**INTIAL LCM FRONT END CODE  
  
INDEX.HTML**<!DOCTYPE html>

<html lang="en">

<script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2.0.0/dist/umd/supabase.min.js"></script>

<head>

<meta charset="UTF-8">

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

<title>LCM DASHBOARD</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Top Bar with Title at the very top -->

<div class="top-bar">

<h1>LCM DASHBOARD</h1>

</div>

<!-- User Profile Section -->

<div class="user-profile">

<h2>User Profile</h2>

<p><strong>Name:</strong> Eric Muthomi Barimbui</p>

<p><strong>Email:</strong> ebarimbui@gmail.com</p>

<!-- MY COMMUNITIES Section -->

<div class="communities-title">

<h3>MY COMMUNITIES</h3>

</div>

<div class="community-row">

<!-- Static Communities Here (like Home, School, Work, etc.) -->

<div class="community-column">

<h4>Home</h4>

<p>Number of Verifiers: <span id="homeVerifiers">5</span></p>

<button class="dropdown-btn">View Verifiers</button>

<div class="dropdown-content">

<p>John</p>

<p>Jane</p>

<p>Alice</p>

</div>

<button class="add-btn">+</button>

<button class="remove-btn">-</button>

</div>

<div class="community-column">

<h4>School</h4>

<p>Number of Verifiers: <span id="schoolVerifiers">3</span></p>

<button class="dropdown-btn">View Verifiers</button>

<div class="dropdown-content">

<p>Mike</p>

<p>Sarah</p>

</div>

<button class="add-btn">+</button>

<button class="remove-btn">-</button>

</div>

<div class="community-column">

<h4>Church</h4>

<p>Number of Verifiers: <span id="churchVerifiers">4</span></p>

<button class="dropdown-btn">View Verifiers</button>

<div class="dropdown-content">

<p>Anna</p>

<p>Luke</p>

<p>Paul</p>

</div>

<button class="add-btn">+</button>

<button class="remove-btn">-</button>

</div>

<div class="community-column">

<h4>Team</h4>

<p>Number of Verifiers: <span id="teamVerifiers">2</span></p>

<button class="dropdown-btn">View Verifiers</button>

<div class="dropdown-content">

<p>Tom</p>

<p>Alice</p>

</div>

<button class="add-btn">+</button>

<button class="remove-btn">-</button>

</div>

</div>

<!-- "What are you up to today?" Section -->

<div class="question-title">

<h3>What are you up to today?</h3>

</div>

<!-- User Input Area with White Border -->

<textarea class="task-input" placeholder="Write down what you're up to today..."></textarea>

<!-- Dropdown Suggestions (suggested as the user types) -->

<div class="suggestions-dropdown">

<ul>

<li>Task Suggestion 1</li>

<li>Task Suggestion 2</li>

<li>Task Suggestion 3</li>

</ul>

</div>

<!-- Row for Category, Verifier, and Submit Buttons -->

<div class="action-buttons">

<!-- Choose Category Button -->

<button class="choose-category-btn" id="chooseCategoryBtn">Choose Category</button>

<div class="category-dropdown-content" id="categoryDropdown">

<!-- Categories will be populated here -->

</div>

<!-- Choose Verifier Button -->

<button class="choose-verifier-btn" id="chooseVerifierBtn">Choose Verifier</button>

