

MINI PACK

First App Store Submission

Get your vibe-coded iPhone app from TestFlight to approved.

For Expo apps · with Claude Code or Cursor

10 prompts · 4 sections

Welcome! How to Use This Pack

Your iPhone app works on your own phone. Now Apple stands between you and launch: signing, TestFlight, a long form in App Store Connect, and a reviewer who might send it back with a guideline number you've never heard of.

Here's the good news. Apple publishes its rules, plus a "Before You Submit" checklist, so you can catch many common problems yourself before anyone at Apple sees your app. This pack walks you through it: reviewing your app the way Apple will, getting a build onto TestFlight, fixing the rules that are easy to miss, and handling a rejection calmly if one comes.

The build prompts are written for apps made with Expo (React Native). If your app is built another way, Sections 1, 3, and 4 still apply.

Apple and Expo change things often. Every guideline number and tool tip here comes from Apple's and Expo's official docs as of October 2026 (the App Review Guidelines page we checked was last updated October 16, 2026). If a menu or rule looks different for you, check the current docs. This pack helps you prepare. It can't promise approval, because only App Review decides that.

How each prompt is laid out

- **Title:** what the prompt is.
- **Use this when...:** one or two sentences on what the prompt is for and when to reach for it.
- **Fill in:** the blanks to replace before you paste.
- **The prompt:** the box you copy.
- **Tips to customize:** optional ways to change it.

Anything in **[BRACKETS IN CAPS]** is a blank to fill in.

Works best with: An Expo app, with Claude Code or Cursor for changes to your code. Prompts 4.1 and 4.2 only need what you paste, so you can also run them in any AI chat. You'll need a paid Apple Developer Program membership to use TestFlight or submit to the App Store.

The words, in one line each. **App Store Connect** is Apple's website where you set up your app, its store page, and its testers. **TestFlight** is Apple's way to share test builds before launch. **App Review** is the Apple team that approves or rejects each version. **EAS** is Expo's service that builds your app in the cloud and uploads it to Apple. A **build** is one uploaded copy of your app.

Quick Start: Find Your Problem

- **About to submit for the first time?** Prompt 1.1.
- **Worried Apple will say it's "just a website"?** Prompt 1.2.
- **Never built your app for the App Store before?** Prompt 2.1.
- **Build or upload failing?** Prompt 2.2.
- **Have Google or Facebook login?** Prompt 3.1.
- **Users can sign up, but can't delete their account in the app?** Prompt 3.2.
- **Need a privacy policy or stuck on App Privacy questions?** Prompt 3.3.
- **Charging for anything?** Prompt 3.4.
- **Filling in the store page and review notes?** Prompt 4.1.
- **Already rejected?** Prompt 4.2.

Golden rule: Give the reviewer everything they need: a working demo account, a backend that's switched on, and notes for anything that isn't obvious.

What's Inside

1. Check It Like a Reviewer (2 prompts)
2. Get It on TestFlight (2 prompts)
3. Fix the Rules That Are Easy to Miss (4 prompts)
4. Store Page, Review, and Rejections (2 prompts)

SECTION 1

Check It Like a Reviewer

Find the problems before Apple does. It's quicker than waiting for a rejection.

1.1 Pre-Review My App Against Apple's Guidelines

Use this when your app is nearly ready and you haven't submitted it yet. It's for getting an honest list of what App Review is likely to flag, sorted by how serious it is, before you wait on a review.

Fill in: [WHAT MY APP DOES] (e.g. "a habit tracker with daily reminders"), [HOW PEOPLE SIGN IN] (e.g. "email and password, plus Google", or "no login"), [HOW I MAKE MONEY] (e.g. "free for now", "a monthly subscription for extra features"), [WHAT DATA IT COLLECTS] (e.g. "email, habit entries, crash reports", or "not sure").

```
Don't change any files yet. I'm about to submit this app to the Apple App Store
for the first time, and I want to find problems before App Review does. I'm a
beginner.
```

```
What my app does: [WHAT MY APP DOES]
How people sign in: [HOW PEOPLE SIGN IN]
How I make money: [HOW I MAKE MONEY]
What data it collects: [WHAT DATA IT COLLECTS]
```

```
Read the project and check it against these App Review Guidelines:
```

1. 2.1 App Completeness: placeholder text, "coming soon" screens, broken links, buttons that do nothing, features that crash.
2. 4.2 Minimum Functionality: does it feel like a real app, or a website in a wrapper?
3. 4.8 Login Services: if there's Google, Facebook, or another social login, is there also an option that meets 4.8?
4. 5.1.1(v): if people can create an account, can they also start deleting it inside the app?
5. 5.1.1(i): is there a privacy policy link inside the app?
6. 3.1.1: does anything unlock paid features or digital content without Apple's in-app purchase?
7. 5.1.1(ii) Permissions: does every permission request (camera, photos, location, and so on) have a clear reason that explains why the app needs it?

```
For each one, say "OK", "Problem", or "Not sure", with the file path and one line
on why. Then list the problems from most to least likely to cause a rejection,
with a short fix for each. If you're not sure what a guideline says now, tell me
to read it on Apple's site instead of guessing.
```

 **Tips to customize:** Run this in plan mode in Cursor or Claude Code, so the AI explains before it edits anything. Have Apple's App Review Guidelines page open at developer.apple.com/app-store/review/guidelines and read the "Before You Submit" list at the top. It also asks you to test for crashes and bugs, keep your contact details up to date, and make sure your backend services are live during review. Apple's guidelines say to test on a real device before you submit, so install the app from TestFlight (Prompt 2.1) and tap through every screen yourself.

1.2 Is My App More Than a Website?

Use this when your app mostly shows web pages, or started life as a website you wrapped into an app. It's for checking it against Guideline 4.2 and adding features that make it worth installing, before Apple calls it a repackaged website.

Fill in: [WHAT MY APP DOES], [HOW IT'S BUILT] (e.g. "Expo, but most screens load my website in a WebView", or "Expo with native screens"), [MY WEBSITE ADDRESS] (if there is one).

Don't change any files yet. Apple's Guideline 4.2 (Minimum Functionality) says an app should include features, content, and UI that take it beyond a repackaged website, and that apps which aren't useful, unique, or "app-like" may be rejected. I want to check my app against it.

What my app does: [WHAT MY APP DOES]

How it's built: [HOW IT'S BUILT]

My website: [MY WEBSITE ADDRESS]

Please:

1. List every screen and say whether it's native or loads a web page.
2. Tell me honestly how "app-like" this feels compared with just visiting the website in Safari, and why.
3. Suggest 3 to 5 changes that would make it more useful as an iPhone app, ranked by value for effort. Only suggest things that fit what the app is for, not features added just to pass review.
4. For the top suggestion, describe what I'd build and which files it touches.

 **Tips to customize:** Guideline 4.2.2 adds that, other than catalogs, apps shouldn't mainly be marketing materials, ads, web clippings, content aggregators, or a collection of links. If your app really is just your website, a well-made mobile website may serve your users better than an App Store listing. Our Make It Pretty pack covers making a site feel great on phones.

SECTION 2

Get It on TestFlight

Get a real build onto a real phone before you ask Apple to review it.

2.1 My First Build and TestFlight Upload With Expo

Use this when your Expo app has never been built for the App Store. It's for getting your first build onto TestFlight step by step, so you and your testers can install it on a real iPhone.

Fill in: [MY COMPUTER] (e.g. "Mac" or "Windows"), [MY BUNDLE ID] (e.g. `com.yourname.habittracker`, or "don't have one yet"), [WHAT I'VE DONE SO FAR] (e.g. "joined the Apple Developer Program, created an Expo account", or "nothing yet").

```
Help me get my Expo app onto TestFlight for the first time. I'm a beginner.
```

```
My computer: [MY COMPUTER]
```

```
My bundle ID: [MY BUNDLE ID]
```

```
What I've done so far: [WHAT I'VE DONE SO FAR]
```

```
What I know from Expo's docs:
```

