**Exercise 5: Task Management System**

**Task.java**

class Task {

    int taskId;

    String taskName;

    String status;

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

        this.taskId = taskId;

        this.taskName = taskName;

        this.status = status;

    }

    void display() {

        System.out.println(taskId + " | " + taskName + " | " + status);

    }

}

**Node.java**

class Node {

    Task task;

    Node next;

    Node(Task task) {

        this.task = task;

        this.next = null;

    }

}

**TaskManagementSystem.java**

public class TaskManagementSystem {

    Node head = null;

    void addTask(int id, String name, String status) {

        Task newTask = new Task(id, name, status);

        Node newNode = new Node(newTask);

        if (head == null) {

            head = newNode;

        } else {

            Node temp = head;

            while (temp.next != null)

                temp = temp.next;

            temp.next = newNode;

        }

        System.out.println("Task added.");

    }

    void searchTask(int id) {

        Node temp = head;

        while (temp != null) {

            if (temp.task.taskId == id) {

                System.out.println("Task found:");

                temp.task.display();

                return;

            }

            temp = temp.next;

        }

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

    }

    void viewTasks() {

        if (head == null) {

            System.out.println("No tasks to display.");

            return;

        }

        Node temp = head;

        System.out.println("Task List:");

        while (temp != null) {

            temp.task.display();

            temp = temp.next;

        }

    }

    void deleteTask(int id) {

        if (head == null) {

            System.out.println("Task list is empty.");

            return;

        }

        if (head.task.taskId == id) {

            head = head.next;

            System.out.println("Task deleted.");

            return;

        }

        Node temp = head;

        while (temp.next != null && temp.next.task.taskId != id)

            temp = temp.next;

        if (temp.next == null) {

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

        } else {

            temp.next = temp.next.next;

            System.out.println("Task deleted.");

        }

    }

    public static void main(String[] args) {

        TaskManagementSystem tms = new TaskManagementSystem();

        Scanner sc = new Scanner(System.in);

        int choice;

        do {

            System.out.println("\n--- Task Management Menu ---");

            System.out.println("1. Add Task");

            System.out.println("2. View Tasks");

            System.out.println("3. Search Task");

            System.out.println("4. Delete Task");

            System.out.println("5. Exit");

            System.out.print("Enter your choice: ");

            choice = sc.nextInt();

            switch (choice) {

                case 1 -> {

                    System.out.print("Enter Task ID: ");

                    int id = sc.nextInt();

                    sc.nextLine();

                    System.out.print("Enter Task Name: ");

                    String name = sc.nextLine();

                    System.out.print("Enter Status: ");

                    String status = sc.nextLine();

                    tms.addTask(id, name, status);

                }

                case 2 -> tms.viewTasks();

                case 3 -> {

                    System.out.print("Enter Task ID to search: ");

                    int id = sc.nextInt();

                    tms.searchTask(id);

                }

                case 4 -> {

                    System.out.print("Enter Task ID to delete: ");

                    int id = sc.nextInt();

                    tms.deleteTask(id);

                }

                case 5 -> System.out.println("Exiting...");

                default -> System.out.println("Invalid choice!");

            }

        } while (choice != 5);

        sc.close();

    }

}

**Output**