<div class="verifier-dropdown-content" id="verifierDropdown">

<!-- Verifiers will be populated here -->

</div>

<!-- Submit Button -->

<button class="submit-btn">Submit</button>

</div>

<!-- Tasks Logged Table -->

<div class="tasks-logged-table">

<h3>TASKS LOGGED 7TH AUGUST 2025</h3>

<table>

<thead>

<tr>

<th>Task</th>

<th>Category</th>

<th>Verifiers</th>

<th>Status</th>

<th>Credits</th>

</tr>

</thead>

<tbody>

<tr>

<td>Made My Bed</td>

<td>Home</td>

<td>🟩🟩⬜⬜⬜</td>

<td>Pending Verification</td>

<td>+1</td>

</tr>

<!-- Other rows -->

</tbody>

</table>

</div>

<!-- Special Credits Table -->

<div class="special-credits-table">

<h3>SPECIAL CREDITS</h3>

<table>

<thead>

<tr>

<th>Credit Type</th>

<th>Amount</th>

<th>Status</th>

</tr>

</thead>

<tbody>

<tr>

<td>Policing Credit</td>

<td>5</td>

<td>Verified</td>

</tr>

<!-- Other rows -->

</tbody>

</table>

</div>

<!-- Daily Stats Table -->

<div class="daily-stats-table">

<h3>DAILY STATS</h3>

<table>

<thead>

<tr>

<th>Date</th>

<th>Total Tasks Logged</th>

<th>Verified Tasks</th>

<th>Policing Credits</th>

</tr>

</thead>

<tbody>

<tr>

<td>7th August 2025</td>

<td>3</td>

<td>2</td>

<td>2</td>

</tr>

<!-- Other rows -->

</tbody>

</table>

</div>

<!-- Monthly Stats Table -->

<div class="monthly-stats-table">

<h3>MONTHLY STATS</h3>

<table>

<thead>

<tr>

<th>Month</th>

<th>Total Tasks Logged</th>

<th>Verified Tasks</th>

<th>Total Credits</th>

</tr>

</thead>

<tbody>

<tr>

<td>August 2025</td>

<td>12</td>

<td>10</td>

<td>30</td>

</tr>

<!-- Other rows -->

</tbody>

</table>

</div>

<!-- Yearly Stats Table -->

<div class="yearly-stats-table">

<h3>YEARLY STATS</h3>

<table>

<thead>

<tr>

<th>Year</th>

<th>Total Tasks Logged</th>

<th>Verified Tasks</th>

<th>Total Credits</th>

</tr>

</thead>

<tbody>

<tr>

<td>2025</td>

<td>100</td>

<td>90</td>

<td>200</td>

</tr>

<!-- Other rows -->

</tbody>

</table>

</div>

</div>

<script src="app.js"></script>

</body>

</html>  
 **STYLES.CSS**/\* Background and general settings \*/

body {

background-color: #121212; /\* Dark background \*/

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

display: flex;

flex-direction: column; /\* Arrange content vertically \*/

justify-content: flex-start; /\* Align items at the top \*/

align-items: center; /\* Center the content horizontally \*/

height: 100vh;

margin: 0;

}

/\* Top Bar \*/

.top-bar {

background-color: #6200ea; /\* Purple \*/

color: white;

padding: 15px;

width: 100%;

display: flex;

justify-content: center; /\* Center the title horizontally \*/

align-items: center;

font-size: 32px; /\* Larger font size \*/

position: absolute; /\* Ensures it stays at the top \*/

top: 0;

left: 0;

right: 0;

}

.top-bar h1 {

margin: 0;

text-align: center; /\* Ensure title is centered \*/

}

/\* User Profile Section \*/

.user-profile {

background-color: #424242; /\* Dark gray \*/

color: white;

padding: 20px;

border-radius: 10px;

margin-top: 100px; /\* Removes gap between sections \*/

width: 95%; /\* Increased width for the grey area \*/

max-width: 1200px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);

}

/\* Communities Title (like LCM DASHBOARD title) \*/

.communities-title {

background-color: #6200ea; /\* Purple background \*/

padding: 15px;

text-align: center;

margin-bottom: 20px;

}

.communities-title h3 {

color: white;

font-size: 28px;

margin: 0;

}

/\* Communities Row Styling \*/

.community-row {

display: flex;

justify-content: space-between;

margin-top: 20px;

width: 100%;

border: 2px solid white; /\* White border around the entire table \*/

border-radius: 10px;

padding: 10px;

box-sizing: border-box; /\* Ensure padding doesn’t affect the layout \*/

align-items: stretch;

}

/\* Community Column Styling \*/

.community-column {

background-color: #424242; /\* Dark gray \*/

padding: 15px;

border-radius: 10px;

width: 18%; /\* Adjust width for each community \*/

text-align: center;

margin-right: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

border: 1px solid white; /\* White border around each column \*/

box-sizing: border-box; /\* Ensure padding and border are included in the width \*/

}

/\* Add background under each column's title \*/

.community-column h4 {

background-color: #6200ea; /\* Purple background \*/

color: white;

font-size: 22px;

padding: 10px;

border-radius: 5px;

margin-bottom: 10px;

}

/\* Column Paragraph Styling \*/

.community-column p {

font-size: 16px;

color: white;

}

/\* Dropdown button and content \*/

.dropdown-btn {

background-color: #6200ea;

color: white;

padding: 10px;

border-radius: 5px;

font-size: 14px;

margin-top: 10px;

cursor: pointer;

}

.dropdown-content {

display: none; /\* Hidden by default \*/

background-color: #212121; /\* Dark background \*/

color: white;

padding: 10px;

border-radius: 5px;

margin-top: 10px;

}

.dropdown-btn:hover + .dropdown-content {

display: block; /\* Display on hover \*/

}

/\* Add/Remove buttons \*/

.add-btn, .remove-btn {

background-color: #6200ea;

color: white;

padding: 10px;

border-radius: 5px;

margin-top: 10px;

cursor: pointer;

width: 40%;

}

.add-btn:hover, .remove-btn:hover {

background-color: #3700b3;

}

/\* "What are you up to today?" Section \*/

.question-title {

background-color: #6200ea; /\* Purple background \*/

padding: 15px;

text-align: center;

margin-bottom: 50px; /\* Increased space from previous section \*/

}

.question-title h3 {

color: white;

font-size: 28px;

margin: 0;

}

/\* Task Input Area with White Border \*/

.task-input {

width: calc(100% - 32px); /\* Adjust width to ensure it stays inside the grey area \*/

padding: 15px;

margin-bottom: 20px;

background-color: #333;

color: white;

border: 1px solid white; /\* White border \*/

border-radius: 10px;

font-size: 16px;

resize: none;

}

/\* Suggestions Dropdown \*/

.suggestions-dropdown {

display: none; /\* Hidden by default \*/

background-color: #212121;

color: white;

border-radius: 5px;

padding: 10px;

}

.suggestions-dropdown ul {

list-style-type: none;

padding: 0;

margin: 0;

}

.suggestions-dropdown ul li {

padding: 5px 0;

}

/\* Action Buttons Row \*/

.action-buttons {

display: flex;

justify-content: space-between; /\* Spread buttons evenly \*/

width: 100%;

margin-top: 20px;

}

/\* Button Styles \*/

button {

background-color: #6200ea;

color: white;

padding: 10px 20px;

border-radius: 5px;

font-size: 14px;

margin-right: 15px;

cursor: pointer;

flex: 1; /\* Allow buttons to grow and spread evenly \*/

}

button:hover {

background-color: #3700b3;

}

/\* Tasks Logged Table \*/

.tasks-logged-table {

margin-top: 20px; /\* Reduced margin to bring it closer to previous section \*/

width: 80%; /\* Adjusted width to center the table \*/

background-color: #424242;

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);

}

/\* Tasks Logged Table Title \*/

.tasks-logged-table h3 {

color: white;

background-color: #6200ea; /\* Purple background for title \*/

padding: 15px;

text-align: center;

margin-bottom: 20px;

border-radius: 10px;

}

/\* Table Styling \*/

table {

width: 100%;

border-collapse: collapse;

color: white;

background-color: #333;

border-radius: 10px;

overflow: hidden;

}

/\* Table Headers \*/

th {

background-color: #6200ea; /\* Purple background for headers \*/

color: white;

padding: 15px;

text-align: center;

