

GUIDES

What it costs to build a custom app

Ballpark ranges, what moves the number, and why the cheapest quote almost never ends up being the cheapest.

The short answer

It depends, and anyone who throws you a number without asking questions is selling you something else. But "it depends" doesn't help you budget either, so here are concrete ranges and — more importantly — **what moves them**.

Before the numbers, something we'd rather say upfront: **you will find quotes considerably cheaper than these**. That's not an exaggeration or a trick. There are people charging a fraction of this who deliver something that works on delivery day. The difference isn't delivery day: it's the six months after.

Ballpark ranges

Type of project	Range	What it usually covers
Custom site or landing page	USD 2,000 – 5,000	Original design, no template; contact form; technical SEO; an editing panel if needed
Simple mobile app	USD 3,500 – 10,000	iOS + Android from a single codebase; no backend or a minimal one; publishing to both stores, preferably on your account, or ours if that's decided
App or platform with a backend	USD 6,500 – 20,000	Admin panel, database, users and permissions, integrations with external services
Platform with AI or heavy integrations	USD 15,000 and up	All of the above plus language models, document or audio processing, or integration with systems you already run

These numbers are illustrative and by no means fixed, in either direction. A project from the third row can come in under one from the second if it has few screens and no awkward integrations. The only way to know is to talk it through and take it apart.

What moves the number

This is what we look at when putting a quote together. If you're comparing proposals from several studios, these are the questions worth asking all of them:

- **How many screens it has.** The most obvious factor and the most underestimated. An eight-screen app and a thirty-screen one have nothing in common, even when the initial brief sounds equally simple.
- **How many interactions live inside each screen.** A screen that shows a list and one that lets you filter, sort, edit inline and drag things around are the same screen in a wireframe and different projects in practice.
- **How many external systems it has to talk to.** Every third-party API — payments, invoicing, messaging, an ERP you already use — is work of its own: reading their docs, handling their errors, and keeping up when they change without warning.
- **Whether there are users, roles and permissions.** Making sure each person only sees their own data looks like a detail and is an entire layer of the system.
- **Whether it has to work offline.** Offline sync multiplies the complexity of any mobile app. It's often worth it, but you should know that's what you're asking for.
- **Whether it goes to the stores.** Publishing is its own work with its own annual costs, [all laid out here](#).
- **How much of this already exists.** Migrating data from an old system is often more work than building the new one.

About the cheaper quotes

You'll find them, and often the person behind them is perfectly capable. We're not saying they're scams. We're saying **the price doesn't cover the same things**.

What usually isn't included: the time spent understanding the problem before writing code, the support after delivery, and the technical decisions you don't notice until the system grows. That last one is the expensive part to fix later: when a database was modelled badly, or the app was built so that every new screen breaks two old ones, the fix isn't a correction — it's a rebuild.

What you're paying for when you don't pick the cheapest

The before. We ask a lot of questions before quoting, and not as a formality: half the briefs we get describe a solution instead of a problem. When we take the problem apart, the scope often shrinks — and so does the quote. That conversation is part of the work, not a step before invoicing.

The after. A system doesn't end the day it ships: that's when it starts being used, which is when the real issues show up. Having someone on the other end at that moment is worth more than any extra feature thrown into a proposal.

The years. Fifteen years building software doesn't make us write code faster. It makes us write it **better**: we've seen how systems break and we dodge from the start the problems that show up in immature builds. It isn't speed — it's not having to go back.

That we don't recycle screens. Every project is designed for that project. There's no template with the colours swapped, no previous app with a different logo on top. It's slower and it costs more, and it's exactly why the result looks like your business and not someone else's.

What to ask before signing — us or anyone else

- ✓ **Whose name do the accounts go under?** Domain, Google Play, Apple Developer. The right answer is: yours. **Why.**
- ✓ **What happens if the relationship ends?** If you can't continue with another team, you're not commissioning a build — you're renting a dependency.
- ✓ **Is the source code mine?** It should be, and you should be able to get it without negotiating.
- ✓ **What does the price include after delivery?** How long for fixes, and with what response time.
- ✓ **What does it NOT include?** The question that clarifies a quote the most.
- ✓ **What does it cost to run per year?** Servers, store accounts, third-party services. It's never zero.

The costs that come after

On top of development, every system has an annual cost of simply existing: servers, domain, store developer accounts, any third-party services you use. On small projects it tends to be low, but **it's never zero**, and it's better in the budget from the start than discovered a year in.

Store accounts have fixed, public numbers: we break them down in [the publishing guide](#).

If you have something in mind and want a real number instead of a range, tell us what you need. We ask first and quote second — and if the scope can be trimmed, we'll tell you.