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

Type: ts | **Size:** 2640 bytes

```
import { useState, useEffect } from 'react';
import { Device } from '@capacitor/device';
import { App } from '@capacitor/app';
import { AdMob } from '@capacitor-community/admob';
/**
 * Hook to detect if app is running in Capacitor and initialize native features
 */
export function useCapacitor() {
  const [isCapacitorApp, setIsCapacitorApp] = useState(false);
  const [isAndroid, setIsAndroid] = useState(false);
  const [isIOS, setIsIOS] = useState(false);
  const [isInitialized, setIsInitialized] = useState(false);
  useEffect(() => {
    // Detect and initialize Capacitor features
    async function checkPlatform() {
      try {
        // Check if window.Capacitor exists
        const isCapacitor = typeof window !== 'undefined' && 'Capacitor' in window;
        setIsCapacitorApp(isCapacitor);
        if (isCapacitor) {
          // Get device info
          const info = await Device.getInfo();
          setIsAndroid(info.platform === 'android');
          setIsIOS(info.platform === 'ios');
          // Initialize AdMob if on a native platform
          if (info.platform === 'android' || info.platform === 'ios') {
            try {
              // Request tracking authorization (primarily for iOS)
              await AdMob.requestTrackingAuthorization();

              // Initialize AdMob with your app ID
              await AdMob.initialize({
                tagForChildDirectedTreatment: true,
              });

              console.log('AdMob initialized successfully');
            } catch (error) {
              console.error('Error initializing AdMob:', error);
            }
          }
          // Add app state change listener (background/foreground)
          App.addListener('appStateChange', ({ isActive }) => {
            console.log('App state changed. Is active:', isActive);
          });
          // Update initialization status
          setIsInitialized(true);
          console.log(`Capacitor initialized for ${info.platform}`);
        } else {
          // Not running in Capacitor
          console.log('Not running in Capacitor environment');
          setIsInitialized(true); // Mark as initialized for web
        }
      } catch (error) {
        console.error('Error initializing Capacitor:', error);
        setIsInitialized(true); // Mark as initialized even on error to prevent infinite loading
      }
    }
    // Run the platform check
    checkPlatform();
    // Cleanup function
    return () => {
      // Remove any listeners if needed
    };
  }, []);
  return {
    isCapacitorApp,
    isAndroid,
    isIOS,
    isInitialized
  };
}
```