

FREE APP AUDIT

THE VIBECODER CHECKLIST

A practical pre-launch review for AI-built apps.

WHAT'S INSIDE

- 01 Visual polish and loading states
- 02 Dark patterns to avoid
- 03 Make your app fast
- 04 Security launch audit
- 05 Privacy, payments and honest UX
- 06 App Store and Google Play planning
- 07 Testing before launch

MAKING IT LOOK PROFESSIONAL

Follow these to make sure your app looks like it was made by a real designer and not AI slop



Use skeleton loaders

Show a layout that mimics the page while its content loads. Users wait longer on average for this type of loading screen than a spinner. Even better if you animate it slightly.



Use simple backgrounds

Use a clean background that keeps the text easy to read. Add a gradient only when it serves the design (be aware that gradients are sometimes a giveaway of AI slop).



Add simple transitions

Animate changes between screens or states so users can follow what happened. Keep movement quick and subtle.



Avoid dark patterns

Do not trick people into paying, sharing data or staying subscribed. See the examples in the page below.



Add an onboarding flow

Help new users set up the app and complete their first useful task. Show only the steps they need at that moment.



Use colors intentionally

AI tends to use purple as a go-to. If your branding uses its own shade of purple use that but otherwise choose a color that aligns with your brand

PRO TIP

Look at these websites for inspiration to make your app look like it was professionally designed:

- [Mobbin](#)
- [Awwwards](#)
- [Dribbble](#)

02 / DARK PATTERNS

DARK PATTERNS TO AVOID

These choices can mislead people about price, consent or what they are buying. Replace them with clear, reversible decisions.

✗ Fake urgency Do not use a fake countdown or claim an offer is ending when it is not.	✗ Preselected consent Do not tick marketing or optional data-sharing boxes for the user.
✗ Hidden fees Show all required charges before the user agrees to pay.	✗ Unfair cookie choices Do not make “Accept all” easy while hiding “Reject”. Follow the user’s choice before loading optional trackers.
✗ Hidden subscriptions Do not call a trial simply “free” while hiding the price that follows.	✗ Guilt-trip messages Do not shame someone for saying no. Use neutral text for both choices.
✗ Hard-to-find cancellation Do not bury cancellation or force users through repeated screens or calls.	✗ Ads that look like reviews Do not make paid content look like an independent review, search result or recommendation.

ALSO AVOID

✗ Fake reviews Do not buy reviews, invent customers or hide paid endorsements. The FTC has a rule against deceptive reviews.
✗ Data you do not need Do not collect health, location, contacts or identity data “just in case”. Ask only for what a feature needs.

MAKE YOUR APP FAST

These three fixes help your app stay quick as users and data grow.

☐ **Implement caching**

Save data or results the app requests often. Set when the cache refreshes, and keep each user's data separate.

☐ **Make slow tasks run in the background**

Run imports, emails, exports and long AI tasks as background jobs. Show progress, let users keep working and retry failures safely.

☐ **Make database queries fast**

Fetch only the fields you need, split long lists into pages and add indexes for common searches. Check slow queries as data grows.

KEEP YOUR APP SAFE

AI-built apps can look complete while leaving security gaps. Verify these in the running app, not just the code, and test access with a second account. Apply platform-specific items where relevant.

☐ 01 Hide API keys Keep API, admin and payment keys on your server. People can inspect browser code and mobile apps.	☐ 06 Enforce server-side auth Check who the user is before every protected action. Hiding a button does not protect the API behind it.
☐ 02 Purge Git secrets Replace any leaked key first. Then remove it from your files and Git history. Deleting the file alone is not enough.	☐ 07 Lock record access Before showing or changing a record, check that the user owns it or has access. Try changing the record ID in a request.
☐ 03 Use public DB key If the app connects to your database directly, use only a public key. Never put an admin or service key in the app; set access rules too.	☐ 08 Block field tampering Accept only the fields users are allowed to change. Set roles, prices, owner IDs and approval states on the server.
☐ 04 Enable row-level security Set database rules for who can read or change each row. Test as a signed-out visitor and as two different users.	☐ 09 Secure session cookies For web logins, use Secure, HttpOnly and a suitable SameSite setting. End sessions on sign-out and keep tokens out of page scripts.
☐ 05 Encrypt sensitive data Use your provider's encryption at rest. Add stronger protection for especially sensitive fields and limit who can use the keys.	☐ 10 Hash passwords Use a trusted login provider. If you store passwords yourself, use a salted password hash such as Argon2id. Never store plain passwords.

