C6CAFY

CARBON FOOTPRINT CALCULATOR FOR FOOD INDUSTRY

Version	Description of Change	Author	Date
V 0.1	Initial Draft.	Adeline Deepa Deysin Mihai	12/02/2025
V 0.2	Second Draft Modifications only in the Appendix	Adeline Deepa Deysin Mihai	25/02/2025
V 0.3	Final Version We deleted: • the number of dishes, because since we are already calculating the CO2 of each food used by the user, we were "overlapping" the calculation. Consequently, we had to correct the calculation formula presented previously. • The download of reports, we did not think it was necessary considering that the CO2 consumption will be available on the website for the customer to access. • the advices. • the calculation with transportation, which is already included in the food issue. • Appendix Modifications • Change in the ERD Diagram	Adeline Deepa Deysin Mihai	09/04/2025

INTRODUCTION 4
Purpose 4
Scope 4
Background 4
References 4
Assumptions Constraints 5
Assumptions 5
Constraints 5
Document Overview 5
METHODOLOGY 5
FUNCTIONAL REQUIREMENTS 7
Context 7
User Requirements 7
Functional Requirements 7
Functional Requirements 8
OTHER REQUIREMENTS 9
Interface Requirements 9
Hardware Interfaces 9
Non-Functional Requirements 9
Security and Privacy 10
Data Retention 11
Error Handling 11
Conventions/Standards 11
Risk Analysis 11
Scope 11
Human Resources 11
Quality 12
Analysis 12
Functional Requirements: 12
Statuses Diagram 14
Non-functional Requirements 14
Glossary 17
GRAPHS: 18
Use Case 18
Flow Charts: 18
Flowcharts Process: 18
Appendix 19

INTRODUCTION

The carbon footprint calculation represents an essential tool in shaping the EU's GREEN DEAL strategies and policies.

This document describes the technical method and operational requirements needed to design a digital application for calculating the carbon footprint of food SMEs in the European Union.

- Calculation/ measure/inform: inform about the footprint and personal tendencies.

Scope SME food industry: restaurants, traiteur, cloud kitchen.

Purpose

The project aims to provide a standardized tool – the **C**arbon Footprint **Ca**lculator for **F**ood **I**ndustry - **C6CAFY®** (digital application) - for calculating food SMEs' carbon footprint (greenhouse gas equivalent).

The aim of the carbon footprint meter software is to enable SMEs that use food in their product chain to have a visual overview and database of their emissions.

This software is aimed at being opened by the user on a free online webpage through a specific login that will enable the user to access his previous data.

Scope

The service specifications document underlines the functional and non-functional requirements, as well as the limitations, for using the software in the operational environment by the food SMEs.

Background

The pink group is creating this software for the master of digitalization. The group takes on no responsibilities other than making it work for Q1 and 2, 2025.

References

Following the applicable EU.

AWAC. (2025). Carbon calculator [Online calculator]. https://calculateurs.awac.be/app/form/ACTIVITY/ACTIVITY. Accessed on January 7, 2025.

Eurostat. (2025). Emissions food [Website]. European Commission. https://ec.europa.eu/eurostat/web/main/search/-/search/estatsearchportlet_WAR_estatsearchportlet_INSTANCE_bHVzuvn1SZ8J?text=emissions+foodAccessed on January 7, 2025.

Intergovernmental Panel on Climate Change (IPCC). *Data*. https://www.ipcc.ch/data/. Accessed on January 7, 2025.

Wikipedia contributors. *Greenhouse gas emissions*. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Greenhouse_gas_emissions. Accessed on January 7, 2025.

WWF Suisse. (2025). Ecological footprint calculator [Online calculator]. https://www.wwf.ch/fr/vie-durable/calculateur-d-empreinte-ecologique. Accessed on January 7, 2025.

Assumptions Constraints

Assumptions

- The software will have to respect all applicable laws (ex. GDPR).
- **Emission Calculation Accuracy –** The methodology for carbon footprint calculations will follow scientific models and conversion factors.
- **Security and Data Privacy –** The system will comply with data protection laws to handle sensitive business data.
- **Support for Multiple Food Categories –** The system will accommodate diverse food products with varying carbon footprints.
- The platform shall be accessible at all times, hence the need for storage.
- **Limitation –** The software does not take into account the food which is thrown away.
- The software shall enable the user to add consumed/used products from a list to its consumed goods list.
- The software shall be able to add all these consumed good carbon footprints and show the results in a numbered way and in a visual way to the user.

Constraints

- **Data Inconsistencies** – Variability in data quality and availability from suppliers and partners may affect the accuracy of carbon footprint calculations.
- **Regulatory Changes** – Compliance standards for carbon emissions are subject to change, which would require continuous updates to the system. Hence, the standards applicable on 06/02/2025 have been chosen.
- **User Training and Adoption** – Employees and stakeholders may require training and onboarding, which could slow implementation.
- **Product Limitations** – The system may face constraints due to different food production practices in the database, thus missing some products within the single market.
- **Time constraint** – Deadline until June 2025.

Document Overview

METHODOLOGY

The software has a database that is filled with external data sets, which are used to determine each product's carbon footprint.

The application will use a straightforward approach to calculating the carbon footprint. The result will refer to a unit (one kilogram) of a final product (with the limitations above-mentioned) generated in the food industry. The unit of measurement will be in **kilogram CO_2e of product** (carbon dioxide equivalents**) per kilogram of product**.

Sequence of actions: The user creates a profile with an authenticator and a password. Once the account is created, it will have access to a dashboard with three sub-sections.

1. Research:

In this section, the user will type the product she/he uses and find it to add to the list of consumed gods.

2. My carbon footprint

In this section, the user will see a green ball that represents they carbon footprint consumption. This is the formula: TOTAL CARBON FOOTPRINT = Σ (Weight X Emissions factor) for each product.

3. Analysis:

The user can access its historical metrics in the last subtab. The aim is for the user to select the months she/he wants to know more about and compare the trends in more detail.

The stored data must be deleted after 12 months (we cannot keep data for more than 12 months).

Process phases:

1. Requirement Analysis & Feasibility Study

- Identify key products.

2. System Architecture & Design

- Cloud-based.
- Develop a data model to support carbon footprint calculations .
- Create UI/UX designs for user accessibility.

3. Development & Implementation

- Import the database and make a calculation formula.

4. Testing & Quality Assurance (QA)

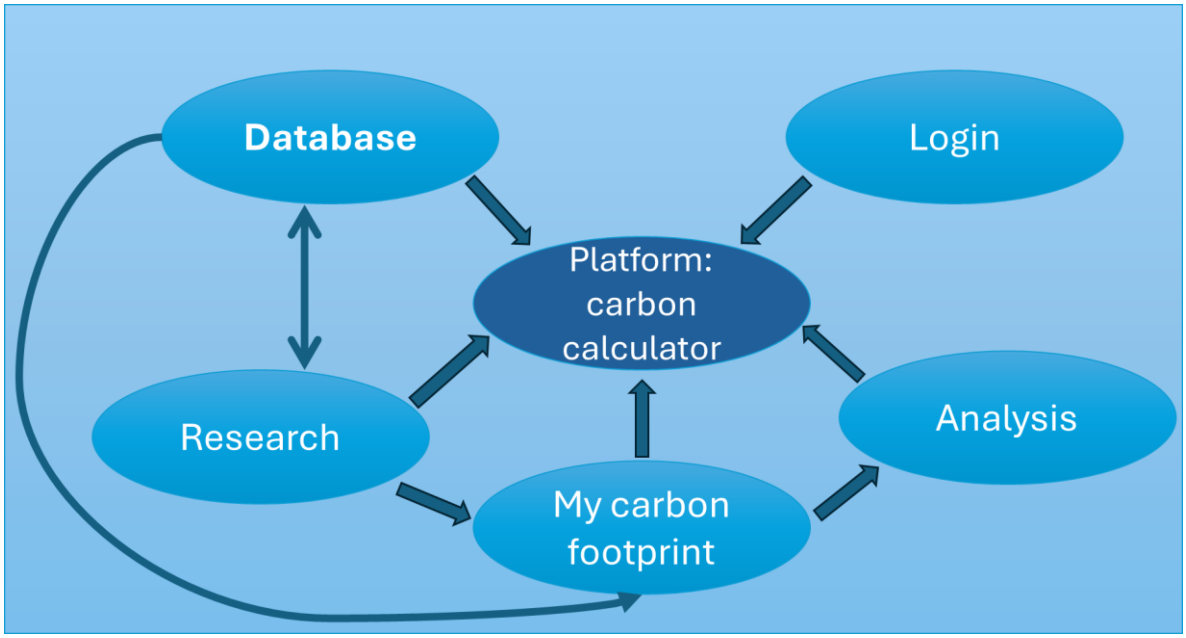
- Perform testing to assess system performance with large datasets.
- Implement user acceptance testing with industry stakeholders.
- Identify and resolve bugs.

5. Deployment & Integration

- Deploy the system in the cloud.

FUNCTIONAL REQUIREMENTS

Context



User Requirements

- Allow users to create an account and log in.
- Collect monthly data on eating and consumption habits based on 1000 user responses.
- Calculate the carbon footprint based on the data provided.
- Store user data in a secure database, based on user acceptance and GDPR compliance (for 12 consecutive months).
- Allow users to view their result history, not only with numbers but also in a visual way.
- The users are in one category: all SMEs in the food sector. The data shared is in the same category (i.e., the food products they use). Thus, the accounts must all be secured in the same way following GDPR and any other applicable law.
- Monthly compilation.
- One thousand users maximum, once a week, filling out the data on average 5 minutes a week and once a month 15 minutes additional to look at the report section.
- The data set will contain a maximum of 150 products.

Functional Requirements

1. User Authentication & Profiles

- **Sign Up/Registration**: Users will create an account by providing basic information (e.g., email, password and username).
- **Login/Logout**: Users can log into their account and log out securely.
- **Password Reset**: Users can reset their password via email if forgotten.
- **Storage of the profile data**: The profile data will be stored.

2. Food Database

- **Food Items Search**: Users can search for food items by name.
- **Detailed Food Information**: Display the carbon footprint data (e.g., carbon emissions per kilogram) for each food item.
- **User**: Types the product category.
- **Add New Food Item**: Admins can add new food items to the database, including their emission factors and relevant details.
- **Another category**: If the product the user seeks is not in the database, the software will generate a message stating, "This product is not listed; please search for an equivalent.".

3. Carbon Footprint Calculation

- **Input Quantity**: Users can input the quantity in kg of a food product they used (e.g., weight in kilograms).
- **Calculation Formula**: Based on user input and predefined emission factors, the system calculates the carbon footprint using the formula mentioned in the methodology.
- **Emission Breakdown**: Show a detailed breakdown of emissions for each monthly report (e.g., "Production: 15 kg CO_2e).
- **Display Carbon Footprint in research**: The calculated carbon footprint is displayed to the user (e.g., "1 kg of beef generates 27 kg CO_2e").
- **Display the aggregated Carbon Footprint in my carbon footprint:** in kg CO_2e.

4. Data Visualization & Reports

- **My Carbon Footprint Dashboard**: Provide a user dashboard that displays the total carbon footprint, a second with the historical trends and one research.
- **Charts/Graphs**: Use bar chart to visually represent carbon footprint data, comparisons, and trends over time.

5. Calculation for Food Items

- **Detailed Input Options**: The user fills in the name of the product .

6. Units

- **Unit Preferences**: Allow users to choose the number of kilograms.

7. Privacy and Data Security

- **GDPR Compliance**: Ensure that users' personal data is stored and processed in accordance with the latest version of the GDPR, as adopted by the EU.

Functional Requirements

Section/ Requirement ID	Requirement Definition
FR1.0.	**Sign Up/Registration**: Users will create an account by providing basic information (e.g., email, password, username).
FR1.1	**Login/Logout**: Users can log into their account and log out securely.
FR1.2	**Password Reset**: Users can reset their password via email if forgotten.
FR1.3	**Storage of the profile data**: The profile data will be stored.

Section/ Requirement ID	Requirement Definition
FR2.0	**Food Items Search**: Users can search for food items by name.
FR2.1	**Detailed Food Information**: For each food item, display the carbon footprint data (e.g., carbon emissions per kilogram).
FR2.2	**Other category**: If the product the user seeks is not in the database, the software will generate a message stating, "This product is not listed, please search for an equivalent".
FR3.0	Users can input the quantity in kg of a food product they used (e.g., weight in kilograms).
FR3.1	**Calculation Formula**: Based on user input and predefined emission factors, the system calculates the carbon footprint using the formula mentioned in the methodology.
FR3.2	**Display Carbon Footprint in research**: The calculated carbon footprint is displayed to the user (e.g., "1 kg of beef generates 27 kg CO_2e").
FR3.3	**Display the aggregated Carbon Footprint in my carbon footprint:** in kg CO_2e .
FR4.0	**My Carbon Footprint Dashboard**: Provide a user dashboard that displays the total carbon footprint, a second with the historical trends and one research.
FR4.1	**Charts/Graphs**: Use bar chart to visually represent carbon footprint data, comparisons, and trends over time.
FR5.0	**Detailed Input Options**: The user fills in the name of the product.
FR6.0	**Unit Preferences**: Allow users to choose the number of kilograms.
FR7.0	**GDPR Compliance**: Ensure that users' data is stored and processed in compliance with the latest version of GDPR, as adopted by the EU.

OTHER REQUIREMENTS

The data should be as close to accurate as possible, and the main aim is to help the consumer understand their footprint and act if they want to.

Interface Requirements

- Log in interface (transition status).
- The platform with the three subtabs from the dashboard is the active interface (active status).

Hardware Interfaces

Any computer with Windows and a research motor shall have access to the platform login page.

Non-Functional Requirements

1. Performance Requirements

- **Response Time**: The application should calculate and display the carbon footprint in under 7 seconds over a 4G connection (equivalents standards).
- **Throughput**: The system should be able to handle multiple users simultaneously (a maximum

of 1000—one thousand) without significant degradation in performance, particularly when calculating emissions or comparing multiple food items within a 4G connection quality.

- **Latency**: Requests for food item data, emission factors, and carbon footprint calculations should have minimal latency (ideally under 1 second for most operations) within a 4G connexion quality.
- **Load Time**: The web application should load in under 5 seconds on 4G connection (equivalents standards), and critical resources should load in parallel to avoid delays.

2. Availability & Reliability

- **Uptime**: The application should have high uptime, with an expected availability of at least 85%.
- **Fault Tolerance**: The system could be placed on hold if any problems arrive.
- **Data Integrity**: The system should ensure that user data and calculations remain accurate and consistent, especially in the case of unexpected failures.

3. Security Requirements

- **Authentication & Authorization**: Users should only be able to access their data and not others through a login.
- **GDPR Compliance**: Ensure that the application complies with privacy laws regarding how user data is stored and processed, particularly GDPR or other data protection regulations.

