<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>EcoRide Mobile App (Utility Integrated)</title>

    <!-- Load Tailwind CSS -->

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

    <!-- Load Lucide Icons for aesthetic icons -->

    <script src="https://unpkg.com/lucide@latest"></script>

    <style>

        @import url('https://fonts.googleapis.com/css2?family=Inter:wght@100..900&display=swap');

        @import url('https://fonts.googleapis.com/css2?family=Caveat:wght@700&display=swap');

        body {

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

            background-color: #f0f4f8;

            display: flex;

            justify-content: center;

            align-items: flex-start;

            min-height: 100vh;

            padding: 20px;

        }

        .font-handwritten {

            font-family: 'Caveat', cursive;

        }

        .mobile-frame {

            width: 100%;

            max-width: 400px;

            min-height: 80vh;

            background-color: #ffffff;

            box-shadow: 0 20px 25px -5px rgba(0, 0, 0, 0.1), 0 10px 10px -5px rgba(0, 0, 0, 0.04);

            border-radius: 2rem;

            overflow: hidden;

            display: flex;

            flex-direction: column;

            position: relative;

        }

        /\* Custom scrollbar for better mobile feel \*/

        .scrollable-content::-webkit-scrollbar {

            width: 0px;

            background: transparent; /\* make scrollbar transparent \*/

        }

        .scrollable-content {

            flex-grow: 1;

            overflow-y: auto;

            padding: 0;

        }

        .screen-container {

            width: 100%;

            padding: 1.5rem;

            animation: fadeIn 0.3s ease-out;

        }

        @keyframes fadeIn {

            from { opacity: 0.5; transform: translateY(10px); }

            to { opacity: 1; transform: translateY(0); }

        }

        .nav-bar {

            border-top: 1px solid #f3f4f6;

        }

        /\* Gradient for header/buttons \*/

        .eco-gradient {

            background-image: linear-gradient(135deg, #4c51bf, #667eea);

        }

    </style>

</head>

<body>

    <div id="app" class="mobile-frame">

        <!-- Application content will be rendered here -->

    </div>

    <script>

        // --- Firebase Globals (Required for persistence in other apps) ---

        // Note: This app does not use Firestore, but these variables are required for the environment.

        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 : '';

        // --- Application State and Constants ---

        const SCREENS = {

            LOGIN: 'login',

            REGISTER: 'register',

            FORGOT\_PASSWORD: 'forgot\_password',

            HOME: 'home',

            LOCATION\_SELECT: 'location\_select',

            CONFIRM\_BOOKING: 'confirm\_booking',

            LIVE\_TRACKING: 'live\_tracking',

            EMERGENCY: 'emergency',

            PROFILE: 'profile',

            AI\_SUPPORT: 'ai\_support',

            IP\_VALIDATOR: 'ip\_validator', // NEW: The combined utility

        };

        const appState = {

            screen: SCREENS.LOGIN,

            isAuthenticated: false,

            // Mock data for ride booking

            pickup: 'Current Location (Simulated)',

            dropoff: null,

            vehicleType: 'Eco',

            rideStatus: 'Waiting for Driver',

            driverName: 'Elias K.',

            driverRating: 4.8,

            licensePlate: 'ECO-738',

            driverPhoto: 'https://placehold.co/100x100/4c51bf/ffffff?text=Elias',

            tripTime: 8, // minutes

            tripCost: 14.50, // USD

        };

        // --- Core Utility Functions ---

        /\*\*

         \* Navigates to a new screen and updates the URL hash for history.

         \* @param {string} screen The screen constant from SCREENS.

         \*/

        function navigateTo(screen, data = {}) {

            if (!screen) {

                console.error("Attempted navigation to null screen.");

                return;

            }

            // Check if user is trying to access a protected screen without auth

            if (!appState.isAuthenticated && screen !== SCREENS.LOGIN && screen !== SCREENS.REGISTER && screen !== SCREENS.FORGOT\_PASSWORD) {

                appState.screen = SCREENS.LOGIN;

            } else {

                appState.screen = screen;

            }

            // Merge any passed data into appState

            Object.assign(appState, data);

            // Set hash for back/forward functionality (simulated)

            window.location.hash = screen;

            render();

        }

        /\*\*

         \* Generic UI message box instead of alert/confirm.