- I need a paid Apple Developer account and an Expo account.
- The bundle identifier goes in app.json under ios.bundleIdentifier.
- I can build with "eas build --platform ios --profile production" and upload with "eas submit --platform ios", or do both at once with "eas build --platform ios --auto-submit".
- "npx testflight" runs the whole flow in one guided command, including Apple credentials and the upload.

```
Please:
```

1. Read my app.json (or app.config) and eas.json and tell me what's missing for an App Store build, without printing any secrets.
2. Give me the exact commands to run, in order, with one line on what each one does.
3. Tell me what questions the commands will ask (Apple sign-in, credentials, encryption) and what to answer as a beginner.
4. Tell me what happens after the upload, and how to install the build on my own iPhone.



Tips to customize: Expo's docs say a build shows up in TestFlight after Apple processes it, usually in 10 to 15 minutes. You can let EAS create and manage your signing certificate and provisioning profile, which is the easiest path for beginners. TestFlight lets you add up to 100 internal testers from your App Store Connect team. External testers (up to 10,000, by email or a

public link) need your first build approved by TestFlight App Review. Each build stops working for testers after 90 days, so upload fresh builds while you test.

2.2 Fix My Failed iOS Build or Upload

Use this when `eas build` or `eas submit` fails, or the upload works but the build never shows up in TestFlight. It's for finding the cause in the log and fixing it without guessing.

Fill in: [THE COMMAND I RAN] (e.g. `eas submit --platform ios`), [THE ERROR] (paste the last 30 to 50 lines of the log, with any passwords or keys removed), [WHAT CHANGED SINCE IT LAST WORKED] (e.g. "added a camera library", or "first build ever").

My iOS build or upload with Expo EAS failed. Help me fix it. Explain before you change anything.

The command I ran: [THE COMMAND I RAN]

The error: [THE ERROR]

What changed since it last worked: [WHAT CHANGED SINCE IT LAST WORKED]

Check these possible causes and say "likely", "possible", or "unlikely" for each, with one line why:

1. Build number: Apple rejects an upload whose build number isn't higher than one I already uploaded.
2. Signing: a problem with my distribution certificate or provisioning profile.
3. Bundle identifier: it doesn't match the app I created in App Store Connect.
4. A library that needs a config plugin or native setup I skipped.
5. A missing or wrong environment variable in the EAS build.
6. An Apple account issue, like an agreement I haven't accepted in App Store Connect.

Then give me the fix for the most likely cause, step by step. If you need to see a file, ask me for it.

 **Tips to customize:** To stop build number clashes for good, Expo's docs suggest setting `"appVersionSource": "remote"` under `cli` in `eas.json` and `"autoIncrement": true` in your production build profile, so EAS raises the number for you. Run `eas credentials --platform ios` to see and manage the signing credentials EAS keeps for you. If an upload went through but the build is missing, open your app's **TestFlight** tab in App Store Connect and check **Build Uploads**, which shows any errors. The full log for every EAS build is on your dashboard at expo.dev.

SECTION 3

Fix the Rules That Are Easy to Miss

Four rules that are easy to miss when you're new. Each one is a clear, fixable task.

3.1 Add Sign in with Apple Next to Google Login

Use this when your app lets people sign up with Google, Facebook, or another social login. It's for adding a login option that meets Guideline 4.8, such as Sign in with Apple, without breaking the login you already have.

Fill in: [MY CURRENT LOGIN] (e.g. "Google and email/password through Supabase Auth"), [MY BACKEND] (e.g. "Supabase", "Firebase", or "my own server").

My Expo app uses a third-party login to create accounts, so I need to meet Apple's Guideline 4.8 (Login Services). I want to add Sign in with Apple.

My current login: [MY CURRENT LOGIN]

My backend: [MY BACKEND]

What I know:

- Guideline 4.8 says apps that use a third-party or social login for the main account must also offer an equivalent login option that limits data collection to name and email, lets people keep their email private, and doesn't collect app activity for advertising without consent.
- Expo has the expo-apple-authentication library. It needs "usesAppleSignIn": true under ios in app.json and a new build, not just a reload.
- Its AppleAuthenticationButton shows Apple's official button. I should only show it when `isAvailableAsync()` returns true, and I shouldn't set its background color or border radius through style (it has its own `cornerRadius` prop).

