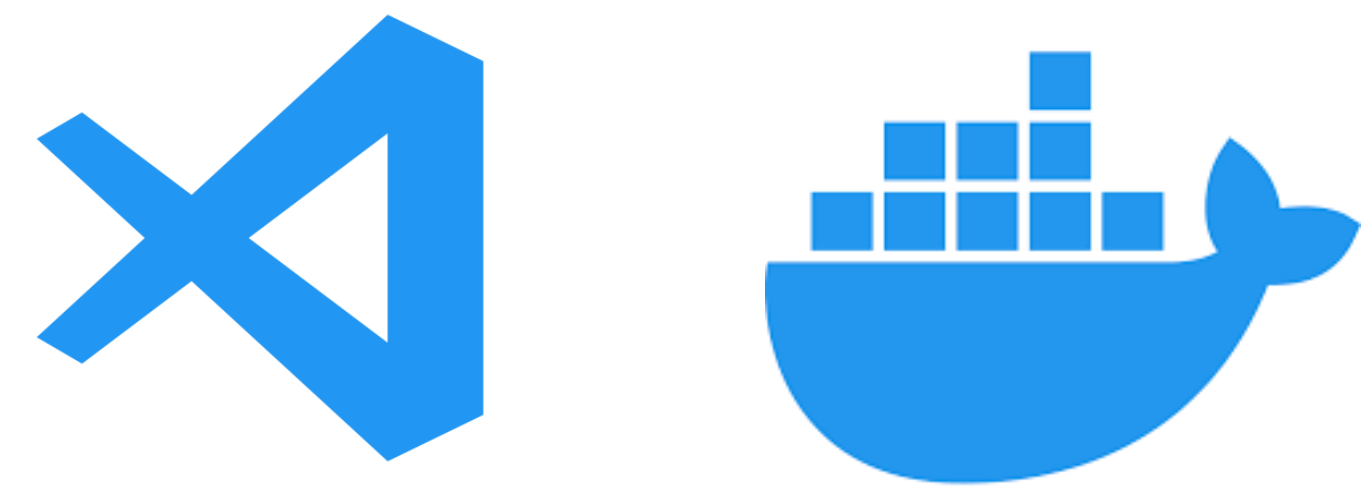
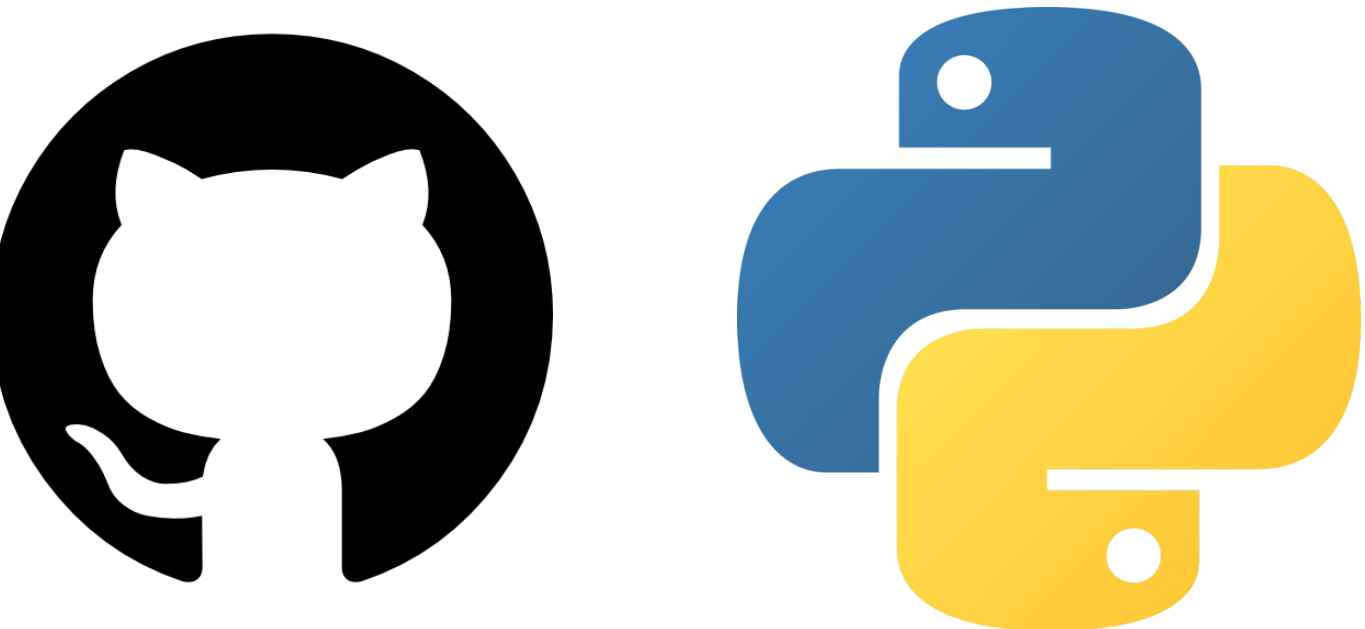


