PYTHON PROJECT

Employee Management System

Abstract

A project that covers Python topics under the Embedded Linux Diploma by Eng. Mohamed Bakr

Habeba Ahmed

Habebaahmed101@gmail.com

Table Content

Problem Statement

It is software for the Employee Management System which can perform the following operations.

- Login using Employee ID and password stored in the System.
- Add Employee Record.
- Remove Employee record.
- Update Employee records.
- Checking duplicate IDs.
- Display user Information.
- Calculate Employee Bonus.
- Calculate Employee Discount.
- Legal Holiday Reminder.
- Exit Function.

Approach

The idea is to form individual functions for every operations, the data structures used is a list of dictionaries, each dictionary is an Employee record with all its information stored in a key-value manner.

Idea

It consists of 4 python files main.py, employee_data.py, operations.py, authentication.py.

The main contains function Calling.

The employee_data.py contains the data manipulation functions.

The operations.py contains the functions that use Employee Data.

The authentication.py contains the function used to authenticate user logging in.

Main.py

The main contains the calling of the functions created in the modules imported which are authentication.py, employee_data.py, and operations.py. The functions are called based on the user input from 1 to 8.

```python
import authentication as authentication
import employee_data as ed
import operations as op
import os


print("Welcome to the Employee Management System")
print("please login to continue")
auth=authentication.login()

if auth:
    while True:
        os.system('cls')
        op.options()
        choice=int(input("Enter your Choice"))
        if choice == 1:
            ed.add_employee()
        elif choice == 2:
            ed.remove_employee()
        elif choice == 3:
            ed.update_employee()
        elif choice == 4:
            op.display_info()
        elif choice == 5:
            op.calculate_bonus()
        elif choice == 6:
            op.calculate_discount()
        elif choice == 7 :
            op.Remind_legal_holidays()
        elif choice == 8 :
            op.exit()
        else:
            print("Wrong Choice, Please try Again! ")
```

Employee_data.py

1. Add_employee

The function has the implementation steps to take the data from the user and store it in the list.

```
def add_employee():
    print(".. Enter Employee Information ..")
    id=int(input("Enter the Employee ID: "))
    if(check_duplicate(id)):
        print(".. ID already exists ..")
    else:
        temp_dict={"ID":id,
                "Name":input("Enter the Employee Name "),
                "Department":input("Enter the Employee Department "),
                "Salary":int(input("Enter the Employee Salary ")),
                "password":int(getpass.getpass("Enter the Employee password ")),
                "absence_days":int(input("Enter the Employee absence days "))}
        Employee_list.append(temp_dict)
        print(".. Employee is added successfully ..")
```

2. Remove_employee

The function has the implementation steps to remove an employee record from the system based on an ID taken from the user.

```
def remove_employee():
    print(".. Enter the Employee Information to be removed ..")
    id=int(input("Enter the employee ID: "))
    for i in range(len(Employee_list)):
        if Employee_list[i].get("ID") == id:
            del Employee_list[i]
            print(".. Employee is removed Successfully ..")
            break
    else:
        print(".. Employee ID is not found ..")
```

3. Update_employee

The function has the implementation steps to update an employee record

In the system, based on an ID taken from the user.

```
def update_employee():
    print(".. Enter Employee Information to be updated ..")
    id=int(input("Enter the employee ID: "))
    for i in range(len(Employee_list)):
        if Employee_list[i].get("ID") == id:
            Employee_list[i].update({"Name":input("Enter the Employee Name ")})
            Employee_list[i].update({"Department":input("Enter the Employee Department ")})
            Employee_list[i].update({"Salary":int(input("Enter the Employee Salary "))})
            Employee_list[i].update({"password":int(input("Enter the Employee Password "))})
            Employee_list[i].update({"absence_days":int(input("Enter the Employee absence days "))})
            print(".. Employee is updated Successfully ..")
            break
    else:
        print("Employee ID is not found")
```

4. Check_duplicates

The function has the implementation steps to take an ID and check if it exists in the system or not so the IDs are unique in the system.

```
def check_duplicate(id):
    for i in range(len(Employee_list)):
        if Employee_list[i].get("ID") == id:
            return 1
```

Operations.py

1. Display_info

The function has the implementation steps to show an employee record based on the ID given by the user.

```
def display_info():
    id=int(input("Enter Employee ID: "))
    for i in range(len(ed.Employee_list)):
        if ed.Employee_list[i].get("ID") == id:
            print("Employee Information")
            print("Employee Name: ",ed.Employee_list[i].get("Name"))
            print("Employee Department: ",ed.Employee_list[i].get("Department"))
            print("Employee Salary: ",ed.Employee_list[i].get("Salary"))
            print("Employee absence days: ",ed.Employee_list[i].get("absence_days"))
            break
    else:
        print(".. Employee not Found ..")
```

2. Calculate_bonus

The function has the implementation steps to calculate bonus as a percentage of Employee Salary.

```
def calculate_bonus():
    id=int(input("Enter Employee ID: "))
    for i in range(len(ed.Employee_list)):
        if ed.Employee_list[i].get("ID") == id:
            bonus=ed.Employee_list[i].get("Salary")*0.1
            print("Bonus Calculation")
            print("Bonus: ",bonus)
            break
    else:
        print(".. Employee not Found ..")

        print(".. Employee not Found ..")
```

3. Calculate_discount

The function has the implementation steps to calculate discount as a percentage of Employee Salary.

```
def calculate_discount():
    id=int(input("Enter Employee ID: "))
    for i in range(len(ed.Employee_list)):
        if ed.Employee_list[i].get("ID") == id:
            discount=ed.Employee_list[i].get("Salary")*0.05
            print("Discount Calculation")
            print("Discount: ",discount)
            break
    else:
        print(".. Employee not Found ..")
```

4. Remind_legal_holidays

The function has the implementation steps to show Employee Holidays left according to the absences days They took.

```
def Remind_legal_holidays():
    id=int(input("Enter Employee ID: "))
    for i in range(len(ed.Employee_list)):
        if ed.Employee_list[i].get("ID") == id:
            legal_holidays=10-(ed.Employee_list[i].get("absence_days"))
            print("Reminder")
            print("Remaining Legal Holidays: ",legal_holidays)
            break
    else:
        print(".. Employee not Found ..")
```

5. Exit

The function has the implementation steps to terminate the program.

```
def exit():
    sys.exit("Thank you for using the Employee Management System. Goodbye!")
```

6. Options

The function has the implementation steps to show the options available to the user to choose from.

```python
def options():
    print("Select an option")
    print("1. Add Employee Information")
    print("2. Remove Employee Information")
    print("3. Update Employee Information")
    print("4. Display Employee Information")
    print("5. Calculate Bonus")
    print("6. Calculate Discount")
    print("7. Remind Legal Holidays")
    print("8. Exit")
```

Authentication.py

Login

The function has the implementation steps to verify the user login to the system using his ID and password.

```python
import employee_data as ed
import getpass

def login():
    for j in range(1,4):
        for i in range(len(ed.Employee_list)):
            id=int(input("Enter Employee ID: "))
            if ed.Employee_list[i].get("ID")==id:
                password =int(getpass.getpass("Enter the Employee password "))
                if ed.Employee_list[i].get("password")==password:
                    print(" Login Successful! ")
                    return 1
                else:
                    print("Login Failed, Please try again. Attempts left: ",3-j)
    return 0
```