

Nordaloom: what was found and what was fixed

Prepared for: Ilze Ozola, owner of Nordaloom, a knitwear shop in Latvia (fictional)

Prepared by: Ilya Shkura

The app: React, TypeScript, Vite, Tailwind, shadcn/ui, Supabase (database, sign-in, storage, server functions), Stripe Checkout

FOUND

27

1 critical, 3 high, 9 medium,
14 low

TESTS FAILING AS RECEIVED

42

of 51 written for this app

AFTER THE FIX

0

failing; 23 of 27 fixed, 4 left
to the owner

Summary for the owner

The shop looked finished and held up in a normal walk-through: prices, stock and totals were right. One defect exposed personal details with no effort: any account could open any guest's order by its number and read the name, address, email and phone. Three more touched money or personal details in ordinary use: after a dropped connection a customer could be shown someone else's order, the owner could record a refund of any size, and a customer could pay by card for an order the shop had already cancelled and not get the money back. All four are fixed, along with 19 smaller ones; four low-priority points are left to the owner, with the reasons. Every fix has a test that failed on the shop as received and passes now.



As received: Built in 17 plain-language messages. The builder's answers included "Orders can't be placed twice", "a customer's orders and addresses are removed from the device when they sign out" and, in the last one, "The dashboard, reports and stock all agree with those orders; I checked each of them against the orders directly."

What was fine: Nobody without an account could read customers' data, and no account could read another account's orders, addresses or returns. Prices and stock are decided by the database, not by the page. The totals of all 2,059 orders added up to the cent, and the reports matched an independent recount. Forged payment notices were refused.

What mattered most

CRITICAL Any signed-in account can open any guest's order by its number

The order page opens: name, street, city, email, phone, what was bought. Counting down 60 order numbers opened 29 orders, every guest order among them; about four in ten of the shop's orders are guest orders.

HIGH After a dropped connection, the next order on the same device shows someone else's order

The second customer pressed "Place order · €53.99" and was shown the first customer's order for €29.99, with that customer's address. Their own two pieces were never ordered.

HIGH A refund of any size can be recorded against a return

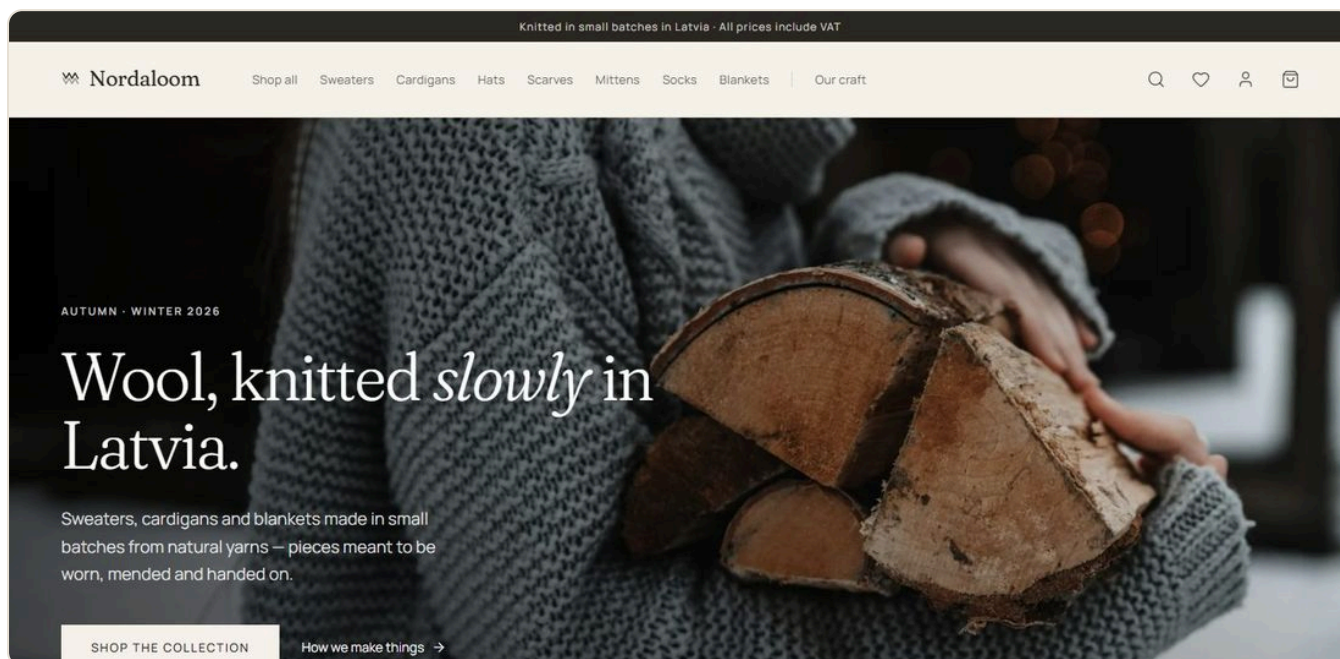
Accepted. The order shows €500 refunded, the customer is emailed "Your refund of €500 is on its way", and the month's refunds grow by €500.

