

File: client/src/components/InterstitialAd.tsx

Type: tsx | Size: 4574 bytes

```
import React, { useState, useEffect } from 'react';
import { motion, AnimatePresence } from 'framer-motion';
import { AD_CLASSES, AD_UNIT_IDS, canShowInterstitial } from '@lib/adManager';
import { prepareInterstitial, showInterstitialIfReady } from '@lib/adMobService';
import { useCapacitor } from '@hooks/use-capacitor';

interface InterstitialAdProps {
  isVisible: boolean;
  onClose: () => void;
}

/**
 * Interstitial Ad Component
 * Displays a full-screen advertisement between user actions
 *
 * On web: Shows a countdown interstitial
 * On Android/iOS: Uses native AdMob interstitials via the Capacitor AdMob plugin
 */
export default function InterstitialAd({ isVisible, onClose }: InterstitialAdProps) {
  const [timeLeft, setTimeLeft] = useState(5); // 5 second countdown
  const { isCapacitorApp, isInitialized } = useCapacitor();

  // Prepare interstitial on component mount for native platforms
  useEffect(() => {
    if (isCapacitorApp && isInitialized) {
      prepareInterstitial().catch(error => {
        console.error('Error preparing interstitial:', error);
      });
    }
  }, [isCapacitorApp, isInitialized]);

  // Handle interstitial display
  useEffect(() => {
    // If not visible or can't show interstitial, don't do anything
    if (!isVisible || !canShowInterstitial()) {
      return;
    }

    // If we're on a mobile device and initialized, show the native ad
    if (isCapacitorApp && isInitialized) {
      // For native platforms, the ad will be shown by AdMobService
      // and we'll immediately close this component
      showInterstitialIfReady().catch(error => {
        console.error('Error showing interstitial:', error);
      });

      // Close the interstitial component since the native ad will take over
      onClose();
      return;
    }

    // For web, show our custom interstitial with countdown
    const timer = setInterval(() => {
      setTimeLeft((prev) => {
        if (prev <= 1) {
          clearInterval(timer);
          return 0;
        }
        return prev - 1;
      });
    }, 1000);

    return () => {
      clearInterval(timer);
      setTimeLeft(5);
    };
  }, [isVisible, isCapacitorApp, isInitialized, onClose]);

  // Only render if:
  // 1. We should show the ad (isVisible is true)
```

```

// 2. We can show an interstitial based on frequency constraints
// 3. We're not on a native platform (since those use native AdMob interstitials)
if (!isVisible || !canShowInterstitial() || (isCapacitorApp && isInitialized)) return null;

// This web placeholder only shows on non-Capacitor platforms (like in development)
return (
  <AnimatePresence>
    {isVisible && (
      <motion.div
        className={`fixed inset-0 z-50 bg-mystical-dark/90 backdrop-blur-sm flex flex-col items-center justi...
        data-ad-unit-id={AD_UNIT_IDS.INTERSTITIAL}
        initial={{ opacity: 0 }}
        animate={{ opacity: 1 }}
        exit={{ opacity: 0 }}
      >
        /* This is a placeholder for web. In Android/iOS, the actual interstitial is shown by AdMob */
        <div className="w-full max-w-md bg-mystical-gradient/30 rounded-xl p-8 border border-mystical-purple...
          <div className="w-16 h-16 rounded-full bg-mystical-purple/20 flex items-center justify-center mb-4...
            <i className="fas fa-sparkles text-mystical-pink text-lg"></i>
          </div>
          <h3 className="font-serif text-xl text-mystical-pink mb-2">Advertisement</h3>
          <p className="text-mystical-lavender text-center mb-6">
            Upgrade to premium to remove ads and unlock more features!
          </p>

          <div className="bg-mystical-dark w-full h-[200px] rounded-lg mb-6 flex items-center justify-center...
            <p className="text-mystical-lavender/50">Ad content will appear here</p>
          </div>

          {timeLeft > 0 ? (
            <p className="text-sm text-mystical-lavender/70">
              Continue in {timeLeft} seconds...
            </p>
          ) : (
            <motion.button
              onClick={onClose}
              className="bg-mystical-purple text-white font-medium py-2 px-6 rounded-lg"
              whileHover={{ scale: 1.05 }}
              whileTap={{ scale: 0.95 }}
            >
              Continue
            </motion.button>
          )}
        </div>
      </motion.div>
    )}
  </AnimatePresence>
);
}

```