         \* @param {string} message

         \* @param {string} title

         \*/

        function showMessage(title, message) {

            const container = document.getElementById('message-box-container');

            if (!container) return;

            container.innerHTML = `

                <div id="message-modal" class="fixed inset-0 bg-gray-900 bg-opacity-70 flex items-center justify-center p-4 z-50 transition-opacity duration-300">

                    <div class="bg-white p-6 rounded-xl shadow-2xl max-w-sm w-full transform transition-all duration-300 scale-100">

                        <h3 class="text-xl font-bold mb-3 text-indigo-700">${title}</h3>

                        <p class="text-gray-600 mb-6">${message}</p>

                        <button onclick="closeMessage()" class="w-full eco-gradient text-white py-2 rounded-xl font-semibold hover:opacity-90 transition duration-150">

                            OK

                        </button>

                    </div>

                </div>

            `;

        }

        function closeMessage() {

            const container = document.getElementById('message-box-container');

            if (container) {

                container.innerHTML = '';

            }

        }

        // --- Authentication Logic (MOCK) ---

        function handleLogin() {

            // Mock successful login

            appState.isAuthenticated = true;

            navigateTo(SCREENS.HOME);

            showMessage("Login Successful!", "Welcome back to EcoRide. Finding sustainable rides for you.");

        }

        function handleLogout() {

            appState.isAuthenticated = false;

            appState.pickup = 'Current Location (Simulated)';

            appState.dropoff = null;

            navigateTo(SCREENS.LOGIN);

        }

        // --- IP Validator Logic (NEW) ---

        /\*\*

         \* Robust function to validate an IPv4 address.

         \* @param {string} ip The IP address string to validate.

         \* @returns {boolean} True if valid IPv4, false otherwise.

         \*/

        function isValidIP(ip) {

            if (typeof ip !== 'string') return false;

            // Trim whitespace and check for empty string

            const trimmedIp = ip.trim();

            if (trimmedIp === '') return false;

            // Regex for IPv4 format: N.N.N.N where N is 1-3 digits

            const ipv4Regex = /^(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})$/;

            if (!ipv4Regex.test(trimmedIp)) {

                return false;

            }

            // Check if each octet is within the 0-255 range and does not have leading zeros unless it's '0'

            return trimmedIp.split('.').every(octet => {

                // Parse the octet as an integer

                const num = parseInt(octet, 10);

                // 1. Check if it's within 0-255 range

                if (num < 0 || num > 255) {

                    return false;

                }

                // 2. Check for leading zeros (e.g., '01', '007')

                // This is valid only if the octet is '0' itself.

                // If the parsed number converts back to a shorter string, it had a leading zero (e.g., '01' -> 1)

                if (octet.length > 1 && octet.startsWith('0')) {

                    return false;

                }

                return true;

            });

        }

        /\*\*

         \* Handles the IP validation button click.

         \*/

        function validateIPAddress() {

            const ipInput = document.getElementById('ip-input');

            const resultElement = document.getElementById('ip-validation-result');

            const ip = ipInput.value;

            if (!ip) {

                resultElement.innerHTML = `<p class="text-sm font-semibold text-red-500 mt-2">Please enter an IP address to check.</p>`;

                return;

            }

            const isValid = isValidIP(ip);

            if (isValid) {

                resultElement.innerHTML = `

                    <div class="bg-green-100 border border-green-400 text-green-700 px-4 py-3 rounded-xl relative mt-4 shadow-md">

                        <strong class="font-bold mr-2">VALID!</strong>

                        <span class="block sm:inline">The IP address "${ip}" is correctly formatted.</span>

                    </div>

                `;

                ipInput.classList.remove('border-red-500');

                ipInput.classList.add('border-green-500');

            } else {

                resultElement.innerHTML = `

                    <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded-xl relative mt-4 shadow-md">

                        <strong class="font-bold mr-2">INVALID!</strong>

                        <span class="block sm:inline">The IP address "${ip}" is NOT a valid IPv4 address. Check the format and octet ranges (0-255).</span>

                    </div>

                `;

                ipInput.classList.remove('border-green-500');

                ipInput.classList.add('border-red-500');

            }

        }

        // --- Screen Rendering Functions ---

        function renderNavBar() {

            const app = document.getElementById('app');

            let navBarHtml = '';

            if (appState.isAuthenticated && appState.screen !== SCREENS.LOGIN) {

                // Determine which icon is active

                const getIconClass = (screen) => appState.screen === screen ? 'text-indigo-600' : 'text-gray-400 hover:text-indigo-500';

                navBarHtml = `

                    <div class="nav-bar flex justify-around p-3 bg-white fixed bottom-0 w-full max-w-[400px] rounded-b-2xl z-40">

                        <button onclick="navigateTo(SCREENS.HOME)" class="flex flex-col items-center transition duration-150">

                            <i data-lucide="home" class="w-6 h-6 ${getIconClass(SCREENS.HOME)}"></i>

                            <span class="text-xs ${getIconClass(SCREENS.HOME)} mt-1">Home</span>

                        </button>

                        <button onclick="navigateTo(SCREENS.AI\_SUPPORT)" class="flex flex-col items-center transition duration-150">

                            <i data-lucide="message-square-text" class="w-6 h-6 ${getIconClass(SCREENS.AI\_SUPPORT)}"></i>

                            <span class="text-xs ${getIconClass(SCREENS.AI\_SUPPORT)} mt-1">Chat</span>

                        </button>

                        <button onclick="navigateTo(SCREENS.EMERGENCY)" class="flex flex-col items-center transition duration-150">

