```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Herbicide Trial Manager - Organization</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.
js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/pptxgenjs/3.12.0/pptxgen.m
in.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/cropperjs/1.5.12/cropper.m
in.js"></script>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/cropperjs/1.5.12/cropper.
min.css">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;7
00&display=swap');
        body { font-family: 'Inter', sans-serif; background-color:
#f3f4f6; }
        .container-app { background-color: #ffffff; border-radius:
1rem; box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px
rgba(0, 0, 0, 0.05); max-width: 1200px; margin: 2rem auto; min-height:
80vh; padding: 2rem; display: flex; flex-direction: column; }
        .card { border-radius: 0.75rem; background-color: #f9fafb;
padding: 1.5rem; margin-top: 1.5rem; box-shadow: 0 1px 3px 0 rgba(0,
0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06); }
        .btn-primary { background-color: #3b82f6; color: white;
padding: 0.75rem 1.5rem; border-radius: 0.5rem; font-weight: 600;
transition: background-color 0.2s ease-in-out; cursor: pointer; }
        .btn-primary:hover { background-color: #2563eb; }
        .btn-secondary { background-color: #e5e7eb; color: #4b5563;
padding: 0.75rem 1.5rem; border-radius: 0.5rem; font-weight: 600;
transition: background-color 0.2s ease-in-out; cursor: pointer; }
        .btn-secondary:hover { background-color: #d1d5db; }
        .message-box { position: fixed; bottom: 2rem; left: 50%;
transform: translateX(-50%); background-color: #10b981; color: white;
padding: 1rem 2rem; border-radius: 0.75rem; box-shadow: 0 4px 6px -1px
rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06); opacity: 0;
transition: opacity 0.3s ease-in-out; z-index: 101; pointer-events:
none; }
        .message-box.show { opacity: 1; pointer-events: auto; }
        .tab-active { background-color: #dbeafe !important; color:
#1e40af !important; border-bottom: 2px solid #2563eb !important; }
        .loading-overlay { position: fixed; top: 0; left: 0; width:
100%; height: 100%; background-color: rgba(255, 255, 255, 0.7);
display: flex; justify-content: center; align-items: center; z-index:
200; }
        .loading-spinner { border: 4px solid rgba(0, 0, 0, 0.1);
width: 36px; height: 36px; border-radius: 50%; border-left-color:
#3b82f6; animation: spin 1s ease infinite; }
        @keyframes spin { 0% { transform: rotate(0deg); } 100% {
transform: rotate(360deg); } }
    </style>
</head>
<body class="bg-gray-100 min-h-screen p-4">
    <div id="loading-overlay" class="loading-overlay hidden"><div
class="loading-spinner"></div></div>

    <!-- Auth Section -->
    <div id="auth-container" class="container-app">
        <div class="max-w-md mx-auto w-full">
            <h2 id="auth-title" class="text-3xl font-bold text-center
text-gray-800 mb-6">Login</h2>
            <div id="auth-error" class="bg-red-100 border
border-red-400 text-red-700 px-4 py-3 rounded relative mb-4 hidden"
role="alert"></div>
            <form id="auth-form" class="space-y-4">
                <input type="email" id="auth-email"
placeholder="Email" required class="w-full p-3 rounded-md
border-gray-300 shadow-sm border focus:border-indigo-300 focus:ring
focus:ring-indigo-200 focus:ring-opacity-50">
                <input type="password" id="auth-password"
placeholder="Password" required class="w-full p-3 rounded-md
border-gray-300 shadow-sm border focus:border-indigo-300 focus:ring
focus:ring-indigo-200 focus:ring-opacity-50">
                <button type="submit" id="auth-submit-btn"
class="btn-primary w-full py-3">Login</button>
            </form>
            <p class="text-center mt-4">
                <a href="#" id="auth-toggle-link" class="text-blue-600
hover:underline">Need an account? Sign Up</a>
            </p>
        </div>
    </div>

    <!-- Main App Container -->
    <div id="app" class="container-app hidden">
        <div class="flex flex-col md:flex-row justify-between
items-start md:items-center mb-6">
            <h1 class="text-4xl font-bold text-gray-800">Herbicide
Trial Manager</h1>
            <div class="flex items-center space-x-4">
                <span id="user-email" class="text-gray-600"></span>
                <button id="logout-btn"
class="btn-secondary">Logout</button>
            </div>
        </div>

        <!-- Organization Setup Section -->
        <div id="org-setup-container" class="text-center p-8
bg-gray-50 rounded-lg hidden">
            <h2 class="text-2xl font-bold mb-4">Welcome!</h2>
            <p class="mb-6 text-gray-600">To start collaborating,
create an organization for your team.</p>
            <div class="max-w-sm mx-auto">
                <input type="text" id="org-name-input"
placeholder="Enter Your Organization's Name" class="w-full p-3
rounded-md border-gray-300 shadow-sm border mb-4">
                <button id="create-org-btn" class="btn-primary w-full
