AI Developer Assignment

Title: Build a Prototype AI Feature for Supply Chain Intelligence
Track Options (choose one)

- Demand Forecasting
- Spoilage Prediction
- ETA Prediction

Objective

Create a simplified ML-based pipeline (data ingestion → preprocessing → model training → prediction) for one supply chain use case. The goal is to evaluate your end-to-end understanding of applied ML — not to perfect the model.

Duration: 3 Days

Estimated Time: 4–6 hours of focused work

What to Build

Choose ONE of the following tracks and complete the steps below.

Option A: Demand Forecasting

Problem: Predict next 7-day SKU-level demand based on past order data.

Input:
CSV with 60 days of order data
Example format:

order_date, sku_id, location, quantity
2024-03-01, mango123, Mumbai, 100

Output:
A JSON or CSV file with forecasted quantity for each of the next 7 days
Example:
{
"sku_id": "mango123",
"location": "Mumbai",
"forecast_next_7_days": [105, 110, 95, 100, 98, 102, 105]
}

Option B: Spoilage Prediction

Problem: Predict the probability that a shipment will spoil based on transit time and temperature logs.

Input:
CSV with logs like
shipment_id, sku_id, transit_hours, avg_temp, shock_events, spoilage_flag
SHP-001, banana78, 16, 29.5, 1, 1

Output:
Train a binary classification model (spoilage_flag) and return:

- Accuracy or ROC-AUC score
- Prediction sample for a new shipment
- Optional summary:
  "This shipment has an 82% spoilage risk due to high temperature and long transit."

Option C: ETA Prediction

Problem: Estimate expected delivery time for a shipment based on past trip data.

Input:
route_id, distance_km, vehicle_type, weather, load_type, actual_eta_hours
R1, 320, van, rain, light, 10.5

Output:

- Train a regression model to predict actual_eta_hours
- Predict ETA for a test input
- Provide a formatted output:
  ```
  {
  "route_id": "R1",
  "predicted_eta_hours": 9.8,
  "confidence": "±1.2 hrs"
  }
  ```

What to Submit

1. Python scripts or Jupyter Notebook
2. CSVs or JSONs used/generated
3. README with:
   - Overview of your approach
   - Model choice and reason
   - Assumptions made
4. Flask/FastAPI endpoint or Streamlit app to demo

Evaluation Criteria

- Code structure and readability
- Data preprocessing and feature engineering
- Reasoning behind model choice
- Output clarity (and optional summarization)
- API/UI integration or interpretability