**Task Management System**

**Main.java**

public class Main {

    public static void main(String[] args) {

        TaskManager manager = new TaskManager();

        manager.addTask(1, "Design UI", "Pending");

        manager.addTask(2, "Implement Login", "In Progress");

        manager.addTask(3, "Database Setup", "Completed");

        System.out.println("\nAll Tasks:");

        manager.listTasks();

        System.out.println("\nSearch Task ID 2:");

        Task found = manager.searchTask(2);

        System.out.println(found != null ? found : "Task not found");

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

        manager.deleteTask(1);

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

        manager.listTasks();

    }

}

**Task.java**

public class Task {

    int taskId;

    String taskName;

    String status;

    Task next; // Pointer to next node

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

        this.taskId = taskId;

        this.taskName = taskName;

        this.status = status;

        this.next = null;

    }

    @Override

    public String toString() {

        return "Task{" +

                "ID=" + taskId +

                ", Name='" + taskName + '\'' +

                ", Status='" + status + '\'' +

                '}';

    }

}

**TaskManager.java**

public class TaskManager {

    private Task head;

    // Add task to the end

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

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

        if (head == null) {

            head = newTask;

        } else {

            Task current = head;

            while (current.next != null) {

                current = current.next;

            }

            current.next = newTask;

        }

        System.out.println("Task added: " + id);

    }

    // Search task by ID

    public Task searchTask(int id) {

        Task current = head;

        while (current != null) {

            if (current.taskId == id)

                return current;

            current = current.next;

        }

        return null;

    }

    // Traverse and print all tasks

    public void listTasks() {

        Task current = head;

        if (current == null) {

            System.out.println("No tasks available.");

            return;

        }

        while (current != null) {

            System.out.println(current);

            current = current.next;

        }

    }

    // Delete task by ID

    public boolean deleteTask(int id) {

        if (head == null)

            return false;

        if (head.taskId == id) {

            head = head.next;

            System.out.println("Task deleted: " + id);

            return true;

        }

        Task current = head;

        while (current.next != null && current.next.taskId != id) {

            current = current.next;

        }

        if (current.next == null)

            return false;

        current.next = current.next.next;

        System.out.println("Task deleted: " + id);

        return true;

    }

}

