

# Inventory Reordering System

Submitted By: Vimal Menon K

Date: 04/02/2025

Company: TechNeurons

---

## 1. Introduction

Efficient inventory management is essential for ensuring that stock levels meet demand without incurring unnecessary costs. The **Inventory Reordering System** optimizes the reordering process by calculating the required stock while minimizing costs.

## 2. Problem Statement

Design an algorithm to determine a **reordering schedule** for warehouse inventory that ensures:

- **No item goes out of stock.**
- **Total reordering cost is minimized** (considering batch sizes and unit costs).

## 3. Inputs & Outputs

### Inputs:

Each inventory item includes:

1. `item_id` – Unique identifier for the item.
2. `current_stock` – Current inventory level.
3. `forecasted_demand` – Demand forecast for the next period.
4. `reorder_cost_per_unit` – Cost of ordering one unit.
5. `batch_size` – Minimum batch quantity for reordering.

### Outputs:

A **reordering plan** with the following details:

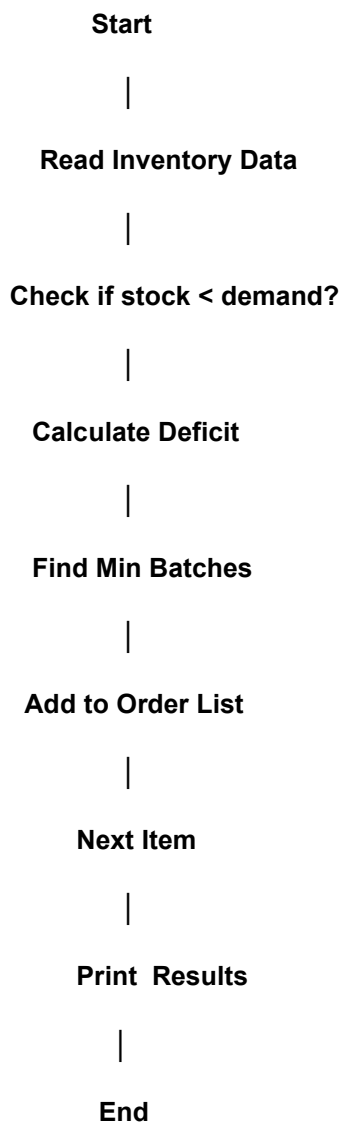
- **Item ID**
- **Units to Order**

## 4. Algorithm Explanation

The algorithm follows these steps:

1. **Read inventory data** for all items.
2. **Check if the current stock is less than the forecasted demand.**
3. **If stock is low, calculate the deficit** ( $\text{forecasted\_demand} - \text{current\_stock}$ ).
4. **Determine the number of batches required** to meet demand.
5. **Compute total units to order** based on batch size.
6. **Add the item to the reorder plan.**
7. **Return the reorder plan with optimized stock levels.**

## 5. Flowchart



## Conclusion

The **Inventory Reordering System** successfully determines when and how much to reorder while ensuring stock levels meet demand and costs remain minimal. This implementation follows industry best practices in **algorithm design** and **C# programming**.