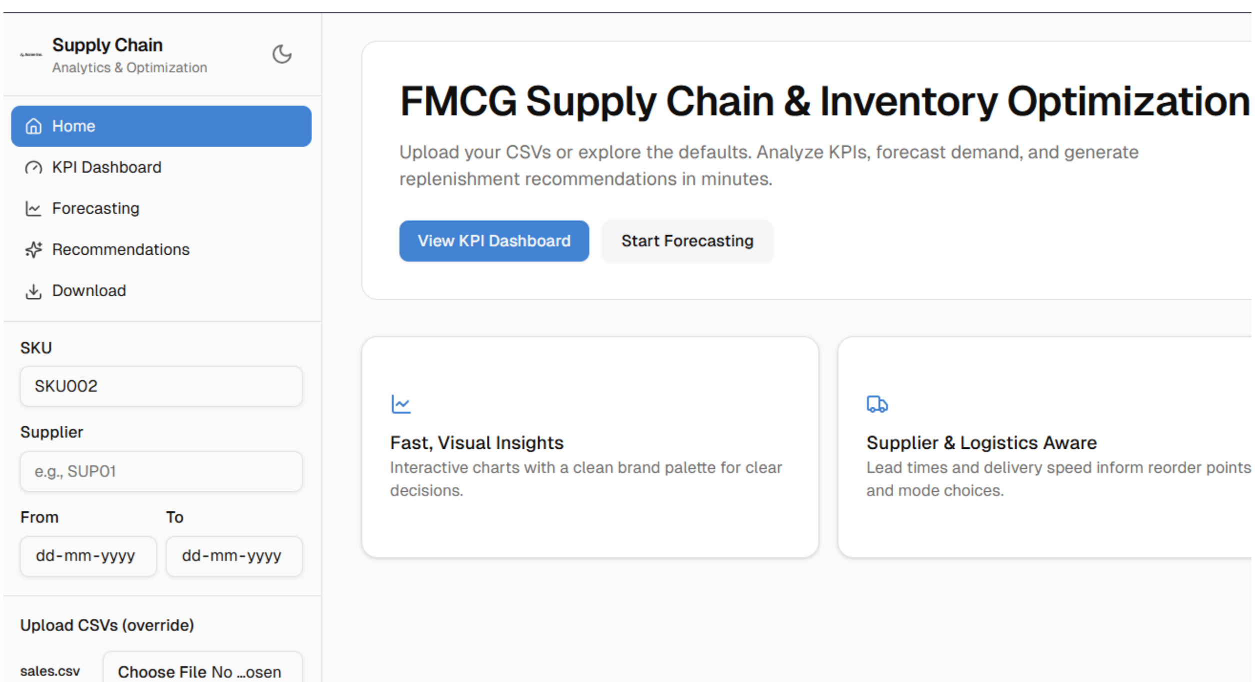
FMCG Supply Chain & Inventory Optimization

Project Overview

Objective:
To build a **data-driven solution** for FMCG companies that optimizes inventory, procurement, and logistics operations by leveraging **data analysis, forecasting, and recommendations**.

Key Goals:

- Analyze historical sales, inventory, procurement, and logistics data
- Identify performance metrics via KPIs
- Forecast future demand per SKU
- Generate actionable reorder recommendations
- Suggest suppliers and transport modes based on urgency, cost, and lead time



Datasets

This project uses **four key datasets** that cover sales, inventory, procurement, and logistics. Each dataset plays a vital role in building the optimization and forecasting model.

