Inventory and Warehouse Management System - Project Report

Student Name: Pallerla Akash
Project Duration: 2 Weeks
Technology: MySQL Database Management (DBeaver)
Date: September 2025

Introduction

This project implements a **database-driven Inventory and Warehouse Management System** using MySQL. The system tracks products, suppliers, and warehouses, and monitors stock levels in real-time. It ensures accurate stock management with **low-stock alerts**, **stock transfers**, and **supplier management**, supporting better decision-making in supply chain operations.

Abstract

The project focuses on building a backend system to streamline warehouse operations. The solution includes schema design, sample data insertion, operational queries, a low-stock trigger, and a stored procedure for stock transfers. With these components, the system prevents stockouts, improves data accuracy, and provides managers with clear inventory insights. The key deliverables include **stock monitoring**, **reorder alerts**, **automated notifications**, and **secure stock transfers**.

Tools Used

- **MySQL 8.x:** Relational database management system
- **DBeaver:** SQL client for design, queries, and testing
- **SQL Features:** DDL, DML, triggers, stored procedures, joins, constraints
- **Other:** Database modeling, integrity constraints, notification logging

Steps Involved in Building the Project

1. **Database Schema Design:** Designed tables: Suppliers, Warehouses, Products, Stock, Notifications. Applied keys, constraints, and reorder thresholds.

2. **Data Insertion:** Added sample suppliers, warehouses, and products. Populated stock records across multiple warehouses.

3. **Queries Development:** Stock overview queries to track availability. Reorder alerts to highlight products below thresholds.

4. **Trigger Creation:** Implemented *trg_low_stock* to insert automatic notifications when stock drops below reorder level.

5. **Stored Procedure:** Built *transfer_stock* to safely transfer items between warehouses with availability validation.

6. **Testing & Validation:** Verified schema integrity with sample operations. Tested low-stock alerts and stock transfers. Ensured data consistency across transactions.

Conclusion

The **Inventory and Warehouse Management System** provides a reliable backend for warehouse operations. By combining schema design, queries, triggers, and stored procedures, it improves accuracy, automates alerts, and ensures efficient stock handling.

Future Enhancements:
- Web dashboard for real-time monitoring.
- Role-based access control.
- Automated supplier order generation.