```html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>496 Challenge 2026 - Group Live</title>

    <script src="https://cdn.tailwindcss.com"></script>
    <script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>
    <script
src="https://cdn.jsdelivr.net/npm/gpxparser@3.0.8/dist/GPXParser.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/canvas-confetti@1.6.0/dist/confetti.browser.min.js"></script
>
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
rel="stylesheet">

    <style>
        #map { height: 200px; border-radius: 1.5rem; }
        .medal-unlocked {
            background: linear-gradient(90deg, #ffd700, #fff5b1, #ffd700);
            background-size: 200%;
            animation: shine 3s linear infinite;
        }
        @keyframes shine { 0% { background-position: -200%; } 100% { background-position:
200%; } }
        .progress-bar-smooth { transition: width 1.5s cubic-bezier(0.34, 1.56, 0.64, 1); }
    </style>
</head>
<body class="bg-[#0a0a0a] text-white font-sans selection:bg-lime-500 selection:text-black">

    <nav class="p-4 bg-gray-900/50 backdrop-blur-md border-b border-gray-800 sticky top-0
z-50">
        <div class="max-w-6xl mx-auto flex justify-between items-center">
            <h1 class="text-xl font-black text-lime-400 italic tracking-tighter">496 CHALLENGE
2026</h1>
            <div class="flex gap-2">
                <input type="text" id="pseudo" placeholder="Pseudo" class="bg-gray-800 border
border-gray-700 rounded-lg px-3 py-1 text-sm outline-none focus:border-lime-500 w-32">
                <button onclick="savePseudo()" class="bg-lime-600 hover:bg-lime-500 text-black
font-bold text-xs px-3 rounded-lg transition-colors">OK</button>
            </div>
        </div>
    </nav>

    <main class="max-w-6xl mx-auto p-4 grid grid-cols-1 lg:grid-cols-12 gap-6 mt-4">
```

```html
    <aside class="lg:col-span-3 order-2 lg:order-1">
        <div class="bg-gray-900/40 p-5 rounded-[2rem] border border-gray-800 shadow-xl">
            <h3 class="text-lime-400 font-black uppercase text-[10px] mb-4 tracking-[0.2em]
flex items-center gap-2">
                <i class="fas fa-trophy"></i> Leaderboard Groupe
            </h3>
            <div id="leaderboard" class="space-y-4">
                <p class="text-gray-600 text-xs animate-pulse italic text-center
py-4">Connexion à Supabase...</p>
            </div>
        </div>
    </aside>

    <div class="lg:col-span-6 order-1 lg:order-2 space-y-6">
        <div class="bg-gray-900 p-8 rounded-[2.5rem] border border-gray-800 shadow-2xl
relative overflow-hidden">
            <div class="flex justify-between items-end mb-4 relative z-10">
                <div>
                    <h2 class="text-6xl font-black italic tracking-tighter"
id="total-display">0.0</h2>
                    <p class="text-gray-500 font-bold uppercase text-[10px] tracking-widest
mt-1">Kilomètres cumulés / 496</p>
                </div>
                <span class="text-lime-400 font-black text-3xl" id="percent-display">0%</span>
            </div>
            <div class="w-full bg-gray-800 h-4 rounded-full overflow-hidden relative z-10">
                <div id="progress-fill" class="progress-bar-smooth h-full bg-lime-500
w-0"></div>
            </div>
        </div>

        <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
            <button onclick="connectStrava()" class="bg-[#FC4C02] hover:bg-[#ff5712] p-6
rounded-[2rem] flex flex-col items-center justify-center transition-all group active:scale-95
shadow-lg shadow-orange-950/20">
                <svg class="w-8 h-8 mb-2 fill-white" viewBox="0 0 24 24"><path d="M15.387
17.944l-2.089-4.116h-3.065L15.387 24l5.15-10.172h-3.066l-2.084 4.116zM10.257
7.833l-3.064 6.134h4.118l1.018-2.043 1.022 2.043h4.112l-7.206-14z"/></svg>
                <span class="font-black uppercase text-[10px] tracking-widest text-white">Sync
Strava</span>
            </button>
            <button onclick="document.getElementById('gpx-input').click()"
class="bg-gray-800 hover:bg-gray-700 p-6 rounded-[2rem] flex flex-col items-center
justify-center transition-all border border-gray-700 active:scale-95">
                <i class="fas fa-file-upload text-2xl text-lime-400 mb-2"></i>
                <span class="font-black uppercase text-[10px] tracking-widest
text-gray-400">Importer GPX</span>
```

```html
            <input type="file" id="gpx-input" class="hidden" accept=".gpx"
onchange="handleGPX(this)">
        </button>
      </div>

      <div id="map-container" class="hidden bg-gray-900 p-2 rounded-[2rem] border
border-gray-800 shadow-lg overflow-hidden">
        <div id="map"></div>
      </div>

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

    <aside class="lg:col-span-3 order-3">
      <div id="medal-card" class="bg-gray-900 p-8 rounded-[2rem] border border-gray-800
text-center transition-all duration-1000 grayscale opacity-20 flex flex-col items-center">
        <div id="medal-icon" class="w-24 h-24 bg-gray-800 rounded-full mb-4 flex
items-center justify-center border-4 border-gray-700 relative">
          <i class="fas fa-medal text-4xl text-gray-500"></i>
        </div>
        <h3 class="font-black italic uppercase text-xs tracking-widest">496 Finisher</h3>
        <p class="text-[10px] text-gray-500 mt-2 uppercase" id="medal-text">Challenge
Verrouillé</p>
      </div>
    </aside>
  </main>

  <script>
    // --- CONFIGURATION (À REMPLIR) ---
    const SUPABASE_URL = 'VOTRE_URL_SUPABASE';
    const SUPABASE_KEY = 'VOTRE_CLE_ANON_SUPABASE';
    const STRAVA_CLIENT_ID = 'VOTRE_STRAVA_ID';
    const STRAVA_CLIENT_SECRET = 'VOTRE_STRAVA_SECRET';

    const supabase = exports.supabase.createClient(SUPABASE_URL,
SUPABASE_KEY);
    const REDIRECT_URI = window.location.origin + window.location.pathname;

    let myPseudo = localStorage.getItem('496_pseudo') || "";
    let userData = JSON.parse(localStorage.getItem('496_data')) || { totalKm: 0, activities: []
};
    let map = null, polyline = null;

    // Initialisation
    document.getElementById('pseudo').value = myPseudo;
    updateUI();
    refreshLeaderboard();
```

```
checkStravaCallback();

// --- LOGIQUE UTILISATEUR ---

function savePseudo() {
    myPseudo = document.getElementById('pseudo').value.trim();
    if(myPseudo) {
        localStorage.setItem('496_pseudo', myPseudo);
        syncWithCloud();
        alert("Pseudo enregistré !");
    }
}

function handleGPX(input) {
    if(!myPseudo) return alert("Choisis un pseudo avant !");
    const file = input.files[0];
    const reader = new FileReader();
    reader.onload = (e) => {
        const gpx = new gpxParser();
        gpx.parse(e.target.result);
        const dist = gpx.tracks[0].distance.total / 1000;
        addActivity(dist, file.name);
        drawMap(gpx.tracks[0].points.map(p => [p.lat, p.lon]));
    };
    reader.readAsText(file);
}

async function connectStrava() {
    if(!myPseudo) return alert("Pseudo requis !");
    window.location.href =
`https://www.strava.com/oauth/authorize?client_id=${STRAVA_CLIENT_ID}&redirect_uri=${
REDIRECT_URI}&response_type=code&scope=activity:read_all`;
}

async function checkStravaCallback() {
    const code = new URLSearchParams(window.location.search).get('code');
    if (code) {
        const tokenResp = await fetch('https://www.strava.com/oauth/token', {
            method: 'POST',
            headers: { 'Content-Type': 'application/json' },
            body: JSON.stringify({ client_id: STRAVA_CLIENT_ID, client_secret:
STRAVA_CLIENT_SECRET, code: code, grant_type: 'authorization_code' })
        });
        const tokenData = await tokenResp.json();

        const afterDate = Math.floor(new Date('2026-01-01').getTime() / 1000);
        const actResp = await
fetch(`https://www.strava.com/api/v3/athlete/activities?after=${afterDate}`, {
```

```javascript
        headers: { 'Authorization': `Bearer ${tokenData.access_token}` }
    });
    const activities = await actResp.json();

    activities.forEach(a => {
        if((a.type === 'Run' || a.sport_type === 'Run') && !userData.activities.find(ex =>
ex.stravaId === a.id)) {
            addActivity(a.distance / 1000, a.name, a.id);
        }
