Carbon Emissions Calculator

Introduction

During this time, when environmental awareness is a crucial substance, one of the most important aspects for us to get to know is our carbon footprint. The Carbon Footprint Calculator is an easy-to-use application for person, small company, and industry to gauge their ecological impact on carbon footprint. This calculator calculates and surveys give result on these three main categories: energy usage, waste managment, and Travel. Add in more, it give extensive assessment and the possibity of generating and download reports, fostering due reflection to reduce emissions and take important steps towards environment.
This report provides a walkthrough of the calculator's functionality, user interface, and the logic of the calculator. It also highlights its potential applications and suggestions.

Purpose

The goal is:

1. Person can calculate their yearly carbon emissions.
2. They can identify major contributors to their carbon footprint.
3. Also people can obey important steps to be a more sustainable lifestyle.

The calculator is devlopde to be accessible, engaging, educational and user friendly, allowing users to explore the environmental impact of their daily activities and operations.

GitHub Repository

https://github.com/Ay1932/Computer-Programming-M602B/tree/main/Final%20Project

Main Parts of Emission

The calculator calculate on these breaks down emissions into three main parts: first, energy usage second, waste management and third, travel. every aspect of our daily routines can have a some impact on carbon emissions, and understanding them helps us make more eco-friendly choices!

1. **Energy Usage**:
    - Electricity, gas, and fuel usage contribute to carbon emissions on our earth.
    - The Carbon calculator calculates by analyzing monthly bills, and applying standard emission factor for each type of energy source.
    - With this all three in energy parts, the Carbon calculator calculates the given equation and gives the final result in the end.

2. **Waste Management**:
    - Waste disposed of in landfills emits greenhouse gases, including methane.
    - Recycling helps lower these emissions, and the calculator incorporates the recycling rate to change the waste emissions accordingly.

3. **Travel**:
    - These emissions from both personal and business travel are determined by yearly travel distance and vehicle fuel efficiency.
    - Fuel efficiency provides an accurate and customized estimate based on the user's travel behaviours or business travel purposes.

Emission Enhancers

Users can select from three types of calculators:

- **Personal**: Standard for individuals.
- **Business**: Increased to account for higher-scale operations.
- **Industrial**: Highest for large-scale activities.

Every type can change the emission values to represent the user’s particular needs and activities.

About Application

User Interface

The application's interface is easy to understand and divided into logical sections:

1. **User Information**:
    - Users have to provide their name, and the year of calculation which they want to calculate, and select their calculator type for personal use or for company.
2. **Energy Usage**:
    - For input fields are for monthly electricity, gas, and fuel expenses.
3. **Waste Management**:
    - Users have to input their monthly waste which they generated in kilograms and select a recycling percentage rate using a slider.
4. **Travel**:
    - Users have to give their yearly travel distance in kilometers and their vehicle's fuel efficiency (liters per 100 km).
5. **Results**:
    - After clicking "Calculate" users can see a detailed breakdown of emissions on energy usage, waste management, and travel.
    - Total emissions are displayed all category-specific emissions.
6. **PDF Report Generation**:
    - A downloadable PDF report summarizes the results for record-keeping or sharing.
7. **Calculation History**:
    - It displays the recent calculations for quick review.
    - It includes an option to clear history.

Calculation Logic with Formula

The calculator uses well-performed formulas to calculate carbon emissions.

1. Energy Emissions:0

Formula: (Monthly Electricity bills) * 12 * (0.0005) + (Monthly Natural gas Bills) * 12 * (0.0053) + (Monthly Fuel Bill) * 12 * (2.32)

- **Electricity**: Measured in kilowatt-hours (kWh).
- **Gas**: Natural gas consumption in cubic meters (m^3).
- **Fuel**: Vehicle or heating fuel in litters.
- **Multiplier**: Adjusts emissions based on calculator type.

2. Waste Emissions:

Formula: (Total Waste Generated Per Month) *12* (0.57-recycling/composting percentage)

- Recycling reduces the waste emission factor, and higher recycling rates.

3. Travel Emissions:

Formula: (Total Kilometres Travelled Per Year) * (1/Average Fuel Efficiency in L/100km) * 2.31

- Efficiency is provided as litter per 100 km.
- Multiplier adjusts emissions for larger-scale operations.

