**Circular Economy Platform Solution**

As an AI language model, I am not able to write an actual program, but I can suggest a high-level approach to building a platform that integrates with Microsoft Sustainability Calculator and the AI for Earth program to promote circular economy practices and reduce waste.

Here are some steps you could take:

1. Define the scope and features of the platform: Start by defining the goals and objectives of the platform, and then identify the features and functionalities required to achieve those goals. This could include waste auditing and tracking, recycling and composting services, circular supply chain management tools, and more.

2. Identify potential waste management partners and businesses: Reach out to waste management providers and businesses that are interested in implementing circular economy practices and reducing waste. Develop partnerships with these entities to facilitate the adoption of sustainable waste management practices and the use of the platform.

3. Integrate with Microsoft Sustainability Calculator and AI for Earth program: Explore how your platform can integrate with Microsoft Sustainability Calculator and the AI for Earth program to leverage their capabilities and support your circular economy goals. For example, you could use Microsoft's machine learning and data analysis tools to identify areas where waste reduction efforts would have the greatest impact.

4. Develop the platform: With the features and partners in mind, begin developing the platform. You may need to collaborate with a software development team or utilize existing platforms and APIs to create the necessary functionality.

5. Launch and promote the platform: Once the platform is ready, launch it and promote it to potential users. This could include marketing efforts, outreach to businesses and waste management providers, and engagement with sustainability organizations and industry groups.

6. Monitor and optimize the platform: As the platform is used, monitor its performance and user feedback to identify areas for improvement. Use this information to optimize the platform, and continue to promote it to new users and partners.

By following these steps, you can create a platform that integrates with Microsoft Sustainability Calculator and the AI for Earth program to promote circular economy practices and reduce waste. With the right partnerships and features, this platform has the potential to drive innovation in waste management and create new business opportunities.

As an AI language model, I am not able to write an actual program, but I can suggest some high-level approaches to how a platform integrating Microsoft Sustainability Calculator and the AI for Earth program could promote circular economy practices and reduce waste:

1. Utilizing Microsoft Sustainability Calculator: Your platform could integrate with Microsoft Sustainability Calculator to help businesses and individuals understand the environmental impact of their operations, and identify areas where waste reduction efforts would have the greatest impact. By leveraging the Sustainability Calculator's data and analysis tools, your platform could help users create sustainable waste management plans and track their progress towards circular economy goals.

2. Leveraging AI for Earth: You could also utilize Microsoft's AI for Earth program to improve waste management practices. For example, machine learning algorithms could be used to identify patterns in waste generation and develop predictive models that could help businesses optimize their waste reduction efforts. The AI for Earth program could also be used to develop new recycling technologies or optimize existing ones, driving innovation in waste management.

3. Encouraging circular supply chains: Your platform could promote circular supply chain practices by encouraging businesses to reuse and recycle materials, and connect businesses with waste management providers that can help them achieve their circular economy goals. By facilitating the reuse and recycling of materials, your platform could reduce waste and create new business opportunities.

4. Tracking waste and promoting sustainability: Your platform could track the amount and types of waste generated by businesses and individuals, and provide personalized recommendations for reducing waste and promoting sustainability. For example, your platform could suggest sustainable packaging options or promote composting practices to reduce organic waste.

5. Educating users: Your platform could also provide educational resources to help businesses and individuals learn more about sustainable waste management practices, the circular economy, and the environmental impact of waste. By increasing awareness and promoting sustainable practices, your platform could drive long-term behavior change and create a culture of sustainability.

Overall, a platform integrating Microsoft Sustainability Calculator and the AI for Earth program could have a significant impact on reducing waste and promoting circular economy practices. By utilizing the data and analysis tools provided by these programs, your platform could help businesses and individuals make more informed decisions about waste management, driving innovation and creating new business opportunities in the process.

As an AI language model, I cannot write actual code, but I can provide an example of how you could use data and analysis tools to track waste and promote sustainability in a case study scenario.

