<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AI - Cuộc sống học đường không bạo lực</title>

    <!-- Tailwind CSS CDN -->

    <script src="https://cdn.tailwindcss.com"></script>

    <!-- Inter Font -->

    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

    <style>

        body {

            font-family: 'Inter', sans-serif;

            background-color: #f0f2f5;

            display: flex;

            justify-content: center;

            align-items: flex-start; /\* Align to top for better scrolling \*/

            min-height: 100vh;

            padding: 2rem 1rem;

        }

        .container {

            max-width: 90%;

            width: 100%;

            background-color: #ffffff;

            border-radius: 1.5rem; /\* Rounded corners for the main container \*/

            box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);

            overflow: hidden; /\* Ensure rounded corners apply to children \*/

        }

        /\* Custom scrollbar for report list \*/

        .report-list-container {

            max-height: 60vh; /\* Limit height for scrolling \*/

            overflow-y: auto;

            scrollbar-width: thin; /\* For Firefox \*/

            scrollbar-color: #cbd5e1 #f7fafc; /\* For Firefox \*/

        }

        .report-list-container::-webkit-scrollbar {

            width: 8px;

        }

        .report-list-container::-webkit-scrollbar-track {

            background: #f7fafc;

            border-radius: 10px;

        }

        .report-list-container::-webkit-scrollbar-thumb {

            background-color: #cbd5e1;

            border-radius: 10px;

            border: 2px solid #f7fafc;

        }

    </style>

</head>

<body>

    <div id="app" class="container p-6 md:p-10">

        <!-- Loading Spinner -->

        <div id="loading-spinner" class="absolute inset-0 flex items-center justify-center bg-gray-100 bg-opacity-75 z-50 hidden">

            <div class="animate-spin rounded-full h-16 w-16 border-t-4 border-b-4 border-blue-500"></div>

        </div>

        <!-- Auth Section -->

        <div id="auth-section" class="flex flex-col items-center justify-center text-center p-6">

            <h1 class="text-3xl md:text-4xl font-bold text-gray-800 mb-6">Chào mừng đến với AI - Cuộc sống học đường không bạo lực</h1>

            <p class="text-gray-600 text-lg mb-8">Một nền tảng hỗ trợ phát hiện và ngăn chặn bạo lực học đường.</p>

            <button id="start-app-btn" class="bg-blue-600 hover:bg-blue-700 text-white font-semibold py-3 px-8 rounded-full shadow-lg transition duration-300 ease-in-out transform hover:scale-105">

                Bắt đầu ứng dụng

            </button>

        </div>

        <!-- Main Application Section (Hidden by default) -->

        <div id="main-app-section" class="hidden">

            <h1 class="text-3xl md:text-4xl font-bold text-gray-800 mb-6 text-center">AI - Cuộc sống học đường không bạo lực</h1>

            <p id="user-id-display" class="text-sm text-gray-500 text-center mb-6">ID người dùng của bạn: <span class="font-mono text-gray-700"></span></p>

            <!-- Role Selection -->

            <div class="flex justify-center gap-4 mb-8">

                <button id="show-student-form-btn" class="flex-1 bg-green-500 hover:bg-green-600 text-white font-semibold py-3 px-6 rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-105">

                    Tôi là Học sinh

                </button>

                <button id="show-teacher-dashboard-btn" class="flex-1 bg-purple-500 hover:bg-purple-600 text-white font-semibold py-3 px-6 rounded-lg shadow-md transition duration-300 ease-in-out transform hover:scale-105">

                    Tôi là Giáo viên/Quản lý

                </button>

            </div>

            <!-- Student Report Form -->

            <div id="student-form" class="bg-blue-50 p-6 rounded-xl shadow-inner mb-8 hidden">

                <h2 class="text-2xl font-semibold text-blue-800 mb-4">Gửi báo cáo ẩn danh</h2>

                <div class="mb-4">

                    <label for="report-text" class="block text-gray-700 text-sm font-bold mb-2">Mô tả sự việc (càng chi tiết càng tốt):</label>