Please:

1. Make a plan to add Sign in with Apple next to my current login, using my backend's own docs for Apple sign-in on native iOS. Show me the plan before you write any code.
2. Make sure it creates the same kind of user account as my other logins.
3. Handle the cases where the person cancels, or hides their email and shares a private relay address instead.
4. Tell me what I must set up outside the code (Apple Developer account, my backend's dashboard), step by step.



Tips to customize: Guideline 4.8 lists a few exceptions, like apps that only use their own email and password accounts, so if you have no social login you may not need this. Apple's own privacy page says Sign in with Apple only lets apps ask for a name and email, lets people hide their email, and doesn't track people. If someone hides their email, Apple gives you a forwarding address, and

Apple's docs say you must register your sending domains (with SPF and DKIM) before emails will reach it. Our Signup Emails pack covers SPF and DKIM. Test on a real iPhone. Expo's docs say the Simulator doesn't behave exactly the same.

3.2 Add Account Deletion Inside the App

Use this when people can create an account in your app but have no way to delete it there. It's for adding the in-app deletion Apple requires under Guideline 5.1.1(v), in a way that actually removes their data.

Fill in: [MY BACKEND] (e.g. "Supabase"), [WHAT I STORE PER USER] (e.g. "profile, habit entries, uploaded photos"), [DATA I MUST KEEP] (e.g. "order records for tax reasons", or "nothing"), [DO I USE SIGN IN WITH APPLE] (yes or no).

Apple requires apps that let people create an account to also let them start deleting it inside the app (Guideline 5.1.1(v)). Add this to my app. Show me the plan before you write code.

My backend: [MY BACKEND]

What I store per user: [WHAT I STORE PER USER]

Data I must keep: [DATA I MUST KEEP]

Do I use Sign in with Apple: [DO I USE SIGN IN WITH APPLE]

What Apple's guidance says:

- The option must be easy to find, usually in account settings.
- It must delete the whole account and its personal data, not just deactivate it.
- I can ask people to confirm, or sign in again, but I can't make it needlessly hard or make them email, call, or chat with support.
- If deletion takes time, or some data is kept for legal reasons, I should tell them.

Please:

1. Add a "Delete account" option in settings with a clear confirmation step.
2. Do the deletion on my server or in a backend function, never with a secret or admin key inside the app.
3. Delete or anonymize every piece of data listed above, except [DATA I MUST KEEP], and tell me if you find other user data I didn't list.
4. Sign the person out and return them to the start screen afterward.
5. If I use Sign in with Apple, add a note on revoking their Apple token.
6. Give me a 3-step test I can run with a throwaway account.

 **Tips to customize:** Apple's account deletion page says apps using Sign in with Apple should use the Sign in with Apple REST API to revoke the user's tokens when they delete their account. It also says that even if your app sends people to a website to create an account, you still need deletion inside the app. If deletion has to finish on your website, link straight to that page, not your home page. Our Ship It Safe pack covers keeping admin keys out of your app.

3.3 Privacy Policy and App Privacy Answers From My Real Code

Use this when you need a privacy policy, or you're stuck on the App Privacy questions in App Store Connect. It's for listing what your app really collects, including what its add-on libraries collect, so your policy and answers match the code.

