import React, { useState, useEffect, useCallback } from 'react';

import { initializeApp } from 'firebase/app';

import {

getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged

} from 'firebase/auth';

import {

getFirestore, doc, collection, query, orderBy, onSnapshot, updateDoc, addDoc, deleteDoc, serverTimestamp

} from 'firebase/firestore';

import {

Lock, CheckCircle, PenTool, Loader, XCircle, Info, Trash2, ShieldCheck, User, Plus, Search, Newspaper

} from 'lucide-react';

// --- GLOBAL VARIABLES (Provided by Canvas Environment) ---

const appId = typeof \_\_app\_id !== 'undefined' ? \_\_app\_id : 'default-app-id';

const firebaseConfig = typeof \_\_firebase\_config !== 'undefined' ? JSON.parse(\_\_firebase\_config) : {

apiKey: "YOUR\_FIREBASE\_API\_KEY",

authDomain: "YOUR\_FIREBASE\_AUTH\_DOMAIN",

projectId: "YOUR\_FIREBASE\_PROJECT\_ID",

storageBucket: "YOUR\_FIREBASE\_STORAGE\_BUCKET",

messagingSenderId: "YOUR\_FIREBASE\_MESSAGING\_SENDER\_ID",

appId: "YOUR\_FIREBASE\_APP\_ID"

};

const initialAuthToken = typeof \_\_initial\_auth\_token !== 'undefined' ? \_\_initial\_auth\_token : null;

// --- UTILITY: Firebase Initialization and Setup ---

let app, db, auth;

try {

app = initializeApp(firebaseConfig);

db = getFirestore(app);

auth = getAuth(app);

} catch (error) {

console.error("Firebase initialization failed:", error);

}

// --- CONSTANTS ---

const ADMIN\_PASSWORD = 'Moab1212L';

// Style Constants: Deep Red & Clean Gray

const PRIMARY\_COLOR = 'red-700'; // Deep Red (Highlights/Buttons)

const ACCENT\_COLOR = 'gray-900'; // Clean Charcoal/Black (Main Background/Text)

/\*\*

\* Main application component for Grassroots Post.

\* Displays headlines and allows authorized users to create and edit reports.

\*/

const App = () => {

const [reports, setReports] = useState([]);

const [isLoading, setIsLoading] = useState(true);

const [error, setError] = useState(null);

const [isAuthReady, setIsAuthReady] = useState(false);

const [userId, setUserId] = useState(null);

const [userContext, setUserContext] = useState(null);

const [isAdmin, setIsAdmin] = useState(false);

// Modals and Forms

const [showAdminModal, setShowAdminModal] = useState(false);

const [showReportModal, setShowReportModal] = useState(false);

const [currentReport, setCurrentReport] = useState(null);

const [showConfirmationModal, setShowConfirmationModal] = useState(false);

const [reportToDelete, setReportToDelete] = useState(null);

// Admin Modal State

const [adminPasswordInput, setAdminPasswordInput] = useState('');

const [adminUserIdInput, setAdminUserIdInput] = useState('');

const [adminMessage, setAdminMessage] = useState('');

// Report Form State

const [title, setTitle] = useState('');

const [headline, setHeadline] = useState('');

const [content, setContent] = useState('');

const [isSaving, setIsSaving] = useState(false);

// --- Auth & Setup Effect ---

useEffect(() => {

if (!auth) return;

const setupAuth = async () => {

try {

if (!auth.currentUser) {

if (initialAuthToken) {

await signInWithCustomToken(auth, initialAuthToken);

} else {

await signInAnonymously(auth);

}

}

} catch (e) {

console.error("Initial sign-in failed:", e);

setError("Failed to sign in. Please try refreshing.");

}

};

const unsubscribe = onAuthStateChanged(auth, (user) => {

setUserContext(user);

if (user) {

setUserId(user.uid);

} else {

setUserId(crypto.randomUUID());

}

setIsAuthReady(true);

});

setupAuth();

return () => unsubscribe();

}, []);

// --- Data Fetching Effect (Firestore Listener) ---

useEffect(() => {

if (!db || !isAuthReady) {

return;

}

const reportsPath = `/artifacts/${appId}/public/data/reports`;

const q = query(collection(db, reportsPath), orderBy("timestamp", "desc"));

const unsubscribe = onSnapshot(q, (snapshot) => {

const fetchedReports = snapshot.docs.map(doc => ({

id: doc.id,

...doc.data(),

timestamp: doc.data().timestamp?.toDate ? doc.data().timestamp.toDate() : new Date(),

