

File: client/src/App.tsx
Type: tsx | Size: 4092 bytes

```
import { Switch, Route } from "wouter";
import { queryClient } from "./lib/queryClient";
import { QueryClientProvider } from "@tanstack/react-query";
import { Toaster } from "@/components/ui/toaster";
import { TooltipProvider } from "@/components/ui/tooltip";
import NotFound from "@/pages/not-found";
import { useState, useEffect } from "react";
import Home from "@/pages/Home";
import Reading from "@/pages/Reading";
import DailyDraw from "@/pages/DailyDraw";
import Compatibility from "@/pages/Compatibility";
import LoveCircles from "@/pages/LoveCircles";
import InteractiveReading from "@/pages/InteractiveReading";
import History from "@/pages/History";
import Profile from "@/pages/Profile";
import Subscription from "@/pages/Subscription";
import SplashScreen from "@/components/SplashScreen";
import Navigation from "@/components/Navigation";
import StarBackground from "@/components/StarBackground";
import BannerAd from "@/components/BannerAd";
import InterstitialAd from "@/components/InterstitialAd";
import { useAds } from "@/hooks/use-ads";
import { useCapacitor } from "@/hooks/use-capacitor";
import { useCapacitorBackButton } from "@/hooks/use-capacitor-back-button";

function Router() {
  // State hooks must be called in the exact same order every render
  const [showSplash, setShowSplash] = useState(true);
  const [activePage, setActivePage] = useState("home");

  // Custom hooks
  const { showInterstitial, closeInterstitial } = useAds();
  const { isCapacitorApp, isAndroid, isIOS, isInitialized } = useCapacitor();

  // Effects
  useEffect(() => {
    // For debugging only
    console.log('Capacitor status:', { isCapacitorApp, isAndroid, isIOS, isInitialized });
  }, [isCapacitorApp, isAndroid, isIOS, isInitialized]);

  // Add back button handler for Android using our custom hook
  useCapacitorBackButton(() => {
    // If not on home screen, navigate to home
    if (activePage !== "home") {
      setActivePage("home");
      return false; // Prevent default back action
    }

    // On home screen, use default behavior (exit app)
    return true;
  });

  useEffect(() => {
    // Show splash screen for at least 2 seconds, or until Capacitor init is done
    let splashTimer = setTimeout(() => {
      // Only hide splash if we're not on a capacitor app or capacitor has initialized
      if (!isCapacitorApp || isInitialized) {
        setShowSplash(false);
      }
    }, 2000);

    // If we're on Capacitor and it's initialized, we can hide the splash
    if (isCapacitorApp && isInitialized) {
      setShowSplash(false);
      clearTimeout(splashTimer);
    }

    return () => clearTimeout(splashTimer);
  }, [isCapacitorApp, isInitialized]);
}
```

```

if (showSplash) {
  return <SplashScreen />;
}

return (
<div className="relative overflow-x-hidden min-h-screen bg-mystical-dark text-mystical-light pb-20">
  <StarBackground />

  {/* Interstitial Ad */}
  <InterstitialAd
    isVisible={showInterstitial}
    onClose={closeInterstitial}
  />

  <main className="pb-20 pt-4">
    {/* Banner Ad at the top */}
    <BannerAd position="top" />

    {activePage === "home" && <Home setActivePage={setActivePage} />}
    {activePage === "reading" && <Reading />}
    {activePage === "daily" && <DailyDraw />}
    {activePage === "compatibility" && <Compatibility />}
    {/* Love Circles feature hidden until we get more sign ups */}
    {/* {activePage === "circles" && <LoveCircles />} */}
    {activePage === "interactive" && <InteractiveReading />}
    {activePage === "history" && <History />}
    {activePage === "profile" && <Profile />}
    {activePage === "subscription" && <Subscription />}
  </main>
  <Navigation activePage={activePage} setActivePage={setActivePage} />
</div>
);
}

function App() {
  return (
    <QueryClientProvider client={queryClient}>
      <TooltipProvider>
        <Toaster />
        <Router />
      </TooltipProvider>
    </QueryClientProvider>
  );
}

export default App;

```