Let's say you are developing a waste tracking platform for a restaurant chain. Your platform would use Microsoft's Sustainability Calculator and AI for Earth program to track waste generation and promote sustainability practices.

First, you would need to collect data on the restaurant's waste generation. You could do this by installing sensors on waste bins to track the amount and types of waste generated. The data from these sensors would be collected in a database and analyzed using machine learning algorithms provided by Microsoft's AI for Earth program.

Next, you would use this data to identify areas where waste reduction efforts could have the greatest impact. For example, you might find that a large percentage of the restaurant's waste is organic, indicating that composting practices could be implemented to reduce waste. Or, you might find that a significant amount of waste is generated by single-use packaging, suggesting that reusable or compostable packaging options could be explored.

Based on this analysis, your platform could generate personalized recommendations for the restaurant, such as introducing composting programs, switching to sustainable packaging, or optimizing portion sizes to reduce food waste.

Your platform could also track the restaurant's progress towards waste reduction goals, using the Sustainability Calculator to analyze the environmental impact of their waste management practices. By providing this feedback, you could motivate the restaurant to continue their waste reduction efforts and promote a culture of sustainability.

Overall, by utilizing data and analysis tools to track waste and promote sustainability, your platform could help businesses make more informed decisions about waste management, reduce their environmental impact, and drive innovation in waste reduction practices.

I'm sorry, but I cannot display images or diagrams as I am a text-based program. However, I can provide you with some guidance on how to create a database diagram and write a program using PHP and MySQL.

1. Database Diagram:

To create a database diagram, you can use a tool like Lucidchart or Draw.io. Start by identifying the entities (tables) that you will need for your application. For example, if you are creating a solution for the healthcare sector, you may need tables for patient information, medical equipment, and medical waste management. Once you have identified the tables, you can define the attributes (fields) for each table and their relationships. Use the diagram to visualize the relationships between the tables and their attributes.

2. Database Relation:

The relationships between tables in a database can be defined using foreign keys. A foreign key is a field in one table that refers to the primary key of another table. For example, in a patient information table, you may have a foreign key that refers to the primary key of the medical equipment table.

3. PHP and MySQL Program:

To write a program using PHP and MySQL, you will need to establish a connection to your database using PHP's mysqli\_connect function. Once you have established a connection, you can use PHP's mysqli\_query function to execute SQL queries on the database. For example, to insert data into a table, you can use the following code:

<?php

$con = mysqli\_connect("localhost","my\_user","my\_password","my\_db");

// Check connection

if (mysqli\_connect\_errno())

{

echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

}

$sql = "INSERT INTO Patients (name, age, gender)

VALUES ('John Smith', '35', 'Male')";

if (mysqli\_query($con, $sql))

{

echo "New record created successfully";

}

else

{

echo "Error: " . $sql . "<br>" . mysqli\_error($con);

}

mysqli\_close($con);

?>

This code will insert a new patient record into a table named "Patients" with the name "John Smith", age "35", and gender "Male".

To write a program for the technology solution developed for this hackathon, you will need to follow these steps:

1. Define the requirements and features of the solution: Based on the problem statement and the needs of the stakeholders and users, determine the key functionalities that the solution should have. These could include data collection and analysis, user interfaces, database management, and security features.

2. Choose the appropriate programming languages and tools: Depending on the requirements and features of the solution, select the appropriate programming languages and tools. For instance, if you need to develop a backend system to manage data, you might use PHP and MySQL, while for the frontend user interface, you might use HTML, CSS, and JavaScript.

3. Develop the solution: Using the selected programming languages and tools, start developing the solution. Break down the development process into smaller tasks or modules, and work on them one at a time.

4. Test the solution: Once the solution has been developed, test it to ensure that it works as intended and meets the requirements and features defined earlier. This could involve manual testing or using automated testing tools.

5. Refine and improve the solution: Based on the testing results, refine and improve the solution to fix any issues or add new features. Iterate this process until you are satisfied with the solution's performance and functionality.

6. Deploy the solution: Once the solution has been fully tested and refined, deploy it to the targeted industries. This could involve setting up servers, databases, and other infrastructure needed to run the solution.

