HIMACHALAM V

113323106036

II ECE A

aut113323eca18

Phase 3: Project Implementation

Title: Energy Efficiency Optimization

Objective

Phase 3's objective is to deploy the central elements of the Energy Efficiency Optimization system according to the plans and strategies formulated during Phase 2. These are the deployment of optimization algorithms, data analytics dashboards, integration of IoT energy-monitoring sensors, and baseline security implementation of energy usage data.

1. Development of Optimization Algorithms

Overview

The main purpose of the system is to study energy consumption patterns and provide optimized usage suggestions for domestic and commercial settings. During Phase 3, the optimization model will be created to address wasteful energy consumption.

Implementation

Data Analytics and Pattern Recognition: Algorithms will be trained to recognize energy usage patterns from past consumption data and suggest optimization measures.

Training Data: Simulated and actual data sets will be utilized, which include typical use cases and lighting, HVAC, and appliance inefficiencies.

Outcome

At this stage, the model should recommend energy-saving activities, like setting thermostat schedules, shutting off non-operational appliances, or minimizing lighting based on occupancy.

2. User Dashboard Development

Overview

The system will have an intuitive dashboard where users can see their energy consumption and get real-time recommendations for optimization.

Implementation

Visualization Tools: Graphs and charts will display usage patterns, peak demand, and comparison to benchmarks.

Interaction: Users can establish goals, get alerts, and view historical performance.

Outcome

The dashboard will allow users to monitor their energy behavior, comprehend optimization recommendations, and act on system insights.

3. IoT Device Integration (Optional)

Overview

While not required in this stage, fundamental integration with IoT sensors and smart energy meters will be added to gather real-time usage data.

Implementation

Device Data Collection: Where available, device data like that from smart plugs and energy meters will be gathered.

API Integration: Manufacturers' APIs (e.g., Shelly, TP-Link, Zigbee) will be utilized to retrieve data into the system.

Outcome

The system will prove capable of obtaining live energy data from accessible IoT sources, which can be utilized to improve optimization strategies.

4. Data Security Implementation

Overview

In order to protect users' energy consumption data, initial data protection practices will be implemented with emphasis on secure transmission and storage.

Implementation

Encryption: All energy data will be encrypted during transit and at rest.

Access Control: Sensitive data will only be accessible to authorized users and system administrators.

Outcome

User data will be safely stored, with controls in place to prevent unauthorized access and ensure data privacy compliance.

5. Testing and Feedback Collection

Overview

The system will be tested initially with early adopters to assess usability, recommendation accuracy, and dashboard effectiveness.

Implementation

Pilot Testing: Pilot users will install the system and give feedback based on real-life usage.

Evaluation Metrics: System responsiveness, recommendation accuracy, and user satisfaction will be evaluated.

Outcome

User feedback insights will inform algorithm optimization improvements and dashboard usability enhancements in Phase 4.

Challenges and Solutions

Data Availability

Challenge: Inability to access real-time usage data across all users.

Solution: Utilization of simulated data and partnership with early adopters for real-world feedback.

Algorithm Performance

Challenge: Early models can over- or under-suggest optimizations.

Solution: Ongoing improvement through feedback and new datasets.

User Engagement

Challenge: Users might not always engage with the dashboard.

Solution: Add notifications and gamification to enhance engagement.

Results of Phase 3

Optimization Engine: Working model to detect energy inefficiencies and recommend improvements.

Interactive Dashboard: Users can see real-time and historical energy data with actionable insights.

IoT Integration: System designed to gather real-time data from energy monitoring equipment.

Secure Data Management: Encryption and limited access applied.

Initial User Testing: Feedback collected to fine-tune in Phase 4.

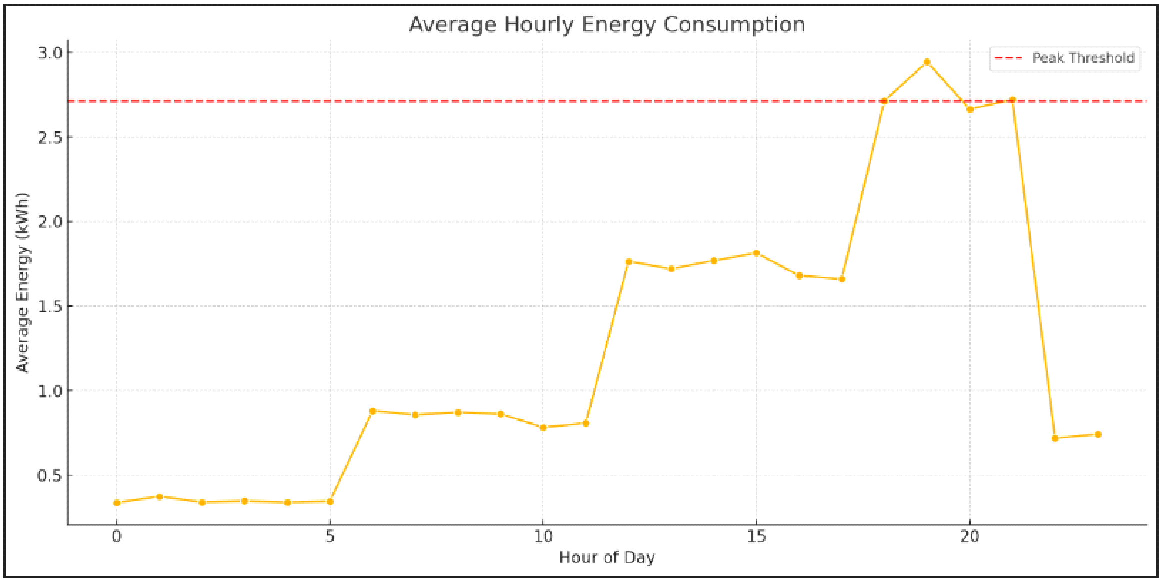
Next Steps for Phase 4

Model Refinement: Add more varied data to create more robust optimization suggestions.

User Customization: Provide personalized energy targets and scenario simulations.

Scalability and Expansion: Set up system for wider rollout and integration with utility companies

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta

np.random.seed(42)

start_date = "2025-04-01"
end_date = "2025-04-30 23:00:00"
date_range = pd.date_range(start=start_date, end=end_date, freq='H')

def simulate_usage(hour):
    if 0 <= hour < 6:
        return np.random.uniform(0.2, 0.5)
    elif 6 <= hour < 12:
        return np.random.uniform(0.5, 1.2)
    elif 12 <= hour < 18:
        return np.random.uniform(1.0, 2.5)
    elif 18 <= hour < 22:
        return np.random.uniform(2.0, 3.5)
    else:
        return np.random.uniform(0.5, 1.0)

usage_data = [simulate_usage(ts.hour) for ts in date_range]

df = pd.DataFrame({
    'Timestamp': date_range,
    'Energy_kWh': usage_data
})
df['Hour'] = df['Timestamp'].dt.hour
df['Day'] = df['Timestamp'].dt.date

hourly_avg = df.groupby('Hour')['Energy_kWh'].mean().reset_index()
top_hours = hourly_avg.sort_values(by='Energy_kWh', ascending=False).head(3)

plt.figure(figsize=(12, 6))
sns.lineplot(data=hourly_avg, x='Hour', y='Energy_kWh', marker='o')
plt.axhline(y=top_hours['Energy_kWh'].min(), color='r', linestyle='--', label='Peak Threshold')
plt.title('Average Hourly Energy Consumption')
plt.xlabel('Hour of Day')
plt.ylabel('Average Energy (kWh)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

def generate_recommendations(df, peak_hours):
    tips = []
    for hour in peak_hours['Hour']:
        tips.append(f"Reduce energy usage during {hour}:00 by shifting activities to off-peak hours or using efficient appliances.")
    return tips

recommendations = generate_recommendations(df, top_hours)

print("\nEnergy Optimization Recommendations:")
for tip in recommendations:
    print(f"- {tip}")
```



The hours with the highest average energy consumption are:

- **7 PM (Hour 19)**: ~2.94 kWh
- **9 PM (Hour 21)**: ~2.72 kWh
- **6 PM (Hour 18)**: ~2.71 kWh

These are typically high-activity times (cooking, lighting, electronics).