System Requirements Specification Index

For

Handling Missing Values in Employee Data

(Topic: Data Cleaning and Preprocessing)

Version 1.0

Employee Data Analysis Console

Project Abstract

The Employee Data Analysis Console is a Python-based application designed for processing employee data stored in CSV files. The application uses Pandas, a powerful data manipulation library, to load, clean, and analyze employee data efficiently. It provides functionalities for displaying the first few rows of data, retrieving the structure and information of the DataFrame, handling missing values, and exporting updated data into a new CSV file. The application is essential for human resources departments to process large datasets, ensure data quality by handling missing values, and allow users to quickly generate updated and clean datasets.

Business Requirements:

The application must meet the following business requirements:

- Efficiently load employee data from CSV files.
- Provide a method to inspect the first few rows of the data.
- Offer the capability to check the structure and data types of the columns.
- Provide a feature for handling missing values, particularly for numeric columns.
- Allow users to export the cleaned dataset to a new CSV file for further use.

Constraints

Input Requirements

- Employee Data File:
  The data must be in CSV format and should include columns such as employee ID, name, department, salary, and hire date. Example: `employee_data.csv` - Contains various fields like "Employee ID", "Name", "Salary", "Hire Date".

- **Missing Data Handling:**
  If the data contains missing values in numeric columns (e.g., salary), these must be replaced with the column mean.

Output Constraints

1. **Display Output Format:**
   - **Display the first 5 rows of the employee data using a method.**
   - **Display the column names and their corresponding data types.**
2. **Data Cleaning:**
   - **Replace missing numeric values with the mean of the respective column.**
   - **Ensure no missing values remain in the dataset before exporting.**
3. **Export Format:**
   - **The cleaned employee data must be saved as a new CSV file.**
   - **The exported file should not include row indices.**

Functionality Requirements

1. **Data Inspection Functions:**
   - `display_head()` **– Display the first 5 rows of the employee dataset.**
   - `dataframe_info()` **– Display the column names and their data types in the dataset.**
2. **Data Cleaning Functions:**
   - `handle_missing_values()` **– Replace missing values in numeric columns with the mean of the respective column.**

3. **Export Function:**
    - `export_updated_csv()` **– Save the cleaned DataFrame to a new CSV file with the updated data.**
4. **Validation Functions:**
    - `check_missing_values()` **– Validate that the cleaned DataFrame has no missing values.**

Required Output Format

- **Show the first 5 rows of the employee data when** `display_head()` **is called.**
- **Show the structure of the DataFrame with column names and data types using** `dataframe_info()`**.**
- **Display a message confirming the successful handling of missing values after calling** `handle_missing_values()`**.**
- **After calling** `export_updated_csv()`**, display the location where the updated CSV file has been saved.**

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:

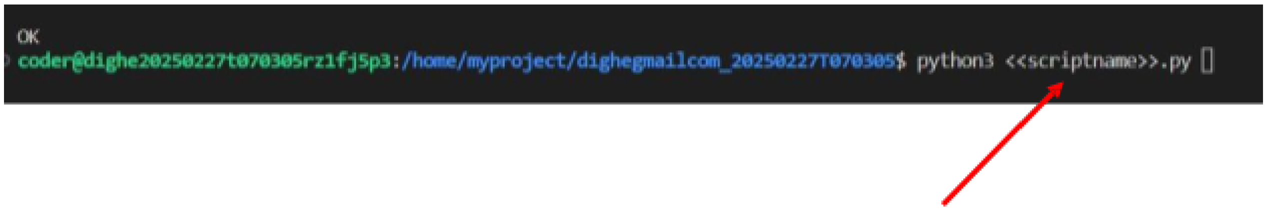
  You can run the application without importing any packages
- To launch application:
  python3 mainclass.py
- To run Test cases:

  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.