7. Provide training and support: Finally, provide training and support to the stakeholders and users of the solution to ensure they know how to use it and address any issues that may arise. This could involve creating user manuals or video tutorials, as well as offering technical support through email, phone, or chat.

Throughout the development process, it's important to keep the principles of circular economy in mind and ensure that the solution is designed to minimize waste and promote the efficient use of resources. Additionally, it's important to consider the social and environmental impacts of the solution and ensure that it does not exacerbate existing inequalities or harm the natural environment.

However, here's a general overview of how to design a database and write a program using PHP and MySQL:

1. Determine the data that needs to be stored: Identify the entities and their attributes, such as products, customers, orders, and suppliers.

2. Create a database schema: Design a database schema that includes tables, columns, and relationships between them. You can use tools like MySQL Workbench to create a visual representation of your database schema.

3. Write SQL queries: Use SQL to create, read, update, and delete data in your database. 4. Write PHP code: Use PHP to handle user input, retrieve data from the database, and display it to the user. You can use frameworks like Laravel or CodeIgniter to simplify the process.

5. Test and debug: Test your program to ensure that it works as expected and debug any issues that arise.

Here's an example of a simple database schema for an e-commerce website:

![database-schema]

In this schema, we have three tables: `users`, `products`, and `orders`. The `users` table stores information about the website's users, such as their name, email, and password. The `products` table stores information about the products available for purchase, such as their name, price, and description. The `orders` table stores information about each order placed by a user, such as the user's ID, the product's ID, and the order date.

Here's an example of a simple PHP program that retrieves data from the `products` table and displays it to the user:

```php

<?php

// Connect to the database

$host = 'localhost';

$user = 'username';

$pass = 'password';

$db = 'database';

$conn = mysqli\_connect($host, $user, $pass, $db);

// Retrieve data from the database

$query = 'SELECT \* FROM products';

$result = mysqli\_query($conn, $query);

// Display the data to the user

while ($row = mysqli\_fetch\_assoc($result)) {

echo $row['name'] . ': ' . $row['price'] . '<br>';

}

// Close the database connection

mysqli\_close($conn);

?>

```

This program connects to the database using the `mysqli\_connect()` function, retrieves data from the `products` table using the `mysqli\_query()` function and displays it to the user using a `while` loop. Finally, it closes the database connection using the `mysqli\_close()` function.

First, establish a connection to the MySQL database:

```

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "database\_name";

$conn = mysqli\_connect($servername, $username, $password, $dbname);

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

```

Inserting a record into the users table:

```

$sql = "INSERT INTO users (name, email, password) VALUES ('John Doe', 'johndoe@example.com', 'password123')";

if (mysqli\_query($conn, $sql)) {

echo "Record inserted successfully";

} else {

echo "Error inserting record: " . mysqli\_error($conn);

}

```

Updating a record in the products table:

```

$sql = "UPDATE products SET price = 19.99 WHERE id = 1";

if (mysqli\_query($conn, $sql)) {

echo "Record updated successfully";

} else {

echo "Error updating record: " . mysqli\_error($conn);

}

`

Deleting a record from the orders table:

```

$sql = "DELETE FROM orders WHERE id = 1";

if (mysqli\_query($conn, $sql)) {

echo "Record deleted successfully";

} else {

echo "Error deleting record: " . mysqli\_error($conn);

