

# Task Management System

- Types of Linked Lists:

## Singly Linked List:

- Structure: Each node contains data and a reference (or link) to the next node in the sequence.
- Advantages: Simple structure, easy to implement. Efficient insertion and deletion operations at the beginning of the list.
- Disadvantages: Cannot traverse backwards. Requires additional memory for the link.

## Doubly Linked List:

- Structure: Each node contains data, a reference to the next node, and a reference to the previous node.
- Advantages: Can be traversed in both directions (forward and backward). Easier to delete a node given only a reference to that node.
- Disadvantages: More complex to implement and requires more memory due to the extra reference to the previous node.

- Time Complexity:

Add Operation:  $O(n)$  in the worst case, as you may need to traverse to the end of the list.

Search Operation:  $O(n)$  because you may need to check each node.

Traverse Operation:  $O(n)$  as every node must be visited.

Delete Operation:  $O(n)$  because you may need to find the node to delete by traversing the list.

- Advantages of Linked Lists over Arrays for Dynamic Data:

Dynamic Size: Linked lists can easily grow and shrink in size by adding or removing nodes. Arrays have a fixed size and require resizing, which is costly.

Efficient Insertions/Deletions: Insertions and deletions in linked lists are generally more efficient, especially when done at the beginning or middle of the list. In arrays, these operations require shifting elements, which is inefficient.

Memory Usage: Linked lists allocate memory as needed for each element, whereas arrays may allocate more memory than required, leading to potential waste.