCommerce sends a second such request. These are POST requests, which the page cache does not handle, so even a page served from the cache costs the server one or two PHP calls. On the audit copy they took 0.18-1.0 s (median 0.26 s, 30 requests in four walk-throughs of the purchase path), longest on the first visit on desktop.

> Recommended fix.
>
> When saving the LiteSpeed Cache settings again (point 2.2), switch on the browser cache as well. Exclude the first images of the home page from lazy loading. Refresh the cart only when the customer has added something to it, from a single source.

Not checked: PageSpeed Insights results from Google's servers (the copy is behind a password); data from real devices (Chrome UX Report, Search Console; the domain has expired); the weight and format of the store's images (the bucket is unavailable); cache hits under traffic (one pair of measurements); behaviour under load; database queries on individual pages.

3. Security

3.1 Versions and published vulnerabilities priority: critical for the core, high for plugins

WordPress 6.8.1 (current patch release of this branch: 6.8.10, latest version: 7.1.2), WooCommerce 9.9.3 (latest 11.1.2), PHP 8.3 (security fixes only, until 31 December 2027). A review of public vulnerability databases (Wordfence, Patchstack, WPScan, NVD, GitHub Security Advisories) against the installed versions of 38 components (35 plugins, the core, the Porto theme, PHP), as of 23 September 2026, without a single request to the store: 92 vulnerabilities in 19 components, 14 components without known vulnerabilities for their version, 5 impossible to check (absent from public databases: paid, custom-built or little-known plugins). CVSS bands: critical 2, high 27, medium 55, low 8. Exploitable without logging in: 45, of which 10 require an action by the victim (a click on a crafted link).

Component	Installed	Latest	Vulnerabilities	Of which without login
Rank Math SEO	1.0.246	1.0.279	13	4
WordPress (core)	6.8.1	6.8.10 / 7.1.2	12	3
Customer Reviews for WooCommerce	5.78.1	5.122.0	11	8
Fluent Forms	6.0.4	6.2.14	10	4
CTX Feed	6.5.66	8.0.26	6	1
YITH WooCommerce Wishlist	4.6.0	4.18.1	5	5
LiteSpeed Cache	7.1	7.9.1	5	4
WooCommerce	9.9.3	11.1.2	5	1
the other 11 components			25	15

The two critical vulnerabilities:

- **WordPress core, CVE-2026-87902** (CVSS 4.0: 9.2, without login): path traversal in page template selection. Fixed in 6.8.10 of 22 September 2026; according to Patchstack, the first exploitation attempt appeared the same day at 19:44 UTC+2, a few hours after the fix was published. According to the published description, code execution requires a theme with a `page-*` directory and a `pearcmd.php` file available on the server. None of the installed themes (Porto, the child theme, four default themes) has such a directory (checked in the files), so that route is closed here. The update still comes first, because it is a fix within the same version branch, and attackers are already trying to exploit this vulnerability.
- **CTX Feed, CVE-2026-66709** (CVSS 9.1): code execution from an account with the Shop Manager role. The store has one such account, without two-factor authentication.

> Recommended fix.
>
> Update WordPress to 6.8.10 straight away. Update the plugins, the theme and WooCommerce as a separate project, with a test of the purchase path on a copy of the store, starting with the plugins with the most serious vulnerabilities (chapter 7).

3.2 Updates had no way to arrive priority: high

25 of the 27 plugins from wordpress.org are behind, among them Elementor 3.29.2 (120 newer releases), WooCommerce 9.9.3 (98), CTX Feed 6.5.66 (93), Customer Reviews (60), Rank Math (42), Fluent Forms (37) and the Stripe payment plugin (36). The directory dates in the copy of the files show that since the move on

17 June 2025 only two plugins have changed (FluentSMTP on 3 September 2025, the invoicing plugin on 4 September 2025). Automatic updates are switched off for all 35 plugins.

By default, the WordPress core installs security fixes by itself, but here it could not: the site directory holds a `.git` code repository, and WordPress then treats the site as run by a developer and skips all background updates (`wp-admin/includes/class-wp-automatic-updater.php`, the `is_vcs_checkout` function). Only the "Site Health" screen says so. The database holds no trace of the automatic updater ever completing an update.

> Recommended fix.
>
> Remove the code repository from the site directory (point 3.5), so that the core installs security fixes by itself again, and update the plugins on a fixed schedule, tested on a copy.

3.3 A firewall without alerts, logins without a second factor priority: high

Wordfence 8.0.5 (free version) has no notification address entered. The plugin's alert counter shows 42; the plugin increments it before it checks the address, and with an empty address it ends the sending without an error (`lib/wordfenceClass.php`, lines 7442-7445). The owner's mail archive holds not a single Wordfence alert from this store.

The last scan (a quick scan, 12 June 2026, 06:11 UTC+2) reported 26 new issues; 27 are open (12 critical plugin updates, 10 medium, 3 for the theme, 1 for WordPress, 1 skipped path). Comparing plugin and theme files against the originals from wordpress.org is switched off, so a change in their code would not be reported unless it matched a known malware pattern.

No account has two-factor authentication. The login log (September 2024 to May 2026) holds 19 successful logins and 156 failed ones, all with non-existent usernames ("admin" 148 times). Two fatal PHP errors occurred on the login page (1 May 2026, 12:11 and 3 May 2026, 18:46 UTC+2): Wordfence failed to handle an IP address containing a disallowed character; they concerned single requests (source: WordPress recovery mode e-mails).

