VIBE CODING · BUILDER · 12 MIN READ

Review AI-written code before you ship

Read the diff, check it does only what you asked, look for security and data risks, test it on a preview, and keep changes small.

Pass the check (4 of 5) online to earn the “Careful Reviewer” badge:
https://digitalbuilders757.com/university/study/review-ai-written-code

You're the reviewer now

Coding agents can change dozens of files in a minute. That speed is the point, but it means mistakes can slip in just as fast: a deleted check, a key in the wrong place, a 'small' change to the database. Professional teams review every change before it ships, usually as a pull request. You can do the same, even working alone.

Read the diff

- Look at the diff: the exact lines added and removed, file by file.
- Does it do what you asked, and only that? Unexpected files or features are a warning sign (scope creep).
- Ask the AI to explain the change in plain words and list every file it touched and why.
- If you can't explain a change, don't ship it yet.

The risky spots

Slow down for anything touching security or data: keys and environment variables, sign-in and permission checks (they must run on the server), database rules (RLS), migrations that change or delete data, payments, and new packages. Check that error handling didn't disappear and that nothing personal is being logged. These are the changes worth a second opinion from a person or another AI review.

Test, then ship small

Run the tests, click through the change on a preview deployment, and check it against the acceptance criteria for the task. Keep each change small, so it's easy to review and easy to roll back. Merge when it works; if something slips through, revert first and fix after.

Key terms

Diff
A view of exactly which lines were added, changed or removed.

Pull request (PR)
A request to merge one branch into another, with review and checks.

Scope creep
Extra requests that sneak in without changing the price or timeline.

Acceptance criteria
Testable conditions that say when a user story is done.

Preview deployment
A private, shareable copy of a branch or pull request for testing.

Migration
A saved, ordered script that changes the database structure.

Row level security (RLS)
Database rules that decide which rows each user may read or change.

Revert
Undo a change by going back to an earlier working version, often a Git commit.

Try it: Review one change properly

1. Ask your coding agent for one small change, and have it work on a branch.
2. Open the diff and list every file it touched; ask the AI to explain any change you don't understand.
3. Check the risky spots list and test the change on a preview link.
4. Merge it with a clear message, or ask for a smaller change if it did too much.

KNOWLEDGE CHECK

Review AI-written code before you ship

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. What is a diff?

A. A type of database backup that stores old versions of your app
B. A deployment error
C. A design file
D. The exact lines added and removed by a change

2. You asked for a new button; the diff also edits the database rules. What do you do?

A. Stop and ask why; unexpected data-rule changes need a careful look
B. Ship it, since the AI probably needed that change to make the button work
C. Delete the database rules so they can't cause problems for the new button
D. Ignore the diff and test later

3. Which change deserves the most careful review?

A. Fixing a typo in a heading
B. A change to sign-in checks or a migration that deletes data
C. Changing a button's colour
D. Adding a comment

4. Why keep each change small?

A. Small changes are always faster to type
B. Big changes aren't allowed on GitHub without a paid plan
C. Easier to review, test and roll back
D. Small changes use fewer AI credits, which is the main reason

5. Something broke right after a merge. Best first step?

A. Keep adding fixes on top in production
B. Delete the repository and restore it from a backup copy
C. Wait for users to stop noticing
D. Revert the change, then investigate

Official sites

GitHub docs: About pull request reviews: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/about-pull-request-reviews
Google: Code review developer guide: https://google.github.io/eng-practices/review/

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

Review AI-written code before you ship

1. D: The exact lines added and removed by a change
Reading the diff is how you see exactly what changed.

2. A: Stop and ask why; unexpected data-rule changes need a careful look
Changes outside the task are scope creep, and data rules are a risky spot.

3. B: A change to sign-in checks or a migration that deletes data
Security and data changes can hurt users and are hard to undo.

4. C: Easier to review, test and roll back
Small, focused changes make problems easy to spot and undo.

5. D: Revert the change, then investigate
Reverting gets users back to working software quickly.