

# Comprehensive Guide to Passing the Apple App Store Review (iOS & iPadOS)

Apple's App Store review process is notoriously strict, but following Apple's official guidelines and best practices can ensure your iOS/iPadOS native app sails through approval. This guide covers every detail – from Apple's five main guideline categories to common rejection pitfalls – so you can avoid rejection and confidently launch on the App Store.

## Preparing for Submission: Pre-Review Checklist

Before you even hit “Submit” in App Store Connect, make sure your app and metadata are 100% ready. Many rejections happen due to easily preventable issues. Use this checklist to catch common problems upfront:

- **Test Thoroughly for Bugs & Crashes:** Run your app on real devices (iPhone and iPad) and fix all crashes or glitches. Apple will reject apps that crash or have obvious bugs[1][2].
- **Complete All Features (No Placeholders):** Ensure no “coming soon” buttons, lorem ipsum text, or blank screens remain. All app functions should be fully implemented – incomplete features or placeholder content will trigger rejection[3].
- **Accurate and Complete Metadata:** Double-check your App Store listing (description, screenshots, etc.) to ensure it precisely matches the app's content and functionality[4][5]. Misleading claims or outdated screenshots can lead to rejection for inaccurate metadata.
- **Provide Access for Reviewers:** If your app requires login or special hardware, include a demo account (and any necessary credentials) or enable a fully

featured demo mode for Apple’s testers[6][1]. Also ensure any backend servers or features are live and accessible during review.

- Include Required App Content: If your app has *In-App Purchases* behind a paywall, implement a “Restore Purchases” button and test it (buy, reinstall, restore to unlock)[7][8]. If your app allows account creation, provide an in-app way to delete the account (a 2022 rule)[9]. Make sure a Privacy Policy URL is provided in App Store Connect and that the policy is easily accessible within the app[10][11].
- Update Contact Info: Ensure the support contact info in your app and App Store listing is up to date. Users (and Apple) should have an easy way to reach you with support issues[12].
- Add Explanatory Notes: Use the “Notes for App Review” field to explain any non-obvious features, special configurations, or use of sensitive APIs. For example, if your app uses location in the background, clarify why and how (e.g. “used only when user selects the Nearby Events feature”)[13]. This helps reviewers understand your app and speeds up approval.

Apple itself emphasizes reviewing these common missteps before submission to smooth the approval process[14][6]. Taking the time to walk through your app like a reviewer – fresh install, no cached data, slow network, etc. – is a smart move[15]. If anything in your app would confuse a first-time user (or reviewer), provide instructions in the review notes or improve the UX before submitting[16][17].

## App Store Review Guidelines Overview

Apple’s official App Store Review Guidelines are organized into five broad categories[18], each addressing different aspects of your app’s content and behavior. All of these must be met for approval. Below, we break down each category – Safety, Performance, Business, Design, and Legal – highlighting the key points and how to comply.

## Safety: User Content and Device Safety

Apple prioritizes apps that respect users' safety, comfort, and wellbeing. The Safety guidelines cover what content/apps are *not* allowed and how to handle user-generated content responsibly[19][20].

- **Objectionable Content is Prohibited:** Apps *must not* contain anything Apple deems offensive, disturbing, or harmful. This includes defamatory or hateful material (e.g. racist or sexist content), realistic violence or abuse, encouragement of illegal acts, overtly sexual or pornographic content, and overly shocking or crude content[21][22]. Hate speech, graphic violence, pornography, and extreme gore will get your app rejected[23][22]. If your app's purpose is to shock or offend, it "isn't the right place" on the App Store[19].
- **User-Generated Content (UGC) Must Be Moderated:** If your app lets users post content (comments, images, etc.), you must include moderation features. Apple requires UGC apps to have: a content filtering system, a user reporting mechanism for abuse, a way to block abusive users, and contact info for users to reach the developer[20][24]. Apps primarily used for objectionable UGC (pornography, bullying, "hot-or-not" rating of people, etc.) will be rejected or removed[25]. *Tip:* Add a "Report" button on user posts and implement a review process for flagged content[26][27].
- **Apps for Kids:** If you target children or use the Kids Category, extra rules apply. No inappropriate content or ads may be present, and you cannot include external links or purchases outside a gated parent-only section[28]. Apps in Kids category must not transmit any personal data of children, and generally should not use third-party analytics or third-party advertising in order to protect child privacy[29][30]. Compliance with all local laws on children's data (like COPPA) is mandatory.
- **Physical Safety and Health:** Apps that could affect someone's physical health or safety face higher scrutiny. For example, medical or health apps must provide

accurate data and appropriate disclaimers – if you claim to measure heart rate or blood oxygen with only the phone sensors, expect rejection[31]. Apps that could risk physical harm (e.g. instructing dangerous device usage) may be rejected[32]. Always disclose and validate any health-related claims, and advise users to consult a doctor where relevant[33].