                    <textarea id="report-text" rows="5" class="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700 leading-tight focus:outline-none focus:ring-2 focus:ring-blue-400 focus:border-transparent transition duration-200 ease-in-out" placeholder="Ví dụ: Bạn A bị một nhóm bạn khác trêu chọc và đẩy ngã trong sân trường..."></textarea>

                </div>

                <button id="submit-report-btn" class="bg-blue-600 hover:bg-blue-700 text-white font-semibold py-3 px-6 rounded-lg shadow-lg transition duration-300 ease-in-out transform hover:scale-105">

                    Gửi báo cáo

                </button>

                <p id="report-message" class="text-sm mt-4 text-center text-gray-600 hidden"></p>

            </div>

            <!-- Teacher/Admin Dashboard -->

            <div id="teacher-dashboard" class="bg-gray-50 p-6 rounded-xl shadow-inner hidden">

                <h2 class="text-2xl font-semibold text-gray-800 mb-4">Bảng điều khiển báo cáo</h2>

                <div id="report-list-container" class="report-list-container border border-gray-200 rounded-lg p-4 bg-white shadow-sm">

                    <p id="no-reports-message" class="text-gray-500 text-center py-4">Chưa có báo cáo nào.</p>

                    <div id="reports-list" class="space-y-4">

                        <!-- Reports will be dynamically loaded here -->

                    </div>

                </div>

            </div>

        </div>

    </div>

    <!-- Firebase SDKs -->

    <script type="module">

        // Import các module Firebase cần thiết

        import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";

        import { getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";

        import { getFirestore, collection, addDoc, onSnapshot, query, orderBy, serverTimestamp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

        // Lấy các biến môi trường từ Canvas

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

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

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

        // Khởi tạo Firebase

        const app = initializeApp(firebaseConfig);

        const db = getFirestore(app);

        const auth = getAuth(app);

        // Lấy các phần tử DOM

        const loadingSpinner = document.getElementById('loading-spinner');

        const authSection = document.getElementById('auth-section');

        const mainAppSection = document.getElementById('main-app-section');

        const startAppBtn = document.getElementById('start-app-btn');

        const userIdDisplay = document.querySelector('#user-id-display span');

        const showStudentFormBtn = document.getElementById('show-student-form-btn');

        const showTeacherDashboardBtn = document.getElementById('show-teacher-dashboard-btn');

        const studentForm = document.getElementById('student-form');

        const teacherDashboard = document.getElementById('teacher-dashboard');

        const reportTextarea = document.getElementById('report-text');

        const submitReportBtn = document.getElementById('submit-report-btn');

        const reportMessage = document.getElementById('report-message');

        const reportsList = document.getElementById('reports-list');

        const noReportsMessage = document.getElementById('no-reports-message');

        let currentUserId = null; // Biến để lưu trữ ID người dùng hiện tại

        // --- Hàm hiển thị/ẩn spinner ---

        function showSpinner() {

            loadingSpinner.classList.remove('hidden');

        }

        function hideSpinner() {

            loadingSpinner.classList.add('hidden');

        }

        // --- Hàm mô phỏng AI phát hiện bạo lực ---

        function simulateAIDetection(text) {

            const lowerText = text.toLowerCase();

            let threatLevel = 0;

            let category = "Bình thường";

            let keywords = [];

            // Các từ khóa và cụm từ mô phỏng dấu hiệu bạo lực

            const violentKeywords = {

                "đánh": 0.8, "đấm": 0.8, "đá": 0.8, "xô đẩy": 0.7, "bắt nạt": 0.9,

                "chửi": 0.7, "sỉ nhục": 0.7, "đe dọa": 0.9, "hăm dọa": 0.8,

                "tẩy chay": 0.7, "cô lập": 0.6, "làm nhục": 0.8,

                "ăn đấm": 0.9, "ăn tát": 0.9, "đòi nợ": 0.7, "bắt ép": 0.8

            };

            for (const keyword in violentKeywords) {

                if (lowerText.includes(keyword)) {

                    threatLevel = Math.max(threatLevel, violentKeywords[keyword]);

                    keywords.push(keyword);

                }

            }

            // Tăng mức độ đe dọa nếu có nhiều từ khóa

            if (keywords.length > 1) {

                threatLevel = Math.min(1.0, threatLevel \* 1.2);

