Name-Kushvinder singh
Student ID-12411069

Standalone Project:-

```
class Employee:
    def __init__(self, name, age, designation):
        self.name = name
        self.age = age
        self.designation = designation
        self.salary = self.set_initial_salary()

    def set_initial_salary(self):
        if self.designation.upper() == 'P':
            return 25000
        elif self.designation.upper() == 'M':
            return 30000
        elif self.designation.upper() == 'T':
            return 20000
        else:
            return 0
```

```
    def display(self):
        info = f"Name: {self.name}, Age: {self.age}, Salary: {self.salary}, Designation:
{self.designation}"
        print(info)
        log_to_file(info)
```

```
    def raise_salary(self, percent):
        old_salary = self.salary
        self.salary += self.salary * (percent / 100)
        info = f"New salary of {self.name} after {percent}% hike: {old_salary} ->
{self.salary}"
        print(info)
        log_to_file(info)
```

```
def log_to_file(text):
    with open("employee_log.txt", "a") as file:
        file.write(text + "\n")
```

```
employees = []
```

```
while True:
    menu = ("\n--- Employee Management System ---\n"
            "1) Create Employee\n"
            "2) Display Employees\n"
            "3) Raise Salary\n"
            "4) Exit")
    print(menu)
    log_to_file(menu)
```

```
    choice = input("Enter your choice (1-4): ")
    log_to_file(f"User choice: {choice}")
```

```
    if choice == '1':
        name = input("Enter your Name: ")
        log_to_file(f"Entered Name: {name}")
```

```
        try:
            age = int(input("Enter your Age (18-60): "))
            log_to_file(f"Entered Age: {age}")
        except ValueError:
            print("Invalid input for age!")
            log_to_file("Invalid input for age!")
            continue
```

```
        if age < 18 or age > 60:
            print("Invalid age! Please enter between 18 and 60.")
            log_to_file("Invalid age! Please enter between 18 and 60.")
            continue
```

```
        print("Designation Options: P (25000), M (30000), T (20000)")
        designation = input("Enter your Designation (P/M/T): ")
        log_to_file(f"Entered Designation: {designation}")
```

```
        if designation.upper() not in ['P', 'M', 'T']:
            print("Invalid Designation! Choose P, M, or T.")
            log_to_file("Invalid Designation! Choose P, M, or T.")
            continue
```

```
        emp = Employee(name, age, designation)
        employees.append(emp)
        msg = f"Employee {name} added successfully!"
        print(msg)
        log_to_file(msg)
```

```
    elif choice == '2':
        if not employees:
            print("No employees to display.")
            log_to_file("No employees to display.")
        else:
            print("\nEmployee List:")
            log_to_file("Employee List:")
            for emp in employees:
                emp.display()
```

```
    elif choice == '3':
        search_name = input("Enter the name of the employee: ")
        log_to_file(f"Search for employee: {search_name}")
        found = False
        for emp in employees:
            if emp.name.lower() == search_name.lower():
                try:
                    percent_hike = float(input("Enter the percentage hike (e.g., 30 for 30%):
"))
                    log_to_file(f"Entered hike percent: {percent_hike}")
                except ValueError:
                    print("Invalid input for hike percent!")
                    log_to_file("Invalid input for hike percent!")
                    continue
                emp.raise_salary(percent_hike)
                found = True
                break
        if not found:
            msg = f"No employee found with the name: {search_name}"
            print(msg)
            log_to_file(msg)
```

```
    elif choice == '4':
        goodbye = "Thank you for using the application."
        print(goodbye)
        log_to_file(goodbye)
        break
```