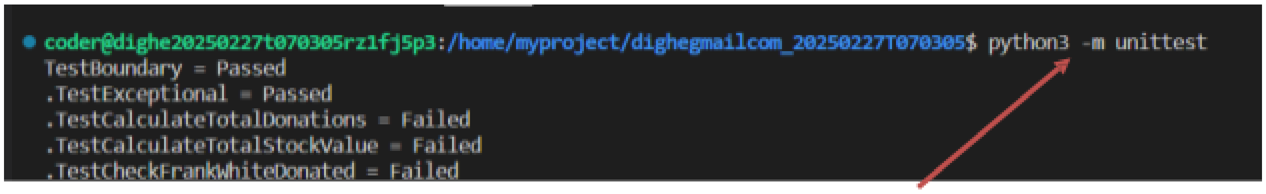
Screen shot to run the program

To run the application



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

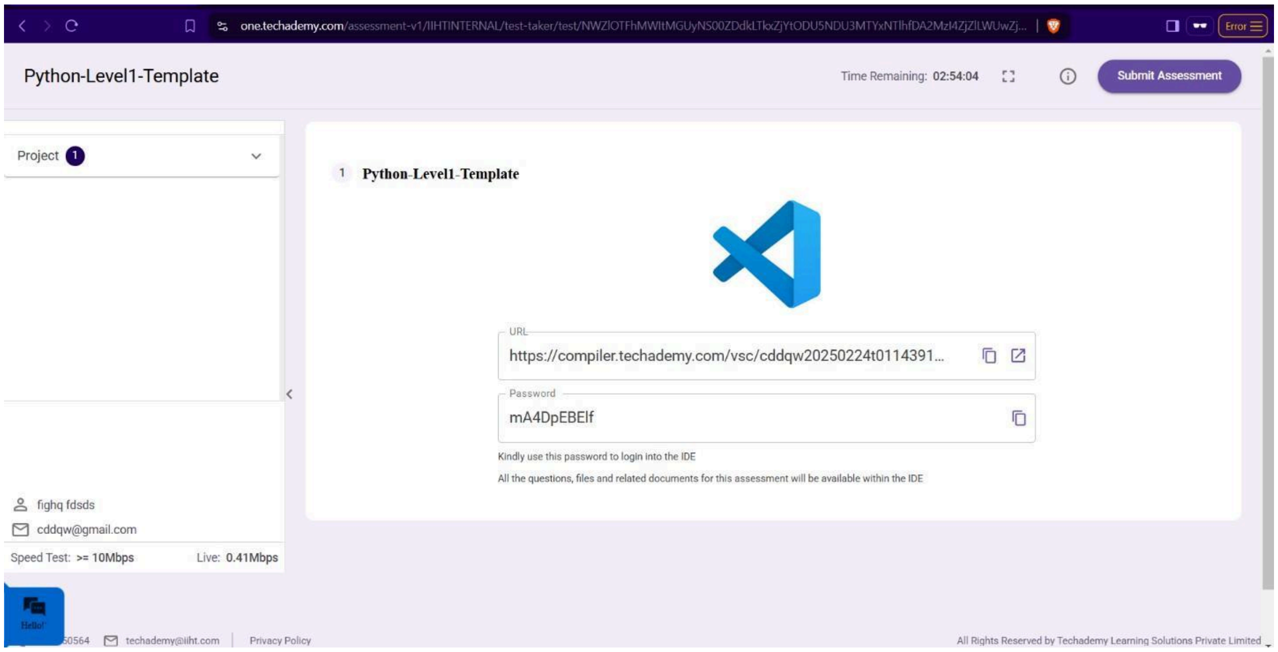
python3 mainclass.py



```
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


one.techademy.com/assessment-v1/IIHTINTERNAL/test-taker/test/NWZlOTFhMWItMGUyNS00ZDdkLTkxZjYtODU5NDU3MTYxNTlhfDA2MzI4ZjZlLWUwZj...

Python-Level1-Template

Time Remaining: 02:54:04

Submit Assessment

Project 1

1 Python-Level1-Template

URL
https://compiler.techademy.com/vsc/cddqw20250224t0114391...

Password
mA4DpEBElf

Kindly use this password to login into the IDE

All the questions, files and related documents for this assessment will be available within the IDE

fighq fdsds
cddqw@gmail.com
Speed Test: >= 10Mbps Live: 0.41Mbps

50564 techademy@iiht.com | Privacy Policy

All Rights Reserved by Techademy Learning Solutions Private Limited

- **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.**