}));

setReports(fetchedReports);

setIsLoading(false);

}, (err) => {

console.error("Error fetching reports:", err);

if (!err.code || err.code !== 'permission-denied') {

setError("Failed to load reports in real-time.");

}

setIsLoading(false);

});

return () => unsubscribe();

}, [db, isAuthReady]);

// --- Handlers for Admin Sign-In ---

const handleAdminSignIn = useCallback(async () => {

setAdminMessage('');

if (!userId || userId === 'default-app-id') {

setAdminMessage('Authentication is not ready. Please try refreshing.');

return;

}

if (adminPasswordInput !== ADMIN\_PASSWORD) {

setAdminMessage('Incorrect secret password.');

return;

}

if (adminUserIdInput !== userId) {

setAdminMessage('Admin User ID does not match your current session ID.');

return;

}

// Success

setIsAdmin(true);

setShowAdminModal(false);

setAdminPasswordInput('');

setAdminUserIdInput('');

}, [adminPasswordInput, adminUserIdInput, userId]);

const handleOpenAdminModal = () => {

setAdminMessage('');

setAdminPasswordInput('');

setAdminUserIdInput('');

setShowAdminModal(true);

};

const handleAdminSignOut = () => {

setIsAdmin(false);

};

// --- Handlers for Report Management ---

const handleCreateReport = () => {

setCurrentReport(null);

setTitle('');

setHeadline('');

setContent('');

setShowReportModal(true);

};

const handleEditReport = (report) => {

setCurrentReport(report);

setTitle(report.title);

setHeadline(report.headline);

setContent(report.content);

setShowReportModal(true);

};

const handleViewReport = (report) => {

if (!isAdmin) {

setCurrentReport(report);

setTitle(report.title);

setHeadline(report.headline);

setContent(report.content);

setShowReportModal(true);

} else {

handleEditReport(report);

}

};

const handleSaveReport = async () => {

if (!title || !headline || !content) {

console.error("All fields must be filled.");

return;

}

if (!db || !isAdmin) {

console.error("Permission denied or database not ready.");

return;

}

setIsSaving(true);

try {

const reportData = {

title,

headline,

content,

authorId: userId,

timestamp: serverTimestamp(),

...(currentReport && { updated: serverTimestamp() })

};

const reportsCollection = collection(db, `/artifacts/${appId}/public/data/reports`);

if (currentReport) {

const reportRef = doc(reportsCollection, currentReport.id);

await updateDoc(reportRef, reportData);

} else {

await addDoc(reportsCollection, reportData);

}

setShowReportModal(false);

} catch (e) {

console.error("Error saving report: ", e);

} finally {

setIsSaving(false);

}

};

const handleDeleteReport = (report) => {

setReportToDelete(report);

setShowConfirmationModal(true);

};

const confirmDelete = async () => {

if (!reportToDelete || !db || !isAdmin) return;

try {

const reportsCollection = collection(db, `/artifacts/${appId}/public/data/reports`);

const reportRef = doc(reportsCollection, reportToDelete.id);

await deleteDoc(reportRef);

setReportToDelete(null);

setShowConfirmationModal(false);

} catch (e) {

console.error("Error deleting report: ", e);

}

};

// --- Components ---

// AdminStatusIndicator is now integrated into the header flow (not fixed)

const AdminStatusIndicator = () => (

<div

className={`p-2 rounded-full cursor-pointer transition-colors duration-300 shadow-md

${isAdmin

? `bg-red-600 hover:bg-red-700 text-white` // Admin logged in: Bright Red Shield

: 'bg-gray-700 hover:bg-gray-800 text-white' // Logged out: Dark Gray Lock (visible against charcoal header)

}`}

onClick={isAdmin ? handleAdminSignOut : handleOpenAdminModal}

title={isAdmin ? "Editor Access Granted - Click to Sign Out" : "Sign In to Admin Editor"}

>

{isAdmin ? <ShieldCheck size={24} /> : <Lock size={24} />}

</div>

);

const ReportModal = ({ report, onClose }) => {

const isEditing = !!report || currentReport;

const isReadOnly = !isAdmin || (report && !currentReport);

return (

<div className="fixed inset-0 bg-black bg-opacity-70 flex items-center justify-center z-[100] p-4">