- **Developer Contact & Support:** As part of user safety, users need a way to reach you. You must provide an easy-to-find support contact in your app or listing[12]. If users can't get help or report issues, Apple may reject the app.
- **Data Security:** Apps are expected to protect user data. If your app handles personal information, use proper security measures (encryption, secure transmission) to safeguard it[34]. Never handle sensitive data irresponsibly or exploit user data; doing so can lead not only to rejection but removal from the store.

In short, keep your app safe and civil. Avoid disallowed content categories entirely, and if your app has social features, build robust moderation tools. Apple “is keeping an eye out for the kids” on the platform[35], so err on the side of caution with content. An app that facilitates bullying, crime, self-harm, or any form of exploitation will be pulled immediately.

## Performance: Completeness, Quality, and Efficiency

Under Performance guidelines, Apple checks that your app is fully functional, well-performing, and uses approved technologies. It's not just about not crashing – Apple looks at everything from app completeness to battery usage. Key points include:

- App Completeness: Submit a final, fully functional build – not a beta, demo, or unfinished product[3]. All app features, links, and URLs must work. Remove placeholder text or “coming soon” stubs[36]. Incomplete apps (or those that feel half-built) are among the fastest rejections[37]. Apple explicitly states they will reject apps that are incomplete, crash, or exhibit obvious technical problems[1].
- Use TestFlight for Betas: If you want to distribute test versions, use Apple's TestFlight. Beta or trial apps do not belong on the App Store[38]. Submitting a clearly beta app (with beta labels or test data) to the public App Store will be rejected.
- Accurate and Up-to-Date Metadata: As part of performance, Apple demands truthful marketing. Your app's listing (description, screenshots, previews) must accurately reflect the app's current experience[39][40]. Don't promise features you don't provide, and keep details updated when the app changes. Misleading metadata – e.g. claiming capabilities your app doesn't have, or using screenshots from another app – is grounds for rejection or even account termination[41].
- All new features in an update should be described in the “What's New” notes; significant changes require specific notes (generic phrases like “bug fixes and improvements” won't cut it for major updates)[42].
- Choose correct categories and age ratings for your app. If you misclassify your app's age rating or category, Apple may correct it or reject the app[43][44]. Answer the age rating questionnaire honestly.
- No keyword stuffing or irrelevant metadata: Don't jam your app name, keywords, or description with popular brands or terms not related to your app. Apple may

reject apps that try to game search results or include other app names/trademarks in metadata[\[45\]](#). App names are limited to 30 characters and should be unique and relevant[\[45\]](#).

- **Optimize for All Devices (Including iPad):** Apple strongly encourages iPhone apps to also run well on iPad[\[46\]](#). Wherever possible, design adaptive layouts so your app isn't iPhone-only (unless a specific iPhone hardware feature is required). Supporting all screen sizes and orientations improves your approval chances and user experience. At minimum, test that your UI isn't broken on iPad or different iPhone models. (Note: iPad-only apps or iPhone-only apps are allowed, but if your iPhone app can run on iPad, it should function properly[\[46\]](#).)
- **Efficient Power & Resource Use:** Apps must not misbehave with device resources. Avoid excessive battery drain, CPU hogging, or generating unnecessary heat[\[47\]](#). For example, cryptocurrency mining in background, or encouraging users to do things like cover the device to overheat it, are forbidden[\[48\]](#). Design your app to be efficient and not abuse background operations. If your app uses background modes (audio, VOIP, location, etc.), ensure it's truly needed and appropriately used[\[49\]](#).
- **Use Only Public APIs:** Your app must use Apple's public APIs and SDKs as intended. Using private, unpublished APIs or hacks is an instant rejection[\[50\]](#). Also, apps must run on the latest official iOS/iPadOS version and should avoid deprecated frameworks[\[51\]](#). Keep your code up to date with Apple's developer guidelines to avoid compatibility issues.
- **No Downloading or Executing Unapproved Code:** An app should be self-contained. Downloading and executing new code from the internet is not allowed (this is often called no "dynamic code" rule)[\[52\]](#). There are limited exceptions (for example, educational coding apps that download user-submitted code for learning purposes, if the code is fully visible to the user)[\[53\]](#). But generally, don't try to bypass Apple's review by loading new features or scripts on the fly.

