INTRODUCTION

Project Report

This project implements a **Warehouse Management System** in C, designed to streamline inventory control, shipment tracking, and order processing. The system integrates **four** core data structures to manage operations efficiently within a unified console interface.

Project Objective

* Demonstrate practical **integration of four data structures** in one application.
* Create an intuitive tool for warehouse inventory and logistics management.
* Implement fundamental **CRUD** operations for each data structure.

Data Structures Used & Functionalities

|  |  |  |
| --- | --- | --- |
| **Data Structure** | **Usage** | **Functionalities Implemented** |
| Array | Store products | Add/view products, prevent duplicates |
| Linked List | Manage shipments | Add/view shipments |
| Queue | Process orders | Add/view shipments |
| Stack | Undo actions (via history) | Push/pop recent actions |

Main Features

* **Product Management** – Add products with ID check; view inventory in table format.
* **Shipment Tracking** – Log shipments (product ID, destination) via linked list.
* **Order Processing** – Place orders using a queue.
* **Undo Actions** – Revert last changes via stack.
* **Dashboard** – View all products, shipments, and orders in one place.

Business Impact

* Centralizes inventory/shipment/order data for real-time tracking.
* Reduces manual errors via structured operations.
* Improves **supply chain efficiency** through organized logistics.

Advantages

* Modular design with clear separation of functionalities.
* Lightweight, terminal-based interface suitable for low-resource environments.
* Educational showcase of data structure integration.

Future Improvements

* **Persistence**: File-based save/load
* **Order Fulfillment**: Auto inventory update on dispatch
* **Error Handling**: Input validation
* **UI**: Interactive menus with edit/delete

Conclusion

This project successfully demonstrates the implementation and integration of multiple data structures in a real-world inspired application. It serves as an example of how core programming concepts can be combined to create a useful and functional system.