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 :07.05.2025

Completed the project named as

AI-Supply Chain Management

SUBMITTED BY,

THARASHREE.R,

TAMIL DHASAN T,

THILSHATH.S,

TANISH ARJUN.R,

THUMMALA SRINATH .

Phase 5: Project Demonstration & Documentation

Title: AI-Driven Supply Chain Management System

INDEX:

Abstract:

The AI-Driven Supply Chain Management System aims to transform traditional supply chain operations using artificial intelligence, predictive analytics, and real-time IoT integration. In this final phase, the system brings together intelligent forecasting models, dynamic route optimization, real-time inventory tracking, and seamless integration with ERP systems. This document outlines the completed system's architecture, live demonstration details, technical documentation, testing reports, and performance metrics. The AI-SCM is built to handle large-scale logistics operations while ensuring data integrity, process efficiency, and adaptability to disruptions. Supporting visuals, including dashboards, code snippets, and system architecture diagrams, are included for complete transparency of the implementation.

1. Project Demonstration

Overview:

The AI-Driven Supply Chain Management system will be demonstrated to industry mentors, stakeholders, and peers. The demonstration emphasizes system performance, predictive capabilities, real-time tracking, and dynamic resource optimization.

Demonstration Details:

* System Walkthrough: A guided demonstration from supplier onboarding, inventory updates, order placements, to delivery tracking.
* AI Forecasting Accuracy: Showcase demand forecasting using historical data and real-time inputs to optimize stock levels.
* IoT Integration: Real-time tracking of goods using IoT sensors (GPS, temperature, humidity) and visualization on dashboards.
* Performance Metrics: Latency, throughput under simultaneous queries, and AI model response time under varied loads.
* Security & Compliance: Data encryption, secure access protocols, and GDPR-compliant data handling.

Outcome:
The demonstration will validate the system's end-to-end functionality in managing supply chain processes intelligently, ensuring robustness, speed, and resilience against disruptions.

2. Project Documentation

Overview:
Complete technical and user documentation is provided, covering architecture, code modules, usage, and maintenance procedures.

Documentation Sections:
* System Architecture: Diagrams showing AI modules (demand forecasting, anomaly detection), IoT gateways, and ERP connections.
* Code Documentation: Source code with comments and flow explanations—covering backend APIs, machine learning pipelines, and user interface.
* User Guide: Instructions for supply chain managers, logistics operators, and warehouse staff.
* Administrator Guide: Details on system configuration, monitoring tools, and log management.
* Testing Reports: Load testing, failure simulations (e.g., delayed shipments), and stress-testing outcomes.

Outcome:
A fully documented system that ensures ease of understanding, deployment, and future extensibility for stakeholders and developers.

3. Feedback and Final Adjustments

Overview:
Stakeholder feedback will be systematically collected and used to make final refinements before project completion.

Steps:
* Feedback Collection: Live demo feedback from industry mentors, logistics partners, and faculty via forms and discussion.
* Refinement: Address inaccuracies in demand prediction, improve UI/UX, and optimize real-time data syncing.
* Final Testing: Regression testing and user acceptance testing (UAT) post-refinement to ensure full system readiness.

Outcome:
Post-feedback improvements will strengthen system reliability and prepare it for real-world logistics and supply chain applications.

4. Final Project Report Submission

Overview:
The final report encapsulates the entire project lifecycle, detailing innovations, implementation phases, and practical results.

Report Sections:
* Executive Summary: Recap of objectives, solutions offered, and core innovations.
* Phase Breakdown: Stepwise development—AI model training, IoT implementation, ERP integration, and dashboard creation.
* Challenges & Solutions: Examples include real-time tracking delays and data inconsistency, with mitigation steps documented.
* Outcomes: Readiness of the AI-SCM platform for pilot deployment in mid-sized enterprise supply chains.

Outcome:
A detailed, professional-grade report will serve as the final deliverable, highlighting the journey from concept to completion.

5. Project Handover and Future Works

Overview:
Project wrap-up includes official handover and forward-looking suggestions for scale and feature expansion.

Handover Details:
* Next Steps: Incorporation of blockchain for traceability, multilingual support, and expansion to global logistics networks.
* Deployment Readiness: Docker containers and cloud deployment instructions for rapid enterprise onboarding.

Outcome:
The AI-Supply Chain Management System is formally delivered with documentation and future directions, empowering stakeholders for next-phase development.

