```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>HealthNest Prototype</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <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;
        }
        /* Custom scrollbar for a cleaner look */
        ::-webkit-scrollbar {
            width: 6px;
            height: 6px;
        }
        ::-webkit-scrollbar-thumb {
            background: #cbd5e1; /* Tailwind gray-300 */
            border-radius: 3px;
        }
        ::-webkit-scrollbar-thumb:hover {
            background: #94a3b8; /* Tailwind gray-400 */
        }
        .screen {
            display: none; /* Hidden by default, JS will manage
visibility */
        }
        .screen.active {
            display: flex; /* Use flex for screen layout */
            flex-direction: column;
        }
        /* Basic transition for screen changes (optional) */
        .screen {
            transition: opacity 0.3s ease-in-out;
        }
        /* Helper for fixed bottom nav */
        .pb-20 {
            padding-bottom: 5rem; /* Adjust based on nav height */
        }
        .h-screen-minus-nav {
            height: calc(100vh - 5rem); /* Adjust based on nav height
*/
        }
    </style>
</head>
```

```
<body class="bg-slate-50 text-slate-800">

    <div id="app-container" class="max-w-md mx-auto h-screen bg-white
shadow-lg flex flex-col">

        <header id="app-header" class="bg-slate-100 p-4 shadow sticky
top-0 z-10">
            <div class="flex items-center">
                <button id="back-button" class="hidden mr-3 p-1
rounded-full hover:bg-slate-200">
                    <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-arrow-left"><path d="m12 19-7-7 7-7"/><path
d="M19 12H5"/></svg>
                </button>
                <h1 id="header-title" class="text-xl font-semibold
text-slate-700">Health Records</h1>
            </div>
        </header>

        <main id="main-content" class="flex-grow overflow-y-auto
pb-20">

            <div id="screen-health-records" class="screen active p-4
space-y-4">
                <div class="relative">
                    <input type="text" id="search-records-input"
placeholder="Search records..." class="w-full p-3 pl-10 border
border-slate-300 rounded-lg focus:ring-2 focus:ring-teal-500
focus:border-teal-500 transition-shadow">
                    <svg xmlns="http://www.w3.org/2000/svg" width="20"
height="20" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-search absolute left-3 top-1/2 -translate-y-1/2
text-slate-400"><circle cx="11" cy="11" r="8"/><path d="m21
21-4.3-4.3"/></svg>
                </div>
                <h2 class="text-lg font-semibold
text-slate-600">Timeline</h2>
                <div id="records-list" class="space-y-3">
                    <p class="text-slate-500 text-center py-4">No
records found.</p>
                </div>
            </div>

            <div id="screen-add-record" class="screen p-6 space-y-6">
                <div class="border-2 border-dashed border-slate-300
rounded-lg p-8 flex flex-col items-center justify-center text-center
```

```
hover:border-teal-400 transition-colors cursor-pointer">
                    <svg xmlns="http://www.w3.org/2000/svg" width="48"
height="48" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-file-plus-2 text-slate-400 mb-2"><path d="M4
22h14a2 2 0 0 0 2-2V7.5L14.5 2H6a2 2 0 0 0-2 2v4"/><path d="M14
2v6h6"/><path d="M3 15h6"/><path d="M6 12v6"/></svg>
                    <p class="text-slate-600 font-medium">Add Image or
PDF</p>
                    <input type="file" id="file-upload-input"
class="hidden" accept="image/*,.pdf">
                    <p id="file-upload-name" class="text-sm
text-slate-500 mt-1"></p>
                </div>
                <div>
                    <label for="document-type" class="block text-sm
font-medium text-slate-700 mb-1">Document type</label>
                    <select id="document-type" class="w-full p-3
border border-slate-300 rounded-lg focus:ring-2 focus:ring-teal-500
focus:border-teal-500 transition-shadow">
                        <option value="">Select type</option>
                        <option value="Blood Test">Blood Test</option>
                        <option
value="Prescription">Prescription</option>
                        <option value="X-Ray Results">X-Ray
Results</option>
                        <option value="Lab Report">Lab Report</option>
                        <option value="Doctor Visit">Doctor Visit