py-3">Create Organization</button>
            </div>
             <p class="mt-6 text-gray-500 text-sm">If you were invited
to an organization, please sign up using the link from your email.</p>
        </div>

        <!-- Main Content (Tabs) -->
        <div id="main-content" class="flex-1 overflow-y-auto hidden">
            <div class="flex space-x-2 border-b-2 border-gray-200
mb-6">
                <button id="tab-dashboard" class="tab-btn py-2
px-4">Dashboard</button>
                <button id="tab-formulations" class="tab-btn py-2
px-4">Formulations</button>
                <button id="tab-trials" class="tab-btn py-2
px-4">Trials</button>
                <button id="tab-organization" class="tab-btn py-2
px-4">Organization</button>
            </div>

            <div id="section-dashboard" class="tab-section
hidden">...</div>
            <div id="section-formulations" class="tab-section
hidden">...</div>
            <div id="section-trials" class="tab-section
```

```
hidden">...</div>

            <div id="section-organization" class="tab-section hidden">
                <h2 class="text-2xl font-semibold text-gray-800
mb-4">Manage Organization</h2>
                <div class="card bg-gray-50">
                    <h3 class="text-xl font-bold text-gray-700
mb-2">Organization: <span id="org-name-display"
class="text-blue-600"></span></h3>
                    <div class="mt-6">
                        <h4 class="font-semibold text-gray-700
mb-2">Invite New Member</h4>
                        <p class="text-sm text-gray-500 mb-2">Enter
the email of a colleague to invite them to your organization.</p>
                        <div class="flex space-x-2">
                            <input type="email"
id="invite-email-input" placeholder="colleague@example.com"
class="flex-grow p-2 rounded-md border-gray-300 shadow-sm border">
                            <button id="invite-btn"
class="btn-primary">Send Invite</button>
                        </div>
                    </div>
                    <div class="mt-8">
                        <h4 class="font-semibold text-gray-700
mb-2">Team Members</h4>
                        <ul id="members-list" class="space-y-2"></ul>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div id="message-box" class="message-box hidden"></div>
    <!-- Modals will go here, similar to your original file -->

    <script type="module">
        // --- SUPABASE INITIALIZATION ---
        const SUPABASE_URL =
'https://guloqbnlxinbfdatxjnp.supabase.co';
        const SUPABASE_ANON_KEY =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6
Imd1bG9xYm5seGluYmZkYXR4am5wIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTkzMzkyMj
EsImV4cCI6MjA3NDkxNTIyMX0.4H_OVnlFjO1Pd3gvHruJeQ4efA8ljiZeQuT321eQS80'
;
        const { createClient } = supabase;
        const _supabase = createClient(SUPABASE_URL,
SUPABASE_ANON_KEY);
        window.supabase = _supabase;
        // --- DOM ELEMENTS & GLOBAL STATE ---
        const authContainer =
document.getElementById('auth-container');
        const appContainer = document.getElementById('app');
        const mainContent = document.getElementById('main-content');
        const orgSetupContainer =
document.getElementById('org-setup-container');
        const loadingOverlay =
document.getElementById('loading-overlay');
        const messageBox = document.getElementById('message-box');

        let userProfile = null;
        let isLogin = true;

        // --- AUTH & USER PROFILE MANAGEMENT ---
        _supabase.auth.onAuthStateChange(async (_event, session) => {
            loadingOverlay.classList.remove('hidden');
            if (session) {
                await fetchUserProfile(session.user.id);
                authContainer.classList.add('hidden');
                appContainer.classList.remove('hidden');
                document.getElementById('user-email').innerText =
session.user.email;

                if (userProfile && userProfile.organization_id) {
                    mainContent.classList.remove('hidden');
                    orgSetupContainer.classList.add('hidden');
                    await loadOrganizationData();
                } else {
                    mainContent.classList.add('hidden');
                    orgSetupContainer.classList.remove('hidden');
                }
            } else {
                userProfile = null;
                authContainer.classList.remove('hidden');
                appContainer.classList.add('hidden');
            }
            loadingOverlay.classList.add('hidden');
        });

        async function fetchUserProfile(userId) {
            const { data, error } = await
_supabase.from('profiles').select('*').eq('id', userId).single();
            if (error) {
                console.error("Error fetching profile:", error);
                showMessage("Could not retrieve user profile.",
"#ef4444");
```

```
            } else {
                userProfile = data;
            }
        }

        const authForm = document.getElementById('auth-form');
