Task Management System using Singly Linked List

Name: [Your Name]

Enrollment Number: [Your Enrollment Number]

Date: [Submission Date]

Introduction

The objective of this assignment is to create a task management system using a singly linked list in C++.

Each task is represented as a node, with fields for a unique task ID, a description, and a priority level.

The linked list will store tasks in order of priority, with the highest priority tasks at the front. The system

includes functionality for adding tasks, viewing all tasks, and removing tasks by their priority or task ID.

Code Explanation

1. Task Structure:

The 'Node' struct represents each task in the linked list, containing the task ID, description, priority, and a pointer to the next node.

2. Adding a Task:

The 'addTask' function adds a new task to the list. If the list is empty or the new task has a higher priority than the current head,

it becomes the new head. Otherwise, the task is inserted in the appropriate position based on its priority.

3. Removing the Highest Priority Task:

The 'removeHighestPriorityTask' function removes the task at the head of the list, which is the task with the highest priority.

4. Removing a Task by ID:

The 'removeTaskById' function traverses the list, finds the task with the specified ID, and removes it from the list.

5. Viewing Tasks:

The 'viewTasks' function iterates through the list and prints the details of each task.

Screenshots

[Include screenshots of the program output here]

Conclusion

Through this assignment, I learned how to implement a basic task management system using a singly linked list in C++.

I explored operations such as adding, viewing, and removing tasks based on their priority and ID. One challenge I faced

was ensuring tasks were added in the correct order based on their priority, which required careful pointer manipulation

in the linked list.