Note</option>
                        <option value="Vaccination">Vaccination
Record</option>
                        <option value="Other">Other</option>
                    </select>
                </div>
                <div>
                    <label for="document-date" class="block text-sm
font-medium text-slate-700 mb-1">Date</label>
                    <input type="date" id="document-date"
class="w-full p-3 border border-slate-300 rounded-lg focus:ring-2
focus:ring-teal-500 focus:border-teal-500 transition-shadow">
                </div>
                <div>
                    <label for="document-name" class="block text-sm
font-medium text-slate-700 mb-1">Record Name / Description</label>
                    <input type="text" id="document-name"
placeholder="e.g., Annual Checkup Report" class="w-full p-3 border
border-slate-300 rounded-lg focus:ring-2 focus:ring-teal-500
focus:border-teal-500 transition-shadow">
                </div>
                <div class="flex gap-3 pt-4">
                     <button id="cancel-add-record" class="w-full
bg-slate-200 text-slate-700 p-3 rounded-lg font-semibold
hover:bg-slate-300 transition-colors">Cancel</button>
                    <button id="save-record-button" class="w-full
bg-teal-500 text-white p-3 rounded-lg font-semibold hover:bg-teal-600
transition-colors">Save Record</button>
                </div>
            </div>

            <div id="screen-profile-switcher" class="screen p-4
space-y-3">
                <h2 class="text-lg font-semibold text-slate-600
px-2">Family Profiles</h2>
                <div id="profiles-list" class="space-y-3">
                    </div>
            </div>

            <div id="screen-emergency-card" class="screen p-6
space-y-6">
                <div id="emergency-card-content" class="bg-amber-50
border border-amber-200 rounded-lg p-6 shadow-md">
                    <div class="flex items-center mb-6">
                        <img id="emergency-profile-avatar"
src="https://placehold.co/60x60/E2E8F0/475569?text=User" alt="Profile
Avatar" class="w-16 h-16 rounded-full mr-4 border-2 border-amber-300">
                        <div>
                            <h2 id="emergency-profile-name"
class="text-2xl font-bold text-amber-700"></h2>
                            <p class="text-sm
text-amber-600">Emergency Information</p>
                        </div>
                    </div>

                    <div class="space-y-4">
                        <div>
                            <h3 class="text-xs font-semibold
text-amber-500 uppercase tracking-wider">Date of Birth</h3>
                            <p id="emergency-dob"
class="text-amber-700 text-lg"></p>
                        </div>
                        <div>
                            <h3 class="text-xs font-semibold
text-amber-500 uppercase tracking-wider">Medical Conditions</h3>
                            <p id="emergency-conditions"
class="text-amber-700 text-lg"></p>
                        </div>
                        <div>
                            <h3 class="text-xs font-semibold
```

```
text-amber-500 uppercase tracking-wider">Medications</h3>
                            <p id="emergency-medications"
class="text-amber-700 text-lg"></p>
                        </div>
                    </div>
                </div>
                <div class="mt-6">
                    <h3 class="text-lg font-semibold text-slate-600
mb-3">Recent Records</h3>
                    <div id="emergency-recent-records"
class="space-y-2">
                        <p class="text-slate-500 text-center py-2">No
recent records available.</p>
                    </div>
                </div>
            </div>

            <div id="message-box" class="fixed bottom-24 left-1/2
-translate-x-1/2 bg-slate-800 text-white px-6 py-3 rounded-lg
shadow-xl text-sm opacity-0 transition-opacity duration-300 z-50">
                <p id="message-text"></p>
            </div>

        </main>

        <nav class="fixed bottom-0 left-0 right-0 max-w-md mx-auto
bg-white border-t border-slate-200 flex justify-around items-center
h-20 shadow-top z-10">
            <button data-screen="screen-health-records"
class="nav-button flex flex-col items-center justify-center
text-teal-600 p-2 rounded-md w-1/4">
                <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-home mb-1"><path d="m3 9 9-7 9 7v11a2 2 0 0 1-2
2H5a2 2 0 0 1-2-2z"/><polyline points="9 22 9 12 15 12 15 22"/></svg>