4. Total Emissions:

Formula: Energy Emissions + Waste Emissions + Travel Emissions

---

PDF Report Generation

The calculator uses the `fpdf` library to create a downloadable PDF report summarizing the user's emissions. Main features of the report:

- Includes user details (name, year, and calculator type).
- Displays category-specific and total emissions.
- Professionally formatted for easy sharing.

History Management

Calculation history is stored in a JSON file (`calculation_history.json`), ensuring data persistence between sessions. The tool:

- Saves the 10 most recent calculations.
- Displays the latest 5 entries in the sidebar.
- Allows users to clear the history with a button click.

Technical Stack

Libraries Used:

1. **Streamlit**:
   - Provides an interactive and responsive UI for the application.
   - Allows real-time updates and downloads.
2. **FPDF**:
   - Used to generate professional PDF reports.
3. **JSON**:
   - Facilitates saving and retrieving calculation history.
4. **datetime**:
   - Adds timestamps to history entries.
5. **BytesIO**:
   - Efficiently handles PDF generation in memory.

Code and Explanation

```python
# Performing calculations
class CalcEmission:
    MULTIPLIERS = {
        "Personal": 1.0,
        "Business": 1.5,
        "Industrial": 2.0,
    }

    @staticmethod
    def energy_emi(electricity, gas, fuel, multiplier):
        """Calculate emissions from energy use."""
        return ((electricity * 12 * 0.0005) + (gas * 12 * 0.0053) + (fuel * 12 * 2.32)) * multiplier

    @staticmethod
    def waste_emi(waste, recycling_rate, multiplier):
        """Calculate emissions from waste management."""
        return (waste * 12 * (0.57 - recycling_rate / 100)) * multiplier

    @staticmethod
    def travel_emi(distance, efficiency, multiplier):
        """Calculate emissions from travel."""
        return (distance * (1 / efficiency) * 2.31) * multiplier

    @staticmethod
    def total_emi(energy, waste, travel):
        """Sum up total emissions."""
        return energy + waste + travel
```

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

Class CalcEmission:

In this class, first those are multiplier which are used for personal use, small business and in the last for the industrial area.

It is divided in 4 part one for energy usage, waste management and travel and in last for total which gives total emission of all these three combining emissions.

Main Application

```
# Main application
class CarbonCalculator:
    def __init__(self):
        self.history = HistoryManager.load()

    def run(self):
        st.title("🌿 Carbon Footprint Calculator")
        st.write("Easily calculate your carbon emissions and learn how to reduce your footprint.")

        # User Input
        st.header("📋 Your Information")
        type = st.selectbox("Select Calculator Type:", ["Personal", "Business", "Industrial"])
        name = st.text_input("Your Name:", "Your Name")
        year = st.number_input("Year:", min_value=2000, max_value=2100, value=datetime.now().year)

        # Energy Input
        st.header("💡 Energy Usage")
        electricity = st.number_input("Monthly Electricity Bill (€):", min_value=0.0, value=0.0)
        gas = st.number_input("Monthly Gas Bill (€):", min_value=0.0, value=0.0)
        fuel = st.number_input("Monthly Fuel Bill (€):", min_value=0.0, value=0.0)

        # Waste Input
        st.header("🗑 Waste Management")
        waste = st.number_input("Monthly Waste (kg):", min_value=0.0, value=0.0)
        recycling_rate = st.slider("Recycling Rate (%):", min_value=0, max_value=100, value=0)

        # Travel Input
        st.header("🚗 Travel")
        travel_distance = st.number_input("Annual Travel Distance (km):", min_value=0.0, value=0.0)
        fuel_efficiency = st.number_input("Fuel Efficiency (L/100km):", min_value=0.1, value=10.0)
```

In this portion, User have to give data to system for calculation for this.

Purpose

The run methods set up app inter face and collect data from user inputs for calculating carbon emission with three categories Energy, Waste and Travel.

Sections of code

Display the title with st.title and for description used st,write.

Rest of the code for takes data from user-end where user give a specific number of energy, waste and travelled kms for individuals, small business or a industrial company.

