Carbon Emissions Calculator

Introduction

One of the most important things we need to understand in this day and age, when environmental consciousness is vital, is our carbon footprint. The Carbon Footprint Calculator is a user-friendly tool that allows individuals, small businesses, and industries to assess their ecological footprint. Energy use, trash management, and travel are the three primary categories that this calculator computes and surveys. Furthermore, it provides thorough evaluation and the option to create and download reports, encouraging thoughtful consideration in order to lower emissions and implement significant environmental actions. A demonstration of the calculator's operation, user interface, and logic is given in this report. It also emphasises its recommendations and possible uses.

Purpose

The purpose of this tool is to:

1. Enable individuals to calculate their annual carbon emissions.
2. Help users identify the primary sources contributing to their carbon footprint.
3. Provide actionable steps for adopting a more sustainable and eco-friendly lifestyle.

This carbon calculator is designed to be intuitive, engaging, and educational, ensuring a user-friendly experience. It empowers individuals to better understand the environmental impact of their daily habits and decisions, promoting informed choices for a greener future.

GitHub Repository

https://github.com/Khenidhruvin2001/Computer-Programming-M602B

Principal Components of Emission

The calculator breaks emissions down into three main categories: energy use, trash management, and transportation. We can make better judgements for the environment if we know how each of our everyday actions impacts carbon emissions!

1. Energy Consumption:

- The use of fuel, petrol and electricity all add to global carbon emissions.
- The carbon calculator computes by using the usual emission factor for each type of energy source and examining monthly invoices.
- Using these three energy components, the carbon calculator computes the provided equation and outputs the outcome.

2. Management of Waste:

- Methane and other greenhouse gases are released when waste is dumped in landfills.
- The calculator adjusts the waste emissions based on the recycling rate, which helps reduce these emissions.

3. User's Travel:

- The annual travel distance and vehicle fuel efficiency determine these emissions from both personal and business travel.
- Based on the user's travel habits or business travel objectives, fuel efficiency offers a precise and personalised estimate.

Enhancers of Emissions

Three different calculator types are available to users:

- **Personal:** Typical for people.
- **Business**: Expanded to accommodate operations on a larger scale.

Each type has the ability to modify emission amounts to reflect the unique requirements and actions of the user.

About the Application

The application features an intuitive and well-organized interface, designed to provide a seamless user experience. Each section is structured to guide users through the process efficiently.

1. **Personal Information**
    - Users are required to enter their name, select the specific year for which they wish to perform calculations, and choose whether the calculation is for individual or business purposes.
2. **Energy Consumption**
    - In this section, users will input their monthly spending on electricity, gas, and fuel. This data helps in assessing their overall energy usage.
3. **Waste Handling**
    - Users need to specify the amount of waste they produce each month in kilograms. They can also adjust a slider to indicate the percentage of waste that is recycled.
4. **Travel Data**
    - Here, users will provide details about their annual travel distance in kilometres and their vehicle's fuel efficiency, measured in litters per 100 kilometres.
5. **Calculation Results**
    - By clicking the "Calculate" button, users can view a detailed analysis of their emissions, categorized by energy consumption, waste management, and travel.
    - The overall emissions, along with category-specific details, will be presented for easy understanding.
6. **PDF Report Creation**
    - The application offers the option to generate a downloadable PDF report summarizing the results. This report can be saved for future reference or shared with others.
7. **History of Calculations**
    - This section keeps a record of recent calculations for quick access.
    - Users can also choose to clear their calculation history whenever needed.

This thoughtfully designed layout ensures that users can navigate the application effortlessly while efficiently managing their data and calculations.

Formula-Based Calculation Logic

The calculator computes carbon emissions using well-performing formulas.

1. **Energy Emissions:**

- Monthly Electricity Bills * 12 * (0.0005) + Monthly Natural Gas Bills * 12 * (0.0053) + Monthly Fuel Bill * 12 * (2.32) = Energy Emissions
- The unit of measurement for electricity is kilowatt-hours (kWh).

- **Gas**: Cubic meters (m^3) of natural gas consumption.
- **Fuel:** Litters of heating or vehicle fuel.
- **Multiplier**: Modifies emissions according to the type of calculator.

2. **Emissions of Waste:**

- Formula**:** (Total Waste Produced Monthly) *12* (0.57-percentage of recycling/composting)
- Recycling increases recycling rates and lowers the waste emission factor.

3. **Travel Emissions:**

- The formula for travel emissions is (total kilometres travelled annually) * (1/average fuel efficiency in L/100 km) * 2.31
- efficiency is expressed as litter per 100 km.
- For larger-scale operations, the multiplier modifies emissions.

