Phase 3: Implementation of Project

Title: ENERGY USAGE OPTIMIZATION

Objective

The goal of Phase 3 is to implement the core components of the AI-Powered Energy Usage Optimization Assistant based on the planning and innovative solutions developed in Phase 2. This includes the development of the AI model for consumption analysis, the chatbot interface for user interaction, initial integration with smart home IoT devices, and the implementation of data security measures.

1. AI Model Development

Overview

The primary feature of the AI-Powered Assistant is its ability to analyze energy usage patterns and provide personalized energy-saving recommendations. In Phase 3, the AI model will be trained and deployed to recognize common usage patterns and suggest optimizations.

Implementation

- **Machine Learning Model for Energy Analysis**: The AI uses machine learning and pattern recognition to understand textual or numerical input (e.g., usage data or user queries). The model will be trained on datasets containing typical energy consumption patterns in households and businesses.

- **Data Source**: A dataset containing historical energy usage data and common inefficiencies will be used. Real-time data integration will be considered in future phases.

Outcome
By the end of this phase, the AI model should provide practical advice such as reducing usage during peak hours, suggesting appliance usage schedules, or highlighting unusual consumption patterns.

2. Chatbot Development

Overview
The AI assistant will be accessible via a chatbot interface, enabling users to interact and receive recommendations in a conversational format.

Implementation

- **User Interaction**: The chatbot will ask questions like "When do you use the most energy?" or "Would you like tips to lower your electricity bill?" and provide responses based on AI analysis.
- **Language Support**: Currently, the chatbot will support English. Support for additional languages will be added in future phases.

Outcome
By the end of Phase 3, the chatbot will be able to engage with users and offer tailored energy-saving advice based on their usage patterns or inputs.

3. IoT Device Integration (Optional)

Overview
Although optional at this stage, initial integration with smart home devices and sensors will be pursued to gather real-time energy data and enhance recommendation accuracy.

Implementation

- **Energy Data Collection**: Data from smart meters, plugs, or thermostats will be used to track real-time energy consumption.
- **API Use**: APIs from platforms like Google Nest, SmartThings, or Home Assistant will be used to interface with these devices.

Outcome
The system will be able to connect to available smart devices and gather basic usage data such as appliance activity or overall power draw. Full integration and real-time optimization will be expanded in future phases.

4. Data Security Implementation

Overview
To protect user data related to energy usage and personal preferences, Phase 3 will implement foundational security practices including encryption and access controls.

Implementation

- **Encryption**: All user-provided data (e.g., home energy data, user preferences) will be stored with encryption.
- **Secure Storage**: Data will be securely stored in a restricted-access environment compliant with relevant privacy standards.

Outcome
By the end of Phase 3, the system will securely store user data and protect all interactions and analytics using basic encryption and security best practices.

5. Testing and Feedback Collection

Overview
Initial testing will be conducted to evaluate the performance of the AI assistant, its ability to provide helpful energy recommendations, and overall user experience.

Implementation

- **Test Groups**: A group of test users will simulate energy inputs and usage patterns to evaluate how the assistant performs in varied scenarios.
- **Feedback Loop**: Feedback will be collected regarding the clarity of recommendations, accuracy of analysis, and ease of using the chatbot.

Outcome
The insights gathered will guide improvements in Phase 4, including enhancements to the AI's intelligence and chatbot design.

Challenges and Solutions

1. Model Accuracy

- *Challenge:* The model may misinterpret certain usage patterns due to limitations in training data.
- *Solution:* Continuous updates and testing will improve the model over time with user feedback loops.

2. User Experience

- *Challenge:* The chatbot may not initially offer the most intuitive experience.
- *Solution:* Feedback will be used to iterate and improve conversational flows and interface design.

3. IoT Device Availability

- *Challenge:* Users may not yet own smart meters or compatible devices.
- *Solution:* Simulated or sample data will be used to showcase the system's full potential.

Outcomes of Phase 3

By the end of Phase 3, the following will be achieved:

1. Basic AI Model: The AI can analyze simple usage patterns and offer energy-saving recommendations.
2. Functional Chatbot Interface: Users can receive actionable insights through a conversational interface.
3. Optional IoT Integration: The system can connect to available smart devices and collect relevant energy usage data.
4. Data Security: User data will be protected through encryption and secure storage.
5. Initial Testing and Feedback: User feedback will be gathered for improvement in future phases.

Next Steps for Phase 4

1. **Improving the AI's Accuracy:** Refine the model using feedback and real-world data.
2. **Expanding Multilingual Support:** Introduce support for additional languages and voice inputs.
3. **Scaling and Optimizing:** Enhance system performance to manage a wider user base and complex energy management scenarios.

SCREENSHOTS OF CODE AND PROGRESS-MUST BE ADDED HERE FOR PHASE 3

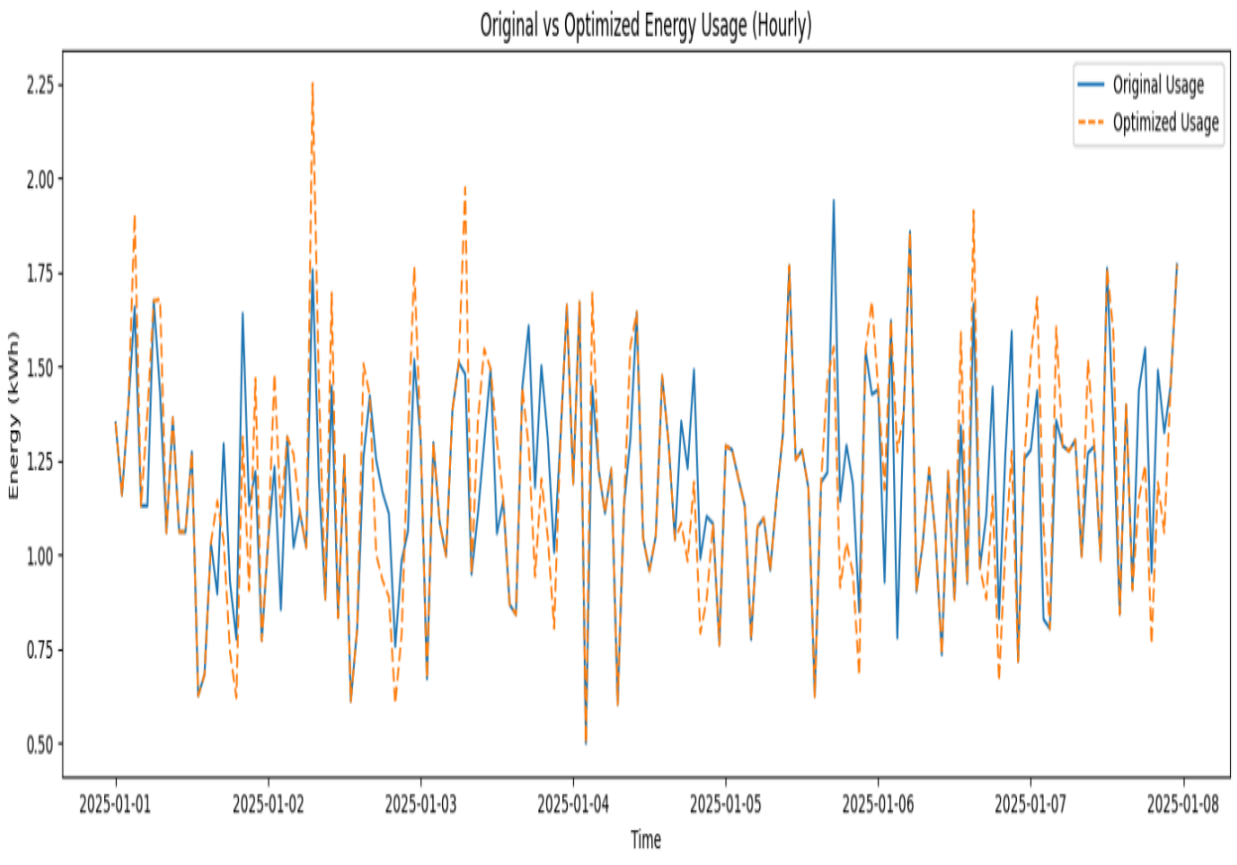
CODE:

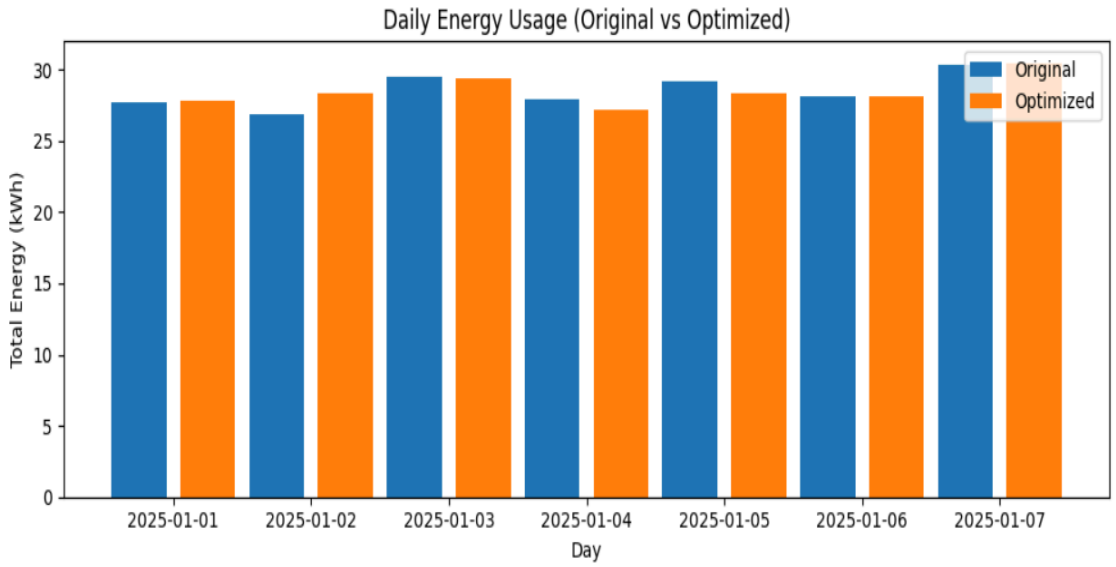
[1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Simulate energy usage data: 7 days of hourly usage (in kWh)
np.random.seed(42)
hours = pd.date_range(start='2025-01-01', periods=168, freq='H')
usage = np.random.normal(loc=1.2, scale=0.3, size=168)  # average 1.2 kWh per hour
usage = np.clip(usage, 0.5, 2.0)

df = pd.DataFrame({'Timestamp': hours, 'Usage_kWh': usage})
df.set_index('Timestamp', inplace=True)

# Define peak hours (e.g., 5 PM - 10 PM)
peak_hours = df.index.hour.isin([17, 18, 19, 20, 21])

# Optimization: Reduce peak usage by 20% and shift to off-peak hours (randomly selected)
df['Optimized_kWh'] = df['Usage_kWh']
df.loc[peak_hours, 'Optimized_kWh'] *= 0.8
shifted_energy = (df['Usage_kWh'][peak_hours] * 0.2).sum()