HIGH A customer can pay by card for an order the shop has already cancelled, and is not refunded

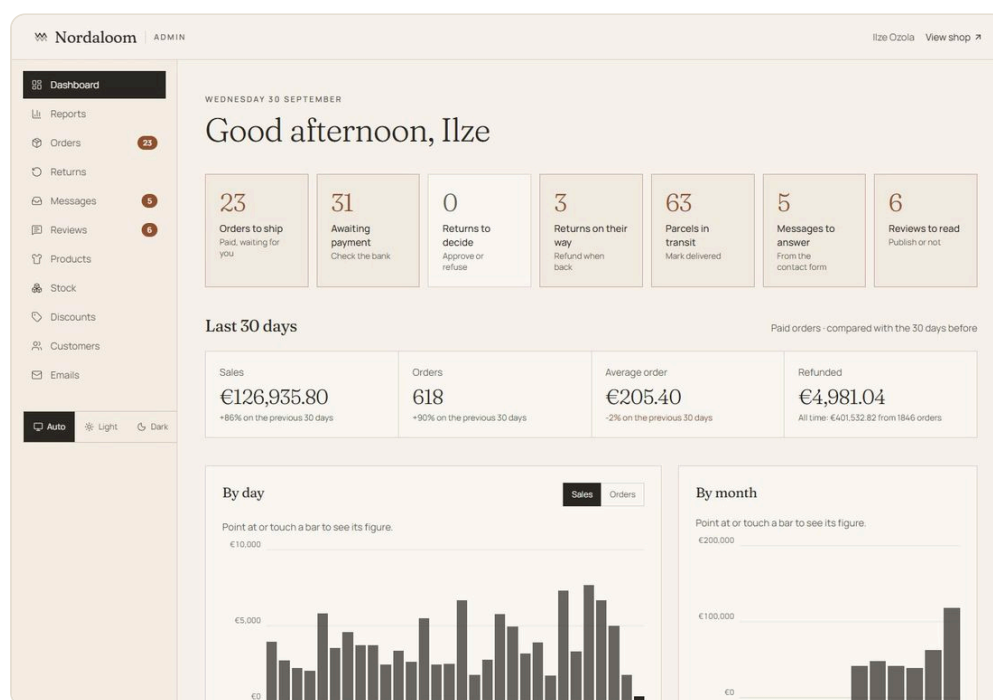
The customer paid €27.99. The shop showed "This order was cancelled", recorded no payment and refunded nothing.

Nordaloom as it was received

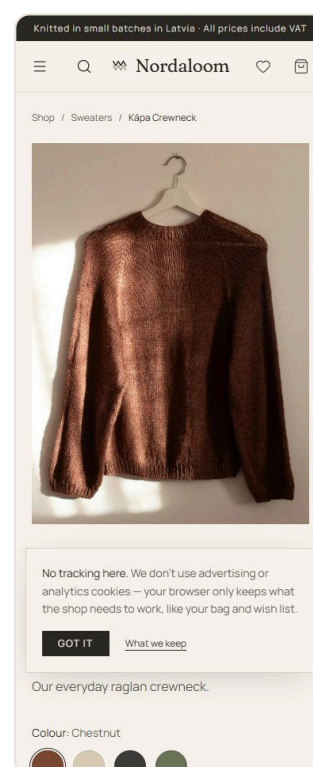
An online shop for knitwear: catalogue with search and filters, bag, checkout by card or bank transfer, customer accounts, returns, reviews, discount codes, emails, an owner's admin with a dashboard and reports, and installation on a phone with offline use.



The shop's home page, as the builder left it after the last message.



The owner's admin: orders to ship, payments awaited, returns, sales by day and by month.



A product page on a phone.

All findings

#	SEVERITY	FINDING	AREA	AFFECTS	STATUS
27	CRITICAL	Any signed-in account can open any guest's order by its number	Data	Customers	Fixed
1	HIGH	After a dropped connection, the next order on the same device shows someone else's order	Orders	Customers	Fixed
2	HIGH	A refund of any size can be recorded against a return	Money	Owner and customers	Fixed
3	HIGH	A customer can pay by card for an order the shop has already cancelled, and is not refunded	Money	Customers	Fixed
4	MEDIUM	Without a connection the installed shop is an empty page on some hosts	Offline	Customers	Fixed
5	MEDIUM	After signing out, the device still opens the customer's order	Data	Customers	Fixed
6	MEDIUM	Anyone can place any number of unpaid orders, to any email address	Stock	Owner	Fixed
7	MEDIUM	The order can be made at a different total than the button showed	Money	Customers	Fixed
8	MEDIUM	An order returned in two parts is offered its delivery charge twice	Money	Owner	Fixed
9	MEDIUM	A discount code from the made-up history gives real customers 20% off	Money	Owner	Fixed
10	MEDIUM	When the shop cancels an unpaid order, the customer is not told	Orders	Customers	Fixed
11	MEDIUM	Dates and deadlines change with the clock of the device	Orders	Owner and customers	Fixed
12	MEDIUM	The bag is wrong without a connection	Offline	Customers	Fixed
13	LOW	The customer page counts refunded money as spent	Money	Owner	Fixed
14	LOW	The shop's settings can be read by anyone, including the welcome discount code	Access	Owner	Fixed
15	LOW	"We couldn't reach the shop" when the order was in fact placed	Orders	Customers	Fixed
16	LOW	The return form shows the list price, not what will come back	Money	Customers	Fixed
17	LOW	A piece the shop stops selling vanishes from the bag with no explanation	Orders	Customers	Fixed
18	LOW	An installed shop does not notice a new version	PWA	Customers	Fixed
19	LOW	An order with no pieces can be placed	Orders	Owner and customers	Fixed
20	LOW	A cancelled order is handed back as "already placed"	Orders	Customers	Fixed
21	LOW	Two cancel-and-refund requests at once put the pieces back twice	Stock	Owner	Fixed
22	LOW	A refund typed as 1.234,56 is recorded as 1.23	Money	Owner and customers	Fixed
23	LOW	A bank transfer that arrives after the order cancelled itself cannot be recorded	Money	Owner	Left to owner
24	LOW	The dashboard and the reports give different totals under the same label	Money	Owner	Left to owner
25	LOW	Whether an email address has ordered before can be found out	Data	Customers	Left to owner
26	LOW	On a new database, whoever signs up first with the owner's address becomes the owner	Access	Owner	Left to owner

27 **CRITICAL** Any signed-in account can open any guest's order by its number

An order opens for the holder of its link, for the account that placed it, and for the owner. For an order placed without an account the second check has nothing to compare, and the shop treated "cannot tell" as "allowed". Anyone who made an account, which takes a minute and no confirmation, could open any such order by its number and read the name, email, phone and address. Order numbers run in sequence.