                            <i data-lucide="siren" class="w-6 h-6 text-red-500"></i>

                            <span class="text-xs text-red-500 mt-1">SOS</span>

                        </button>

                        <button onclick="navigateTo(SCREENS.PROFILE)" class="flex flex-col items-center transition duration-150">

                            <i data-lucide="user" class="w-6 h-6 ${getIconClass(SCREENS.PROFILE)}"></i>

                            <span class="text-xs ${getIconClass(SCREENS.PROFILE)} mt-1">Profile</span>

                        </button>

                    </div>

                `;

            }

            app.insertAdjacentHTML('beforeend', navBarHtml);

        }

        function renderHeader(title, backScreen = null) {

            return `

                <header class="p-4 flex items-center justify-between bg-white sticky top-0 z-10 border-b border-gray-100 rounded-t-2xl">

                    ${backScreen ?

                        `<button onclick="navigateTo('${backScreen}')" class="text-indigo-600 p-2 rounded-full hover:bg-indigo-50 transition duration-150">

                            <i data-lucide="chevron-left" class="w-6 h-6"></i>

                        </button>`

                        : `<div class="w-6 h-6"></div>` // Spacer

                    }

                    <h1 class="text-xl font-bold text-gray-800">${title}</h1>

                    ${appState.screen === SCREENS.HOME ?

                        `<button onclick="navigateTo(SCREENS.PROFILE)" class="p-2 rounded-full hover:bg-gray-100 transition duration-150">

                            <i data-lucide="settings" class="w-6 h-6 text-gray-500"></i>

                        </button>`

                        : `<div class="w-6 h-6"></div>` // Spacer

                    }

                </header>

            `;

        }

        function renderLoginScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                <div class="scrollable-content flex flex-col justify-between h-full">

                    <div class="h-full eco-gradient flex flex-col items-center justify-center p-8 rounded-t-2xl">

                        <i data-lucide="leaf" class="w-16 h-16 text-white mb-2"></i>

                        <h1 class="text-4xl font-handwritten text-white font-bold mb-2">EcoRide</h1>

                        <p class="text-white text-opacity-80 text-center">Your sustainable journey starts here.</p>

                    </div>

                    <div class="bg-white p-6 rounded-b-2xl shadow-inner">

                        <h2 class="text-2xl font-bold text-gray-800 mb-6 text-center">Sign In</h2>

                        <form onsubmit="event.preventDefault(); handleLogin();">

                            <div class="mb-4">

                                <label for="email" class="block text-sm font-medium text-gray-700 mb-1">Email</label>

                                <input type="email" id="email" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" placeholder="user@example.com" required>

                            </div>

                            <div class="mb-6">

                                <label for="password" class="block text-sm font-medium text-gray-700 mb-1">Password</label>

                                <input type="password" id="password" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" placeholder="••••••••" required>

                            </div>

                            <button type="submit" class="w-full eco-gradient text-white py-3 rounded-xl font-semibold text-lg shadow-lg shadow-indigo-500/50 hover:opacity-90 transition duration-150">

                                Login

                            </button>

                        </form>

                        <div class="mt-6 text-center">

                            <button onclick="navigateTo(SCREENS.FORGOT\_PASSWORD)" class="text-sm text-indigo-600 hover:text-indigo-800 transition duration-150">Forgot Password?</button>

                            <p class="text-sm text-gray-500 mt-2">

                                Don't have an account?

                                <button onclick="navigateTo(SCREENS.REGISTER)" class="font-medium text-indigo-600 hover:text-indigo-800 transition duration-150">Sign up</button>

                            </p>

                        </div>

                    </div>

                </div>

            `;

        }

        function renderRegistrationScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Create Account", SCREENS.LOGIN)}

                <div class="scrollable-content screen-container">

                    <p class="text-gray-500 mb-6 text-center">Join EcoRide for a greener journey.</p>

                    <form onsubmit="event.preventDefault(); showMessage('Registration Success', 'Your account is ready! Please log in.'); navigateTo(SCREENS.LOGIN);">

                        <div class="mb-4">

                            <label for="reg-name" class="block text-sm font-medium text-gray-700 mb-1">Full Name</label>

                            <input type="text" id="reg-name" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" required>

                        </div>

                        <div class="mb-4">

                            <label for="reg-email" class="block text-sm font-medium text-gray-700 mb-1">Email</label>

                            <input type="email" id="reg-email" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" required>

                        </div>

                        <div class="mb-6">

                            <label for="reg-password" class="block text-sm font-medium text-gray-700 mb-1">Password</label>

                            <input type="password" id="reg-password" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" placeholder="Min 6 characters" required>

                        </div>

                        <button type="submit" class="w-full eco-gradient text-white py-3 rounded-xl font-semibold text-lg hover:opacity-90 transition duration-150">

                            Register

                        </button>

                    </form>

                </div>

            `;

        }

        function renderForgotPasswordScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Reset Password", SCREENS.LOGIN)}

                <div class="scrollable-content screen-container">

                    <p class="text-gray-500 mb-6 text-center">Enter your email and we'll send you a link to reset your password.</p>

                    <form onsubmit="event.preventDefault(); showMessage('Email Sent', 'If your account exists, a password reset link has been sent to your email.'); navigateTo(SCREENS.LOGIN);">

                        <div class="mb-6">

                            <label for="reset-email" class="block text-sm font-medium text-gray-700 mb-1">Email Address</label>

                            <input type="email" id="reset-email" class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" required>

                        </div>

                        <button type="submit" class="w-full eco-gradient text-white py-3 rounded-xl font-semibold text-lg hover:opacity-90 transition duration-150">

                            Send Reset Link

                        </button>

                    </form>

                </div>

            `;

        }

        function renderHomeScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("EcoRide", null)}

                <div class="scrollable-content p-4">

                    <!-- Pickup and Dropoff Selection -->

                    <div class="bg-white p-6 rounded-2xl shadow-xl border border-gray-100">

                        <h2 class="text-xl font-bold text-gray-800 mb-4">Where to today?</h2>

                        <!-- Pickup Input (Simulated) -->

                        <div class="flex items-center p-3 bg-gray-50 rounded-xl mb-3 border border-gray-200">

                            <i data-lucide="map-pin" class="w-5 h-5 text-green-600 mr-3"></i>

                            <input type="text" value="${appState.pickup}" class="bg-transparent w-full focus:outline-none text-gray-700 font-medium" readonly>

                        </div>

                        <!-- Dropoff Input -->

                        <button onclick="navigateTo(SCREENS.LOCATION\_SELECT)" class="w-full flex items-center p-3 bg-white border border-indigo-300 rounded-xl shadow-md hover:bg-indigo-50 transition duration-150">

                            <i data-lucide="send" class="w-5 h-5 text-indigo-600 mr-3"></i>

                            <span class="text-gray-700 font-medium">${appState.dropoff ? appState.dropoff : 'Enter Destination...'}</span>

                        </button>

                    </div>

                    <!-- Suggested Rides/Categories -->

                    <h2 class="text-lg font-bold text-gray-800 mt-8 mb-4">Popular Options</h2>

                    <div class="grid grid-cols-2 gap-4">

                        <div class="bg-white p-4 rounded-xl shadow-md flex items-center hover:shadow-lg transition duration-150">

                            <i data-lucide="leaf" class="w-8 h-8 text-green-600 mr-3"></i>

                            <div>

                                <p class="font-semibold text-gray-800">Eco Car</p>

                                <p class="text-sm text-gray-500">Low Emission</p>

                            </div>

                        </div>

                        <div class="bg-white p-4 rounded-xl shadow-md flex items-center hover:shadow-lg transition duration-150">

                            <i data-lucide="bolt" class="w-8 h-8 text-yellow-600 mr-3"></i>

                            <div>

                                <p class="font-semibold text-gray-800">Electric</p>

                                <p class="text-sm text-gray-500">Silent & Fast</p>

                            </div>

                        </div>

                        <div class="bg-white p-4 rounded-xl shadow-md flex items-center hover:shadow-lg transition duration-150">

                            <i data-lucide="users" class="w-8 h-8 text-indigo-600 mr-3"></i>

                            <div>

                                <p class="font-semibold text-gray-800">XL Ride</p>

                                <p class="text-sm text-gray-500">Group Travel</p>

                            </div>

                        </div>

                        <div class="bg-white p-4 rounded-xl shadow-md flex items-center hover:shadow-lg transition duration-150">

                            <i data-lucide="bike" class="w-8 h-8 text-pink-600 mr-3"></i>

                            <div>

                                <p class="font-semibold text-gray-800">Bike Share</p>

                                <p class="text-sm text-gray-500">Quick Trips</p>

                            </div>

                        </div>

                    </div>

                    <!-- Space for bottom navigation bar -->

                    <div class="h-16"></div>

                </div>

            `;

        }

        function renderLocationSelectionScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Select Dropoff", SCREENS.HOME)}

                <div class="scrollable-content p-4">

                    <div class="mb-4">

                        <label for="search-location" class="sr-only">Search</label>

                        <div class="relative">

                            <i data-lucide="search" class="w-5 h-5 text-gray-400 absolute left-3 top-1/2 transform -translate-y-1/2"></i>

                            <input type="text" id="search-location" oninput="filterLocations(this.value)" class="w-full pl-10 pr-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150" placeholder="Search for destination...">

                        </div>

                    </div>

                    <div id="location-list" class="space-y-3">

                        ${[

                            { name: "Central Business Tower", detail: "450m | Est. $12.50" },

                            { name: "Historical Museum Park", detail: "1.2km | Est. $18.00" },

                            { name: "Metro Station A", detail: "2.1km | Est. $8.90" },

                            { name: "University Campus West", detail: "3.5km | Est. $25.00" },

                        ].map(loc => `

                            <button onclick="navigateTo(SCREENS.CONFIRM\_BOOKING, { dropoff: '${loc.name}' })" class="w-full text-left p-4 bg-white border border-gray-100 rounded-xl shadow-sm hover:bg-indigo-50 transition duration-150">

