

COMSATS UNIVERSITY ISLAMABAD ATTOCK CAMPUS

BS(Software Engineering)

NAME:	Muhammad Haroon
REG. NO:	SP23-BSE-042
SUBJECT:	DS
ASSIGNMENT NO:	01
DATE:	09-24-2024
SUBMITTED TO:	Mr. Muhammad Kamran

Introduction

The objective of this assignment is to design and implement a task management system using linked lists in C++. This system allows the user to manage tasks, which include adding tasks with different priorities, viewing all tasks, removing the highest priority task, and removing a task by its ID. The system ensures that tasks with higher priority are handled first, providing a flexible yet structured way to manage tasks efficiently.

Code Explanation

Task Struct:

The 'Task' struct stores information for each task. It contains:

- `taskId`: The unique identifier for each task.
- `description`: A brief description of the task.
- `priority`: A number indicating the task's importance (higher numbers indicate higher priority).
 - `next`: A pointer to the next task in the linked list.

TaskList Class:

This class manages the operations related to the task list. The linked list is implemented using dynamic memory allocation.

addTask:

This function inserts a new task into the list. The task is inserted based on its priority, ensuring tasks with higher priority appear at the front of the list. If the list is empty or the new task has the highest priority, it becomes the new head. Otherwise, it finds the correct position for insertion.

removeHighestPriorityTask:

This function removes the task with the highest priority, which is always located at the head of the list. It adjusts the head to point to the next task and deallocates the memory used by the removed task.

removeTaskById:

This function removes a task by its ID. If the task to be removed is the head, it adjusts the head. Otherwise, it searches for the task in the list and adjusts the pointers to bypass the removed task, then frees the allocated memory.

viewAllTasks:

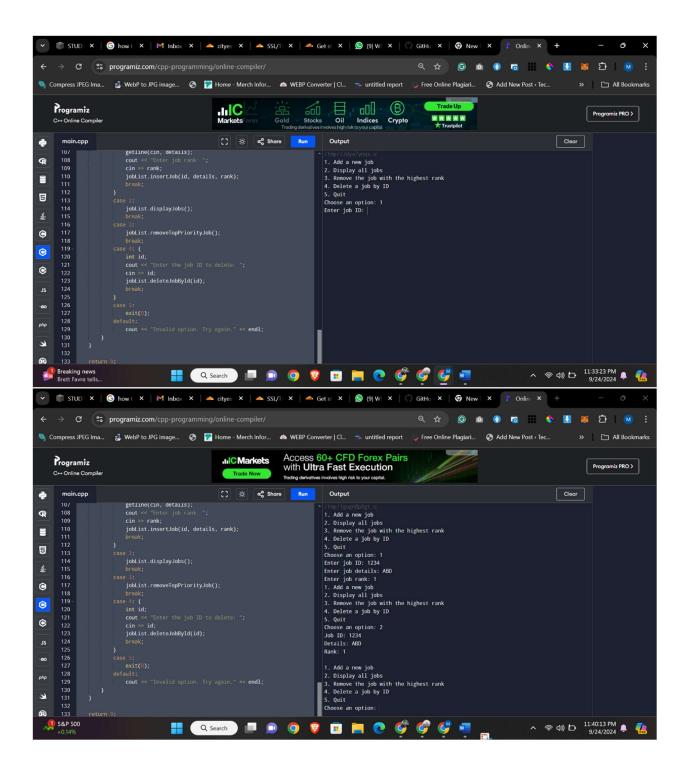
This function traverses the list and prints all tasks, displaying the task ID, description, and priority.

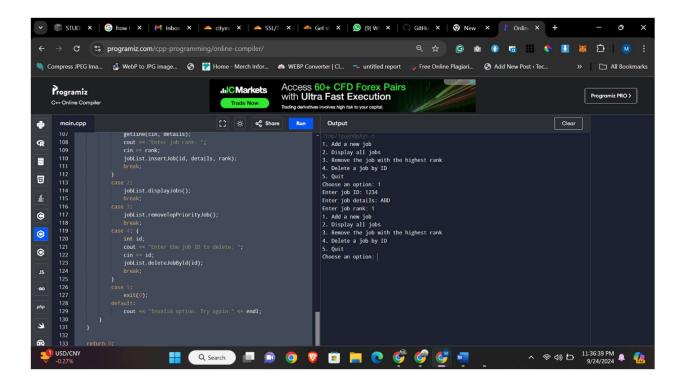
3. Main Function:

The main function presents a menu to the user to perform various operations:

- Option 1 allows the user to add a new task.
- Option 2 displays all tasks currently in the list.
- Option 3 removes the task with the highest priority.
- Option 4 removes a task based on the task ID.
- Option 5 exits the program.

Output Screenshots





Conclusion

Through this assignment, I learned how to manage dynamic memory in C++ and implement a linked list to store and organize data efficiently. The biggest challenge was handling edge cases, such as inserting at the head or removing the last task in the list. This experience has improved my understanding of data structures and how to apply them in practical scenarios.