File: client/src/hooks/use-ads.ts

Type: ts | **Size:** 1869 bytes

```ts
import { useState, useEffect, useCallback } from 'react';
import {
  incrementActionCount,
  initAdManager,
  isAdFree,
  setAdFree as setAdFreeManager
} from '@/lib/adManager';
import { showInterstitialIfReady, trackAction as trackNativeAction } from '@/lib/adMobService';
import { useCapacitor } from './use-capacitor';
/**
 * Hook for managing ad display in the application
 * Handles both web and native (Android/iOS) ad implementations
 */
export function useAds() {
  const [showInterstitial, setShowInterstitial] = useState(false);
  const [adFree, setAdFree] = useState(false);
  const { isCapacitorApp, isInitialized } = useCapacitor();

  // Initialize ad manager
  useEffect(() => {
    initAdManager();
    setAdFree(isAdFree());
  }, []);

  /**
   * Record a user action that may trigger an interstitial ad
   */
  const trackAction = useCallback(() => {
    if (adFree) return false;

    // If running in Capacitor and initialized, use native ads
    if (isCapacitorApp && isInitialized) {
      // Use native implementation for showing interstitials
      showInterstitialIfReady();
      return true;
    } else {
      // For web, use the web implementation
      // Check if we should show an interstitial
      const shouldShowAd = incrementActionCount();
      if (shouldShowAd) {
        setShowInterstitial(true);
        return true;
      }
    }

    return false;
  }, [adFree, isCapacitorApp, isInitialized]);

  /**
   * Close the interstitial ad
   */
  const closeInterstitial = useCallback(() => {
    setShowInterstitial(false);
  }, []);

  /**
   * Update the ad-free status
   */
  const updateAdFree = useCallback((status: boolean) => {
    setAdFree(status);
    setAdFreeManager(status);
  }, []);

  return {
    showInterstitial,
    trackAction,
    closeInterstitial,
    adFree,
    updateAdFree
  };
}
```