/App.tsx

See all issues in this file

```
// FIX: This file was a placeholder. Added the main App component with routing, layout, and context providers. This resolves module resolution errors in other files that import it.
// FIX: Refactored dashboard routing to use a single '/home' route with a consistent header provided by DashboardPage. Removed redundant '/admin' and '/dashboard' routes for clarity and consistency.
import React, { Suspense, useContext, useState, useCallback, ReactNode, Component } from 'react';
```

Remove this unused import of 'Component'.

```
import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';
import { AuthContext, ToastContext, Toast } from './context/AuthContext';
import { Role, User } from './types';
import Sidebar from './components/layout/Sidebar';
import OnboardingPage from './pages/OnboardingPage';
import SuspenseLoader from './components/ui/SuspenseLoader';
import ToastContainer from './components/ui/Toast';
import Tutorial from './components/onboarding/Tutorial';
import DashboardPage from './pages/DashboardPage';
// Lazy load pages for better performance
const MusicManagementPage = React.lazy(() => import('./pages/music/MusicManagementPage'));
const PublishingManagementPage = React.lazy(() => import('./pages/publishing/PublishingManagementPage'));
const EcommerceManagementPage = React.lazy(() => import('./pages/ecommerce/EcommerceManagementPage'));
const FinancesManagementPage = React.lazy(() => import('./pages/finances/FinancesManagementPage'));
const MarketingManagementPage = React.lazy(() => import('./pages/marketing/MarketingManagementPage'));
const SettingsPage = React.lazy(() => import('./pages/SettingsPage'));
const ComingSoonPage = React.lazy(() => import('./pages/ComingSoonPage'));
const HomePage = React.lazy(() => import('./pages/HomePage'));
const AnalyticsPage = React.lazy(() => import('./pages/AnalyticsPage'));
const StrategyManagementPage = React.lazy(() => import('./pages/strategy/StrategyManagementPage'));
const CollaborationManagementPage = React.lazy(() => import('./pages/collaboration/CollaborationManagementPage'));
const CreativeLabManagementPage = React.lazy(() => import('./pages/creative/CreativeLabManagementPage'));
const FanHubManagementPage = React.lazy(() => import('./pages/fanhub/FanHubManagementPage'));
const PrometheusGenesisPage = React.lazy(() => import('./pages/prometheus/PrometheusGenesisPage'));
// --- Providers (moved from context/Providers.tsx to resolve loading issue) ---
```

```
const AuthProvider: React.FC<{ children: ReactNode }> = ({ children }) => {
  const [user, setUser] = useState<User | null>(null);
  const [showTutorialFor, setShowTutorialFor] = useState<Role | null>(null);
  const login = useCallback((role: Role) => {
    if (role === Role.ADMIN) {
      setUser({
        id: 'admin-001',
        name: 'Alex Admin',
        role: Role.ADMIN,
        avatar: 'https://i.pravatar.cc/150?u=admin-alex',
        email: 'alex.admin@hardbanrecords.com',
        bio: 'Administrator overseeing the entire HardbanRecords platform operations.',
        location: 'Warsaw, Poland',
        website: 'https://hardbanrecords.com'
      });
    } else if (role === Role.MUSIC_CREATOR) {
      setUser({
        id: 'creator-001',
        name: 'Casey Creator',
        role: Role.MUSIC_CREATOR,
        avatar: 'https://i.pravatar.cc/150?u=casey-creator',
        email: 'casey@synthwave.com',
        bio: 'Electronic music producer specializing in Synthwave and Chillwave. Creating
dreamscapes through sound.',
        location: 'Los Angeles, USA',
        website: 'https://caseycreator.com'
      });
    } else { // BOOK_AUTHOR
      setUser({
        id: 'author-001',
        name: 'Pat Publisher',
        role: Role.BOOK_AUTHOR,
        avatar: 'https://i.pravatar.cc/150?u=pat-publisher',
        email: 'pat@publishing.com',
        bio: 'Bestselling author of Sci-Fi and Fantasy novels. Creating new worlds one page at
