MEASURE ENERGY CONSUMPTION

PHASE 1

Problem Definition :

The problem at hand is to accurately measure and monitor energy consumption in various settings, such as residential, commercial, and industrial sectors. Accurate energy consumption measurement is crucial for several reasons, including environmental sustainability, cost management, and regulatory compliance. This problem definition outlines the key aspects and challenges associated with measuring energy consumption.

Design Thinking :

1. Empathize:

a. User Research: Begin by conducting extensive research to understand the needs, pain points, and behaviors of different stakeholders, including residential consumers, businesses, and energy utilities. Consider their motivations, challenges, and attitudes toward energy consumption.

b. Interviews and Surveys: Conduct interviews and surveys to gather insights directly from users. Ask about their current methods of measuring and managing energy consumption, as well as their frustrations and desires.

c. Observation: Observe users in their real-life contexts to gain deeper insights into their energy consumption habits and the challenges they face.

2. Define:

a. Problem Statement: Based on your research, define a clear problem statement that encapsulates the challenges and opportunities related to measuring energy consumption. For example, "How might we create a user-friendly and cost-effective solution for tracking and reducing energy consumption in residential households?"

b. User Personas: Create user personas that represent the different types of users you've identified, including their needs, goals, and pain points.

3. Ideate:

Brainstorming: Conduct brainstorming sessions with a cross-functional team to generate creative ideas for measuring and managing energy consumption. Encourage wild ideas and avoid premature judgment.

4. Prototype:

Prototyping: Create low-fidelity prototypes of your energy consumption measurement solution. This could be sketches, wireframes, or even physical mock-ups of hardware components.

5. Test:

a. Real-World Testing: Implement a pilot program or deploy a limited version of your solution in a real-world setting to assess its performance and gather data on energy consumption.

b. Data Analysis: Analyze the data collected to determine the effectiveness of your solution in accurately measuring and reducing energy consumption.

6. Implement:

a. Full-Scale Deployment: Roll out your energy consumption measurement solution on a larger scale, taking into account the lessons learned during testing and iteration.

b. Marketing and Education: Develop marketing and educational materials to promote the benefits of your solution to users and encourage them to adopt energy-efficient behaviors.

7. Monitor and Maintain:

a. Continuous Monitoring: Continuously monitor the performance of your solution and gather user feedback to address any issues and make necessary updates.

PHASE 2

Measure Energy Consumption

My design for a system to measure energy consumption and put it into innovation to solve the problem is as follows:

Step 1: Develop a smart energy sensor

This sensor would be attached to electrical appliances and devices to measure their energy consumption in real time. The sensor would be able to communicate with a central hub or server to transmit the data.

Step 2: Develop a machine learning algorithm

This algorithm would be able to analyze the energy consumption data to identify patterns and trends. It would also be able to detect anomalies and inefficiencies.

Step 3: Develop a user-friendly interface

This interface would allow users to view their energy consumption data in a variety of ways. It would also provide users with insights and recommendations on how to reduce their energy consumption.

Benefits of this system:

* **Improved energy efficiency:** The system would help users to identify and address areas where they are wasting energy. This could lead to significant savings on energy costs.
* **Reduced environmental impact:** By reducing energy consumption, the system would help to reduce greenhouse gas emissions and other environmental impacts.
* **Increased awareness:** The system would help users to become more aware of their energy consumption habits. This could lead to behavioral changes that further reduce energy consumption.

Steps to transform the design into innovation:

* **Develop and test the smart energy sensor:** The sensor would need to be designed and tested to ensure that it is accurate and reliable. It would also need to be made affordable and easy to install.
* **Develop and train the machine learning algorithm:** The algorithm would need to be trained on a large dataset of energy consumption data. This would allow it to learn the patterns and trends that are associated with different types of appliances and devices.
* **Develop the user-friendly interface:** The interface would need to be designed to be easy to use and understand. It should provide users with the information they need to make informed decisions about their energy consumption.
* **Partner with energy providers:** Energy providers could partner with the company to offer the system to their customers. This would help to reach a wider audience and make the system more accessible.
* **Market the system to businesses and consumers:** The system could be marketed to businesses and consumers as a way to save money on energy costs and reduce their environmental impact.

Conclusion:

This system has the potential to make a significant contribution to improving energy efficiency and reducing environmental impact. By transforming the design into innovation, the system could be made available to a wider audience and help people to make more informed decisions about their energy consumption.