                <span class="text-xs font-medium">Home</span>
            </button>
            <button data-screen="screen-health-records"
class="nav-button flex flex-col items-center justify-center
text-slate-500 hover:text-teal-600 p-2 rounded-md w-1/4">
                 <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-search mb-1"><circle cx="11" cy="11" r="8"/><path
d="m21 21-4.3-4.3"/></svg>
                <span class="text-xs font-medium">Search</span>
            </button>
            <button data-screen="screen-add-record" class="nav-button
```

```
flex flex-col items-center justify-center text-slate-500
hover:text-teal-600 p-2 rounded-md w-1/4">
                <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-plus-circle mb-1"><circle cx="12" cy="12"
r="10"/><path d="M8 12h8"/><path d="M12 8v8"/></svg>
                <span class="text-xs font-medium">Add</span>
            </button>
            <button data-screen="screen-profile-switcher"
class="nav-button flex flex-col items-center justify-center
text-slate-500 hover:text-teal-600 p-2 rounded-md w-1/4">
                <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-users mb-1"><path d="M16 21v-2a4 4 0 0 0-4-4H6a4
4 0 0 0-4 4v2"/><circle cx="9" cy="7" r="4"/><path d="M22 21v-2a4 4 0
0 0-3-3.87"/><path d="M16 3.13a4 4 0 0 1 0 7.75"/></svg>
                <span class="text-xs font-medium">Profiles</span>
            </button>
        </nav>
    </div>

    <script>
        // Mock Data
        let profiles = [
            { id: 1, name: "Eliza Miller", avatarSeed: "Eliza", dob:
"01/01/1990", conditions: "Migraines (occasional)", medications:
"Sumatriptan (as needed)", isCurrentUser: true },
            { id: 2, name: "John Smith", avatarSeed: "John", dob:
"03/10/1975", conditions: "Hypertension, Type 2 Diabetes",
medications: "Lisinopril 10mg, Metformin 500mg", isCurrentUser: false
},
            { id: 3, name: "Christopher Doe", avatarSeed: "Chris",
dob: "15/06/2005", conditions: "Asthma (mild)", medications:
"Albuterol Inhaler (as needed)", isCurrentUser: false }
        ];

        let records = {
            1: [ // Eliza's records
                { id: 101, type: "Lab Report", name: "Annual Blood
Panel", date: "2024-07-10", fileUrl: "#", profileId: 1 },
                { id: 102, type: "Prescription", name: "Sumatriptan
Refill", date: "2024-05-12", fileUrl: "#", profileId: 1 },
                { id: 103, type: "Doctor Visit", name: "Follow-up on
Migraines", date: "2024-03-01", fileUrl: "#", profileId: 1 }
            ],
            2: [ // John's records
                { id: 201, type: "Blood Test", name: "A1C Check",
date: "2023-11-20", fileUrl: "#", profileId: 2 },
                { id: 202, type: "Prescription", name: "Lisinopril
Prescription", date: "2023-10-15", fileUrl: "#", profileId: 2 },
                { id: 203, type: "X-Ray Results", name: "Chest X-Ray
(Routine)", date: "2023-09-05", fileUrl: "#", profileId: 2 }
            ],
            3: [ // Christopher's records
                { id: 301, type: "Doctor Visit", name: "Asthma
Checkup", date: "2024-02-15", fileUrl: "#", profileId: 3 },
                { id: 302, type: "Vaccination", name: "Flu Shot",
date: "2023-10-01", fileUrl: "#", profileId: 3 }
            ]
        };

        let currentProfileId = profiles.find(p => p.isCurrentUser)?.id
|| profiles[0]?.id || 1;
        let previousScreenId = null; // For back button functionality

        // DOM Elements
        const screens = document.querySelectorAll('.screen');
        const navButtons = document.querySelectorAll('.nav-button');
        const headerTitle = document.getElementById('header-title');
        const backButton = document.getElementById('back-button');

        const recordsListEl = document.getElementById('records-list');
        const searchInput =
document.getElementById('search-records-input');

        const addRecordFileLabel =
document.getElementById('file-upload-name');
        const addRecordFileInput =
document.getElementById('file-upload-input');
        const addRecordFileUploadTrigger =
document.querySelector('#screen-add-record .border-dashed');
        const docTypeInput = document.getElementById('document-type');
        const docDateInput = document.getElementById('document-date');
        const docNameInput = document.getElementById('document-name');
        const saveRecordButton =
document.getElementById('save-record-button');
        const cancelAddRecordButton =