4. **Total Emissions:**

- Energy Emissions + Waste Emissions + Travel Emissions

Creation of PDF Reports

The tool summarises the user's emissions data in a downloadable PDF report using the **ReportLab** library. Among the report's main components are:

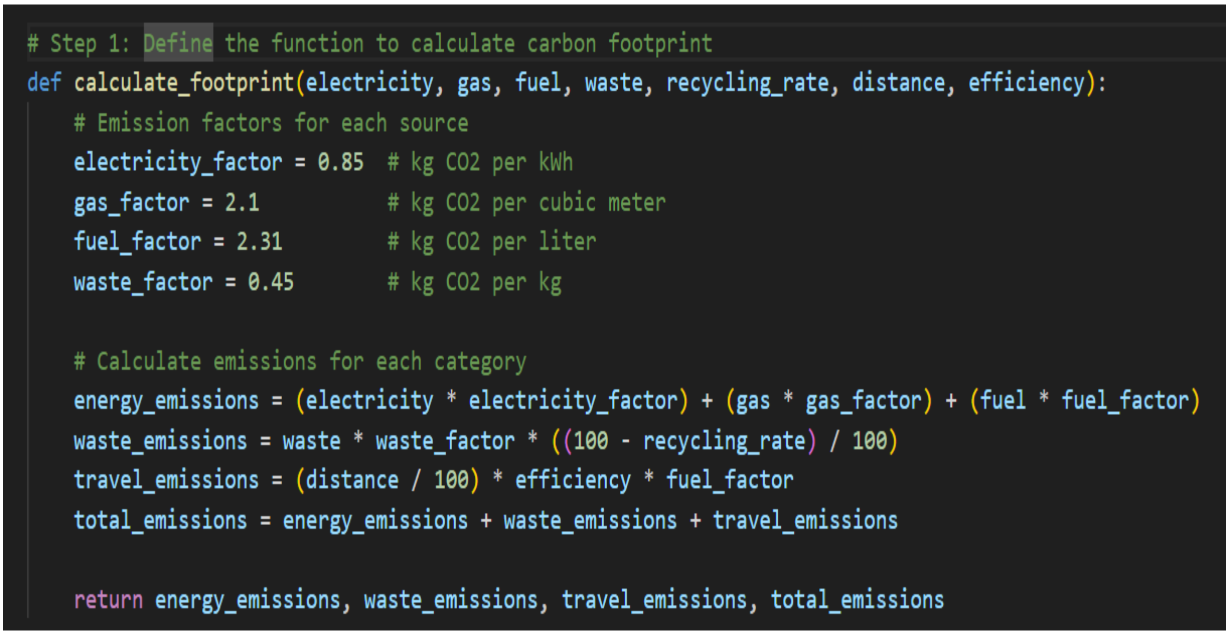
- Details about the user, including name, year, and calculator type.
- both the overall emissions and comprehensive breakdowns of emissions by category.
- a polished, well-structured format that makes sharing and presenting information simple.
- **History Management**:
- To ensure that data is retained across sessions, calculation records are kept in a JSON file called **calculation_history.json**. Features of the system include:
- The 10 most recent computations are stored.
- The most recent five calculations are shown in the sidebar for easy access.
- an option to simply click a button to erase the calculation history.

Technical Stack

Libraries Utilized**:**

1. **Streamlit**
    - Delivers an interactive and user-friendly interface for the application.
    - Supports real-time updates and file downloads**.**
2. **ReportLab**
    - Utilized for creating high-quality and professional PDF reports.
3. **JSON**
    - Enables efficient storage and retrieval of calculation history data.
4. **Datetime**
    - Adds timestamps to each history record for accurate tracking.
5. **BytesIO**
    - Facilitates efficient in-memory PDF generation and management.

Code and Explanation



```python
# Step 1: Define the function to calculate carbon footprint
def calculate_footprint(electricity, gas, fuel, waste, recycling_rate, distance, efficiency):
    # Emission factors for each source
    electricity_factor = 0.85  # kg CO2 per kWh
    gas_factor = 2.1           # kg CO2 per cubic meter
    fuel_factor = 2.31         # kg CO2 per liter
    waste_factor = 0.45        # kg CO2 per kg

    # Calculate emissions for each category
    energy_emissions = (electricity * electricity_factor) + (gas * gas_factor) + (fuel * fuel_factor)
    waste_emissions = waste * waste_factor * ((100 - recycling_rate) / 100)
    travel_emissions = (distance / 100) * efficiency * fuel_factor
    total_emissions = energy_emissions + waste_emissions + travel_emissions

    return energy_emissions, waste_emissions, travel_emissions, total_emissions
```

The first class of code is for define where all the provided data are being calculate and shows in the result.

Function calculate_footprint

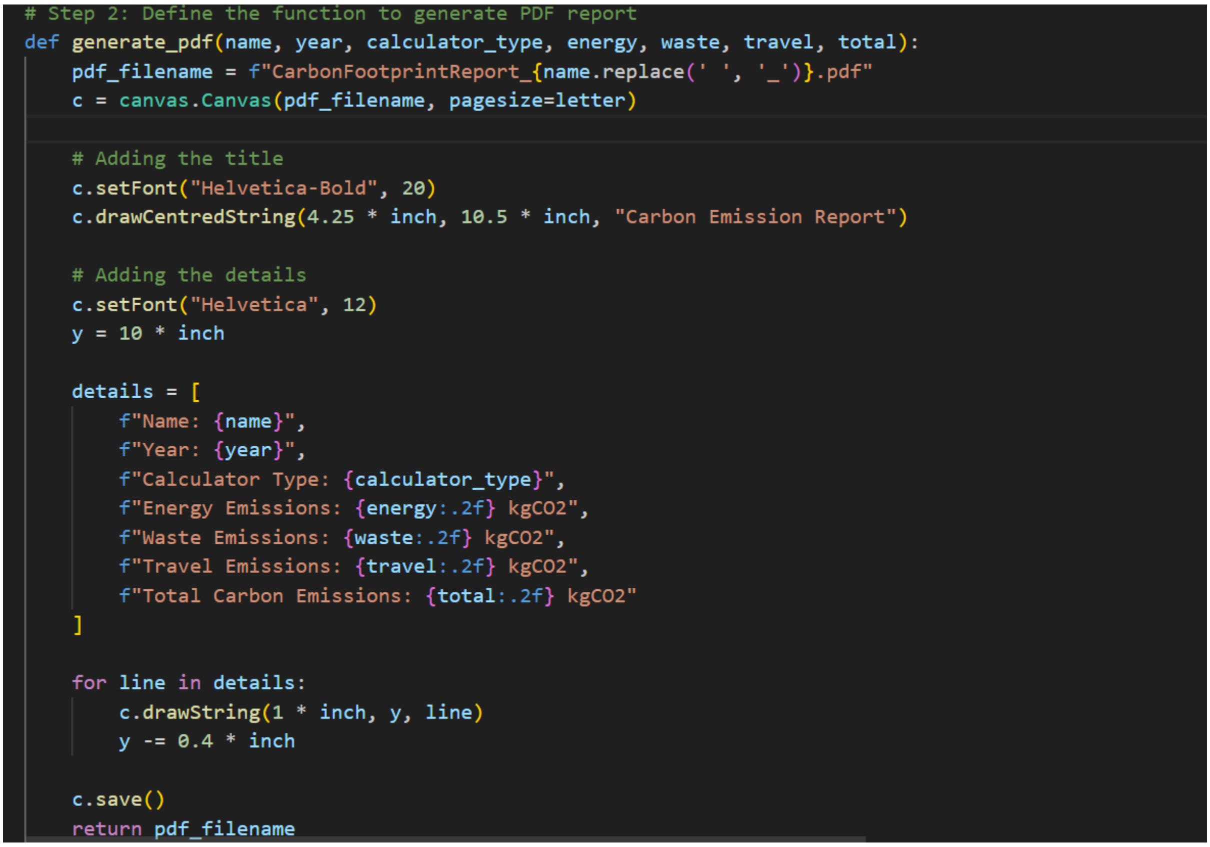
Calculating carbon emissions based on a number of inputs, including waste, recycling_rate, distance, efficiency, electricity, petrol, and fuel, is the purpose of The calculate_footprint function.

To determine the carbon emissions, it establishes emission factors for every category (such as waste, fuel, petrol, and electricity).

After that, it determines emissions for every category:
Energy_emissions: Fuel, petrol, and electricity-related emissions.
waste_emissions: Waste-related emissions after the recycling rate is taken into account.
travel_emissions: Fuel factor, efficiency, and distance-based travel emissions.
Lastly, it computes the total emissions by adding up all of these.

Energy_emissions, waste_emissions, travel_emissions, and total_emissions are the four emission values that are returned by the function.

Generate PDF



```
# Step 2: Define the function to generate PDF report
def generate_pdf(name, year, calculator_type, energy, waste, travel, total):
    pdf_filename = f"CarbonFootprintReport_{name.replace(' ', '_')}.pdf"
    c = canvas.Canvas(pdf_filename, pagesize=letter)

    # Adding the title
    c.setFont("Helvetica-Bold", 20)
    c.drawCentredString(4.25 * inch, 10.5 * inch, "Carbon Emission Report")

    # Adding the details
    c.setFont("Helvetica", 12)
    y = 10 * inch

    details = [
        f"Name: {name}",
        f"Year: {year}",
        f"Calculator Type: {calculator_type}",
        f"Energy Emissions: {energy:.2f} kgCO2",
        f"Waste Emissions: {waste:.2f} kgCO2",
        f"Travel Emissions: {travel:.2f} kgCO2",
        f"Total Carbon Emissions: {total:.2f} kgCO2"
    ]

    for line in details:
        c.drawString(1 * inch, y, line)
        y -= 0.4 * inch

    c.save()
    return pdf_filename
```

Aims of this code

Calculating carbon emissions based on a number of inputs, including waste, recycling_rate, distance, efficiency, electricity, petrol, and fuel, is the purpose of the calculate_footprint function.