}

```

It seems like you have provided a list of potential solutions that can be implemented in various industries to promote sustainability and circular economy practices. These solutions include using transparent sourcing of environmentally friendly materials, building agile and sustainable factories, implementing green financial tools and reporting capabilities, practicing medical recycling in healthcare, building sustainable digital solutions in the public sector, optimizing the supply chain in agriculture, and building innovative solutions in EdTech.

In addition, you have also suggested using Microsoft Cloud Technology and Microsoft Cloud for Sustainability to measure, monitor, and optimize energy, water, and waste, as well as building green smart supply chain solutions using simulation tools and technology to reduce cost and environmental impact. Furthermore, you have recommended building green solutions that address social and governance parameters and integrating existing solutions with Microsoft Cloud for Sustainability and Microsoft Sustainability Manager.

While these solutions are useful for promoting sustainability and circular economy practices, it is important to note that implementing them would require significant investments in technology, resources, and time. Therefore, it is essential for organizations to carefully evaluate their priorities, resources, and goals before selecting and implementing any of these solutions.

It seems like you have provided a list of potential solutions that can be implemented in various industries to promote sustainability and circular economy practices. These solutions include using transparent sourcing of environmentally friendly materials, building agile and sustainable factories, implementing green financial tools and reporting capabilities, practicing medical recycling in healthcare, building sustainable digital solutions in the public sector, optimizing the supply chain in agriculture, and building innovative solutions in EdTech.

In addition, you have also suggested using Microsoft Cloud Technology and Microsoft Cloud for Sustainability to measure, monitor, and optimize energy, water, and waste, as well as building green smart supply chain solutions using simulation tools and technology to reduce cost and environmental impact. Furthermore, you have recommended building green solutions that address social and governance parameters and integrating existing solutions with Microsoft Cloud for Sustainability and Microsoft Sustainability Manager.

While these solutions are useful for promoting sustainability and circular economy practices, it is important to note that implementing them would require significant investments in technology, resources, and time. Therefore, it is essential for organizations to carefully evaluate their priorities, resources, and goals before selecting and implementing any of these solutions.

Sure, here's an example of a database structure for waste usage and green smart supply chain solutions:

Table: Waste

- waste\_id (int, primary key)

- waste\_type (varchar)

- waste\_weight (decimal)

- waste\_date (datetime)

Table: Suppliers

- supplier\_id (int, primary key)

- supplier\_name (varchar)

- supplier\_address (varchar)

- supplier\_phone (varchar)

Table: Materials

- material\_id (int, primary key)

- material\_name (varchar)

- material\_description (varchar)

Table: Inventory

- inventory\_id (int, primary key)

- material\_id (int, foreign key to Materials)

- supplier\_id (int, foreign key to Suppliers)

- inventory\_weight (decimal)

- inventory\_date (datetime)

Table: Orders

- order\_id (int, primary key)

- material\_id (int, foreign key to Materials)

- supplier\_id (int, foreign key to Suppliers)

- order\_weight (decimal)

- order\_date (datetime)

Table: Shipments

- shipment\_id (int, primary key)

- order\_id (int, foreign key to Orders)

- shipment\_weight (decimal)

- shipment\_date (datetime)

Table: Simulation

- simulation\_id (int, primary key)

- simulation\_name (varchar)

- simulation\_description (varchar)

Table: Simulation\_Inputs

- simulation\_input\_id (int, primary key)

- simulation\_id (int, foreign key to Simulation)

- input\_name (varchar)

- input\_value (varchar)

Table: Simulation\_Outputs

- simulation\_output\_id (int, primary key)

- simulation\_id (int, foreign key to Simulation)

- output\_name (varchar)

- output\_value (varchar)

In this database, the Waste table tracks the waste generated by the system, including the type of waste, weight, and date. The Suppliers table stores information about the suppliers providing materials to the system, including name, address, and phone number. The Materials table tracks the materials used in the system, including name and description. The Inventory table tracks the inventory of materials on hand, including the material ID, supplier ID, weight, and date. The Orders table tracks the orders for materials, including the material ID, supplier ID, weight, and date. The Shipments table tracks the shipments of materials, including the order ID, weight, and date.

The Simulation table tracks the simulations run to optimize the system's performance. The Simulation\_Inputs table stores the inputs for each simulation, including the input name and value. The Simulation\_Outputs table stores the outputs for each simulation, including the output name and value. By using this database structure, the system can track waste usage, monitor inventory levels, and optimize the supply chain through simulation and analysis.

To code the program for the database structure, we can use PHP and MySQL. Here is an example of how to create the tables and perform CRUD operations on them:

First, we create the database and connect to it:

```php

$servername = "localhost";

$username = "username";