A brand-new account opens the address of an order someone else placed without an account

AS RECEIVED

The order page opens: name, street, city, email, phone, what was bought. Counting down 60 order numbers opened 29 orders, every guest order among them; about four in ten of the shop's orders are guest orders.

AFTER THE FIX

"We couldn't open this order."

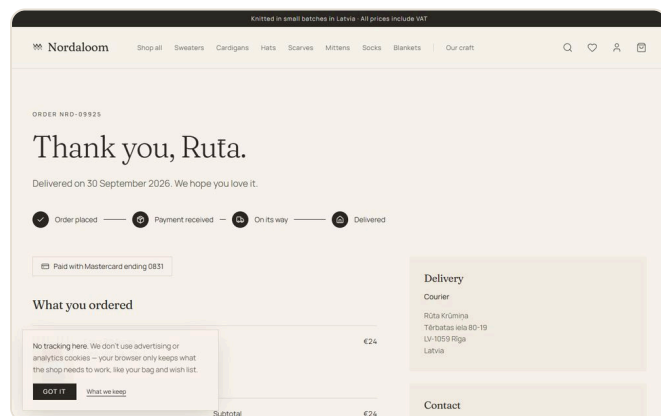
HOW TO REPRODUCE

1. Sign up with any email address.
2. Open the order page of a guest's order with only its number in the address.
3. Do the same for the numbers before it.

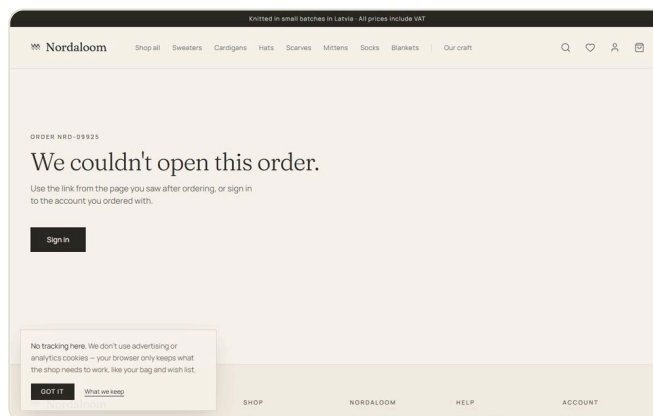
THE FIX

Each of the three checks is now either true or not; an order with no account never passes the account check. The page's data also no longer carries the keys the shop uses to recognise a repeated order.

Test: database test: a stranger's account asks for a guest's order by number, with no key, with a made-up key, and for the 40 numbers before it. Failed before the fix, passes after. **Fixed**. Data · affects: Customers



AS RECEIVED



AFTER THE FIX

1 **HIGH** After a dropped connection, the next order on the same device shows someone else's order

The checkout keeps a key on the device for 12 hours, so that an order is not made twice when the connection drops. The shop answered any order that carried the key with the first order made under it, without looking at who was ordering or what was in the bag. The next customer saw "Thank you, your order is placed" with the first customer's name, street, phone and email; their own bag was emptied and nothing was ordered for them.

Two customers on one device; the first one's answer was lost on the way back

AS RECEIVED

The second customer pressed "Place order · €53.99" and was shown the first customer's order for €29.99, with that customer's address. Their own two pieces were never ordered.

AFTER THE FIX

The second customer is told only that an earlier attempt from this device went through, with none of its details, and gets their own order for €53.99 on pressing again. The first order stays the first customer's.

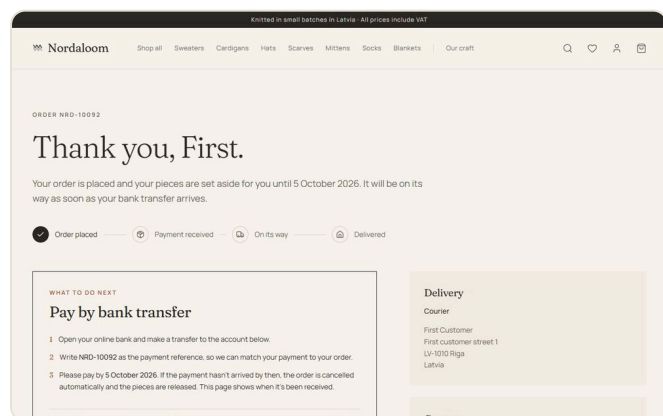
HOW TO REPRODUCE

1. Fill the checkout as one customer and press Place order; let the request reach the shop and drop the answer.
2. On the same device put other pieces in the bag, fill the checkout as another person and press Place order.
3. Look at the page that opens and at the orders in the admin.

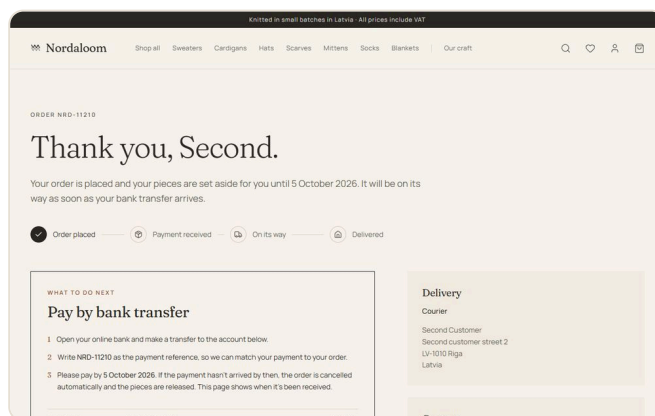
THE FIX

The key now stands for one exact order by one person: the same order sent again is returned as before, anything else is a new order. After a lost answer, before anything changed is sent, the checkout asks the shop whether the earlier attempt went through: someone ordering with the same email address is shown that order, anyone else is told only that an earlier attempt from the device went through.

