

System Requirements

Specification Index

For

Employee Data Aggregation by Department

(Topic: Data Aggregation and Grouping)

Version 1.0

Employee Data Analysis Console

Project Abstract

The Employee Data Analysis Console is a Python application developed to manage and analyze employee data stored in CSV files. This system provides various functionalities to process and analyze the employee data, such as displaying the first few rows, extracting detailed information about the dataset, aggregating data by department, and exporting statistics to a new CSV file. The application is designed to help HR departments efficiently manage employee records, perform insightful analysis, and generate departmental statistics to support decision-making.

Business Requirements

The Employee Data Analysis Console will be used to:

- Load and analyze employee data stored in CSV files.
- Provide basic insights about the dataset including the first few rows and column data types.
- Group employee data by department to generate aggregate statistics like mean and median values for age and salary.
- Allow HR professionals to export aggregated departmental statistics to CSV for reporting purposes.

Constraints

Input Requirements

- CSV File:
 - The file must be a valid CSV format.
 - It must contain at least the following columns: Employee ID, Age, Salary, and Department.

- The file must be accessible from the given path in the system.

Functionality Constraints

- **Display Head:**
 - Must return the first 5 rows of the employee dataset in the DataFrame.
- **DataFrame Info:**
 - Must return the column names and data types of the DataFrame.
- **Group by Department:**
 - The employee data must be grouped by the Department column.
 - The system should compute the following statistics for each department:
 - **Age Mean:** Mean age of employees in the department.
 - **Salary Mean:** Mean salary of employees in the department.
 - **Age Median:** Median age of employees in the department.
 - **Salary Median:** Median salary of employees in the department.
 - **Employee Count:** Count of employees in the department.
- **Export Aggregated Data:**
 - The system must allow the aggregated statistics to be saved in a new CSV file.
 - The default output file should be named "department_statistics.csv".

Output Constraints

1. Display Format:

- The output must display the first 5 rows of the DataFrame when requested by the Display Head function.
- The DataFrame Info function must display column names and data types of the DataFrame.
- The Group by Department function must return aggregate statistics for each department.
 - Age_Mean: Mean of the ages in the department.
 - Salary_Mean: Mean of the salaries in the department.
 - Age_Median: Median of the ages in the department.
 - Salary_Median: Median of the salaries in the department.
 - Employee_Count: Count of employees in the department.
- The Export Aggregated Data function must save the aggregated data in a CSV file and return the aggregated data as output.

Template Code Structure

1. Data Loading Functions:

- `__init__(self, file_path)` — Load CSV data into a Pandas DataFrame.
- `display_head()` — Return the first 5 rows of the DataFrame.
- `dataframe_info()` — Return column names and data types of

the DataFrame.

- `group_by_department()` — Group the employee data by department and compute aggregate statistics.
- `export_aggregated_data(output_file="department_statistics.csv")` — Export aggregated statistics to a CSV file.

2. Input Section:

- Load CSV file containing employee data.

3. Data Processing Section:

- Display the first few rows of the data.
- Analyze the structure of the DataFrame (column names, data types).
- Group data by department and calculate statistics.

4. Output Section:

- Display aggregated departmental statistics.
- Save the results to a CSV file for reporting.

Execution Steps to Follow:

- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B** -command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
- These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
- To setup environment:
You can run the application without importing any packages
- To launch application:
python3 mainclass.py
- To run Test cases:
python3 -m unittest
- Before Final Submission also, you need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

Screen shot to run the program


To run the application

A screenshot of a terminal window with a dark background. The prompt is 'coder@dighe20250227t070305rz1fj5p3: /home/myproject/dighegmailcom_20250227T070305\$'. The command 'python3 <<scriptname>>.py' is entered. A red arrow points from the text 'python3 mainclass.py' below to the '<<scriptname>>.py' part of the command in the terminal.

```
OK
coder@dighe20250227t070305rz1fj5p3: /home/myproject/dighegmailcom_20250227T070305$ python3 <<scriptname>>.py
```

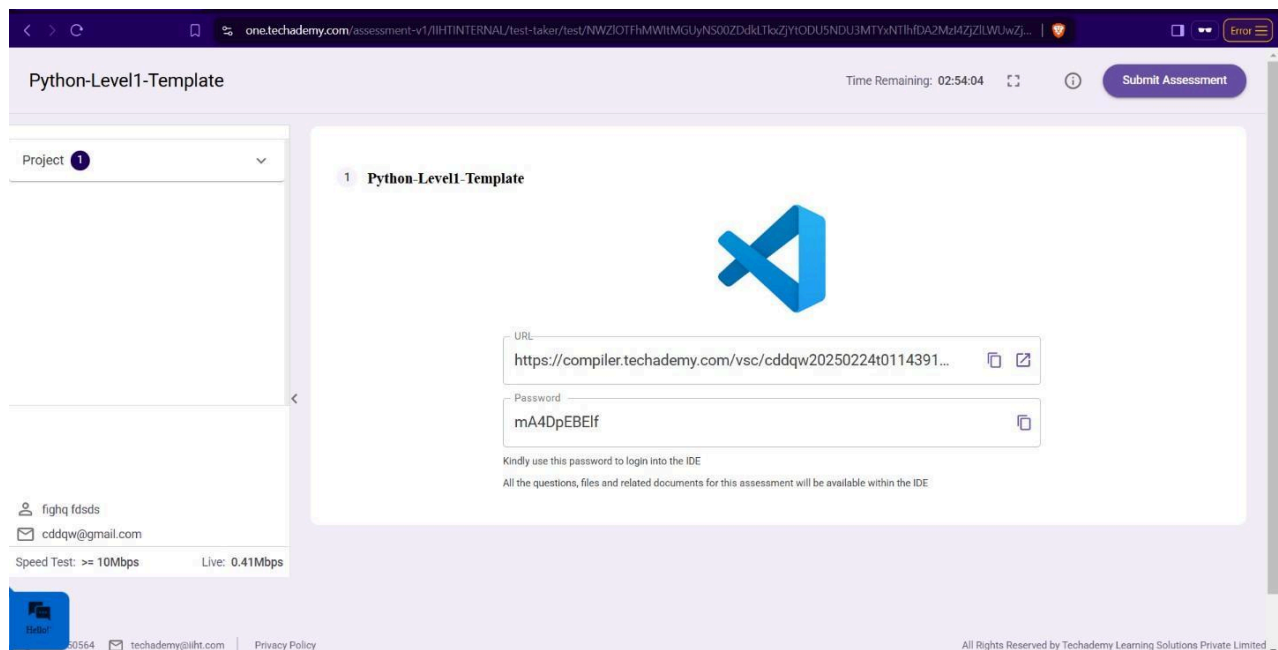
python3 mainclass.py

```
● coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 -m unittest
TestBoundary = Passed
.TestExceptional = Passed
.TestCalculateTotalDonations = Failed
.TestCalculateTotalStockValue = Failed
.TestCheckFrankWhiteDonated = Failed
```



To run the testcase

`python3 -m unittest`



- **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.**