4. Usability Requirements

- **Intuitive User Interface (UI)**: The application should have a clean and easy-to-navigate interface that allows users to calculate and visualize their carbon footprints with minimal effort.
- **Responsive Design**: The application should be fully responsive and work seamlessly across the following devices: desktops.
- **Accessibility**: Ensure the application is accessible to users with disabilities by meeting WCAG (Web Content Accessibility Guidelines) standards.
- **User Documentation**: Provide an email address that users can contact for assistance regarding the login interface.

5. Maintainability

- **Code Quality**: The codebase will be modular, well-documented, and follow best practices to facilitate future updates and feature additions.
- **Error Logging**: Implement detailed logging of errors and exceptions to aid in debugging and monitoring the system. Logs should be easily accessible for developers and system administrators.
- **Continuous Integration/Continuous Deployment (CI/CD)**: Implement a CI/CD pipeline to automate testing, building, and deployment processes, ensuring smooth updates without affecting user experience.
- **Monitoring and Alerting**: Set up monitoring tools (e.g., New Relic, Datadog) to track the application's health, including performance, uptime, and error rates. Alerts should be set up for critical issues (e.g., server downtime).

Security and Privacy

State the consequences of the following breaches of security in the subject application:

a) Loss or corruption of data: A message shall appear on the platform to excuse the loss or error. Before starting, the user will be informed that the software is only indicative and that no responsibility will be taken by the owner in case of errors.

b) Disclosure of personal sensitive information: The software owner will do all it can to avoid disclosing data, and if data has been disclosed, it will inform the user and try to solve the situation.

c) Software corruption or the introduction of malware, such as viruses: The software owner will do all it can to secure the software, and if there is a problem, it will inform the user and try to solve the situation.

The Security Section describes the need to control access to the data, including who may view and alter application data.

Data Retention

The system shall offer storage for the last 12 consecutive months if the user consents.

Error Handling

If an error occurs, the software owner will receive a mail alert and act within a reasonable time to repair the error.

Conventions/Standards

Window standards shall be followed.

Risk Analysis

Scope

- Ambiguity: misunderstanding and reworks.
- Creep: uncontrolled changes or additions to the project without adjustments of delays and resources.

Changing database standard → mitigation: chose one reference date for the standard in time.

Forgotten things in the food chain → mitigation: We won't address them and mention them in the software's general conditions.

Human Resources

- Availability: turnover, burnout, external teams.
- Skill gaps: missing expertise.
- Communication: cultural and language barriers.
- Motivation.
- Fixed number of persons and no way to have extra members → mitigation: reducing the scope of the project.
- Communication and low skills → mitigation: regular meetings to make sure we all work on the same and share know-how.
- External reach to database → mitigation: putting it in the first tasks to make sure if there is a

problem that we have the time to react.

- Communication and putting pieces together → mitigation: use GitHub (or similar) and engage us to always describe in detail why we coded a part and what it is supposed to do and be linked to so that we can all understand each other's work.

Quality

- Defects and bugs.
- Technical debt: poor development quality.
- Insufficient testing.
- Insufficient testing → mitigation: use third-party free testing available on the internet.

Analysis

Functional Requirements:

Analyse every requirement and provide solutions:

Section/ Requirement ID	Requirement Definition
FR1.0.	**Sign Up/Registration**: Users will create an account by providing basic information (e.g., email, password, username). On the entrance page, there is a login button and one register. If you push on the register, make appear a form requesting: 1. Name SME. 2. Email. 3. Password.
FR1.1	**Login/Logout**: Users can log into their account and log out securely. They create an account for that Name and keep further data registered under that account. If pushed on "log in", generate two fields, one called Name and one password. If the Name and password match one account, open the dashboard.
FR1.2	**Password Reset**: Users can reset their password via email if forgotten. On the login page, under the name and password field, write in the hyperlink "password forgotten." If clicked on, send a mail to reset the password to the address under the name. The mail generates a hyperlink that empties the password field in the account, redirects the user to the registration, and enables restarting while keeping the key to the account.
FR1.3	**Storage of the profile data**: The profile data will be stored in the PR's cloud service.
FR2.0	**Food Items Search**: Users can search for food items by name. Once logged in, the user has three tabs, one of which is the search tab. In the search tab, there is one field named "add a product." When the user clicks on it, he/she can start typing the purchased product. The software will look into the database and create products starting with the same letters under the research field in alphabetical order. The user can click on the right product. Then, a field with "kg" will appear, and the user will have to add the number of kg bought. The software will generate a window with "X kg of selected product" and a button to add or change. If add is selected, the information is sent to my carbon footprint and

	added to the monthly calculation.
FR 2.1	**Detailed Food Information**: In the report section, display the carbon footprint data (e.g., carbon emissions per kilogram) for each food item.
FR 2.2	**Other category**: If the product the user seeks is not in the database, the software will generate a message stating, “This product is not listed, please search for an equivalent”.
FR 3.0	Users can input the quantity of a food product they use in kg. The input is numerical and is specified as kilograms. The UI should include an INPUT FIELD where users can type the weight of the food product. The field will be labelled “Enter the quantity in Kg.” The input format will be FLOAT type and positive. It will be limited to 1 ten, 1 one, and 1 decimal place. **Optional:** (Validation: ensure the input is a valid numeric value). **Unit display:** automatically displays the unit “Kg” next to the input field, so the users know the unit of measurement. **Storage:** store the input as a numeric value in the backend.
FR 3.1	**Calculation Formula**: Based on user input and predefined emission factors, the system calculates the carbon footprint using the formula mentioned in the methodology. This is: TOTAL CARBON FOOTPRINT = Σ (Weight X Emissions factor) for each product.
FR 3.2	**Display Carbon Footprint in research**: The calculated carbon footprint is displayed to the user (e.g., "1 kg of beef generates 27 kg CO_2e"). **User input:** The user enters the quantity of the product they consume (e.g., 1 kg of beef). **Calculation:** For the quantity entered of the selected product, calculate the total carbon footprint using the emission factors for that product. **Display the carbon footprint:** After the calculation, display the carbon footprint in the format of a circle with the value of the total carbon footprint.
FR 3.3	**Display the aggregated Carbon Footprint in my carbon footprint:** in kg CO_2e **User Input**: The user will input multiple products and their quantities throughout the session. Calculate each product's carbon footprint based on the emission factors and the quantity the user enters. **Accumulate Total Carbon Footprint:** After calculating the carbon footprint for each product, it is accumulated to calculate the total carbon footprint for all products entered. **Display the Aggregated Carbon Footprint:** Once the user finishes entering products, the system will display the total (aggregated) carbon footprint for all the products.
FR 4.0	**Charts/Graphs**: The total carbon footprint will be represented in a circle. Bar chart will be used to represent carbon footprints for different months. The X-axis will be the date. The y-axis will represent the range of carbon footprint emissions The bar will have the month and the emission's value.
FR 5.0	**Detailed Input Options**: The user will provide the quantity of the ingredient (kg).
FR 6.0	**Unit Preferences**: A predefined default option in Kilograms for easy use and consistent calculations. The system should allow users to choose the number of kilograms when entering data about food products.
FR 7.0	**GDPR Compliance**: Ensure that users' personal data is stored and processed in compliance with the latest version of GDPR, as adopted by the European Parliament. The system must include an explicit consent notice during registration, informing users how their data will be used and stored for 1 year. There must be functionality for users to request the deletion of their personal data (right to be forgotten) easily and quickly.