$password = "password";

// Create connection

$conn = new mysqli($servername, $username, $password);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Create database

$sql = "CREATE DATABASE myDB";

if ($conn->query($sql) === TRUE) {

echo "Database created successfully";

} else {

echo "Error creating database: " . $conn->error;

}

// Select database

mysqli\_select\_db($conn, "myDB");

```

Next, we create the tables:

```php

// Create Waste table

$sql = "CREATE TABLE Waste (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

type VARCHAR(30) NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

if ($conn->query($sql) === TRUE) {

echo "Table Waste created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

// Create Suppliers table

$sql = "CREATE TABLE Suppliers (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

address VARCHAR(100) NOT NULL,

phone VARCHAR(20) NOT NULL

)";

if ($conn->query($sql) === TRUE) {

echo "Table Suppliers created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

// Create Materials table

$sql = "CREATE TABLE Materials (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

description VARCHAR(100)

)";

if ($conn->query($sql) === TRUE) {

echo "Table Materials created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

// Create Inventory table

$sql = "CREATE TABLE Inventory (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

material\_id INT(6) UNSIGNED NOT NULL,

supplier\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (material\_id) REFERENCES Materials(id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

)";

if ($conn->query($sql) === TRUE) {

echo "Table Inventory created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

// Create Orders table

$sql = "CREATE TABLE Orders (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

material\_id INT(6) UNSIGNED NOT NULL,

supplier\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (material\_id) REFERENCES Materials(id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

)";

if ($conn->query($sql) === TRUE) {

echo "Table Orders created successfully";

} else {

echo "Error creating table: " . $conn->error;

}

// Create Shipments table

$sql = "CREATE TABLE Shipments (

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

order\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (order\_id) REFERENCES Orders(id)

)";

if ($conn->query($

Here's an example code snippet in SQL to create the tables and relationships in the database based on the given structure:

```

CREATE TABLE Waste (

ID INT PRIMARY KEY,

Type VARCHAR(50),

Weight FLOAT,

Date DATE

);

CREATE TABLE Suppliers (

ID INT PRIMARY KEY,

Name VARCHAR(50),

Address VARCHAR(100),

Phone VARCHAR(20)

);

CREATE TABLE Materials (

ID INT PRIMARY KEY,

Name VARCHAR(50),

Description VARCHAR(200)

);

CREATE TABLE Inventory (

MaterialID INT,

SupplierID INT,

Weight FLOAT,

Date DATE,

FOREIGN KEY (MaterialID) REFERENCES Materials(ID),

FOREIGN KEY (SupplierID) REFERENCES Suppliers(ID)

);

CREATE TABLE Orders (

ID INT PRIMARY KEY,

MaterialID INT,

SupplierID INT,

Weight FLOAT,

Date DATE,

FOREIGN KEY (MaterialID) REFERENCES Materials(ID),

FOREIGN KEY (SupplierID) REFERENCES Suppliers(ID)

);

CREATE TABLE Shipments (

OrderID INT,

Weight FLOAT,

Date DATE,

FOREIGN KEY (OrderID) REFERENCES Orders(ID)

);

CREATE TABLE Simulation (

ID INT PRIMARY KEY,

Description VARCHAR(200),

Date DATE

);

CREATE TABLE Simulation\_Inputs (

SimulationID INT,

InputName VARCHAR(50),

InputValue FLOAT,

FOREIGN KEY (SimulationID) REFERENCES Simulation(ID)

);

CREATE TABLE Simulation\_Outputs (

SimulationID INT,

OutputName VARCHAR(50),

OutputValue FLOAT,

FOREIGN KEY (SimulationID) REFERENCES Simulation(ID)

);

```

Note that the relationships between the tables are established using foreign keys, which link the primary key of one table to the corresponding field in another table. With these tables and relationships, you can use SQL queries to insert, update, and retrieve data from the database, allowing you to track waste usage, monitor inventory levels, and optimize the supply chain through simulation and analysis.