document.getElementById('cancel-add-record');

        const profilesListEl =
document.getElementById('profiles-list');

        const emergencyProfileAvatar =
document.getElementById('emergency-profile-avatar');
        const emergencyProfileName =
document.getElementById('emergency-profile-name');
```

```
        const emergencyDob = document.getElementById('emergency-dob');
        const emergencyConditions =
document.getElementById('emergency-conditions');
        const emergencyMedications =
document.getElementById('emergency-medications');
        const emergencyRecentRecordsEl =
document.getElementById('emergency-recent-records');

        const messageBox = document.getElementById('message-box');
        const messageText = document.getElementById('message-text');

        // Utility Functions
        function showMessage(message, duration = 3000) {
            messageText.textContent = message;
            messageBox.classList.remove('opacity-0');
            setTimeout(() => {
                messageBox.classList.add('opacity-0');
            }, duration);
        }

        function formatDate(dateString) {
            if (!dateString) return 'N/A';
            const date = new Date(dateString);
            // Make sure date is valid before formatting
            if (isNaN(date.getTime())) return dateString; // Return
original if invalid
            return date.toLocaleDateString('en-US', { year: 'numeric',
month: 'long', day: 'numeric' });
        }

        function getIconForRecordType(type) {
            // Simple icon mapping, can be expanded with more SVGs
            switch (type.toLowerCase()) {
                case 'blood test': return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-droplets text-red-500"><path d="M7 16.3c2.2 0 4-1.83 4-4.05
0-1.16-.57-2.26-1.71-3.19S7.29 6.75 7 5.3c-.29 1.45-1.14 2.84-2.29
3.76S3 11.1 3 12.25c0 2.22 1.8 4.05 4 4.05Z"/><path d="M12.56
6.6A10.97 10.97 0 0 0 0 14 3.02c.5 2.5 2 4.9 4 6.5s3 3.5 3 5.5a6.98 6.98
0 0 1-11.91 4.97"/></svg>`;
                case 'prescription': return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-pilcrow-left text-blue-500"><path d="M14 4v16"/><path d="M18
4v16"/><path d="M9.5 16H14V4H9.5a4.5 4.5 0 0 0 0 0 9h.01Z"/></svg>`;
                case 'x-ray results': return `<svg
```

```
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-bone text-slate-500"><path d="M17.24 7.24a5.002 5.002 0 0
0-7.075-7.075L10 0l-1.5 1.5L7.24 2.76a5.002 5.002 0 0 0-7.075 7.075L0
10l1.5 1.5L2.76 12.76a5.002 5.002 0 0 0 7.075 7.075L10 20l1.5-1.5
1.24-1.24a5.002 5.002 0 0 0 7.075-7.075L20 10l-1.5-1.5Z"/></svg>`;
                case 'lab report': return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-flask-conical text-green-500"><path d="M10 2v7.527a2 2 0 0
1-.211.896L4.72 20.55a1 1 0 0 0 .9 1.45h12.76a1 1 0 0 0
.9-1.45l-5.069-10.127A2 2 0 0 1 14 9.527V2"/><path d="M8.5
2h7"/></svg>`;
                case 'doctor visit': return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-clipboard-list text-purple-500"><rect width="8" height="4"
x="8" y="2" rx="1" ry="1"/><path d="M16 4h2a2 2 0 0 1 2 2v14a2 2 0 0
1-2 2H6a2 2 0 0 1-2-2V6a2 2 0 0 1 2-2h2"/><path d="M12 11h4"/><path
d="M12 16h4"/><path d="M8 11h.01"/><path d="M8 16h.01"/></svg>`;
                 case 'vaccination': return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-syringe text-sky-500"><path d="m18 2 4 4"/><path d="m17 7
3-3"/><path d="M19 9 8.7 19.3c-1 1-2.5 1-3.4 0l-.6-.6c-1-1-1-2.5
0-3.4L15 4"/><path d="m9 11 4 4"/><path d="m5 19-3 3"/></svg>`;
                default: return `<svg
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round" class="lucide
lucide-file-text text-slate-500"><path d="M15 2H6a2 2 0 0 0-2 2v16a2 2
0 0 0 2 2h12a2 2 0 0 0 2-2V7Z"/><path d="M14 2v4a2 2 0 0 0 2
2h4"/><path d="M10 9H8"/><path d="M16 13H8"/><path d="M16
17H8"/></svg>`;
            }
        }

        // Screen Management
        function switchScreen(screenId, newTitle, showBack = false,
passedData = null) {
            const currentActiveScreen =
document.querySelector('.screen.active');
            if (currentActiveScreen) {
                previousScreenId = currentActiveScreen.id; // Store
previous screen for back button
```

```
            } else {
                previousScreenId = 'screen-health-records'; // Default
if no active screen
            }

            screens.forEach(screen => {
                screen.classList.remove('active');
