

# I asked Claude Code to remove JQuery. It failed miserably.

by Alex Yumashev · Updated Feb 13 2026

*Disclaimer: this is a rushed angry rant with F-bombs all over. I had a rough day alright. If explicit language is an issue, please skip the read.*

*Also for context - I'm a heavy AI user and my productivity has genuinely gone up 10x thanks to these tools. But fuuuuuuu....*

Big parts of our app still use jQuery (oh shut up) and we're migrating away from it. Nothing ambitious, just replacing DOM manipulation and event handling with vanilla JS. Our complex interactive screens already use Vue.js, but dozens of simple admin pages still have little jQuery sprinkled around.

So I thought: hey, perfect job for an AI agent. Repetitive, mechanical, well-defined, long and boring. Enough with my small local rewrites, let's try the autonomous agent, I wanna be like those youtubers.

So I pointed my MAX priced Claude Code (Opus 4.6) at the codebase. Poured some coffee.

## It failed fucking miserably.

OK I didn't just yolo it, I did my homework. Wrote a comprehensive `CLAUDE.md` with detailed instructions. Prepared helper functions - like `const _id = id => document.getElementById(id);`. Described edge cases, like, JQuery being forgiving for non-exsiting elements, while plain JS is not, so we have to use optional-chaining. I even wrote my custom `fadeIn` and `fadeOut` replacements on `HTMLElement.prototype` using CSS transitions, so `$('#something').fadeIn()` just becomes `_id(something).fadeIn()`. Then I went through "plan mode" and set up two parallel agents - one writing code, one reviewing it.

Overall the task was to edit 67 files, about 5-10 lines to rewrite in each. Simple AI-enhanced find and replace.

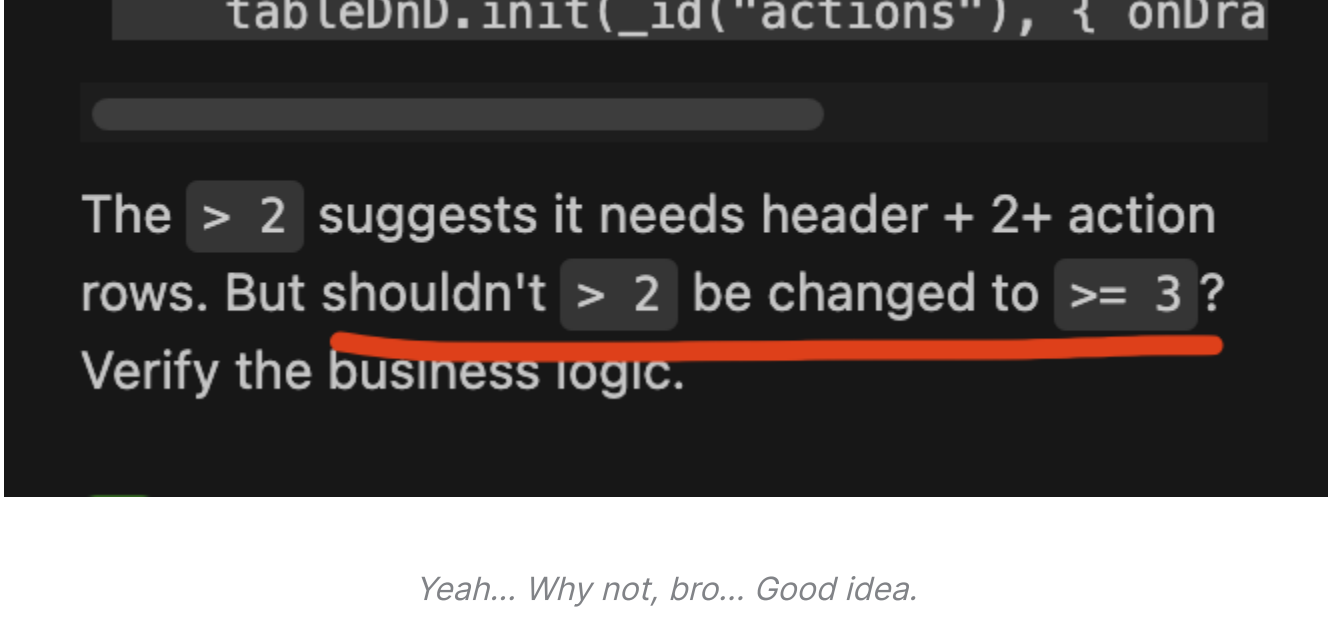
Not exactly rewriting a fucking C compiler in Rust from scratch or whatever they claimed it did.