    });
    window.history.pushState({}, document.title, REDIRECT_URI);
    }
}

// --- GESTION DES ACTIVITÉS ---

function addActivity(km, name, stravaId = null) {
    const act = { id: Date.now(), stravaId, km: parseFloat(km.toFixed(2)), name, date:
new Date().toLocaleDateString('fr-FR', {day:'numeric', month:'short'}) };
    userData.activities.unshift(act);
    saveAndRefresh();
}

function deleteActivity(id) {
    if(confirm("Supprimer cette activité ?")) {
        userData.activities = userData.activities.filter(a => a.id !== id);
        saveAndRefresh();
    }
}

function saveAndRefresh() {
    userData.totalKm = userData.activities.reduce((sum, a) => sum + a.km, 0);
    localStorage.setItem('496_data', JSON.stringify(userData));
    updateUI();
    syncWithCloud();
}

// --- INTERFACE ---

function updateUI() {
    document.getElementById('total-display').innerText = userData.totalKm.toFixed(1);
    const percent = Math.min((userData.totalKm / 496) * 100, 100);
    document.getElementById('progress-fill').style.width = percent + "%";
    document.getElementById('percent-display').innerText = Math.round(percent) + "%";

    // Liste
    const list = document.getElementById('activities-list');
    list.innerHTML = userData.activities.map(a => `
```

```html
            <div class="bg-gray-900/60 p-4 rounded-2xl border border-gray-800 flex justify-between items-center group">
                <div>
                    <p class="text-[10px] text-gray-500 font-bold uppercase">${a.date}</p>
                    <p class="font-bold text-sm truncate w-40">${a.name}</p>
                </div>
                <div class="flex items-center gap-4">
                    <span class="text-lime-400 font-black text-lg">+${a.km}</span>
                    <button onclick="deleteActivity(${a.id})" class="opacity-0 group-hover:opacity-100 text-gray-600 hover:text-red-500 transition-all"><i class="fas fa-trash"></i></button>
                </div>
            </div>
        `).join('');

        // Médaille
        if(userData.totalKm >= 496) {
            document.getElementById('medal-card').classList.remove('grayscale', 'opacity-20');
            document.getElementById('medal-card').classList.add('medal-unlocked', 'border-yellow-500');
            document.getElementById('medal-text').innerText = "Vainqueur du 496 !";
            confetti({ particleCount: 150, spread: 70, origin: { y: 0.6 }, colors:['#84cc16','#ffd700','#ffffff'] });
        }
    }

    async function syncWithCloud() {
        if(myPseudo) {
            await supabase.from('scores').upsert({ pseudo: myPseudo, distance: userData.totalKm });
            refreshLeaderboard();
        }
    }

    async function refreshLeaderboard() {
        const { data } = await supabase.from('scores').select('*').order('distance', { ascending: false });
        if(data) {
            document.getElementById('leaderboard').innerHTML = data.map((u, i) => `
                <div class="flex justify-between items-center ${u.pseudo === myPseudo ? 'bg-lime-900/10 p-2 rounded-xl border border-lime-500/20' : ''}">
                    <div class="flex items-center gap-3">
                        <span class="text-[10px] font-bold text-gray-600">${i+1}.</span>
                        <span class="text-xs font-bold ${u.pseudo === myPseudo ? 'text-lime-400' : 'text-gray-300'}">${u.pseudo}</span>
                    </div>
                    <span class="text-xs font-black">${u.distance.toFixed(1)} km</span>
```

```
              </div>
          `).join('');
        }
      }

      function drawMap(points) {
          document.getElementById('map-container').classList.remove('hidden');
          if(!map) {
              map = L.map('map', {zoomControl: false}).setView(points[0], 13);

L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png').addTo(map);
          }
          if(polyline) map.removeLayer(polyline);
          polyline = L.polyline(points, {color: '#84cc16', weight: 4, opacity: 0.8}).addTo(map);
          map.fitBounds(polyline.getBounds());
      }
    </script>
</body>
</html>
```