



DS&A SEMESTER PROJECT

Last date of Submission: 28 MAY 2025

Submitted to: MR SHAKEEL AHMED

GROUP MEMBERS

NAME	REGISTRATION NO
MOHAMMAD SAAD SHAFIQUE	4779-FOC/BSSE/F23
MUBASHIR NAZIR	4792-FOC/BSSE/F23
MUHAMMAD TAHIR	4795-FOC/BSSE/F23
MUZAMMIL KHALIQ	4824-FOC/BSSE/F23



PROJECT REPORT

Project Introduction:

- **PROJECT TITLE:** INVENTORY & STOCK CONTROL SYSTEM
- **PROGRAMMING LANGUAGE:** C++
- **PROGRAMMING TOOLS USED:** VSCODE and GITHUB
- **APPLICATION DOMAIN:** Businesses involving any kind of inventory & stock
- **DS&A CONCEPT USED:** Linked Lists, Binary Search Trees
- **PROJECT FEATURES:** Product Addition, Deletion, Display, Sorting and Supplier Addition, Deletion, Display and Stock Addition, Display, Sorting and also save and load all data.

Project Objective:

Our project, *Inventory & Stock Control System*, is designed to streamline the management of inventory and stock for businesses of any scale. It offers essential features such as product addition, deletion, updating, and display to ensure efficient product handling. In addition to inventory tracking, the system also incorporates stock control functionalities to monitor and manage product quantities effectively.

From a technical perspective, the system leverages key Data Structures & Algorithms (DS&A) concepts, including Linked Lists and Binary Search Trees, along with search algorithms like Linear Search and Binary Search, to ensure optimized data storage and retrieval.

Data Structures and Algorithms Used:

1. Binary Search Tree (BST) – for Products

We used a Binary Search Tree to store and manage product data. This allows efficient insertion, searching, and deletion of products based on their unique product ID. In-order traversal is used to display products in sorted order.



2. Singly Linked List – for Suppliers and Stock

We implemented singly linked lists to store supplier and stock records. This structure allows flexible insertion and traversal without the overhead of resizing, making it ideal for dynamic data like suppliers and stock entries.

3. Bubble Sort – for Products and Suppliers

Bubble Sort was applied to sort product and supplier arrays by their respective IDs in ascending order. It is simple to implement and useful for small datasets.

4. Merge Sort – for Stock (by Quantity)

Merge Sort was used to sort stock records based on quantity in descending order. This algorithm is efficient for large datasets due to its divide-and-conquer approach and stable sorting.

5. Linear and Binary Search

- **Linear Search** was used to find products and suppliers in unsorted arrays.
- **Binary Search** was applied on sorted arrays to quickly find entries using their IDs.

Code Screenshots:

Project Main Interface:

```
F:\4th Semester Documents\DSA\DSA CODES\project.exe

--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
```



Add Product:

```
F:\4th Semester Documents\DSA\DSA CODES\project.exe

--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 1
Enter Product ID: 1
Enter Product Name: Samsung A15
Enter Product Price: 40000
Enter Product Category: SmartPhone
Product added successfully.
```

```
0. Exit
Enter your choice: 1
Enter Product ID: 2
Enter Product Name: Digital Watch
Enter Product Price: 4000
Enter Product Category: SmartWatches
Product added successfully.
```

```
Enter your choice: 1
Enter Product ID: 3
Enter Product Name: HP Laptop H450
Enter Product Price: 120000
Enter Product Category: Laptops
Product added successfully.
```

Display Product:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 2
--- Products List ---
Product ID: 1, Name: Samsung A15, Price: $40000, Category: SmartPhone
Product ID: 2, Name: Digital Watch, Price: $4000, Category: SmartWatches
Product ID: 3, Name: HP Laptop H450, Price: $120000, Category: Laptops
-----
```



Search Product by ID:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 3
Enter Product ID to search: 2
Product ID: 2, Name: Digital Watch, Price: $4000, Category: SmartWatches
```

Remove Product:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 4
Enter Product ID to remove: 2
Product removed successfully.
```

Add Supplier:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 5
Enter Supplier ID: 10
Enter Supplier Name: Mubashir
Enter Supplier Contact Info: 03003030300
Supplier added successfully.
```



```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 5
Enter Supplier ID: 20
Enter Supplier Name: Muzammil
Enter Supplier Contact Info: 03030303030
Supplier added successfully.
```

