Energy Consumption Prediction System – Project Documentation

1. Project Overview

This project aims to predict energy consumption (in kilowatt-hours, kWh) based on key environmental and usage factors such as temperature, humidity, occupancy, lighting, and HVAC (Heating, Ventilation, and Air Conditioning) status. It features a machine learning model deployed via a user-friendly Streamlit web interface.

2. Dataset Description

The dataset (Energy_consumption.csv) includes historical energy usage records with the following features:

- **Temperature (°C):** Ambient room temperature
- **Humidity (%):** Relative humidity
- **Occupancy:** Number of people present
- **Lighting:** Whether lighting is in use (On/Off)
- **HVAC:** Whether HVAC is active (On/Off)
-

3. Data Preprocessing

Before training the model, the dataset likely underwent preprocessing, including:

- **Encoding categorical features:** Lighting and HVAC values were encoded as binary (0 = Off, 1 = On).
- **Handling missing values:** Any missing entries in the dataset were removed or imputed.
- **Feature scaling (optional):** Standardization or normalization might have been applied

```
# Load dataset
df = pd.read_csv("Energy_consumption.csv")
```

```
# Preprocessing
df['LightingUsage'] = df['LightingUsage'].map({'Off': 0, 'On': 1})
df['HVACUsage'] = df['HVACUsage'].map({'Off': 0, 'On': 1})
```

4. Model Training

Although the model training code is not included, the `app.py` file references a trained model (energy_model.pkl) loaded using pickle. The typical training process would have involved:

- Splitting the dataset into training and test sets
- Training a regression model (e.g., Linear Regression, Random Forest, etc.)
- Evaluating it using metrics like MAE, RMSE, or R^2 score
- Saving the trained model using pickle.dump()

```
# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Save model
with open("energy_model.pkl", "wb") as f:
    pickle.dump(model, f)
```

5. Web App Interface

Developed using **Streamlit**, the web app allows users to:

- Input environmental parameters: ○ Temperature (0–50°C) ○
- Humidity (0–100%) ○ Occupancy (0–20)

. Lighting and HVAC status (On/Off)

- Submit the data to receive an **estimated energy consumption** value.

Code Highlights:

- User inputs are collected using number input, slider, and select box.
- Inputs are formatted as a NumPy array and passed to the model's predict() method.
- The prediction is displayed with st.success

- ## 6. Usage Instructions

To run the app locally:

1. Ensure Python, Streamlit, NumPy, and pickle are installed.
2. Place the following files in the same directory:

3 app.py
4 energy_model.pkl

```
import streamlit as st
import numpy as np
import pickle

# Load model
with open("energy_model.pkl", "rb") as f:
    model = pickle.load(f)

st.title("🔋 Energy Consumption Predictor")

st.markdown("Enter the input values to predict energy usage.")
```

3. Run the app:
   bash CopyEdit
   streamlit run app.py

4. Use the web interface to enter values and get a prediction.

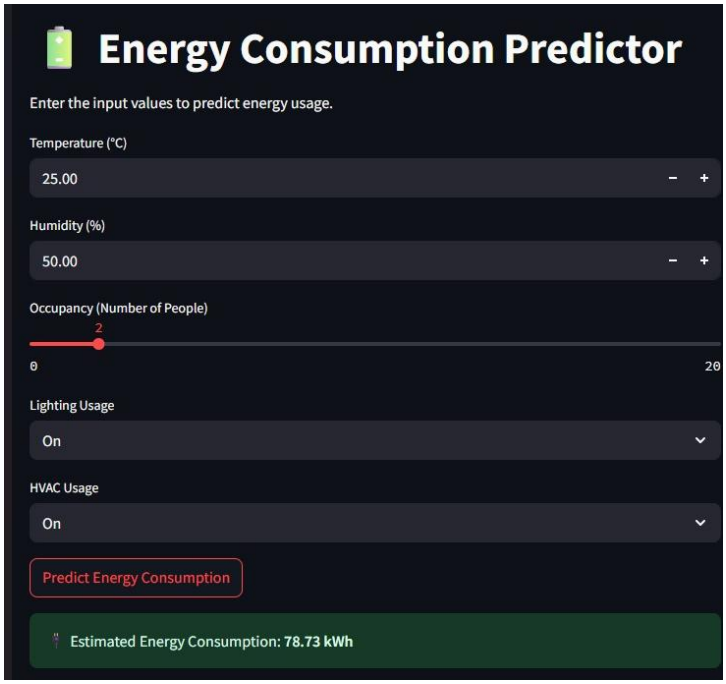
```
# Input fields
temp = st.number_input("Temperature (°C)", min_value=0.0, max_value=50.0, value=25.0)
humidity = st.number_input("Humidity (%)", min_value=0.0, max_value=100.0, value=50.0)
occupancy = st.slider("Occupancy (Number of People)", 0, 20, 5)
lighting = st.selectbox("Lighting Usage", ["Off", "On"])
hvac = st.selectbox("HVAC Usage", ["Off", "On"])

# Convert to model format
lighting_val = 1 if lighting == "On" else 0
hvac_val = 1 if hvac == "On" else 0

# Predict
if st.button("Predict Energy Consumption"):
    features = np.array([[temp, humidity, occupancy, lighting_val, hvac_val]])
    prediction = model.predict(features)[0]
    st.success(f"🔌 Estimated Energy Consumption: **{prediction:.2f} kWh**")
```

7. Conclusion and Future Work

This project provides a useful tool for estimating energy consumption, potentially helping facilities optimize resource use. Future enhancements could include:



- Adding time-based features (e.g., time of day, day of the week)
- Including historical consumption trends
- Enhancing the UI/UX with graphs or visual analytics
- Expanding the dataset for better generalization