
📂 Dataset 1: `employees.csv`

```
EmployeeID,Name,Department,Salary,JoiningDate
1,Ali,HR,50000,2021-03-15
2,Neha,IT,60000,2022-01-10
3,Ravi,Finance,55000,2020-07-23
4,Sara,IT,70000,2023-05-19
5,Vikram,HR,52000,2022-09-30
```

📂 Dataset 2: `projects.csv`

```
ProjectID,EmployeeID,ProjectName,HoursAllocated
101,2,AI Chatbot,120
102,3,ERP System,200
103,3,Payroll Automation,150
104,5,Cloud Migration,100
```

🐍 Python Full-Spectrum Assessment (No Solutions)

🔹 Section 1: Python Basics & Control Flow

Q1. Write a Python program to print all odd numbers between 10 and 50.

Q2. Create a function that returns whether a given year is a leap year.

Q3. Write a loop that counts how many times the letter `a` appears in a given string.

🔹 Section 2: Collections (Lists, Tuples, Sets, Dicts)

Q4. Create a dictionary from the following lists:

```
keys = ['a', 'b', 'c']
values = [100, 200, 300]
```

Q5. From a list of employee salaries, extract:

- The maximum salary
- All salaries above average
- A sorted version in descending order

Q6. Create a set from a list and remove duplicates. Show the difference between two sets:

```
a = [1, 2, 3, 4]
b = [3, 4, 5, 6]
```

🔹 Section 3: Functions & Classes

Q7. Write a class `Employee` with `__init__`, `display()`, and `is_high_earner()` methods. An employee is a high earner if salary > 60000.

Q8. Create a class `Project` that inherits from `Employee` and adds `project_name` and `hours_allocated`.

Q9. Instantiate 3 employees and print whether they are high earners.

Section 4: File Handling

Q10. Write to a file the names of employees who belong to the 'IT' department.

Q11. Read from a text file and count the number of words.

Section 5: Exception Handling

Q12. Write a program that accepts a number from the user and prints the square. Handle the case when input is not a number.

Q13. Handle a potential `ZeroDivisionError` in a division function.

Section 6: Pandas – Reading & Exploring CSVs

Q14. Load both `employees.csv` and `projects.csv` using Pandas.

Q15. Display:

- First 2 rows of `employees`
- Unique values in the `Department` column
- Average salary by department

Q16. Add a column `TenureInYears` = current year - joining year.

Section 7: Data Filtering, Aggregation, and Sorting

Q17. From `employees.csv`, filter all IT department employees with salary > 60000.

Q18. Group by Department and get:

- Count of employees
- Total Salary
- Average Salary

Q19. Sort all employees by salary in descending order.

Section 8: Joins & Merging

Q20. Merge `employees.csv` and `projects.csv` on `EmployeeID` to show project allocations.

Q21. List all employees who are not working on any project (left join logic).

Q22. Add a derived column `TotalCost = HoursAllocated * (Salary / 160)` in the merged dataset.
