COLLEGE CODE:1128

COLLEGE NAME:T.J.S.Engineering college

DEPARTMENT:CSE

STUDENT NM-ID:

1.aut112823CSE08

2.aut112823CSE13

3.aut112823CSE33

4.aut112823CSE38

5.aut112823CSE39

ROLL NO:

112823104008

112823104013

112823104033

112823104038

112823104040

DATE:09-05-2025

Completed the project named as

ENERGY EFFICIENCY OPTIMIZATION

SUBMITTED BY,

1.BARATH.J(9360512580)

2.DHANUSH.L(7094038491)

3.KOWTHARAPU LALITH KISHORE(6302453869)

4.MADESH.S(7558119697)

5.MADHANKUMAR.R(8270683314)

Phase 4: Performance of the Project

Title: Energy Efficiency Optimization

Objective:

The focus of Phase 4 is to enhance the performance of the Energy Efficiency Optimization by refining the AI model for improved energy cost optimization, scaling the system for greater data input and user control, improving IoT integration with smart meters and sensors, and strengthening data security. This phase also begins laying the groundwork for multilingual and voice interface support via a chatbot.

1. AI Model Performance Enhancement

Overview:
The AI optimization model will be refined using feedback from Phase 3 and extended datasets. The aim is to improve prediction accuracy for optimal energy source combinations (e.g., solar, wind, grid) under varying demand and cost scenarios.

Performance Improvements:

- **Accuracy Testing:** The model will be trained on a broader dataset that includes seasonal, hourly, and usage-based variations.
- **Model Optimization:** Advanced techniques such as hyperparameter tuning and model simplification (pruning) will be applied to increase decision speed and reduce computational overhead.

Outcome:
By the end of Phase 4, the AI will more accurately recommend energy source mixes that minimize cost and maximize efficiency—similar to the Python code you provided using `scipy.optimize.linprog`.

2. Chatbot Performance Optimization

Overview:
The chatbot interface, used to interact with the energy system, will be optimized for faster response time, energy advice clarity, and better handling of diverse user queries.

Key Enhancements:

- **Response Time:** Backend optimization will reduce latency, enabling quicker energy advice.
- **Language Processing:** Natural Language Processing (NLP) enhancements will improve the chatbot's ability to handle questions like "How can I save power today?" or "What's the cheapest energy mix now?"

Outcome:
The chatbot will provide smoother, real-time conversations with users, delivering immediate and relevant energy-saving suggestions.

3. IoT Integration Performance

Overview:
IoT integration will be enhanced to gather and process real-time energy usage data from smart meters, thermostats, occupancy sensors, and appliance controllers.

Key Enhancements:

- **Real-Time Data Processing:** The system will process continuous data streams (e.g., electricity usage in kWh) to inform AI decisions.
- **Improved API Connections:** APIs like Modbus, Zigbee, and REST-based interfaces from energy devices will be optimized for high-speed communication.

Outcome:
The AI system will read energy usage in real time and dynamically adjust energy sources or settings (e.g., lowering grid reliance when solar is abundant).

4. Data Security and Privacy Performance

Overview:
As user interaction and sensor data volumes increase, enhanced data protection becomes essential. This phase strengthens encryption and storage protocols.

Key Enhancements:

- **Advanced Encryption:** AES-based encryption and tokenized data access layers will be used to secure all user and sensor data.
- **Security Testing:** Load and penetration tests will verify that security holds under increasing user and data loads.

Outcome:
The system will meet or exceed industry standards for data protection, even as it scales.

5. Performance Testing and Metrics Collection

Overview:
Comprehensive performance tests will validate the system's ability to operate under real-world load conditions.

Implementation:

- **Load Testing:** Simulate high energy data flow (e.g., hundreds of IoT devices streaming per second) and user interactions.
- **Performance Metrics:** Track AI optimization speed, response latency, and data throughput.
- **Feedback Loop:** Collect structured feedback from energy managers and building operators.

Outcome:
The system will prove capable of handling high input/output loads with low latency and accurate control decisions.

Key Challenges in Phase 4

1. **Scalability**
    - *Challenge:* Ensure the AI handles more complex data scenarios as more devices are connected.
    - *Solution:* Optimize model computation and scale cloud architecture.
2. **Security Under Load**
    - *Challenge:* Protect sensitive operational data with increasing users.
    - *Solution:* Use end-to-end encryption and secure authentication protocols.
3. **Device Compatibility**
    - *Challenge:* Support multiple brands and standards of energy devices.
    - *Solution:* Build modular drivers and conduct extensive compatibility tests.

