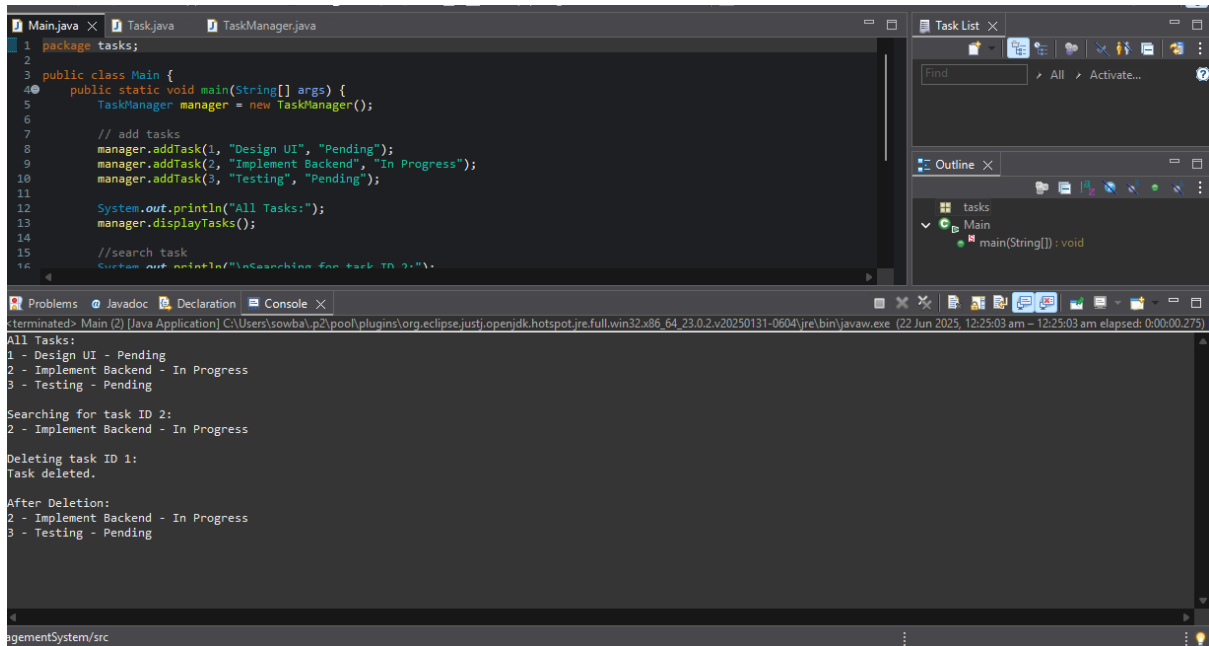


Exercise 5: Task Management System

OUTPUT:



```
1 package tasks;
2
3 public class Main {
4     public static void main(String[] args) {
5         TaskManager manager = new TaskManager();
6
7         // add tasks
8         manager.addTask(1, "Design UI", "Pending");
9         manager.addTask(2, "Implement Backend", "In Progress");
10        manager.addTask(3, "Testing", "Pending");
11
12        System.out.println("All Tasks:");
13        manager.displayTasks();
14
15        //search task
16        Scanner scanner = new Scanner(System.in);
17        System.out.println("Enter task ID to search:");
18        int taskID = scanner.nextInt();
19        manager.searchTask(taskID);
20
21        //delete task
22        System.out.println("Enter task ID to delete:");
23        int deleteID = scanner.nextInt();
24        manager.deleteTask(deleteID);
25
26        System.out.println("After Deletion:");
27        manager.displayTasks();
28    }
29 }
```

terminated> Main (2) [Java Application] C:\Users\sowba\p2\prof\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_23.0.2.v20250131-0604\jre\bin\javaw.exe (22 Jun 2025, 12:25:03 am - 12:25:03 am elapsed: 0:00:00.275)

All Tasks:
1 - Design UI - Pending
2 - Implement Backend - In Progress
3 - Testing - Pending

Searching for task ID 2:
2 - Implement Backend - In Progress

Deleting task ID 1:
Task deleted.

After Deletion:
2 - Implement Backend - In Progress
3 - Testing - Pending

The Task Management System uses a singly linked list for dynamic task handling. Linked lists allow efficient addition and deletion, especially when tasks change frequently. Unlike arrays, they offer flexibility in memory usage and avoid shifting elements during operations.