README

Drinks Sustainability Tool

A comprehensive, one-stop-shop sustainability platform designed to empower Small and Medium-sized Enterprise (SME) drinks brands to measure, manage, and report their environmental impact with confidence.

Table of Contents

Why This Project?

SME drinks brands face a significant challenge: they are under increasing pressure from consumers and regulators to operate sustainably, but are often constrained by limited budgets, a lack of in-house expertise, and the sheer complexity of gathering and analyzing environmental data across their supply chains.

This platform aims to solve that problem by providing an affordable, intuitive, and credible tool that transforms complex Life Cycle Assessment (LCA) data into a clear, actionable, and commercial advantage.

Key Features (MVP)

- **Guided Onboarding:** A "gamified," step-by-step journey to simplify data collection for both clients and their suppliers.
- **Automated LCA Engine:** Asynchronously calculates product-level environmental footprints (carbon, water, waste) using a server-hosted OpenLCA instance for scientific credibility.
- **Interactive Dashboard:** Clean, simple visualizations of company-wide and SKU-level environmental impacts.
- **Supplier Collaboration Portal:** A dedicated, simplified portal for contract manufacturers to submit their data directly, enabling accurate Scope 3 emissions tracking.

- **Human-in-the-Loop Validation:** A core feature allowing clients to submit their reports for expert review by the Avallen Solutions team, ensuring accuracy and building trust.

Tech Stack

Our technology stack is chosen for scalability, security, and developer efficiency. For a detailed breakdown and rationale, please refer to the TECHSTACK.md document.

- **Frontend:** React (with Vite)
- **Backend:** Python (with Flask)
- **Database:** PostgreSQL
- **LCA Engine:** OpenLCA (Server-Hosted)
- **Authentication:** Replit Auth
- **Payments:** Stripe
- **Asynchronous Tasks:** Celery with Redis
- **Deployment:** Replit

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes.

Prerequisites

- A Replit account.
- Access to the project's Replit workspace.
- Necessary environment variables (secrets) configured in the Replit workspace (e.g., DATABASE_URL, STRIPE_SECRET_KEY, etc.).

Installation

1. **Fork the Repl:** Fork the main project repository within Replit to create your own development instance.
2. **Install Dependencies:** Open the Shell tab and run the installation commands for both the Python backend and the React frontend.

   ```
   # For the Python backend (from the root directory)
   pip install -r requirements.txt

   # For the React frontend (navigate to the frontend directory)
   cd frontend
   npm install
   ```

Running the Application

The application is configured to run with a single command from the Replit "Run" button. The .replit configuration file handles the concurrent startup of the Flask backend and the Vite frontend development server.

1. **Configure Secrets:** Ensure all required API keys and credentials are set in the Replit "Secrets" tool.
2. **Click "Run":** Press the main "Run" button at the top of the Replit workspace.
3. **Access the App:** The application will be available in the "WebView" tab within Replit.

Contributing

We welcome contributions to make this platform even better. Please follow these steps to contribute:

1. **Fork the Project:** Create your own fork of the repository.
2. **Create a Feature Branch:** (git checkout -b feature/AmazingFeature)
3. **Commit Your Changes:** (git commit -m 'Add some AmazingFeature')
4. **Push to the Branch:** (git push origin feature/AmazingFeature)
5. **Open a Pull Request:** Open a pull request against the main branch of the original repository for review.

Please make sure to update tests as appropriate and adhere to the project's coding standards.