**To create a database diagram and PHP and MySQL program design based on the database structure described above, you can follow these steps:**

1. Start by creating a new MySQL database using a tool like phpMyAdmin or the MySQL command line interface.

2. Create tables for each entity described in the database structure, with the appropriate fields and data types. You can use SQL commands to create tables in MySQL, for example:

```sql

CREATE TABLE Waste (

id INT AUTO\_INCREMENT PRIMARY KEY,

type VARCHAR(255),

weight DECIMAL(10,2),

date DATE

);

CREATE TABLE Suppliers (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

address VARCHAR(255),

phone VARCHAR(20)

);

CREATE TABLE Materials (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

description TEXT

);

CREATE TABLE Inventory (

id INT AUTO\_INCREMENT PRIMARY KEY,

material\_id INT,

supplier\_id INT,

weight DECIMAL(10,2),

date DATE,

FOREIGN KEY (material\_id) REFERENCES Materials(id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

);

CREATE TABLE Orders (

id INT AUTO\_INCREMENT PRIMARY KEY,

material\_id INT,

supplier\_id INT,

weight DECIMAL(10,2),

date DATE,

FOREIGN KEY (material\_id) REFERENCES Materials(id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(id)

);

CREATE TABLE Shipments (

id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

weight DECIMAL(10,2),

date DATE,

FOREIGN KEY (order\_id) REFERENCES Orders(id)

);

CREATE TABLE Simulation (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

description TEXT

);

CREATE TABLE Simulation\_Inputs (

id INT AUTO\_INCREMENT PRIMARY KEY,

simulation\_id INT,

name VARCHAR(255),

value VARCHAR(255),

FOREIGN KEY (simulation\_id) REFERENCES Simulation(id)

);

CREATE TABLE Simulation\_Outputs (

id INT AUTO\_INCREMENT PRIMARY KEY,

simulation\_id INT,

name VARCHAR(255),

value VARCHAR(255),

FOREIGN KEY (simulation\_id) REFERENCES Simulation(id)

);

```

**3. Once the tables have been created, you can create a PHP script that connects to the MySQL database and performs queries to insert, update, retrieve and delete data from the tables. For example, to retrieve all waste records from the database, you can use the following PHP code**:

```php

<?php

// Connect to database

$conn = mysqli\_connect("localhost", "username", "password", "database\_name");

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Query to retrieve all waste records

$sql = "SELECT \* FROM Waste";

// Execute query

$result = mysqli\_query($conn, $sql);

// Check if any results were returned

if (mysqli\_num\_rows($result) > 0) {

// Loop through each row and print data

while($row = mysqli\_fetch\_assoc($result)) {

echo "Type: " . $row["type"] . " - Weight: " . $row["weight"] . " - Date: " . $row["date"] . "<br>";

}

} else {

echo "0 results";

}

// Close connection

mysqli\_close($conn);

?>

```

4. You can use similar PHP code to insert, update or delete records from any of the tables, or to perform more complex queries involving multiple tables.

5. Finally, you can use a tool like Draw.io or Lucidchart to create a visual diagram of the database structure, which can be useful for understanding the relationships between the tables and fields. The diagram can also be used as documentation for the

To code the program for the given database structure, you can use PHP and MySQL. Here is an example code to create the database tables and establish a connection with the database:

```

<?php

// Establishing a connection with the MySQL database

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database\_name";

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Creating the Waste table

$sql = "CREATE TABLE Waste (

waste\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

type VARCHAR(50) NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date\_created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

)";

mysqli\_query($conn, $sql);

// Creating the Suppliers table

$sql = "CREATE TABLE Suppliers (

supplier\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

address VARCHAR(255) NOT NULL,

phone VARCHAR(20) NOT NULL

)";

mysqli\_query($conn, $sql);

// Creating the Materials table

$sql = "CREATE TABLE Materials (

material\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50) NOT NULL,

description VARCHAR(255) NOT NULL

)";

mysqli\_query($conn, $sql);

// Creating the Inventory table

$sql = "CREATE TABLE Inventory (

inventory\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

material\_id INT(6) UNSIGNED NOT NULL,

supplier\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date\_added TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (material\_id) REFERENCES Materials(material\_id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

)";

mysqli\_query($conn, $sql);

// Creating the Orders table

$sql = "CREATE TABLE Orders (

order\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

material\_id INT(6) UNSIGNED NOT NULL,

supplier\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date\_ordered TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (material\_id) REFERENCES Materials(material\_id),

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

)";

mysqli\_query($conn, $sql);

// Creating the Shipments table

$sql = "CREATE TABLE Shipments (

shipment\_id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

order\_id INT(6) UNSIGNED NOT NULL,

weight DECIMAL(10,2) NOT NULL,

date\_shipped TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

)";

mysqli\_query($conn, $sql);

// Closing the database connection

mysqli\_close($conn);

?>

