COLLEGE CODE: 3126

COLLEGE NAME: THANGAVELU ENGINEERING COLLEGE

DEPARTMENT: B.E Computer science and engineering

STUDENTNM-ID -033796DB1B778F49BA74B49D1A88333E

ROLL NO - 312623104050

DATE: 14/05/2025

TECHNOLOGY PROJECT NAME: AI-EBPL Supply chain management

SUBMITTED BY,

NAME : VINOTHINI k

MOBILE NO: 7695989981

Phase 4: Performance of the project

Title : AI-Supply Chain Management

Objective:

The goal of Phase 4 is to enhance the system's performance by optimizing the AI models for improved demand forecasting, enhancing real-time inventory tracking via IoT integration, securing data transfers across the supply chain, and ensuring scalability under high transaction volumes.

1.AI Model Performance Enhancement

Overview:
The AI model used for demand forecasting and inventory optimization will be refined using updated data from previous phases.

Performance Improvements:

- **Accuracy Testing:** Retrain with a broader dataset including historical sales, seasonal demand, and real-time market dynamics.
- **Model Optimization:** Implement advanced machine learning techniques like hyperparameter tuning to boost model precision and speed.

Outcome:
Improved demand prediction accuracy and optimized stock levels across the supply chain with fewer shortages or overstock issues.

2. Chatbot Performance Optimization

Overview:
The chatbot interface used for supplier and customer queries will be enhanced to provide real- time updates on inventory status, order tracking, and delivery schedules.

Key Enhancements:

- **Response Time:** Optimized to handle high volumes of interactions during peak times.
- **Language Processing:** Enhanced NLP for better query understanding and initial support for multilingual interaction.

Outcome:
Faster and more intuitive communication, improving supplier coordination and customer satisfaction.

3. IoT Integration Performance

Overview:
Optimize integration with IoT-enabled logistics and inventory systems for real-time data capture (e.g., RFID tags, GPS in delivery trucks).

Key Enhancements:

- **Real-Time Data Processing:** Streamlined data capture from warehouses and transport vehicles.
- **API Improvements:** Enhanced connectivity with logistics platforms and smart warehouse tools.

Outcome:
Seamless visibility into supply chain operations, enabling proactive decision-making.

4. Data Security and Privacy Performance

Overview:
Ensure end-to-end security in data exchanges between vendors, manufacturers, logistics providers, and retailers.

Key Enhancements:

- **Advanced Encryption:** Implementation of encryption protocols for transactional and sensitive data.
- **Security Testing:** Simulate attacks and perform audits under heavy loads.

Outcome:
A highly secure system that complies with industry standards, ensuring data integrity throughout the supply chain.

5. Performance Testing and Metrics Collection

Overview:
Robust performance testing to verify system stability and responsiveness under increased demand and complexity.

Implementation:

- **Load Testing**: Simulate end-to-end supply chain scenarios involving high transaction volumes.
- **Metrics Collection:** Monitor processing speed, system uptime, and query success rate.

Feedback Loop: Gather operational feedback from suppliers and logistics partners.

Outcome:
A reliable, scalable SCM system ready for real-world deployment.

1. **Key Challenges and Solution Scalability**

- **Challenge**: Adapting to larger supplier networks and customer bases.
- **Solution:** Optimize architecture and cloud infrastructure.

2. **Security Under Load**

- **Challenge:** Ensuring encrypted data exchange under heavy transactional loads.
- **Solution:** Use robust protocols like TLS 1.3 and conduct regular audits.

3. **IoT Device Compatibility**

- **Challenge:** Managing various IoT hardware/software platforms.
- **Solution:** Use standardized APIs and conduct multi-device tests.

Final Outcomes of phase 4

- Accurate demand forecasting
- Efficient and user-friendly chatbot interaction
- Real-time logistics visibility
- High-level data security
- System scalability

Next Steps

Deploy the final version, collect operational data, and continuously improve based on live feedback.

SOURCE CODE:

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

# Simulated monthly sales data
months = np.array(range(1, 13)).reshape(-1, 1) # Jan to Dec
sales = np.array([120, 135, 150, 145, 160, 170, 175, 180, 190, 200, 210, 220]) # units sold per month

# Build a linear regression model model =
LinearRegression() model.fit(months,
sales)

# Predict next 3 months
future_months = np.array(range(13, 16)).reshape(-1, 1)
future_sales = model.predict(future_months)


# Combine for plotting
all_months = np.append(months, future_months) all_sales =
np.append(sales, future_sales)

# Plotting the data
plt.figure(figsize=(10, 6))
plt.plot(all_months, all_sales, marker='o', linestyle='-', label='Sales Forecast')
plt.axvline(x=12.5, color='red', linestyle='--', label='Forecast Start') plt.title('Monthly
Product Sales Forecast')
plt.xlabel('Month')
plt.ylabel('Units Sold')
plt.xticks(ticks=range(1, 16), labels=[ 'Jan',
    'Feb', 'Mar', 'Apr', 'May', 'Jun',
    'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',
    'Jan+1', 'Feb+1', 'Mar+1'
])
plt.grid(True) plt.legend()
plt.tight_layout() plt.show()
```

OUTPUT:


Monthly Product Sales Forecast
Sales Forecast
Forecast Start
Units Sold
240
220
200
180
160
140
120
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan+1
Feb+1
Month