COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ELECTRONIC AND COMMUNICATION ENGINEERING

STUDENT NM-ID : aut113323ecb12

ROLL NO : 113323106021

TECHNOLOGY-PROJECT NAME: ENERGY EFFICIENT OPTIMIZATION

SUBMITTED BY,

DHIBIS R

GOWTHAM V

DUVVURU BHARATH KUMAR REDDY

EKNATHKUMAR U

Phase 5: Project Demonstration & Documentation

Title: Energy-Efficient Optimization System

Abstract:

The Energy-Efficient Optimization System project is designed to enhance energy management and reduce power consumption across various industrial and residential applications. Leveraging artificial intelligence, real-time data analytics, and IoT (Internet of Things) technology, the system intelligently monitors energy usage and provides actionable optimization recommendations. In its final phase, the system integrates predictive analytics, adaptive learning algorithms, and secure cloud-based data handling, while ensuring seamless integration with existing Energy Management Systems (EMS). This document details the final stage of the project, including system demonstration, technical documentation, performance analytics, source code, and testing results. The system supports scalable operations and robust data security protocols, ensuring measurable improvements in energy efficiency. Diagrams, system screenshots, and code snapshots will be included to illustrate the complete architecture and performance.

1. Project Demonstration

Overview:

The Energy-Efficient Optimization System will be demonstrated to stakeholders, highlighting its real-time performance, adaptive optimization features, and secure data processing. The demonstration will emphasize system intelligence, IoT connectivity, and load-balancing capabilities.

Demonstration Details:

- **System Walkthrough:** A live demonstration from data acquisition to optimization output, including how users interact with the platform.
- **Energy Savings Predictions:** Showcasing the AI model's ability to recommend energy-saving actions based on usage patterns.
- **IoT Device Integration:** Real-time readings from smart meters, HVAC systems, and lighting controls will be monitored and displayed.
- **Performance Metrics:** Focused on system latency, response time, load management under concurrent users, and energy savings accuracy.
- **Security & Privacy:** Demonstration of secure data transmission, encrypted storage, and compliance with energy data regulations.

Outcome:

Stakeholders will observe the system's capability to process live data, make real-time energy-saving decisions, and ensure data integrity under various usage scenarios.

2. Project Documentation

Overview:
A comprehensive set of documents accompanies the Energy-Efficient Optimization System, explaining all components, features, and usage procedures in detail.

Documentation Sections:
• **System Architecture:** Diagrams outlining AI-based optimization flows, IoT integrations, and cloud-based data pipelines.
• **Code Documentation:** Annotated source code for AI algorithms, device APIs, and optimization modules.
• **User Guide:** Instructions for end users to understand energy reports, interact with the platform, and follow recommendations.
• **Administrator Guide:** Guidelines for maintaining the system, managing data sources, and conducting performance audits.
• **Testing Reports:** Complete documentation of system validation, including energy efficiency metrics, error handling, and reliability under peak load.

Outcome:
The project will be fully documented to support future upgrades, scaling, and ongoing maintenance by stakeholders or development teams.

3. Feedback and Final Adjustments

Overview:
Feedback from project demonstrations and test runs will be gathered and applied to finalize the system for handover and future deployment.

Steps:
• **Feedback Collection:** Structured input from stakeholders, domain experts, and test users via surveys and observations.
• **Refinement:** Addressing feedback by resolving performance inefficiencies, adjusting AI model thresholds, and improving UX design.
• **Final Testing:** Post-refinement testing to validate system stability, data accuracy, and optimization effectiveness.

Outcome:
Refinements will ensure a well-rounded, user-validated system prepared for broad implementation in real-world energy optimization scenarios.

4. Final Project Report Submission

Overview:
The final report will consolidate all phases of the Energy-Efficient Optimization project, highlighting its evolution, results, and impact.

