Project Report: Salary Calculator in Python

1. Title

Python Salary Calculator Application

---

2. GitHub Repository

Source Code:
https://github.com/AdiChakote/ds-salarycalculator

You can access the full project code, version history, and documentation in this repository.

---

3. Introduction

This project is a simple **console-based salary calculator** developed in **Python**. It computes the **monthly salary** of an employee based on:

- Employee name
- Daily salary
- Number of days worked

The application also applies a **10% bonus** if the employee worked 26 or more days, or a **10% deduction** if they worked 20 or fewer days. The project demonstrates core Python programming concepts such as functions, arithmetic operations, and conditional logic.

---

4. Objective

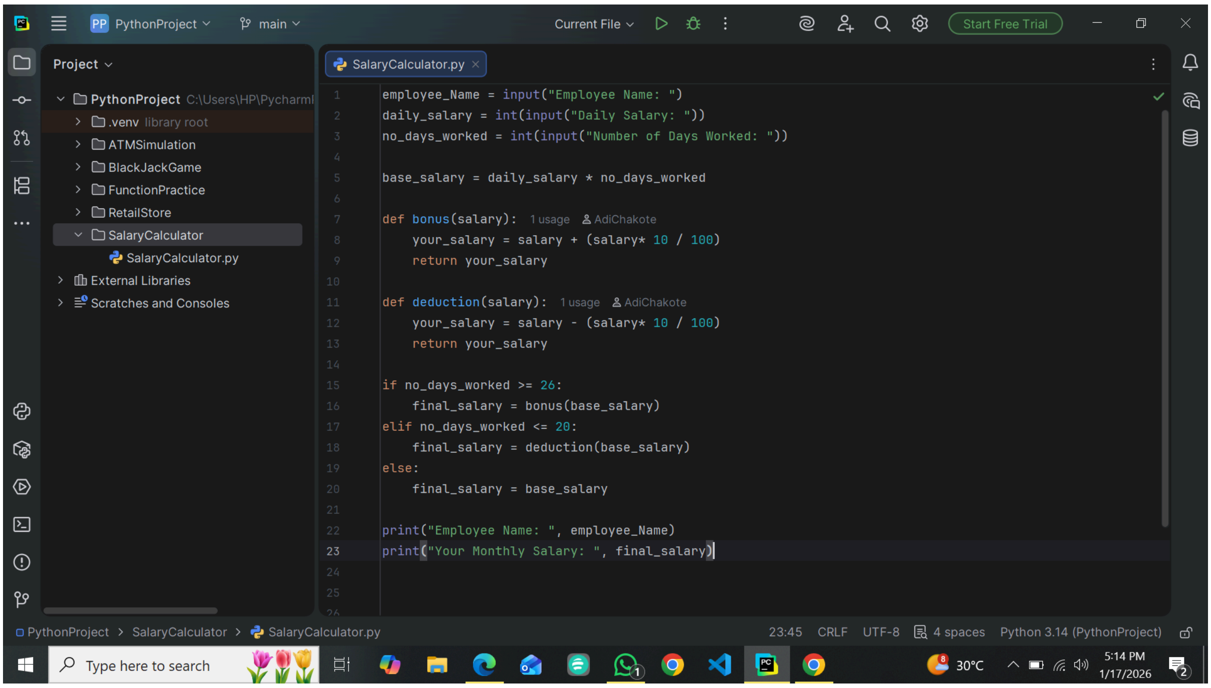
The primary objective of this project is to:

- Accept user input for employee details
- Calculate base salary
- Apply bonus or deduction based on attendance
- Display the final monthly salary
- Provide a simple and intuitive user experience

5. Tools & Technologies

- **Programming Language:** Python
- **IDE/Editor:** Any Python editor (e.g., VS Code, PyCharm, IDLE)
- **Version Control:** Git and GitHub
- **Platform:** Command-line interface

