<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Hiking Trails Map | Speed-Proof</title>

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.header {

text-align: center;

color: white;

margin-bottom: 30px;

padding: 20px;

}

.header h1 {

font-size: 2.5rem;

margin-bottom: 10px;

text-shadow: 2px 2px 4px rgba(0,0,0,0.3);

}

.header p {

font-size: 1.2rem;

opacity: 0.9;

}

.map-container {

background: white;

border-radius: 15px;

overflow: hidden;

box-shadow: 0 20px 40px rgba(0,0,0,0.1);

margin-bottom: 30px;

}

#hikingMap {

height: 600px;

width: 100%;

border-radius: 15px;

}

.controls {

background: white;

padding: 20px;

border-radius: 15px;

margin-bottom: 20px;

box-shadow: 0 10px 30px rgba(0,0,0,0.1);

}

.control-group {

display: flex;

gap: 15px;

flex-wrap: wrap;

justify-content: center;

}

.btn {

padding: 12px 24px;

border: none;

border-radius: 8px;

background: linear-gradient(135deg, #667eea, #764ba2);

color: white;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

box-shadow: 0 4px 15px rgba(102, 126, 234, 0.3);

}

.btn:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(102, 126, 234, 0.4);

}

.btn-outline {

background: transparent;

border: 2px solid #667eea;

color: #667eea;

}

.features {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 20px;

margin-top: 30px;

}

.feature-card {

background: white;

padding: 25px;

border-radius: 15px;

text-align: center;

box-shadow: 0 10px 30px rgba(0,0,0,0.1);

transition: transform 0.3s ease;

}

.feature-card:hover {

transform: translateY(-5px);

}

.feature-icon {

font-size: 2.5rem;

margin-bottom: 15px;

}

.feature-card h3 {

color: #333;

margin-bottom: 10px;

}

.feature-card p {

color: #666;

line-height: 1.6;

}

.footer {

text-align: center;

color: white;

padding: 20px;

margin-top: 40px;

opacity: 0.8;

}

/\* Responsive design \*/

@media (max-width: 768px) {

#hikingMap {

height: 400px;

}

.header h1 {

font-size: 2rem;

}

.control-group {

flex-direction: column;

}

.btn {

width: 100%;

}

}

</style>

</head>

<body>

<div class="container">

<div class="header">

<h1>🚶 Hiking Trails Explorer</h1>

<p>Discover and plan your next adventure with Speed-Proof</p>

</div>

<div class="controls">

<div class="control-group">

<button class="btn" onclick="locateUser()">📍 Find My Location</button>

<button class="btn btn-outline" onclick="showPopularTrails()">🏞️ Popular Trails</button>

<button class="btn btn-outline" onclick="showEmergencyInfo()">🆘 Emergency Info</button>

</div>

</div>

<div class="map-container">

<div id="hikingMap"></div>

</div>

<div class="features">

<div class="feature-card">

<div class="feature-icon">🗺️</div>

<h3>Interactive Trails</h3>

<p>Explore detailed hiking routes with elevation profiles and difficulty ratings</p>

</div>

<div class="feature-card">

<div class="feature-icon">📱</div>

<h3>Mobile Ready</h3>

<p>Access trail maps offline with the Speed-Proof app during your hikes</p>

</div>

<div class="feature-card">

<div class="feature-icon">🆘</div>

<h3>Emergency Ready</h3>

<p>Generate precise GPS coordinates for emergency situations with one tap</p>

</div>

</div>

</div>

<div class="footer">

<p>&copy; 2024 Speed-Proof.com | Your trusted hiking companion</p>

</div>

<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

<script>

// Initialize the map

const map = L.map('hikingMap').setView([40.7128, -74.0060], 10); // Default to NYC

// Add OpenStreetMap tiles (FREE!)

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

attribution: '© OpenStreetMap contributors',

maxZoom: 18

}).addTo(map);

// Add terrain layer for better hiking context

L.tileLayer('https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png', {

attribution: '© OpenTopoMap contributors',

maxZoom: 17,

opacity: 0.7

}).addTo(map);

// Sample hiking trails data (you can replace with real data)

const sampleTrails = [

{

name: "Appalachian Trail Section",

coords: [[40.7128, -74.0060], [40.8138, -74.0760], [40.9148, -74.1460]],

difficulty: "moderate",

length: "12.5 km"

