SAMPLE DELIVERABLE - READ THIS FIRST

This is a real diagnostic, run on a real store, with every identifying detail removed. It is published as a sample so you can see exactly what you are buying before you buy it.

WHAT IT COST TO PRODUCE: about one hour. No account on their store, no password, no paid tools. Everything in it came off pages anybody can open, which is the point - you find out what is wrong before you hand anyone access to anything.

WHAT THE CLIENT GOT: the document below, plus a written structure spec and a quoted fix. The diagnostic is the part that is priced up front. The fix is quoted after, because nobody can price a repair before they have looked at it.

Every number in here is checkable. Where a number is a ceiling rather than an exact figure, it says so. Where a finding is a strong explanation rather than proof, it says that too.

SAMPLE: CATALOG DIAGNOSTIC A real diagnostic, client details removed | Fix My Sh*t by TraciUnfiltered

THE SHORT VERSION

There are two separate problems and they are not the same size.

The first is a live bug. A script on your site, header.js, throws an error every time a page loads. Your year, make and model Search button is wired to run JavaScript rather than to go to an address, and when I click it nothing happens at all - not a filtered page, not even a request leaving the browser. Those two things sit right next to each other and they probably explain one another. Your developer can act on this today with the error message printed on page five.

The second is the data underneath. Of your 666 products, at most 41 carry any year value your sidebar Year filter can read. That is about six percent. And the choices that filter offers are not years at all - they are ranges somebody typed, like 11-16, 13-18 and 19+, two of which are not years but intercooler outlet sizes.

None of this is a mistake you made. Shopify does not support year, make and model fitment natively, so fitment ends up spread across tags, titles and variant options, and the filters can only work with what they find. It is fixable, and it is a great deal cheaper to fix at 666 products than at three thousand.

HOW I CHECKED

Everything below came off your public storefront on Monday, September 14, 2026. No account access and no paid tools. Four sources: your product feed at the store's public product feed; the individual product record for every product I name, pulled one at a time so I am quoting stored values rather than a summary; your live collection and filter pages read directly; and the browser's own console and network activity.

Three things I checked so the numbers below are not guesses. The filter lists have no "show more" control hiding extra values, so the counts are the complete lists. The filter counts come off the page itself. The 666 total is printed twice on the page.

What I cannot give you from outside is how many of the 666 have wrong or missing data. I can prove the pattern and I can count what the filters see. Counting the errors is a store-access job, not a guess I am willing to make.

FINDING 1. ALMOST NOTHING IS IN THE YEAR FILTER

This is the headline and most of the rest is detail.

Your sidebar Year filter prints a count next to each choice. Added up, every year value together covers 41 product slots. That 41 is generous, because a product carrying two year values is counted twice - the transmission deep pan kit carries both 13-18 and 19+, so it is in there twice by itself.

So at most 41 products out of 666 can be found through that filter. Call it six percent. The other 625 are invisible to it, permanently, no matter which app is running it.

To be precise about what that does and does not cover: this is the sidebar Year filter. The year, make and model search at the top of the page is a different piece of software with its own vehicle data, and I cannot see its coverage from outside your admin. It may reach products this filter does not. But it is also the thing that does not currently return a result, which is Finding 3.

FINDING 2. THE YEAR FILTER OFFERS RANGES SOMEBODY TYPED, NOT YEARS

These are the exact choices in your Year filter, with each one's product count, read off the live page. This is the complete list:

03-07 (3), 03-09 (1), 07-09 (1), 07.5-09 (2), 10-12 (4), 11-16 (3), 11-22 (1), 13-18 (5), 17-19 (1), 17-22 (2), 17-23 (1), 99-16 (1), 2011-2016 (2), 2011-2019 (1), 2017-2019 (1), 2017-2026 (1), 2020-2022 (2), 2023-2025 (1), 19+ (2), 23+ (1), 19-CURRENT (3), 11-16 (3 Inch IC Oulet) (1), 17-25 (2.5 Inch IC Outlet) (1)

Four things are wrong in that list and they are all the same problem wearing different hats.

A customer cannot pick his own year. He owns a 2014. There is no 2014. He has to work out whether his truck counts as 11-16, or 13-18, or 11-22, or 2011-2016. That is your job, not his.

The same range is written twice, two different ways. 11-16 and 2011-2016 are the same trucks in two separate buckets. So are 17-19 and 2017-2019. Parts that fit the same truck sit in filter entries that never see each other.

Three spellings of open-ended. 19+, 23+ and 19-CURRENT. 19+ and 19-CURRENT mean the identical thing and both sit in the list as separate choices.

Two entries are not years at all. 11-16 (3 Inch IC Oulet) and 17-25 (2.5 Inch IC Outlet) are intercooler outlet sizes living inside the year filter. They do not even agree with each other on how to spell outlet - one of them says Oulet.

The filter is doing exactly what it was told. The instruction is the problem.

FINDING 3. THE YEAR, MAKE AND MODEL SEARCH IS BUILT CORRECTLY AND IT IS NOT FIRING

You have a proper year, make and model selector at the top of your collection pages, and it is set up better than I assumed before I looked. Year runs 1990 through 2026. Pick 2014 and the Make list fills in on its own. Pick Ford and the Model list fills in with F-250, F-350, F-450 and F-550 Super Duty. Somebody built that, and the vehicle data behind it is real.

Here is what testing it turned up, in order.

The Search control is a link whose address is javascript:void(0). That is normal for this kind of widget - it means the button does nothing by itself and relies on a script to attach the behavior.

I selected 2014, Ford, F-250 Super Duty and clicked it. The page did not change, the web address did not change, and - this is the useful part - the browser sent no network request at all. Not a failed one. None. So nothing is being asked and nothing is failing to answer. Nothing is listening to that button.

Then I loaded the page fresh, touched nothing, and looked at the browser console. There is a JavaScript error on every page load:

> TypeError: Cannot read properties of null (reading 'addEventListener') at header (/assets/header.js)

In plain English: that script is trying to attach a click handler to something on the page that is not there, and it crashes at that line. When a script crashes partway through, everything after that line never runs. The year, make and model widget lives in the header, and header.js is the header's script.

Say plainly what that is and is not. It is a real error, on every page load, with no input from me. It is a strong explanation for a button that nobody is listening to. It is NOT proof that this particular error is what kills that particular button - proving that needs someone inside the theme code. But it is exactly where I would have your developer start, and it is a twenty-minute look rather than a project.

One more thing, unrelated to the crash and worth two minutes on its own. Your Make list for 2014 reads: Chevrolet, Ford, RAM, Ram. RAM and Ram are two separate entries for the same manufacturer. A customer who picks Ram is looking at a different shelf than one who picks RAM, and neither of them knows it.

FINDING 4. YEAR IS STORED THREE WAYS AT ONCE

Across the products I read in full, year lives in three incompatible places.

In tags, as individual four-digit years. The 2011-2014 Ford F250/350 injectors (part number withheld) carry tags 2011, 2012, 2013, 2014.

As a variant option. The 2011-2019 6.7 Powerstroke intake piping kit has an option named Vehicle Year. The transmission deep pan and Thermal Bypass Install Kit has an option named YEAR. Shopify treats Vehicle Year and YEAR as two unrelated options, because they are spelled differently.

Nowhere at all. The intake piping kit has no tags. The transmission pan kit has no tags. The Intake Boot Kit for 1994-1997 Ford F250/F350, 7.3L Powerstroke has no tags.

Three storage locations means no single query can find every part that fits one truck. That is why this stays broken even on the products where the data does exist.

FINDING 5. THE YEAR DATA THAT EXISTS DOES NOT MATCH THE TITLES

Three examples, each pulled from that product's own record so you can check them one at a time.

Hot Side Intercooler Pipe for 2016-2026 Ford Powerstroke 6.7L is tagged 2015, 2016, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026. 2017 and 2018 are missing. 2015 is in there and the title says it should not be. A man with a 2017 filtering by year never sees this part.

Hot Side Intercooler Pipe for 2011-2016 Ford Powerstroke 6.7L is tagged 2011 through 2015. 2016 is missing.

Intake Boot Kit for 1994-1997 Ford F250/F350, 7.3L Powerstroke has no tags at all.

That same 2016-2026 pipe still carries the web address of its old title: its link reads hot-side-intercooler-pipe-for-2023-2024. The title got fixed and the address did not. Worth knowing before anybody rewrites links in bulk.

These are not carelessness. This is what happens when a person types year ranges into a tag field several hundred times. The error rate is arithmetic, not character. Which is also why the fix is not "be more careful next time."

FINDING 6. YEAR USED AS A VARIANT IS WHERE FINDING 2 COMES FROM

The 2011-2019 6.7 Powerstroke intake piping kit has three options: Color (Raw, Polished, Black), Air Intake Option (Standard, Big AF), and Vehicle Year (2011-2014, 2015-2016, 2017-2019). Three times two times three is 18 variants for one physical kit. The transmission pan kit's YEAR option holds 13-18 and 19+.

Now look at those option values and look back at the Year filter list in Finding 2. They are the same strings. The filter is not inventing those ranges. It is offering whatever got typed into a variant option box, because that is all it has.

That connects the two halves of this. Fitment is not a variant. A variant is something you stock, count and ship differently. A model year is something a customer either has or does not have. Using year as a variant inflates the SKU count, splits inventory and sales reporting across rows that are the same physical part, and then hands the filter a list of typed ranges instead of years.

FINDING 7. THREE SMALLER THINGS WORTH A LOOK

The 07.5-24 Ram 6.7L intake horn is listed at $0.00. I checked its record directly: one variant, Default Title, price 0.00. It is live on the site right now.

The Product type facet counts add to 659 against 666 products, so roughly seven products have no product type at all. Separately, Fuel Systems and Components holds 319 of the 666 - nearly half the store in one bucket, which will stop being a useful filter shortly. And A camshaft brand is listed as a product type, though it is a brand rather than a category.

Some product titles use typographic quote characters instead of plain ones. The 1.45 inch tappets use a double prime, and several titles from one turbo supplier mix straight and curly apostrophes inside the same title. Those characters break search matching and they corrupt when a CSV is opened in Excel and saved again. Cheap to normalize now, annoying later.

WHY THIS HAPPENED, PLAINLY

Fitment is many-to-many. One part fits many vehicles. One vehicle takes many parts. Shopify's product model cannot express that relationship, and there is no native year, make and model support to fall back on.

So every store in this position improvises: some fitment in tags, some in titles, some in variant options, and an app asked to make sense of the result. You did the normal thing. It stops scaling at exactly the point you are standing on.

WHAT I WOULD DO, IN ORDER

Phase 0, and it is not mine. Give your developer the header.js error above. If the year, make and model search starts working, you get a real improvement this week that costs you nothing and does not wait on any of the rest of this.

Phase 1. This document. Done.

Phase 2. The structure spec, in writing, before anything else is loaded. One real decision sits at the front of it: what does your buyer pick first. My read is engine, then year, then truck, because your customer types 6.7 Powerstroke, not 2016 Ford F-250, and your whole navigation is already built that way. Once that order is agreed, everything else follows mechanically.

Phase 3. One supplier converter. Take a single vendor's CSV, turn it into a file Shopify accepts with fitment in the right fields, and run it. One vendor first, so the mistakes are cheap.

Phase 4. The catalog load, with before-and-after numbers written down. The first number is already on the board: at most 41 of 666 findable through the sidebar year filter.

Phase 5. Fitment live. The filter works the way you described it. That is where the pilot ends.

Nothing bulk loads before Phase 2 is settled. Loading first and cleaning up after is how 666 products arrived in this shape, and three thousand would arrive in a worse one.

WHAT HAPPENS NEXT, in the real version

Phase 2 is the written structure spec - the one decision that everything else follows from. Phase 3 is one converter built against a single supplier file, so the mistakes are cheap. Phase 4 is the load, with before-and-after numbers written down. Phase 5 is live.

The diagnostic is $350 and it stands on its own. You can take this document to anybody. The fix is quoted separately once the diagnostic says what the fix actually is.

Fix My Sh*t

TraciUnfiltered

ducttapeworld.com