Statuses Diagram

Non-functional Requirements

1- **Performance Requirements**

- Response Time: The application should calculate and display the carbon footprint in under 7 seconds over a 4G connection (equivalent standards)

Analysis:

1. Optimising the calculation process: The carbon footprint calculation is simple, so the code's algorithm will naturally be fast. Ensure the application avoids unnecessary computation or complex operations. It should perform multiplications based on user input, which should take very little time.

2. Server-side optimisation: If the application fetches data from a remote server (e.g., a database of products), make sure the server query times are minimised. Use **caching** for commonly used data (e.g., emission factors). Optimise **database performance** (e.g., indexing, efficient lookups).

3. Client-side optimisation: Minimise the processing time on the browser's client side. Use efficient data handling techniques in the front end.

4. Network optimisation: If the application requires network requests (e.g., fetching data), ensure the API response times are low and that network calls are optimised (e.g., using gzip compression, minimising payload size). Asynchronous operations can be used to handle network requests without blocking the UI, ensuring that the user can interact with the system while waiting for results (example: If using a web application, consider updating the results dynamically (e.g., after each input) without reloading the page).

5. Testing and monitoring: Load and stress testing should be conducted to ensure that the application can handle multiple users or requests and still stay under the 7-second threshold. Response times in production should be monitored to identify and address performance bottlenecks quickly.

- **Throughput**: The system should be able to handle multiple users simultaneously (a maximum of 1000—one thousand) without significant degradation in performance, particularly when calculating emissions or comparing multiple food items within a 4G connection quality.

Analysis:

The **Throughput** requirement specifies that the system should handle up to 1000 simultaneous users (concurrent users) without significant degradation in performance, especially during tasks like calculating emissions or comparing multiple food items. This ensures the application can scale and perform effectively even under high load, maintaining acceptable performance for all users.

1. The Throughput requirement specifies that the system should handle up to 1000 simultaneous users without significant degradation in performance, especially during tasks like calculating emissions or comparing multiple food items. This ensures the application can scale and perform effectively even under high load, maintaining acceptable performance for all users.

2. Emission Calculations and Comparisons: While carbon footprint calculations are simple operations, the system must handle cases where multiple users perform them simultaneously.

3. Impact of 4G Network (equivalent standards): The application should work efficiently, assuming that users are operating on 4G connections, which typically offer good performance but can have occasional fluctuations in speed and latency.

4. Potential Scaling: As the number of users grows, the system's performance should scale horizontally (e.g., by adding more servers) or vertically (e.g., improving server hardware).

- **Latency**: Requests for food item data, emission factors, and carbon footprint calculations should have minimal latency (ideally under 1 second for most operations) within a 4G connexion quality.

Analysis:

1. Data caching: Frequently requested data, such as food item names and emission factors, should be cached in memory to minimise retrieval time. Use in-memory caching systems like Redis or Memcached to store emission factors, food product details, and recent results. For example, if a user repeatedly calculates the carbon footprint for beef or chicken, the system can quickly retrieve emission factors from the cache rather than querying a database or API each time.

2. Optimise database access: Efficient database queries - If the application stores food data in a relational database, ensure that the database queries are optimised for speed (Use indexes on key fields - e.g., product names or categories.

3. Frontend optimisation: Client-Side Caching - Use local storage or session storage to cache the emission factors and previous calculation results on the client side, reducing the need to repeatedly fetch the same data from the server. AJAX/Fetch API - Implement asynchronous communication between the front and back end using AJAX or Fetch API (in web applications). This way, users can continue interacting with the application while the data is fetched in the background. Pre-fetching Data - Pre-load common data, such as emission factors for popular products, before the user requests it. This could be done in the background while the user interacts with the UI.

4. API Optimisation: Ensure that our API endpoints for data retrieval (e.g., fetching food item details or emission factors) are optimised for speed.

5. Monitoring and alerting: Continuously monitor the system's response times to detect performance degradation early. Tools like New Relic, Prometheus, and Datadog can provide insights into the system's latency. Set up automated alerts if response times exceed a threshold (e.g., 1 second), allowing you to identify and fix issues proactively.

- **Load Time**: The web application should load in under 5 seconds on 4G equivalent standards, and critical resources should load in parallel to avoid delays.

Analysis:

The Load Time requirement specifies that the web application should fully load in under 5 seconds on 4G equivalent standards. Additionally, critical resources (such as scripts, stylesheets, and images) should be loaded in parallel to avoid delays in page rendering.

Initial Load Time: The application must be rendered and accessible to users in under 5 seconds.

Critical Resources: These are resources required to make the page viewable and usable, including HTML, Python, and any initial assets.

Parallel Loading: Resources not immediately needed for rendering the initial page can be loaded asynchronously or lazily to speed up the initial page load.

2- Availability & Reliability

- **Uptime**: The application should have a high uptime, with an expected availability of at least 85%

Analysis:

To meet the 85% uptime requirement, we should implement the following strategies:

1. Set up redundancy (e.g., load balancing, database replication) to ensure no single points of failure.

2. Use cloud hosting.

3. Proactively monitor system health and set up alerts to respond quickly to failures. Use real-time monitoring tools like Prometheus, Datadog, New Relic, or AWS CloudWatch to keep track of server health, application performance, and resource usage. Set up alerts to notify you when critical systems (e.g., application servers, databases, network connections) are down. For example, set up email, SMS, or Slack notifications when a server or service is down.

4. Perform regular maintenance and load tests of the application to handle high traffic efficiently. Conduct stress testing and load testing to simulate heavy user traffic and ensure that the application can handle large numbers of users without degrading performance or crashing. Tools like Apache JMeter, Locust, and Gatling can be used to generate traffic spikes and identify bottlenecks in your system.

5. Ensure disaster recovery systems are in place to recover from significant failures.

- **Fault Tolerance**: The system could be placed on hold if any problems arrive.

  If a problem or bug occurs, the system shall send a message to one of the developers, in which, if possible, a report will be sent (where the error occurs and what it is). Legally, no fault shall be held upon the software developers.

- **Data Integrity**: The system should ensure that user data and calculations remain accurate and consistent, especially in the case of unexpected failures. However, if some data is missing due to unexpected failures, the software shall inform the user about it.

4. Security Requirements

- **Authentication & Authorisation**: Users should only be able to access their data and not others through a login. The software should only allow access to the data related to the account that was opened through the login. Hence, the data shall be held separately in the cloud, and no bridge between accounts will be made.
- **GDPR Compliance**: Ensure that the application complies with privacy laws regarding how user data is stored and processed, particularly GDPR or other data protection regulations. The software shall request consent to keep the data for over one year when the user creates an account.

5. Usability Requirements

- **Intuitive User Interface (UI)**: The application should have a clean and easy-to-navigate interface that allows users to calculate and visualise their carbon footprints with minimal effort. This is achieved using a few words and three tabs (search, my carbon footprint, and reports). Each tab shall use the same design and writing format, and the visualisation of both my carbon footprint shall be easy to understand for the user.
- **Responsive Design**: The application should be fully responsive and work seamlessly on a desktop. The software code should be formatted for the desktop.
- **Accessibility:** The user can access the website from Monday through Saturday for 24 hours. However, the website will not be available on Sunday and will be down for maintenance. This information will be present on the website's homepage.
- **Profile Documentation**: It will contain the historic data Username, email ID of the user and carbon footprint details.

6. Maintainability

- **Code Quality**: The codebase will be modular, well-documented, and follow best practices to facilitate future updates and feature additions.
- **Error Logging**: Implement detailed logging of errors and exceptions to aid in debugging and monitoring the system. Logs should be easily accessible for developers and system administrators.
- **Continuous Integration/Continuous Deployment (CI/CD):** Implement a CI/CD pipeline to

automate testing, building, and deployment processes, ensuring smooth updates without affecting user experience.

