1. Introduction

1.1. Problem Definition

The current "Guided Sustainability Report" feature fails to produce a coherent, visually professional report. The output is jumbled, poorly formatted, and non-customizable, frustrating users and undermining the platform's value. The existing approach, which attempts to programmatically generate a static PDF, is fundamentally unsuited for creating complex, multi-format, and editable reports.

1.2. Vision & Goals

The vision is to replace the current functionality with a new, dynamic, and interactive "Report Builder" that empowers users to effortlessly generate professional, brand-aligned sustainability reports in multiple formats, including PDF and PowerPoint. The system will be a "click of a few buttons" experience, transforming raw data into a compelling, customizable narrative.

Primary Goals:

- **Remove Existing Functionality:** Completely remove all code and functionality related to the previous "Guided Sustainability Report" feature to prevent conflicts and start fresh.
- **Enable Customization:** Allow users to easily select, reorder, and exclude report sections.
- **Ensure Visual Quality:** Generate reports that are visually professional, consistent with the platform's brand, and optimized for high-quality export.
- **Provide Multi-Format Export:** Support the generation of both high-quality PDF and editable PowerPoint (PPTX) files.

2. User Flow & User Stories

2.1. High-Level User Flow

1. User navigates to a new "Reports" section on the main dashboard.
2. User clicks a "Create New Report" button.
3. The Report Builder interface loads, showing a library of available report "blocks."
4. User drags and drops blocks into a central canvas to assemble the report.
5. User can customize the data and narrative within each block.
6. User clicks an "Export" button and selects the desired format (PDF, PPTX).
7. The system generates and downloads the report file to the user's computer.

2.2. User Stories

As a user, I want to...

- ...generate a comprehensive annual sustainability report with a single click so that I can easily share my progress with stakeholders.
- ...see a clear, concise overview of my company's key environmental metrics (Carbon, Water, Waste) at the top of my report so that the most important information is

immediately visible.
- ...be able to drag and drop different data visualizations and text sections into my report so that I can customize the narrative and flow.
- ...export my report as a high-quality PDF so that it looks professional when printed or shared digitally.
- ...export my report as an editable PowerPoint file so that I can easily make final adjustments or use it for presentations.
- ...remove and replace the existing, unusable report generation feature so that the platform is clean and provides a functional experience.

3. Feature Requirements

3.1. Report Builder Frontend

- **Modular UI:** The frontend must be a single-page application (SPA) with a drag-and-drop interface.
- **Block Library:** A sidebar will display a library of pre-designed "blocks."
  - **Data Blocks:** Pre-configured visualizations (e.g., charts, tables) for metrics like Carbon Footprint, Water Usage, and Waste Generated.
  - **Narrative Blocks:** Pre-formatted text sections for things like the "Executive Summary," "Introduction," and "Next Steps."
- **WYSIWYG Editor:** Each text block will have a rich-text editor for user customization.
- **Real-time Preview:** The central canvas will show a live, high-fidelity preview of the report as it is being built.

3.2. Data Management & API

- **Endpoint:** A new API endpoint will be created to accept the user's report configuration (an ordered list of blocks with their specific data inputs).
- **Data Aggregation:** The API will be responsible for fetching all relevant data from the backend database (as defined in the existing "Sustainability Tool" documents) and passing it to the export engine.

3.3. Export Engine

- **HTML-to-PDF Conversion:**
  - Use a Puppeteer-based service to render the final, client-side HTML into a high-quality PDF.
  - The service must handle page breaks, headers, and footers correctly.
- **HTML-to-PowerPoint Conversion:**
  - Use a Node.js library like pptxgenjs to create an editable PowerPoint file.
  - The service will map the data and images from the HTML blocks to the corresponding slides and elements in the PPTX file.
  - This conversion must preserve charts as native PowerPoint objects if possible, and images as high-quality PNGs.

4. Technical Considerations

- **Frontend Framework:** Use a modern frontend framework (e.g., React or Angular) to build the interactive UI.
- **Backend Framework:** Continue using Node.js with Express.js.
- **PDF Library:** Puppeteer for PDF generation.
- **PowerPoint Library:** pptxgenjs for PowerPoint generation.
- **Styling:** A central CSS file will manage all styling, ensuring consistency with the platform's existing brand and the LCA_Report_31_2025-09-08.pdf document.
- **Database Integration:** The new features will connect to the existing database to fetch all necessary data points.
- **Refactoring:** A key part of this project is the removal of the old report generation code to ensure a clean codebase. This will be an explicit instruction.
- **Rollback Plan:** In the event of an issue, we must have a clear rollback plan to revert to the previous working state.