- Network Requirements: Ensure your app works on an IPv6-only network (Apple tests for this)[49]. This has been a requirement for several years – most modern networking libraries handle it, but double-check if you use low-level networking code.
- Web Content and Browsers: If your app includes web browsing, you must use WebKit (Safari's engine) for web content[54]. Apps are not allowed to use their own web rendering engines or extend web content in ways that violate WebKit rules (unless you get a specific entitlement for an alternate browser engine, which is rare)[54].
- No Encouraging Device Hacks: Apps must not encourage users to disable system features or hack their device. For instance, don't ask users to jailbreak, turn off security settings, or restart their device as a requirement[55]. Also, your app shouldn't simulate or interfere with standard UI elements like the volume or mute switch beyond their normal behavior[56].
- Stability and Load Time: While not explicitly written, Apple reviewers will check that your app loads reasonably quickly and remains responsive. Extremely slow load times or unresponsive UI might be interpreted as poor performance or bugs[2]. Optimize launch time and make sure there are no dead-end screens.

In summary, polish your app to perfection before submission. Reviewers will compare your app's actual behavior to your claims. They will reject unstable, unoptimized apps outright. By delivering a *complete*, high-quality app that follows Apple's technical rules, you greatly increase your chances of first-try approval[57][58].

## Business: Monetization and Payment Rules

Apple's Business guidelines govern how you can monetize your app and what content is acceptable. Key principle: if you're making money *through the app*, Apple wants a say (and a cut). Here's how to comply:

- Use In-App Purchase (IAP) for Digital Goods: If your app unlocks features or content digitally (extra levels, premium features, virtual currency, subscriptions, etc.), you must use Apple's in-app purchase system for those transactions[59]. You cannot direct users to external payment methods for content used in the app. Using outside payment mechanisms for digital content is a sure rejection[59][60]. (Physical goods or services to be used outside the app can use other payment methods, but digital content within the app = IAP.)
- Examples: Selling game coins, an e-book PDF, or a premium account upgrade *must* go through IAP. You cannot, say, use PayPal in your app or send users to a web page to buy these. If you try to bypass this, Apple will reject the app for violating guideline 3.1.1[59][60].
- Tip: Also implement a "Restore Purchases" function for IAP so users (and reviewers) can easily restore prior purchases on a new device[7]. Lack of a restore mechanism for IAP is a common mistake that can lead to rejection or at least reviewer frustration.
- Allowed Payment Exceptions: In limited cases, Apple has introduced StoreKit External Purchase Entitlements (as of 2022-2023) that let certain apps link out to external purchase options in specific regions. For example, "reader" apps (content apps like Netflix, Spotify) can allow account signup externally, and certain apps with entitlements can link to a website for purchases[61][62]. However, these require Apple's approval and are limited. Unless you apply and qualify, assume all in-app digital sales must go through IAP. (Note: Apps on the U.S. App Store have slightly different rules due to legal changes – e.g. in the U.S.,

apps may include some external purchase info for content, but this is evolving. Always check Apple's latest stance if this applies to you.)

- No "Payment Gate" Tricks: Don't design your app to mislead users about pricing. Clearly disclose what content or features require a purchase. For instance, if your app is free to download but requires a subscription or in-app purchase for full use, make that clear in the description and within the app. Apple will reject apps that hide fees or make it hard to figure out pricing[63][64]. A common rejection example: labeling your app "Free" or saying "free trial" but then immediately forcing purchase without clarity[65].
- Fair Pricing: You set your own app price or IAP prices, but don't try to scam users with irrationally high prices for low value. Apple can reject apps that appear to "cheat users" with overpriced content[66]. If you're charging a lot, be sure your app delivers value to justify it.
- No Manipulative Monetization: Guidelines forbid apps from engaging in fraudulent or manipulative monetization schemes. For example, loot boxes or any randomized in-app purchases must disclose the odds of winning each item before purchase[67]. Also, you cannot incentivize users to review your app or download another app in exchange for benefits – any such attempt to manipulate App Store rankings or reviews (including buying fake reviews) will get your app rejected and can get you banned from the Developer Program[68].
- Acceptable Business Models: Some special cases are allowed if done right:
- Apps can display or promote your other apps (cross-promotion) as long as your app isn't just a catalogue of apps[69].
- Aggregator apps (that gather third-party content) need to provide real value (like editorial content) so they're not just a storefront of others' apps[70].
- Cryptocurrency: Wallet apps and crypto exchanges are allowed, but must follow applicable laws. However, apps may not mine cryptocurrency in the background on the device[71].