a time.',
        location: 'London, UK',
        website: 'https://patpublisher.com'
      });
    }
    setShowTutorialFor(role);
  }, []);
  const logout = useCallback(() => {
    setUser(null);
    setShowTutorialFor(null);
  }, []);
  const updateUser = useCallback((updatedData: Partial<User>) => {
    setUser(currentUser => currentUser ? { ...currentUser, ...updatedData } : null);
  }, []);
  return (
    <AuthContext.Provider value={{ user, login, logout, updateUser, showTutorialFor,
setShowTutorialFor }}>
      {children}
    </AuthContext.Provider>
  );
};
let toastCount = 0;
const ToastProvider: React.FC<{ children: ReactNode }> = ({ children }) => {
  const [toasts, setToasts] = useState<Toast[]>([]);
  const addToast = useCallback((message: string, type: 'success' | 'error') => {
    const id = toastCount++;
    setToasts(currentToasts => [...currentToasts, { id, message, type }]);
  }, []);
  const removeToast = useCallback((id: number) => {
    setToasts(currentToasts => currentToasts.filter(toast => toast.id !== id));
  }, []);
  return (
    <ToastContext.Provider value={{ toasts, addToast, removeToast }}>
      {children}
    </ToastContext.Provider>
  );
};
// --- Main App Components ---
const AppContent: React.FC = () => {
  const { user, showTutorialFor, setShowTutorialFor } = useContext(AuthContext);
  if (!user) {
    return <OnboardingPage />;
  }
  const getDashboardTitle = () => {
    if (!user) return 'Dashboard';
    switch (user.role) {
      case Role.ADMIN:
        return 'Admin Dashboard';
      case Role.MUSIC_CREATOR:
        return 'Creator Dashboard';
      case Role.BOOK_AUTHOR:
        return 'Author Dashboard';
      default:
        return 'Dashboard';
    }
  };
  return (
    <>
      {showTutorialFor && <Tutorial role={showTutorialFor} onClose={() =>
setShowTutorialFor(null)} />}
      <div className="flex bg-dark-bg min-h-screen text-gray-200">
```

```
        <Sidebar />
        <main className="flex-1 ml-64">
          <Suspense fallback={<div className="h-screen w-full flex items-center
justify-center"><SuspenseLoader /></div>}>
            <Routes>
              <Route path="/" element={<Navigate to="/home" replace />} />
              <Route path="/home" element={<DashboardPage title={getDashboardTitle()}
component={<HomePage />} />} />
              <Route path="/music/*" element={<MusicManagementPage />} />
              <Route path="/publishing/*" element={<PublishingManagementPage />} />
              <Route path="/ecommerce/*" element={<EcommerceManagementPage />} />
              <Route path="/fanhub/*" element={<FanHubManagementPage />} />
              <Route path="/finances/*" element={<FinancesManagementPage />} />
              <Route path="/marketing/*" element={<MarketingManagementPage />} />
              <Route path="/settings" element={<SettingsPage />} />
              <Route path="/analytics/*" element={<AnalyticsPage />} />
              <Route path="/strategy/*" element={<StrategyManagementPage />} />
              <Route path="/creative/*" element={<CreativeLabManagementPage />} />
              <Route path="/prometheus/*" element={<PrometheusGenesisPage />} />
              {/* Fallback for any other marketing pages not explicitly defined */}
              <Route path="/marketing/*" element={<ComingSoonPage title="Marketing Tool" />}
/>
            </Routes>
          </Suspense>
        </main>
      </div>
    </>
  );
};
const App: React.FC = () => {
  return (
    <AuthProvider>
      <ToastProvider>
        <Router>
          <AppContent />
        </Router>
        <ToastContainer />
      </ToastProvider>
    </AuthProvider>
  );
};
export default App;
```

nastrrpny bład:

/App.tsx

See all issues in this file

```
// FIX: This file was a placeholder. Added the main App component with
routing, layout, and context providers. This resolves module resolution
errors in other files that import it.
