

## PROJECT::Student Record System (LinkedList / ArrayList)

### \*\*1. Problem Statement\*\*

A college wants to maintain student records that include ID, Name, and Marks. The system should allow administrators to add new students, display all students, search for a student by ID, delete a student by ID, and update student details by ID. This avoids the need for manual registers and makes student management efficient.

---

### \*\*2. Sample Input\*\*

...

```
1 → Add Student
101 Alice 88.5
1 → Add Student
102 Bob 76
2 → Display Students
3 → Search Student by ID (101)
5 → Update Student by ID (102 → Bobby, 80)
4 → Delete Student by ID (101)
2 → Display Students
6 → Exit
```

...

---

### \*\*3. Sample Output\*\*

...

```
Student added successfully.
Student added successfully.
101 - Alice - 88.5
102 - Bob - 76.0
Record Found: 101 - Alice - 88.5
Record updated successfully.
Record deleted successfully.
102 - Bobby - 80.0
Exiting...
```

...

---

### \*\*4. Code\*\*

```

``java
import java.util.*;

class Student {
    int id;
    String name;
    double marks;

    Student(int id, String name, double marks) {
        this.id = id;
        this.name = name;
        this.marks = marks;
    }

    @Override
    public String toString() {
        return id + " - " + name + " - " + marks;
    }
}

public class StudentRecordSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        List<Student> students = new ArrayList<>();

        while (true) {
            System.out.println("\n===== Student Record System =====");
            System.out.println("1. Add Student");
            System.out.println("2. Display Students");
            System.out.println("3. Search Student by ID");
            System.out.println("4. Delete Student by ID");
            System.out.println("5. Update Student by ID");
            System.out.println("6. Exit");
            System.out.print("Enter choice: ");

            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1: // Add
                    System.out.print("Enter ID: ");
                    int id = sc.nextInt();
                    sc.nextLine();
                    System.out.print("Enter Name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter Marks: ");
                    double marks = sc.nextDouble();

```

```
students.add(new Student(id, name, marks));  
System.out.println("Student added successfully.");  
break;
```

case 2: // Display

```
if (students.isEmpty()) {  
    System.out.println("No records found.");  
} else {  
    for (Student s : students) {  
        System.out.println(s);  
    }  
}  
break;
```

case 3: // Search

```
System.out.print("Enter ID to search: ");  
int searchId = sc.nextInt();  
boolean found = false;  
for (Student s : students) {  
    if (s.id == searchId) {  
        System.out.println("Record Found: " + s);  
        found = true;  
        break;  
    }  
}  
if (!found) {  
    System.out.println("Record not found!");  
}  
break;
```

case 4: // Delete

```
System.out.print("Enter ID to delete: ");  
int delId = sc.nextInt();  
Iterator<Student> it = students.iterator();  
boolean deleted = false;  
while (it.hasNext()) {  
    if (it.next().id == delId) {  
        it.remove();  
        deleted = true;  
        System.out.println("Record deleted successfully.");  
        break;  
    }  
}  
if (!deleted) {  
    System.out.println("Record not found!");  
}  
break;
```

```

case 5: // Update
    System.out.print("Enter ID to update: ");
    int updlId = sc.nextInt();
    sc.nextLine();
    boolean updated = false;
    for (Student s : students) {
        if (s.id == updlId) {
            System.out.print("Enter new name: ");
            s.name = sc.nextLine();
            System.out.print("Enter new marks: ");
            s.marks = sc.nextDouble();
            updated = true;
            System.out.println("Record updated successfully.");
            break;
        }
    }
    if (!updated) {
        System.out.println("Record not found!");
    }
    break;

case 6: // Exit
    System.out.println("Exiting...");
    sc.close();
    return;

default:
    System.out.println("Invalid choice, try again.");
    }
    }
}
...

```

---

## **\*\*5. Actual Output (Sample Run)\*\***

```

...
===== Student Record System =====
1. Add Student
2. Display Students
3. Search Student by ID
4. Delete Student by ID
5. Update Student by ID
6. Exit
Enter choice: 1
Enter ID: 101

```

Enter Name: Alice  
Enter Marks: 88.5  
Student added successfully.

===== Student Record System =====

Enter choice: 1  
Enter ID: 102  
Enter Name: Bob  
Enter Marks: 76  
Student added successfully.

===== Student Record System =====

Enter choice: 2  
101 - Alice - 88.5  
102 - Bob - 76.0

===== Student Record System =====

Enter choice: 3  
Enter ID to search: 101  
Record Found: 101 - Alice - 88.5

===== Student Record System =====

Enter choice: 5  
Enter ID to update: 102  
Enter new name: Bobby  
Enter new marks: 80  
Record updated successfully.

===== Student Record System =====

Enter choice: 4  
Enter ID to delete: 101  
Record deleted successfully.

===== Student Record System =====

Enter choice: 2  
102 - Bobby - 80.0

===== Student Record System =====

Enter choice: 6  
Exiting...

```