                                <p class="font-semibold text-gray-800">${loc.name}</p>

                                <p class="text-sm text-indigo-600">${loc.detail}</p>

                            </button>

                        `).join('')}

                    </div>

                </div>

            `;

        }

        function renderConfirmBookingScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Confirm Ride", SCREENS.LOCATION\_SELECT)}

                <div class="scrollable-content p-4">

                    <div class="bg-white p-6 rounded-2xl shadow-xl border border-gray-100 mb-6">

                        <h2 class="text-xl font-bold text-gray-800 mb-4">Trip Details</h2>

                        <div class="space-y-3 border-b pb-4 mb-4">

                            <div class="flex items-center text-gray-600">

                                <i data-lucide="map-pin" class="w-5 h-5 text-green-600 mr-3"></i>

                                <span>Pickup: <strong class="text-gray-800">${appState.pickup}</strong></span>

                            </div>

                            <div class="flex items-center text-gray-600">

                                <i data-lucide="send" class="w-5 h-5 text-indigo-600 mr-3"></i>

                                <span>Dropoff: <strong class="text-gray-800">${appState.dropoff}</strong></span>

                            </div>

                        </div>

                        <div class="flex justify-between items-center mb-4">

                            <span class="font-medium text-lg text-gray-800">Vehicle:</span>

                            <span class="font-bold text-lg text-green-600">${appState.vehicleType}</span>

                        </div>

                        <div class="flex justify-between items-center">

                            <span class="font-medium text-lg text-gray-800">Estimated Cost:</span>

                            <span class="font-bold text-2xl text-indigo-600">$${appState.tripCost.toFixed(2)}</span>

                        </div>

                    </div>

                    <!-- Payment Method -->

                    <div class="bg-white p-6 rounded-2xl shadow-xl border border-gray-100 mb-6">

                        <h2 class="text-xl font-bold text-gray-800 mb-4">Payment</h2>

                        <div class="flex items-center justify-between p-3 bg-gray-50 rounded-xl border border-gray-200">

                            <div class="flex items-center">

                                <i data-lucide="credit-card" class="w-5 h-5 text-indigo-600 mr-3"></i>

                                <span class="text-gray-700">Visa ending in \*\*42</span>

                            </div>

                            <i data-lucide="chevron-right" class="w-4 h-4 text-gray-500"></i>

                        </div>

                    </div>

                    <!-- Action Button -->

                    <button onclick="navigateTo(SCREENS.LIVE\_TRACKING)" class="w-full eco-gradient text-white py-3 rounded-xl font-semibold text-lg shadow-lg shadow-indigo-500/50 hover:opacity-90 transition duration-150">

                        Book EcoRide Now

                    </button>

                    <p class="text-center text-xs text-gray-500 mt-4">By booking, you agree to our Terms of Service.</p>

                </div>

            `;

        }

        function renderLiveTrackingScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Live Tracking", SCREENS.HOME)}

                <div class="scrollable-content flex flex-col h-full relative">

                    <!-- Mock Map Area -->

                    <div class="flex-grow bg-gray-200 flex items-center justify-center p-4">

                        <div class="text-center">

                            <i data-lucide="map" class="w-12 h-12 text-gray-400 mb-2"></i>

                            <p class="text-gray-500">Live Map Simulation</p>

                            <p class="text-sm font-semibold text-indigo-600 mt-1">Driver is ${appState.tripTime} mins away.</p>

                        </div>

                    </div>

                    <!-- Driver Info Card -->

                    <div class="bg-white p-6 rounded-t-2xl shadow-2xl border-t border-gray-100">

                        <div class="flex items-center justify-between mb-4">

                            <div class="flex items-center">

                                <img src="${appState.driverPhoto}" onerror="this.onerror=null; this.src='https://placehold.co/100x100/4c51bf/ffffff?text=Elias';" class="w-16 h-16 rounded-full object-cover mr-4 border-2 border-green-500 shadow-md">

                                <div>

                                    <p class="text-xl font-bold text-gray-800">${appState.driverName}</p>

                                    <p class="text-sm text-gray-500">${appState.licensePlate} (${appState.vehicleType})</p>

                                    <div class="flex items-center text-yellow-500 text-sm mt-1">

                                        <i data-lucide="star" class="w-4 h-4 fill-yellow-500 mr-1"></i>

                                        <span>${appState.driverRating}</span>

                                    </div>

                                </div>

                            </div>

                            <button onclick="showMessage('Calling Driver', 'Simulating call to ${appState.driverName}')" class="p-3 bg-green-500 rounded-full text-white shadow-lg hover:bg-green-600 transition duration-150">

                                <i data-lucide="phone" class="w-6 h-6"></i>

                            </button>

                        </div>

                        <div class="flex justify-between text-center mt-2 border-t pt-4">

                            <button onclick="showMessage('Share Link Generated', 'Your ride details have been copied to the clipboard for sharing.')" class="flex-1 p-2 bg-indigo-50 rounded-xl text-indigo-600 font-medium mr-2 hover:bg-indigo-100">