Sections on Code

1.Setup for PDF Files

- The user's name is used to dynamically generate the PDF filename.
- The report creation process begins with the initialisation of the PDF canvas.

2. Title of Display

- The title "Carbon Emission Report" is added to the PDF using a bold font.

3.Section on Details

- A bold font is used to add the title "Carbon Emission Report" to the PDF.
- Name, Year, Calculator Type, and other user-inputted data are added.
- Formatted to two decimal places, the computed emission values for Energy, Waste, Travel, and Total Carbon Emissions are shown.

4.Procedure for Gathering Data

- The user is prompted by the system to enter particular values for every category of emissions.
- After the user clicks the "Calculate Emission" button, the system computes the emissions.

5. Data Writing to PDF

- With appropriate formatting and spacing, the gathered and computed data are entered into the PDF.

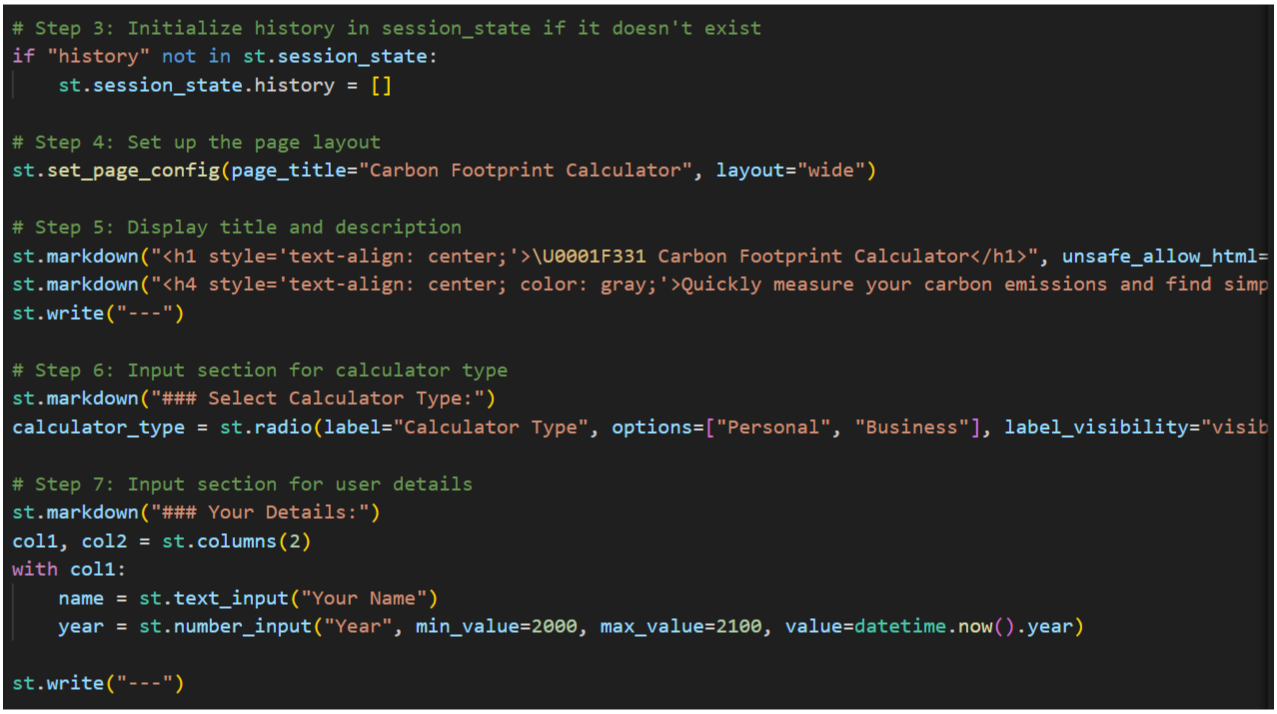
6. Retaining and Saving

- The user can view the generated report after the PDF is saved and the filename is returned.

The Way It Operates

- Data about travel distance, waste generation, and energy consumption are entered by users.
- When the calculate button is clicked, the emissions are calculated after the system has processed the inputs.
- All of the computed emissions are then presented in a clean PDF report.