// FIX: Refactored dashboard routing to use a single '/home' route with
a consistent header provided by DashboardPage. Removed redundant
'/admin' and '/dashboard' routes for clarity and consistency.
import React, { Suspense, useContext, useState, useCallback, ReactNode,
Component } from 'react';
import { BrowserRouter as Router, Routes, Route, Navigate } from
'react-router-dom';
import { AuthContext, ToastContext, Toast } from
'./context/AuthContext';
import { Role, User } from './types';
import Sidebar from './components/layout/Sidebar';
import OnboardingPage from './pages/OnboardingPage';
import SuspenseLoader from './components/ui/SuspenseLoader';
import ToastContainer from './components/ui/Toast';
import Tutorial from './components/onboarding/Tutorial';
import DashboardPage from './pages/DashboardPage';
// Lazy load pages for better performance
const MusicManagementPage = React.lazy(() =>
import('./pages/music/MusicManagementPage'));
const PublishingManagementPage = React.lazy(() =>
import('./pages/publishing/PublishingManagementPage'));
const EcommerceManagementPage = React.lazy(() =>
import('./pages/ecommerce/EcommerceManagementPage'));
const FinancesManagementPage = React.lazy(() =>
import('./pages/finances/FinancesManagementPage'));
const MarketingManagementPage = React.lazy(() =>
import('./pages/marketing/MarketingManagementPage'));
const SettingsPage = React.lazy(() => import('./pages/SettingsPage'));
const ComingSoonPage = React.lazy(() =>
import('./pages/ComingSoonPage'));
const HomePage = React.lazy(() => import('./pages/HomePage'));
const AnalyticsPage = React.lazy(() =>
import('./pages/AnalyticsPage'));
const StrategyManagementPage = React.lazy(() =>
import('./pages/strategy/StrategyManagementPage'));
```

```
const CollaborationManagementPage = React.lazy(() =>
import('./pages/collaboration/CollaborationManagementPage'));
const CreativeLabManagementPage = React.lazy(() =>
import('./pages/creative/CreativeLabManagementPage'));
const FanHubManagementPage = React.lazy(() =>
import('./pages/fanhub/FanHubManagementPage'));
const PrometheusGenesisPage = React.lazy(() =>
import('./pages/prometheus/PrometheusGenesisPage'));
// --- Providers (moved from context/Providers.tsx to resolve loading
issue) ---
const AuthProvider: React.FC<{ children: ReactNode }> = ({ children })
=> {
 const [user, setUser] = useState<User | null>(null);
 const [showTutorialFor, setShowTutorialFor] = useState<Role |
null>(null);
 const login = useCallback((role: Role) => {
   if (role === Role.ADMIN) {
     setUser({
       id: 'admin-001',
       name: 'Alex Admin',
       role: Role.ADMIN,
       avatar: 'https://i.pravatar.cc/150?u=admin-alex',
       email: 'alex.admin@hardbanrecords.com',
       bio: 'Administrator overseeing the entire HardbanRecords
platform operations.',
       location: 'Warsaw, Poland',
       website: 'https://hardbanrecords.com'
     });
   } else if (role === Role.MUSIC_CREATOR) {
     setUser({
       id: 'creator-001',
       name: 'Casey Creator',
       role: Role.MUSIC_CREATOR,
       avatar: 'https://i.pravatar.cc/150?u=casey-creator',
       email: 'casey@synthwave.com',
       bio: 'Electronic music producer specializing in Synthwave and
Chillwave. Creating dreamscapes through sound.',
       location: 'Los Angeles, USA',
       website: 'https://caseycreator.com'
     });
   } else { // BOOK_AUTHOR
```

```
     setUser({
       id: 'author-001',
       name: 'Pat Publisher',
       role: Role.BOOK_AUTHOR,
       avatar: 'https://i.pravatar.cc/150?u=pat-publisher',
       email: 'pat@publishing.com',
       bio: 'Bestselling author of Sci-Fi and Fantasy novels. Creating
new worlds one page at a time.',
       location: 'London, UK',
       website: 'https://patpublisher.com'
     });
   }
   setShowTutorialFor(role);
 }, []);
 const logout = useCallback(() => {
   setUser(null);
   setShowTutorialFor(null);
 }, []);
 const updateUser = useCallback((updatedData: Partial<User>) => {
   setUser(currentUser => currentUser ? { ...currentUser,
...updatedData } : null);
 }, []);
 return (
   <AuthContext.Provider value={{ user
, login, logout, updateUser, showTutorialFor, setShowTutorialFor }}>
```

The object passed as the value prop to the Context provider changes every render. To fix this consider wrapping it in a useMemo hook.