Display All Supplier:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 6
--- Suppliers List ---
Supplier ID: 20, Name: Muzammil, Contact: 03030303030
Supplier ID: 10, Name: Mubashir, Contact: 03030303000
-----
```

Search Supplier by ID:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 7
Enter Supplier ID to search: 20
Supplier ID: 20, Name: Muzammil, Contact: 03030303030
```



Add Stock:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 8
Enter Product ID: 1
Enter Supplier ID: 10
Enter Quantity: 120
Stock added/updated successfully.
```

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 8
Enter Product ID: 4
Enter Supplier ID: 20
Enter Quantity: 200
Stock added/updated successfully.
```

Display All Stocks:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 9
--- Stock List ---
Product ID: 4, Supplier ID: 20, Quantity: 200
Product ID: 1, Supplier ID: 10, Quantity: 120
```




Sort Products by ID:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 10
--- Products Sorted by ID ---
Product ID: 1, Name: Samsung A15, Price: $40000, Category: SmartPhone
Product ID: 3, Name: HP Laptop H450, Price: $120000, Category: Laptops
Product ID: 4, Name: Earbuds Pro, Price: $3000, Category: Accessories
-----
```

Sort Stocks by Quantity:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 11
--- Stocks Sorted by Quantity ---
Product ID: 3, Supplier ID: 30, Quantity: 400
Product ID: 4, Supplier ID: 20, Quantity: 200
Product ID: 1, Supplier ID: 10, Quantity: 120
-----
```

Save all Data:

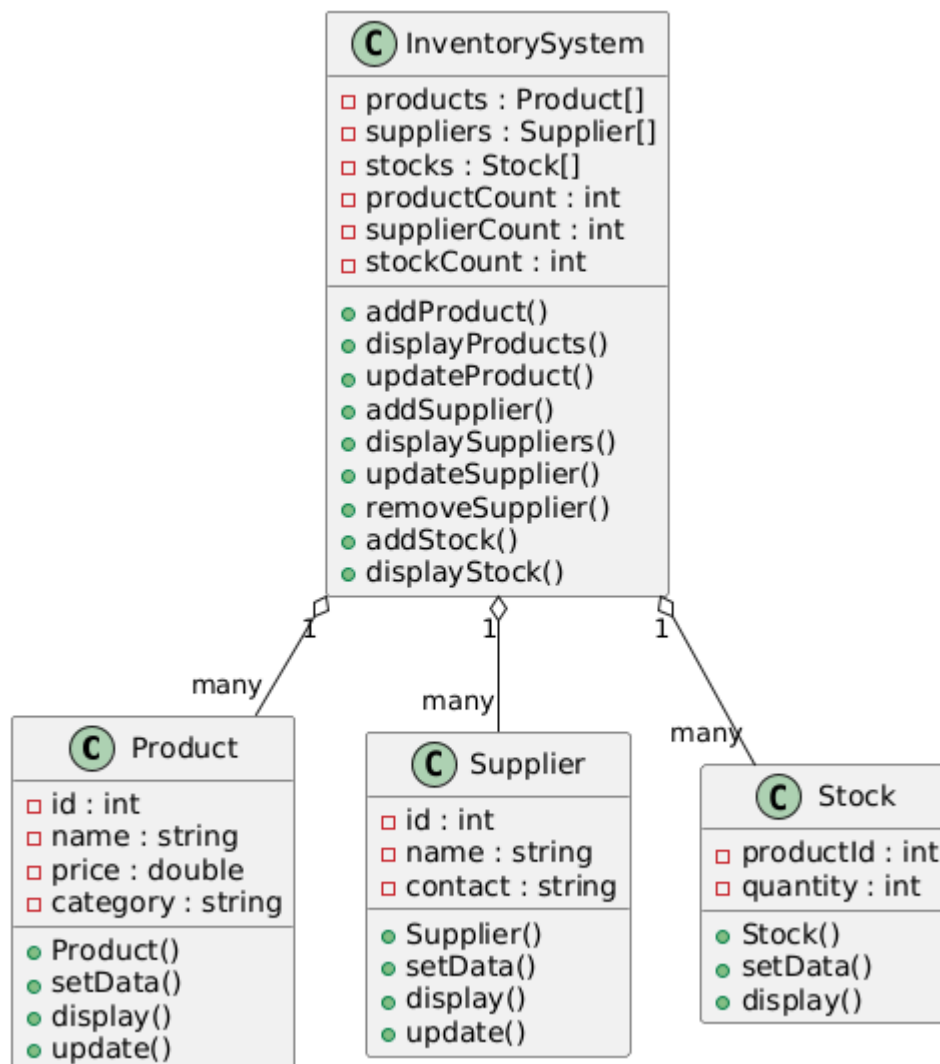
```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 12
Data saved successfully.
```




Load all Data:

```
--- Inventory Management System ---
1. Add Product
2. Display All Products
3. Search Product by ID
4. Remove Product
5. Add Supplier
6. Display All Suppliers
7. Search Supplier by ID
8. Add Stock
9. Display All Stocks
10. Sort Products
11. Sort Stocks
12. Save All Data
13. Load All Data
0. Exit
Enter your choice: 13
Data loaded successfully.
```

UML Class Diagram:





Performance Analysis and Optimization Summary:

1. Binary Search Tree (BST) – Products

- **Time Complexity:**
 - Insertion/Search/Deletion: Average **$O(\log n)$** , Worst-case **$O(n)$** (if unbalanced)
- **Optimization:**
 - In-order traversal is used for displaying sorted products without needing external sorting.
 - Unique product ID ensures data consistency and efficient lookup.

2. Singly Linked List – Suppliers and Stock

- **Time Complexity:**
 - Insertion at head: **$O(1)$**
 - Searching: **$O(n)$**
- **Optimization:**
 - Linked list is used where order is less critical, and fast insertions are required without resizing.

3. Sorting Algorithms

Algorithm	Applied On	Time Complexity	Remarks
Bubble Sort	Products, Suppliers	$O(n^2)$	Simple to implement, used on small datasets
Merge Sort	Stock (by quantity)	$O(n \log n)$	Efficient and stable for large data

4. Searching Algorithms

Method	Use Case	Time Complexity	Remarks
Linear Search	Unsorted arrays	$O(n)$	Simple, used where sorting is not guaranteed
Binary Search	Sorted arrays (by ID)	$O(\log n)$	Fast lookup on sorted data



5. File Handling Optimization

- Data is persisted using text files (products.txt, suppliers.txt, stocks.txt).
- On loading, duplicate ID exceptions are handled gracefully.
- Only necessary data is loaded and written, avoiding redundancy.

Member Wise Contribution:

- 1. Mohammad Saad Shafique → Coding + Professional Presentation**
- 2. Mubashir Nazir → Coding + Professional Presentation**
- 3. Muhammad Tahir → Coding + Design Documentation**
- 4. Muzammil Khaliq → Coding + Design Documentation**

GitHub Repository Link: [GitHub Repository](#)

LinkedIn Post Link: [LinkedIn Post](#)

Summary:

The "Inventory and Stock Control System" is a C++-based console application developed as part of our Data Structures and Algorithms course. It allows users to manage products, suppliers, and stock quantities through efficient data structures and algorithms. Key features include product insertion/search/deletion using a Binary Search Tree (BST), supplier and stock management using singly linked lists, and data persistence through file handling.

The system demonstrates practical implementation of object-oriented principles, custom exception handling, and algorithmic logic, while offering a user-friendly, menu-driven interface.

Conclusions:

- The project successfully showcases the application of core data structures such as BSTs and linked lists in solving real-world problems.
- Sorting algorithms like Bubble Sort and Merge Sort were integrated effectively for data organization and performance.



- Searching was optimized using both linear and binary search techniques.
- File handling ensured data persistence between sessions, and exception handling added robustness to the system.
- The project met all academic requirements, including design, functionality, and documentation.

Future Improvements:

To enhance the system further, we propose the following upgrades:

- **User Authentication:** Add login/logout functionality for multi-user access and security.
- **Graphical User Interface (GUI):** Convert the console-based application to a GUI using libraries like Qt or wxWidgets.
- **Database Integration:** Replace text files with a relational database (e.g., MySQL or SQLite) for better data management and scalability.
- **Self-Balancing Trees:** Use AVL or Red-Black Trees instead of BST to maintain balance and ensure optimal performance.
- **Reporting Module:** Generate reports on stock levels, supplier performance, and product movement.
- **Cloud Sync/Backup:** Store data in the cloud for access across systems and for automatic backup.
- **Mobile App:** Build a lightweight mobile version for real-time inventory access.