},

{

name: "River Valley Loop",

coords: [[40.7528, -74.0560], [40.7928, -74.0160], [40.7728, -73.9760], [40.7528, -74.0560]],

difficulty: "easy",

length: "8.2 km"

},

{

name: "Mountain Ridge Trail",

coords: [[40.8128, -74.1060], [40.8528, -74.0860], [40.8928, -74.0660], [40.8728, -74.0260]],

difficulty: "hard",

length: "15.7 km"

}

];

// Add sample trails to map

sampleTrails.forEach(trail => {

const polyline = L.polyline(trail.coords, {

color: getColorByDifficulty(trail.difficulty),

weight: 6,

opacity: 0.7

}).addTo(map);

// Add popup with trail info

polyline.bindPopup(`

<div style="padding: 10px;">

<h3 style="margin: 0 0 10px 0; color: #333;">${trail.name}</h3>

<p style="margin: 5px 0;"><strong>Difficulty:</strong>

<span style="color: ${getColorByDifficulty(trail.difficulty)}; font-weight: bold;">

${trail.difficulty.toUpperCase()}

</span>

</p>

<p style="margin: 5px 0;"><strong>Length:</strong> ${trail.length}</p>

<button onclick="simulateEmergency('${trail.name}')"

style="background: #ff4757; color: white; border: none; padding: 8px 15px; border-radius: 5px; cursor: pointer; margin-top: 10px;">

🆘 Simulate Emergency GPS

</button>

</div>

`);

});

// Add some sample markers for points of interest

const pointsOfInterest = [

{lat: 40.7828, lng: -74.0660, name: "Trailhead Parking", type: "parking"},

{lat: 40.8128, lng: -74.0360, name: "Waterfall View", type: "viewpoint"},

{lat: 40.8328, lng: -74.0160, name: "Emergency Shelter", type: "shelter"}

];

pointsOfInterest.forEach(poi => {

const icon = L.divIcon({

html: getIconForType(poi.type),

className: 'custom-div-icon',

iconSize: [30, 30]

});

L.marker([poi.lat, poi.lng], {icon: icon})

.addTo(map)

.bindPopup(`<strong>${poi.name}</strong><br>Type: ${poi.type}`);

});

// Function to locate user

function locateUser() {

if (navigator.geolocation) {

navigator.geolocation.getCurrentPosition(

position => {

const userLat = position.coords.latitude;

const userLng = position.coords.longitude;

map.setView([userLat, userLng], 13);

// Add user location marker

L.marker([userLat, userLng])

.addTo(map)

.bindPopup("<strong>Your Current Location</strong>")

.openPopup();

alert(`📍 Found you! Coordinates: ${userLat.toFixed(4)}, ${userLng.toFixed(4)}`);

},

error => {

alert("Unable to get your location. Please ensure location services are enabled.");

}

);

} else {

alert("Geolocation is not supported by this browser.");

}

}

// Function to show popular trails

function showPopularTrails() {

const bounds = new L.LatLngBounds();

sampleTrails.forEach(trail => {

trail.coords.forEach(coord => {

bounds.extend(coord);

});

});

map.fitBounds(bounds, {padding: [20, 20]});

}

// Function to show emergency info

function showEmergencyInfo() {

alert(`🆘 Speed-Proof Emergency Features:\n\n• One-tap GPS coordinate generation\n• Offline location sharing\n• Emergency contact integration\n• Battery-efficient tracking\n\nTest these features in the Speed-Proof app!`);

}

// Simulate emergency GPS feature

function simulateEmergency(trailName) {

const center = map.getCenter();

const coords = `${center.lat.toFixed(6)}, ${center.lng.toFixed(6)}`;

alert(`🚨 EMERGENCY GPS SIMULATION\n\nTrail: ${trailName}\nCoordinates: ${coords}\nTimestamp: ${new Date().toLocaleString()}\n\nThis would generate emergency coordinates in the Speed-Proof app!`);

}

// Helper function to get color by difficulty

function getColorByDifficulty(difficulty) {

switch(difficulty) {

case 'easy': return '#2ecc71';

case 'moderate': return '#f39c12';

case 'hard': return '#e74c3c';

default: return '#3498db';

}

}

// Helper function to get icon HTML for POI types

function getIconForType(type) {

const icons = {

parking: '🅿️',

viewpoint: '🏞️',

shelter: '🏠'

};

return `<div style="font-size: 20px;">${icons[type] || '📍'}</div>`;

}

// Add scale control

L.control.scale({imperial: true, metric: true}).addTo(map);

</script>

</body>

</html>