**Project file**

**On**

**Employee salary management system**

**Submitted for:**

**Document automation Using Python**

**In**

**Computer Programming**

**Submitted to:**

Mr. Muhammad Sultan

**Submitted by:**

Harshil Patel (200552678)

Om Patel(200556124)

Mishant Dalwadi(200555960)

Nirmit Tandel(200548670)

**Automation Wizard: Employee salary management system:**

**Introduction:**

The offered code is a Python script that updates or produces an Excel file with the employee details, handles employee data, computes salaries, and creates a structured report. It also uses a bar graph to display employee pay. The programme shows how to use datetime, files, classes, and third-party libraries like matplotlib and openpyxl.

**Problem Statement:**

The code's main objective is to automate the management of employee data, wage calculation, and report generation processes. It also intends to construct and maintain an Excel file for the purpose of storing employee data and displaying the salary distribution as a bar graph.

**Code Analysis:**

**1. Employee Class:**

The Employee class is defined to represent individual employees, with attributes such as name, hours worked, and hourly rate.

**2. calculate\_salary Function:**

This function computes the salary of an employee based on the hours worked and hourly rate, considering a tax factor of 1.13.

**3. format\_employee\_data Function:**

Formats employee data into a human-readable report, including employee details, taxes, and salary. It also calculates and displays the total salary for all employees.

**4. create\_excel\_report Function:**

This function creates or updates an Excel file with employee data. It utilizes openpyxl to interact with Excel files, handling the addition of headers, employee details, and total salary.

**5. create\_graph Function:**

Uses the matplotlib library to create a bar graph visualizing employee salaries.

**6. get\_valid\_input Function:**

A utility function for obtaining valid user input, with optional data type validation.

**7. Example Usage:**

The script prompts the user to input the number of employees, then iteratively collects details for each employee (name, hours worked, hourly rate).

It generates a formatted report, updates an Excel file with employee details, and creates a bar graph visualizing employee salaries.

**Conclusion:**

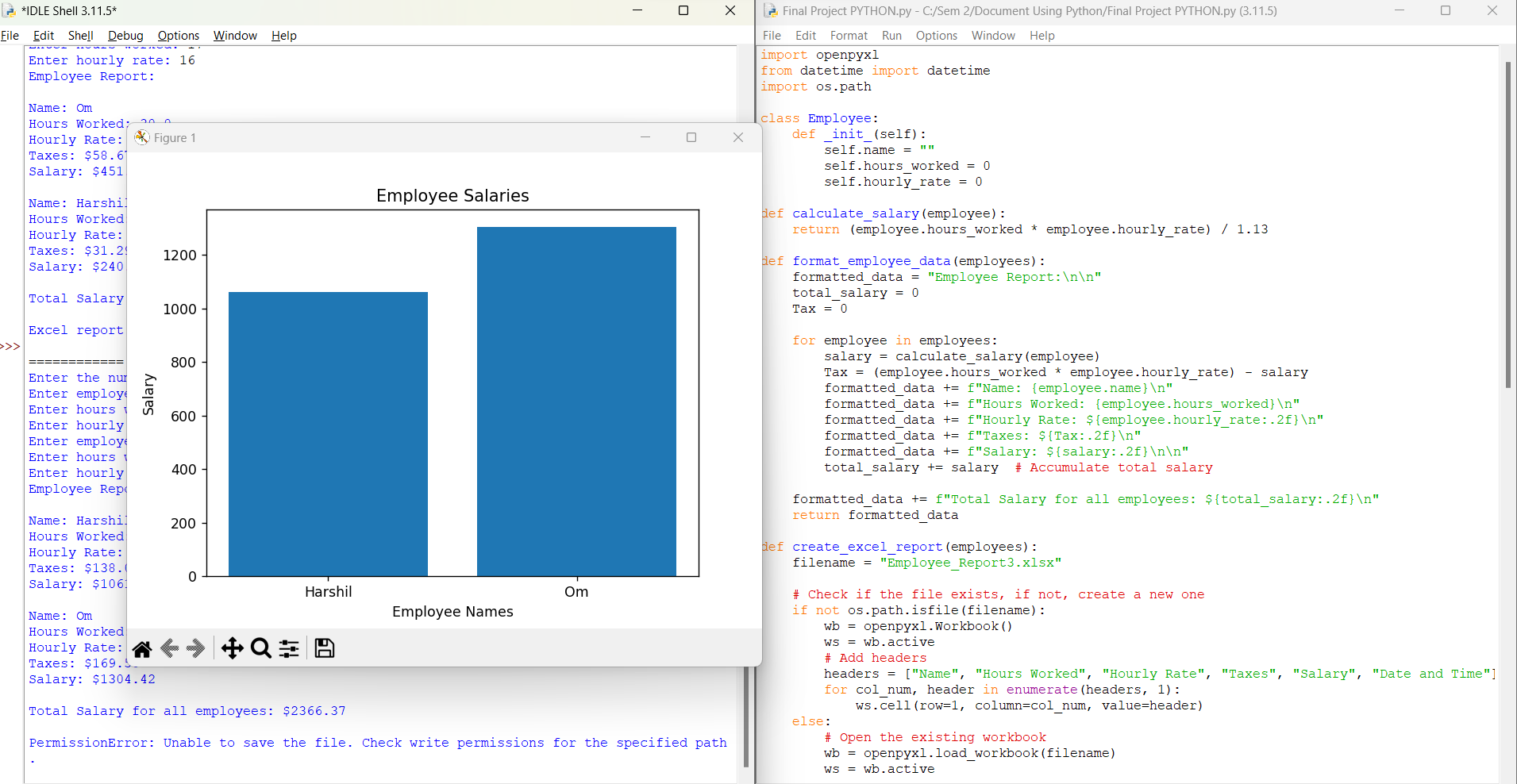
Managing employee data, computing pay, and offering insights into salary distribution through visualizations are all successfully handled by the code. This script provides an effective method of handling and analyzing employee data. It is well-organized and modular, and it may be expanded for larger-scale applications. External libraries are also used to improve functionality and offer a more user-friendly experience.

Because of its modular nature, the script is easy to maintain and can be enhanced in the future. Still, more advancements might be made in error management for file-related functions and possibly adding more sophisticated data visualization methods for a more thorough examination of employee pay.

How it Works….

A screenshot of a computer

Description automatically generated



This graph will be generated at the end.