Fill in: [WHAT MY APP DOES], [SERVICES I KNOW I USE] (e.g. "Supabase, Sentry for crash reports, RevenueCat", or "not sure"), [MY CONTACT EMAIL] (for the policy).

```
Don't change any app code. Help me write an honest privacy policy and prepare my
App Privacy answers for App Store Connect, based on what this app actually does.
```

```
What my app does: [WHAT MY APP DOES]
```

```
Services I know I use: [SERVICES I KNOW I USE]
```

```
My contact email: [MY CONTACT EMAIL]
```

```
Please:
```

1. Read the code and package.json and list every kind of data that leaves the phone: what it is, where it goes (my backend or a third-party library), and why. Include analytics, crash reporting, ads, and login libraries.
2. For each one, tell me whether it's linked to the user's identity and whether it's used for tracking or advertising.
3. Write a plain-English privacy policy that covers: what data is collected and how, how it's used, that any third party I share data with protects it the same way, how long I keep it, and how someone can ask for it to be deleted or withdraw consent.
4. Make a table I can follow while answering App Privacy in App Store Connect: data type, purpose, linked to user (yes/no), used for tracking (yes/no).
5. Mark anything you're unsure about as "Check this" instead of guessing.

 **Tips to customize:** Apple's guidelines (5.1.1(i)) say the privacy policy link must be in the App Store Connect field and inside your app where it's easy to find. Apple's App Privacy page says you must also declare what third-party libraries in your app collect, and that data handled only on the phone, never sent to a server, doesn't count as collected. Host the policy on a real web page, not a document link that needs a login. This prompt drafts a starting point, not legal advice, so have a lawyer review it if you handle sensitive data or serve users in many countries.

3.4 Selling Something? In-App Purchase or Not

Use this when your app charges money or plans to. It's for finding out whether Apple requires its in-app purchase system for what you sell, so you don't build Stripe checkout into an app that could be rejected for it.

Fill in: [WHAT I SELL] (e.g. "a monthly plan that unlocks unlimited habits", "physical stickers", or "1-on-1 coaching calls"), [HOW I CHARGE NOW] (e.g. "Stripe on my website", or "not set up yet"), [WHERE I'LL SELL THE APP] (e.g. "United States only", or "worldwide").

```
Don't change any files yet. Help me understand whether Apple requires in-app
purchase for what I sell in my iPhone app. I'm a beginner.
```

```
What I sell: [WHAT I SELL]
```

```
How I charge now: [HOW I CHARGE NOW]
```

```
Where I'll sell the app: [WHERE I'LL SELL THE APP]
```

```
What I know from Apple's App Review Guidelines:
```

- 3.1.1: unlocking features or functionality inside the app, like subscriptions, premium content, or a full version, must use in-app purchase.
- 3.1.3(e): physical goods and services used outside the app must NOT use in-app purchase.
- 3.1.3(d): real-time services between two people, like tutoring, may use other payment methods.
- The rules about linking to outside payment pages differ by country, and the United States storefront has its own exception.

```
Please:
```

1. Sort each thing I sell into "must use in-app purchase", "must not use in-app purchase", or "check with Apple", with the guideline number and one line why.
2. Find any place in my code that takes payment or links to a payment page, and say whether it's likely a problem.
3. If I need in-app purchase, give me a short plan for adding it to an Expo app, and tell me to check the library's and Apple's current docs before I start.
4. Tell me anything I must explain in my App Review notes about purchases.

 **Tips to customize:** Payment rules differ by country, and the United States storefront has its own exceptions. Read sections 3.1.1, 3.1.1(a), and 3.1.3 on Apple's guidelines page before you decide, and treat the AI's answer as a first pass. Guideline 2.1(b) says any in-app purchases must be complete, visible to the reviewer, and working when you submit. If you sell physical products through Stripe, our Get Paid pack covers checkout and webhooks.

SECTION 4

Store Page, Review, and Rejections

The last mile: what the reviewer sees, and what to do if they say no.

4.1 My Store Listing and Review Notes

Use this when you're filling in your app's page in App Store Connect before you submit. It's for writing the description, a screenshot plan, and review notes with a demo account, all in a way that follows Apple's rules.

Fill in: [WHAT MY APP DOES], [WHO IT'S FOR] (e.g. "people who want to build one small habit at a time"), [MAIN SCREENS] (e.g. "today list, add habit, streak calendar, settings"), [HOW TO REACH THE MAIN FEATURES] (e.g. "sign in, tap +, add a habit"), [ANYTHING NON-OBVIOUS] (e.g. "reminders only fire after 24 hours", or "nothing").