```
     {children}
   </AuthContext.Provider>
 );
};
let toastCount = 0;
const ToastProvider: React.FC<{ children: ReactNode }> = ({ children })
=> {
 const [toasts, setToasts] = useState<Toast[]>([]);
 const addToast = useCallback((message: string, type: 'success' |
'error') => {
   const id = toastCount++;
   setToasts(currentToasts => [...currentToasts, { id, message, type
}]);
```

```
 }, []);
 const removeToast = useCallback((id: number) => {
   setToasts(currentToasts => currentToasts.filter(toast => toast.id
!== id));
 }, []);
 return (
   <ToastContext.Provider value={{ toasts, addToast, removeToast }}>
     {children}
   </ToastContext.Provider>
 );
};
// --- Main App Components ---
const AppContent: React.FC = () => {
 const { user, showTutorialFor, setShowTutorialFor } =
useContext(AuthContext);
 if (!user) {
   return <OnboardingPage />;
 }
 const getDashboardTitle = () => {
   if (!user) return 'Dashboard';
   switch (user.role) {
     case Role.ADMIN:
       return 'Admin Dashboard';
     case Role.MUSIC_CREATOR:
       return 'Creator Dashboard';
     case Role.BOOK_AUTHOR:
       return 'Author Dashboard';
     default:
       return 'Dashboard';
   }
 };
 return (
   <>
     {showTutorialFor && <Tutorial role={showTutorialFor} onClose={()
=> setShowTutorialFor(null)} />}
     <div className="flex bg-dark-bg min-h-screen text-gray-200">
       <Sidebar />
       <main className="flex-1 ml-64">
         <Suspense fallback={<div className="h-screen w-full flex
items-center justify-center"><SuspenseLoader /></div>}>
           <Routes>
```

```
             <Route path="/" element={<Navigate to="/home" replace />}
/>
             <Route path="/home" element={<DashboardPage
title={getDashboardTitle()} component={<HomePage />} />} />
             <Route path="/music/*" element={<MusicManagementPage />}
/>
             <Route path="/publishing/*"
element={<PublishingManagementPage />} />
             <Route path="/ecommerce/*"
element={<EcommerceManagementPage />} />
             <Route path="/fanhub/*" element={<FanHubManagementPage />}
/>
             <Route path="/finances/*" element={<FinancesManagementPage
/>} />
             <Route path="/marketing/*"
element={<MarketingManagementPage />} />
             <Route path="/settings" element={<SettingsPage />} />
             <Route path="/analytics/*" element={<AnalyticsPage />} />
             <Route path="/strategy/*" element={<StrategyManagementPage
/>} />
             <Route path="/creative/*"
element={<CreativeLabManagementPage />} />
             <Route path="/prometheus/*"
element={<PrometheusGenesisPage />} />
             {/* Fallback for any other marketing pages not explicitly
defined */}
             <Route path="/marketing/*" element={<ComingSoonPage
title="Marketing Tool" />} />
           </Routes>
         </Suspense>
       </main>
     </div>
   </>
 );
};
const App: React.FC = () => {
 return (
   <AuthProvider>
     <ToastProvider>
       <Router>
         <AppContent />
       </Router>
       <ToastContainer />
     </ToastProvider>
   </AuthProvider>
 );
};
export default App;
```

NASTEPNY BLAD:

/App.tsx

See all issues in this file

```
// FIX: This file was a placeholder. Added the main App component with
routing, layout, and context providers. This resolves module resolution
errors in other files that import it.
