**Task Management System**

# **Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**

* **Singly Linked List:** Each node in Singly Linked List contains data and a reference to the next node. The last node in the list points to null, indicating the end of the list.
* **Doubly Linked List:** Each node in Doubly Linked List contains data, a reference to the next node, and a reference to the previous node. This allows traversal in both directions.
* **Circular Linked List:** Similar to a singly linked list, but the last node points back to the head node instead of null, forming a circle.

# **Analyse the time complexity of each operation.**

* **Add:**
  + As I am using a linked list without tail pointer, the add operation is taking O(n) complexity to add new task in TaskList.
  + With the use of tail pointer, it can be done using constant time complexity.
* **Search:**
  + **Best case:** O(1), if the target task is at first node.
  + **Average case:** O(n), if the target task is at middle of tasklist.
  + **Worst case:** O(n), if the targeted task is at last node.
* **Traverse:**
  + Time complexity is **O(n)**, as we have to iterate over every element in list.
* **Delete:**
  + Delete operation takes task name search it in list and delete it. For deletion it requires constant time but there is an underlying search cost. So, the time complexity here is same as searching.

# **Discuss the advantages of linked lists over arrays for dynamic data.**

* Arrays are static, we can not increase or decrease their size in run time. But, in case of linked list it can be done effortlessly.
* Addition & Deletion in 1st position takes constant time complexity in case of Linked List, but in case of array it is linear.