COLLEGE CODE :3105

COLLEGE NAME : DHANALAKSHMI SRINIVASAN COLLOGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID : 7B6706D96D25E45A9B44E4B8E531E9A1

ROLL NO :310523104154

DATE :06.05.2025

Completed the project named as

AI-Supply Chain Management

SUBMITTED BY,

NAME : THARASHREE.R

MOBILE NO:6379313305

Phase 4: Performance of the Project

Title: AI-Supply Chain Management System

Objective

The focus of Phase 4 is to enhance the performance of the AI-based Supply Chain Management (SCM) system by improving prediction accuracy, optimizing inventory management algorithms, increasing the scalability of data processing pipelines, and strengthening system security. This phase also emphasizes real-time data integration from IoT sensors, improving supplier communication through AI chat interfaces, and preparing the system for multilingual support.

1. AI Model Performance Enhancement

Overview:

The AI model for demand forecasting and logistics optimization will be refined using enhanced datasets and feedback from prior phases. The goal is to improve the model's precision in predicting stock levels, demand surges, and logistics bottlenecks.

Performance Improvements:

- Demand Forecasting: Retraining the AI model with broader datasets including seasonal trends, economic factors, and market dynamics.
- Algorithm Tuning: Applied hyperparameter tuning and ensemble learning techniques for more reliable and faster forecasting.

Outcome:

The SCM AI system will deliver more accurate demand predictions and optimized delivery schedules, reducing overstocking and understocking incidents by a significant margin.

2. Chatbot Performance Optimization

Overview:

The AI chatbot used for vendor and logistics partner interactions will be optimized for faster and more contextual responses. It will also support multi-language input for broader usability.

Key Enhancements:

- Response Time: Improved backend architecture for quicker message processing.
- Contextual Awareness: Enhanced NLP models to handle context retention in conversations and region-specific supply chain terminology.

Outcome:

The chatbot will offer seamless communication across various supplier networks, improving coordination efficiency and enabling 24/7 automated interaction.

3. IoT Integration Performance

Overview:

Integration with IoT devices (e.g., RFID scanners, GPS trackers, warehouse sensors) will be optimized for real-time inventory tracking and environmental monitoring (e.g., temperature-sensitive goods).

Key Enhancements:

- Real-Time Tracking: Enhanced data collection from IoT endpoints with minimal latency.
- APIs and Protocols: Optimized data flow using lightweight protocols (MQTT, CoAP) for better reliability and speed.

Outcome:

Supply chain visibility will be significantly improved, enabling proactive responses to delays, spoilage risks, or inventory mismatches.

4. Data Security and Privacy Performance

Overview:

Robust security mechanisms will be ensured to protect sensitive supplier data, financial transactions, and real-time inventory flows.

Key Enhancements:

- Encryption Protocols: Upgraded to AES-256 for data-at-rest and TLS 1.3 for data-in-transit.
- Security Audits: Regular vulnerability assessments and penetration testing.

Outcome:

The system will ensure enterprise-grade data protection, complying with standards like ISO 27001 and GDPR where applicable.

5. Performance Testing and Metrics Collection

Overview:

End-to-end performance tests will evaluate system behavior under load, focusing on data processing speed, throughput, and fault tolerance.

Implementation:

- Load Testing: Simulations of high-order volumes and real-time data influx from thousands of devices.
- Metrics Collection: Capturing response times, system uptime, and AI decision latency.
- User Feedback: Gathering insights from supply chain managers and vendors for final-phase refinements.

Outcome:

The system will operate efficiently even under complex and high-volume conditions, proving its readiness for deployment in large-scale industrial environments.

Key Challenges in Phase 4

1. Scalability of AI Models
- Challenge: Maintaining accuracy with growing data streams.

- Solution: Modular AI architecture and distributed training pipelines.

2. Real-Time IoT Data Handling
- Challenge: Handling and processing millions of data points per hour.
- Solution: Implemented edge computing and stream processing frameworks (Apache Kafka, Spark Streaming).

3. Vendor System Compatibility
- Challenge: Integrating with various third-party systems.
- Solution: Standardized data formats (JSON, EDI) and RESTful APIs for interoperability.

Outcomes of Phase 4

1. Improved Forecasting Accuracy: Enhanced AI models provide better inventory and logistics predictions.
2. Optimized Vendor Communication: Streamlined interactions through a multilingual, AI-driven chatbot.
3. Real-Time Supply Chain Visibility: IoT integration ensures prompt issue detection and resolution.
4. Enterprise-Grade Security: Fully secured supply chain data ecosystem.

Next Steps for Finalization

In the next and final phase, the AI-SCM system will undergo full deployment with selected enterprise partners. Feedback from real-world usage will guide final adjustments

Python Code Implementation

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import random
import time
import hashlib
```

```
data = {'Month': [1, 2, 3, 4, 5, 6], 'Sales': [120, 150, 170, 200, 220, 260]}
df = pd.DataFrame(data)
model = LinearRegression()
model.fit(df[['Month']], df['Sales'])
future = pd.DataFrame({'Month': [7, 8, 9]})
predictions = model.predict(future)
print("Predicted Sales:", predictions)
plt.plot(df['Month'], df['Sales'], label='Actual')
plt.plot(future['Month'], predictions, label='Forecast', linestyle='--')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.legend()
plt.title('Demand Forecasting')
plt.show()

def simulate_iot_event():
    event = {
        'timestamp': time.time(),
        'temperature': round(random.uniform(10.0, 30.0), 2),
        'humidity': round(random.uniform(40.0, 80.0), 2),
        'device_id': f'Device-{random.randint(1, 10)}'
    }
    return event

for _ in range(5):
    event = simulate_iot_event()
    print(f"Received IoT data: {event}")
    time.sleep(1)

responses = {
    'order status': "Your order is currently being processed and will ship
soon.",
    'delay': "We apologize for the delay. We're actively resolving the issue.",
    'invoice': "Your invoice has been emailed to the registered address.",
```

```
    'default': "I'm sorry, could you please clarify your request?"
}

def chatbot(query):
    for keyword in responses:
        if keyword in query.lower():
            return responses[keyword]
    return responses['default']

while True:
    user_input = input("Vendor: ")
    if user_input.lower() in ['exit', 'quit']:
        break
    print("Bot:", chatbot(user_input))

def secure_transaction(transaction_id, amount, vendor):
    record = f"{transaction_id}|{amount}|{vendor}"
    hash_object = hashlib.sha256(record.encode())
    return hash_object.hexdigest()

transaction_hash = secure_transaction("TX12345", 1000, "VendorA")
print("Transaction Hash:", transaction_hash)
```

Performance Metrics Screenshot: