

From prototype to production: 25 checks before launch

A checklist for an app built with an AI tool, by a freelancer or left by an agency. What it's missing, how to check it and when to start over.

For founders who have a prototype and are about to put it in front of real users or customers. It takes an hour, even without a developer. Where you can't answer, you have a question for whoever built the app.

Vibe coding gets you a prototype. We build software that runs a business. Since 2023 we have delivered 15+ production projects. We use this list on every takeover.

A missing item isn't a reason to rewrite. It's a to-do list.

A. Login, permissions and secrets

- ☐ 1. **Access control is on every entry point.** In the API too, not just on the main screen. Check: log in as a normal user and try to open someone else's record via its address.
- ☐ 2. **Passwords and keys aren't in the code.** If they ever were, they have been replaced.
- ☐ 3. **Only an admin sees admin functions.** Not just a hidden button.
- ☐ 4. **Password reset and invitations can't be abused.** The link expires and works only once.
- ☐ 5. **Forms are protected against bots** and limit the number of attempts.

B. Data

- ☐ 6. **Backups run automatically.** And you have restored one for real at least once.
- ☐ 7. **Database changes go through migrations.** Not manual edits on the live database.
- ☐ 8. **Test data and live data are separate.** Experiments don't happen on customers.
- ☐ 9. **Personal data is only where it's needed.** You know where it is and who can access it.

C. Deployment and environments

- ☐ 10. **Deploying is one command or one button.** Not done by hand from one computer.
- ☐ 11. **There is a way back.** You can roll back to the previous version in minutes.
- ☐ 12. **You have a test environment** separate from live operation.

- ☐ 13. **Someone else can run the app from scratch.** Following the instructions, on a clean computer.

D. Quality

- ☐ 14. **The main flows have tests.** Sign-up, payment, the thing the customer pays for.
- ☐ 15. **Tests really fail when something breaks.** A green tick without tests is decoration.
- ☐ 16. **CI blocks the merge** when tests fail. It doesn't just report.
- ☐ 17. **Integrations have a verified API and limits.** Payment gateway, email, third-party services. Check them before you build a plan on them.

E. Operations

- ☐ 18. **An alert reports an outage,** not a customer.
- ☐ 19. **Errors are logged** and someone reads them.
- ☐ 20. **Background jobs are watched.** Whatever runs on its own will one day quietly stop.
- ☐ 21. **You know what it costs to run** and what happens to the cost with 10 times more users.

F. Ownership and accounts

- ☐ 22. **The repository is in your company's name,** with its full history.
- ☐ 23. **Domain, cloud and payment accounts are in the company's name.** Not on the personal account of whoever clicked it together.
- ☐ 24. **More than one person can access the accounts.**
- ☐ 25. **The contract says who owns the code** and what you get when you leave the vendor.

Why monitoring is on the list

Because whatever runs unwatched will one day quietly stop. It happened to us too. On a project for a client with an online marketplace, a crawler stopped running in December 2025. It was noticed late, from a chart. Meanwhile traffic fell from 250K to 66K page views a month.

What was missing was one alert that would have reported the outage that day. That's why items 18 to 20 are mandatory, not a bonus.

Scoring

Count the ticked items. But look at items 1, 2, 6 and 22 first. If even one of them is missing, fix it before launch, whatever the score.

State	Verdict	What it means
The base holds, tests check something, deployment is repeatable	Continue	Keep building features and fill in the missing items along the way
The app works, but security, tests or monitoring are missing	Harden	Fix those first, then new features
Fixes would take longer than a new version	Start over	The prototype stays as the brief; the code is rewritten

The most common outcome isn't a rewrite. It's a few security fixes, a working deployment and then normal work. You can only say so honestly after looking at the code.

Even with a rewrite, the prototype stays. It shows how the product should work better than any specification.

What to prepare before the conversation

1. access to the repository or an export of the project from the AI tool,
2. who uses the app today and for what,
3. this list, with the items you weren't sure about marked.

Want to know what to do with your prototype?

In 15 minutes with Filip we go through the items you're missing. We tell you whether to continue, harden or start over.

Book 15 minutes with Filip: digitaliko.com/contact

Digitaliko s.r.o. · www.digitaliko.com · Custom software in weeks. Yours from day one.