6. Source Code



```python
employee_Name = input("Employee Name: ")
daily_salary = int(input("Daily Salary: "))
no_days_worked = int(input("Number of Days Worked: "))

base_salary = daily_salary * no_days_worked

def bonus(salary):
    your_salary = salary + (salary* 10 / 100)
    return your_salary

def deduction(salary):
    your_salary = salary - (salary* 10 / 100)
    return your_salary

if no_days_worked >= 26:
    final_salary = bonus(base_salary)
elif no_days_worked <= 20:
    final_salary = deduction(base_salary)
else:
    final_salary = base_salary

print("Employee Name: ", employee_Name)
print("Your Monthly Salary: ", final_salary)
```

7. Code Listing

```python
employee_Name = input("Employee Name: ")
daily_salary = int(input("Daily Salary: "))
no_days_worked = int(input("Number of Days Worked: "))

base_salary = daily_salary * no_days_worked

def bonus(salary):
    your_salary = salary + (salary* 10 / 100)
    return your_salary

def deduction(salary):
    your_salary = salary - (salary* 10 / 100)
    return your_salary

if no_days_worked >= 26:
    final_salary = bonus(base_salary)
elif no_days_worked <= 20:
    final_salary = deduction(base_salary)
else:
    final_salary = base_salary

print("Employee Name: ", employee_Name)
print("Your Monthly Salary: ", final_salary)
```

---

8. Feature Explanation

8.1 Input Section

```
employee_Name = input("Employee Name: ")
daily_salary = int(input("Daily Salary: "))
no_days_worked = int(input("Number of Days Worked: "))
```

- Collects employee name, daily salary, and total working days.
- `int()` ensures numeric input for salary and days.

8.2 Base Salary Calculation

```
base_salary = daily_salary * no_days_worked
```

- Computes total salary before any bonus or deduction.

8.3 bonus() Function

```
def bonus(salary):
    your_salary = salary + (salary * 10 / 100)
    return your_salary
```

- Adds **10% bonus** to the base salary.
- Returns the updated salary.

8.4 deduction() Function

```
def deduction(salary):
    your_salary = salary - (salary * 10 / 100)
    return your_salary
```

- Applies a **10% deduction** for fewer working days.
- Returns the updated salary.

8.5 Conditional Salary Adjustment

```
if no_days_worked >= 26:
    final_salary = bonus(base_salary)
elif no_days_worked <= 20:
    final_salary = deduction(base_salary)
else:
    final_salary = base_salary
```

- If the employee worked **26 or more days**, they receive a **10% bonus**.
- If they worked **20 days or fewer**, a **10% deduction** is applied.
- Otherwise, base salary remains unchanged.

8.6 Final Output

```
print("Employee Name: ", employee_Name)
print("Your Monthly Salary: ", final_salary)
```

- Displays employee name and final computed salary.

---

9. Sample Output

Example Input

```
Employee Name: John
Daily Salary: 400
Number of Days Worked: 27
```

Output

```
Employee Name: John
Your Monthly Salary: 11880.0
```

---

10. Features Summary

The program:

- Takes user input for employee salary details
- Computes base salary
- Applies conditional bonus or deduction
- Prints final salary
- Uses modular functions to structure logic

---

11. Limitations

While functional, the current version lacks:

- **Input validation** (e.g., negative values handled)
- **Loop support** for multiple employees
- **Error handling** for incorrect inputs
- **Formatted output** (currency symbols, rounding)

---

12. Future Scope

You can expand this project in several ways:

12.1 Add Validation

Ensure salary and days are valid numbers:

- Reject negatives
- Handle non-integer entries

12.2 Multiple Employee Support

Allow the user to calculate salary for:

- Many employees in one run
- Save results to a file (CSV/JSON)

12.3 Advanced Salary Structure

Include:

- Tax deduction
- Overtime pay
- Allowances (HRA/DA/TA)

12.4 Graphical User Interface

Use libraries like:

- **Tkinter**
- **PyQt**
  to make this a desktop application.

12.5 Web App Version

Use frameworks:

- **Flask**
- **Django**
  to convert this into a web-based salary calculator.

---

13. Conclusion

This salary calculator demonstrates fundamental programming skills and basic business logic. It serves as a strong foundation for more advanced payroll systems and can be improved with validation, UI, file support, and deployment as a complete application.