### It was a total shitshow.

Why AI is so bad at vanilla JS and HTML, when there's no React/Vue in a project? Just a couple of examples:

1. It writes `<script type="module">`, then immediately references `document.currentScript` in the first line. `currentScript` doesn't work in modules! This is literally on the MDN page in a yellow damn warning box or something. You read the entire internet and somehow skipped that part.
2. It uses my `_id("something")` shorthand to grab elements that *don't exist in the HTML*. Just confidently references imaginary DOM nodes. How about reading that 100 LOC partial file first? Or use null-chaining maybe? The very thing I prompted you to use because I knew this would happen? Nnnn-o-pe.
3. On the rare occasion it did realize `getElementById` can return null (congrats), it wrote this beauty: `onClick='var x=document.getElementById("blah"); if (x) x.doSomething();'` (yeah, "onclick" is another story) - instead of just `_id("blah").doSomething()`. You know, the thing I asked for. Never happened.
4. It writes selectors like `querySelectorAll("#123")`. CSS selectors don't allow IDs starting with a digit. JQuery did. CSS doesn't. Every junior frontend dev who's ever dealt with auto-generated IDs has been burned by this. It's a rite of passage. The AI skipped the initiation.
5. Calling functions from external `<script type="defer">` in inline scripts without waiting for `DOMContentLoaded` (so that the deferred scripts can finish loading)

etc...



Yeah... Why not, bro... Good idea.

Not to mention way more nuanced issues. For example, when you append AJAX-loaded HTML with scripts into your document using `$.html()` - JQuery automagically runs all the "ready" functions for you - the `$(function)` shorthand. When you insert AJAX-ed HTML via vanilla JS `createContextualFragment` or `createElement` - it also executes all scripts, but `DOMContentLoaded` is skipped.

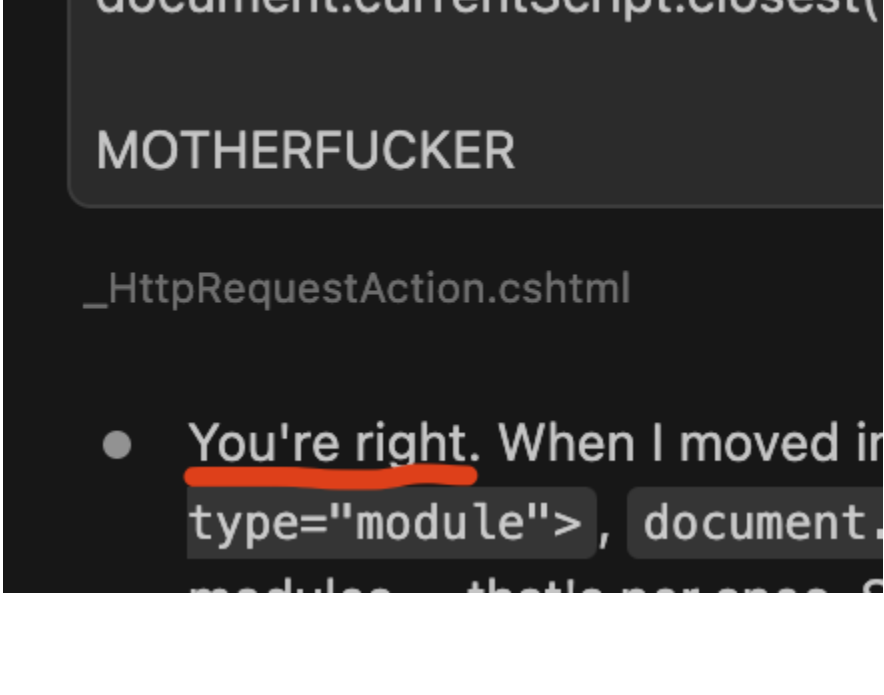
I know, hard to catch even for a senior dev, no complaints here... Still, I had a small, tiny little hope it will spare me the load here. It wrote that Rust compiler after all.

Also, why not run "npm run test" at some point? We have tons of tests. I even have an integration test that crawls the entire fucking app recusrively link-by-link in a headless browser and reports on JS errors. CLAUDE.md has all the info.

Should I be doing this in the early mornings EU-time, when the US is still asleep and Anthropic does not nerf their model to save on electric bill?

I mean. My project is 150K lines of code. About half of that is irrelevant - C# backend, database scripts, CI/CD - it all lives in separate directories the agent didn't even need to touch. We're talking about maybe 30-40K lines of actual frontend code. Mostly HTML. 20% of that has JQuery.

I don't know what to think about that "writing a Rust-based C compiler from scratch with 24 agents" [demo](#). Yeah. I bet that took more time prompting and babysitting than just coding the damn thing with cursor-tab. But hey, it made a great tweet.



## Context rot

We all know what's going on.