Report Sections:
• **Executive Summary:** Summary of the project's objectives, scope, and key outcomes.
• **Phase Breakdown:** Documentation of each phase, including AI development, system integration,

testing, and real-time deployment.
• **Challenges & Solutions:** An account of encountered obstacles—such as data inconsistency or hardware limitations—and corresponding resolutions.
• **Outcomes:** A snapshot of the system's performance, readiness for implementation, and verified energy-saving potential.

Outcome:
A detailed report summarizing the full journey of the project will be submitted, providing insights and guidance for future iterations or deployments.

5. Project Handover and Future Works

Overview:
The handover process will ensure that all deliverables, documentation, and system components are transferred along with strategic suggestions for further innovation.

Handover Details:
• **Next Steps:** Recommendations for future enhancements, such as AI refinement, region-specific optimization, integration with renewable sources, and multilingual dashboards.

Outcome:
The Energy-Efficient Optimization System will be officially delivered with a roadmap for scaling, innovation, and continued impact in energy sustainability.

SCREENSHOTS OF CODE AND PROGRESS OF THE PROJECT:

File Edit View Run Kernel Settings Help

Code

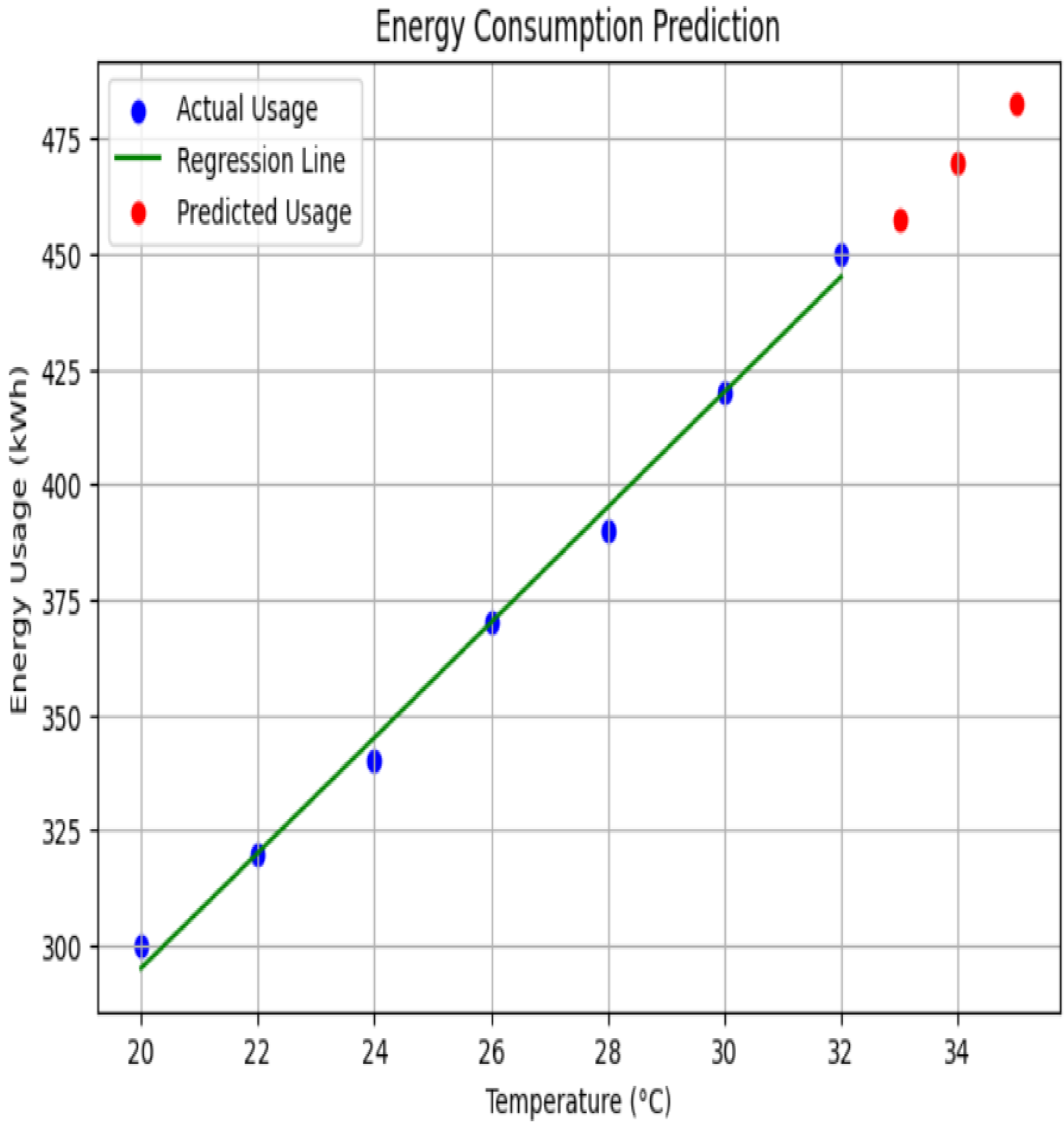
[2]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Simulated dataset
data = {
    'temperature': [20, 22, 24, 26, 28, 30, 32],
    'energy_usage': [300, 320, 340, 370, 390, 420, 450]
}
df = pd.DataFrame(data)