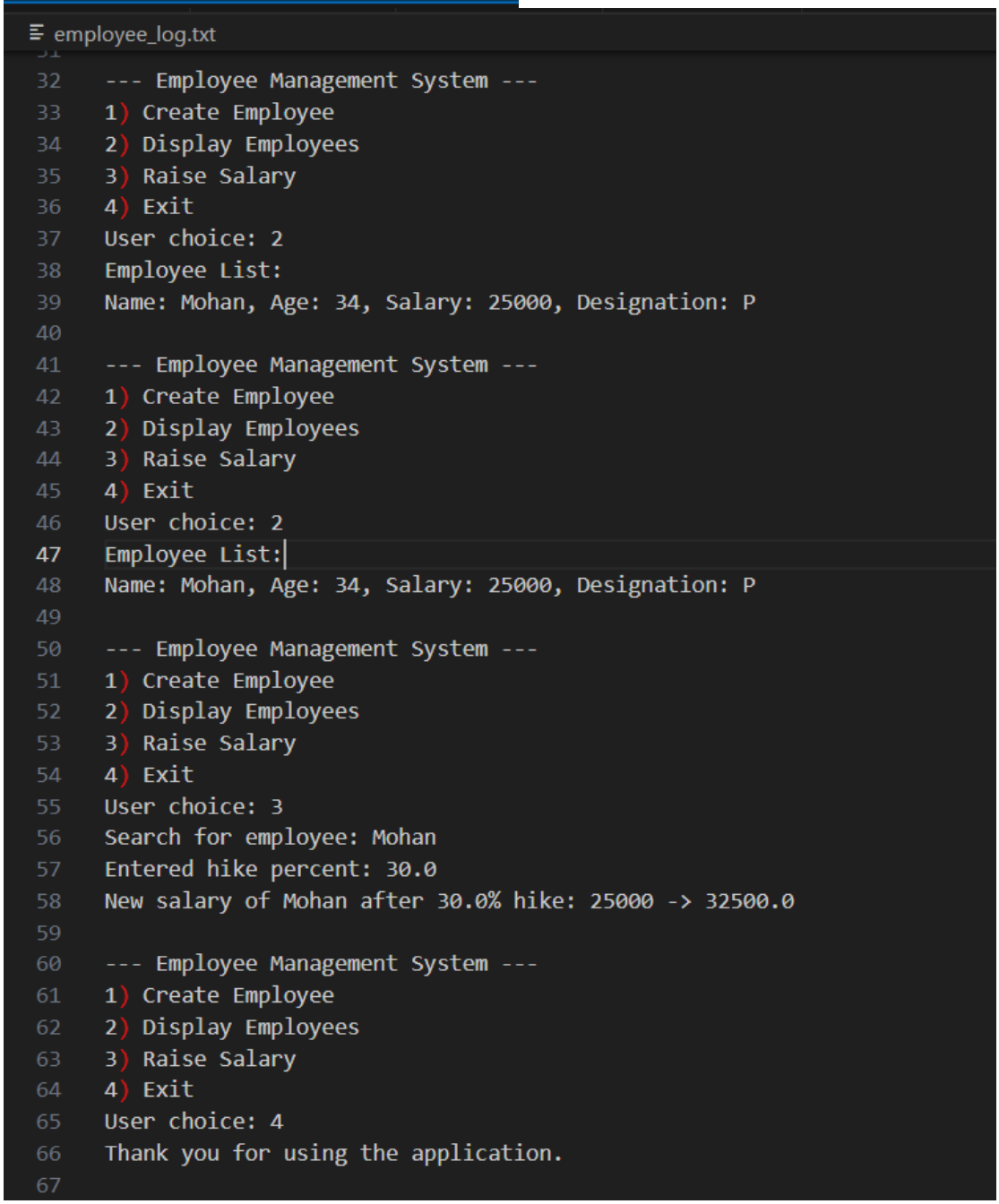
```
    else:
        msg = "Invalid choice! Please select from 1 to 4."
        print(msg)
        log_to_file(msg)
```

Output-:

```
--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit
Designation Options: P (25000), M (30000), T (20000)
Employee Mohan added successfully!

--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit

Employee List:
Name: Mohan, Age: 34, Salary: 25000, Designation: P

--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit

Employee List:
...
2) Display Employees
3) Raise Salary
4) Exit
Thank you for using the application.
```

Created a File :

employee_log.txt



```
--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit
User choice: 2
Employee List:
Name: Mohan, Age: 34, Salary: 25000, Designation: P

--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit
User choice: 2
Employee List:
Name: Mohan, Age: 34, Salary: 25000, Designation: P

--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit
User choice: 3
Search for employee: Mohan
Entered hike percent: 30.0
New salary of Mohan after 30.0% hike: 25000 -> 32500.0

--- Employee Management System ---
1) Create Employee
2) Display Employees
3) Raise Salary
4) Exit
User choice: 4
Thank you for using the application.

```