PHASE 3

DATASET LINK: https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
RED = "\033[91m"
GREEN = "\033[92m"
YELLOW = "\033[93m"
BLUE = "\033[94m"
RESET = "\033[0m"

df = pd.read_csv("C:/Users/PMCTECH/Documents/AEP_hourly (1).csv")

df["Datetime"] = pd.to_datetime(df["Datetime"])

# DATA CLEANING
print(BLUE + "\nDATA CLEANING" + RESET)
# --- Check for missing values
missing_values = df.isnull().sum()
print(GREEN + "Missing Values : " + RESET)
print(missing_values)
# --- Handle missing values
df.dropna(inplace=True)
# --- Check for duplicate values
duplicate_values = df.duplicated().sum()
print(GREEN + "Duplicate Values : " + RESET)
print(duplicate_values)
# --- Drop duplicate values
df.drop_duplicates(inplace=True)

# DATA ANALYSIS
print(BLUE + "\nDATA ANALYSIS" + RESET)
# --- Summary Statistics
summary_stats = df.describe()
print(GREEN + "Summary Statistics : " + RESET)
print(summary_stats)

# Data Visualization
# Line plot for energy consumption over time
plt.figure(figsize=(12, 6))
plt.plot(df.index, df["AEP_MW"], label="Energy Consumption (AEP_MW)")
plt.xlabel("Datetime")
plt.ylabel("Energy Consumption (MW)")
plt.title("Energy Consumption Over Time")
plt.grid()
plt.legend()
plt.show()

# SAVING THE FILE
```

```
df.to_csv("/kaggle/working/cleaned_AEP_hourly.csv", index=False)
print(BLUE + "\nDATA ANALYSIS" + RESET)
print(GREEN + "Data Cleaned and Saved !" + RESET)
```

Output:

```
DATA CLEANING
Missing Values :
Datetime    0
AEP_MW      0
dtype: int64
Duplicate Values :
0

DATA ANALYSIS
Summary Statistics :
              AEP_MW
count  121273.000000
mean    15499.513717
std      2591.399065
min      9581.000000
25%     13630.000000
50%     15310.000000
75%     17200.000000
max     25695.000000
```

PHASE 4

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

RED = "\033[91m"
GREEN = "\033[92m"
YELLOW = "\033[93m"
BLUE = "\033[94m"
RESET = "\033[0m"

df = pd.read_csv("C:/Users/PMCTECH/Documents/AEP_hourly.csv")

df["Datetime"] = pd.to_datetime(df["Datetime"])

# DATA CLEANING
print(BLUE + "\nDATA CLEANING" + RESET)
# --- Check for missing values
missing_values = df.isnull().sum()
print(GREEN + "Missing Values : " + RESET)
print(missing_values)
# --- Handle missing values
df.dropna(inplace=True)
# --- Check for duplicate values
duplicate_values = df.duplicated().sum()
print(GREEN + "Duplicate Values : " + RESET)
print(duplicate_values)
# --- Drop duplicate values
df.drop_duplicates(inplace=True)

# DATA ANALYSIS
print(BLUE + "\nDATA ANALYSIS" + RESET)
# --- Summary Statistics
summary_stats = df.describe()
print(GREEN + "Summary Statistics : " + RESET)
print(summary_stats)
```

```
# Data Visualization
# Line plot for energy consumption over time
plt.figure(figsize=(12, 6))
plt.plot(df.index, df["AEP_MW"], label="Energy Consumption (AEP_MW)")
plt.xlabel("Datetime")
plt.ylabel("Energy Consumption (MW)")
plt.title("Energy Consumption Over Time")
plt.grid()
plt.legend()
plt.show()

# SAVING THE FILE
df.to_csv("C:/Users/PMCTECH/Documents/AEP_hourly.csv", index=False)
print(BLUE + "\nDATA ANALYSIS" + RESET)
print(GREEN + "Data Cleaned and Saved !" + RESET)
```

#feature engineering

```
import psutil

# Measure energy consumption before feature engineering
before_energy = psutil.cpu_percent()

# Your feature engineering code here

# Measure energy consumption after feature engineering
after_energy = psutil.cpu_percent()

energy_consumption_feature_engineering = after_energy - before_energy
print(f"Energy consumption during feature engineering:
{energy_consumption_feature_engineering}%")
```

#model training

```
import psutil
```

```
# Measure energy consumption before model training
before_energy = psutil.cpu_percent()

# Your model training code here

# Measure energy consumption after model training
after_energy = psutil.cpu_percent()

energy_consumption_training = after_energy - before_energy
print(f"Energy consumption during model training: {energy_consumption_training}%")
```

#model evaluation

```
import psutil

# Measure energy consumption before model evaluation
before_energy = psutil.cpu_percent()

# Your model evaluation code here

# Measure energy consumption after model evaluation
after_energy = psutil.cpu_percent()

energy_consumption_evaluation = after_energy - before_energy
print(f"Energy consumption during model evaluation: {energy_consumption_evaluation}%")
```

OUTPUT:

```
DATA CLEANING
Missing Values :
Datetime    0
AEP_MW      0
dtype: int64
Duplicate Values :
0
```

DATA ANALYSIS

```
Summary Statistics :
             AEP_MW
count  121273.000000
mean    15499.513717
std      2591.399065
min      9581.000000
25%     13630.000000
50%     15310.000000
75%     17200.000000
max     25695.000000
```

DATA ANALYSIS

```
Data Cleaned and Saved !
Energy consumption during feature engineering: 0.0%
Energy consumption during model training: 0.0%
Energy consumption during model evaluation: 0.0%
```