04 / DATA AND SAFETY / CONTINUED

☐ 11 Rate limit login Slow repeated sign-in and password-reset attempts. Test that real users can still recover their accounts.	☐ 16 Restrict file uploads Accept only needed file types and sizes. Check the actual file, rename it and keep it out of executable folders.
☐ 12 Add bot protection Add limits or challenges to signup, forms and costly AI features when bots abuse them. Do not slow down every visitor.	☐ 17 Trim API responses Send only the fields each screen needs. Check that responses do not include private notes, tokens or another user's data.
☐ 13 Parameterize queries Pass user input to the database as parameters. Never build SQL by joining raw input into a query.	☐ 18 Add security headers For websites, set and test a Content Security Policy, frame protection and nosniff. Check that they do not break your app.
☐ 14 Validate all input On the server, check the type, length and allowed values of every request. Browser checks alone can be bypassed.	☐ 19 Force HTTPS Serve the site and APIs over HTTPS, redirect HTTP and fix mixed content. Turn on HSTS after HTTPS works across your domain.
☐ 15 Escape user content Show user text as text. If you allow rich HTML, clean it before displaying it.	☐ 20 Scan dependencies Scan packages before launch and in later builds. Fix serious issues and remove packages you no longer use.

PRIVACY AND FAIR TERMS

Write down what the app actually does with people's data and money. A generated policy only helps when its claims are true.

☐ Record what data you collect List every data field, tracking tool and vendor, why you use it and how long you keep it. If GDPR applies, explain the data categories and user rights in your privacy notice.	☐ Ask before using tracking cookies Where EU or UK rules apply, ask before nonessential tracking and offer a real reject choice.
☐ Have a privacy policy Say what data you collect, why, who gets it and how long you keep it. It must match the app: the FTC can challenge false privacy claims.	☐ Let users delete accounts Give users a clear way to delete an account. Explain any data you must keep and check Apple and Google's rules.
☐ Use a free policy generator The UK ICO offers a free privacy notice generator. Other services have basic free options too. Check every answer against what your app actually does.	☐ Make it easy to cancel Put cancellation in the account or plan area. Show when access ends and confirm it. For store-billed plans, point users to the store's subscription settings.
☐ Have Terms of Use Set rules against illegal use and cover ownership, payments and valid liability limits. Terms can help, but they do not remove all legal risk.	☐ Show subscription terms before payment Before purchase, show the price, billing period, renewal date, trial end and how to cancel.
☐ Check your app name for trademarks Search Google and app stores, then WIPO and your country's trademark register. Similar names can still cause trouble.	☐ Send a trial ending reminder Before the first charge, remind users of the date, price and cancellation path where possible. Check local and store rules.

PRO TIP Use a free service such as [Termly](#) to draft your privacy policy or Terms of Use and add a cookie consent banner. Its free plan currently covers one legal policy, so use another free route for the second. Check the generated text against your app.

PLAN THE STORE WORK EARLY

This page is for native or store-distributed mobile apps. A web-only SaaS does not need these store accounts.

App Store Connect **\$99 / year**

THE ROUTE

Join the Apple Developer Program, create an app record, prepare screenshots and privacy details, upload a build, test with TestFlight, then submit to App Review.

WHEN TO ENROLL

A free Apple developer account supports testing on your own device. Join the paid program when you need TestFlight or App Store distribution - early enough to prepare the listing and review.

Google Play Console **\$25 once**

THE ROUTE

Create a developer account, verify identity, prepare the store listing, privacy and Data safety details, upload a build, use a test track, then request production access where required.

WHEN TO ENROLL

Do not leave this until launch week. New personal accounts created after 13 Nov 2023 need a closed test with 12 opted-in testers for 14 continuous days before applying for production access.

TIMING RULE

Do not pay while the app is still just an idea. Do not announce a fixed store launch date before account setup, testing and review are ready.

BEFORE YOU SUBMIT

Check these before launch. Store-specific items apply when you distribute through the App Store or Google Play.

☐ Use the required payment system For digital goods in mobile apps, check Apple In-App Purchase and Google Play Billing rules before building checkout. Rules vary by product and region.	☐ Update your business details Check your identity, contact details, tax and bank information, privacy links and support email in both store consoles.
☐ Make good store screenshots Show real app screens and the main benefit first. AppScreens has a free option for up to five screenshots for your own app.	☐ Add a support page Link to a live page where users can contact you. Check that it works on a phone without signing in.
☐ Check your store listing Make sure the description, screenshots, age rating, data-safety answers and subscription text match the app.	☐ Add analytics Track signups, the first useful action, purchases, cancellations and errors. Leave out secrets and data you do not need.

PRO TIP Prepare a demo account beforehand. This not only makes it easy for you to do your own testing but store reviewers will require this too when they test your app for approval. Have multiple demo accounts if your app supports multiple subscription states.

TEST BEFORE YOU LAUNCH

Try these in a test environment so failures do not surprise your first users.

☐ Test without internet

Turn off Wi-Fi and mobile data during a key task. Explain what is offline, keep entered work and let users retry when the connection returns.

☐ Test service outages

Make your database, AI service or payment provider fail. Keep the user's work, show a clear message and offer a safe retry.

☐ Test payment problems

Try a failed card, a timeout and two quick taps on Pay. Check that nobody is charged twice and the app explains what to do next.

☐ Stress test the app

Simulate expected traffic, then push beyond it. Watch speed, errors, database load and background jobs in a test environment.

☐ Test purchases and subscriptions

For store apps, use Apple Sandbox and Google Play testers. Try purchase, refund, restore, trial conversion, renewal, failed payment, cancellation and plan changes.