COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : aut113323aia27

ROLL NO : 113323243052

DATE : 03-05-2025

TECHNOLOGY-ENERGY USAGE OPTIMIZATION

SUBMITTED BY,

MARRIREDDY VYSHNAVI

TEAM MEMBERS

GOLLA PRATHIMA
GOPASANI VENKATA VAISHNAVI
KONDURU BHAVYA SREE

Phase 5: Project Demonstration & Documentation

Title: ENERGY USAGE OPTIMIZATION

Abstract:

The **AI-Powered Energy Usage Optimization System** aims to enhance energy efficiency and reduce consumption costs through intelligent data analysis, real-time monitoring, and smart device control.
Leveraging artificial intelligence, machine learning, and IoT technologies, the system continuously analyzes energy usage patterns, predicts future consumption, and provides actionable insights. In this final phase, the system integrates with smart meters, sensors, and ERP platforms to demonstrate live performance, scalability, and secure data handling. This report encompasses the complete system demonstration, detailed documentation, testing reports, and source code. The project ensures optimized energy consumption, cost savings, and environmental sustainability in industrial and residential settings. Visual materials such as screenshots, system diagrams, and codebase excerpts will be provided to support understanding.

INDEX

1. Project Demonstration

Overview:

The AI-Powered Energy Usage Optimization System will be presented to stakeholders, showcasing its real-time analytics, AI-driven recommendations, integration with IoT-enabled energy devices, and performance under dynamic load conditions.

Demonstration Details:

- **System Walkthrough:** Live demo from user login to real-time energy data monitoring and optimization recommendations.
- **AI Analytics Accuracy:** Showcases how the AI model identifies high-consumption patterns and suggests intelligent load scheduling.
- **IoT Integration:** Live data from sensors measuring power usage, temperature, and occupancy will be demonstrated.
- **Performance Metrics:** Includes latency, load testing results, and system behavior with multiple concurrent users.
- **Security & Privacy:** Demonstrates encryption methods, access controls, and secure storage of energy data.

Outcome:

By the end of the session, stakeholders will understand the system's ability to optimize energy usage, reduce operational costs, and adapt dynamically to usage patterns in real-world environments.

2. Project Documentation

Overview:

A detailed documentation set is provided to describe every element of the AI-Powered Energy Usage Optimization System, from system design and algorithms to administrative control and testing.

Documentation Sections:

- **System Architecture:** Diagrams outlining AI model components, IoT sensor networks, data pipelines, and integration with ERP or BMS (Building Management Systems).
- **Code Documentation:** Annotated source code covering AI training, real-time data ingestion, device control APIs, and user interface logic.
- **User Guide:** Instructions for homeowners or facility managers on using the system, reading insights, and adjusting settings.
- **Administrator Guide:** Guidance on system setup, maintenance, data calibration, and report generation.
- **Testing Reports:** Includes results from power consumption benchmarking, AI accuracy tests, and system stress testing.

Outcome:

The documentation ensures smooth system deployment and provides a roadmap for future enhancements or integrations.

3. Feedback and Final Adjustments

Overview:

Following the demonstration, feedback from project supervisors, technical experts, and test users will be collected and used to finalize system improvements.

Steps:

- **Feedback Collection:** Surveys and discussions post-demo to gather insight on usability, performance, and feature effectiveness.
- **Refinement:** Address identified issues, such as optimization delays, inaccurate recommendations, or UI difficulties.
- **Final Testing:** Conduct final evaluations on energy savings, system robustness, and user satisfaction.

Outcome:

The final version will be optimized for real-world deployment, providing reliable, accurate, and user-friendly energy management.

4. Final Project Report Submission

Overview:

The final report presents a comprehensive summary of the project, documenting its development process, performance outcomes, and long-term value.

Report Sections:

- **Executive Summary:** High-level description of project objectives, innovation, and key results.
- **Phase Breakdown:** Overview of system evolution from AI model training to IoT implementation and ERP integration.
- **Challenges & Solutions:** Documents issues such as data irregularities, model overfitting, and sensor malfunctions, with corresponding fixes.
- **Outcomes:** Details energy savings percentages, system stability, and deployment readiness.

Outcome:

A professional report will be submitted for evaluation and archival, covering all critical milestones and deliverables.

5. Project Handover and Future Works

Overview:

Details the official system handover and outlines potential enhancements for broader deployment or increased functionality.

Handover Details:

- **Next Steps:** Include AI enhancements for demand forecasting, integration with renewable energy sources, and voice-activated energy controls.

- **Scalability Plan:** Guidelines for deploying the system in large buildings, industrial setups, or smart cities.
- **Maintenance Guide:** Instructions for software updates, sensor recalibration, and AI model retraining.

Outcome:

The project will be officially delivered to stakeholders, along with complete documentation, installation support, and strategic recommendations for ongoing improvement.

Screenshots of source code and Working final project.

CODE :

```
[1]: 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)
```

```
# 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()
```

OUTCOME :