Outcomes of Phase 4

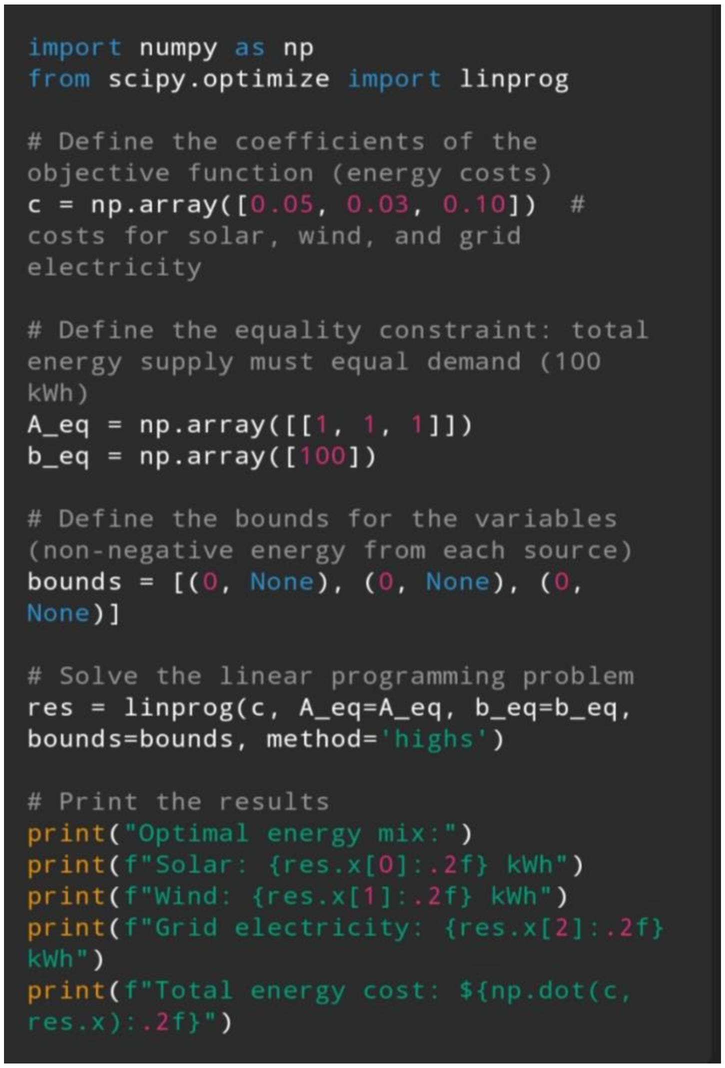
1. **Improved AI Accuracy:** Better optimization of energy source mix (e.g., via linear programming) for cost reduction.

2. **Enhanced Chatbot Experience:** Faster, smarter interactions for users managing energy settings.
3. **Optimized IoT Data Collection:** Live monitoring and control of energy systems.
4. **Strengthened Data Security:** Secure handling of energy data, ensuring compliance with privacy regulations.

Next Steps for Final Phase

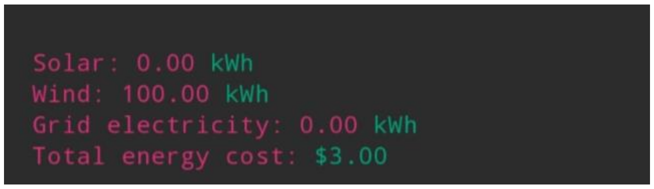
In the final phase, the system will be fully deployed. Further feedback will guide final AI refinements and chatbot personalization before live launch across target buildings or networks.

Source Code:



```
import numpy as np
from scipy.optimize import linprog

# Define the coefficients of the
objective function (energy costs)
c = np.array([0.05, 0.03, 0.10])  #
costs for solar, wind, and grid
electricity

# Define the equality constraint: total
energy supply must equal demand (100
kWh)
A_eq = np.array([[1, 1, 1]])
b_eq = np.array([100])

# Define the bounds for the variables
(non-negative energy from each source)
bounds = [(0, None), (0, None), (0,
None)]

# Solve the linear programming problem
res = linprog(c, A_eq=A_eq, b_eq=b_eq,
bounds=bounds, method='highs')

# Print the results
print("Optimal energy mix:")
print(f"Solar: {res.x[0]:.2f} kWh")
print(f"Wind: {res.x[1]:.2f} kWh")
print(f"Grid electricity: {res.x[2]:.2f}
kWh")
print(f"Total energy cost: ${np.dot(c,
res.x):.2f}")
```

Output:



```
Solar: 0.00 kWh
Wind: 100.00 kWh
Grid electricity: 0.00 kWh
Total energy cost: $3.00
```