

SCOPING YOUR WEB APPLICATION: AN HONEST PRE-BUILD CHECKLIST



Scoping Your Web Application: An Honest Pre-Build Checklist

The most expensive web application decisions get made before anyone writes a line of code — in the gap between “we need an app” and a clear picture of what it must do, who it’s for, and who owns it at the end. This checklist closes that gap.

Work through it before you commission anything. The clearer your answers, the tighter your scope, and the fewer surprises later — and the easier it is for any honest development shop, ours included, to tell you what to build, what to buy, and what to skip. There are no prices in here on purpose: scope comes first, and cost follows scope.

1. Start With the Problem, Not the Features

It’s tempting to open with a feature list. Start with the problem instead — the features fall out of it, and half the ones you imagined turn out to be unnecessary.

Answer First:

- What specific problem does this app solve, in one sentence?
- What do people do today instead, and what does that cost in time or errors?
- Is this process a competitive advantage, or plumbing you could buy?
- What happens to the business if you do nothing?

The Rule:

If the problem needs a paragraph to explain, the app will need a rewrite. Get it down to one sentence first.

2. Who Uses It: Users and Roles

Every web application is shaped by who logs in. Different users see different things — and “roles” is where scope quietly doubles.

List Your Roles:

- Who are the distinct types of user — customers, staff, admins, partners?
- What can each role see, and what can each role change?
- Who approves, and who only views?
- Will external users log in, or only your team?
- Roughly how many users, and how fast will that grow?

Why It Matters:

Each role is its own set of screens, permissions, and tests. Naming the roles early is the single best way to keep scope honest.

3. Map the Workflows It Must Support

An app isn't a set of screens; it's the work that moves through them. Trace the real workflow end to end before you design anything.

Trace Each Workflow:

- What starts the process, and what finishes it?
- What are the steps in between, in order?
- Where does a human decide, and where can the app decide automatically?
- What are the exceptions — the rejected, late, or missing paths?
- Which steps live today in spreadsheets, email, or someone's head?

The Rule:

The exceptions are where projects overrun. Write them down now, while they're cheap to plan for.

4. List Every System It Must Talk To

Almost no web application lives alone. Integrations are often the hardest, riskiest part of a build, so surface them first — not halfway through.

Inventory Your Integrations:

- What systems hold data this app needs — accounting, CRM, payment, email?
- Does data flow one way, or in both directions?
- Does each system have an API, or will data move by export and import?
- How fresh must the data be — real-time, hourly, nightly?
- Who owns the credentials and rate limits for each connection?

Why It Matters:

Every connection adds design, testing, and failure cases. A clear list turns an open-ended risk into a plannable one.

5. Decide Data and Code Ownership Up Front

Ownership is a scoping question, not a legal afterthought. Settle it before the first line of code — it changes how the app is built.

Get These In Writing:

- Who owns the source code and intellectual property when it's done?
- Who owns the data, and can you export it in a usable form?
- Can you change developers later without rebuilding from scratch?
- Where does the code live, and do you have access to it?
- Are there per-seat or licensing fees to use what you paid to build?

The Bottom Line:

Software you don't own is a subscription you paid to create. Insist on owning the code, the data, and the roadmap — in the contract, before work starts.

6. Plan Hosting and Scale

A web application has to run somewhere, reliably, and grow without falling over. Decide the shape of that before launch — not during the first outage.

Decide Early:

- Where will it be hosted — your cloud account, ours, or on-premises?
- How many users at peak, and how much data over a few years?
- What uptime does the business actually need?
- What is the plan for backups and recovery?
- Does any data carry compliance or security requirements?

The Rule:

Host it in an account you control. Hosting you can't access is just another form of lock-in.

7. Budget for Life After Launch

Software is never “done.” The build is the first cost, not the only one — and maintenance is the part most people forget to plan for.

Plan For:

- Who maintains the app after launch, and how is that arranged?
- How are bugs reported, triaged, and fixed?
- Who handles security patches and dependency updates?
- How do new features get requested and prioritized?
- What does support look like — hours, response times, contacts?

Why It Matters:

An app with no maintenance plan degrades quietly until it breaks loudly. Decide who keeps it healthy before you need them.

8. Define What Success Looks Like

If you can't tell whether the app worked, you can't tell whether it was worth building. Define success in measurable terms before you start.

Set Your Measures:

- What will be faster, cheaper, or possible that isn't today?
- How will you measure it — time saved, errors avoided, revenue enabled?
- What does "good enough to launch" look like versus "everything we imagined"?
- Which features are must-have, and which are nice-to-have?
- When will you review whether it delivered?

The Bottom Line:

A must-have list and a nice-to-have list are the two most valuable documents in any web project. Write them before you ask for a proposal.

The Final Word

A web application is only as good as the thinking that goes in before the build. Answer these questions honestly and you'll walk into any conversation with a development shop knowing what you need — and able to tell whether they're solving your problem or selling you a project.

Next Step:

Fill in your answers to sections 1, 2, and 5 first — the problem, the users, and ownership. Those three decide most of the scope. When you're ready to turn the checklist into a plan, QOS Software will scope it with you honestly, and tell you where an off-the-shelf product would serve you better. Call **877.800.7672** or visit **qossoftware.com**.