- **Monitoring and Alerting:** Set up monitoring tools (e.g., New Relic, Datadog) to track the application's health, including performance, uptime, and error rates. Alerts should be set up for critical issues (e.g., server downtime).

  Security Monitoring: Monitor unauthorized access attempts or suspicious activity, such as multiple failed login attempts; Set up alerts for potential security breaches, such as data leaks or access to sensitive information.

  Set up automated alerts to notify development teams of critical issues, such as System availability below 85%, frequent endpoint or calculation errors, and resource usage (CPU, memory, disk).

  7. Portability

- **Responsive Design:** Ensure your user interface (UI) is responsive and adapts to different screen resolutions and window sizes.
- **Portability Documentation:** Include a section in your user documentation explaining minimum system requirements and how to ensure the best Windows experience. Provide clear instructions for troubleshooting common compatibility issues.

Glossary

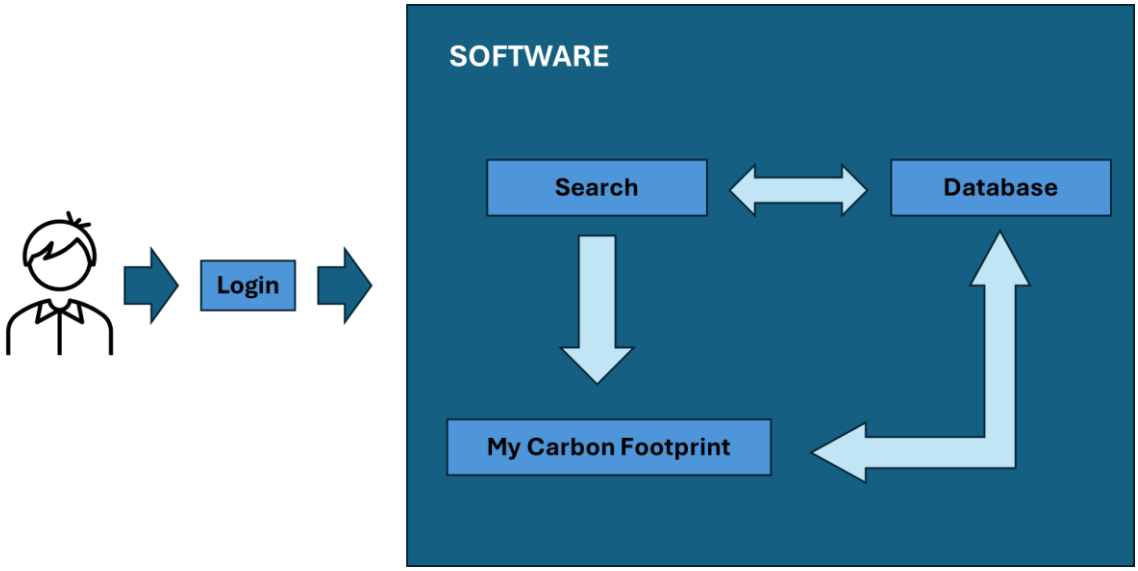
1. **Carbon Footprint**: The carbon footprint of a product is the quantity of greenhouse gases (GHG), expressed in carbon dioxide equivalent (CO2e), emitted across the supply chain for a single unit of that product. (FAO, 2019) .
2. **Greenhouse Gas**- Greenhouse gases are those gaseous constituents of the atmosphere, both natural and anthropogenic, that absorb and emit radiation at specific wavelengths within the spectrum of infrared radiation emitted by the Earth's surface, the atmosphere, and clouds. This property causes the greenhouse effect. Water vapour (H2O), carbon dioxide (CO2), nitrous oxide (N2O), methane (CH4), and ozone (O3) are the primary greenhouse gases in the Earth's atmosphere (IPCC,2018)**.**
3. **Range Limit:** Food products' carbon footprint range limit varies significantly based on factors such as production methods, transportation, and processing. Here is a general breakdown of greenhouse gas (GHG) emissions, measured in kilograms of CO_2-equivalent (kg CO_2e) per kilogram of food product (Wikipedia)

- Very Low (0.1 – 0.5 kg CO_2e/kg).
- Low (0.5 – 2 kg CO_2e/kg).
- Medium (2 – 5 kg CO_2e/kg).
- High (5 – 15 kg CO_2e/kg).
- Very High (15+ kg CO_2e/kg).

4. **Product**: The product in this document refers to the final food product or a meal.
5. **Ingredient**: The ingredient in this document refers to the ingredients used in making the food product.

GRAPHS:

Use Case



Flow Charts:

Flowcharts Process:

1. Start- Opening the website. The user can also create a new account or reset password.
2. User Login- Log in with your e-mail ID and password.
3. User Inputs Details
    a) Product name/ Ingredients from the dropbox.
    b) Weight/Quantity (Kg) of the ingredient/product.
4. Fetch Carbon Footprint Data
    a) Database lookup for emission factors (per ingredient).
5. Calculate Carbon Emission
    a) Apply emission factors (kg CO_2e per unit).
    b) Multiply the emission factor by the quantity of product.
    c) Display the final emission result of the product.
6. Additional Features
    a) A bar chart showing the total emission value of all ingredients in each month.
    b) The final carbon footprint emission value will be displayed after calculation.
7. Generate Report
    a) Display total carbon footprint (kg CO_2e).

8. End.

Appendix

Ingredient/Product		Carbon footprint (Kg CO_2e)
1	All Spice	1.1
2	Almond Extract	2.3
3	Baking Powder	0.5–1
4	Basil	1–2
5	Bay Leaves	1–2
6	Beet Sugar	1–2
7	Sodium Bi Carbonate	0.5–1
8	Black Pepper	1–3
9	Breakfast Cereal	1–3
10	Brown Sugar	1–2
11	Cacao Powder	2–5
12	Cassava Starch	0.5–1
13	Caynne Pepper	1–3
14	Chilli Flakes	1–3
15	Chilli Powder	1–3
16	Cinnamon	1–3
17	Cloves	1–3
18	Coconut Sugar	1–2
19	Coffee Beans	15–20
20	Coriander	1–2
21	Corn Flour	0.5–1
22	Cumin	1–3
23	Dill	1–2
24	Wheat Flour	1–2
25	Food Colouring	1–3
26	Garam Masala	1–3
27	Granola	1–3
28	Herbs And Spices	1–3
29	Hot Chocolate Powder	2–5
30	Icing Sugar	1–2
31	Mint	1–2
32	Miso Paste	1–3
33	Mollasses	1–2
34	Museli	1–3
35	Nutmeg	1–3
36	Oat Flakes	0.5–1
37	Oregano	1–2
38	Papparika	1–3
39	Parsley	1–2

40	Plain Flour	1–2
41	Porridge	1–2
42	Rosemery	1–2
43	Rye Flour	1–2
44	Sage	1–2
45	Salt	0.5–1
46	Sugar	1–2
47	Tapioca Starch	0.5–1
48	Thyme	1–2
49	Turmeric	1–3
50	Vannila	1–3
51	Vannilla Extract	2–4
52	Vegetable Fat	2–4
53	White Sugar	1–2
54	Yeast	0.5–1
55	Chilli Paste	1–3
56	Almond Milk	0.5–1
57	Butter	10–15
58	Butter Milk	1–3
59	Cheddar	8–13
60	Cheese	8.7
61	Clotted Cream	10–15
62	Concentrated Milk	1–3
63	Cottage Cheese	5–8
64	Cream Cheese	5–8
65	Double Cream	10–15
66	Emmental	8–13
67	Feta	5–8
68	Goat Chesese	5–8
69	Ice Cream	2–5
70	Margarine	2–4
71	Milk	1–3
72	Mozzerella	5–8
73	Oatmilk	0.5–1
74	Parmesan	8–13
75	Plant Based Cream	1–2
76	Plant Based Milk	0.5–1
77	Single Cream	10–15
78	Sour Cream	5–8
79	Soy Milk	0.5–1
80	Vegan Cheese	1–3
81	Whipping Cream	10–15
82	Yogurt	1–3