Test: database test and browser test: the same key with another bag and another person. Failed before the fix, passes after.
Fixed. Orders · affects: Customers



AS RECEIVED



AFTER THE FIX

2

HIGH

A refund of any size can be recorded against a return

When the owner marks a return as refunded, the amount typed was checked only for being a number. One slipped digit or decimal point went straight into the order, the dashboard, the reports and the email that tells the customer how much is on its way. A recorded refund cannot be corrected afterwards.

A return of one pair of socks from an order of €29.99; 500 typed in the amount field

AS RECEIVED

Accepted. The order shows €500 refunded, the customer is emailed "Your refund of €500 is on its way", and the month's refunds grow by €500.

AFTER THE FIX

Refused: "That's more than is left to refund on this order (€29.99)."

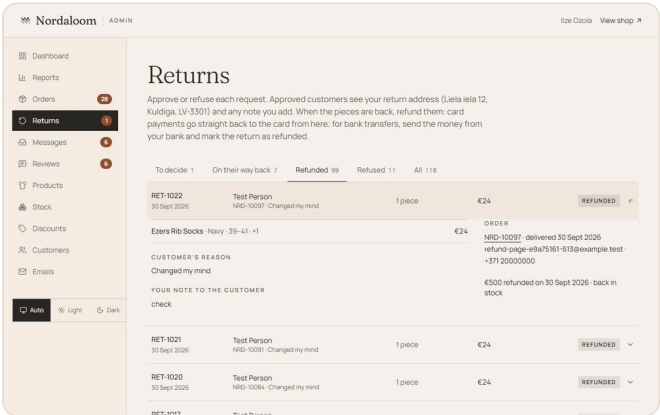
HOW TO REPRODUCE

1. As the owner, approve a return.
2. On Returns, type 500 as the amount and press "Mark as refunded".

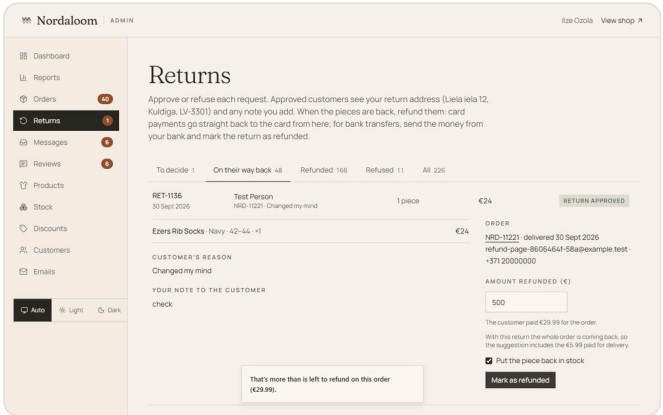
THE FIX

The database refuses a refund larger than what is left of what the customer paid; the page says how much that is before anything is sent.

Test: database test (a refund above what was paid, and two refunds at the same moment) and browser test on the Returns page. Failed before the fix, passes after. **Fixed**. Money · affects: Owner and customers



AS RECEIVED



AFTER THE FIX

3 HIGH A customer can pay by card for an order the shop has already cancelled, and is not refunded

When the owner cancelled an order that was waiting for a card payment, the payment page the customer had open at Stripe stayed open. If the customer then paid, the card was charged, the shop showed the order as cancelled and nothing was returned. Where the shop did try to return such a payment and the first attempt failed, the repeat was answered "already paid" and the money was never sent back.

Order NRD-10008 for €27.99, cancelled by the owner while the customer was on the payment page (Stripe in test mode)

AS RECEIVED

The customer paid €27.99. The shop showed "This order was cancelled", recorded no payment and refunded nothing.

AFTER THE FIX

Cancelling closes the payment page first, so it can no longer be paid. The €27.99 already taken was sent back when the customer's page was opened again, and the customer got an email saying so.

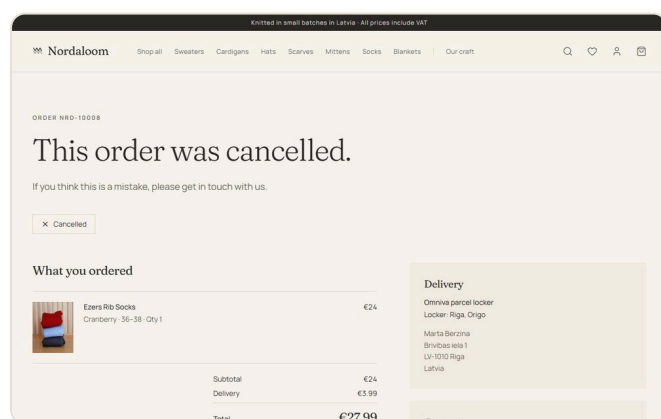
HOW TO REPRODUCE

1. As a customer, go to the card payment page for an order and stop there.
2. As the owner, cancel that order in the admin.
3. As the customer, pay on the page that is still open.

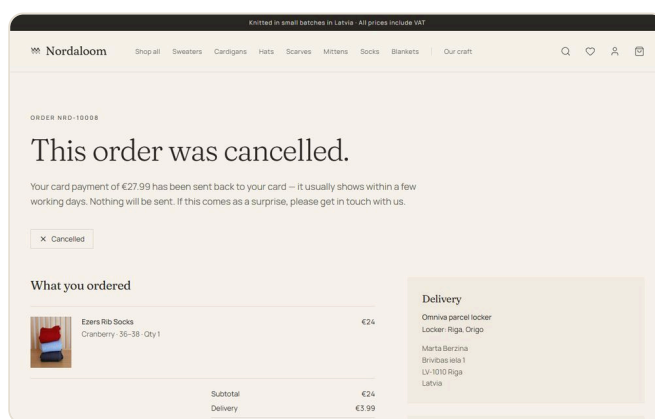
THE FIX

A card order is cancelled only after Stripe has confirmed its payment page is closed. A payment that still arrives for a cancelled order is refunded, again on every later notice or visit until the refund has gone through, and the customer is told by email.