            }

            // Phân loại dựa trên mức độ đe dọa

            if (threatLevel >= 0.8) {

                category = "Nguy hiểm cao";

            } else if (threatLevel >= 0.5) {

                category = "Cần chú ý";

            } else if (threatLevel > 0) {

                category = "Có dấu hiệu";

            } else {

                // Nếu không có từ khóa bạo lực, thêm một chút ngẫu nhiên để mô phỏng AI không hoàn hảo

                if (Math.random() < 0.05) { // 5% khả năng bị gắn nhãn "có dấu hiệu" ngẫu nhiên

                    category = "Có dấu hiệu";

                    threatLevel = Math.random() \* 0.4 + 0.1; // 0.1 - 0.5

                }

            }

            // Làm tròn mức độ đe dọa

            threatLevel = parseFloat(threatLevel.toFixed(2));

            return {

                threatLevel: threatLevel,

                category: category,

                detectedKeywords: keywords.length > 0 ? keywords.join(', ') : 'Không có'

            };

        }

        // --- Xử lý trạng thái xác thực Firebase ---

        onAuthStateChanged(auth, async (user) => {

            if (user) {

                currentUserId = user.uid;

                userIdDisplay.textContent = currentUserId;

                authSection.classList.add('hidden');

                mainAppSection.classList.remove('hidden');

                hideSpinner();

                // Lắng nghe các báo cáo theo thời gian thực

                listenForReports();

            } else {

                // Nếu chưa đăng nhập, hiển thị màn hình đăng nhập

                authSection.classList.remove('hidden');

                mainAppSection.classList.add('hidden');

                hideSpinner();

            }

        });

        // --- Xử lý nút "Bắt đầu ứng dụng" ---

        startAppBtn.addEventListener('click', async () => {

            showSpinner();

            try {

                if (initialAuthToken) {

                    await signInWithCustomToken(auth, initialAuthToken);

                } else {

                    await signInAnonymously(auth);

                }

            } catch (error) {

                console.error("Lỗi đăng nhập:", error);

                reportMessage.textContent = "Lỗi đăng nhập. Vui lòng thử lại.";

                reportMessage.classList.remove('hidden');

                hideSpinner();

            }

        });

        // --- Chuyển đổi giữa form học sinh và dashboard giáo viên ---

        showStudentFormBtn.addEventListener('click', () => {

            studentForm.classList.remove('hidden');

            teacherDashboard.classList.add('hidden');

            showStudentFormBtn.classList.add('bg-green-600');

            showStudentFormBtn.classList.remove('bg-green-500');

            showTeacherDashboardBtn.classList.remove('bg-purple-600');

            showTeacherDashboardBtn.classList.add('bg-purple-500');

        });

        showTeacherDashboardBtn.addEventListener('click', () => {

            studentForm.classList.add('hidden');

            teacherDashboard.classList.remove('hidden');

            showTeacherDashboardBtn.classList.add('bg-purple-600');

            showTeacherDashboardBtn.classList.remove('bg-purple-500');

            showStudentFormBtn.classList.remove('bg-green-600');

            showStudentFormBtn.classList.add('bg-green-500');

        });

        // --- Xử lý gửi báo cáo ---

        submitReportBtn.addEventListener('click', async () => {

            const reportText = reportTextarea.value.trim();

            if (!reportText) {

                reportMessage.textContent = "Vui lòng nhập nội dung báo cáo.";

                reportMessage.className = 'text-sm mt-4 text-center text-red-600';

                reportMessage.classList.remove('hidden');

                return;

            }

            showSpinner();

            submitReportBtn.disabled = true; // Vô hiệu hóa nút để tránh gửi nhiều lần

            reportMessage.classList.add('hidden'); // Ẩn thông báo cũ

            try {

                // Mô phỏng AI phân tích

                const aiResult = simulateAIDetection(reportText);

