📦 DSA Toolkit (C++ Terminal App)

Developer: Mohd Meraaz
Timeline: May 2025
Technologies Used:

- C++
- File Handling
- Custom DSA Algorithms (No STL)

📌 Overview

The **DSA Toolkit** is a terminal-based C++ application designed to reinforce core Data Structures and Algorithms (DSA) without relying on the Standard Template Library (STL). It serves as a practical tool for students and professionals to interactively learn, implement, and test key data structures like arrays, stacks, queues, and linked lists.

The toolkit includes a multi-functional menu-driven system that allows users to perform a wide range of operations, including sorting, traversal, and a unique undo functionality using a stack.

🎯 Objectives

- Strengthen core DSA understanding without STL dependencies.
- Provide a practical tool to explore array, stack, queue, and linked list operations.
- Implement undo functionality using stack as a real-world application.
- Reinforce modular programming practices using functions and switch-case control logic.

🧠 Key Features

- 📌 **Array Operations**: Insertion, Deletion, Sorting, Traversal
- 📌 **Stack**: Push, Pop, Peek, Undo Feature
- 📌 **Queue**: Enqueue, Dequeue, Display
- 📌 **Linked List**: Insert at beginning/end/position, Delete, Reverse, Display
- 📌 **Undo Functionality**: Implemented using stack to reverse recent actions

- 📌 **File Handling**: Optional saving/loading of data (future-ready)

🧪 Challenges & Solutions

🔄 1. Designing a Versatile Switch-Case System

Challenge: Handling multiple operations cleanly
Solution: Structured the code with separate functions for each data structure and operation to ensure modularity and smooth user experience.

🚫 2. No STL Usage

Challenge: Manually managing memory and data structures
Solution: Implemented arrays and pointers from scratch for every DSA operation to build stronger control over memory and logic flow.

↩️ 3. Undo Functionality

Challenge: Efficiently tracking and reversing actions
Solution: Used a custom stack to record each significant change and allow state restoration, showcasing the practical use of stacks.

🔗 Project Links

- 📂 GitHub Repository: DSA Toolkit

💡 Learning Outcomes

- In-depth understanding of memory and pointer manipulation in C++
- Real-world application of stack for undo features
- Importance of modular code and clean control structures
- File handling basics in terminal-based applications
- Confidence in implementing DSA without relying on libraries

📈 Future Improvements

- Add Graph and Tree data structures