AI Driven Warehouse Stock Management

Students: Angela Banov, Vianne Novo Grifol, Steven S. Palacios, Matthew Cobos, Momin Salar Khan
Mentor: Ravi Venugopal, Giggso | **Instructor:** Dr. Masoud Sadjadi, Florida International University



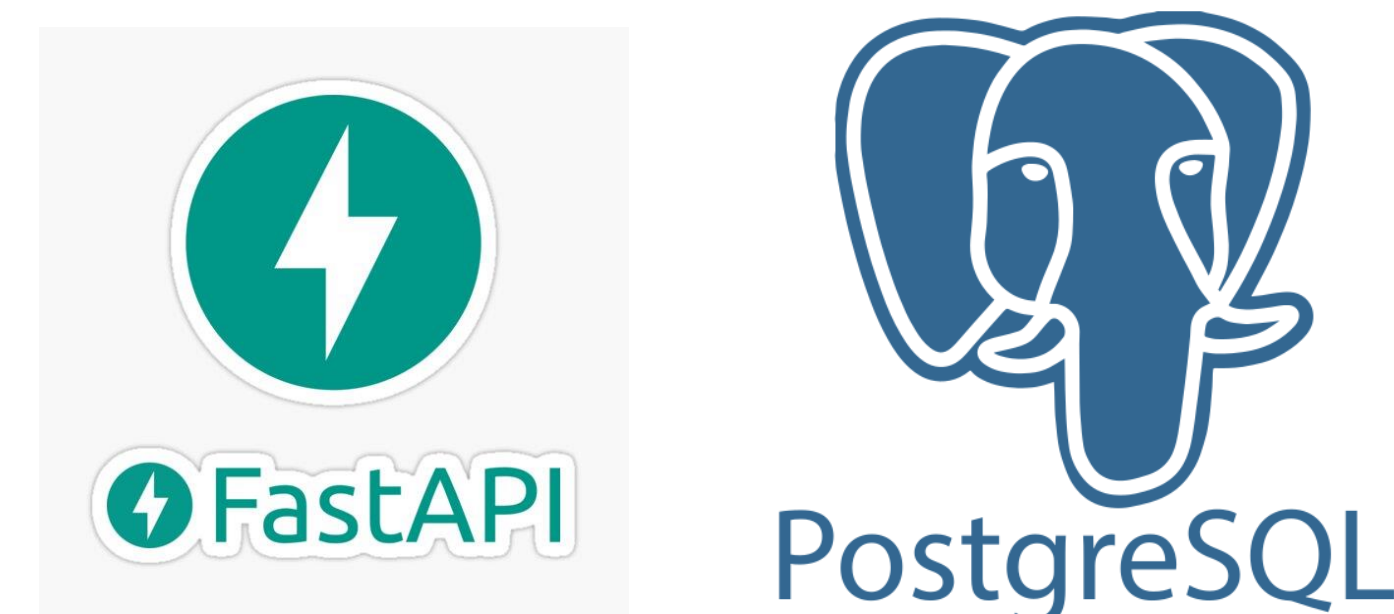
SQLAlchemy



Pydantic



Streamlit



pandas



Problem

Many warehouses struggle to balance inventory across distribution centers. One site may hold excess stock while another faces shortages or stockouts, which increases costs, slows fulfillment, and reduces customer satisfaction. Traditional systems provide static inventory views but do not monitor conditions continuously or recommend what to move and when. Teams must analyze data manually, which is slow, reactive, and prone to error. Our project creates a smarter assistant that tracks inventory in real time and highlights actionable transfers, so products reach the right place at the right time.

Current System

ERP and WMS platforms provide raw inventory data but do not guide decision making. Staff must manually determine product availability, shipping location, and fulfillment priority. No automated detection of overstock or shortage across warehouse networks. No system-level recommendations for transfers or redistribution. Results in slow responses, inefficient stock usage, and reduced service reliability.

Requirements

- The system must reach at least ninety percent accuracy when checking product availability and when recommending inventory transfers.
- The platform must connect to an operational data store so it can access orders, inventory levels, product details, and warehouse information in real time.
- Users must have a web dashboard where they can review transfer suggestions, explore current stock conditions, and approve actions with a single click.
- The technical stack needs to be realistic for engineering teams, which is why we are using Python, FastAPI, PostgreSQL, and Streamlit.
- The project will receive continuous guidance from our sponsor through daily check-ins, code reviews, and ongoing feedback to ensure alignment with real supply chain workflows.

Object Design

- Data Model mirrors real warehouse operations
- Inventory links products to warehouses and track quantities
- Products and warehouses store core attributes for analysis
- Orders and order items capture customer purchases and revenue
- Spare parts references main products with availability and compatibility details
- Pydantic schemas ensure consistent data formats across the API endpoint and future implemented designs.

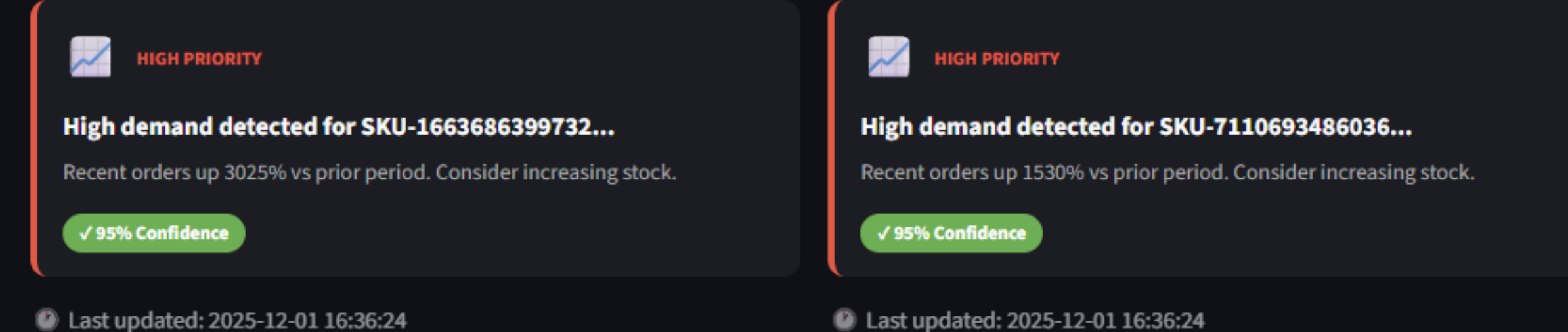
AI-Driven Warehouse Stock Management

Executive overview with KPIs, category analytics, and operational performance.

Total Inventory	Processing Orders	Total Products
671,787	227	500

AI Smart Suggestions

AI-powered insights updated in real-time



Warehouse AI Assistant

Powered by LangChain & Groq (FREE) - Ask me anything about your inventory!

Hi! I'm your AI warehouse assistant. I can help you with:

- Order statistics and trends
- Inventory levels and SKU information
- Product performance analysis
- Forecasting insights

What would you like to know?

Show me recent order trends

Based on the provided data, here are some recent order trends:

- Total Orders:** 1000 orders were placed between 2025-06-07 and 2025-10-05.
- Average Order Value (AOV):** The average order value is approximately \$14,228.40.
- Order Distribution:** Orders were shipped from 5 different warehouses.
- Top SKUs:** The top 5 SKUs by ordered quantity are:
 - SKU-5203141742685 (51 units)
 - SKU-3486753002282 (46 units)
 - SKU-1218499873823 (42 units)
 - SKU-9304489678738 (41 units)
 - SKU-3852502915067 (40 units)