- NFTs: Apps may facilitate sale of NFTs (non-fungible tokens) via IAP and allow viewing NFTs. But simply owning an NFT shouldn't unlock in-app features unless that happens through an IAP mechanism[72]. And apps can't direct users to external purchase mechanisms for NFTs except in allowed regions[73].
- If you offer personal loans or financial services, there are very specific rules (e.g. loan apps must disclose APR% and cannot have sky-high rates >36% or demand repayment < 60 days)[74]. Ensure compliance with those if relevant.
- Unacceptable Business Models: Apple will reject:
  - Apps that replicate the App Store or act as a mini app marketplace for third-party apps/plugins[75] (except approved enterprise app catalogs, etc.). Your app shouldn't look like an app store within an app.
  - Apps that exist mainly to display ads or make users click ads (ad farms)[76].
  - Apps that pay users for downloads or social sharing (e.g. "download our app for rewards") violate Apple's policies on artificial promotion[77].
  - Charitable fundraising within the app is restricted: unless you are a registered nonprofit using Apple Pay, you cannot solicit donations inside the app (you'd have to direct to a web/SMS for donations)[78].
  - Apps that force users to rate the app or perform App Store actions to unlock functionality are not allowed[79]. You can ask for a rating (using the official API) but you can't make, say, posting a 5-star review a requirement to use features.
  - Subscriptions: If you offer auto-renewable subscriptions, you must follow rules on transparency: clearly let users know the subscription terms, what they get, length of subscription, and how to manage/cancel. Also, subscriptions must provide ongoing value (no "subscribe for nothing" schemes)[80][81]. Free trial periods are allowed, but you must communicate what happens after the trial (e.g. charges) and you can implement a free trial via IAP free tier as Apple outlines[82]. Ensure your subscription works across the user's devices and they get what they pay for without extra steps[80][83].

- No Lottery or Gambling without Permissions: Apps offering real-money gambling, betting, lotteries, etc. have strict requirements (they must be legal in all regions offered, geo-restricted appropriately, free to download, not use IAP for credit, etc.)[\[84\]](#)[\[85\]](#). If you're not specifically in that space, avoid it; if you are, be prepared with proper licenses and implementation.

The bottom line for business: play by Apple's rules when making money. Use in-app purchase when required, be transparent with users about costs, and don't try to trick or pressure users for payments. If your monetization looks scammy or sidesteps Apple's systems, your app won't be approved[\[60\]](#)[\[66\]](#).

## Design: User Experience and Interface Guidelines

Apple is renowned for its design standards – and expects apps on its platform to reflect a high-quality user experience. The Design guidelines cover everything from originality to minimum app functionality and proper use of system features:

- Follow Human Interface Guidelines: While not explicitly a rule that “violating HIG = rejection,” it’s strongly implied that apps should be intuitive, easy to use, and polished[86]. Clunky interfaces, ugly design, or confusing navigation increase the risk of rejection (or at least poor featuring). Test your UI on different devices and orientations, fix layout issues, and ensure text is readable. Apple wants apps that feel at home on iOS – pay attention to things like using proper iOS UI controls, spacing, etc.
- No Copycats or Clones: Your app should be original. Don’t just clone an existing popular app or rip off another developer’s idea/interface[87]. Copycat apps (even if not outright IP infringement) make the App Store worse, and Apple will reject them[87]. This includes not using another app’s icons or names to mislead users[88]. Impersonating other apps or companies violates the Developer Code of Conduct and can get you banned[89].
- Minimum Functionality (No “Wrapped” Websites): Apps need to feel like real, standalone apps, not just a thin shell for a website or a single piece of media. Apple’s guideline 4.2 clearly states that an app must provide utility or entertainment value beyond what a web page or a book or a song would[90]. If your app is basically just your website in a webview, or a few buttons that link elsewhere, it may be rejected as having insufficient value. Ensure your app has native features, interactivity, or content that justifies being an app.
- For example, simply packaging a blog’s RSS feed or a web store in an app with little enhancement is risky. Add native navigation, push notifications, offline features – something to elevate the experience beyond a web browser. Apps that

are primarily marketing materials, content aggregators, or collections of links with minimal native functionality will likely be rejected[91].