> Recommended fix.
>
> Enter a notification address in Wordfence that someone reads, turn on two-factor authentication for the administrator and the shop manager, lock out login attempts with non-existent usernames straight away, and switch on the comparison of plugin and theme files.

3.4 Service keys in the code and in the database priority: high

An OpenAI API key is stored in plain text in the code of a custom-built plugin ("WooCommerce SEO Optimizer", `wp-content/plugins/woo-seo-optimizer/index.php`, line 163) and in two settings in the database; it is the same key (hashes compared, without printing the value). On the day of the audit the key was active (checked with a read-only request for the list of models), so anyone with a copy of the files or the database could use it at the account owner's expense; that was the only critical priority in this point. The owner revoked the key on 24 September 2026, and the same request then returned an authorisation error (09:07 UTC+2).

The database also holds live Stripe keys, a CookieYes account token, Rank Math access tokens to Google services and the SMTP mailbox password, and `wp-config.php` holds the keys to the Cloudflare R2 bucket.

> Recommended fix.
>
> Revoke the other keys with the providers (in a trading store: replace them), remove the OpenAI key from the plugin's code and from the database, and keep keys in the server configuration, not in the code.

3.5 Files in the site directory that should not be there `priority: critical`

- The `.git` repository: 64 commits from 5 September 2024 to 13 June 2025, about 400 MB. Its history includes, among other things, the code of the custom-built plugin (the file changed in 5 commits); it contains neither `wp-config.php` nor database dumps. Up to the day of the shutdown, the store's `.htaccess` file did not block access to `/.git`, and the server serves hidden files (next point), so the repository may have been downloadable from outside. This cannot be confirmed, because the access logs from the time the store was running no longer exist.
- 5,998 `.listing` files left by FTP copying; each holds a list of the files in its directory with sizes and dates. The server serves them to anyone: `/.listing` on the audit copy answers with status 200 (2,313 bytes, 23 September 2026, 14:08 UTC+2). That is a listing of directory contents despite directory listing being switched off.
- `wp-content/debug.log` (111 KB, from 4 March 2025), `cf-images.log` in the uploads directory (4.8 MB, 104,629 lines), copies of `.htaccess`, code editor configuration files.

> `Recommended fix.`
>
> Remove everything on this list from the site directory (keep the repository outside it, if it is needed) and leave a server rule in place that blocks hidden files and copies.

3.6 Server and WordPress configuration

- No `.htaccess` file in `wp-content/uploads` blocking PHP execution `priority: high`: a PHP file uploaded through a vulnerability in one of the plugins will run.
- No security headers (HSTS, `Content-Security-Policy`, `X-Frame-Options`, `X-Content-Type-Options`, `Referrer-Policy`, `Permissions-Policy`); the `x-powered-by` header gives away the PHP version `priority: medium`. None of the versions of the store's `.htaccess` file sets these headers, and the response of the audit copy on the same server does not have them (23 September 2026, 14:09 UTC+2).
- Editing PHP files from the WordPress dashboard is enabled (no `DISALLOW_FILE_EDIT`) `priority: medium`.
- In order: account registration closed, three accounts (administrator, shop manager, customer), no application passwords, unique keys and salts in `wp-config.php`, the invoice directory closed off by a server rule, no PDF files in the directory of the second invoicing plugin.

> `Recommended fix.`
>
> Add a rule in the `wp-content/uploads` directory that blocks PHP execution, set the security headers and hide the PHP version, and switch off file editing from the dashboard with the `DISALLOW_FILE_EDIT` constant.

3.7 Backups `priority: high`

WordPress holds one Duplicator entry from 27 September 2024, and the file of that backup does not exist. There is no other backup plugin. Hosting-side backups were not checked.

> `Recommended fix.`
>
> Confirm the scope and retention of the hosting backups, restore one at a separate address, and remove the dead Duplicator entry.

Not checked: backups and accounts in the hosting panel and FTP; the five plugins that are not in public vulnerability databases; whether the OpenAI key is in the repository history; XML-RPC availability and username disclosure; access logs from the time the store was running (they do not exist). Vulnerabilities were not tested with requests: the findings come from reading versions against published advisories.