Implementation

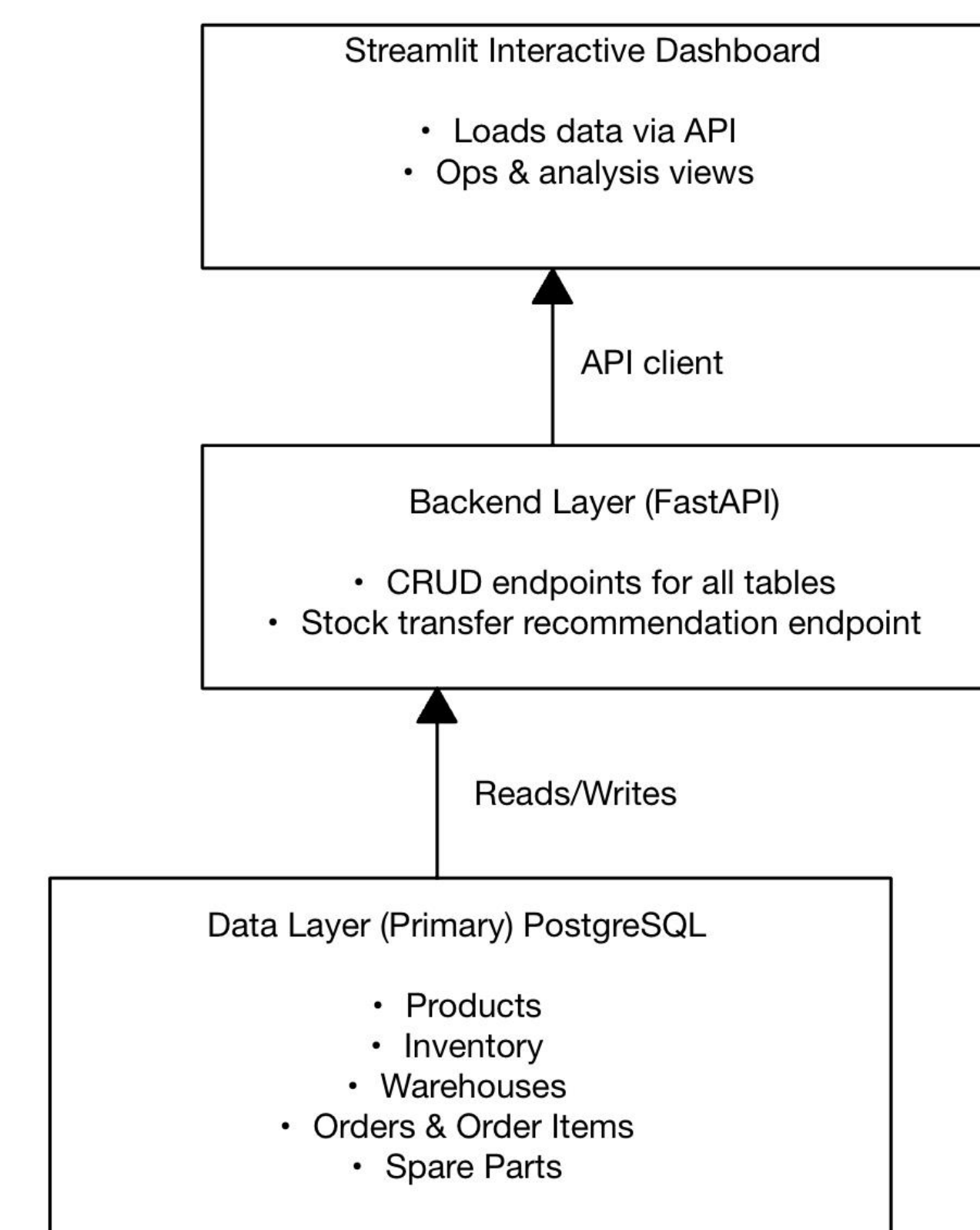
Built a **Python-based stack** running locally with **Docker**.

- Backend:**
 - FastAPI + SQLAlchemy** connecting to **PostgreSQL**
 - Defines models for inventory, products, warehouses, orders, and spare parts
 - Custom **Warehouse API** client returns each dataset as a **Pandas DataFrame**
- Frontend:**
 - Multi-page **Streamlit** dashboard
 - Shows KPIs, category analytics, and warehouse-level views
 - Offers search, filtering, and drill-down into specific products, locations, and order statuses
 - Uses **Altair** for charts and interactive visualizations
- AI & Forecasting:**
 - Forecasting engine using **SARIMA**, **ETS**, and **Prophet**
 - AI module generates smart suggestions:
 - high-demand detection
 - restock alerts

Verification

- We validated the system for **data accuracy** and **usability** using realistic mock datasets.
- Confirmed all **backend API endpoints** return valid results and that the **Streamlit client connects without errors**.
- In the ledger service, tested **receive, ship, and adjust events** to ensure **on-hand and reserved quantities update correctly**.
- On the dashboard, verified **product counts, total inventory, and warehouse totals** match the database.
- Tested **search and filter functionality** for fast product lookup across the network.
- Compared outputs to the source **mock dataset** and refined joins and calculations to **maintain consistency and reliability**.

System Design



Summary

Our project establishes the foundation for an **AI-driven warehouse stock management platform** designed for intelligent, data-guided decision making. We developed a **working PostgreSQL database and FastAPI backend** that expose clean endpoints for **inventory, products, warehouses, orders, and spare parts**. On top of this, we implemented a **Streamlit dashboard** that visualizes key operational metrics and enables users to explore stock conditions across distribution centers. While this phase focused on **data integration, backend services, and visualization**, the architecture was intentionally designed with **agent orchestration and optimization capabilities** in mind. In future phases, the static recommendation endpoint will evolve into an **autonomous agent that monitors events, forecasts demand, and proposes high-accuracy transfers**, enabling companies like Medula to shift from manual actions to **proactive, network-wide inventory balancing**.

Acknowledgement

The material presented in this poster is based upon the work supported by Florida International University. We thank **Dinesh Cyril Selvaraj** and **Sriram A** for their assistance, cooperation and mentorship that we received throughout this process.