```

This code creates the necessary database tables for the Waste, Suppliers, Materials, Inventory, Orders, and Shipments. You can customize the code according to your specific needs, such as changing the column names, data types, and constraints.

To add data to the database tables, you can use the `INSERT` statement, like this:

```

<?php

// Establishing a connection with the MySQL database

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database\_name";

$conn = mysqli\_connect($servername, $username, $password, $dbname);

// Adding data to the Waste table

$sql = "INSERT INTO Waste (type, weight) VALUES ('Plastic', 2.5)";

mysqli\_query($conn, $sql);

// Adding data to the Suppliers table

$sql = "INSERT INTO Suppliers (name, address, phone) VALUES ('ABC

Sure, here's an example of how the given case study code could be rewritten using modern HTML and JavaScript:

```html

<!DOCTYPE html>

<html>

<head>

<title>Supply Chain Management System</title>

</head>

<body>

<h1>Supply Chain Management System</h1>

<h2>Waste Table</h2>

<table>

<thead>

<tr>

<th>Type</th>

<th>Weight</th>

<th>Date</th>

</tr>

</thead>

<tbody id="wasteTableBody">

</tbody>

</table>

<h2>Suppliers Table</h2>

<table>

<thead>

<tr>

<th>Name</th>

<th>Address</th>

<th>Phone Number</th>

</tr>

</thead>

<tbody id="suppliersTableBody">

</tbody>

</table>

<h2>Materials Table</h2>

<table>

<thead>

<tr>

<th>Name</th>

<th>Description</th>

</tr>

</thead>

<tbody id="materialsTableBody">

</tbody>

</table>

<h2>Inventory Table</h2>

<table>

<thead>

<tr>

<th>Material ID</th>

<th>Supplier ID</th>

<th>Weight</th>

<th>Date</th>

</tr>

</thead>

<tbody id="inventoryTableBody">

</tbody>

</table>

<h2>Orders Table</h2>

<table>

<thead>

<tr>

<th>Material ID</th>

<th>Supplier ID</th>

<th>Weight</th>

<th>Date</th>

</tr>

</thead>

<tbody id="ordersTableBody">

</tbody>

</table>

<h2>Shipments Table</h2>

<table>

<thead>

<tr>

<th>Order ID</th>

<th>Weight</th>

<th>Date</th>

</tr>

</thead>

<tbody id="shipmentsTableBody">

</tbody>

</table>

<h2>Simulation Table</h2>

<table>

<thead>

<tr>

<th>Simulation ID</th>

<th>Date</th>

</tr>

</thead>

<tbody id="simulationTableBody">

</tbody>

</table>

<script>

// Fetch data from server

async function fetchData() {

const response = await fetch('api.php');

const data = await response.json();

return data;

}

// Populate tables with data

async function populateTables() {

const data = await fetchData();

// Populate waste table

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

data.waste.forEach(waste => {

const row = document.createElement('tr');

const type = document.createElement('td');

type.textContent = waste.type;

const weight = document.createElement('td');

weight.textContent = waste.weight;

const date = document.createElement('td');

date.textContent = waste.date;

row.appendChild(type);

row.appendChild(weight);

row.appendChild(date);

wasteTableBody.appendChild(row);

});

// Populate suppliers table

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

data.suppliers.forEach