                screen.style.opacity = '0'; // For transition
            });
            const targetScreen = document.getElementById(screenId);
            if (targetScreen) {
                targetScreen.classList.add('active');
                setTimeout(() => targetScreen.style.opacity = '1',
50); // Trigger transition
                headerTitle.textContent = newTitle;
                if (showBack) {
                    backButton.classList.remove('hidden');
                } else {
                    backButton.classList.add('hidden');
                }

                // Update nav button active state
                navButtons.forEach(btn => {
                    if (btn.dataset.screen === screenId && !showBack)
{ // Don't highlight nav if it's a sub-screen like Emergency Card
                        btn.classList.remove('text-slate-500',
'hover:text-teal-600');
                        btn.classList.add('text-teal-600');
                    } else {
                        btn.classList.remove('text-teal-600');
                        btn.classList.add('text-slate-500',
'hover:text-teal-600');
                    }
                });

                // Scroll to top of new screen
                document.getElementById('main-content').scrollTop = 0;

                // Special handling for screens needing data refresh
                if (screenId === 'screen-health-records')
renderHealthRecords();
                if (screenId === 'screen-profile-switcher')
renderProfilesList();
                if (screenId === 'screen-emergency-card' && passedData
&& passedData.profileId) {
                    renderEmergencyCard(passedData.profileId);
                }
                if (screenId === 'screen-add-record') {
                    resetAddRecordForm();
                }


            } else {
                console.error("Screen not found:", screenId);
                if (currentActiveScreen) { // Fallback to previous
screen if target is not found
                    currentActiveScreen.classList.add('active');
                     setTimeout(() =>
currentActiveScreen.style.opacity = '1', 50);
                }
            }
        }

        // Render Functions
        function renderHealthRecords(searchTerm = '') {
            const userRecords = records[currentProfileId] || [];
            const filteredRecords = userRecords.filter(record =>

record.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

record.type.toLowerCase().includes(searchTerm.toLowerCase())
            ).sort((a, b) => new Date(b.date) - new Date(a.date)); //
Sort by date descending

            if (filteredRecords.length === 0) {
                recordsListEl.innerHTML = `<p class="text-slate-500
text-center py-4">${searchTerm ? 'No matching records found.' : 'No
records yet. Tap "Add" to create one.'}</p>`;
                return;
            }

            recordsListEl.innerHTML = filteredRecords.map(record => `
                <div class="bg-white p-4 rounded-lg shadow border
border-slate-200 hover:shadow-md transition-shadow cursor-pointer flex
items-start space-x-3" onclick="viewRecordDetail(${record.id},
${record.profileId})">
                    <div class="flex-shrink-0 w-10 h-10 flex
items-center justify-center rounded-full bg-slate-100">
                        ${getIconForRecordType(record.type)}
                    </div>
                    <div class="flex-grow">
                        <h3 class="font-semibold
text-slate-700">${record.name}</h3>
                        <p class="text-sm
text-slate-500">${record.type}</p>
                    </div>
                    <div class="text-right flex-shrink-0">
                        <p class="text-xs
```

```
text-slate-400">${formatDate(record.date)}</p>
                        <svg xmlns="http://www.w3.org/2000/svg"
width="16" height="16" viewBox="0 0 24 24" fill="none"
stroke="currentColor" stroke-width="2" stroke-linecap="round"
stroke-linejoin="round" class="lucide lucide-chevron-right
text-slate-400 mt-1 inline-block"><path d="m9 18 6-6-6-6"/></svg>
                    </div>
                </div>
            `).join('');
        }

        function viewRecordDetail(recordId, profileId) {
            const record = records[profileId]?.find(r => r.id ===
recordId);
            if (record) {
                showMessage(`Viewing: ${record.name} (${record.type})
- Details not implemented yet.`);
                // In a real app, you'd navigate to a detailed view
screen.