Test: payment-code tests against a stand-in for Stripe (page left open, payment at the same moment as the cancellation, a failed refund repeated); the scenario was also run once in a browser against Stripe in test mode, before and after the fix. Failed before the fix, passes after. **Fixed**. Money · affects: Customers



AS RECEIVED



AFTER THE FIX

The other findings

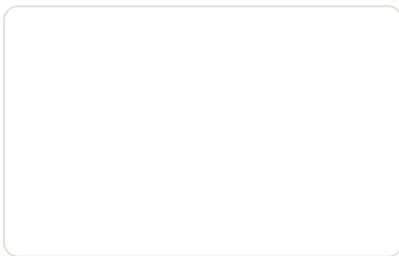
4 **MEDIUM** Without a connection the installed shop is an empty page on some hosts

WHAT HAPPENS

The shop saves its files on the phone so that it opens without a connection. On a host that answers with a "Vary: Origin" header, the saved files were never found again: the page asks for its script in a slightly different way than it was saved. The customer saw a blank white page, with no message. The builder's own way of serving the built shop behaves like this.

HOW TO REPRODUCE

1. Build the shop and serve the build.
2. Open the shop and one more page, then switch the connection off.
3. Reload.



AS RECEIVED

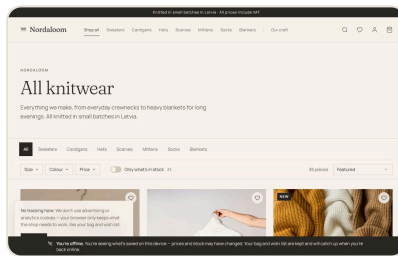
The built shop, opened once, then opened again with no connection

AS RECEIVED

A blank page: no catalogue, no bag, no "you're offline" note.

AFTER THE FIX

The saved shop opens, with the offline note at the bottom.



AFTER THE FIX

THE FIX

Saved files are matched by their address alone, and each build saves its complete set of files when it is installed.

Test: installed-app test: open, go offline, reload, on a host that answers with that header. Failed before the fix, passes after. **Fixed**. Offline - affects: Customers

5 **MEDIUM** After signing out, the device still opens the customer's order

WHAT HAPPENS

The shop kept a link back into each order on the device, with the order's access key, and copies of the customer's bag and wish list. Signing out removed none of them. The next person on a shared phone or tablet could open the order page and read the name, street, email and phone.

HOW TO REPRODUCE

1. Sign in, place an order, sign out.
2. Open the order's address again on the same device.

A customer orders, signs out; the next person opens the order's address on the same device

AS RECEIVED

The order page opens with the first customer's name, street, email and phone.

AFTER THE FIX

"We couldn't open this order." Nothing of the customer's is left on the device.

THE FIX

Signing out, in any tab or when the sign-in runs out, removes the links into orders, the copies of the bag and wish list and the saved pages; a signed-in customer's order link is no longer kept on the device at all.

Test: browser test (the order after sign-out) and installed-app test (everything the device holds after sign-out). Failed before the fix, passes after. **Fixed**. Data - affects: Customers

6 **MEDIUM** Anyone can place any number of unpaid orders, to any email address

WHAT HAPPENS

An order by bank transfer needs no account and no payment. Each one sends an email with the shop's bank details to whatever address was typed, and holds its pieces for five days. Nothing limited how many, or how large: a script could empty the shelf and send the shop's emails to strangers.

HOW TO REPRODUCE

1. Without signing in, call the order function ten times with ten email addresses.

Ten orders in a row from one visitor, each to a different address

AS RECEIVED

All ten placed: ten emails with bank details sent, ten pieces held for five days. One order could take every piece on sale: 2,353 pieces, €394,322 (tried inside a transaction and undone).

AFTER THE FIX

At most three orders waiting for a transfer per email address, five per account, 40 pieces per order, and eight unpaid orders a day per visitor once that limit is switched on for the host.

THE FIX

Limits per email address, per account, per order and per visitor. The per-visitor limit needs one setting on the live host and is off until it is set, because a wrong value would refuse honest customers.

Test: database tests for each limit, including that a stranger cannot use the limits to stop a customer from ordering. Failed before the fix, passes after. **Fixed**. Stock · affects: Owner

7 **MEDIUM** The order can be made at a different total than the button showed

WHAT HAPPENS

The shop prices every order itself, which is right, but it never compared its total with the one on the button the customer pressed. If a price, a code or a delivery charge changed while the checkout was open and the page had not caught up, the order was made at the new amount without a word.

HOW TO REPRODUCE

1. Open the checkout with live updates not arriving (a sleeping phone).
2. Raise the price of the piece in the admin.
3. Press Place order.

Button: "Place order · €29.99"; the owner raised the price by €4 a moment before

AS RECEIVED

An order for €33.99, first seen on the confirmation page.

AFTER THE FIX

Nothing ordered. "A price changed while you were checking out: your order now comes to €33.99, not €29.99"; the customer presses again to agree.

THE FIX

The checkout sends the total it shows; the shop refuses to order at any other, returns its own, and the page shows the new total and asks again.

Test: database test and browser test with live updates blocked. Failed before the fix, passes after. **Fixed**. Money · affects: Customers

8 **MEDIUM** An order returned in two parts is offered its delivery charge twice

WHAT HAPPENS

When a whole order comes back, the delivery charge is refunded too. The Returns page counted "the whole order" across all returns of the order, including ones not yet decided, and added the delivery charge to the suggestion for each of them.

HOW TO REPRODUCE

1. Return the two pieces of an order as two separate requests and approve both.
2. Read the suggested amounts on the Returns page.

Two pairs of socks at €24, delivery €5.99, sent back as two returns

AS RECEIVED

Suggested refunds:
€29.99 and €29.99,
€59.98 on an order of
€53.99.