                // Lưu báo cáo vào Firestore

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

                await addDoc(reportsCollectionRef, {

                    reporterId: currentUserId, // ID của người gửi (ẩn danh)

                    reportText: reportText,

                    aiThreatLevel: aiResult.threatLevel,

                    aiCategory: aiResult.category,

                    aiDetectedKeywords: aiResult.detectedKeywords,

                    timestamp: serverTimestamp() // Thời gian gửi báo cáo

                });

                reportTextarea.value = ''; // Xóa nội dung sau khi gửi

                reportMessage.textContent = "Báo cáo của bạn đã được gửi thành công. Cảm ơn bạn!";

                reportMessage.className = 'text-sm mt-4 text-center text-green-600';

                reportMessage.classList.remove('hidden');

            } catch (e) {

                console.error("Lỗi khi gửi báo cáo: ", e);

                reportMessage.textContent = "Đã xảy ra lỗi khi gửi báo cáo. Vui lòng thử lại.";

                reportMessage.className = 'text-sm mt-4 text-center text-red-600';

                reportMessage.classList.remove('hidden');

            } finally {

                hideSpinner();

                submitReportBtn.disabled = false; // Kích hoạt lại nút

            }

        });

        // --- Lắng nghe báo cáo theo thời gian thực cho Dashboard giáo viên ---

        function listenForReports() {

            // Sắp xếp theo thời gian gửi giảm dần (mới nhất lên đầu)

            // Lưu ý: Firestore yêu cầu tạo chỉ mục cho trường 'timestamp' nếu bạn muốn sắp xếp.

            // Trong môi trường Canvas, có thể không cần chỉ mục cụ thể cho các trường hợp đơn giản.

            // Nếu gặp lỗi, có thể bỏ orderBy và sắp xếp bằng JS sau khi fetch.

            // query(collection(db, `artifacts/${appId}/public/data/reports`), orderBy('timestamp', 'desc'))

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

            onSnapshot(q, (snapshot) => {

                reportsList.innerHTML = ''; // Xóa danh sách cũ

                if (snapshot.empty) {

                    noReportsMessage.classList.remove('hidden');

                } else {

                    noReportsMessage.classList.add('hidden');

                    snapshot.docs.forEach((doc) => {

                        const report = doc.data();

                        const reportId = doc.id;

                        const timestamp = report.timestamp ? new Date(report.timestamp.toDate()).toLocaleString('vi-VN') : 'Đang cập nhật';

                        // Xác định màu sắc dựa trên mức độ đe dọa

                        let threatColor = 'text-gray-700';

                        if (report.aiThreatLevel >= 0.8) {

                            threatColor = 'text-red-600 font-bold';

                        } else if (report.aiThreatLevel >= 0.5) {

                            threatColor = 'text-orange-500 font-semibold';

                        } else if (report.aiThreatLevel > 0) {

                            threatColor = 'text-yellow-600';

                        }

                        const reportItem = `

                            <div class="bg-white p-4 rounded-lg shadow-sm border border-gray-100">

                                <p class="text-sm text-gray-500 mb-1">ID báo cáo: <span class="font-mono">${reportId.substring(0, 8)}...</span></p>

                                <p class="text-sm text-gray-500 mb-2">Người gửi (ID ẩn danh): <span class="font-mono">${report.reporterId.substring(0, 8)}...</span></p>

                                <p class="text-gray-800 mb-2 font-medium">Nội dung: <span class="font-normal">${report.reportText}</span></p>

                                <p class="text-sm ${threatColor}">Mức độ đe dọa AI: <span class="font-bold">${(report.aiThreatLevel \* 100).toFixed(0)}%</span> - Thể loại: <span class="font-bold">${report.aiCategory}</span></p>

                                <p class="text-xs text-gray-400 mt-2">Từ khóa phát hiện: ${report.aiDetectedKeywords}</p>

                                <p class="text-xs text-gray-400 mt-1">Thời gian: ${timestamp}</p>

                            </div>

                        `;

                        reportsList.innerHTML += reportItem;

                    });

                }

            }, (error) => {

                console.error("Lỗi khi lắng nghe báo cáo: ", error);

                reportsList.innerHTML = `<p class="text-red-600 text-center py-4">Không thể tải báo cáo. Vui lòng kiểm tra kết nối hoặc quyền truy cập.</p>`;

                noReportsMessage.classList.add('hidden');

            });

        }

    </script>

</body>

</html>