- There is a known rule: apps that are just a song, a video, or a PDF should go to the appropriate store (Apple Music, Apple Books), not the App Store[92].
- Avoid Spam and Template Apps: Don't flood the App Store with multiple similar apps. If you have just slight variations (e.g. white-label apps for multiple clients or the same app re-skinned for different topics), consider a single container app with in-app content selection. Apple will reject spam submissions and may consider it an attempt to game the system[93]. Also, if you use an app generation service, make sure the result is truly custom – apps that are obviously made from a generic template with no customization are often rejected[94].
- Proper Use of Notifications and System Features: Push notifications should be used respectfully. Apple forbids using push for spam, advertising, or any purpose users didn't consent to[95]. Users must opt-in to any promotional pushes, and there must be an in-app option to opt-out[95]. Also, your app should not require enabling push just to function[96]. Use notifications for meaningful updates, not as a marketing bullhorn (unless users explicitly agree).
- Login Options (Sign in with Apple): If your app offers third-party social logins (Google, Facebook, etc.) as the *only* way to log in, you must also offer Sign in with Apple or an equivalent privacy-friendly option[97][98]. Apple wants users to have a choice that doesn't force sharing personal data. (Exceptions exist for apps that are exclusive clients of a service – e.g. a Google app can require Google login – but most apps need to comply by providing an alternative)[99].
- Don't Abuse Apple Icons/Emoji: Your app's design shouldn't misuse Apple copyrighted material. For instance, don't use Apple's emoji in your app's UI in a way that implies they're your graphics (Unicode emoji that render as Apple style are allowed, but you can't, say, use the actual Apple Emoji images in a custom way)[100]. Don't design your app to look confusingly like an Apple system app or mimic iOS UI elements in a misleading way[101].

- Augmented Reality (ARKit) and New Tech: If using ARKit or other new frameworks, make sure you use them fully. An example from Apple: an AR app that just drops a static 3D model on camera with no interactivity might be deemed too minimal and not accepted[102]. So leverage new technologies to create rich experiences.
- Keep UI Consistent with Metadata: As mentioned earlier, your screenshots and description must match the actual app UI. If you update your app's design, update your screenshots too. Reviewers have flagged apps for having mismatched UI in their marketing versus the build[103][104]. Treat your App Store description as a “contract” – every claim in it should be truthful and verifiable in the app[105].
- User Experience Matters: Apple wants apps that feel good to use. If your app has confusing flows, broken links, or an ugly interface, it might not technically violate a written rule but can still lead to rejection under the catch-all of not providing a “high-quality experience”[106][107]. In fact, Apple has guideline 5.6.4 (recently cited in a rejection) that *apps must provide the high-quality experience users expect*[108]. So focus on quality: correct any typos, ensure your content is not placeholder text, and make navigation logical. Test with users if possible.

Ultimately, design your app to meet Apple's high standards of quality and originality. A well-designed app not only avoids rejection but also delights users (and Apple's editorial team, if you aim to be featured). Don't give Apple a reason to say your app is subpar or a duplicate of something else[109][110].

## Legal: Privacy, Data Protection, and Other Regulations

The Legal section of Apple's guidelines is about complying with laws and protecting users' rights. Many of the most common rejections in recent years are due to privacy issues, so pay special attention here. Key legal requirements:

- **Privacy Policy Required:** Every app must have a privacy policy. You need to provide a URL to a privacy policy in App Store Connect, and the policy must also be accessible within the app (for example, in a Settings menu)[111][112]. The privacy policy should clearly explain what data your app collects (if any), how that data is used, and how users can contact you or request deletion[113][114]. *Note:* Hiding or forgetting the privacy policy link is an easy way to get rejected – Apple checks for it[10].
- **User Data & Permissions:** If your app collects user data or accesses sensitive resources (camera, contacts, location, health data, etc.), you must ask for permission and explain why. The permission request message (NSPrivacyUsageDescription in your app) needs to clearly justify the need (“This app requires location to show local weather updates,” for example)[115][116]. Apps that request personal data or device data without a clear reason, or before the user understands the benefit, often get rejected[117][118]. Also, *never* require users to agree to share data or enable tracking just to use the app – features not relevant to core app functionality should be optional[119][120].
- **Data Minimization:** Only ask for data that is truly needed for your app's core function[121]. For instance, if your app just needs an email for account creation, don't also request access to Contacts or Photos without justification. Unnecessary permission requests raise red flags.
- **Access and Control:** If a user declines a permission, your app should still function (perhaps with limited features) and not repeatedly nag or attempt to trick them into enabling it[119][122]. Provide alternative ways if possible (e.g. allow manual entry of location if GPS is off).

- Account Deletion: As of late 2021/2022, if your app allows account creation, it must allow users to initiate deletion of their account from within the app[\[123\]](#)[\[124\]](#). This is now enforced. The option should be easy to find (e.g. in account settings) and actually process account/data deletion per applicable laws.
- App Tracking Transparency (ATT): If your app tracks users or shares data that could be used to track, you must implement the ATT prompt (requesting permission to track across apps) and respect the user's choice. If the user says no, you cannot use IDFA or similar tracking. Apple strictly enforces this. Apps that circumvent or ignore ATT will be rejected or removed[\[125\]](#)[\[126\]](#).
- Data Sharing and Third-Party Code: You must disclose what data is shared with third parties and ensure any third-party SDKs in your app (analytics, ads, etc.) adhere to equivalent privacy standards[\[127\]](#). For example, if you use an analytics SDK that collects user info, mention this in your privacy policy and ensure the SDK isn't doing more than you're aware of. Also, do not share data with third parties without explicit user consent, unless it's for a permitted use[\[125\]](#).
- Personal Data Protection: Apps should never harvest passwords or private data from the user's device illegitimately[\[128\]](#). Also, if your app compiles personal information from public sources, be careful – if users haven't given it directly and explicitly, you may be violating guidelines[\[129\]](#).
- Legal Entity for Regulated Content: If your app is in a highly regulated domain – e.g. banking/financial services, healthcare, gambling, cryptocurrency exchange – Apple expects the app to be submitted by the company or institution with proper licenses, not an individual developer with no authority[\[130\]](#). For instance, a banking app should come from the bank itself. If you are aggregating or offering such services, make sure you are authorized.
- Intellectual Property: Ensure everything in your app (code, content, images, music) is either your own or you have rights to use it[\[131\]](#)[\[132\]](#). Copyright and trademark infringement is a quick rejection. Don't use famous characters, logos,

or content you don't own. Even your app's name and icon should not violate others' trademarks or closely resemble other apps[\[132\]](#)[\[133\]](#). If your app integrates content from third-party services (YouTube videos, for example), check those services' terms of use and get permissions if needed[\[134\]](#).

- **No Illegal Downloads:** Apps must not facilitate piracy or unauthorized downloads of media. For example, an app that lets users download YouTube videos or music from Apple Music without permission will be rejected[\[135\]](#).
- **Don't Imply Apple Endorsement:** Avoid any wording or imagery that suggests Apple endorses your app or that Apple is the source of your content. (E.g. don't use Apple logos in your screenshots or say "Apple recommends...")[\[136\]](#). Also, don't make your app look like Apple's own apps – no mimicking the App Store interface or iMessage design, etc., in a way that confuses users[\[101\]](#).
- **Gambling, VPNs, and Other Regulated Services:** If your app includes gambling (real money wagering) or contests, you must provide clear rules and ensure it's legal in all locales you target[\[137\]](#). Gambling apps require geo-restriction to legal jurisdictions and cannot use in-app purchases for currency[\[84\]](#). For VPN apps, Apple requires you use the proper VPN API and that only companies (not individual dev accounts) publish VPN apps[\[138\]](#)[\[139\]](#). VPN apps must not log or sell user data and, in some regions, you must provide a license number in the review notes if a license is required locally[\[139\]](#). Similarly, Mobile Device Management (MDM) apps have to be submitted by reputable companies (like enterprise solution providers or schools), and must strictly adhere to user data protection rules[\[140\]](#)[\[141\]](#).
- **Legal Compliance by Region:** You are responsible for ensuring your app complies with all local laws in regions you distribute. For example, privacy laws (GDPR in Europe, etc.), export regulations (encryption use might require declaring compliance), and so on. If certain content is illegal somewhere, use Apple's tools to geo-restrict your app's availability. Apple's guideline is clear: apps that solicit or

facilitate criminal activity will be rejected, and in extreme cases (e.g. aiding child exploitation) Apple will contact authorities[142][143].

In summary, respect user privacy and legal boundaries at all times. The App Store has increasingly tightened privacy rules – *privacy violations are now the #1 cause of rejections* in many cases[144]. If your app is transparent with users, forthright about data usage, and diligent about permissions and policies, you'll avoid the most common legal pitfalls. Always err on the side of user consent and control: let users know what you're doing, and give them choices (opt out, delete account, etc.). And of course, never include anything illegal, illicit, or copyrighted without permission in your app[145][135].