The user's input section, calculator type, and history



```
# Step 3: Initialize history in session_state if it doesn't exist
if "history" not in st.session_state:
    st.session_state.history = []

# Step 4: Set up the page layout
st.set_page_config(page_title="Carbon Footprint Calculator", layout="wide")

# Step 5: Display title and description
st.markdown("<h1 style='text-align: center;'>\U0001F331 Carbon Footprint Calculator</h1>", unsafe_allow_html=
st.markdown("<h4 style='text-align: center; color: gray;'>Quickly measure your carbon emissions and find simp
st.write("---")

# Step 6: Input section for calculator type
st.markdown("### Select Calculator Type:")
calculator_type = st.radio(label="Calculator Type", options=["Personal", "Business"], label_visibility="visib

# Step 7: Input section for user details
st.markdown("### Your Details:")
col1, col2 = st.columns(2)
with col1:
    name = st.text_input("Your Name")
    year = st.number_input("Year", min_value=2000, max_value=2100, value=datetime.now().year)

st.write("---")
```

User Input Details

- The purpose of this section is to allow users to input data for calculating carbon emissions.

Purpose

- This method establishes the application interface and gathers user data to compute carbon emissions. The calculation is divided into three categories: Energy, Waste, and Travel.

Code Breakdown

The title is presented using st.title, and a description is shown via st.write. The remaining code collects data from the user, including:

- **Energy Consumption** (such as electricity usage)
- **Waste Generation** (daily or weekly waste output)
- **Travel Distance** (kilometers traveled by vehicle)

These inputs are based on whether the user is an individual, a small business, or an industrial company.

Calculation Process

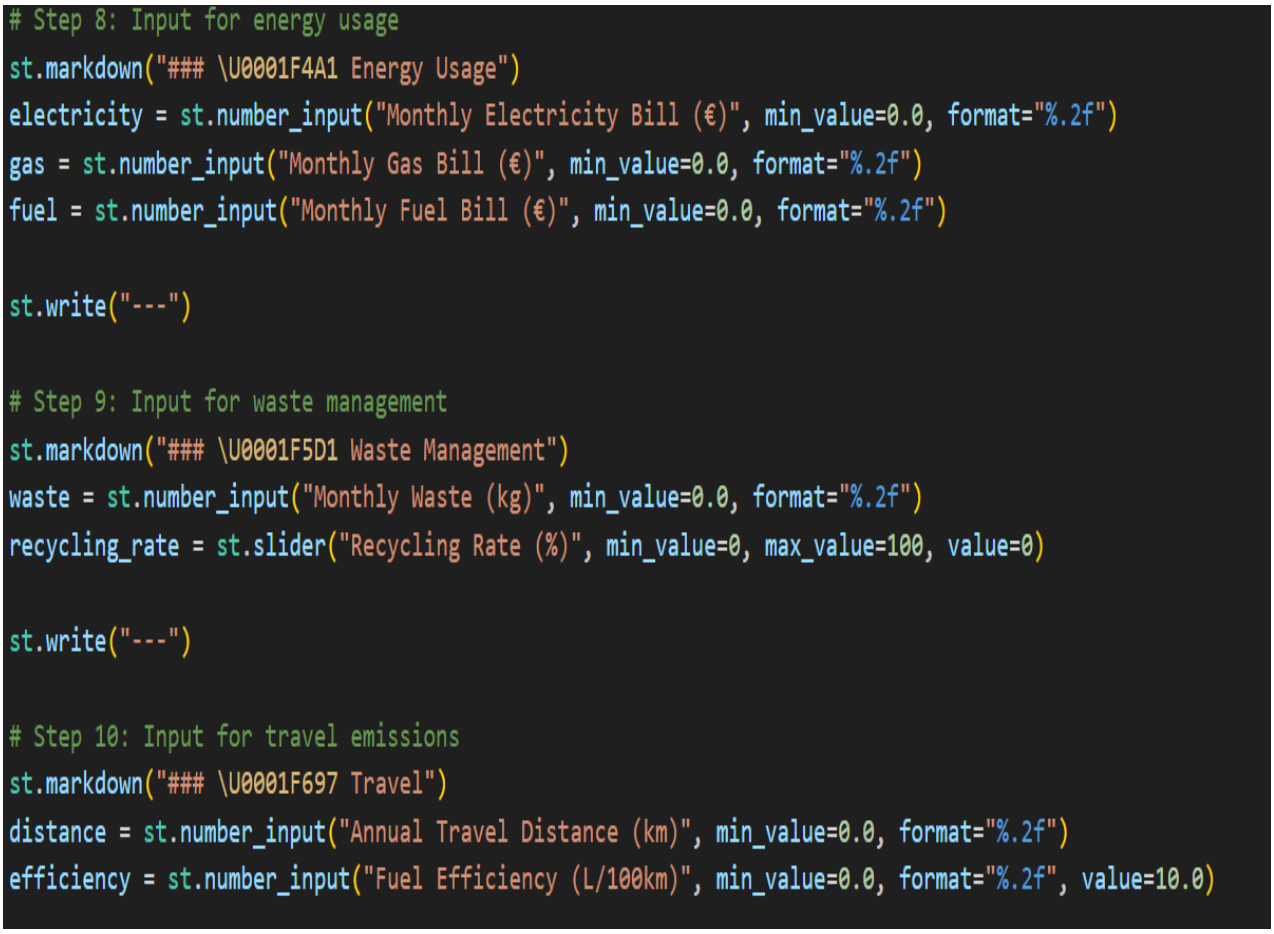
- Once the "Calculate Emission" button is pressed, the input data is processed using predefined formulas. The output displays the total emissions, along with separate breakdowns for energy, waste, and travel emissions, depending on whether the user is an individual or a company.

User Input and Collection of Data

This code segment is designed to collect user input related to energy consumption, waste management, and travel emissions. The inputs include monthly electricity, gas, and fuel expenses, waste quantity, recycling rate, travel distance, and fuel efficiency.

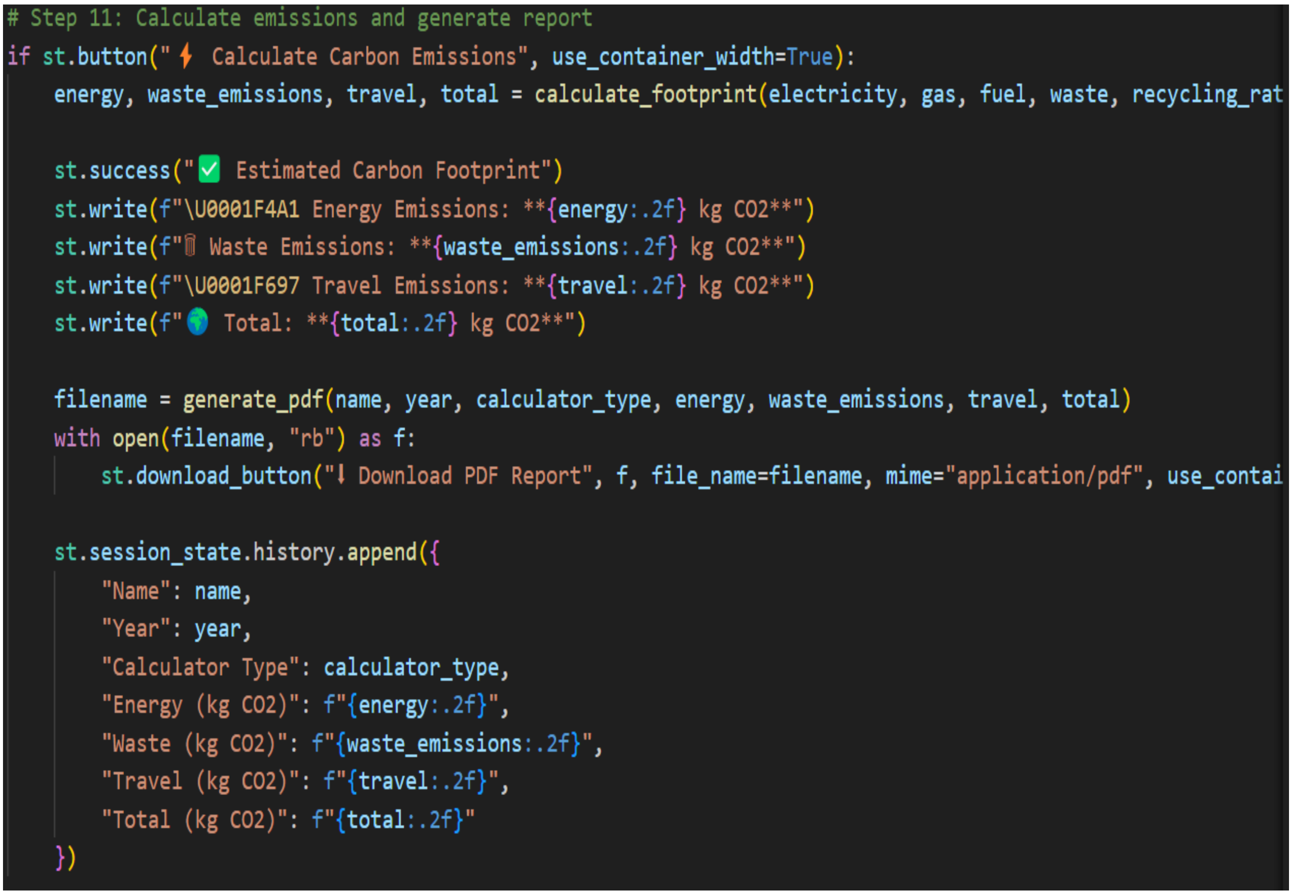
Once the data is collected, it will be used to generate a PDF report when the user opts to download the final summary. The report will cover essential details such as Name, Year, Type, Energy, Waste, Travel, and the overall carbon emissions impacting the environment.

