**MODULE-2**

**DATA STRUCTURE AND ALGORITHM**

SUPERSET ID:6407550

**Exercise 5: Task Management System**

**Task.java**

package linkedlist;

public class Task {

int taskId;

String taskName;

String status;

public Task(int taskId, String taskName, String status) {

this.taskId = taskId;

this.taskName = taskName;

this.status = status;

}

}

**TaskNode.java**

package linkedlist;

public class TaskNode {

Task task;

TaskNode next;

public TaskNode(Task task) {

this.task = task;

this.next = null;

}

}

**TaskLinkedList.java**

package linkedlist;

public class TaskLinkedList {

private TaskNode head;

// Add task to end of list

public void addTask(Task task) {

TaskNode newNode = new TaskNode(task);

if (head == null) {

head = newNode;

return;

}

TaskNode temp = head;

while (temp.next != null) {

temp = temp.next;

}

temp.next = newNode;

}

// Search task by ID

public Task searchTask(int taskId) {

TaskNode current = head;

while (current != null) {

if (current.task.taskId == taskId) {

return current.task;

}

current = current.next;

}

return null;

}

// Delete task by ID

public boolean deleteTask(int taskId) {

if (head == null) return false;

if (head.task.taskId == taskId) {

head = head.next;

return true;

}

TaskNode prev = head;

TaskNode curr = head.next;

while (curr != null) {

if (curr.task.taskId == taskId) {

prev.next = curr.next;

return true;

}

prev = curr;

curr = curr.next;

}

return false;

}

// Traverse and display all tasks

public void displayTasks() {

TaskNode current = head;

while (current != null) {

System.out.println("ID: " + current.task.taskId +

", Name: " + current.task.taskName +

", Status: " + current.task.status);

current = current.next;

}

}

}

**TaskManagerApp.java**

package linkedlist;

public class TaskManagerApp {

public static void main(String[] args) {

TaskLinkedList taskList = new TaskLinkedList();

taskList.addTask(new Task(1, "Design UI", "Pending"));

taskList.addTask(new Task(2, "Develop Backend", "In Progress"));

taskList.addTask(new Task(3, "Write Test Cases", "Pending"));

System.out.println("All Tasks:");

taskList.displayTasks();

System.out.println("\nSearching for Task with ID 2:");

Task found = taskList.searchTask(2);

if (found != null)

System.out.println("Found: " + found.taskName + " - " + found.status);

else

System.out.println("Task not found.");

System.out.println("\nDeleting Task with ID 1:");

boolean deleted = taskList.deleteTask(1);

System.out.println("Deleted: " + deleted);

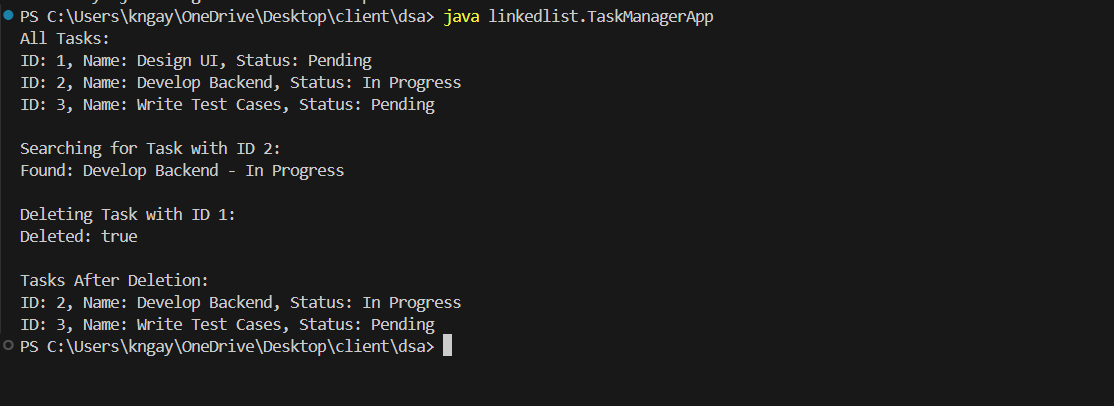
System.out.println("\nTasks After Deletion:");

taskList.displayTasks();

}

}

**OUTPUT:**

****