# Distribute shifted energy to off-peak hours
off_peak_indices = df[~peak_hours].sample(n=len(df[peak_hours]), replace=True).index
for i in off_peak_indices:
    df.at[i, 'Optimized_kWh'] += shifted_energy / len(off_peak_indices)
```

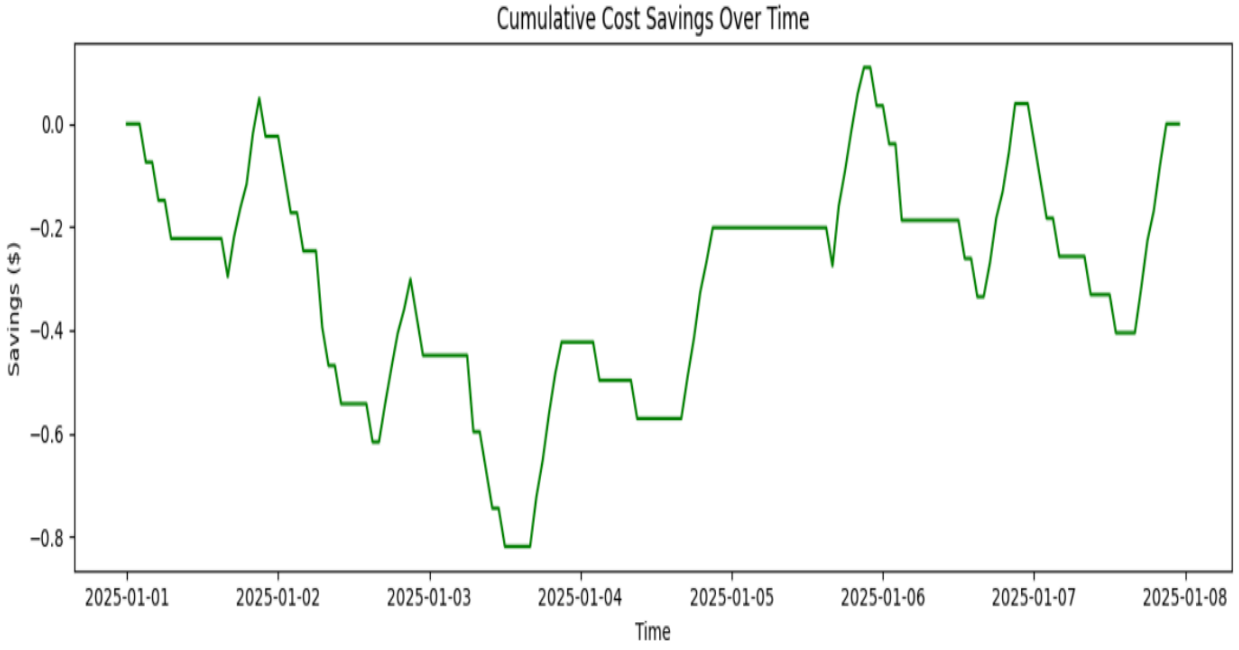
```python
# Calculate savings (assuming higher cost during peak hours)
df['Cost_Original'] = df['Usage_kWh'] * (0.30 if peak_hours.any() else 0.15)
df['Cost_Optimized'] = df['Optimized_kWh'] * (0.30 if peak_hours.any() else 0.15)
df['Savings'] = df['Cost_Original'] - df['Cost_Optimized']

# Plot 1: Hourly Usage Comparison
plt.figure(figsize=(14, 5))
plt.plot(df.index, df['Usage_kWh'], label='Original Usage')
plt.plot(df.index, df['Optimized_kWh'], label='Optimized Usage', linestyle='--')
plt.title('Original vs Optimized Energy Usage (Hourly)')
plt.xlabel('Time')
plt.ylabel('Energy (kWh)')
plt.legend()
plt.tight_layout()
plt.show()

# Plot 2: Daily Total Energy Usage
daily_usage = df.resample('D').sum()
plt.figure(figsize=(10, 4))
plt.bar(daily_usage.index - pd.Timedelta(hours=6), daily_usage['Usage_kWh'], width=0.4, label='Original')
plt.bar(daily_usage.index + pd.Timedelta(hours=6), daily_usage['Optimized_kWh'], width=0.4, label='Optimized')
plt.title('Daily Energy Usage (Original vs Optimized)')
plt.xlabel('Day')
plt.ylabel('Total Energy (kWh)')
plt.legend()
plt.tight_layout()
plt.show()
```

```python
# Plot 3: Cumulative Savings Over Time
df['Cumulative_Savings'] = df['Savings'].cumsum()
plt.figure(figsize=(12, 4))
plt.plot(df.index, df['Cumulative_Savings'], color='green')
plt.title('Cumulative Cost Savings Over Time')
plt.xlabel('Time')
plt.ylabel('Savings ($)')
plt.tight_layout()
plt.show()
```

Out put:




Daily Energy Usage (Original vs Optimized)
Original
Optimized
Total Energy (kWh)
0
5
10
15
20
25
30
2025-01-01
2025-01-02
2025-01-03
2025-01-04
2025-01-05
2025-01-06
2025-01-07
Day


Cumulative Cost Savings Over Time
Savings ($)
0.0
−0.2
−0.4
−0.6
−0.8
2025-01-01
2025-01-02
2025-01-03
2025-01-04
2025-01-05
2025-01-06
2025-01-07
2025-01-08
Time