Inventory and Warehouse Management System

# Abstract

The Inventory and Warehouse Management System is designed to facilitate real-time tracking and management of product inventories across multiple warehouses. This project showcases a prototype for a larger-scale solution, integrating Python as a frontend interface with MySQL for backend data storage and operations. It includes features such as low-stock alerts, stock distribution visualization, and stock transfer functionalities.

# Tools Used

- MySQL  
- Python (mysql-connector-python, SQLAlchemy, pandas, matplotlib, seaborn)  
- dbdiagram.io for database visualization

# Objective

To design and implement a warehouse inventory tracking system with functionalities such as:  
- Managing product, warehouse, and supplier data  
- Tracking stock levels in warehouses  
- Alert generation for low stock  
- Stock transfer between warehouses  
- Visual representation of stock distribution

# Schema Design

The schema consists of the following tables:  
1. Products: Stores product details.  
2. Suppliers: Contains supplier information.  
3. Warehouses: Stores warehouse location data.  
4. Stock: Tracks quantity of products in warehouses.

# Steps Performed

1. Designed schema with relationships among products, warehouses, suppliers, and stock.  
2. Inserted sample records using Python and Faker.  
3. Developed Python scripts for operations like inserting data, viewing alerts, and generating visualizations.  
4. Created stored procedures and triggers for stock transfer and alert mechanisms.  
5. Developed an interactive menu-driven program integrating all features.

# Key Functionalities

- Menu-driven interface for interacting with the warehouse system.  
- Automatic alerting for stock levels below threshold.  
- Visualization of warehouse-wise and product-wise stock.  
- Option to transfer stock between warehouses.  
- Check availability of products across all warehouses.

# 

# Conclusion

The project provides a simplified but robust prototype of an inventory management system suitable for retail enterprises. It demonstrates how backend database systems can be integrated with Python for data visualization, automation, and interactive operations.