Help me prepare my App Store listing and App Review notes. Follow Apple's metadata rules.

What my app does: [WHAT MY APP DOES]

Who it's for: [WHO IT'S FOR]

Main screens: [MAIN SCREENS]

How to reach the main features: [HOW TO REACH THE MAIN FEATURES]

Anything non-obvious: [ANYTHING NON-OBVIOUS]

Rules I know from Apple's guidelines:

- App name: 30 characters or less, no popular app names or trademarked terms that aren't mine.
- No prices or names of other platforms (like Android) in the name, subtitle, or screenshots.
- Screenshots must show the app in use, not just the title, login, or splash screen.
- The demo account for App Review must not expire, and the backend must be on during review.

Please write:

1. 3 app name ideas and 3 subtitle ideas, with character counts.
2. A short description in plain words, with no claims I can't back up.
3. A plan for 5 screenshots: which screen, and a short caption for each.
4. App Review notes: how to sign in with the demo account, the exact steps to reach each main feature, and an explanation of anything non-obvious. Leave the username and password blank for me to fill in myself.
5. A checklist of everything else I must have ready: support URL, privacy policy URL, age rating answers, category, and contact details.

💡 **Tips to customize:** App Store Connect has a "Sign-in required" box for the demo account's username and password, and a Notes field (up to 4,000 bytes) for everything else. Fill in the demo account in App Store Connect yourself and don't paste real passwords into an AI chat. Use fake names and data in screenshots, not real customers. Apple's screenshot specifications page lists the required sizes, which can change, so check it before you make yours. If your app.json sets `supportsTablet` to true, your app supports iPad and you'll need iPad screenshots too. Apple says your support URL must lead to real contact details.

4.2 I Got Rejected: Fix Plan and Reply to App Review

Use this when App Review rejected your app and you have the message in front of you. It's for turning it into a clear fix plan and a calm, short reply, and for knowing whether you need a new build at all.

Fill in: [THE REJECTION MESSAGE] (paste all of it, including guideline numbers), [WHAT MY APP DOES], [WHAT I THINK HAPPENED] (e.g. "the reviewer couldn't log in", or "no idea").

My app was rejected by Apple's App Review. Help me fix it and reply. Don't change any code until I approve the plan.

The rejection message: [THE REJECTION MESSAGE]

What my app does: [WHAT MY APP DOES]

What I think happened: [WHAT I THINK HAPPENED]

Please:

1. For each guideline mentioned, explain in plain words what Apple is asking for, quoting the exact part of their message.
2. Sort each issue into: needs a code change and a new build, can be fixed in App Store Connect (metadata, demo account, review notes), or a misunderstanding I should explain.
3. Give me a step-by-step fix plan for each issue, in order, with the files to change.
4. Write a short, polite reply to App Review that says what I changed or explains the misunderstanding. No arguing, no excuses, and nothing I haven't actually done.
5. Tell me whether I should reply only, or upload a new build and resubmit.
6. If anything in the message is unclear, write one question I can ask App Review.

💡 **Tips to customize:** Reply in App Store Connect: open your app, click the link about unresolved issues at the top, click **Resolve** next to the submission, then **Reply to App Review**. You can attach screenshots. Apple's help says that if the fix is only metadata or something on your server, like turning on the demo account, you can fix it and reply without a new build. Resubmit only after a build fix, and only once every issue is resolved, because resubmitting starts a new review. If you think the rejection is wrong, Apple also has an appeal process through the App Review Board.

Bonus: Magic Phrases for App Store Prep

- "Check this against Apple's App Review Guidelines and name the guideline number."
- "If you're not sure what Apple requires now, tell me to check the docs instead of guessing."
- "Don't change any files yet. Explain first."
- "Do this on the server, never with a secret key inside the app."
- "Write it for a reviewer who has never seen my app."
- "Mark anything you're unsure about as 'Check this'."

First App Store Submission

That's the whole pack: **10 prompts across 4 sections**. Review your app like Apple will, get it onto TestFlight, fix the rules that are easy to miss, and handle the store page and any rejection with a plan.

Happy building!