<div className="bg-white rounded-xl shadow-2xl w-full max-w-2xl max-h-[90vh] flex flex-col overflow-hidden">

{/\* Header - Styled to match new theme \*/}

<div className="p-4 border-b bg-gray-50 flex justify-between items-center">

<h2 className="text-2xl font-bold text-gray-800">

{isEditing ? (isAdmin ? "Edit Report" : "View Report") : "Create New Report"}

</h2>

<button

onClick={onClose}

className="p-1 rounded-full text-gray-500 hover:bg-gray-200"

title="Close"

>

<XCircle size={24} />

</button>

</div>

{/\* Body \*/}

<div className="p-6 overflow-y-auto space-y-4">

<div>

<label htmlFor="title" className="block text-sm font-medium text-gray-700 mb-1">Title</label>

<input

id="title"

type="text"

value={title}

onChange={(e) => setTitle(e.target.value)}

readOnly={isReadOnly}

className={`w-full p-3 border rounded-lg shadow-sm focus:ring-${PRIMARY\_COLOR} focus:border-${PRIMARY\_COLOR} ${isReadOnly ? 'bg-gray-100' : 'bg-white'}`}

placeholder="The Main Title of the Report"

/>

</div>

<div>

<label htmlFor="headline" className="block text-sm font-medium text-gray-700 mb-1">Headline (Short Summary)</label>

<input

id="headline"

type="text"

value={headline}

onChange={(e) => setHeadline(e.target.value)}

readOnly={isReadOnly}

className={`w-full p-3 border rounded-lg shadow-sm focus:ring-${PRIMARY\_COLOR} focus:border-${PRIMARY\_COLOR} ${isReadOnly ? 'bg-gray-100' : 'bg-white'}`}

placeholder="A compelling, short headline for the front page"

/>

</div>

<div>

<label htmlFor="content" className="block text-sm font-medium text-gray-700 mb-1">Report Content</label>

<textarea

id="content"

rows="10"

value={content}

onChange={(e) => setContent(e.target.value)}

readOnly={isReadOnly}

className={`w-full p-3 border rounded-lg shadow-sm focus:ring-${PRIMARY\_COLOR} focus:border-${PRIMARY\_COLOR} resize-none ${isReadOnly ? 'bg-gray-100' : 'bg-white'}`}

placeholder="Write the full body of the report here..."

/>

</div>

</div>

{/\* Footer - Styled to match new theme \*/}

<div className="p-4 border-t bg-gray-50 flex justify-end">

{isAdmin && !isReadOnly && (

<button

onClick={handleSaveReport}

disabled={isSaving}

className={`px-6 py-2 rounded-lg font-semibold text-white transition-all duration-200 flex items-center ${isSaving ? `bg-${PRIMARY\_COLOR}/50` : `bg-${PRIMARY\_COLOR} hover:bg-red-800 shadow-md`}`}

>

{isSaving ? (

<>

<Loader size={20} className="animate-spin mr-2" />

Saving...

</>

) : (

<>

<CheckCircle size={20} className="mr-2" />

Save Report

</>

)}

</button>

)}

{!isAdmin && isReadOnly && (

<button

onClick={onClose}

className="px-6 py-2 rounded-lg font-semibold bg-gray-600 hover:bg-gray-700 text-white shadow-md"

>

Close Viewer

</button>

)}

</div>

</div>

</div>

);

};

const AdminSignInModal = ({ onClose }) => (

<div className="fixed inset-0 bg-black bg-opacity-70 flex items-center justify-center z-[110] p-4">

<div className="bg-white rounded-xl shadow-2xl w-full max-w-md p-6 space-y-4">

<h2 className={`text-2xl font-bold text-${ACCENT\_COLOR} flex items-center`}>

<ShieldCheck className={`mr-2 text-${PRIMARY\_COLOR}`} size={24} /> Admin Access

</h2>

<p className="text-sm text-gray-600">

Enter the secret password and confirm your unique session ID to gain editor privileges.

