TECHSTACK

Drinks Sustainability Tool

Version: 1.1
Date: 2025-07-18
Author: Replit Coach Too
Status: Draft

1. Introduction

This document outlines the technology stack selected for the "Drinks Sustainability Tool." The choices herein are based on the functional and non-functional requirements detailed in the Product Requirements Document (PRD), prioritizing scalability, security, developer efficiency, and a robust user experience.

2. Core Philosophy

Our technology selection is guided by the following principles:

- **Managed & Integrated Services:** We will prioritize using services that are managed and well-integrated within the Replit ecosystem to reduce operational overhead and accelerate development.
- **Scalability:** The architecture must be able to grow from an MVP to a full-featured platform supporting thousands of users without requiring a complete rewrite.
- **Security by Design:** We will use technologies with strong security track records and built-in security features.
- **Right Tool for the Job:** We will select technologies best suited for their specific purpose, such as a data-science-oriented language for the backend and a component-based framework for the interactive frontend.

3. Technology Breakdown

Component	Technology / Service	Rationale
Frontend Framework	**React (with Vite)**	React is the industry standard for building dynamic, responsive, and component-based user interfaces. It is ideal for creating the interactive dashboards and complex forms required for the platform. Vite provides an extremely fast development environment and optimized

		builds.
Backend Framework	**Python (with Flask)**	Python is the premier language for data analysis, machine learning, and scientific computing, making it the perfect choice for a backend that will interface with a complex calculation engine like OpenLCA. Flask is a lightweight and flexible framework that allows for rapid development of robust APIs and background tasks.
Database	**PostgreSQL**	As specified in the PRD, a relational database is required. PostgreSQL is a powerful, open-source, and highly reliable object-relational database system. It excels at handling complex queries and ensuring data integrity, which is critical for storing financial and environmental data. Replit has native support for PostgreSQL.
LCA Calculation Engine	**OpenLCA (Server-Hosted)**	This is a core requirement from the project's foundational documents. We will use a server-hosted instance of OpenLCA, which will be called by our Python backend. This ensures our calculations are based on a credible, industry-recognized, and scientifically robust engine.
Authentication	**Replit Auth**	As mandated by the PRD, Replit Auth will be used for all user authentication. It provides a secure, simple, and fully managed login system out-of-the-box, handling user

		sign-up, login, and session management, which drastically reduces development time and security risks.
Payment Processing	**Stripe**	To handle the tiered, revenue-based subscription model, a secure and reliable payment processing system is required. Stripe is the industry standard, offering robust APIs, excellent developer documentation, and seamless integration for managing recurring subscriptions and online payments.
Asynchronous Tasks	**Celery (with Redis)**	To handle the asynchronous generation of complex LCA reports without blocking the user interface, a task queue is essential. Celery is a robust and widely-used task queue for Python. Redis will serve as the message broker, providing a fast and reliable communication channel between the Flask application and the Celery workers.
Data Visualization	**Recharts**	For the React frontend, Recharts is an excellent library for building composable charts. It provides a wide range of chart types (bar, line, pie) needed for the dashboard and makes it easy to create beautiful, interactive, and responsive data visualizations.
Deployment & Hosting	**Replit Deployments**	The entire application, including the frontend, backend, database, and background workers, will be hosted and deployed on

		Replit. This provides a unified environment for development, testing, and production, simplifying the CI/CD pipeline and infrastructure management.