1. Sales Dataset (`sales.csv`)

- Contains **historical sales data** for each SKU.
- Helps in demand forecasting and KPI tracking.
- **Key Columns:**
  - `sku_id` → Unique product identifier
  - `date` → Transaction date
  - `sales_value` → Sales amount/value on that day

2. Inventory Dataset (`inventory.csv`)

- Represents **stock levels** for each SKU.
- Used to calculate stock turnover and reorder points.
- **Key Columns:**
  - `sku_id` → Unique product identifier
  - `opening_stock` → Stock at the beginning of the period
  - `closing_stock` → Stock at the end of the period
- **Derived Metric:**
  - `avg_daily_stock = (opening_stock + closing_stock) / 2`

3. Procurement Dataset (`procurement.csv`)

- Provides details about **suppliers and lead times**.
- Crucial for calculating reorder points and supplier performance.
- **Key Columns:**
  - `sku_id` → Unique product identifier
  - `supplier` → Supplier name or ID
  - `lead_time_days` → Delivery lead time in days

4. Logistics Dataset (`logistics.csv`)

- Captures data about **transportation modes, costs, and delivery times**.
- Supports cost optimization and transport mode recommendations.
- **Key Columns:**
  - `transport_mode` → e.g., Road, Rail, Air
  - `cost` → Transport cost for the trip
  - `distance_km` → Distance covered in kilometers
  - `delivery_time_days` → Average delivery duration
- **Derived Metric:**
  - `cost_per_km = cost / distance_km`

Data Preprocessing & EDA

Steps Performed:

1. Checked for missing values and duplicates
2. Converted date columns to datetime format
3. Normalized categorical variables where necessary (e.g., transport mode)
4. Computed new metrics:
    - Average stock per SKU
    - Cost per km for transport
    - Average lead time per supplier

Visualizations:

- Bar charts for **average inventory per SKU**
- Horizontal bar charts for **average lead time per supplier**
- Bar charts for **average logistics cost per km by mode**
- Delivery time comparisons by transport mode

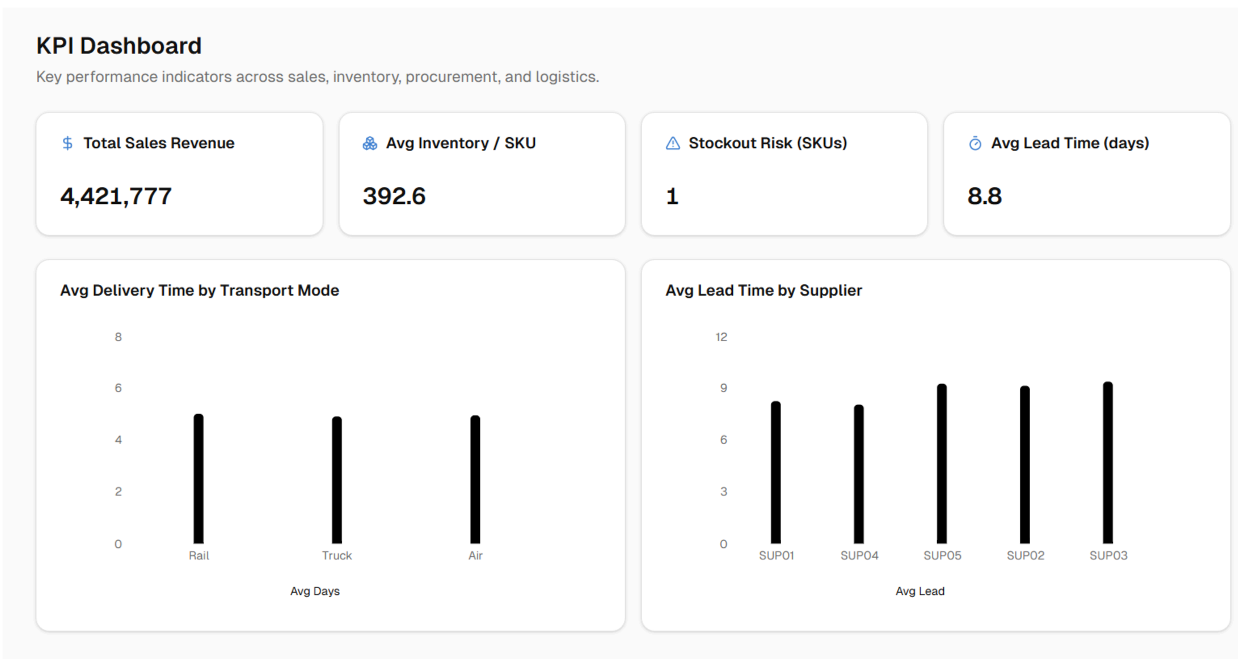
Insights:

- Identified SKUs with **high inventory or stockouts risk**
- Suppliers with **long lead times**
- Transport modes with **high cost vs delivery time trade-off**

Key Performance Indicators (KPIs)

To evaluate supply chain efficiency, the following KPIs were calculated:

- **Total Sales → Overall revenue from all SKUs (`sum(sales_value)`)**
- **Avg Inventory per SKU → Average stock level across SKUs (`mean(avg_daily_stock)`)**
- **Stockout Risk → Identified SKUs with stock below safety threshold (`avg_daily_stock < reorder_point`)**
- **Avg Lead Time → Supplier efficiency (`mean(lead_time_days)`)**
- **Avg Delivery Time → Logistics performance (`mean(delivery_time_days)`)**
- **Avg Cost per KM → Transportation cost efficiency (`mean(cost_per_km)`)**



Forecasting

- **Method: Time-series forecasting using Facebook Prophet.**
- **Objective: Forecast sales per SKU for the next 30 days.**

- **Steps Taken:**
    - **Grouped sales data by SKU & date.**
    - **Renamed columns for Prophet: `ds` (date), `y` (sales).**
    - **Trained Prophet model per SKU.**
    - **Generated future dataframe for 30 days.**
    - **Plotted forecasts with confidence intervals.**
- **Output:**
    - **Line charts of predicted sales per SKU.**
    - **Aggregate demand predictions for all SKUs.**

Reorder Point & Recommendations

- **Steps Taken:**
    1. **Merged forecasted demand, inventory, and supplier lead times.**
    2. **Calculated reorder point for each SKU.**
    3. **Checked if `current stock < reorder point` → order needed.**
    4. **Recommended order quantity = reorder point – current stock.**

Sample Output Table:

SKU ID	Supplier	Avg Demand	Stock	Lead Time	Reorder Point	Order Qty	Order Needed
SKU_1	Supplier_A	40	120	5	240	120	Yes

Transport Mode Recommendation

- **Rule-Based Approach:**
  - **High urgency (stockout risk) → fastest mode (Air)**
  - **Low urgency → cheapest mode (Rail/Road)**
- **Advanced Option: Scoring-based optimization balancing cost vs. delivery time using weighted scores.**

Sample Output:

SKU ID	Urgency	Transport Mode
SKU_1	High	Air
SKU_2	Low	Rail

Interactive Dashboard

- **Features:**
  - **Sidebar navigation: *Home, KPI Dashboard, Forecasting, Recommendations***
  - **KPI cards with metrics like *Total Sales, Avg Inventory, Lead Time***
  - **Interactive charts (bar, line, pie) for sales & logistics analysis**
  - **Recommendations table with urgency highlighting**
  - **Option to download results as CSV**
- **Visual Enhancements:**
  - **Color-coded KPIs**
  - **Filters for SKU, Supplier, Date Range**

Tools & Libraries

- **Python Libraries: pandas, numpy, matplotlib, seaborn**
- **Dashboard/UI: Streamlit**
- **Environment: Jupyter Notebook / Google Colab / PyCharm**

Conclusion

This project delivers a decision-support system for FMCG supply chain management, enabling:

- **Accurate demand forecasting**
- **Reorder point calculation to minimize stockouts**
- **Supplier and transport mode optimization**
- **Interactive dashboards for real-time insights**

Impact:

- **Improved operational efficiency**
- **Reduced overstock and understock costs**
- **Data-driven decision-making for procurement & logistics**

Links:

- **Google Colab Analysis: Colab Notebook**
- **Live UI: Web Link**
- **Github : Check Here**