</p>

<div>

<label htmlFor="admin-uid" className="block text-sm font-medium text-gray-700 mb-1">

Admin User ID (Copy the ID below)

</label>

<input

id="admin-uid"

type="text"

value={adminUserIdInput}

onChange={(e) => setAdminUserIdInput(e.target.value)}

className={`w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:ring-${PRIMARY\_COLOR} focus:border-${PRIMARY\_COLOR}`}

placeholder="Paste your unique ID here"

/>

<p className={`mt-1 text-xs text-${ACCENT\_COLOR} flex items-center`}>

<User size={14} className="mr-1" /> Your current session ID: <span className="font-bold break-words">{userId}</span>

</p>

</div>

<div>

<label htmlFor="admin-password" className="block text-sm font-medium text-gray-700 mb-1">

Secret Password

</label>

<input

id="admin-password"

type="password"

value={adminPasswordInput}

onChange={(e) => setAdminPasswordInput(e.target.value)}

onKeyDown={(e) => e.key === 'Enter' && handleAdminSignIn()}

className={`w-full p-3 border border-gray-300 rounded-lg shadow-sm focus:ring-${PRIMARY\_COLOR} focus:border-${PRIMARY\_COLOR}`}

/>

</div>

{adminMessage && (

<p className={`text-sm p-2 rounded-lg font-medium ${adminMessage.includes('Incorrect') || adminMessage.includes('does not match') ? 'bg-red-100 text-red-700' : 'bg-green-100 text-green-700'}`}>

<Info size={16} className="inline mr-1" /> {adminMessage}

</p>

)}

{/\* Buttons - Styled to match new theme \*/}

<div className="flex justify-end space-x-3 pt-2">

<button

onClick={onClose}

className="px-4 py-2 text-gray-600 bg-gray-200 hover:bg-gray-300 rounded-lg font-medium transition"

>

Cancel

</button>

<button

onClick={handleAdminSignIn}

className={`px-4 py-2 bg-${PRIMARY\_COLOR} hover:bg-red-800 text-white rounded-lg font-semibold transition shadow-md flex items-center`}

>

<ShieldCheck size={20} className="mr-2" />

Grant Editor Access

</button>

</div>

</div>

</div>

);

const ConfirmationModal = ({ onConfirm, onCancel }) => (

<div className="fixed inset-0 bg-black bg-opacity-70 flex items-center justify-center z-[110] p-4">

<div className="bg-white rounded-xl shadow-2xl w-full max-w-sm p-6 space-y-4">

<h2 className="text-xl font-bold text-red-600 flex items-center">

<Trash2 className="mr-2" size={20} /> Confirm Deletion

</h2>

<p className="text-gray-700">

Are you sure you want to delete the report titled: <span className="font-bold">{reportToDelete?.title}</span>? This action cannot be undone.

</p>

<div className="flex justify-end space-x-3 pt-2">

<button

onClick={onCancel}

className="px-4 py-2 text-gray-600 bg-gray-200 hover:bg-gray-300 rounded-lg font-medium transition"

>

Cancel

</button>

<button

onClick={onConfirm}

className="px-4 py-2 bg-red-600 hover:bg-red-700 text-white rounded-lg font-semibold transition shadow-md flex items-center"

>

<Trash2 size={20} className="mr-2" />

Delete Permanently

</button>

</div>

</div>

</div>

);

// --- Main Render ---

if (isLoading || !isAuthReady) {

return (

<div className="flex items-center justify-center h-screen bg-gray-50">

<Loader size={32} className={`animate-spin text-${PRIMARY\_COLOR}`} />

<p className="ml-2 text-gray-600">Loading Application...</p>

</div>

);

}

if (error) {

return (

<div className="flex items-center justify-center h-screen bg-red-50 p-4">

<p className="text-red-700 font-medium">Error: {error}</p>

</div>

);

}

return (

<div className="min-h-screen bg-gray-50 font-sans antialiased pb-16">

{/\* AdminStatusIndicator is now integrated into the header below, removing the standalone call here \*/}

{/\* Header - Charcoal Background, Red Border \*/}

<header className={`bg-${ACCENT\_COLOR} shadow-md border-b-4 border-${PRIMARY\_COLOR}`}>

<div className="max-w-6xl mx-auto px-4 py-3 flex justify-between items-center">

{/\* Left Block: Title and Nav Links \*/}

<div className="flex items-center space-x-8">

<h1 className={`text-3xl font-extrabold text-white uppercase tracking-tight`}>

The Grassroots Post

</h1>

<nav className="hidden sm:flex space-x-4 text-gray-400 text-sm font-semibold">

<a href="#" className={`hover:text-${PRIMARY\_COLOR} transition`}>MATCH REPORTS</a>

<a href="#" className={`hover:text-${PRIMARY\_COLOR} transition`}>BEHIND THE GOALPOSTS</a>

<a href="#" className={`hover:text-${PRIMARY\_COLOR} transition`}>CONTACT</a>

</nav>

</div>

{/\* Right Block: Search and Admin Button (hidden on mobile) \*/}

<div className="hidden sm:flex items-center space-x-4">

<button className={`flex items-center text-sm font-medium text-white hover:text-${PRIMARY\_COLOR} transition`}>

<Search size={16} className="mr-1" /> Search

</button>

<AdminStatusIndicator />

</div>

</div>

</header>

{/\* Hero / Title Section - Clean White Background \*/}

<div className="max-w-6xl mx-auto px-4 pt-10 pb-4">

<h2 className={`text-6xl font-black text-${ACCENT\_COLOR} leading-none`}>

THE GRASSROOTS GAME

</h2>

<p className="text-lg text-gray-500 mt-3">

Unrivaled match reporting and deep investigative features on youth football in Birmingham and the Midlands.

</p>

</div>

{/\* Main Content: Reports Feed \*/}

<main className="max-w-6xl mx-auto p-4 space-y-8">

{/\* ADMIN: Create New Report Box - Red Accent \*/}

{isAdmin && (

<div

onClick={handleCreateReport}

className={`w-full bg-white border-2 border-dashed border-${PRIMARY\_COLOR} rounded-xl p-6 text-center cursor-pointer hover:bg-red-50 transition duration-200 shadow-md mb-8`}

>

<div className={`flex items-center justify-center text-${ACCENT\_COLOR}`}>

<Plus size={24} className={`mr-3 text-${PRIMARY\_COLOR}`} />

<span className="text-xl font-bold">

Write New Report (Match or Investigation)

</span>

</div>

<p className="text-sm text-gray-600 mt-1">

Click here to write and publish your next article to the front page.

</p>

</div>

)}

{/\* Featured Investigation Section (New styling) \*/}

<div className="grid grid-cols-1 lg:grid-cols-3 gap-6">

<div className={`lg:col-span-2 bg-white rounded-xl shadow-xl p-6 border-t-4 border-${PRIMARY\_COLOR}`}>

<h3 className="text-2xl font-bold text-gray-800 mb-4 flex items-center">

<Newspaper size={20} className={`mr-2 text-${PRIMARY\_COLOR}`} /> Featured Investigation

</h3>

<div className="grid grid-cols-1 sm:grid-cols-4 gap-6">

<div className="sm:col-span-1 bg-gray-100 h-32 rounded-lg flex items-center justify-center text-sm text-gray-500 font-medium border border-gray-200">

VISUAL: Action Shot Placeholder

</div>

<div className="sm:col-span-3">

<span className={`text-xs font-black text-${PRIMARY\_COLOR} uppercase tracking-widest`}>EXCLUSIVE</span>

<h4 className={`text-2xl font-extrabold text-${ACCENT\_COLOR} mt-1 leading-snug hover:text-red-800 transition cursor-pointer`}>

The Price of Potential: How Pitch Fees Are Squeezing Birmingham's Talent Pipeline

</h4>

<p className="text-base text-gray-600 mt-3 line-clamp-3">

An investigation into the rising costs of local facilities and the systemic barriers preventing low-income families from participating in the grassroots game...

</p>

<p className="text-xs text-gray-400 mt-3">

Published Today by [Your Name]

</p>

</div>

</div>

</div>

{/\* Sidebar (New styling) \*/}

<div className="space-y-6">

<div className={`bg-${ACCENT\_COLOR} rounded-xl p-5 text-white shadow-xl`}>

<h3 className={`text-xl font-extrabold border-b border-gray-700 pb-2 text-white`}>BEHIND THE GOALPOSTS</h3>

<p className="text-sm mt-3 text-gray-400">

Deep dives into the urgent financial, ethical, and structural issues surrounding grassroots football development in the Midlands.