                                <i data-lucide="share2" class="w-5 h-5 mx-auto mb-1"></i>

                                <span class="text-xs block">Share Ride</span>

                            </button>

                            <button onclick="navigateTo(SCREENS.EMERGENCY)" class="flex-1 p-2 bg-red-50 rounded-xl text-red-600 font-medium ml-2 hover:bg-red-100">

                                <i data-lucide="siren" class="w-5 h-5 mx-auto mb-1"></i>

                                <span class="text-xs block">SOS</span>

                            </button>

                        </div>

                    </div>

                    <!-- Spacer for bottom nav -->

                    <div class="h-16"></div>

                </div>

            `;

        }

        function renderEmergencyScreen() {

             const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("Emergency SOS", appState.isAuthenticated ? SCREENS.LIVE\_TRACKING : SCREENS.HOME)}

                <div class="scrollable-content p-6 flex flex-col items-center text-center">

                    <div class="w-40 h-40 eco-gradient rounded-full flex items-center justify-center mb-6 shadow-xl">

                         <i data-lucide="siren" class="w-20 h-20 text-white animate-pulse"></i>

                    </div>

                    <h2 class="text-2xl font-bold text-red-700 mb-2">Need Immediate Help?</h2>

                    <p class="text-gray-600 mb-8">Tap and hold the button below to alert emergency services and contact your primary emergency contact.</p>

                    <button

                        onmousedown="startEmergencyHold(this)"

                        ontouchstart="startEmergencyHold(this)"

                        onmouseup="cancelEmergencyHold(this)"

                        ontouchend="cancelEmergencyHold(this)"

                        class="w-full max-w-xs bg-red-600 text-white py-4 rounded-full font-extrabold text-xl shadow-2xl transition-all duration-300 hover:bg-red-700 active:scale-95"

                        id="emergency-button"

                    >

                        HOLD TO CALL SOS

                    </button>

                    <p id="hold-status" class="mt-4 text-sm text-red-500 font-semibold invisible">Holding... 3 seconds remaining</p>

                    <div class="mt-10 pt-6 border-t border-gray-200 w-full max-w-xs">

                        <h3 class="font-semibold text-gray-700 mb-3">Other Contacts</h3>

                        <button onclick="showMessage('Calling 911', 'Simulating call to emergency services.')" class="w-full p-3 mb-3 bg-white border border-gray-300 rounded-xl flex items-center justify-center hover:bg-gray-50">

                            <i data-lucide="phone-call" class="w-5 h-5 text-indigo-600 mr-3"></i>

                            Call Police / Ambulance

                        </button>

                         <button onclick="showMessage('Calling Friend', 'Simulating call to your designated emergency contact.')" class="w-full p-3 bg-white border border-gray-300 rounded-xl flex items-center justify-center hover:bg-gray-50">

                            <i data-lucide="user-cog" class="w-5 h-5 text-indigo-600 mr-3"></i>

                            Call Emergency Contact

                        </button>

                    </div>

                </div>

            `;

            let holdTimer = null;

            let countdown = 3;

            const holdDuration = 3000; // 3 seconds

            window.startEmergencyHold = function(button) {

                if (holdTimer) return;

                const status = document.getElementById('hold-status');

                status.classList.remove('invisible');

                status.textContent = `Holding... ${countdown} seconds remaining`;

                button.classList.add('bg-red-800');

                // Start countdown display

                const interval = setInterval(() => {

                    countdown--;

                    status.textContent = `Holding... ${countdown} seconds remaining`;

                    if (countdown <= 0) {

                        clearInterval(interval);

                    }

                }, 1000);

                holdTimer = setTimeout(() => {

                    clearInterval(interval);

                    showMessage('SOS Activated', 'Emergency alert has been sent to the necessary contacts and authorities. Help is on the way.');

                    cancelEmergencyHold(button);

                    navigateTo(SCREENS.HOME); // Return to home after activation

                }, holdDuration);

            };

            window.cancelEmergencyHold = function(button) {

                if (holdTimer) {

                    clearTimeout(holdTimer);

                    holdTimer = null;

                    countdown = 3;

                    button.classList.remove('bg-red-800');

                    const status = document.getElementById('hold-status');

                    if (status) {

                        status.classList.add('invisible');

                    }

                }

            };

            // Clear handlers on screen exit (important for memory)

            app.onunload = () => {

                window.startEmergencyHold = null;

                window.cancelEmergencyHold = null;

            };

        }

        function renderProfileScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("My Profile", SCREENS.HOME)}

                <div class="scrollable-content p-4">

                    <!-- User Info Card -->

                    <div class="flex flex-col items-center p-6 bg-white rounded-2xl shadow-xl border border-gray-100 mb-8">