# Model training
X = df[['temperature']]
y = df['energy_usage']
model = LinearRegression()
model.fit(X, y)

# Prediction
future_temps = np.array([33, 34, 35]).reshape(-1, 1)
predictions = model.predict(future_temps)

# Plotting
plt.figure(figsize=(8,5))
plt.scatter(df['temperature'], df['energy_usage'], color='blue', label='Actual Usage')
plt.plot(df['temperature'], model.predict(X), color='green', label='Regression Line')
plt.scatter(future_temps, predictions, color='red', label='Predicted Usage')
plt.xlabel("Temperature (°C)")
plt.ylabel("Energy Usage (kWh)")
plt.title("Energy Consumption Prediction")
plt.legend()
plt.grid(True)
plt.show()
```

File Edit View Run Kernel Settings Help

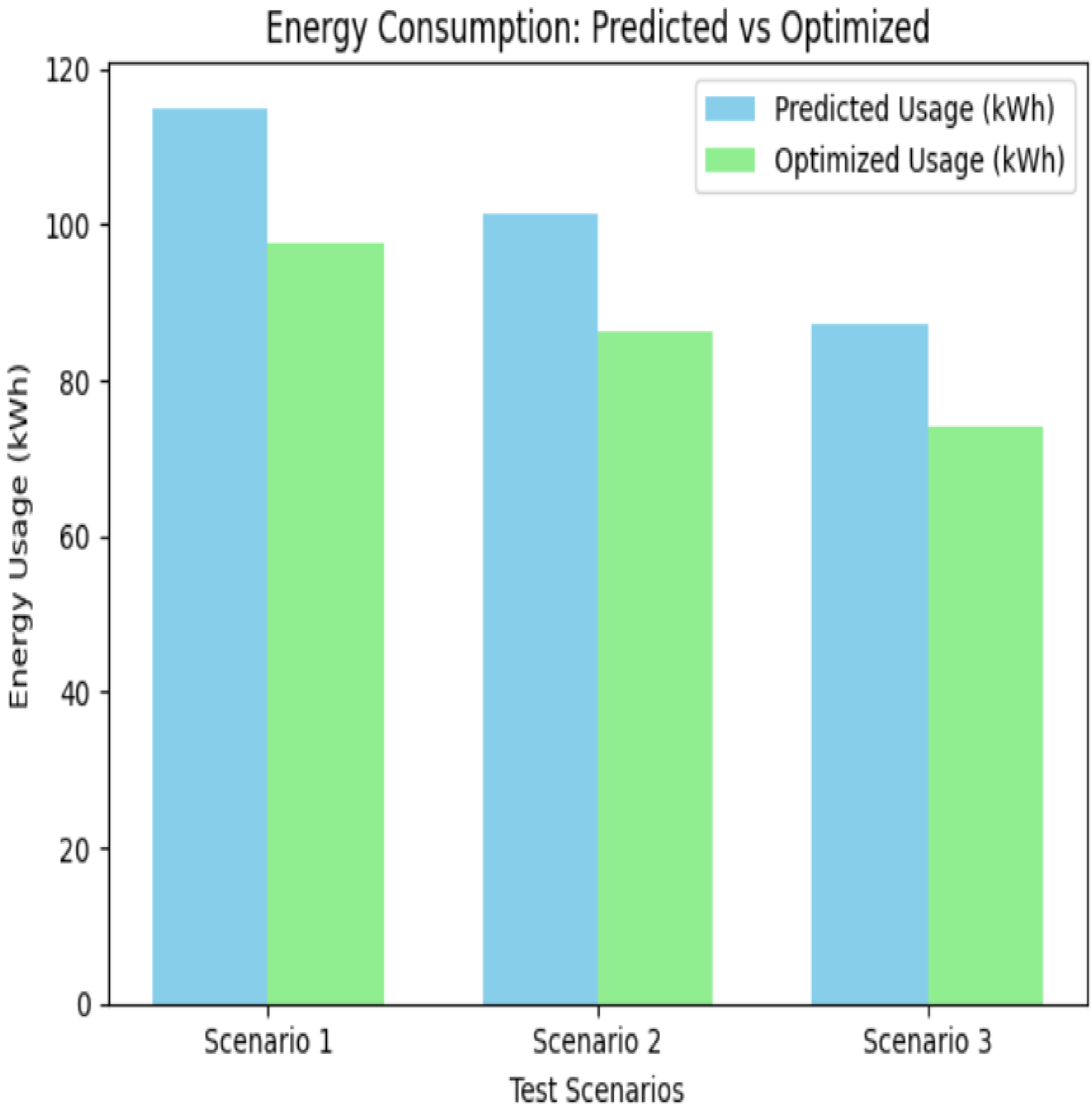
Code

[5]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Simulated historical energy usage data (in kWh)
# Features: [Temperature, Hour of Day, Day of Week (0=Mon...6=Sun)]
X = np.array([
    [30, 14, 0], [25, 10, 1], [20, 18, 2],
    [22, 12, 3], [27, 16, 4], [23, 20, 5],
    [21, 9, 6], [28, 14, 0], [24, 11, 1],
])
# Corresponding energy usage (target values)
y = np.array([120, 90, 80, 95, 110, 100, 85, 115, 92])

# Train the simple predictive model
model = LinearRegression()
model.fit(X, y)