83	Plant Based Yogurt	0.5–1
84	Eggs	2–4
85	Americano	0.7
86	Apple Juice	0.5–1
87	Beer	0.7
88	Beer Bottle	0.8
89	Beer Can	0.6
90	Black Instant Coffee	0.7
91	Black Tea	0.6
92	Bottled Water	0.3
93	Brandy	2–4
94	Capuchino	1.6
95	Cider	0.7
96	Cider Bottle	0.8
97	Cider Can	0.6
98	Espresso	0.7
99	Flat White	1.6
100	Fruit Juice	0.5–1
101	Fruit Tea	0.6
102	Grapefruit Juice	0.5–1
103	Green Tea	0.6
104	Herbal Tea	0.6
105	Hot Chocolate Powder	2–5
106	Hot Chocolate With Milk	2.7
107	Hot Chocolate With Water	0.08
108	Instant Coffee	1.4
109	Instant Coffee With Milk	1.6
110	Jasmine Tea	0.6
111	Latte	1.6
112	Lemon Juice	0.5–1
113	Lime Juice	0.5–1
114	Mocha	1.6
115	Orange Juice	0.5–1
116	Other Spirits	2–4
117	Pineapple Juice	0.5–1
118	Rum	2–4
119	Soft Drink	0.6
120	Tequila	2–4
121	Sherry	2–4
122	Vodka	2–4
123	Water	0.02
124	Whiskey	2–4
125	White Tea	0.6

126	Wine	1.3
127	Bagel	1–2
128	Baguette	1–2
129	Bread	1–2
130	Bread Crumbs	1–2
131	Brown Bread	1–2
132	Cake	2–5
133	Chocolate Cake	3–6
134	Crepe	1–2
135	Doughnut	1–3
136	Muffin	1–3
137	Fruit Cake	2–5
138	Pancake	1–2
139	Pastry	2–5
140	Pitta Bread	1–2
141	Scones	1–3
142	Plain Naan Bread	1–2
143	Sponge Cake	2–5
144	Tortilla Wrap	1–2
145	Waffle	1–3
146	White Bread	1–2
147	Fruit Pie	2–5
148	Cheesecake	3–6
149	Biscuits	1–3
150	Chocolate Cake	3 to 5
151	Chocolate Chips	2–5
152	Chocolate Snack Bar	2–5
153	Crackers	1–3
154	Fudge	2–5
155	Gluten Free Biscuits	1–3
156	Marshmellows	1–3
157	Musseli Snack Bars	1–3
158	Sweets	1–3
159	Toffee	2–5
160	Vegan Chocolate	2–5
161	Chicory	0.5–1
162	Anchioves	3–5
163	Caviar	10–20
164	Clams	2–4
165	Cod	3–5
166	Crayfish	3–5
167	Fish fingers	3–5
168	fish sticks	3–5

169	Haddock	3–5
170	Herring	3–5
171	Kipper	3–5
172	Lobster	5–10
173	Mackrel	3–5
174	Molluscs	2–4
175	Mussels	2–4
176	Oysters	2–4
177	Pollock	3–5
178	Prawn	5–10
179	Salmon	3–6
180	Sardines	3–5
181	Scallops	5–10
182	Scampi	5–10
183	Seaweed	0.5–1
184	Shrimp	5–10
185	Smoked salmon	5–10
186	Squid	3–5
187	Tuna	3–6
188	Whitebait	3–5
189	Apple	0.3–0.6
190	Apricot	0.5–1
191	Avocado	0.5–1
192	Banana	0.5–1
193	Berries	0.5–1
194	Blackberry	0.5–1
195	Blueberry	0.5–1
196	Bottled tomatoes	1–2
197	Canned mandarines	1–2
198	Canned mango	1–2
199	Canned peaches	1–2
200	Canned pear	1–2
201	Canned pineapple	1–2
202	Canned plums	1–2
203	Canned tomatoes	1–2
204	Cherries	0.5–1
205	Coconut	1–2
206	Cranberries	0.5–1
207	Dates	0.5–1
208	Dried apple	1–2
209	Dried mango	1–2
210	Dried pear	1–2
211	Dried pineapple	1–2

212	Dried plums	1–2
213	Frozen apple	0.5–1
214	Frozen plums	0.5–1
215	Frozen pineapple	0.5–1
216	Frozen mango	0.5–1
217	Frozen blackberry	0.5–1
218	Frozen Blueberry	0.5–1
219	Frozen cherry	0.5–1
220	Frozen cranberry	0.5–1
221	Frozen peaches	0.5–1
222	Frozen rasberry	0.5–1
223	Frozen strawberry	0.5–1
224	Grapefruit	0.5–1
225	Grapes	0.5–1
226	Kiwi	0.5–1
227	Lemon	0.5–1
228	Lime	0.5–1
229	Mandarines	0.5–1
230	Mango	0.5–1
231	Melons	0.5–1
232	Nectarines	0.5–1
233	Olives	1–2
234	Oranges	0.5–1
235	Papaya	0.5–1
236	Peaches	0.5–1
237	Pear	0.5–1
238	Pineapple	0.5–1
239	Plum	0.5–1
240	Strawberry	0.5–1
241	Raisins	1–2
242	Rasberry	0.5–1
243	Tomato	0.5–1
244	Barley	0.5–1
245	Basmati rice	1–2
246	Black Beans	0.5–1
247	Brownrice	1–2
248	Buckwheat	0.5–1
249	Bulgar	0.5–1
250	Chickpeas	0.5–1
251	Corn	0.5–1
252	Couscous	0.5–1
253	Faba Beans	0.5–1
254	Frozen pulses	0.5–1

255	Haricot Beans	0.5–1
256	Jasmine rice	1–2
257	Kidney beans	0.5–1
258	Long-grained rice	1–2
259	Lentils	0.5–1
260	Macaroni	1–2
261	Maize	0.5–1
262	Mixed grain	0.5–1
263	Noodles	1–2
264	Oats	0.5–1
265	Pasta	1–2
266	Polenta	0.5–1
267	Quinoa	1–2
268	Rice	1–2
269	Soybeans	0.5–1
270	Sphaggetti	1–2
271	Sushi rice	1–2
272	Wheat	0.5–1
273	White rice	1–2
274	Wholemeal pasta	1–2
275	Bacon	5–10
276	Beef	27–60
277	Chicken	4–6
278	Cocktail sausage	5–10
279	Duck	5–10
280	Falafal	1–2
281	Goat	5–10
282	Goose	5–10
283	Guinea fowl	5–10
284	Ham	5–10
285	Lamb	20–40
286	Pork	5–12
287	Rabbit	5–10
288	Steak	27–60
289	Tofu	1–2
290	Turkey	4–6
291	Veal	10–20
292	Pork Sausage	5–10
293	Alfalfa	0.5–1
294	Amond	2–4
295	Cashew	2–4
296	Chestnut	0.5–1
297	Chia seeds	1–2