        authForm.addEventListener('submit', async (e) => {
            e.preventDefault();
            loadingOverlay.classList.remove('hidden');
            const email = document.getElementById('auth-email').value;
            const password =
document.getElementById('auth-password').value;
            try {
                let response;
                if (isLogin) {
                    response = await
_supabase.auth.signInWithPassword({ email, password });
                } else {
                    response = await _supabase.auth.signUp({ email,
password });
                    if (!response.error) {
                        showMessage('Success! Please check your email
for a verification link.', '#3b82f6');
                    }
                }
                if (response.error) throw response.error;
            } catch (error) {
                showMessage(error.message, '#ef4444');
            } finally {
                loadingOverlay.classList.add('hidden');
            }
        });


document.getElementById('logout-btn').addEventListener('click', () =>
_supabase.auth.signOut());

document.getElementById('auth-toggle-link').addEventListener('click',
(e) => {
            e.preventDefault();
            isLogin = !isLogin;
            document.getElementById('auth-title').innerText = isLogin
? 'Login' : 'Sign Up';
            document.getElementById('auth-submit-btn').innerText =
isLogin ? 'Login' : 'Sign Up';
            e.target.innerText = isLogin ? 'Need an account? Sign Up'
: 'Have an account? Login';
        });
        // --- ORGANIZATION MANAGEMENT ---

document.getElementById('create-org-btn').addEventListener('click',
async () => {
            loadingOverlay.classList.remove('hidden');
            const orgName =
document.getElementById('org-name-input').value.trim();
            if (!orgName) {
                loadingOverlay.classList.add('hidden');
                return showMessage('Please enter an organization
name.', '#ef4444');
            }

            // 1. Create the organization
            const { data: orgData, error: orgError } = await
_supabase.from('organizations').insert({ name: orgName
}).select().single();
            if (orgError) {
                loadingOverlay.classList.add('hidden');
                return showMessage(orgError.message, '#ef4444');
            }

            // 2. Link the current user to the new organization
            const { error: profileError } = await
_supabase.from('profiles').update({ organization_id: orgData.id
}).eq('id', userProfile.id);
            if (profileError) {
                 loadingOverlay.classList.add('hidden');
                return showMessage(profileError.message, '#ef4444');
            }

            showMessage("Organization created successfully!",
"#10b981");

            // Refresh UI
            await fetchUserProfile(userProfile.id);
            mainContent.classList.remove('hidden');
            orgSetupContainer.classList.add('hidden');
            await loadOrganizationData();
            loadingOverlay.classList.add('hidden');
        });


document.getElementById('invite-btn').addEventListener('click', async
() => {
            const emailToInvite =
document.getElementById('invite-email-input').value.trim();
```

```
            if(!emailToInvite) return showMessage('Please enter an
email to invite.', '#ef4444');

            const { data, error } = await
_supabase.auth.admin.inviteUserByEmail(emailToInvite);

            if (error) {
                showMessage(`Error sending invite: ${error.message}.
Note: This feature requires your Supabase instance to have an email
provider configured.`, '#ef4444');
            } else {
                showMessage(`Invite sent to ${emailToInvite}!`,
'#10b981');
                document.getElementById('invite-email-input').value =
'';
            }
        });

        async function loadOrganizationData() {
            if (!userProfile || !userProfile.organization_id) return;

            // Fetch org name
            const { data: org, error: orgError } = await
_supabase.from('organizations').select('name').eq('id',
userProfile.organization_id).single();
            if (org)
document.getElementById('org-name-display').innerText = org.name;

            // Fetch members
            const { data: members, error: membersError } = await
_supabase.from('profiles').select('id').eq('organization_id',
userProfile.organization_id);
            if(members) {
                 const membersList =
document.getElementById('members-list');
                 membersList.innerHTML = '';
                 // This is a simplified display. We can't get emails
directly for security reasons
                 // without more complex table joins and policies.
                 members.forEach(member => {
                     const li = document.createElement('li');
                     li.className = "bg-white p-2 rounded-md
shadow-sm";
                     // We show the User ID as a placeholder for the
email
                     li.textContent =
```