                        <img src="https://placehold.co/100x100/667eea/ffffff?text=User" class="w-20 h-20 rounded-full object-cover mb-3 border-4 border-indigo-200 shadow-md">

                        <h2 class="text-2xl font-bold text-gray-800">Jane Doe</h2>

                        <p class="text-gray-500 mb-4">jane.doe@example.com</p>

                        <button onclick="showMessage('Edit Profile', 'Simulating profile editing page.')" class="text-sm text-indigo-600 font-medium flex items-center hover:text-indigo-800 transition duration-150">

                            <i data-lucide="edit-3" class="w-4 h-4 mr-1"></i> Edit Profile

                        </button>

                    </div>

                    <!-- Account Settings -->

                    <h3 class="text-lg font-semibold text-gray-700 mb-3">Account & Utilities</h3>

                    <div class="space-y-3 mb-8">

                        <!-- NEW IP Validator Link -->

                        <button onclick="navigateTo(SCREENS.IP\_VALIDATOR)" class="flex items-center w-full p-4 bg-white hover:bg-gray-50 rounded-xl shadow-md transition duration-150">

                            <i data-lucide="globe" class="w-6 h-6 text-indigo-600 mr-4"></i>

                            <span class="text-gray-700 font-medium">IP Address Validator (Tool)</span>

                            <i data-lucide="chevron-right" class="w-5 h-5 text-gray-400 ml-auto"></i>

                        </button>

                        <button onclick="showMessage('Payment', 'Simulating payment methods screen.')" class="flex items-center w-full p-4 bg-white hover:bg-gray-50 rounded-xl shadow-md transition duration-150">

                            <i data-lucide="credit-card" class="w-6 h-6 text-green-600 mr-4"></i>

                            <span class="text-gray-700 font-medium">Payment Methods</span>

                            <i data-lucide="chevron-right" class="w-5 h-5 text-gray-400 ml-auto"></i>

                        </button>

                        <button onclick="showMessage('History', 'Simulating ride history list.')" class="flex items-center w-full p-4 bg-white hover:bg-gray-50 rounded-xl shadow-md transition duration-150">

                            <i data-lucide="history" class="w-6 h-6 text-yellow-600 mr-4"></i>

                            <span class="text-gray-700 font-medium">Ride History</span>

                            <i data-lucide="chevron-right" class="w-5 h-5 text-gray-400 ml-auto"></i>

                        </button>

                    </div>

                    <!-- Logout Button -->

                    <button onclick="handleLogout()" class="w-full bg-red-500 text-white py-3 rounded-xl font-semibold text-lg hover:bg-red-600 transition duration-150">

                        Log Out

                    </button>

                     <!-- Spacer for bottom nav -->

                    <div class="h-16"></div>

                </div>

            `;

        }

        function renderAiSupportScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("EcoRide AI Chat", SCREENS.HOME)}

                <div class="scrollable-content p-4 flex flex-col justify-between h-full">

                    <div class="flex flex-col space-y-4 mb-4 flex-grow overflow-y-auto">

                        <div class="flex justify-start">

                            <div class="bg-indigo-100 text-indigo-800 p-3 rounded-xl rounded-bl-none max-w-xs shadow-sm">

                                Hello! I'm your EcoRide AI assistant. How can I help with your trip or account?

                            </div>

                        </div>

                        <div class="flex justify-end">

                            <div class="bg-green-100 text-green-800 p-3 rounded-xl rounded-br-none max-w-xs shadow-sm">

                                My driver was late, can I get a partial refund?

                            </div>

                        </div>

                        <div class="flex justify-start">

                            <div class="bg-indigo-100 text-indigo-800 p-3 rounded-xl rounded-bl-none max-w-xs shadow-sm">

                                I see. I've noted the issue and initiated a 15% credit to your account. Is there anything else?

                            </div>

                        </div>

                    </div>

                    <div class="border-t pt-4 bg-white">

                        <div class="flex">

                            <input type="text" placeholder="Type your message..." class="flex-grow px-4 py-3 border border-gray-300 rounded-l-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150">

                            <button onclick="showMessage('Sending Message', 'Simulating sending your message to AI.')" class="eco-gradient text-white px-5 py-3 rounded-r-xl font-semibold hover:opacity-90 transition duration-150">

                                <i data-lucide="send" class="w-5 h-5"></i>

                            </button>

                        </div>

                    </div>

                    <!-- Spacer for bottom nav -->

                    <div class="h-16"></div>

                </div>

            `;

        }

        function renderIPValidatorScreen() {

            const app = document.getElementById('app');

            app.innerHTML = `

                <div id="message-box-container"></div>

                ${renderHeader("IP Address Validator", SCREENS.PROFILE)}

                <div class="scrollable-content p-4">

                    <div class="bg-white p-6 rounded-2xl shadow-xl border border-gray-100">

                        <h2 class="text-xl font-bold text-gray-800 mb-4 flex items-center">

                            <i data-lucide="check-square" class="w-6 h-6 text-indigo-600 mr-2"></i>

                            Validate IPv4 Address

                        </h2>

                        <p class="text-gray-600 mb-6">Enter an IPv4 address (e.g., 192.168.1.1) to check if it's correctly formatted and within range (0-255 per octet).</p>

