Easy Level: Employee Management System Problem Statement 📝 Write a Java program to implement an ArrayList that stores employee details (ID, Name, and Salary). Allow users to:

Add employees

Update employee details

Remove employees

Search for employees

Code:

```
import java.util.*;

class Employee {
    private int id;
    private String name;
    private double salary;

    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    public double getSalary() {
        return salary;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setSalary(double salary) {
        this.salary = salary;
    }
    @Override
    public String toString() {
        return "Employee ID: " + id + ", Name: " + name + ", Salary: " +
salary;
    }
}
public class EmployeeManagement {
    private static final List<Employee> employees = new ArrayList<>();
    private static final Scanner scanner = new Scanner(System.in);
    public static void main(String[] args) {
        while (true) {
            System.out.println("\nEmployee Management System");
            System.out.println("1. Add Employee");
            System.out.println("2. Update Employee");
            System.out.println("3. Remove Employee");
            System.out.println("4. Search Employee");
```

```
            System.out.println("5. Display All Employees");
            System.out.println("6. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1 -> addEmployee();
                case 2 -> updateEmployee();
                case 3 -> removeEmployee();
                case 4 -> searchEmployee();
                case 5 -> displayEmployees();
                case 6 -> {
                    System.out.println("Exiting...");
                    return;
                }
                default -> System.out.println("Invalid choice! Please try
again.");
            }
        }
    }
    private static void addEmployee() {
        System.out.print("Enter Employee ID: ");
        int id = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        System.out.print("Enter Employee Name: ");
        String name = scanner.nextLine();
        System.out.print("Enter Employee Salary: ");
        double salary = scanner.nextDouble();

        employees.add(new Employee(id, name, salary));
        System.out.println("Employee added successfully!");
    }
    private static void updateEmployee() {
        System.out.print("Enter Employee ID to update: ");
        int id = scanner.nextInt();
        for (Employee emp : employees) {
            if (emp.getId() == id) {
                scanner.nextLine(); // Consume newline
                System.out.print("Enter new Name: ");
                String newName = scanner.nextLine();
                System.out.print("Enter new Salary: ");
                double newSalary = scanner.nextDouble();
                emp.setName(newName);
                emp.setSalary(newSalary);
                System.out.println("Employee updated successfully!");
                return;
            }
        }
        System.out.println("Employee not found!");
    }
    private static void removeEmployee() {
        System.out.print("Enter Employee ID to remove: ");
        int id = scanner.nextInt();
        employees.removeIf(emp -> emp.getId() == id);
        System.out.println("Employee removed successfully!");
    }
    private static void searchEmployee() {
        System.out.print("Enter Employee ID to search: ");
        int id = scanner.nextInt();
        for (Employee emp : employees) {
            if (emp.getId() == id) {
                System.out.println(emp);
                return;
```

```
            }
        }
        System.out.println("Employee not found!");
    }
    private static void displayEmployees() {
        if (employees.isEmpty()) {
            System.out.println("No employees found.");
        } else {
            for (Employee emp : employees) {
                System.out.println(emp);
            }
        }
    }
}
```

Output:

```
Run    EmployeeManagement ×

"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\P

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Choose an option: 1
Enter Employee ID: 12477
Enter Employee Name: Anshuman
Enter Employee Salary: 99999
Employee added successfully!

Employee Management System
1. Add Employee
2. Update Employee
3. Remove Employee
4. Search Employee
5. Display All Employees
6. Exit
Choose an option: 5
Employee ID: 12477, Name: Anshuman, Salary: 99999.0
```