

The App Store Rejection Checklist

Step-by-step fixes for the 9 most common App Store rejections — what to check, what to change, and how to resubmit.

22%

of all submissions were rejected in 2025

63%

of developers hit at least one rejection a year

85%

of appeals to the App Review Board are denied — fixing and resubmitting is almost always faster

How to use this: find the guideline number from your rejection email in App Store Connect, jump to that section, and work through the checklist before you resubmit.

GUIDELINE 2.1

App Completeness

The single most common rejection — crashes, bugs, placeholder content, or a demo login the reviewer couldn't get past.

- ☐ Reproduce on a clean install: delete the app, install the exact review build via TestFlight, test on iPhone and iPad.
 - ☐ Test on an IPv6-only network (NAT64) — hard-coded IPv4 and legacy networking code fail here.
 - ☐ Symbolicate any crash log Apple attached in Xcode and fix the root cause.
 - ☐ Fix the demo account: working credentials, full feature access, no SMS/2FA blocking the reviewer.
 - ☐ Sweep for placeholder text, dead buttons, and confirm support/privacy URLs actually load.
 - ☐ Keep server-gated features fully populated for the entire review window.
-

GUIDELINE 2.3.3

Inaccurate Screenshots

Screenshots must show the app actually in use — not a splash screen, login page, or pure marketing art.

- ☐ Identify every flagged device size — check all sizes uploaded, not just the one Apple named.
 - ☐ Capture real UI: content loaded, genuine data, a screen mid-task.
 - ☐ Keep marketing framing (device frames, captions) but make sure real UI is clearly visible.
 - ☐ Match every screenshot slot to its device — iPad slots need real iPad captures.
 - ☐ Audit app preview videos too: screen captures only, no concept animation.
-

GUIDELINE 2.3.10

Accurate Metadata

Metadata must stay focused on the iOS experience — no other-platform mentions, no pricing, no irrelevant keywords.

- ☐ Sweep every field: name, subtitle, description, promotional text, keywords, What's New, screenshots.
 - ☐ Remove Android/other-platform mentions and drop pricing from the description.
 - ☐ Replace flagged copy, don't just delete it — keep the text natural.
 - ☐ Re-export any screenshot showing a Play Store badge or a price.
 - ☐ If Apple said "app and its metadata", check the binary too — that needs a new build.
-

GUIDELINE 2.5.1

Software Requirements

Code-level only — a metadata edit never fixes this. Private APIs, non-public frameworks, deprecated tech.

- ☐ Copy the exact flagged symbol, selector, or framework name from the rejection — verbatim.
 - ☐ Search your own source first; rename or replace any colliding method.
 - ☐ If it's not your code, grep/strings/nm your bundled frameworks to find the dependency.
 - ☐ Update or remove the offending third-party SDK.
 - ☐ Replace deprecated tech (e.g. `UIWebView` → `WKWebView`) — grep the whole project and Pods.
 - ☐ Validate in Xcode Organizer (or Transporter) before resubmitting — it catches most of these early.
-

GUIDELINE 3.1.1

In-App Purchase

Apple's core monetization rule: digital content consumed in the app must go through Apple's IAP system.

- ☐ Classify what you're selling: digital-in-app → IAP required. Physical/external → regular payments.
 - ☐ Create the products in App Store Connect and submit them together with the app version.
 - ☐ Implement StoreKit 2 with a working Restore Purchases path — its absence is a common flag on its own.
 - ☐ Remove external payment links and "cheaper on our site" copy for digital goods.
 - ☐ Test the full purchase flow with a sandbox account before resubmitting.
-

GUIDELINE 4.2

Minimum Functionality

Apple's "why is this an app?" rejection — web wrappers and thin utilities with too little native behavior.

- ☐ Diagnose honestly which bucket you're in: web wrapper, thin utility, or content-as-app.
 - ☐ Add native capabilities the website can't deliver: push notifications, offline caching, native navigation.
 - ☐ Deepen the core loop: persistence, history, settings, iCloud sync, a widget, App Intents/Shortcuts.
 - ☐ Move rendering native where feasible instead of one full-screen web view.
 - ☐ Seed content and provide a demo account — no empty state on first launch.
 - ☐ Enumerate your native features explicitly in the App Review notes field.
-

GUIDELINE 4.3

Spam / Design

Questions the app's right to exist — duplicate, template, or category-saturating apps, including 4.3(a).

- ☐ Work out which trigger matched: binary similarity, concept similarity, or metadata.
 - ☐ If built from a template, rework core screens and add features the template doesn't have.
 - ☐ If you publish many client apps, consolidate into one configurable app or separate ASC accounts.
 - ☐ Use a unique name, an original icon, and screenshots of your actual UI.
 - ☐ Document your differentiation concretely in the reviewer notes.
 - ☐ If the flag is plainly wrong, reply in the App Review thread with evidence before escalating.
-

GUIDELINE 5.1.1

Data Collection & Storage

The second most common rejection — purpose strings, forced registration, account deletion, privacy policy accuracy.

- ☐ Write specific, honest purpose strings for every flagged Info.plist permission key.
 - ☐ Delete unused permission requests and the code paths that trigger them.
 - ☐ Let users reach non-account features without being forced to register.
 - ☐ Add a real in-app "Delete account" flow that actually starts deletion, not just a support email.
 - ☐ Confirm the privacy policy URL loads and matches your App Privacy questionnaire answers.
 - ☐ Rebuild and resubmit — purpose strings live in the binary, so this needs a new build.
-

GUIDELINE 5.1.2

Data Use & Sharing

Governs tracking and third-party sharing after collection — enforced primarily through ATT since iOS 14.5.

- ☐ Inventory every ads/analytics/attribution SDK in your binary and check each one's ATT/IDFA behavior.
 - ☐ Add `NSUserTrackingUsageDescription` and call `ATTrackingManager.requestTrackingAuthorization` correctly.
 - ☐ Configure SDKs into a real non-tracking mode on ATT denial — never fall back to fingerprinting.
 - ☐ Stop any undeclared third-party data sharing, or gate it behind explicit, informed consent.
 - ☐ Reconcile your App Privacy label with what the app and its SDKs actually do after the fix.
 - ☐ Verify with a proxy or Instruments that declining ATT actually stops outbound tracking.
-

Forward the rejection instead

AscAuto reads Apple's rejection email, classifies the guideline, drafts the fix — a reviewer reply or a pull request — and resubmits after your one-click approval. Nothing goes to Apple without you seeing it first.

ascauto.org · ascauto.org/rejections