            }
        }


        function resetAddRecordForm() {
            docTypeInput.value = '';
            docDateInput.value = '';
            docNameInput.value = '';
            addRecordFileInput.value = ''; // Clear file input
            addRecordFileLabel.textContent = ''; // Clear file name
label
            // Set date to today by default
            const today = new Date().toISOString().split('T')[0];
            docDateInput.value = today;
        }

        function handleSaveRecord() {
            const type = docTypeInput.value;
            const date = docDateInput.value;
            const name = docNameInput.value.trim();
            const file = addRecordFileInput.files[0]; // Get the file

            if (!type || !date || !name) {
                showMessage("Please fill in all fields: Type, Date,
and Name.", 3000);
                return;
            }

            const newRecordId = Date.now(); // Simple unique ID
            const newRecord = {
                id: newRecordId,
                type: type,
                name: name,
                date: date,
                fileUrl: file ? URL.createObjectURL(file) : "#", //
Store a temporary URL for the image if uploaded
                fileName: file ? file.name : null,
                profileId: currentProfileId
            };

            if (!records[currentProfileId]) {
                records[currentProfileId] = [];
            }
            records[currentProfileId].push(newRecord);

            showMessage("Record saved successfully!", 2000);
            resetAddRecordForm();
            switchScreen('screen-health-records', 'Health Records');
// Go back to records list
        }

        function renderProfilesList() {
            profilesListEl.innerHTML = profiles.map(profile => `
                <div class="bg-white p-4 rounded-lg shadow border
border-slate-200 hover:shadow-md transition-shadow cursor-pointer flex
items-center space-x-4"
onclick="handleProfileSelection(${profile.id})">
                    <img
src="https://placehold.co/48x48/E2E8F0/475569?text=${profile.avatarSee
d.substring(0,1)}" alt="${profile.name}" class="w-12 h-12 rounded-full
border-2 ${profile.id === currentProfileId ? 'border-teal-500' :
'border-slate-300'}">
                    <div>
                        <h3 class="font-semibold text-slate-700
text-lg">${profile.name}</h3>
                        <p class="text-sm text-slate-500">${profile.id
=== currentProfileId ? 'Current Profile' : 'View Records / Emergency
Card'}</p>
                    </div>
                    <svg xmlns="http://www.w3.org/2000/svg" width="20"
height="20" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"
class="lucide lucide-chevron-right text-slate-400 ml-auto"><path d="m9
18 6-6-6-6"/></svg>
                </div>
            `).join('');
        }

        function handleProfileSelection(profileIdToView) {
            // Option 1: Switch current user and go to their records
            // currentProfileId = profileIdToView;
            // profiles.forEach(p => p.isCurrentUser = p.id ===
profileIdToView);
            // switchScreen('screen-health-records', 'Health
Records');
            // showMessage(`Switched to ${profiles.find(p=>p.id ===
profileIdToView).name}'s profile.`);

            // Option 2: Go to Emergency Card for selected profile (as
per mockups)
            switchScreen('screen-emergency-card', 'Emergency Card',
true, { profileId: profileIdToView });
        }

        function renderEmergencyCard(profileIdForCard) {
            const profile = profiles.find(p => p.id ===
profileIdForCard);
            if (!profile) {
                showMessage("Profile not found.", 3000);
                switchScreen(previousScreenId ||
'screen-health-records', 'Health Records'); // Go back
                return;
            }

            emergencyProfileAvatar.src =
`https://placehold.co/64x64/E2E8F0/475569?text=${profile.avatarSeed.su
bstring(0,1)}`;
            emergencyProfileAvatar.alt = profile.name;
            emergencyProfileName.textContent = profile.name;
            emergencyDob.textContent = profile.dob ?