}

/\* Table Cells \*/

th, td {

padding: 15px;

text-align: center;

border: 1px solid white;

}  
 **APP.JS**// Initialize Supabase client

const supabase = createClient('https://zjxsymoobzgtxurwdxsl.supabase.co', 'your-anon-key');

// Get the buttons and dropdowns

const categoryButton = document.getElementById('chooseCategoryBtn');

const verifierButton = document.getElementById('chooseVerifierBtn');

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

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

// Handle Category Button Click

categoryButton.addEventListener('click', async () => {

// Toggle visibility of the category dropdown

categoryDropdown.style.display = categoryDropdown.style.display === 'block' ? 'none' : 'block';

// Fetch categories from Supabase

const { data: categories, error: categoryError } = await supabase

.from('Categories')

.select('\*'); // Ensure we're selecting all categories

if (categoryError) {

console.error('Error fetching categories:', categoryError);

} else {

// Clear the existing category dropdown before populating

categoryDropdown.innerHTML = '';

// Populate the category dropdown with all categories

categories.forEach(category => {

const option = document.createElement('p');

option.textContent = category.name; // Assuming 'name' is the field in your Categories table

option.addEventListener('click', () => {

// Set the category as the selected category

categoryDropdown.style.display = 'none'; // Hide dropdown after selection

categoryButton.textContent = `Category: ${category.name}`; // Update button text

});

categoryDropdown.appendChild(option);

});

}

});

// Handle Verifier Button Click

verifierButton.addEventListener('click', async () => {

// Toggle visibility of the verifier dropdown

verifierDropdown.style.display = verifierDropdown.style.display === 'block' ? 'none' : 'block';

// Get selected category

const selectedCategory = categoryButton.textContent.replace('Category: ', '').trim();

if (!selectedCategory) {

alert('Please select a category first');

return;

}

// Fetch verifiers from Supabase based on the selected category

const { data: verifiers, error: verifierError } = await supabase

.from('Verifiers')

.select('\*')

.eq('category', selectedCategory); // Filter verifiers by the selected category

if (verifierError) {

console.error('Error fetching verifiers:', verifierError);

} else {

// Clear the existing verifier dropdown before populating

verifierDropdown.innerHTML = '';

// Populate the verifier dropdown with verifiers for the selected category

verifiers.forEach(verifier => {

const option = document.createElement('p');

option.textContent = verifier.name; // Assuming 'name' is the field in your Verifiers table

option.addEventListener('click', () => {

// Set the verifier as the selected verifier

verifierDropdown.style.display = 'none'; // Hide dropdown after selection

verifierButton.textContent = `Verifier: ${verifier.name}`; // Update button text

});

verifierDropdown.appendChild(option);

});

}

});

// Handle Submit Button Click

document.querySelector('.submit-btn').addEventListener('click', async () => {

const taskDescription = document.querySelector('.task-input').value;

const selectedCategory = categoryButton.textContent.replace('Category: ', '').trim();

const selectedVerifier = verifierButton.textContent.replace('Verifier: ', '').trim();

if (!taskDescription || !selectedCategory || !selectedVerifier) {

alert('Please fill out all fields before submitting.');

return;

}

// Insert the task into the Supabase Tasks table

const { data, error } = await supabase

.from('Tasks')

.insert([{ task\_description: taskDescription, category: selectedCategory, verifier: selectedVerifier }]);

if (error) {

console.error('Error inserting task:', error);

} else {

console.log('Task successfully inserted:', data);

alert('Task submitted successfully!');

document.querySelector('.task-input').value = ''; // Clear the task input field

}

});  
 **supabaseClient.js**import { createClient } from '@supabase/supabase-js';

// Replace with your Supabase URL and Anon Key

const supabaseUrl = 'https://your-project.supabase.co'; // Your Supabase URL

const supabaseKey = 'your-anon-key'; // Your Supabase Anon Key

const supabase = createClient(supabaseUrl, supabaseKey);

export default supabase;