When the calculate emission button will be clicked then the given data will be procced with the equations and gives the final answer of total emission, energy emission, waste emission

and travel emission of individual and other companies. For that code is as per image of the project.

```
if st.button("Calculate Emissions"):
    multiplier = CalcEmission.MULTIPLIERS[type]

    energy = CalcEmission.energy_emi(electricity, gas, fuel, multiplier)
    waste = CalcEmission.waste_emi(waste, recycling_rate, multiplier)
    travel = CalcEmission.travel_emi(travel_distance, fuel_efficiency, multiplier)
    total = CalcEmission.total_emi(energy, waste, travel)

    # Show Results
    st.subheader("Your Results")
    st.write(f"**Name:** {name}")
    st.write(f"**Year:** {year}")
    st.write(f"**Calculator Type:** {type}")
    st.write(f"**Energy Emissions:** {energy:.2f} kgCO2")
    st.write(f"**Waste Emissions:** {waste:.2f} kgCO2")
    st.write(f"**Travel Emissions:** {travel:.2f} kgCO2")
    st.write(f"**Total Emissions:** {total:.2f} kgCO2")
```

Class PDFgenerator

In this class, the proceed data will be displayed in pdf when the user wants to download the final report.

In the explanation, a pdf will be generated with all aspects of data such as Name, Year, Type, Energy, Waste, Travel, and last Total emission emitted in the environment.

All the given shown code is for generating pdf and users can see the report for carbon emission.

```
# Create a PDF report
class PDFGenerator:
    @staticmethod
    def generate(name, year, type, energy, waste, travel, total):
        """Generate a PDF report of the emissions calculation."""
        pdf = FPDF()
        pdf.add_page()
        pdf.set_font("Arial", size=12)

        pdf.cell(200, 10, txt="Carbon Emission Report", ln=True, align="C")
        pdf.ln(10)

        pdf.cell(200, 10, txt=f"Name: {name}", ln=True)
        pdf.cell(200, 10, txt=f"Year: {year}", ln=True)
        pdf.cell(200, 10, txt=f"Calculator Type: {type}", ln=True)
        pdf.cell(200, 10, txt=f"Energy Emissions: {energy:.2f} kgCO2", ln=True)
        pdf.cell(200, 10, txt=f"Waste Emissions: {waste:.2f} kgCO2", ln=True)
        pdf.cell(200, 10, txt=f"Travel Emissions: {travel:.2f} kgCO2", ln=True)
        pdf.cell(200, 10, txt=f"Total Carbon Emissions: {total:.2f} kgCO2", ln=True)

        buffer = BytesIO()
        pdf_output = pdf.output(dest='S').encode('latin1')
        buffer.write(pdf_output)
        buffer.seek(0)

        return buffer
```

Class HistoryManager:

```python
# Manage calculation history
class HistoryManager:
    @staticmethod
    def load():
        """Load calculation history from a file."""
        try:
            with open("calculation_history.json", "r") as file:
                return json.load(file)
        except FileNotFoundError:
            return []

    @staticmethod
    def save(history):
        """Save calculation history to a file."""
        with open("calculation_history.json", "w") as file:
            json.dump(history, file, indent=4)
```

In this class, user can see the history of recent calculations of carbon footprint in the left side of the screen. For this I used json module which is already preinstall with python and user can also cleared this history by clicking clear history.

Example Usage

Input Example:

- Type: Personal
- Name: Ayush Pandav
- Year: 2023
- Monthly Energy Bills: Electricity (€50), Gas (€30), Fuel (€20)
- Monthly Waste: 40 kg
- Recycling Rate: 50%
- Annual Travel Distance: 12,000 km
- Fuel Efficiency: 8 L/100km

Output Example:

- **Energy Emissions**: 559.0.1 kg CO_2
- **Waste Emissions**: 33.6 kg CO_2
- **Travel Emissions**: 3,465 kg CO_2
- **Total Emissions**: 4,057.8 kg CO_2

Benefits

For Individuals:

- Helps to identify aspects of improvement in daily.
- Motivates recycling and energy conservation.

For Businesses:

- Highlights inability in operations.
- Aids in tracking progress to sustainability goals.

For Industries:

- Gives an overview of carbon emissions.
- Further compliance with environmental methods.

Enhancements and Future Features

1. **Other Categories**:
    - Food burning, water usage, and gadgets emissions.
2. **Data Validation**:
    - Realistic input with appropriate pressure.
3. **Visualization**:
    - Include charts to show emissions breakdown.
4. **Multi-User Support**:
    - Add account-based history for personalized experiences.
5. **Integration with APIs**:
    - Pull real-time data for energy and waste emission factors.

Conclusion

The Carbon Footprint Calculator is a useful tool for people, companies, and sectors who want to understand their environmental impact. Users can significantly advance toward sustainability by using it to simplify complicated calculations and provide actionable insights. It could become an even more comprehensive and powerful resource with additional improvements.