// FIX: Refactored dashboard routing to use a single '/home' route with
a consistent header provided by DashboardPage. Removed redundant
'/admin' and '/dashboard' routes for clarity and consistency.
import React, { Suspense, useContext, useState, useCallback, ReactNode,
Component } from 'react';
import { BrowserRouter as Router, Routes, Route, Navigate } from
'react-router-dom';
import { AuthContext, ToastContext, Toast } from
'./context/AuthContext';
import { Role, User } from './types';
import Sidebar from './components/layout/Sidebar';
import OnboardingPage from './pages/OnboardingPage';
import SuspenseLoader from './components/ui/SuspenseLoader';
import ToastContainer from './components/ui/Toast';
import Tutorial from './components/onboarding/Tutorial';
import DashboardPage from './pages/DashboardPage';
// Lazy load pages for better performance
const MusicManagementPage = React.lazy(() =>
import('./pages/music/MusicManagementPage'));
const PublishingManagementPage = React.lazy(() =>
import('./pages/publishing/PublishingManagementPage'));
const EcommerceManagementPage = React.lazy(() =>
import('./pages/ecommerce/EcommerceManagementPage'));
const FinancesManagementPage = React.lazy(() =>
import('./pages/finances/FinancesManagementPage'));
```

```
const MarketingManagementPage = React.lazy(() =>
import('./pages/marketing/MarketingManagementPage'));
const SettingsPage = React.lazy(() => import('./pages/SettingsPage'));
const ComingSoonPage = React.lazy(() =>
import('./pages/ComingSoonPage'));
const HomePage = React.lazy(() => import('./pages/HomePage'));
const AnalyticsPage = React.lazy(() =>
import('./pages/AnalyticsPage'));
const StrategyManagementPage = React.lazy(() =>
import('./pages/strategy/StrategyManagementPage'));
const CollaborationManagementPage = React.lazy(() =>
import('./pages/collaboration/CollaborationManagementPage'));
const CreativeLabManagementPage = React.lazy(() =>
import('./pages/creative/CreativeLabManagementPage'));
const FanHubManagementPage = React.lazy(() =>
import('./pages/fanhub/FanHubManagementPage'));
const PrometheusGenesisPage = React.lazy(() =>
import('./pages/prometheus/PrometheusGenesisPage'));
// --- Providers (moved from context/Providers.tsx to resolve loading
issue) ---
const AuthProvider: React.FC<{ children: ReactNode }> = ({ children })
=> {
 const [user, setUser] = useState<User | null>(null);
 const [showTutorialFor, setShowTutorialFor] = useState<Role |
null>(null);
 const login = useCallback((role: Role) => {
   if (role === Role.ADMIN) {
     setUser({
       id: 'admin-001',
       name: 'Alex Admin',
       role: Role.ADMIN,
       avatar: 'https://i.pravatar.cc/150?u=admin-alex',
       email: 'alex.admin@hardbanrecords.com',
       bio: 'Administrator overseeing the entire HardbanRecords
platform operations.',
       location: 'Warsaw, Poland',
       website: 'https://hardbanrecords.com'
     });
   } else if (role === Role.MUSIC_CREATOR) {
     setUser({
       id: 'creator-001',
       name: 'Casey Creator',
       role: Role.MUSIC_CREATOR,
       avatar: 'https://i.pravatar.cc/150?u=casey-creator',
       email: 'casey@synthwave.com',
       bio: 'Electronic music producer specializing in Synthwave and Chillwave. Creating dreamscapes through sound.',
       location: 'Los Angeles, USA',
       website: 'https://caseycreator.com'
     });
   } else { // BOOK_AUTHOR
     setUser({
       id: 'author-001',
       name: 'Pat Publisher',
       role: Role.BOOK_AUTHOR,
       avatar: 'https://i.pravatar.cc/150?u=pat-publisher',
       email: 'pat@publishing.com',
       bio: 'Bestselling author of Sci-Fi and Fantasy novels. Creating new worlds one page at a time.',
       location: 'London, UK',