298	Fennel seeds	1–2
299	Mixed nuts	2–4
300	Mustard seeds	1–2
301	Peanuts	1–2
302	Pecans	2–4
303	Pistachio	2–4
304	Pumpkin seeds	1–2
305	Sesame seeds	1–2
306	Sunflower seeds	1–2
307	Walnut	2–4
308	Coconut oil	2–4
309	Mustard oil	2–4
310	olive oil	3–6
311	Sunflower oil	2–4
312	Maize oil	2–4
313	Groundnut oil	2–4
314	Palm oil	3–6
315	Peanut oil	2–4
316	Rapeseed oil	2–4
317	Sesame oil	2–4
318	Soybean oil	2–4
319	Potato	0.2–0.5
320	Sweet potato	0.2–0.5
321	Baked beans	1–2
322	Chips	1–2
323	Corn crisp	1–2
324	Custard	1–2
325	Pizza	2–5
326	Frui salad	0.5–1
327	Tomato paste	1–2
328	Apple sauce	0.5–1
329	Barbeque sauce	1–2
330	Beef gravy	1–2
331	Beef stock	1–2
332	Bolognese sauce	1–2
333	Chicken gravy	1–2
334	Chicken stock	1–2
335	Fish sauce	1–2
336	Hummus	1–2
337	Mayonaise	1–2
338	Mustard	1–2
339	Oyster sauce	1–2
340	Pesto	1–2

341	Salad cream	1–2
342	Salad dressing	1–2
343	Soy sauce	1–2
344	Sweet chilli sauce	1–2
345	Tahini	1–2
346	Tomato ketchup	1–2
347	Tomato sauce	1–2
348	Vegetable stock	1–2
349	Vinegar	1–2
350	Salsa	1–2
351	Coleslaw	1–2
352	Sriracha	1–2
353	Chocolate spread	2–5
354	Golden syrup	1–2
355	Hazelnut paste	2–4
356	Hazelnut spread	2–4
357	Honey	1–2
358	Jam	1–2
359	Maple syrup	1–2
360	Marmalade	1–2
361	Peanut butter	1–2
362	Artichoke	0.5–1
363	Asparagus	0.5–1
364	Aubergine	0.5–1
365	Beetroot	0.5–1
366	Brocolli	0.5–1
367	Brussels sprout	0.5–1
368	Cabbage	0.5–1
369	Canned asparagus	1–2
370	Canned carrot	1–2
371	Canned beetroot	1–2
372	Canned green beans	1–2
373	Canned sweet corn	1–2
374	Carrot	0.5–1
375	Cauliflower	0.5–1
376	Celery	0.5–1
377	Chilli pepper	0.5–1
378	Cucumber	0.5–1
379	Canned peas	1–2
380	Dried peas	0.5–1
381	Frozen asparagus	0.5–1
382	Frozen brussels sprout	0.5–1
383	Frozen carrot	0.5–1

384	Frozen green peas	0.5–1
385	Frozen sweet corn	0.5–1
386	Ginger	0.5–1
387	Garlic	0.5–1
388	Green beans	0.5–1
389	Green peppers	0.5–1
390	Jalapeno peppers	0.5–1
391	Kale	0.5–1
392	Leek	0.5–1
393	Lettuce	0.5–1
394	Mushroom	0.5–1
395	Onion	0.5–1
396	Sweet corn	0.5–1
397	Peas	0.5–1
398	Red onions	0.5–1
399	Red peppers	0.5–1
400	Rhubarb	0.5–1
401	Salad leaves	0.5–1
402	Shallots	0.5–1
403	Spinach	0.5–1
404	Squash	0.5–1
405	Turnip	0.5–1

