A Streamlit app is a **web app**, not a native iOS/Android app, so you can’t ship it to the **Apple App Store** or **Google Play** as-is. You have three practical paths:

### **1) Ship it as a web app (fastest)**

* Host the Streamlit app (Streamlit Cloud / Spaces / your server) and share a URL or wrap it as a **PWA** for “Add to Home Screen.”
* **Pros:** quickest MVP; no store review. **Cons:** not discoverable in stores; limited native features.

### **2) Wrap it in a native shell (interim)**

* Use **Capacitor/Cordova** (WebView) or **React Native (Expo) + WebView** to load your hosted Streamlit UI inside a native container, then submit to stores.
* **Caveats:** Apple may reject “websites in a wrapper” without sufficient native value; mobile UX may feel cramped; file uploads and caching need testing.

### **3) Build a proper mobile app (recommended for store)**

* Keep your scoring logic in a small **API** (e.g., **FastAPI** service running the CleanScore code).
* Build a native front end (**React Native/Expo** or **Flutter**) that calls that API.
* **Pros:** best chance of approval, performance, offline UX, push, analytics. **Cons:** more engineering effort.

## **Minimum checklist for App Store/Play**

* **Distinct native value:** real mobile UI (not just an embedded webpage).
* **Privacy Policy & Data handling:** link in-app; explain uploads/processing.
* **Account/identity & support flow** if you collect user data.
* **No broken flows on mobile:** file upload, tier thresholds, downloads (export to email/Share sheet or cloud).
* **Branding & metadata:** app name, icons, screenshots, description; comply with **App Store Review Guidelines** and **Play Developer Program Policies**.
* **Billing (if any):** use in-app purchase rules (Apple) or compliant alternatives.

## **A pragmatic plan**

1. **MVP now:** deploy Streamlit on the web.
2. **Stabilize backend:** extract CleanScore logic into a **FastAPI** endpoint.
3. **Mobile UI:** build a lightweight **React Native** app calling that API; add native file picker and share/export.
4. **Compliance pass:** privacy policy, data retention, support email, accessibility checks; submit to stores.

If you want, I can sketch the FastAPI endpoint (Python) and a matching React Native screen (Expo) that hits it—so you have a clean upgrade path from your current Streamlit MVP to a store-ready app.