The purpose of this code is to gather the necessary information for compiling a comprehensive carbon emission report in PDF format, allowing users to view and analyse their environmental impact.



```
# Step 8: Input for energy usage
st.markdown("### \U0001F4A1 Energy Usage")
electricity = st.number_input("Monthly Electricity Bill (€)", min_value=0.0, format="%.2f")
gas = st.number_input("Monthly Gas Bill (€)", min_value=0.0, format="%.2f")
fuel = st.number_input("Monthly Fuel Bill (€)", min_value=0.0, format="%.2f")

st.write("---")

# Step 9: Input for waste management
st.markdown("### \U0001F5D1 Waste Management")
waste = st.number_input("Monthly Waste (kg)", min_value=0.0, format="%.2f")
recycling_rate = st.slider("Recycling Rate (%)", min_value=0, max_value=100, value=0)

st.write("---")

# Step 10: Input for travel emissions
st.markdown("### \U0001F697 Travel")
distance = st.number_input("Annual Travel Distance (km)", min_value=0.0, format="%.2f")
efficiency = st.number_input("Fuel Efficiency (L/100km)", min_value=0.0, format="%.2f", value=10.0)
```

Determine Emissions and Produce a Report



```
# Step 11: Calculate emissions and generate report
if st.button("⚡ Calculate Carbon Emissions", use_container_width=True):
    energy, waste_emissions, travel, total = calculate_footprint(electricity, gas, fuel, waste, recycling_rat

    st.success("✅ Estimated Carbon Footprint")
    st.write(f"\U0001F4A1 Energy Emissions: **{energy:.2f} kg CO2**")
    st.write(f"🗑 Waste Emissions: **{waste_emissions:.2f} kg CO2**")
    st.write(f"\U0001F697 Travel Emissions: **{travel:.2f} kg CO2**")
    st.write(f"🌍 Total: **{total:.2f} kg CO2**")

    filename = generate_pdf(name, year, calculator_type, energy, waste_emissions, travel, total)
    with open(filename, "rb") as f:
        st.download_button("⬇ Download PDF Report", f, file_name=filename, mime="application/pdf", use_contai

    st.session_state.history.append({
        "Name": name,
        "Year": year,
        "Calculator Type": calculator_type,
        "Energy (kg CO2)": f"{energy:.2f}",
        "Waste (kg CO2)": f"{waste_emissions:.2f}",
        "Travel (kg CO2)": f"{travel:.2f}",
        "Total (kg CO2)": f"{total:.2f}"
    })
```

When the user selects the "Calculate Carbon Emissions" button in this step, the code uses the calculate_footprint function to determine the total emissions as well as emissions for energy, waste, and travel. A success message and formatted emission values are then shown alongside the results.

Following computation, all of the emission data is included in a PDF report that is produced using the generate_pdf function. After that, the user can click the "Download PDF Report" button to download this report.

For future reference or record-keeping, the computed data (such as name, year, calculator type, and all emission results) is also saved into the session state (st.session_state.history.append).

