Smart Queue System for Efficient T-Shirt Distribution

Author: Ishtiaq Ahmed

Instructor: M. Shayan Shah

PROBLEM STATEMENT

The *Code-a-Thon* event gives free T-shirts to the first 100 participants. A system is needed to manage the queue of students arriving, waiting, and leaving dynamically. The queue must support:

- Adding new students
- Serving students from the front
- Removing students who leave early
- Displaying and counting the queue

Because the number of students is not fixed and students can leave from any position, the system must use a **dynamic and efficient structure** instead of arrays.

PROPOSED SOLUTION

To efficiently manage the dynamic queue, a **Linked List** was used to represent the students. Each node in the list stores:

- Student Name
- Student ID
- Pointer to the next node

Advantages of Using a Linked List

- Fast insertion and deletion (no data shifting required)
- Dynamic size — grows and shrinks as needed
- Maintains *First Come, First Serve* order naturally

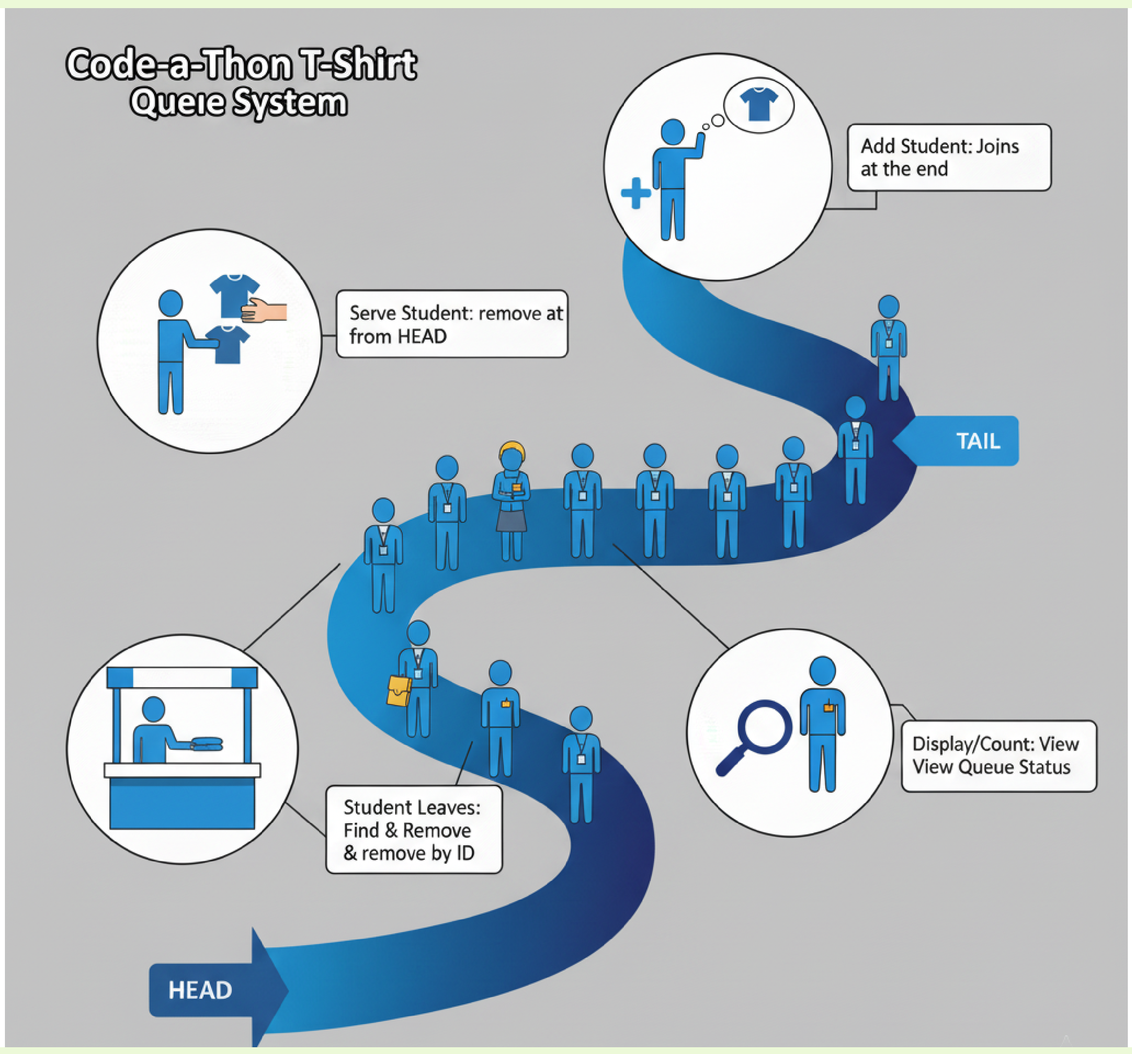
CONCLUSION

The *Queue Management System* demonstrates how a **linked list** can efficiently manage real-time, dynamic data. It enables quick additions, removals, and serving operations — making it ideal for real-world event queue management.
This project highlights how **data structures** serve as the foundation of practical and efficient software solutions.

WORKING METHODOLOGY

Functions Used

- **add_student()** – Adds a new student to the queue
- **serve_student()** – Serves (removes) the first student in the queue
- **student_leaves()** – Removes a specific student from the queue
- **display()** – Displays the current list of students
- **count_students()** – Shows the total number of students in the queue



ALGORITHM FLOW

Flow of the Program

1. Start the program
2. Display the menu options
3. User selects an operation
4. The corresponding linked list function executes
5. The updated queue is displayed
6. Repeat until the user chooses to exit