4. Store configuration

4.1 Stock levels and the import from the wholesaler priority: critical

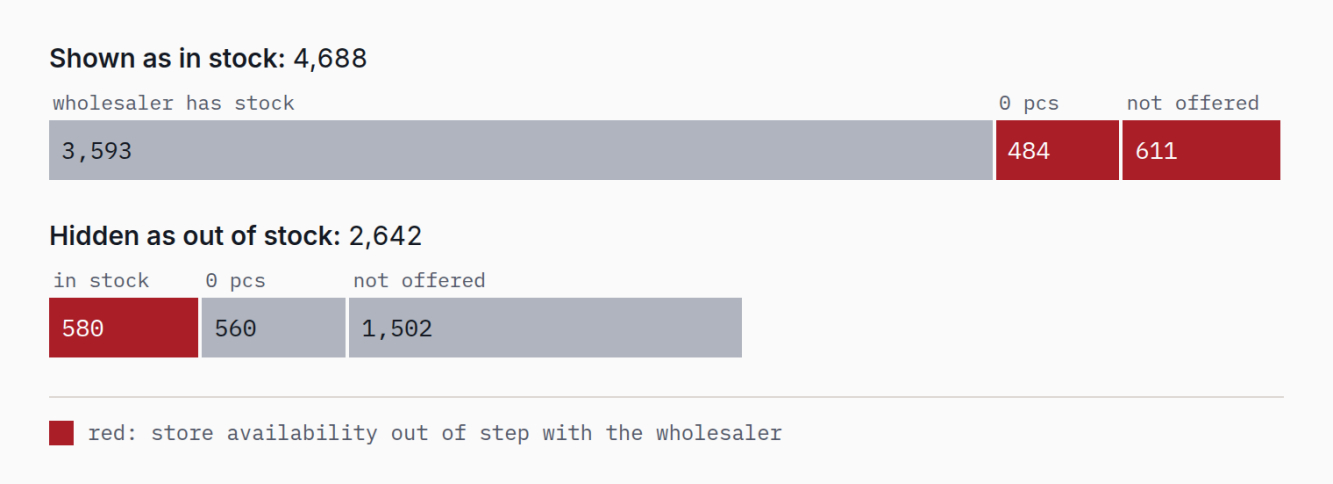
Products and stock levels come from the wholesaler through WP All Import Pro, and the import is started by an external scheduler (cron-job.org). The last completed import started on 15 June 2025 at 00:00 UTC+2, and the run took about 1 h 52 min (6,308 products updated, 9,157 rows skipped).

The same day at 15:02 UTC+2, the scheduler disabled the import processing job after 26 consecutive server errors (HTTP 500) and sent an e-mail about it. The import started at midnight on 16 June 2025 was never processed and stayed marked as started until the store stopped trading. The scheduler disabled the job that starts the import in the night from 7 to 8 July 2026 (8 July 2026, 00:01 UTC+2, error 404). The scheduler had disabled the processing job the same way before (30 December 2024, 30 March 2025). The cause of the 500 errors cannot be established today, because the server logs from that time no longer exist; it coincides with the store's move to a new server (12-17 June 2025).

The store's stock against the wholesaler's feed downloaded on 23 September 2026 at about 08:45 UTC+2 (matched by SKU, 7,330 products):

Status in the store	In stock at the wholesaler	0 units at the wholesaler	Not in the wholesaler's range	Total
in stock	3,593	484	611	4,688
out of stock (hidden)	580	560	1,502	2,642

The wholesaler does not have 1,095 of the 4,688 products marked as in stock (23%) today; the store hides 580 products it does have (the setting “Hide out of stock items from the catalog” is on). The comparison measures the drift after 15 months without synchronisation, not the state on the day of the shutdown. It covers availability only, because the store recalculates prices on import (among other things, it adds the shipping cost to products shipped separately).



The same table as a chart. The red segments are the drift the customer sees: 1,095 products marked as in stock that the wholesaler does not have, and 580 hidden ones that the wholesaler has in stock.

> Recommended fix.
>
> Find the cause of the processing errors, run the import from the server's cron, and send a notification to an address someone reads when an import does not finish within a day.

4.2 State on the day of the shutdown

Coming soon mode on, all payment methods switched off. That is consistent with the store being closed, not a fault. In the history of 11 orders (August 2024 to September 2025), customers chose cash on delivery (7) and Przelewy24 handled by the Stripe plugin (4). The Stripe plugin stayed in live mode with its keys in the database (point 3.4).

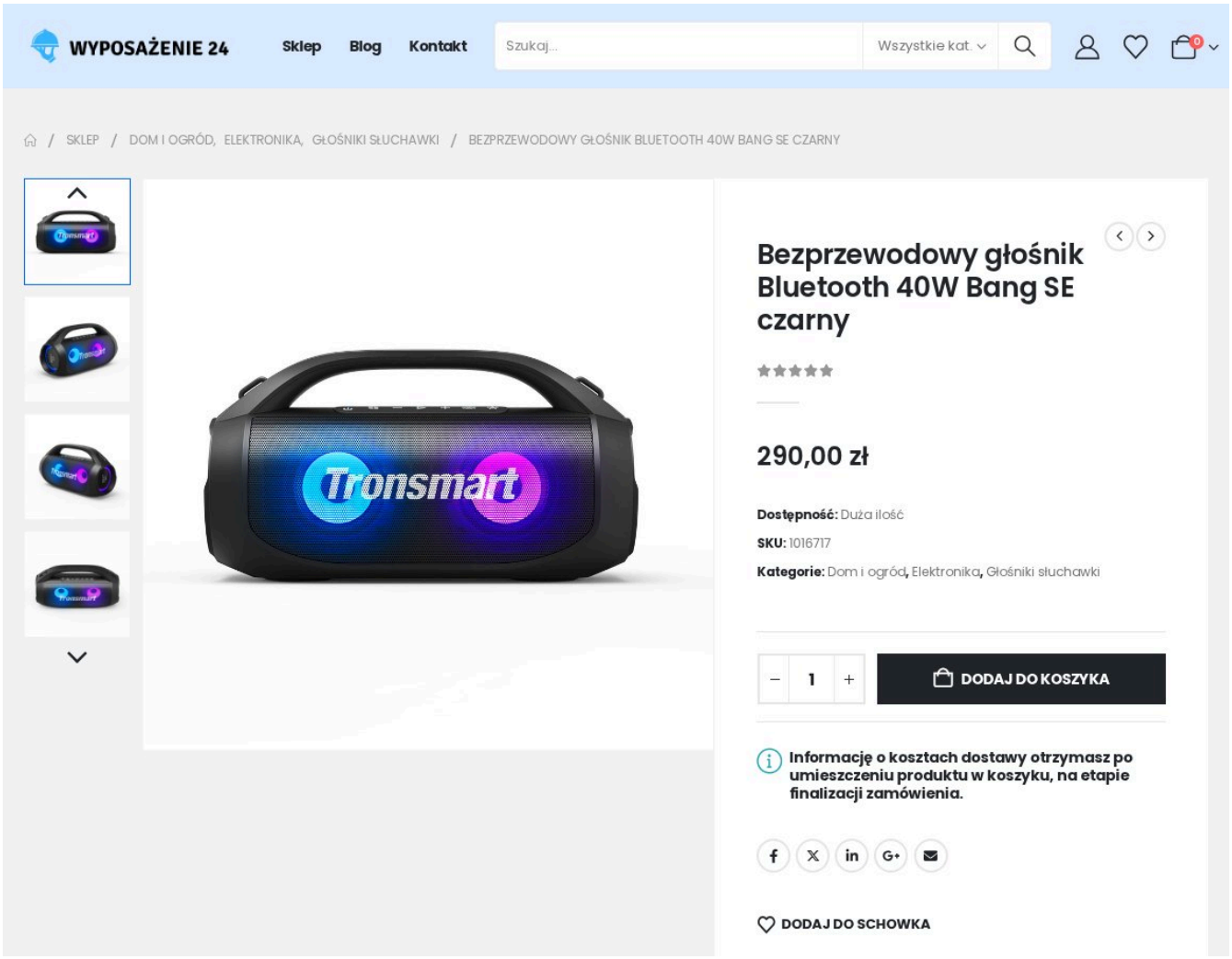
The bank transfer settings still hold WooCommerce's sample data: “Przykładowe konto bankowe” (“sample bank account”), number 555 555 555, no IBAN `priority: critical the moment bank transfer is switched on`. With bank transfer off this has no effect, but once it is on, the customer gets these details on the order confirmation page and in the e-mail, and does not know where to pay (checked with a test order, point 4.3).

> Recommended fix.
>
> Before switching bank transfer on, enter the store's bank account and check the e-mail to the customer with a test order.

4.3 Checkout and the purchase path

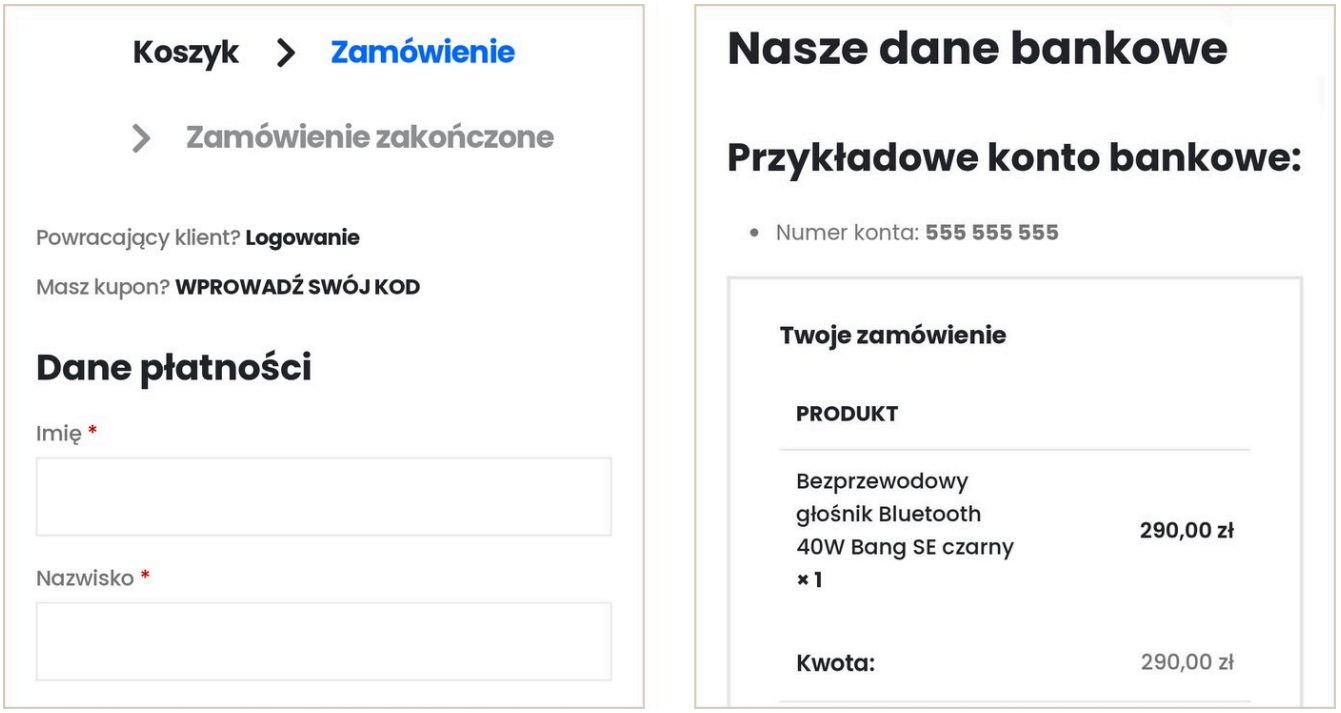
Guest checkout on, a one-step checkout (the classic one, on the WooCommerce shortcode), the shipping cost visible in the cart (example: product 290.00 PLN, shipping 19.00 PLN, total 309.00 PLN including 23% VAT).

The purchase path (home page, shop, product, add to cart, cart, checkout) was checked on the audit copy: two walk-throughs on a phone (390 px) and two on desktop (1440 px), 23 September 2026, 15:12-15:25 UTC+2. All pages respond, the server returns the page document in 0.38-0.94 s (without the page cache, which is switched off on the copy; one exception: the first page of the walk-through with the test order, 3.2 s), and adding to the cart works without reloading the page. In every walk-through the first page loaded twice (point 2.3). In the state from the day of the shutdown, the checkout shows a message that no payment method is available (point 4.2).



A product page on desktop (1440 px). The customer sees the delivery cost only in the cart, as the box under the button says; the store's interface is in Polish. Audit copy, 23 September 2026.

The test order (phone, 15:17-15:20 UTC+2, fictitious details, bank transfer switched on for the test): 1.6 s from the click on the order button to the confirmation page loading. The order got the “On hold” status (awaiting payment), the product's stock went down by one unit, and the store generated an e-mail to the customer and a notification to the store. The confirmation page and the e-mail to the customer give the sample account details from point 4.2. After the order was cancelled, the stock went back up.

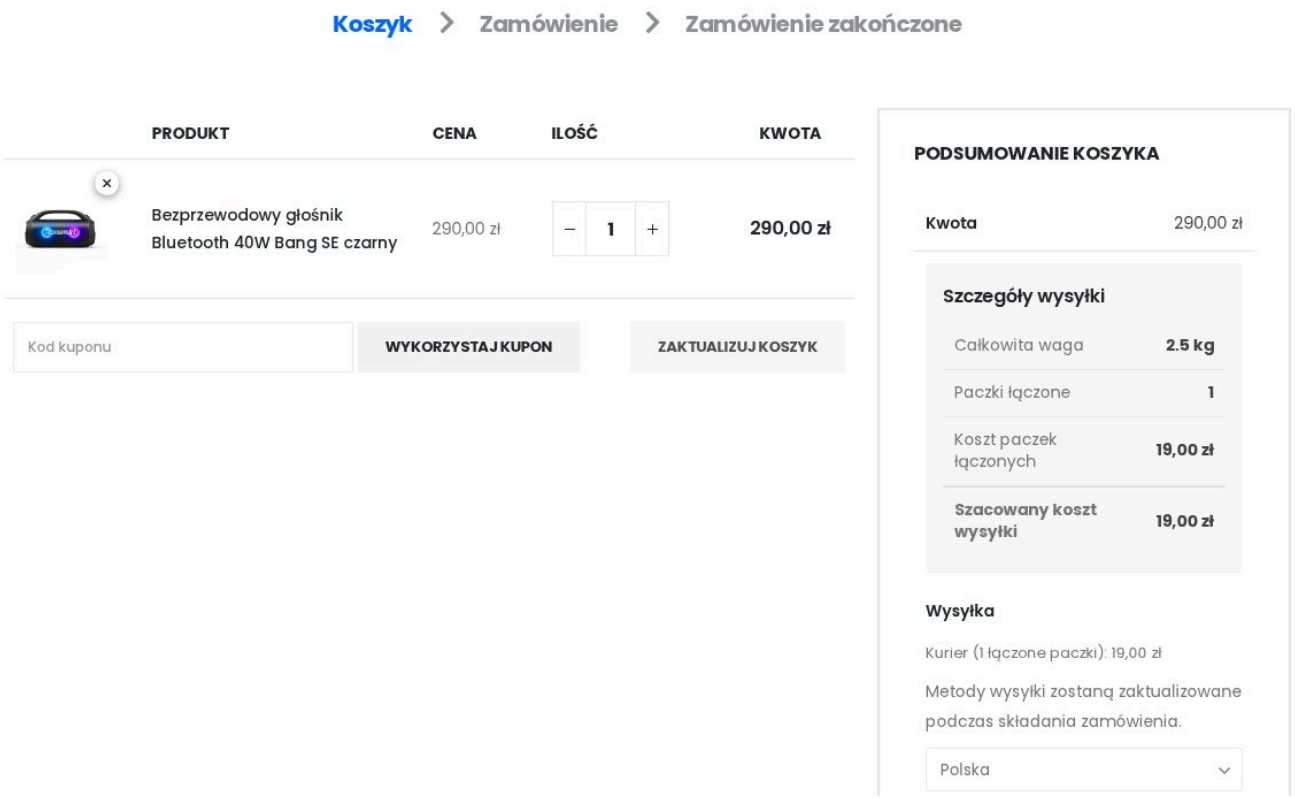


Left, the checkout on a phone (390 px): one step, purchase without creating an account. Right, the confirmation page of the test order with WooCommerce's sample account details (point 4.2); the store's interface is in Polish. Audit copy, 23 September 2026.

On both devices, the only JavaScript error comes from the consent banner script (CookieYes), which is tied to the old domain; with this script blocked, the page reports no JavaScript errors. That is an effect of the copy, not a fault of the store.

4.4 Shipping

The WooCommerce settings have one zone (Poland) and “Kurier” (courier) at 1.00 PLN, but the actual cost is calculated by the child theme's code (`porto-child/functions.php`, `woocommerce_package_rates` filters, lines 180 and 808): 19 PLN per parcel (a custom field in the settings), an individual cost from a product field, combined and separate parcels. For products shipped separately, the import adds the shipping cost to the price (line 144) and marks them as “darmowa dostawa” (free delivery). It works, but the WooCommerce settings show something different from what the customer pays, and changing the shipping rates needs a developer `priority: medium, chapter 5`. The cart label “Kurier (1 łączone paczki)”, literally “Courier (1 combined parcels)”, is ungrammatical `priority: low`.



The cart with a product at 290.00 PLN. The shipping details (weight, combined parcels, 19.00 PLN) are calculated by the child theme's code, not the WooCommerce settings; below them, the label “Kurier (1 łączone paczki)”. Audit copy, 23 September 2026.

> Recommended fix.
>
> When moving the store logic to a plugin (point 5.5), move the shipping rates to settings that show what the customer pays, and fix the grammar of the cart label.

4.5 Taxes, e-mails, prices

- Taxes: VAT 23%, prices include tax, calculated from the shipping address. No remarks.
- Transactional e-mails: default templates, all switched on. The sender in the WooCommerce settings is an address in a different domain from the store's, while the FluentSMTP plugin sends through a mailbox on the hosting server from an address in the store's domain `priority: low; make them the same`. The FluentSMTP log has one entry (1 June 2026), so it does not tell whether e-mails reached customers.
- Lowest price in the last 30 days (the EU Omnibus rule): the Price History plugin, 30 days, shown with a sale price, price history for all 7,330 products. The store had no products on sale, so the display could not be checked.

> Recommended fix.
>
> Set the same sender address, in the store's domain, in WooCommerce and in the SMTP plugin.

4.6 Analytics and consent

Google Analytics 4 and Microsoft Advertising (UET) tags are inserted into the page code and loaded through the script delay mechanism of LiteSpeed Cache; the child theme adds Bing enhanced conversions in the header and the Google review invitation in the footer, both only on the order confirmation page. The Rank Math analytics module does not insert a second GA code (in order). Consent banner: CookieYes.

- Purchases in Google Analytics 4 `priority: medium`: the GA4 tag in the child theme carries only its configuration, with no store events, so GA4 receives neither the purchase nor the order value and will show neither the revenue nor which traffic source sells. The plugins that could send these events are switched off or have no identifiers (the Rank Math module without code insertion, tracking in CTX Feed without an account number). Whether a purchase was defined by a rule in the GA4 panel (still without the order value) was not checked.
- Bing enhanced conversions `priority: medium`: the code passes a hash of the customer's e-mail address and phone number. The number was meant to go in international format, but the condition that adds the "+" is never met, and the code does not add the country code, so for numbers entered without "+48" matching by phone probably does not work.

> Recommended fix.
>
> Send store events with the order value to GA4, in line with the consent setting. In Bing enhanced conversions, fix the phone number format and check the matching in the Microsoft Advertising panel.

Not checked: whether the tags wait for consent (the CookieYes banner is tied to the old domain and does not work on the audit copy); delivery of e-mails to inboxes (on the copy, mail is captured to a file); online payments in action (Stripe gateways switched off, keys removed from the copy; bank transfer was checked with the test order); how current the CTX Feed product feeds are (Google Shopping and a second feed).

---

5. Housekeeping

5.1 Plugins and themes `priority: medium`

35 plugins: 32 active, 3 inactive (WP MailHog SMTP, a developer tool for capturing mail; Better Search Replace; Preserve Page and Taxonomy Hierarchy 0.1 from outside wordpress.org). Filenames to Latin has had no release since 9 August 2020. Two form plugins (Contact Form 7 and Fluent Forms), both in use (submissions stored in both).

The custom-built plugin "WooCommerce SEO Optimizer" (header from a template, "Author: Your Name") sends product titles and descriptions to the OpenAI API; the key is in the code, point 3.4. Four inactive default themes (Twenty Twenty-Two to Twenty Twenty-Five).

> Recommended fix.
>
> Delete the inactive plugins and three of the four default themes, keep one form plugin, and replace the abandoned plugin with a maintained equivalent.

5.2 Database `priority: medium`

The database has 153 tables and about 0.9 GB; about two thirds of that is Offload Media plugin data: the `wp_acoofm_items` table (about 490 MB) and 71 expired `acoofm_cached_data_*` transients in the options table (about 163 MB together, expired on 12-13 June 2026). The `wp_postmeta` table is about 194 MB.

Options loaded on every request (autoload) take up about 1.4 MB, against the 800 KB threshold at which WordPress warns in “Site Health” (`wp-admin/includes/class-wp-site-health.php` , line 2648). The two largest, about 0.5 MB each, are transients without an expiry: `_transient_dirsize_cache` (the cache of directory sizes) and `_transient_ptk_patterns` (block patterns fetched by WooCommerce). WordPress 6.8 will write the first one again with an expiry, no longer loaded on every request; WooCommerce 9.9.3 writes the second one without an expiry, so it will come back after deletion (to be checked after the WooCommerce update). 97 revisions, 0 orphaned metadata.