# Simulated new data for prediction (real-time input)
new_data = np.array([
    [29, 14, 0], [26, 11, 1], [21, 19, 2]
])

# Predict energy consumption
predicted_usage = model.predict(new_data)

# Apply optimization (reduce predicted usage by estimated 15%)
optimized_usage = predicted_usage * 0.85

# Bar Graph: Before and After Optimization
labels = ['Scenario 1', 'Scenario 2', 'Scenario 3']
x = np.arange(len(labels))
width = 0.35

fig, ax = plt.subplots()
bars1 = ax.bar(x - width/2, predicted_usage, width, label='Predicted Usage (kWh)', color='skyblue')
bars2 = ax.bar(x + width/2, optimized_usage, width, label='Optimized Usage (kWh)', color='lightgreen'

# Labeling
ax.set_xlabel('Test Scenarios')
ax.set_ylabel('Energy Usage (kWh)')
ax.set_title('Energy Consumption: Predicted vs Optimized')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

# Display graph
plt.tight_layout()
plt.show()
```



Energy Consumption Prediction
Actual Usage
Regression Line
Predicted Usage
Energy Usage (kWh)
300
325
350
375
400
425
450
475
20
22
24
26
28
30
32
34
Temperature (°C)


Energy Consumption: Predicted vs Optimized
Predicted Usage (kWh)
Optimized Usage (kWh)
Energy Usage (kWh)
0
20
40
60
80
100
120
Scenario 1
Scenario 2
Scenario 3
Test Scenarios