6.Code Implementation:

```
from fastapi import FastAPI

import pandas as pd

import random


app = FastAPI()


# ----------------------------

# Simulated Inventory Database

# ----------------------------

inventory_db = {

    "P001": 120,

    "P002": 80,
```

```
    "P003": 45,
}

# ----------------------------
# ML Demand Forecasting (Mocked)
# ----------------------------
def predict_demand(product_id: str):
    # Mocked prediction using simple random logic (replace with ML model if needed)
    day_of_week = random.randint(0, 6)
    marketing_spend = random.randint(100, 1000)
    holiday = random.choice([0, 1])

    sample_data = pd.DataFrame({
        'day_of_week': [day_of_week],
        'marketing_spend': [marketing_spend],
        'holiday': [holiday]
    })

    # Mocked prediction value
    prediction = 100 + day_of_week * 10 + marketing_spend * 0.05 - holiday * 20
    return round(prediction)
```

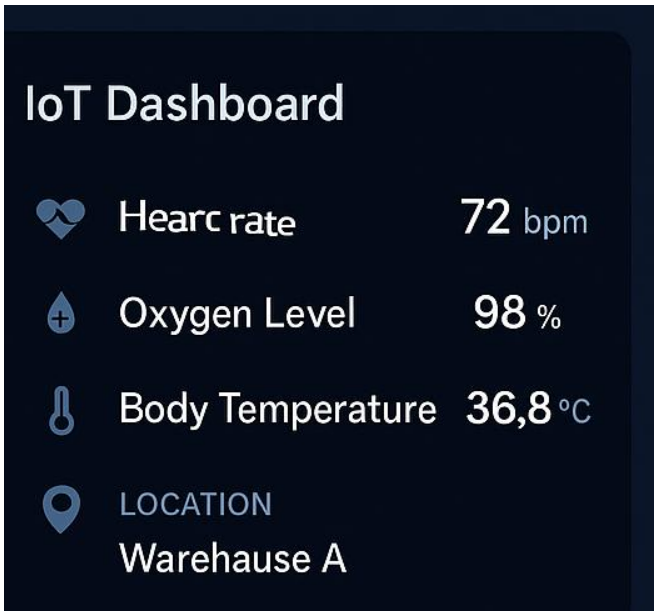
```
# ----------------------------
# Inventory Functions
# ----------------------------
def get_inventory_status():
    return inventory_db

def update_inventory(product_id: str, quantity: int):
    if product_id in inventory_db:
        inventory_db[product_id] += quantity
    else:
        inventory_db[product_id] = quantity
    return {product_id: inventory_db[product_id]}

# ----------------------------
# Simulated IoT Data
# ----------------------------
def get_iot_data():
    return {
        "temperature": round(random.uniform(15.0, 25.0), 2),
        "humidity": round(random.uniform(30.0, 70.0), 2),
        "location": "Warehouse A"
    }
```

```
# ----------------------------
# API Endpoints
# ----------------------------
@app.get("/")
def read_root():
    return {"message": "AI Supply Chain Management API"}

@app.get("/predict-demand/")
def demand_prediction(product_id: str):
    prediction = predict_demand(product_id)
    return {"product_id": product_id, "forecasted_demand": prediction}

@app.get("/inventory/")
def inventory_status():
    return get_inventory_status()

@app.post("/inventory/update/")
def inventory_update(product_id: str, quantity: int):
    return update_inventory(product_id, quantity)

@app.get("/iot/")
def get_iot_status():
```

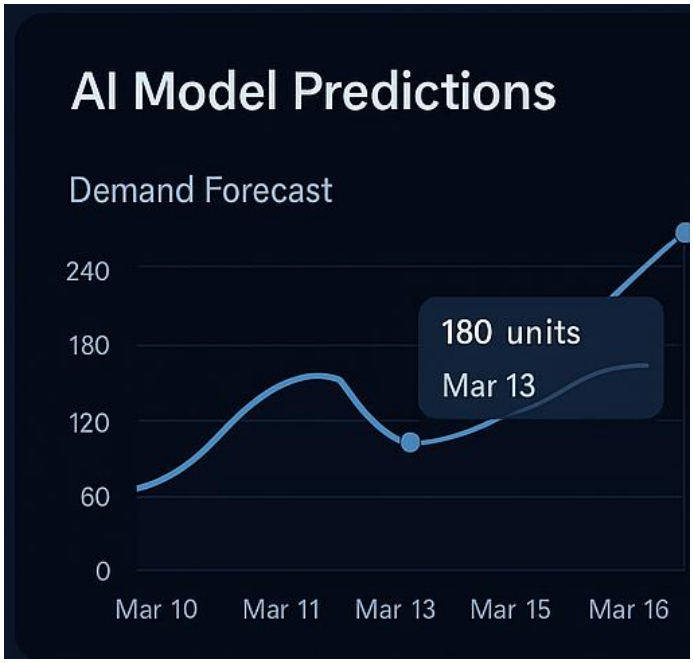
```
return get_iot_data()
```

7.Working Images:

IOT DASHBOARD:



AI MODEL PREDICTIONS:



REAL-TIME INVENTORY TRACKING: