

What this is: A submission playbook for your Expo / React Native app on the Apple App Store, Google Play Store, and (bonus) Amazon Appstore. The strategy below uses the **Reader App Model** — a fully store-approved approach that lets you keep 100% of subscription revenue (minus Stripe's 2.9% + \$0.30) by routing all signups through your website.

■ Strategy: The Reader App Model

The plan in one sentence: Apps are listed as FREE downloads on both stores. Users sign up + pay on *thegettingaheadguidebook.com* via Stripe. The app itself only handles login + day-to-day use. Apple and Google both explicitly allow this model — it's how Netflix, Spotify, Kindle, Dropbox, YNAB, Copilot Money, and Monarch Money distribute their iOS / Android apps.

How each type of user experiences it

	Existing customer	Brand-new prospect
Step 1	Downloads free app from store	Downloads free app from store
Step 2	Sees login screen → enters credentials	Sees login screen with soft prompt: <i>"New here? Create your account at thegettingaheadguidebook.com"</i>
Step 3	✓ Full access	Opens website, signs up, pays via Stripe
Step 4		Returns to app, logs in → ✓ Full access

Key insight: The "sign up at the website" friction only hits BRAND-NEW prospects who discovered you via app store search alone. Existing customers who heard about you from your blog, YouTube channel, Reddit, or email list already signed up on the website FIRST — they're just downloading the app to use on the go. For them, the experience is exactly what they'd expect.

Why this is the recommended path

- **You keep 100% of subscription revenue** (minus Stripe's 2.9% + \$0.30) — no Apple 15–30%, no Google 15–30%
- **No StoreKit / Google Play Billing code complexity** — just your existing Stripe flow on the web
- **One refund process, one customer dashboard, one source of truth** — all in Stripe
- **Pricing changes don't require app review** — change a price on the website and it's live in seconds
- **Full marketing visibility** — abandoned cart emails, A/B tests, attribution data, etc. (Apple and Google give you none of this)
- **Fully compliant with BOTH stores** — zero risk of rejection or de-platforming



Apple's Specific Rules for Reader Apps

Apple's App Store Guideline 3.1.3(a) explicitly carves out "Reader Apps" — free apps that let users access content or services they purchased outside the app. Finance apps qualify. The catch: **you cannot mention pricing, billing, subscriptions, or free trials ANYWHERE in the iOS app.**

Things you CAN do in the iOS app

- Show a login screen
- Say "New here? Visit our website to learn more."
- Include an **External Link Account Entitlement** link to your pricing page (Apple now allows ONE such link — application required)
- Let users manage their existing subscription via a link to Stripe's customer portal
- Use the full app once a user is logged in
- Show "Forgot password" reset flow (in-app or web)

Things you must NOT do in the iOS app

- Show pricing ("\$.95/mo" anywhere)
- Show a "Start Free Trial" button that goes to your website (Apple historically rejects this — use "Sign In" instead and let users find signup elsewhere)
- Show a "Subscribe" button of any kind
- Mention "7-day free trial" copy
- Show feature comparison tables that mention plan tiers
- Process payments through any system other than Apple IAP

Code Changes Your Existing App Needs

Your app already has a Pricing screen and a Signup screen. On iOS, both must be hidden. The cleanest way in Expo is platform-conditional routing — when the user opens the app on iOS, they get routed straight to the Login screen with NO option to navigate to Pricing or Signup.

Step 1: In `/app/frontend/app/_layout.tsx`, conditionally hide Pricing and Signup routes on iOS:



```
import { Platform } from "react-native";

// Inside your Stack.Screen list:
{Platform.OS !== "ios" && (
  <>
    <Stack.Screen name="pricing" />
    <Stack.Screen name="signup" />
  </>
)}
```

Step 2: In `/app/frontend/app/login.tsx`, conditionally swap the "Don't have an account? Sign up" link for an iOS-friendly prompt:

```
import { Platform, Linking } from "react-native";

{Platform.OS === "ios" ? (
  <Text style={s.helper}>
    New to The Getting Ahead Guidebook?{" "}
    <Text
      onPress={() => Linking.openURL(
        "https://thegettingaheadguidebook.com"
      )}>
      Visit our website to learn more.
    </Text>
  </Text>
) : (
  // Existing Android/web "Sign up" link
)}
```

Step 3: Hide any "View Pricing" or "\$8.95/mo" text elsewhere in the app on iOS (e.g., expired-trial screens):

```
// Anywhere pricing text appears:
{Platform.OS !== "ios" && (
  <Text>$8.95 / month – cancel anytime</Text>
)}
```

Ask your developer to scan for these strings before iOS submission: "\$8.95", "\$79", "Subscribe", "Free trial", "7-day trial", "Pricing", "Plan". Each one needs to be wrapped in `{Platform.OS !== "ios" && ...}` or moved out of the user flow entirely on iOS.



Submission — Apple App Store (iOS)

Step	Action
1	Apple Developer Program enrollment (\$99/yr). LLC accounts need a D-U-N-S number (free, 1–2 wks to issue).
2	Bundle ID in App Store Connect: com.thegettingaheadguidebook.app
3	Apply Reader-App code changes above. Test on a real iOS device — NO pricing/signup screens reachable.
4	Build via EAS: eas build --platform ios --profile production
5	Upload via: eas submit --platform ios
6	TestFlight first — invite 5–10 testers for 7+ days
7	Fill out App Store Connect metadata: description, screenshots, keywords, privacy nutrition labels
8	Submit for App Review — typically 24–72 hrs. Be ready to respond to reviewer questions within 24 hrs.

Submission — Google Play (Android)

Step	Action
1	Google Play Developer account (\$25 one-time fee)
2	Package name: com.thegettingaheadguidebook.app (NEVER change this once published)
3	Google Play is more flexible on Reader Apps than Apple. You CAN show pricing on Android — but for consistency I recommend keeping the same restricted UI as iOS.
4	Build signed AAB: eas build --platform android --profile production
5	Internal testing track first — invite up to 100 testers, no review delay
6	Closed testing track — at least 20 testers for 14 days (REQUIRED for new dev accounts as of 2023)
7	Production track — 1–7 days review
8	Fill out Play Console: store listing, screenshots, feature graphic (1024×500), data safety form, content rating

■ Bonus: Amazon Appstore (Kindle Fire)

Once you have a signed Android AAB, you can submit the same binary to Amazon Appstore for free — reaching ~3% of mobile users (Kindle Fire tablets and Amazon-powered Android devices). Amazon is the most flexible of the three stores on payment rules.

Step	Action
1	Sign up at developer.amazon.com (free, no \$25 like Google)



Step	Action
2	Upload your same .aab or .apk from EAS
3	Fill out listing (mostly identical to Play Store fields)
4	Submit for review — typically 1–3 business days

Worth it for a solo founder? Yes — 20-min setup, \$0 cost, marginal extra reach into a demographic that skews older/family-oriented (your exact target).



Pre-Submission Checklist (All Stores)

- App icon — 1024×1024 PNG, no transparency, no rounded corners
- Privacy policy URL (live on your domain)
- Terms of service URL
- Account deletion flow (Apple Guideline 5.1.1(v), Google Play target API 33+)
- Support email or contact URL
- Screenshots — at least 3 per device size, real app content only
- Short description (80 chars) and long description (~4,000 chars)
- Keywords / tags (think: "family budget envelope sinking fund cash flow")
- Content rating questionnaire answered
- Tested on a REAL device, not just the simulator
- Bumped versionCode (Android) / buildNumber (iOS) above last upload
- **iOS-specific:** pricing/signup screens hidden via Platform.OS
- **iOS-specific:** scanned codebase for "\$8.95", "Subscribe", "Free trial" strings
- Privacy nutrition labels (iOS) / Data safety form (Android) filled out accurately

Top Rejection Reasons & How to Avoid Them

Apple App Store

Guideline	What triggers it	How to avoid
3.1.1 (IAP)	Charging for digital service without Apple IAP	Use Reader App model — no pricing or signup in iOS app
3.1.3(a)	Reader App showing prices or signup CTA	Hide all pricing strings on iOS. Use Login-only screen.
4.3 (Spam)	App looks too similar to competitors	Differentiate screenshots + description; emphasize unique angle
5.1.1 (Privacy)	Missing privacy policy or requesting unused permissions	Verify privacy policy URL + audit Info.plist entries
5.1.1(v) (Deletion)	No in-app account deletion	Already implemented in your app — verify it actually works
2.1 (Functionality)	Crashes on launch, login broken, core feature fails	Test fresh install on real device before submitting
2.3 (Metadata)	Screenshots show features that aren't in the app	Use literal app screenshots, no marketing puffery



Google Play

Issue	What triggers it	How to avoid
Data safety mismatch	Declared data collection doesn't match what app does	Audit AsyncStorage / cookies / analytics before filling form
Target API too low	2026 requires API 34+ targeting	Run eas doctor to verify before building
Permission overreach	Requesting unused permissions	Strip unused permissions from app.json
Account deletion missing	No in-app deletion flow	Already implemented — verify it works
Misleading screenshots	Screenshots that aren't from the running app	Use literal app screenshots
Financial sensitivity	Requires extra disclosure for any bank data reading	Currently fine — you don't read bank data. Watch this if you add Plaid later.



Versioning Strategy

Every Play Store / App Store upload needs a higher version number than the previous one. (Sideload-only APK distribution does NOT have this requirement — only public store uploads enforce it.)

Platform	Field in app.json	Rule
iOS	expo.ios.buildNumber (string)	Must increase numerically: 1 → 2 → 3 → ...
Android	expo.android.versionCode (int)	Must increase: 14 → 15 → 16 → ...
Both (user-facing)	expo.version (semver)	Bump for releases: 1.1.0 → 1.2.0 for features, 1.1.0 → 1.1.1 for bugfixes

After Submission — First Week

- Set up email alerts for review status changes (both stores)
- Have a landing page ready for the day it goes live
- Schedule announcement email to your existing list
- Reddit/social posts ready to drop (don't pre-announce — surprise launch performs better)
- Plan first-week update: collect crash reports + reviews, ship a v1.0.1 bugfix release within 7 days
- Monitor Sentry / crash reporting daily for the first 30 days
- Reply to EVERY app store review (positive and negative) within 24 hrs — it correlates strongly with rankings
- Email customers a "We're on the App Store!" announcement so they download it (and rate it 5 stars)

Quick Reference Summary

	iOS	Android	Kindle Fire
Store	Apple App Store	Google Play	Amazon Appstore
Developer fee	\$99/yr	\$25 one-time	Free
Reader App allowed?	Yes (recommended)	Yes (recommended)	Yes
Payment in app?	Never	Never (for consistency)	Never
Pricing visible in app?	■ No	■ OK but hide for consistency	■ OK
Platform cut	0% (Reader App)	0% (Reader App)	0% (Reader App)
Review time	24–72 hrs	1–7 days	1–3 business days



Bottom line: Free downloads on all three stores. All signups + payments route to your website. You keep 100% of subscription revenue (minus Stripe's 2.9% + \$0.30). Zero platform fees. The Reader App model is battle-tested by every major SaaS — and it's how YNAB, Copilot Money, and Monarch all work today.