## Common Reasons for Rejection (and How to Avoid Them)

Even with the guidelines above, it helps to know the typical mistakes that cause Apple to reject apps. Here are some of the most frequent rejection reasons developers face – make sure your app doesn't fall into these:

- **Crashes and Bugs:** The app crashes during testing, freezes, or exhibits major bugs. *Solution:* Test thoroughly on multiple devices/OS versions and fix all crashes. Even a single crash on review can fail your submission[2].
- **Incomplete App / Broken Features:** Some functionality is broken, not implemented, or requires configuration the reviewer doesn't have. For example, a login screen with no demo credentials, or a feature that says "coming soon." *Solution:* Finish all core features and remove or hide anything not ready. Provide test login info or demo mode for any gated sections[1][6]. Ensure all links and buttons work.
- **Long Load Times or Poor Performance:** If your app takes too long to load or certain content never appears (maybe due to network timeouts), the reviewer might give up. *Solution:* Optimize your loading and handle poor network conditions gracefully (use spinners, error messages). Test on slow connections – Apple often reviews under suboptimal conditions to catch issues[146].
- **Privacy Violations:** Not providing a privacy policy, requesting permissions without justification, or attempting to track users without consent (violating ATT). *Solution:* Include a clear privacy policy URL and in-app link[10]. Only request necessary permissions and always include usage descriptions explaining why. Implement the App Tracking Transparency prompt if you use tracking. Don't collect more data than needed[144][115].
- **Misleading or Insufficient Metadata:** The App Store description or screenshots don't match the app, or are incomplete. E.g., you advertise features that don't exist, or your screenshots show functionality that isn't in the app. *Solution:* Keep your marketing honest and up-to-date[41][5]. Also, ensure your screenshots are

actual app screens (Apple forbids fake UI in screenshots). If your app requires a login or location access to show its main content, mention that so the reviewer knows what to expect.

- Payment/IAP Issues: Common sub-issues include: no “Restore Purchases” for IAP, using external payment links, subscription info not clear, or offering digital goods without using IAP. *Solution:* Review all monetization flows. Provide a restore mechanism[7]. Make sure all purchases of digital content use Apple’s IAP[59]. If your app has subscriptions or a free trial, clearly describe how it works and ensure prices match between the app and metadata[147][148].
- Lack of Lasting Value (Spam/Copypat): Apple might reject apps that seem overly simple, single-use, or duplicative of others – *even if they function*. For example, yet another flashlight app or a basic app that just loads a web page might be deemed not valuable enough. Or an app that users pay for but provides trivial content might be flagged (as in a real case where paid content was judged not to have long-term value)[149]. *Solution:* Ensure your app has unique, useful features. If it’s a content app, update it regularly or provide enough content to engage users long-term. Avoid cookie-cutter app templates.
- Missing App Content for Review: Sometimes apps get rejected because the reviewer couldn’t access parts of the app. Examples: the app is region-locked or requires a device-specific accessory and the reviewer had no way to test; or the app requires an invite code with none provided. *Solution:* If your app requires special conditions (certain location, hardware, login), *explain this in the review notes and provide a way to test*. You can temporarily demo features or provide test hardware if needed. Always make the reviewer’s job easy[17] – if they can’t experience your app, they can’t approve it.
- UI/Design Flaws: If your app’s interface is very poor – e.g. lots of misspelled words, overlapping text, or it doesn’t adapt to different screen sizes – Apple may reject it as not meeting the standards. *Solution:* Polish your UI. Check for layout

issues on various devices (especially iPad). Fix typos and ensure your app isn't just a rough draft design-wise[\[150\]](#)[\[107\]](#).

- **Violating New Rules (Not Staying Updated):** Apple occasionally adds new requirements (for example, the account deletion requirement, or rules around cryptocurrency/NFTs, or AI transparency). If you're unaware, you might violate them. *Solution:* Keep an eye on Apple's developer news and update your app accordingly. For instance, in 2025 Apple emphasized privacy labels, AI disclosures, and accessibility – missing these considerations can hurt your review[\[151\]](#)[\[152\]](#). Always read the latest App Store Review Guidelines before submission, since Apple can change them anytime[\[153\]](#)[\[154\]](#).
- **App Store Connect Mistakes:** Sometimes the issue isn't the app binary but how you fill out the submission in App Store Connect. Missing info in the "Content Rights" section (if your app contains third-party content, you must declare you have rights), incorrect age rating questionnaire answers, or forgetting to provide an encryption export compliance info (if your app uses encryption) can all delay approval. *Solution:* Double-check every App Store Connect field. If you use encryption beyond routine HTTPS, answer the export compliance questions. If you have any user-generated content or other people's content, mark the appropriate checkboxes and provide documentation if needed. Provide detailed review notes whenever you think something might be unclear.

If your app *does* get rejected, Apple will cite a guideline number in their message.

Carefully read their feedback, fix the issue, and resubmit. You can also politely appeal if you believe a rejection is a misunderstanding (use the App Review Appeal process)[\[155\]](#) – but in most cases, it's faster to address the problem. Remember, each resubmission goes back in the queue, so it's best to get it right the first time.