                        <div class="mb-6">

                            <label for="ip-input" class="block text-sm font-medium text-gray-700 mb-1">IP Address</label>

                            <input

                                type="text"

                                id="ip-input"

                                class="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-indigo-500 focus:border-indigo-500 transition duration-150"

                                placeholder="e.g., 255.255.255.0"

                                onkeydown="if(event.key === 'Enter') validateIPAddress()"

                            >

                        </div>

                        <button onclick="validateIPAddress()" class="w-full eco-gradient text-white py-3 rounded-xl font-semibold text-lg shadow-lg shadow-indigo-500/50 hover:opacity-90 transition duration-150">

                            Validate

                        </button>

                        <div id="ip-validation-result" class="mt-4 min-h-16">

                            <!-- Results displayed here -->

                        </div>

                    </div>

                    <h3 class="text-lg font-semibold text-gray-700 mt-8 mb-3">IP Validation Rules:</h3>

                    <ul class="text-sm text-gray-600 space-y-2 bg-white p-4 rounded-xl shadow-sm">

                        <li class="flex items-start">

                            <i data-lucide="check" class="w-4 h-4 text-green-500 mr-2 mt-1 flex-shrink-0"></i>

                            Must contain exactly four sets of numbers (octets).

                        </li>

                        <li class="flex items-start">

                            <i data-lucide="check" class="w-4 h-4 text-green-500 mr-2 mt-1 flex-shrink-0"></i>

                            Each octet must be separated by a dot (.).

                        </li>

                         <li class="flex items-start">

                            <i data-lucide="check" class="w-4 h-4 text-green-500 mr-2 mt-1 flex-shrink-0"></i>

                            Each octet must be a number between 0 and 255, inclusive.

                        </li>

                        <li class="flex items-start">

                            <i data-lucide="check" class="w-4 h-4 text-green-500 mr-2 mt-1 flex-shrink-0"></i>

                            Octets cannot have unnecessary leading zeros (e.g., '01' is invalid, but '0' is valid).

                        </li>

                    </ul>

                    <!-- Spacer for bottom nav -->

                    <div class="h-16"></div>

                </div>

            `;

            // Attach event listener for Enter key on the new input after rendering

            const ipInput = document.getElementById('ip-input');

            if (ipInput) {

                ipInput.addEventListener('keydown', (event) => {

                    if (event.key === 'Enter') {

                        event.preventDefault();

                        validateIPAddress();

                    }

                });

            }

        }

        // --- Main Render Function ---

        function render() {

            const app = document.getElementById('app');

            // Clear any existing nav bar before re-rendering

            app.querySelector('.nav-bar')?.remove();

            switch (appState.screen) {

                case SCREENS.HOME:

                    renderHomeScreen();

                    break;

                case SCREENS.REGISTER:

                    renderRegistrationScreen();

                    break;

                case SCREENS.FORGOT\_PASSWORD:

                    renderForgotPasswordScreen();

                    break;

                case SCREENS.LOCATION\_SELECT:

                    renderLocationSelectionScreen();

                    break;

                case SCREENS.CONFIRM\_BOOKING:

                    renderConfirmBookingScreen();

                    break;

                case SCREENS.LIVE\_TRACKING:

                    renderLiveTrackingScreen();

                    break;

                case SCREENS.EMERGENCY:

                    renderEmergencyScreen();

                    break;

                case SCREENS.PROFILE:

                    renderProfileScreen();

                    break;

                case SCREENS.AI\_SUPPORT:

                    renderAiSupportScreen();

                    break;

                case SCREENS.IP\_VALIDATOR: // NEW CASE FOR IP VALIDATOR

                    renderIPValidatorScreen();

                    break;

                case SCREENS.LOGIN:

                default:

                    renderLoginScreen();

                    break;

            }

            // Render the navigation bar if authenticated

            renderNavBar();

            // Re-create lucide icons after HTML is updated

            lucide.createIcons();

        }

        // --- Back/Forward Navigation Handler (uses hash) ---

        window.addEventListener('hashchange', () => {

            const screen = window.location.hash.substring(1);

            if (screen && screen !== appState.screen) {

                appState.screen = screen in SCREENS ? screen : SCREENS.LOGIN;

                render();

            }

        });

        // Initialize App on load

        window.onload = () => {

            // Check hash on load for deep linking or refresh

            if (window.location.hash) {

                const screen = window.location.hash.substring(1);

                if (screen in SCREENS) {

                    // Start in logged-in state if screen is not a public one

                    if (screen !== SCREENS.LOGIN && screen !== SCREENS.REGISTER && screen !== SCREENS.FORGOT\_PASSWORD) {

                        appState.isAuthenticated = true;

                    }

                    appState.screen = screen;

                }

            }

            render();

        };

    </script>

</body>

</html>