Employee Management System (EMS)

Objective - Create a simplified **Employee Management System (EMS)**. This project will cover control structures, functions, and object-oriented programming concepts to manage employee data.

Steps to Implement

Step 1 - Plan the Data Storage

- Use a **dictionary** to store employee data where the **keys** is the `emp_id` (Employee ID) and the **value** is another dictionary containing:
  - `name`: Employee's name.
  - `age`: Employee's age.
  - `department`: Employee's department.
  - `salary`: Employee's monthly salary.
- **Initialize** the dictionary with some sample employee data for testing (e.g., `{101: {'name': 'Satya', 'age': 27, 'department': 'HR', 'salary': 50000}}`).

Step 2 - Define the Menu System

- Create a **menu** that displays the following options:
  1. **Add Employee**
  2. **View All Employees**
  3. **Search for Employee**
  4. **Exit**
- Implement a **loop** to continuously display the menu until the user chooses to **Exit**.

Step 3 - Add Employee Functionality

1. **Prompt the User** to enter the following details for a new employee:
   - `emp_id` (Employee ID)
   - `name` (Employee Name)
   - `age` (Employee Age)
   - `department` (Employee Department)
   - `salary` (Employee Salary)

2. **Validate Input**: Make sure the Employee ID is unique. If it already exists in the dictionary, ask the user to enter a new ID.

3. **Store the Employee** data in the dictionary using the entered `emp_id` as the key and the other details as values.

4. Display a message indicating the employee was successfully added.

Step 4 - View All Employees

1. **Display all employees** stored in the dictionary.
2. Format the display in a **table-like structure**, showing employee details (ID, name, age, department, salary).
3. If there are no employees in the system, display a message like:
   - *"No employees available."*

Step 5 - Search for an Employee by ID

1. **Prompt the User** to enter the `emp_id` they want to search for.
2. **Search the Dictionary**:
   - If the employee exists, display their details (name, age, department, salary).
   - If the employee does not exist, display a message like:
     - *"Employee not found."*

Step 6 - Exit the Program

1. Add an **Exit option** in the menu.
2. If the user chooses **Exit**, display a **thank-you message** and **exit** the program.

Project Code Structure

To keep the project organized, break it into functions:

1. **`main_menu()`**: Displays the main menu and calls the appropriate function based on user input.
2. **`add_employee()`**: Adds a new employee to the dictionary.
3. **`view_employees()`**: Displays all employee details.
4. **`search_employee()`**: Searches for an employee by ID.

Note : Submit the project code as a Python script (.py).