## Final Tips for a Smooth Approval

Passing App Store review is achievable if you plan ahead and pay attention to detail.

Keep these best-practice tips in mind:

- Plan for Review from Day 1: It's easier to build your app with Apple's guidelines in mind than to retrofit fixes later. From the start, design your app to meet content, design, and privacy standards[\[57\]](#)[\[154\]](#).
- Use App Store Connect Effectively: Fill out all required info, upload accurate screenshots, and monitor your submission. If Apple needs clarification (they might send you questions via Resolution Center), respond promptly. Ensure your contact info is current in case they reach out[\[156\]](#).
- Document Special Cases: If your app uses innovative tech (AI, AR, etc.), consider adding a note on how you use it responsibly (e.g. "AI chatbot with no deceptive intent, content moderated"). If you use AI to generate content or recommendations, be transparent about it to both users and Apple[\[157\]](#).
- Focus on Accessibility: Apple increasingly values accessible apps. While not usually a cause for outright rejection, having VoiceOver support, scalable fonts, and good contrast can only help your app's perception. In fact, Apple's 2025 updates highlight accessibility – things like text that's too small or poor contrast could be flagged during review[\[158\]](#)[\[159\]](#). It's good practice (and good for users) to meet basic accessibility guidelines.
- Stay Up-to-Date: Apple's rules evolve. Check the official App Store Review Guidelines regularly, especially before major iOS releases or if you've been developing for a long time without submitting. What was acceptable a year ago might have subtle changes now (for example, new rules around social login, data sharing, or ad tracking). Following Apple's developer news and documentation will keep you ahead of surprises[\[160\]](#).
- Be Patient and Polite: App review can take time (usually around 1-3 days, but sometimes longer). Don't spam the review team with messages or

resubmissions; that won't help. If it's taking unusually long and you have a hard deadline (like a launch event), you can request an expedited review, but use that sparingly. And if you communicate with App Review, always be professional and clear.

- **Learn from Others:** Many developers share their App Store rejection stories in blogs and forums. It's useful to read these (Apple's guidelines can be a bit general, so real cases shed light on how rules are applied). Knowing, for example, that writing "Download now for free – limited time!" in your app description is considered misleading marketing can save you a rejection. In short, leverage community knowledge on App Store reviews.

By ensuring your app is complete, compliant, and user-friendly, you greatly improve your chances of approval. Apple's review might seem strict, but it's ultimately about maintaining a high-quality App Store. As Apple themselves put it: they want apps that are safe, useful, high-quality, and respectful to users<sup>[161][162]</sup>. If you deliver that, you'll likely get that coveted "Approved" status and be on your way to release. Good luck with your app submission, and welcome to the App Store!

## Sources:

- Apple App Store Review Guidelines (Apple Developer)[163][1]
- Apple Developer Documentation and News[164][155]
- NextNative App Store Review Checklist (2025)[165][10]
- App Store Review Tips – AppFollow Blog (2024/2025)[65][148]
- Common Rejection Reasons – Twinr & Others[144][107]
- Netscape Labs – App Store Guidelines Update 2025[151][58]

---

[1] [3] [4] [6] [12] [14] [18] [19] [20] [21] [22] [23] [24] [25] [28] [29] [30] [31] [32] [33] [34] [35] [36] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [59] [61] [62] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [109] [111] [112] [113] [114] [115] [116] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137] [138] [139] [140] [141] [142] [143] [145] [155] [156] [163] [164] App Review Guidelines - Apple Developer

<https://developer.apple.com/app-store/review/guidelines/>

[2] [5] [13] [63] [64] [65] [103] [104] [105] [106] [108] [147] [148] [149] iOS App Store Review Guidelines: How to Pass & Avoid Rejection

<https://appfollow.io/blog/app-store-review-guidelines>

[7] [8] [9] [10] [11] [15] [16] [17] [26] [27] [37] [146] [165] App Store Review Guidelines (2025): Checklist + Top Rejection Reasons

<https://nextnative.dev/blog/app-store-review-guidelines>

[57] [58] [60] [151] [152] [153] [154] [157] [158] [159] [160] [161] [162] Apple App Store Review Guidelines Update 2025

<https://netscapelabs.com/2025/08/27/apple-app-store-review-guidelines-update-2025-what-developers-really-need-to-know/>

[107] [110] [117] [118] [144] [150] Apple App Store Rejection Reasons In 2025 (And Fixes)

<https://twinr.dev/blogs/apple-app-store-rejection-reasons-2025/>