Creiden

PRODUCTION READINESS CHECKLIST

Is Your AI-Built App Ready for Real Users?

A production readiness checklist for apps built with AI tools.

WHAT'S INSIDE

Prepared by Creiden. 15 years of software delivery, 400+ projects, 15+ countries.

creiden.com

How to use this

AI tools made it possible to build a working application in days. That part is real, and the thing you built is real. What these tools do not do is prepare an application for the moment strangers start using it, paying through it, and storing their data in it.

The gap is not about code quality in the abstract. It is about the things that only matter once you have real users: who can see whose data, what happens when something fails at 2am, what your bill looks like when traffic spikes, and whether anyone can rebuild your environment if it breaks.

This checklist covers 12 areas. Go through it with your app open. Tick what is genuinely handled. Leave the rest blank and be honest, because a blank box now is much cheaper than the same blank box after launch.

Two kinds of items

SELF You can check this yourself, right now, without technical help. Open the app, try it, look at it.

ENG This needs someone technical to verify properly. If nobody on your team can answer it, that itself is the finding.

What this is not

This is a self-assessment, not an audit. It tells you where the risk probably is. It does not tell you how deep any single risk goes, and it does not replace someone reading your actual code and configuration.

> **Time needed:** About 20 minutes for the SELF items. The ENG items need someone technical.

1 Architecture and Environments

You need a place to break things that is not the place your users live.

- [ ] ENG Production and staging environments are properly separated
- [ ] ENG Application components and external dependencies are correctly configured
- [ ] ENG Production data and services are connected to the correct environments
- [ ] SELF You can name every third-party service your app depends on
- [ ] ENG Critical dependencies and single points of failure are identified

2 Code Ownership and Platform Independence

If the tool you built with disappeared tomorrow, or doubled its price, you need to still have a business.

- [ ] SELF You have the full source code in a repository you control
- [ ] SELF The repository is owned by the company, not a personal account
- [ ] ENG The application can be deployed without the original AI building tool
- [ ] SELF You own the accounts for hosting, database, and domain
- [ ] SELF At least two people have access to every critical account
- [ ] ENG Setup instructions exist so a new developer can run the project locally

3 AI-Built Code Quality and Maintainability

AI writes code that works. It does not always write code that anyone can safely change six months later.

- [ ] ENG The codebase is understandable to a developer who did not build it
- [ ] ENG Critical logic is not duplicated across multiple places
- [ ] ENG No hardcoded values (keys, IDs, URLs, prices) that should be configurable
- [ ] ENG Fragile or improvised implementations that create production risk are identified
- [ ] ENG Generated code does not introduce obvious security or reliability risks
- [ ] SELF Someone on your team, or on retainer, can explain how any part of the app works

4 Authentication and Access Control

The most common serious problem in AI-built apps is that the app looks locked but is not.

- [ ] SELF Login, signup or invitation, and password recovery all work end to end
- [ ] SELF Logged-out users cannot reach protected pages by pasting the URL directly
- [ ] ENG Users can only access records they are authorized to access
- [ ] ENG Roles and permissions are enforced on the server, not only hidden in the interface
- [ ] ENG Sessions expire and unauthorized access is handled securely
- [ ] SELF You have tested the app logged in as a second, non-admin user

5 Data and Storage

Real users mean real personal data, and that comes with obligations.

- [ ] ENG One customer's data cannot be reached by another customer
- [ ] ENG Database access controls are correctly configured
- [ ] ENG Sensitive data is not exposed through the interface or API responses
- [ ] ENG Uploaded files are protected and not publicly guessable
- [ ] ENG Backups exist and restoring from them has actually been tested

6 Security and Secrets

The moment you are public, automated traffic finds you. Usually within hours.

- [ ] ENG Production credentials and secrets are securely managed
- [ ] ENG No keys or secrets are exposed in the browser or the repository
- [ ] ENG Environment variables are separated per environment
- [ ] ENG Third-party integrations are securely configured
- [ ] ENG Known security vulnerabilities have been identified and addressed

7 Backend and API

Anything enforced only in the browser is not enforced at all.

- [ ] ENG APIs and functions enforce authentication and authorization
- [ ] ENG Server-side validation exists for all critical operations
- [ ] ENG Sensitive business logic (pricing, credits, permissions) runs server-side
- [ ] ENG Webhooks and integrations are secured and verified
- [ ] ENG Critical operations handle errors and retries without corrupting data

8 User-Facing Experience

Your users will not send a bug report. They will leave.

- [ ] SELF Every critical user journey works from start to finish
- [ ] SELF Loading, success, empty, and error states are all handled
- [ ] SELF Forms validate input and give clear feedback when something is wrong
- [ ] SELF The app works on the phones and screen sizes your users actually have

9 Reliability and Error Handling

Things will fail. What matters is what your user sees when they do.

- [ ] SELF Failures never show raw technical errors or stack traces to users
- [ ] ENG Backend and API failures are handled gracefully
- [ ] SELF When something fails, the user is told what to do next
- [ ] ENG Critical actions cannot produce duplicate, lost, or inconsistent data

10 Performance and Scalability

Demo data is small and forgiving. Real data is neither.

- [ ] SELF Critical pages load in an acceptable time on a normal connection
- [ ] ENG Large data sets, lists, and file uploads are handled efficiently
- [ ] ENG Obvious scalability constraints are identified

11 Deployment, Testing and Change Safety

The real question is not whether you can launch. It is whether you can change anything afterwards without fear.

- [ ] ENG The production deployment process is defined and controlled
- [ ] ENG Automated build, lint, or type checks run before deployment
- [ ] ENG Changes can be tested somewhere safe before reaching production
- [ ] ENG Critical functionality has meaningful test coverage
- [ ] ENG Production configuration is reproducible from scratch
- [ ] SELF You can roll back a bad release quickly, and someone has done it at least once

12 Monitoring and Incident Readiness

If your users are the ones telling you the app is down, you found out too late.

- [ ] ENG Production errors are captured somewhere you can see them
- [ ] ENG Critical failures trigger an alert to a real person
- [ ] SELF It is clear who responds when production breaks, including at night and on weekends
- [ ] SELF A basic recovery procedure is written down, not just in someone's head

Your result

Count the boxes you left unticked.

0 to 4	In good shape. Close the remaining gaps and launch.
5 to 12	Material risk. The app works, but specific areas are likely to fail under real usage.
13 to 25	Not ready for real users yet. Normal for an AI-built app that has not had an engineering pass. Usually weeks of work, not months.
More than 25	Do not put real users, real data, or real money through this yet. Get a technical assessment first.

Any unticked box in sections 4, 5, or 6 should be treated as urgent regardless of your total, because those are the ones that cause damage you cannot undo.

Creiden has spent 15 years building and maintaining production software. We now help teams take AI-built applications the rest of the way, from working prototype to something that holds up with real users.

If you want a second opinion on what you found, you can reach us here.