       website: 'https://patpublisher.com'
     });
   }
   setShowTutorialFor(role);
 }, []);
 const logout = useCallback(() => {
   setUser(null);
   setShowTutorialFor(null);
 }, []);
 const updateUser = useCallback((updatedData: Partial<User>) => {
   setUser(currentUser => currentUser ? { ...currentUser, ...updatedData } : null);
 }, []);
 return (
   <AuthContext.Provider value={{ user, login, logout, updateUser, showTutorialFor, setShowTutorialFor }}>
     {children}
   </AuthContext.Provider>
 );
};
let toastCount = 0;
```

```
const ToastProvider: React.FC<{ children: ReactNode }> = ({ children })
=> {
 const [toasts, setToasts] = useState<Toast[]>([]);
 const addToast = useCallback((message: string, type: 'success' |
'error') => {
   const id = toastCount++;
   setToasts(currentToasts => [...currentToasts, { id, message, type
}]);
 }, []);
 const removeToast = useCallback((id: number) => {
   setToasts(currentToasts => currentToasts.filter(toast => toast.id
!== id));
 }, []);
 return (
   <ToastContext.Provider value={{ toasts
, addToast, removeToast }}>
```

The object passed as the value prop to the Context provider changes every render. To fix this consider wrapping it in a useMemo hook.

```
     {children}
   </ToastContext.Provider>
 );
};
// --- Main App Components ---
const AppContent: React.FC = () => {
 const { user, showTutorialFor, setShowTutorialFor } =
useContext(AuthContext);
 if (!user) {
   return <OnboardingPage />;
 }
 const getDashboardTitle = () => {
   if (!user) return 'Dashboard';
   switch (user.role) {
     case Role.ADMIN:
       return 'Admin Dashboard';
     case Role.MUSIC_CREATOR:
       return 'Creator Dashboard';
     case Role.BOOK_AUTHOR:
       return 'Author Dashboard';
     default:
       return 'Dashboard';
   }
 };
 return (
   <>
     {showTutorialFor && <Tutorial role={showTutorialFor} onClose={() => setShowTutorialFor(null)} />}
     <div className="flex bg-dark-bg min-h-screen text-gray-200">
       <Sidebar />
       <main className="flex-1 ml-64">
         <Suspense fallback={<div className="h-screen w-full flex items-center justify-center"><SuspenseLoader /></div>}>
           <Routes>
             <Route path="/" element={<Navigate to="/home" replace />} />
             <Route path="/home" element={<DashboardPage title={getDashboardTitle()} component={<HomePage />} />} />
             <Route path="/music/*" element={<MusicManagementPage />} />
             <Route path="/publishing/*" element={<PublishingManagementPage />} />
             <Route path="/ecommerce/*" element={<EcommerceManagementPage />} />
             <Route path="/fanhub/*" element={<FanHubManagementPage />} />
             <Route path="/finances/*" element={<FinancesManagementPage />} />
             <Route path="/marketing/*" element={<MarketingManagementPage />} />
             <Route path="/settings" element={<SettingsPage />} />
             <Route path="/analytics/*" element={<AnalyticsPage />} />
             <Route path="/strategy/*" element={<StrategyManagementPage />} />
             <Route path="/creative/*" element={<CreativeLabManagementPage />} />
             <Route path="/prometheus/*" element={<PrometheusGenesisPage />} />
             {/* Fallback for any other marketing pages not explicitly defined */}
             <Route path="/marketing/*" element={<ComingSoonPage title="Marketing Tool" />} />
           </Routes>
         </Suspense>
       </main>
     </div>
   </>
 );
};
const App: React.FC = () => {
 return (
   <AuthProvider>
     <ToastProvider>
       <Router>
         <AppContent />
       </Router>
       <ToastContainer />
     </ToastProvider>
   </AuthProvider>
 );
};
export default App;
```