

KPI & Goal-Setting Dashboard Guide

For the Replit Development Agent

Version: 1.0
Date: 2025-08-29
Author: Replit Coach Too
Status: Draft

1. Objective

This document provides a detailed technical guide for the Replit Agent to build the **KPI & Goal-Setting Dashboard**. This feature will replace the current, basic dashboard with a dynamic, strategic interface that allows users to track a library of preset and custom KPIs, set meaningful goals, and visualize their progress over time.

2. Part 1: Backend & Database Enhancements

2.1. New Table: kpis

This table will store the definitions for all KPIs, both preset and custom.

| Column Name | Data Type | Constraints | Description |
|-------------|--------------|-----------------------------------------|-------------------------------------------------------------------------------|
| id | UUID | PRIMARY KEY | Unique identifier for the KPI. |
| company_id | UUID | FOREIGN KEY (companies.id), NULLABLE | If NULL, it's a preset KPI. If populated, it's a custom KPI for that company. |
| kpi_name | VARCHAR(255) | NOT NULL | The display name of the KPI (e.g., "Energy Intensity"). |
| kpi_type | VARCHAR(50) | NOT NULL | e.g., 'Environmental', 'Social', 'Engagement'. |
| unit | VARCHAR(50) | NOT NULL | The unit of the final calculated value |

| | | | |
|--------------|-------|----------|----------------------------------------------------|
| | | | (e.g., 'kWh/Litre', '%'). |
| formula_json | JSONB | NOT NULL | Stores the calculation logic. See structure below. |

Structure for formula_json:

```
// For a preset KPI
{
  "numerator": "total_kwh_consumption",
  "denominator": "total_litres_produced"
}

// For a custom KPI created by a user
{
  "numerator_data_point": "company_data.value WHERE data_type='operational_water_use'",
  "denominator_data_point": "products.total_production_volume"
}
```

2.2. New Table: project_goals (for Qualitative Goals)

| Column Name | Data Type | Constraints | Description |
|-------------|--------------|----------------------------|---------------------------------------------------------------------------------------|
| id | UUID | PRIMARY KEY | Unique ID for the project goal. |
| company_id | UUID | FOREIGN KEY (companies.id) | Links to the company. |
| goal_title | VARCHAR(255) | NOT NULL | e.g., "Achieve B Corp Certification". |
| milestones | JSONB | | An array of milestone objects, e.g., [{"text": "Complete BIA", "is_complete": true}]. |

2.3. Modification to companies Table

| Column Name | Data Type | Constraints | Description |
|------------------------|-----------|-------------|----------------------------------------------------|
| ... (existing columns) | | | |
| dashboard_layout | JSONB | NULLABLE | Stores the user's custom arrangement of KPI cards. |

2.4. New Backend Service & API Endpoints

- **New Service: KpiCalculationService**
 - This service will contain the logic to read a formula_json from the kpis table, fetch the required data from other tables (e.g., company_data, products), and calculate the final value for any given KPI.
- **New API Endpoints:**
 - GET /api/dashboard/kpis: Fetches all preset and custom KPIs for a user, along with their current calculated values.
 - POST /api/kpis/custom: Creates a new custom KPI.
 - POST /api/goals/project: Creates a new qualitative project goal.
 - PUT /api/dashboard/layout: Saves the user's custom dashboard layout.

3. Part 2: Frontend Implementation

The existing dashboard page (/app/dashboard) will be completely overhauled.

3.1. The KPI Library & Display

- **UI:** The main dashboard will be a customizable grid of KpiCard components.
- **Logic:** The page will fetch data from GET /api/dashboard/kpis. For each KPI returned, it will render a KpiCard.
- **KpiCard Component:**
 - Displays the kpi_name and its current value.
 - Includes a trend indicator (e.g., an up or down arrow with a percentage change from the previous period).
 - If a goal is set for this KPI, it will also display the progress bar/gauge and target value from the company_goals table.

3.2. The Custom KPI Builder

- **UI:** A "Create a Custom KPI" button on the dashboard will launch a wizard modal.

- **Wizard Steps:**
 1. **Name Your KPI:** A text input for kpi_name.
 2. **Define Calculation:**
 - **Numerator:** A dropdown menu populated with all available raw data points from the user's account (e.g., "Total Water Consumption (Litres)", "Total Waste Generated (kg)").
 - **Denominator:** A second dropdown for selecting a denominator to create an intensity metric.
 3. **Set Unit:** A text input for the final unit.
- **Logic:** On submission, the wizard calls POST /api/kpis/custom. The new custom KPI will then automatically appear on the dashboard.

3.3. Qualitative Goal Tracking Module

- **UI:** A new section on the dashboard titled "Projects & Initiatives".
- **Functionality:**
 - A button to "Add a Project Goal".
 - Each project goal is displayed as a card with a title and a progress bar representing the percentage of completed milestones.
 - Users can click on a project to view, add, and check off individual milestones.

3.4. Dashboard Customisation

- **Technology:** Implement a drag-and-drop library like **react-beautiful-dnd**.
- **Logic:** The user can drag and reorder the KpiCard components on the dashboard. When the user finishes dragging, the new layout (an ordered list of kpi_ids) is saved to the backend via the PUT /api/dashboard/layout endpoint. The next time the page loads, it will render the cards in the user's preferred order.