Inventory Management System

# **Explain why data structures and algorithms are essential in handling large inventories.**

* **Efficient Data storing & retrieval:**
  + Choosing right data structure, e.g. Arrays, Linked Lists, Hash Tables, Trees, Graphs are important as they significantly impact the performance of searching, storing & retrieving inventory data.
  + Algorithms are crucial in searching, sorting and organizing items in inventory which helps in stream lining inventory operations.
* **Scalability & resource utilization:**
  + As inventory scales, more data processing and computation resources are required. DSA help to cut the resource costs and optimize operations.
* **Complex Queries, Operations & Analysis:**
  + Data structures and algorithms also play a critical role in analytics. They help businesses analyse inventory data for insights and decision-making.
  + Complex queries, like most sold items, history of a products, are often required. DSA also plays a major role here.

# **Discuss the types of data structures suitable for this problem.**

The data structure which is suitable for this problem is **HashMap.** The reason is HashMap allows to insert, search and delete items at **Constant Time Complexity - O(1)** in **average cases.**

# **Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.**

* **Add Operation:**
  + **Best Case Complexity:** O(1)
  + **Average Case Complexity:** O(1)
  + **Worst Case Complexity:** O(n) [n = number of items in HashMap]
* **Update Operation:**
  + Same as the add operation as they use same method.
* **Delete Operation:**
  + Same as the above-mentioned operations.

# **Discuss how you can optimize these operations.**

* **Hashing Collisions Handling:** HashMap has O(1) average complexity. But collisions can lead to O(n) in the worst case. Properly implemented hash functions and efficient capacity planning can minimize collisions and maintain performance**.**
* **Concurrency Handling**: In a multithreaded environment, it is better to use ‘**ConcurrentHashMap’.**