AFTER THE FIX

€24 for the first return,
€29.99 for the one that
completes the order:
€53.99.

THE FIX

Delivery is suggested only with the return that completes the order, and no suggestion exceeds what is left of what was paid.

Test: database test (two refunds cannot exceed what was paid) and browser test of the suggested amounts. Failed before the fix, passes after. **Fixed.** Money · affects: Owner

9 **MEDIUM** A discount code from the made-up history gives real customers 20% off

WHAT HAPPENS

The shop was loaded with six months of made-up trading for showing it. One of that history's codes, THANKYOU20, was left active with no end date, so any new email address could use it. One real use also made the documented "remove the history before opening" command stop with an error.

HOW TO REPRODUCE

1. Type THANKYOU20 at checkout as a new customer.
2. Run the command that removes the made-up history.

THANKYOU20 typed by a new customer on a €24 pair of socks

AS RECEIVED

€4.80 off, for every new
address; afterwards the
history can no longer be
removed.

AFTER THE FIX

Refused. The history's
codes are paused, and
removing the history
keeps a code a real order
has used instead of
stopping.

THE FIX

The history's codes are created paused, and the removal no longer stops on a code in use.

Test: database tests: the code is refused for a real customer; the history is removed although a real order used one of its codes. Failed before the fix, passes after. **Fixed.** Money · affects: Owner

10 MEDIUM When the shop cancels an unpaid order, the customer is not told

WHAT HAPPENS

A customer who chose bank transfer holds an email with the bank details, the amount and a pay-by date. When the owner cancelled the order, no email went out. The customer could still transfer money for an order that no longer existed.

HOW TO REPRODUCE

1. Place an order by bank transfer.
2. Cancel it in the admin and look in the mailbox.

An order for €29.99 by bank transfer, cancelled by the owner

AS RECEIVED

The only email the customer has is "how to pay".

AFTER THE FIX

An email: "We've cancelled your order. Please don't send the payment for it."

THE FIX

Every cancellation is emailed. The email says who cancelled and what happens to any money already paid.

Test: database test: the cancellation puts an email in the outbox, and its text tells the customer not to pay. Failed before the fix, passes after. **Fixed**. Orders · affects: Customers

11 MEDIUM Dates and deadlines change with the clock of the device

WHAT HAPPENS

The pages showed dates in the time zone of whoever was looking, the emails in Latvian time. A customer abroad could read a payment deadline one day later than the moment the order cancels itself, and the page and the email disagreed.

HOW TO REPRODUCE

1. Open an order page in a browser set to Tokyo time.
2. Compare the date with the email.

A payment deadline at 23:30 Latvian time on 3 October, read on a phone set to Tokyo time

AS RECEIVED

"Please pay by 4 October 2026"; the order cancels itself on the 3rd.

AFTER THE FIX

"Please pay by 3 October 2026"; the same as in the email.

THE FIX

Dates and deadlines are shown in the shop's time on every device; the last day of a discount code is a Latvian day, also on the nights the clocks change.

Test: browser tests with the device set to Tokyo time: the customer's deadline and the owner's order time. Failed before the fix, passes after. **Fixed**. Orders · affects: Owner and customers

12 **MEDIUM** The bag is wrong without a connection

WHAT HAPPENS

A piece added offline was counted but not listed or priced. After the app was started offline the bag read "Loading your bag..." with a total of €0. For a signed-in customer, a piece added in the first seconds of an offline start disappeared.

HOW TO REPRODUCE

1. Put a piece in the bag, switch the connection off, add another piece and open the bag.
2. Restart the app without a connection.

One pair of socks in the bag (€24); a second pair added without a connection

AS RECEIVED

"2 pieces", one line, subtotal €24. After restarting the app offline: "Loading your bag...", €0.

AFTER THE FIX

Both lines, subtotal €48, also after a restart; nothing added is lost.

THE FIX

The bag lists and prices pieces from what the device has already saved, shows its own copy at once, and keeps whatever is changed before the account's copy arrives.

Test: three installed-app tests: a piece added offline, the bag after an offline restart, a piece added straight after an offline start. Failed before the fix, passes after. **Fixed**. Offline · affects: Customers

13 **LOW** The customer page counts refunded money as spent

WHAT HAPPENS

In the admin, a customer's "Spent" and the customer list added up paid orders and never took refunds off.

HOW TO REPRODUCE

1. Refund a customer's only order and open the customer in the admin.

A customer whose only order, €29.99, was returned and refunded

AS RECEIVED

Spent €29.99.

AFTER THE FIX

Spent €0 · Refunded €29.99.

THE FIX

Spent is what was paid less what was refunded; a Refunded figure is shown beside it.

Test: database test of the customer list and browser test of the customer page. Failed before the fix, passes after. **Fixed**. Money · affects: Owner

14 **LOW** The shop's settings can be read by anyone, including the welcome discount code

WHAT HAPPENS

The settings row was readable in full with the public key that every visitor's browser holds.

HOW TO REPRODUCE

1. Read the settings table with the public key.

One request with the shop's public key

AS RECEIVED

Returns the 10% welcome code without subscribing to the newsletter, and the internal address of the server functions.

AFTER THE FIX

Only the six values the shop pages show.

THE FIX

Visitors can read only the columns the pages need; the owner's pages read the rest through a function that checks who is asking.

Test: database tests: the hidden values are refused, the public ones still load, the owner's page still gets all of them. Failed before the fix, passes after. **Fixed**. Access · affects: Owner

15 **LOW** "We couldn't reach the shop" when the order was in fact placed

WHAT HAPPENS

When the answer to an order was lost, the message said the shop could not be reached. The order existed, held its pieces and had sent its email; a customer who did not press again never learned of it.

HOW TO REPRODUCE

1. Press Place order and drop the answer on its way back.

The order reaches the shop, the answer is lost

AS RECEIVED

"We couldn't reach the shop – the connection dropped."

AFTER THE FIX

"We couldn't confirm your order – the connection dropped, and it may already have reached us."
After a reload the page says which order went through.

THE FIX

The message says the order may have gone through, and the checkout finds out and says so the next time it is opened.

Test: browser test: the message after a lost answer. Failed before the fix, passes after. **Fixed**. Orders · affects: Customers

16 **LOW** The return form shows the list price, not what will come back

WHAT HAPPENS

A customer who bought with a discount code was shown the full price of the pieces on the return form, and learned the real amount only after the refund.

HOW TO REPRODUCE

1. Order with a discount code, have the order delivered, open "Request a return" and tick a piece.

One €24 pair of socks bought with a 15% code

AS RECEIVED

"Value of pieces €24".

AFTER THE FIX

"You paid for these pieces €20.40".

THE FIX

The form shows what was paid for the chosen pieces, the same sum the owner is shown when refunding.

Test: browser test of the return form on a discounted order. Failed before the fix, passes after. **Fixed**. Money · affects: Customers

17 **LOW** A piece the shop stops selling vanishes from the bag with no explanation

WHAT HAPPENS

If the owner hid a product while a customer had it at the checkout, pressing the button gave "Your bag is empty" and nothing said why.

HOW TO REPRODUCE

1. Open the checkout with a piece in the bag.
2. Hide the product in the admin and press Place order.

A pair of socks at an open checkout; the owner switches off "Show in the shop"

AS RECEIVED

"Your bag is empty. Browse the shop."

AFTER THE FIX

The shop names the piece: "... is no longer sold, so we've taken it out of your bag."

THE FIX

The shop names the piece it took out and says why.

Test: browser test: the product is hidden while the checkout is open. Failed before the fix, passes after. **Fixed**. Orders · affects: Customers

18 **LOW** An installed shop does not notice a new version

WHAT HAPPENS

The part of the app that saves files on the phone carried a version number set by hand. A new build did not change it, so an open app was never told, files of old versions were never removed, and on a slow connection the old saved page kept coming back.

HOW TO REPRODUCE

1. Open the built shop, publish a second build, ask the app to look for an update.

A new build is published while the shop is open on a phone

AS RECEIVED

No "new version" message; on a slow connection the old version opens again and again.

AFTER THE FIX

"A new version of the shop is ready – Refresh"; the next start opens the new one and the old files are removed.

THE FIX

Every build is a new version with its own list of files; the phone saves the whole new build, then drops the old one and offers to refresh.

Test: installed-app tests with two real builds served in turn: the message, the files kept afterwards, a slow connection. Failed before the fix, passes after. **Fixed**. PWA · affects: Customers

19 **LOW** An order with no pieces can be placed

WHAT HAPPENS

Sent straight to the shop's database with no list of pieces, an order was accepted: nothing in it, the delivery charge as its total, and a "how to pay" email.

HOW TO REPRODUCE

1. Call the order function with the list of pieces left out.

An order request with no list of pieces

AS RECEIVED

Order placed: 0 pieces, total €5.99.

AFTER THE FIX

Refused like an empty bag.

THE FIX

A missing list is treated as an empty bag.

Test: database test: an order without a list of pieces. Failed before the fix, passes after. **Fixed**. Orders · affects: Owner and customers

20 **LOW** A cancelled order is handed back as "already placed"

WHAT HAPPENS

If an order had been cancelled and the customer pressed the button again on a checkout that was still open, the shop answered with the cancelled order as if it had just been placed, and the page emptied the bag.

HOW TO REPRODUCE

1. Place an order, cancel it in the admin, send the same order again from the same checkout.

An order cancelled by the shop; the customer presses Place order again on the open checkout

AS RECEIVED

The cancelled order comes back as placed; the bag is emptied and nothing is ordered.

AFTER THE FIX

A new order is made.

THE FIX

A cancelled order is never returned as placed.

Test: database test: the same order sent again after it was cancelled. Failed before the fix, passes after. **Fixed**. Orders · affects: Customers

21 **LOW** Two cancel-and-refund requests at once put the pieces back twice

WHAT HAPPENS

Cancelling a paid card order returns the money and puts its pieces back on the shelf. When two such requests for one order met, the money went back once but the pieces were added twice, so the shop showed stock it did not have.

HOW TO REPRODUCE

1. Send two cancel-and-refund requests for one paid card order so that the second starts before the first has finished.

A paid order of two pieces, cancelled twice at the same moment; 14 on the shelf before the order

AS RECEIVED

16 on the shelf.

AFTER THE FIX

14.

THE FIX

The order is locked while it is cancelled, so the second request finds it already cancelled.

Test: payment-code test: two cancellations of one order overlapping in the database. Failed before the fix, passes after. **Fixed.**
Stock · affects: Owner

22 **LOW** A refund typed as 1.234,56 is recorded as 1.23

WHAT HAPPENS

The refund field accepted an amount written with a thousands separator and read only its beginning. The return was marked refunded with the wrong amount, and the customer's email named it.

HOW TO REPRODUCE

1. On Returns, type 1.234,56 and press "Mark as refunded".

"1.234,56" typed as the refund amount

AS RECEIVED

Recorded, and emailed to the customer, as a refund of €1.23.

AFTER THE FIX

"Please type the amount as a plain number, like 29.99."

THE FIX

Only a plain amount is accepted.

Test: browser test of the refund field. Failed before the fix, passes after. **Fixed.** Money · affects: Owner and customers

23 **LOW** A bank transfer that arrives after the order cancelled itself cannot be recorded

WHAT HAPPENS

An unpaid order cancels itself after five days and its pieces go back on sale. If the transfer lands a day later, the admin offers no way to put the payment against the order.

HOW TO REPRODUCE

1. Let an order pass its payment deadline, then try to mark it paid.

A customer pays on the evening of the fifth day; the transfer arrives on the sixth

AS RECEIVED

The shop shows the order as cancelled, with the money in the bank.

LEFT AS IT IS

Left as it is: whether to make the order again or send the money back is the owner's decision each time.

LEFT TO THE OWNER

It is described in the shop's notes for the owner.

Check: seen in the code; no automated test. **Left to the owner.**
Money · affects: Owner

24 LOW The dashboard and the reports give different totals under the same label

WHAT HAPPENS

"Refunded, last 30 days" counts 30 × 24 hours back on the dashboard and calendar days in the reports. In the reports, one block counts orders that were paid and later cancelled as paid, and the headline leaves them out.

HOW TO REPRODUCE

1. Compare "Last 30 days" on the dashboard and in the reports.

Refunds for the last 30 days

AS RECEIVED

Dashboard €5,619.79, Reports €5,483.80. The difference is one refund of €135.99 made on the evening of the day before the period.

LEFT AS IT IS

Left as it is: both figures are correct for their own period; the owner chooses which definition the labels should follow.

LEFT TO THE OWNER

Nothing was changed in the app: the choice is the owner's.

Check: seen in a browser and checked against the database; no automated test. **Left to the owner.** Money · affects: Owner

25 LOW Whether an email address has ordered before can be found out

WHAT HAPPENS

A first-order discount code answers differently for an address that has ordered, and signing up with an existing address says it is taken. One address at a time, anyone can learn who is a customer.

HOW TO REPRODUCE

1. Check a first-order code with someone else's email address.

The welcome code checked for an address that has ordered

AS RECEIVED

"Not your first order", which tells the asker the address belongs to a customer.

LEFT AS IT IS

Left as it is: closing it means changing how first-order codes and sign-up behave, which is the owner's call.

LEFT TO THE OWNER

Nothing was changed in the app: the choice is the owner's.

Check: run against the shop's public interface; no automated test. **Left to the owner.** Data · affects: Customers

26

LOW

On a new database, whoever signs up first with the owner's address becomes the owner

WHAT HAPPENS

The owner's role is given to the account with the owner's email address, and email confirmation is switched off. On a database where that account does not exist yet, anyone can create it.

HOW TO REPRODUCE

1. On a fresh database, sign up with the owner's address.

A fresh copy of the shop, before the owner has signed up

AS RECEIVED

The first person to sign up with the owner's address gets the admin.

LEFT AS IT IS

Left as it is, with a note: create the owner's account before the shop is opened, and switch email confirmation on.

LEFT TO THE OWNER

None in the code; it is a step for going live.

Check: read in the code and the settings; on the running shop the owner's account exists and the attempt was refused. **Left to the owner.** Access · affects: Owner

Checked and fine

- Other customers' data, apart from the guest orders in finding 27: accounts' orders, addresses, returns, reviews in waiting, emails and messages could not be read or changed from outside or from another account.
- Nothing is written without signing in except what should be: an order, a newsletter sign-up, a contact message.
- No secret key in the page or in the built files.
- Money: for all 2,059 orders the total equals pieces less discount plus delivery, and VAT is right to the cent; discount rules (first order, once per customer, minimum, last day, limit) hold.
- Stock: the last piece cannot be sold twice; two orders at once for it give one order.
- A double press makes one order; the same order sent again after a lost answer makes one order.
- Reports and dashboard: September recounted independently matches both; the spreadsheet export matches the page.
- Payments: forged or unsigned payment notices are refused; only the shop's own server code can mark an order paid.
- Phone screen: no page scrolls sideways at 390 × 844; the app's icons and settings for installing are in place.
- Offline: an order is never sent without a connection, a guest's bag is kept, and nothing from the admin is saved on the device.

Left to the owner

- The limit on unpaid orders per visitor is off until one setting says how many proxies stand in front of the shop on its host; a wrong value would refuse honest customers, so it is set after looking at the live host. The limits slow abuse down; a CAPTCHA on guest checkout is the next step if it happens anyway.
- A guest whose email address others have typed into three unpaid orders cannot order by bank transfer until those are paid or cancelled; they can pay by card or sign in.

- The Returns page reads the whole list at once; after about 1,000 returns its counts will stop growing. Read in the code, not run.
- "Payment received", "Mark as shipped" and "Mark as delivered" act on one click and cannot be undone.
- An email can be sent twice if marking it as sent fails; a card order paid with a wrong amount stays open until looked at by hand. Both were read in the code, not run.
- The customer page still counts returned pieces under "Pieces bought".
- Installing on a real phone, the iPhone in particular, was not tried; the checks ran in a desktop browser at phone size.

How it was checked

- A copy of the shop on a local Supabase with test accounts and the builder's six months of made-up trading; card payments against Stripe in test mode.
- The shop was checked the way someone who wants to misuse it would: from outside with only its public key, through its code and database, as the owner and customers in a browser, and installed on a phone and taken offline.
- Every finding became a test that fails on the shop as received. After each fix all tests ran again.
- Each fix was reviewed independently before the next one, until a review of the last fix found nothing above low.

Limits

- One shop, built once. The numbers describe this shop, not a rate for apps built this way.
- The report covers the shop as it was received. The reviews of the fixes found more, part of it brought by the fixes themselves; those rounds are in the full record and not in these numbers. The work stopped at the round whose review found nothing above low; its low points are listed for the owner. The test counts above are the tests that were run on the shop as received; 33 more were written for the later rounds.
- Finding 27 and four of the low ones were found during the reviews of the fixes, although they were in the shop from the start. Two more defects of the shop as received were found and fixed then without a run on the original, and are not counted: a second card payment for an order that is already paid was not sent back, and a bag changed offline on one device removed pieces added on another.
- Stripe was used in test mode only. The shop was not put on a public host; two of the fixes (the limit per visitor and the offline files) depend on how that host answers and should be looked at there.

Sample engagement: the shop is fictional and holds test data; it was built with Claude Code for this case; every finding, test and fix is a real run.