* **Name: Reilly Lambey**
* **Student ID: 20221566895**
* **Course: CMPS2131**
* **Date:14-Feb-2025**

Overview of the Project

Using a singly linked list, the Task Manager is a terminal application that assists users in organizing chores. Users can add, remove, and manage tasks according to their due dates and priorities. Features like searching, filtering, and task display that are easy to use are all provided by the system.

Task Class/Struct Properties of Data Structures:

Task Description (string): Holds information about the task.

Task priority is tracked by the Priority Level (string) (High, Medium, Low).

Due Date (string): Holds the due date for the work.

Completion Status (boolean): Indicates if the task has been finished.

Task Next Pointer: \* Indicates the task that comes next this one in the list.

Justification: To manage the linked list, prioritize activities, save task data, and keep track of deadlines, these features are essential.

Linked List Management TaskList Class:

The first duty is indicated by the head pointer.

The tail pointer indicates the final task.

Tasks are added according to priority (High > Medium > Low) using the insertion logic.

Task Priority Ordering: Tasks are arranged according to priority. Insertion time (FIFO) is used to prioritize tasks of the same priority.

Essential Features

Managing Tasks

Include a Task: work information are entered by users, and the work is added according to priority.

Eliminate a Task: Tasks that have been finished are taken from the list.

Mark a Task as Complete: Users have the ability to modify a task's completion status.

Task Search: Users have the option to look for tasks by description.

Show All Tasks: This option shows every task in order of priority.

Tasks of a particular priority level are displayed when you filter them by priority.

Show Tasks Due Within N Days: Shows tasks due within a specified number of days.

Handling Errors and Edge Cases

Empty List Operations: If the list is empty, show "No tasks found".

Invalid Inputs: Verify inputs (such as those with an incorrect date format or priority).

Failures with Memory Allocation: After memory allocation, look for nullptr.

Plan for Memory Management

To deallocate memory when tasks are eliminated, use delete; to allocate memory for tasks, use new.

To avoid leaks, make sure all memory is released prior to program closure.

User Interface Plan Menu-Driven Interface: A numbered menu is used to select options (such as add, remove, and search).

Input Prompts: Users interactively enter task information.

Clear Output: The results are presented in a structured way.

Testing Strategy

Add a Task: Confirm that tasks are added in the appropriate order of importance.

Eliminate a Task: Make sure memory is deallocated and tasks are eliminated.

Search for a Task: Confirm tasks are found by description.

Sort Tasks by Priority: Confirm that the filtering is effective.

Display Assignments Due in N Days: Verify that jobs that are due within the allotted time are shown.

Extra Features (Optional)

Sort jobs by Due Date: In addition to priority, sort jobs by due date.

Tasks can be saved to a file and loaded when the application launches.

Allow users to reverse the most recent action (such as adding or removing).