> Recommended fix.
>
> After backing up the database, delete the expired transients and `_transient_dirsize_cache` ; clean up the Offload Media table together with the decision from point 5.3.

5.3 Product images `priority: medium, project`

There are 27,542 attachments, and about 29% of them have their files on the server. The Offload Media plugin moved the rest to an external bucket (Cloudflare R2) under an address in the store's domain, which stopped responding when the domain expired (no DNS record, 23 September 2026). The images therefore depend on the domain, the bucket and the licence of a single plugin. Whether the bucket itself still exists, the Cloudflare account will tell. The wholesaler's feed provides images for 5,217 products (71%); for the 2,113 products outside its range, the images exist only in the bucket or in the files on the server.

> Recommended fix.
>
> Decide whether the images stay in the bucket or return to the server, and move them so that they depend neither on the domain nor on the licence of a single plugin.

5.4 Background jobs `priority: medium`

The hosting account has no system cron for the store (read on 23 September 2026), so WordPress jobs run only on visits. Action Scheduler: 250 failed jobs from September 2024 to February 2026, 110 of them in June 2025, mostly the rebuild of the product lookup tables (125) and the clearing of the related products cache (84); 5 jobs pending since 12 June 2026; 22,844 log rows.

> Recommended fix.
>
> Run WordPress jobs from the server's cron every few minutes instead of on visits, and review and clear the failed jobs.

5.5 Store logic in the child theme `priority: medium, project`

The `porto-child/functions.php` file holds 1,239 lines of store logic: shipping (point 4.4), adding costs on import, stock status texts, free shipping labels, Bing conversions, Google review invitations. Changing or breaking the child theme changes shipping prices.

> Recommended fix.
>
> Move the store logic to a separate plugin with settings and a test of the shipping calculation.

5.6 Minor

A setting of the Debug Log Manager plugin points to a path on the previous hosting (a warning in the PHP log); working files in the site directory: point 3.5.

> Recommended fix.
>
> Correct the path in the Debug Log Manager settings, or delete the plugin if it is not used.

Not checked: which features of each plugin are actually used; tables left by deleted plugins, other than those on the list of the largest.

---

6. Customer behaviour, in the Extended variant

Not covered: the sample report covers the Basic variant, and on the day of the audit the store had no traffic. In the Extended variant, this chapter describes the month of tracking: where customers click, at which checkout step they drop out, from which pages they reach the cart and on which devices, each finding with its number of sessions.

7. What to fix, in what order

To click today, without a contractor

- Wordfence: in the e-mail alert settings, enter an address someone reads; in the login module (Login Security), turn on two-factor authentication for the administrator and the shop manager.
- LiteSpeed Cache: on the Cache tab, click “Save Changes” without changing any setting (the plugin will write its server rules again, including the one for phones); in the General settings, switch off Guest Mode.
- Plugins: delete the three inactive ones (WP MailHog SMTP, Better Search Replace, Preserve Page and Taxonomy Hierarchy).
- The service that starts the import (cron-job.org): turn the import jobs back on and check in the job history that a run ends without an error.

Simple (doable within a week, without a project):

Fix	Priority	Area	What it changes	To be quoted by
Update WordPress 6.8.1 to 6.8.10, after backing up the files and the database	critical	Security	Closes 12 core vulnerabilities, including the critical one attackers have been trying to exploit since 22 September 2026	WordPress administrator
Remove from the site directory the `.git` repository, 5,998 `.listing` files, copies of `.htaccess`, logs and editor files; a rule blocking hidden files	critical	Security	Nobody downloads the code history or the list of the server's files	WordPress administrator
Resume the stock import: cause of the processing errors, running from the server's cron, a notification of a failed run	critical	Store configuration	Customers do not order 1,095 products that are not there; 580 available ones return to the catalogue	WordPress developer
Bank transfer: enter the store's bank account in place of WooCommerce's sample data, before the method returns to the checkout	critical	Store configuration	A customer paying by transfer gets the real account number on the confirmation page and in the e-mail	owner (on their own)
Remove the OpenAI key from the plugin's code and from the database (revoked with the provider on 24 September 2026); revoke the other service keys (Stripe, Google, Cloudflare, CookieYes, SMTP)	high	Security	Nobody with a copy of the files or the database uses services at the owner's expense	owner (revoking), WordPress developer (code)
Wordfence: notification address, two-factor authentication for the administrator and the shop manager, lockout for non-existent usernames	high	Security	Alerts arrive; taking over an account needs a second factor	owner (on their own)
LiteSpeed Cache: save the settings again (with Browser Cache), switch off Guest Mode, check `x-litespeed-cache: hit` on a phone	high	Performance	Phones get pages from the cache; the first visit without a double load; static files with a long cache lifetime	owner (on their own), check: administrator
Turn on the Redis object cache with a connection test	high	Performance	Fewer database queries on pages outside the page cache (cart, checkout, account)	WordPress administrator

Fix	Priority	Area	What it changes	To be quoted by
Block PHP execution in `wp-content/uploads`	high	Security	A file uploaded through a vulnerability will not run	WordPress administrator
Confirm the hosting backups with a test restore; remove the dead Duplicator entry	high	Security	It is known what the store can be restored from	WordPress administrator
The first image on the home page without lazy loading	medium	Performance	The browser downloads the largest element of the page straight away	WordPress developer
A system cron instead of jobs started by visits; clear the 250 failed jobs	medium	Housekeeping	Background jobs run on time, regardless of traffic	WordPress administrator
Security headers, hiding the PHP version, `DISALLOW_FILE_EDIT`	medium	Security	Less information for scanners; no code editing from the dashboard	WordPress administrator
Delete 3 inactive plugins and 3 default themes; replace Filenames to Latin	medium	Housekeeping	Less code to update and fewer places for a vulnerability	WordPress administrator
Database: expired transients and `_transient_dirsize_cache`	medium	Housekeeping	Database smaller by about 163 MB; options loaded on every request smaller by about 0.5 MB (another 0.5 MB after the WooCommerce update, to be checked)	WordPress administrator
Side cart: refresh the cart contents only when the cart is not empty, one source of refreshing	medium	Performance	A page served from the cache does not call PHP on every visit	WordPress developer
Analytics: a purchase event with the order value in GA4; the phone number format in Bing enhanced conversions	medium	Store configuration	GA4 shows revenue and the sources of sales; Microsoft matches conversions by phone as well	WordPress developer
Shipping label in the cart, one e-mail sender in WooCommerce and SMTP, the old hosting path in the settings	low	Store configuration	Order in what the customer and the administrator see	WordPress administrator

Larger (a project to be quoted):

Fix	Priority	Area	What it changes	To be quoted by
Updating the plugins, the theme and WooCommerce on a copy with a test of the purchase path, then production; order by vulnerabilities (CTX Feed, Customer Reviews, Rank Math, Fluent Forms), major version jumps separately (Elementor 3 to 4, WooCommerce 9 to 11)	high	Security	Covers 80 of the 92 published vulnerabilities (plugins and theme; for one Porto theme vulnerability WPScan gives no fix); the store is back on supported versions	WordPress developer
Slimming the page code: Elementor and Porto modules, plugin assets only on the pages that use them, one icon library, self-hosted fonts, one form plugin	high	Performance	Fewer render-blocking files; Lighthouse points to 134-158 KB of unused CSS and 74-168 KB of unused JS per page	WordPress developer (front end)
Moving the store logic from the child theme (1,239 lines) to a plugin with settings and tests; shipping settings that match what the customer pays	medium	Housekeeping	Updating or changing the theme does not change shipping prices; rates change without a developer	WordPress developer
Images: decide between the bucket and the server, clean up after Offload Media	medium	Housekeeping	Images do not depend on the domain and the licence of a single plugin; database smaller by about 490 MB	WordPress developer

Audit conditions and deviations

- **The store:** owned by the auditor, shut down on 29 July 2026, domain expired on 6 September 2026. From 12-13 June 2026, the database holds no trace of WordPress at work (background jobs, firewall scans); the cause was not investigated.
- **The audit copy:** the store's files and a dump of its database from 23 September 2026, restored that day on the same server (shared hosting, LiteSpeed server, PHP 8.3) at a separate password-protected address, with a header and a `robots.txt` file forbidding indexing. The copy will be removed from the server once work on the report is finished. Some of the code and database reads were done on a second copy (the same code and the same database) on the auditor's computer; the purchase path and the test order were done on the audit copy.
- **Customer data:** anonymised before the copy went onto the server (11 orders and 3 refunds, 8 customer records, contacts, form submissions, order notes); logs with IP addresses cleared. Check after anonymisation: 0 e-mail addresses outside the domains of the store and its owner in the whole database (before: 306 rows). Service keys removed from the database and the code of the copy. Findings about customers' personal data were passed to the owner separately; they are not in this version of the report.
- **Changes on the copy:** coming soon mode switched off, so that the store could be measured; mail captured to a file; connections to external services blocked; WordPress scheduled tasks switched off. For the test order (23 September 2026, 15:17-15:20 UTC+2), bank transfer was switched on; the order was cancelled, the stock went back up and bank transfer was switched off. In the browser used for the purchase path, sending data to Google Analytics and Microsoft Advertising was blocked, so that the test order would not end up in the store's statistics. The core, plugins and theme were not updated: the neglect is part of what the audit examines.

- **Images and fonts:** images from the files on the server (about 29% of attachments) and from the wholesaler's feed; the rest return a 404 error. Some fonts point to the old domain and do not load (point 2.5), so the pages of the copy are lighter than in the store.
- **Measurements:** Lighthouse instead of PageSpeed Insights (the copy is behind a password), from the auditor's computer; the page cache switched off on the copy, because it answered before the password check. The measurement setup was compared with PageSpeed Insights on 23 September 2026 on the auditor's public website, three runs each: mobile score 89 against 93 (runs 82-95 against 84-98), desktop 99 in both. The difference between the medians on mobile is within the run-to-run spread of both tools, so the results from the copy should be read with a tolerance of a few points. Individual metrics differ in both directions (mobile: TBT 189 ms against 71 ms, LCP 2.86 s against 3.08 s), and PageSpeed Insights uses a newer version of Lighthouse (13.5 against 12.8). The comparison site is light, so agreement for script-heavy pages was not checked. Findings about plugin code come from reading the code of the installed versions, findings about the state of the data from the store's database, and findings about the server from read-only access to the hosting account.
- **Chapter 6:** does not exist (Basic variant, no traffic in the store).