ERD

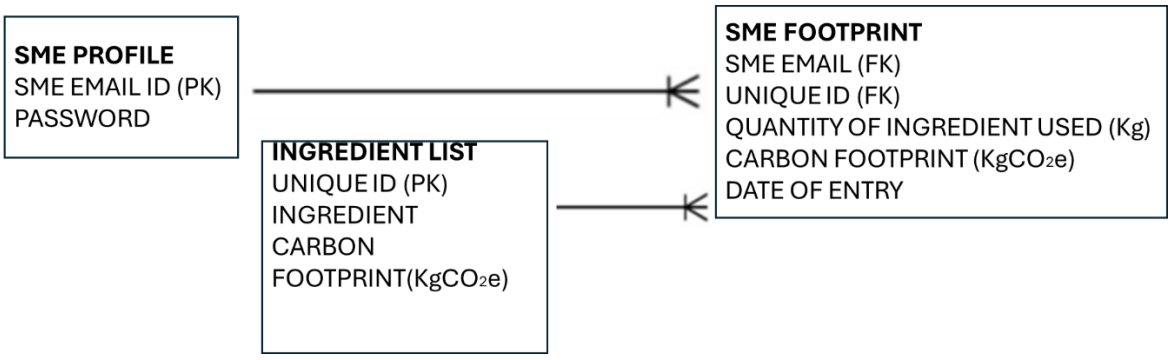


Figure 1: ERD

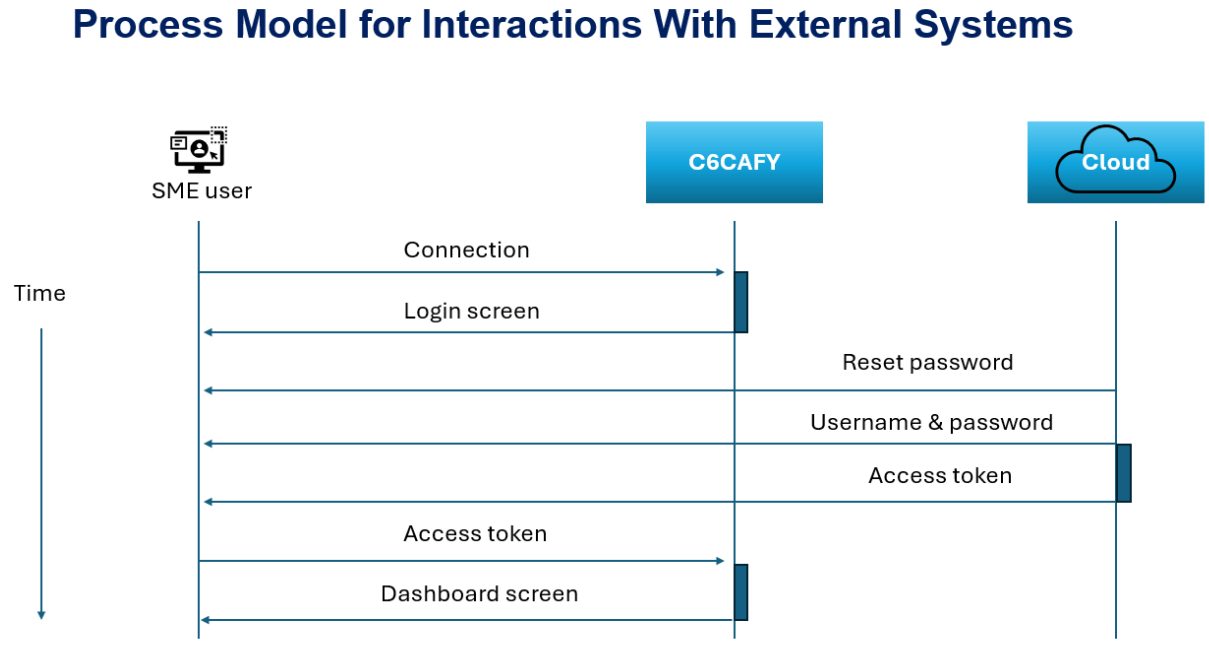


Figure 2: Process Model for Interactions With External Systems

WEB PAGES

Figure 3: Login page

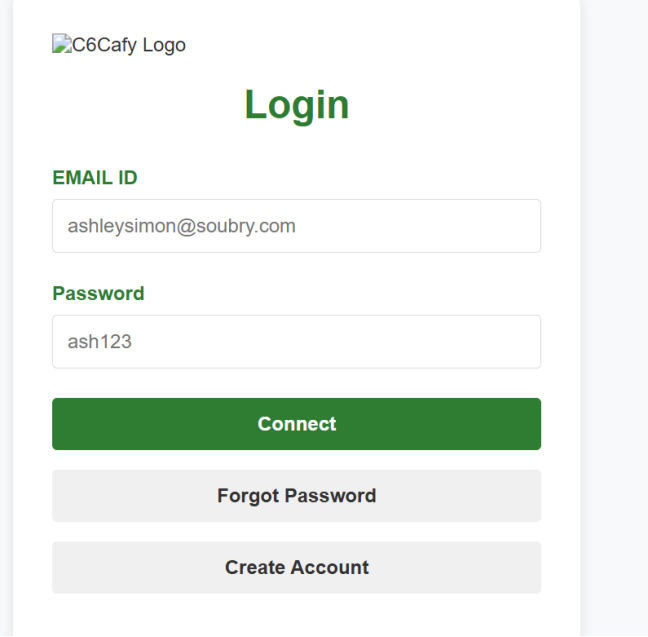


Figure 4: Carbon Calculator

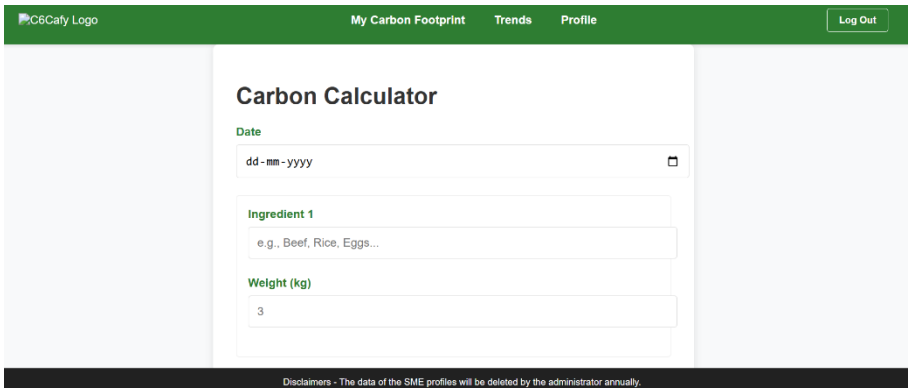


Figure 5: My Carbon Footprint

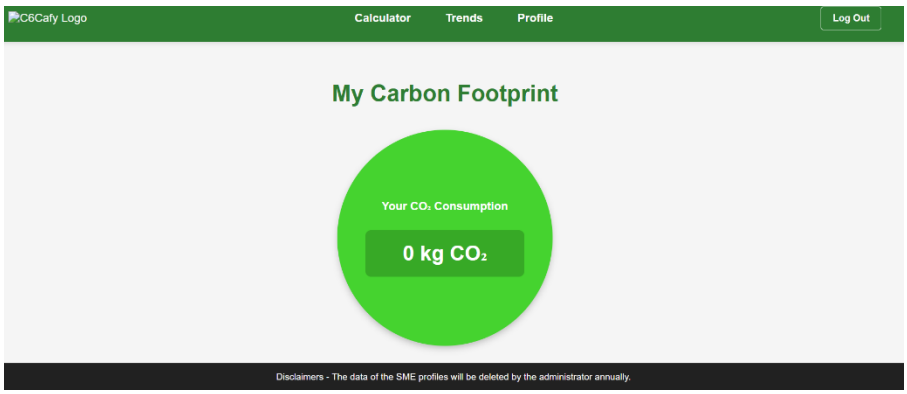


Figure 6: SME Profile

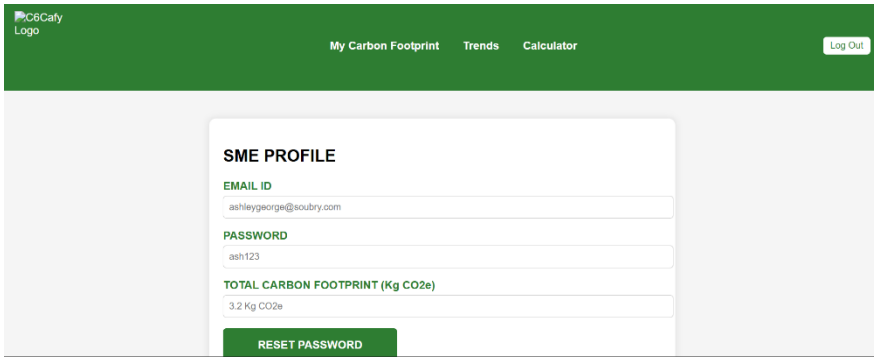


Figure 7: SME Trends

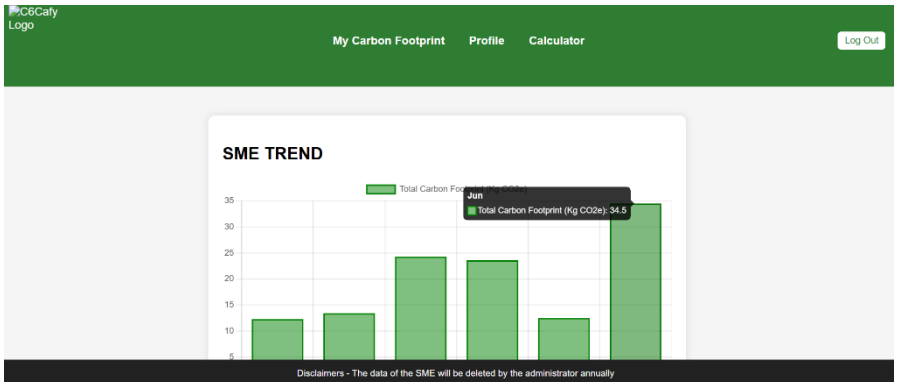


Figure 8: Create Account

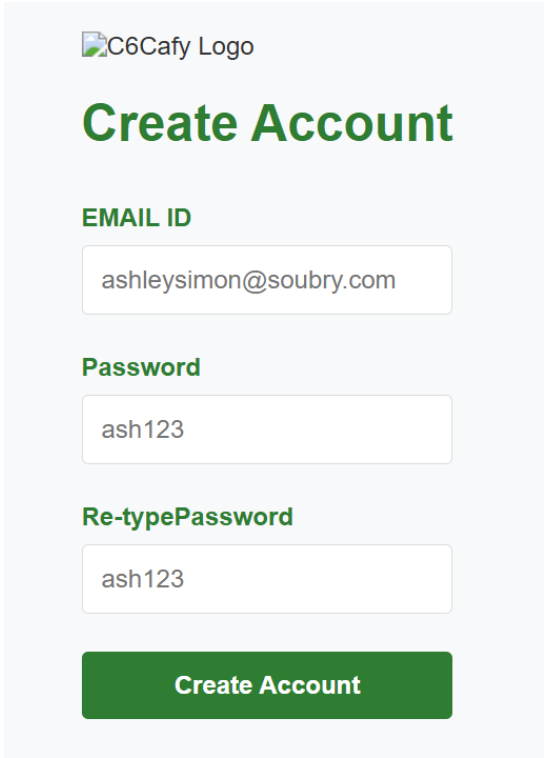


Figure 9: Reset Password