</p>

<button className={`mt-4 w-full bg-${PRIMARY\_COLOR} text-white py-2 rounded-lg font-bold text-sm hover:bg-red-800 transition shadow-md`}>

Read All Investigative Reports

</button>

</div>

<div className="bg-white rounded-xl p-5 shadow-lg border border-gray-100">

<h3 className={`text-lg font-bold text-${ACCENT\_COLOR}`}>About The Reporter</h3>

<p className="text-sm text-gray-600 mt-1">

[Your Name] covers the non-league scene, providing match-day insight and the crucial stories affecting the future of youth development.

</p>

<p className={`text-sm text-${PRIMARY\_COLOR} mt-2 cursor-pointer hover:text-red-800 transition font-semibold`}>

Get in touch for Pitches & Tips →

</p>

</div>

</div>

</div>

<h3 className={`text-3xl font-extrabold text-${ACCENT\_COLOR} mt-12 mb-6 border-b border-${PRIMARY\_COLOR}/50 pb-2`}>

LATEST MATCH REPORTS

</h3>

{reports.length === 0 && (

<div className="text-center p-12 bg-white rounded-xl shadow-lg mt-8">

<Info size={40} className="text-gray-400 mx-auto mb-4" />

<p className="text-lg text-gray-500">

{isAdmin ? "No reports found. Start by creating a new one!" : "No reports have been published yet."}

</p>

</div>

)}

<div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6">

{reports.map((report) => (

<div key={report.id} className="bg-white rounded-xl shadow-lg p-5 hover:shadow-xl transition duration-300 border border-gray-100 flex flex-col justify-between">

<div className="space-y-2">

<div className="h-28 bg-gray-100 rounded-lg flex items-center justify-center text-xs text-gray-500 font-medium mb-2">

Team Badge Placeholder

</div>

<p className={`text-xs font-black text-${PRIMARY\_COLOR} uppercase tracking-widest flex justify-between`}>

<span>MATCH REPORT</span>

<span className="text-gray-400 font-medium">{new Date(report.timestamp).toLocaleDateString()}</span>

</p>

<h4 className={`text-xl font-bold text-${ACCENT\_COLOR} line-clamp-3 mb-2 hover:text-red-800 transition cursor-pointer`}>{report.title}</h4>

</div>

<div className="mt-4">

<button

onClick={() => handleViewReport(report)}

className={`flex items-center text-sm font-semibold text-${PRIMARY\_COLOR} hover:text-red-800 transition`}

>

Read the Full Report →

</button>

{isAdmin && (

<div className="flex space-x-4 mt-3 pt-3 border-t border-gray-100">

<button

onClick={() => handleEditReport(report)}

className="flex items-center text-xs font-medium text-blue-600 hover:text-blue-800 transition"

>

<PenTool size={14} className="mr-1" />

Edit

</button>

<button

onClick={() => handleDeleteReport(report)}

className="flex items-center text-xs font-medium text-red-600 hover:text-red-800 transition"

>

<Trash2 size={14} className="mr-1" />

Delete

</button>

</div>

)}

</div>

</div>

))}

</div>

<div className="text-center pt-8">

<button className={`px-8 py-3 border-2 border-${ACCENT\_COLOR} rounded-full text-${ACCENT\_COLOR} font-bold hover:bg-gray-100 transition shadow-sm`}>

Load More Match Reports

</button>

</div>

</main>

{/\* Footer - Charcoal Background \*/}

<footer className={`bg-${ACCENT\_COLOR} text-white mt-20 py-10`}>

<div className="max-w-6xl mx-auto px-4 text-center">

<h5 className={`text-2xl font-extrabold uppercase tracking-wider mb-3 text-${PRIMARY\_COLOR}`}>THE GRASSROOTS POST</h5>

<p className="text-sm text-gray-400 mb-6">

©2024 Independent Football Journalism for the Midlands. All Rights Reserved.

</p>

<div className="flex justify-center space-x-10 text-base font-medium">

<a href="#" className="hover:text-gray-200 transition">Our Mission</a>

<a href="#" className="hover:text-gray-200 transition">Pitches</a>

<a href="#" className="hover:text-gray-200 transition">Archive</a>

</div>

</div>

</footer>

{showReportModal && (

<ReportModal

report={currentReport}

onClose={() => setShowReportModal(false)}

/>

)}

{showAdminModal && (

<AdminSignInModal onClose={() => setShowAdminModal(false)} />

)}

{showConfirmationModal && (

<ConfirmationModal

onConfirm={confirmDelete}

onCancel={() => setShowConfirmationModal(false)}

/>

)}

</div>

);

};

export default App;