What the User Does in This Step

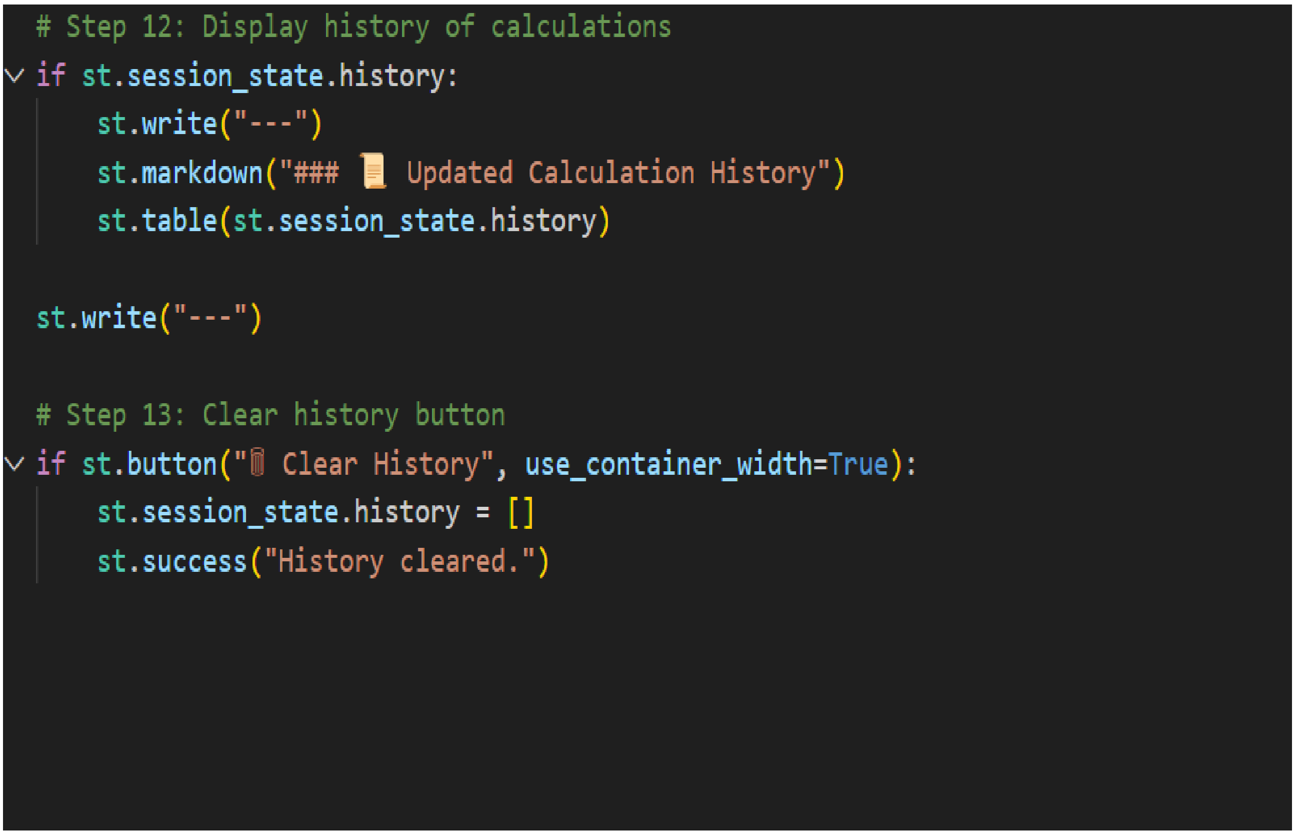
1.Initiates Calculation:
Clicks the "Calculate Carbon Emissions" button to process and calculate the emissions data.

2.Views Emission Results:
The calculated emissions for energy, waste, travel, and the total are displayed on the screen.

3.Downloads the Report:
Clicks the "Download PDF Report" button to download the final report containing all the calculated emissions.

4.Data Storage:
The calculated data is automatically saved into the session history for future use and reference.

Display and Control Calculation History



```
# Step 12: Display history of calculations
if st.session_state.history:
    st.write("---")
    st.markdown("### 📜 Updated Calculation History")
    st.table(st.session_state.history)

st.write("---")

# Step 13: Clear history button
if st.button("🗑 Clear History", use_container_width=True):
    st.session_state.history = []
    st.success("History cleared.")
```

In this section, users can view their recent carbon footprint calculation history, which is displayed in a table format. The history is managed using st.session_state.history, which automatically records previous inputs and results.

Users also have the option to clear this history by clicking the **"Clear History"** button, which removes all stored data and confirms the action with a success message.

Use Case Example

Input Example:

Type**:** Personal
Name: Dhruvin Kheni
Year: 2024
Monthly Energy Bills:

- Electricity: €60
- Gas: €25
- Fuel: €15

Monthly Waste: 35 kg
Recycling Rate: 60%
Annual Travel Distance: 15,000 km
Fuel Efficiency: 7 L/100km

Output Example

- Energy Emissions: 580.5 kg CO_2
- Waste Emissions: 28.0 kg CO_2
- Travel Emissions: 3,750 kg CO_2
- Total Emissions: 4,358.5 kg CO_2

Advantages

For Individuals:

- Helps in pinpointing areas where daily habits can be improved.
- Promotes energy-saving practices and encourages recycling efforts.

For Businesses:

- Identifies gaps and inefficiencies in operational activities.
- Assists in tracking and achieving sustainability targets.

Enhancements and Future Features

1. **Additional Categories**
    - Incorporate emissions from food consumption, water usage, and electronic devices.

2. **Input Validation**
    - Ensure accurate and realistic data entry with proper validation checks.

3. **Enhanced Visualization**
    - Introduce interactive charts to display detailed emission breakdowns.

4. **Multi-User Functionality**
    - Implement account-based history tracking for a personalized user experience.

5. **API Integration**
    - Connect with external APIs to retrieve real-time data on energy and waste emission factors.

Conclusion

A helpful tool for individuals, businesses, and sectors looking to comprehend their environmental impact is the Carbon Footprint Calculator. By using it to streamline complex computations and offer useful insights, users can make significant progress towards sustainability. With more enhancements, it could grow into an even more extensive and potent resource.