Employee Management System – Python Project (Functions Only, No File Handling)

---

1. Problem Statement

A small organization with fewer than 8 employees wants to manage their employee data (name, department, role, salary) using a simple Python program.
The system should allow adding, viewing, searching, updating, and deleting employee details using only **functions** (no OOP, no database, no file storage).

---

2. Features & Functionalities

1. **Add Employee**
    - Input: ID, Name, Department, Role, Salary
    - Restrict maximum employees to **8 only**
2. **View Employees**
    - Display all employees in a tabular format
3. **Search Employee**
    - Find employee by ID or Name
4. **Update Employee**
    - Modify existing employee details
5. **Delete Employee**
    - Remove employee record
6. **Exit Program**
    - End the application safely

---

3. Requirements

- Python 3.8+
- Knowledge of:
    - Lists & dictionaries
    - Functions
    - Loops & conditionals

---

4. Data Structure

Employees will be stored in a **list of dictionaries** (maximum 8 records).

Example:

```
employees = [
   {"id": 1, "name": "Alice", "department": "HR", "role": "Manager", "salary": 50000},
   {"id": 2, "name": "Bob", "department": "IT", "role": "Developer", "salary": 60000}
]
```

---

5. Functions to Implement

- add_employee(employees) – Add a new employee (check limit = 8).
- view_employees(employees) – Show all employees.
- search_employee(employees) – Find employee by ID or name.
- update_employee(employees) – Edit employee details.
- delete_employee(employees) – Delete employee by ID.
- menu() – Display menu and call functions.

---

6. Menu Example

===== Employee Management System =====

1. Add Employee

2. View Employees

3. Search Employee

4. Update Employee

5. Delete Employee

6. Exit

Enter your choice:

---

7. Step-by-Step Implementation Guide

Step 1 – Setup

- Create a Python file (main.py).
- Create an empty list: employees = [].
- Define all required functions with just pass for now.
- Create a loop to display the menu.

Step 2 – Add Employee Function

- Input details: ID, Name, Department, Role, Salary.
- Convert salary into an integer/float.
- Check if employees list has less than 8 records.
- Append dictionary to the list.
- Print success message.

Step 3 – View Employees Function

- Check if the list is empty → print "No employees found".
- Else, loop through the list and print each employee in a neat format.

Step 4 – Search Employee Function

- Ask user for ID or Name.
- Loop through employees.
- If found → display details.
- If not found → print "Employee not found".

Step 5 – Update Employee Function

- Ask for ID.
- If ID exists → ask which field to update (Department, Role, or Salary).
- Update the value.
- Print success message.

Step 6 – Delete Employee Function

- Ask for ID.
- If ID exists → remove from list.
- Print success message.

Step 7 – Exit Option

- Break the loop when user enters 6.

Step 8 – Testing

- Add 2–3 employees.
- Try searching, updating, deleting.
- Check limit (only 8 employees allowed).
- Verify outputs.

---

8. Evaluation Criteria

- Correct implementation of CRUD operations – 50%
- Proper use of functions (no OOP) – 20%
- Data validation (check limit, no duplicate ID) – 15%
- Code readability & comments – 15%