

The build checklist

Requirements, docs, build, QA.

The questions to ask before you hire anyone to build your website, app or automation.

Five phases, plain questions, red flags, and a handover list. Print it or keep it open on the call. Tick a box only when you have heard a clear answer, or better, seen it on paper.

Requirements

Before anyone writes a line, you both agree what success looks like.

☐ What problem does this solve for my business, and how will we measure it in 90 days?

☐ Who are the people using it, and what is the one thing each must be able to do?

☐ What is in scope, and what is clearly out? Is that written down?

☐ What will you need from me (content, logins, decisions), and by what date?

☐ What does it cost, what is the payment schedule, and what changes the price?

☐ Who owns the code, the design and the accounts when we finish?

You should receive: A one-page brief you can read in two minutes and sign.

Docs and plan

Good builders write the plan down before building. It protects you if people change.

☐ Will I get a product requirements document (what it does, for whom, how we test it)?

☐ Is there a short architecture note: what the parts are and how they connect?

☐ Which services and tools will it depend on, and who pays for them?

☐ How is security handled: logins, stored data, secret keys, backups?

☐ What are the speed and accessibility targets, in numbers?

☐ Where are decisions recorded, so I can see why a choice was made?

You should receive: A doc pack: requirements, design rules, architecture, security, decisions.

Build

You should be able to see real progress every week, not a surprise at the end.

- ☐ Can I see a working version every week on a private link?
- ☐ How do you show progress: what is done, what is next, what is blocked?
- ☐ How do you handle a change I ask for halfway through?
- ☐ Is the code in a repository that I own or will be given?
- ☐ Are secrets (keys, passwords) kept out of the code?
- ☐ Is the code commented and tidy enough for another developer to take over?

You should receive: A weekly update note and a live preview link.

QA (quality checks)

Testing is part of the job, not an extra. Ask to see the results.

- ☐ What do you test, and can I see the written results with dates?
- ☐ Is it checked on a real phone, a tablet and a laptop, in more than one browser?
- ☐ What happens when something goes wrong: a wrong password, no internet, an empty form?
- ☐ What is the measured load speed, and what was the target?
- ☐ Can people using a keyboard or a screen reader use it?
- ☐ Who tests it besides the person who built it?

You should receive: A testing record, a performance report and an accessibility check.

Launch and after

Going live is a checklist, not a button. Make sure you can run it without the builder.

☐ Is there a step-by-step launch plan, and a way to undo it if something breaks?

☐ Will you hand over every login and transfer ownership to me?

☐ Is there a runbook: how to deploy, update, and fix common problems?

☐ What is covered after launch, for how long, and what does support cost?

☐ How do I change text or prices myself, without a developer?

☐ When will we review the results against the goal from phase 1?

You should receive: Live site or app, a runbook, and a review date in the diary.

Eight things to watch for

None of these alone means walk away. Two or three together do.

No written scope. If it is not on paper, every extra becomes an argument.

A price with no breakdown. You cannot compare quotes or trim scope without one.

No plan or docs, just "we will start". Nobody can take over, including the builder in six months.

You only see it at the end. Problems found late are the expensive ones.

"Testing is included" but no results. Ask to see the record. If none exists, it was not done.

The builder keeps the accounts. Domain, hosting, store and code should be in your name.

Pressure to pay everything up front. Pay in stages tied to things you can see.

Nobody can explain it in plain words. If they cannot explain it to you, they may not understand it.

What you should receive

On the day it goes live, ask for all twelve. Tick them off.

Requirements document

What it does, for whom, and how each part is tested.

Design rules

Colours, type, spacing, and motion, so new pages match.

Architecture note

The parts, how they connect, and the services used.

Source code

In a repository you own, with a readme that explains setup.

Logins and ownership

Domain, hosting, accounts and stores transferred to you.

Security note

How data, logins and keys are protected.

Testing record

What was tested, on which devices, with dates and results.

Accessibility check

What was checked and what was fixed.

Performance report

Measured speed against the agreed target.

Runbook

How to deploy, update, and fix common problems.

Customize guide

The text, prices and settings you can change yourself.

Decision log and changelog

Why choices were made, and what changed in each version.

Want a build done this way? Message @ferngalestudio with the word BUILD, or email hello@ferngale.com. A real person answers within a day.