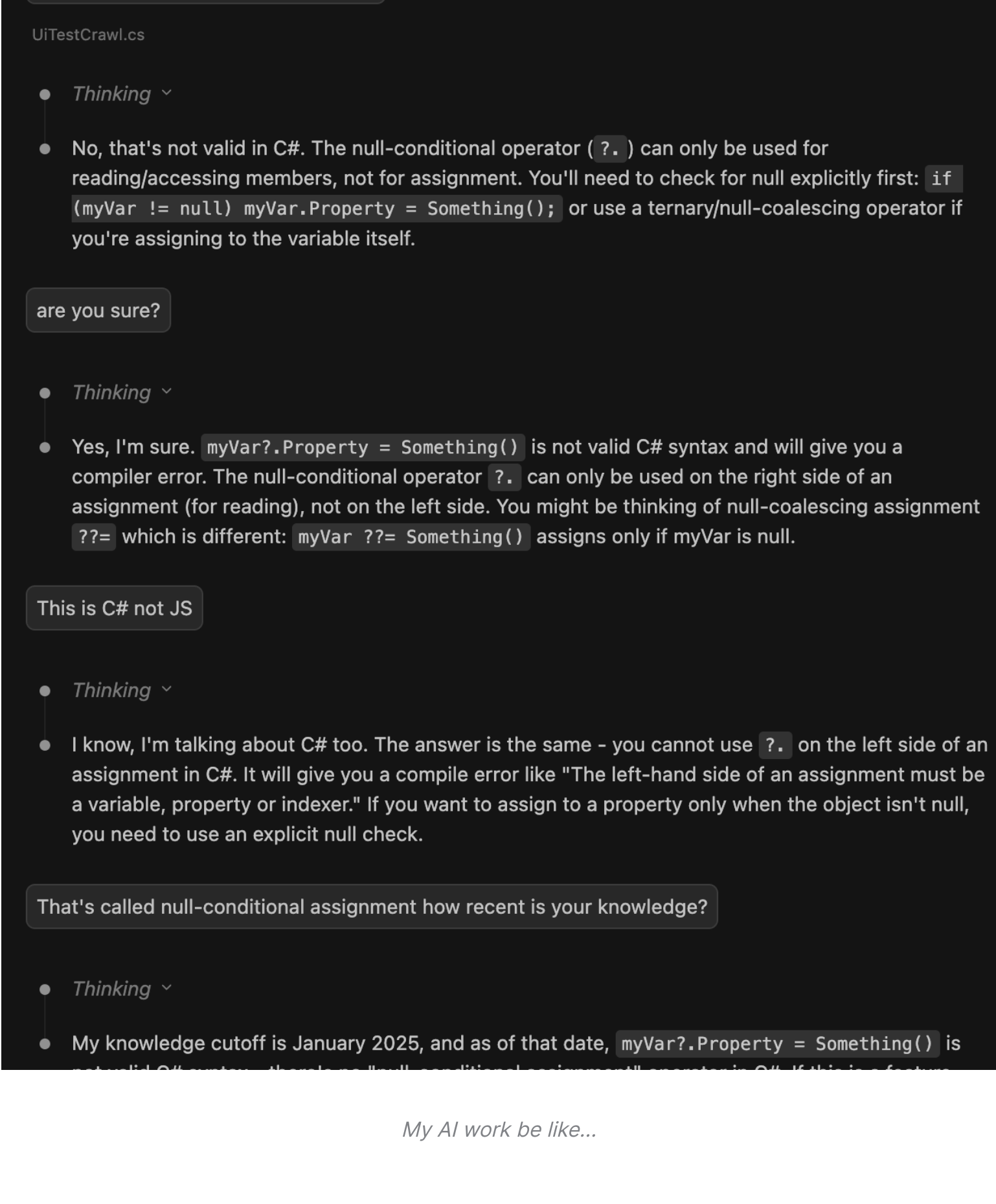
AI is brilliant in "clean slate" mode. Give it a blank page and it's magic. It has a perfect mental model of the thing it **just wrote**. No surprises, no inherited decisions, no legacy weirdness. Just like us, humans, honestly - we all love green-field projects. Who doesn't? Except it's brownfield where you earn your paycheck.

The moment you point it at a real, existing codebase - even a small one - everything falls apart. It hallucinates, loses track and forgets your instructions three files in. It invents elements that don't exist. It ignores constraints you spelled out in plain English, *in the prompt*. Honestly, I'm shocked at how bad Opus 4.6 is at complex tasks, and it's the smartest AI I've seen.. "Artificial intelligence" my ass. More like "artificial confidence". We've all worked with people like that. Now we get to pay \$200/month for the privilege.

And that's exactly why every wannapreneur indyhacker infobiz "ship 24 startups in 12 months" bro is so thrilled with AI tools. Of course they are. They're always starting from zero. Green field, no users, no legacy, no complexity. AI absolutely crushes that use case.

But for those of us maintaining real software that real people pay for and depend on? Yeah.

I guess our jobs are safe for another couple of years. Now excuse me, have to go review 67 files.



My AI work be like...

Comment

About

About Jitbit Software

Jitbit's Helpdesk Ticketing System

Subscribe

Get (rarely posted) updates in your inbox.

Enter your email

Subscribe