formatDate(profile.dob) : 'N/A'; // Format date here if DOB is stored
as YYYY-MM-DD
            emergencyConditions.textContent = profile.conditions ||
'None specified';
            emergencyMedications.textContent = profile.medications ||
'None specified';

            // Render recent records for this profile
            const profileRecords = (records[profileIdForCard] ||
[]).sort((a,b) => new Date(b.date) - new Date(a.date)).slice(0,3); //
Get latest 3
            if(profileRecords.length > 0) {
                emergencyRecentRecordsEl.innerHTML =
profileRecords.map(record => `
                    <div class="bg-white p-3 rounded-md shadow-sm
border border-slate-200 flex items-start space-x-3">
                        <div class="flex-shrink-0 w-8 h-8 flex
items-center justify-center rounded-full bg-slate-100 text-sm">
```

```
                           ${getIconForRecordType(record.type)}
                        </div>
                        <div>
                            <p class="font-medium text-slate-700
text-sm">${record.name}</p>
                            <p class="text-xs
text-slate-500">${formatDate(record.date)}</p>
                        </div>
                    </div>
                `).join('');
            } else {
                emergencyRecentRecordsEl.innerHTML = `<p
class="text-slate-500 text-center py-2">No recent records available
for ${profile.name}.</p>`;
            }
        }


        // Event Listeners
        navButtons.forEach(button => {
            button.addEventListener('click', () => {
                const screenId = button.dataset.screen;
                let title = "HealthNest"; // Default title
                if (screenId === 'screen-health-records') title =
'Health Records';
                else if (screenId === 'screen-add-record') title =
'Add New Record';
                else if (screenId === 'screen-profile-switcher') title
= 'Select Profile';

                switchScreen(screenId, title);
            });
        });

        backButton.addEventListener('click', () => {
            // Determine the title for the previous screen
            let title = 'Health Records'; // Default
            if (previousScreenId === 'screen-health-records') title =
'Health Records';
            else if (previousScreenId === 'screen-profile-switcher')
title = 'Select Profile';
            // Add more else-if for other screens if they can be
"backed" into with specific titles

            switchScreen(previousScreenId || 'screen-health-records',
title, false);
        });

        searchInput.addEventListener('input', (e) => {
            renderHealthRecords(e.target.value);
        });

        if(addRecordFileUploadTrigger) {
            addRecordFileUploadTrigger.addEventListener('click', () =>
{
                addRecordFileInput.click();
            });
        }

        addRecordFileInput.addEventListener('change', (event) => {
            if (event.target.files && event.target.files[0]) {
                const file = event.target.files[0];
                addRecordFileLabel.textContent = `File: ${file.name}`;
            } else {
                addRecordFileLabel.textContent = '';
            }
        });

        saveRecordButton.addEventListener('click', handleSaveRecord);
        cancelAddRecordButton.addEventListener('click', () => {
             switchScreen('screen-health-records', 'Health Records');
        });


        // Initial Load
        // Set current profile based on isCurrentUser flag
        const initialProfile = profiles.find(p => p.isCurrentUser);
        if (initialProfile) {
            currentProfileId = initialProfile.id;
        } else if (profiles.length > 0) {
            currentProfileId = profiles[0].id; // Fallback to the
first profile
            profiles[0].isCurrentUser = true;
        }

        switchScreen('screen-health-records', 'Health Records'); //
Show home screen initially

    </script>
</body>
</html>
```