System Requirements Specification Index

For

CSV File Handling and Data Conversion using Pandas and NumPy

(Topic: Integration with Other Libraries)

Version 1.0

Sales Data Processor Console

Project Abstract

The Sales Data Processor is a Python-based console application developed to process and analyze sales data from CSV files. The application allows users to read data from a CSV file containing sales information, convert the sales data into a NumPy array, and ensure proper handling of any errors such as missing or empty files. This tool is designed to be used by data analysts who need to perform further analysis or transformation on sales data, ensuring that the data is properly loaded and converted before any business logic or statistical analysis is performed.

Business Requirements:

1. **Data Processing**
   The application must handle reading and processing sales data from a CSV file. It must be capable of handling different types of errors, including missing files, empty files, and invalid data.
2. **Data Conversion**
   The application must convert the `total_sales` column in the CSV to a NumPy array for further analysis and computations.

Constraints

- **Input Requirements:**
  - **CSV File:**
    - **Must contain a column labeled `total_sales` with numeric data.**
    - **File must be in CSV format.**
  - **File Path:**
    - **Must be a valid file path.**
    - **File must be accessible and readable by the system.**
    - **Example: `"sales_data.csv"`**
  - **Sales Data:**
    - **Must be numeric for conversion into a NumPy array.**
    - **Example: `1000`, `2000`, `5000`**
- **Conversion Constraints:**
  - **CSV Reading:**
    - **File must exist at the provided path.**
    - **Must handle empty or invalid CSV files.**
  - **Sales Column Conversion:**

        - **Must convert `total_sales` column to NumPy array.**
        - **If the column doesn't exist, raise an appropriate error.**
- **Output Constraints:**
    - **Conversion Output:**
        - **The output must be a NumPy array of sales data.**
        - **If the `total_sales` column is missing, return an error message.**

Required Output Format:

1. **CSV Reading Results:**
    - **Display an error message if the file cannot be found.**
    - **Display an error message if the file is empty.**
    - **Provide feedback on the successful reading of the file.**
2. **Sales Data Conversion:**
    - **Convert `total_sales` into a NumPy array.**
    - **If the `total_sales` column is missing, raise an error message indicating the issue.**

Template Code Structure:

1. **Class: `SalesDataProcessor`**
    - **Methods:**
        - **`__init__(file_path)`: Initialize with the file path to the CSV.**
        - **`read_csv()`: Read the CSV file into a pandas DataFrame.**
        - **`convert_sales_column()`: Convert the `total_sales` column to a NumPy array.**
2. **Input Section:**
    - **Get file path from the user or configuration.**
    - **Ensure the file exists and is accessible.**
3. **Conversion Section:**
    - **Convert the `total_sales` column from the DataFrame to a NumPy array.**
    - **Handle missing or invalid columns gracefully.**
4. **Output Section:**
    - **Display the converted sales data (NumPy array).**
    - **Handle errors and display meaningful messages when required.**

Execution Steps to Follow:

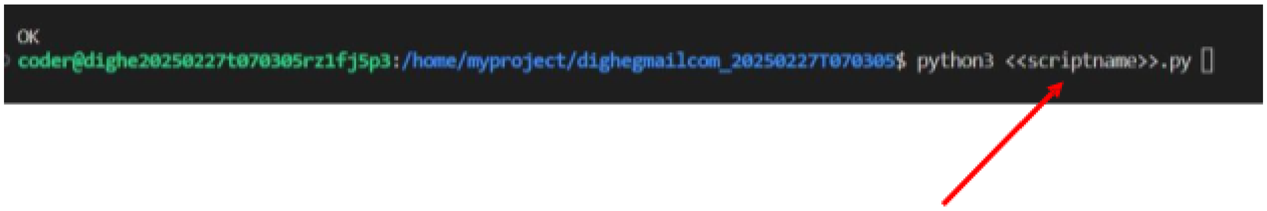
- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want 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.

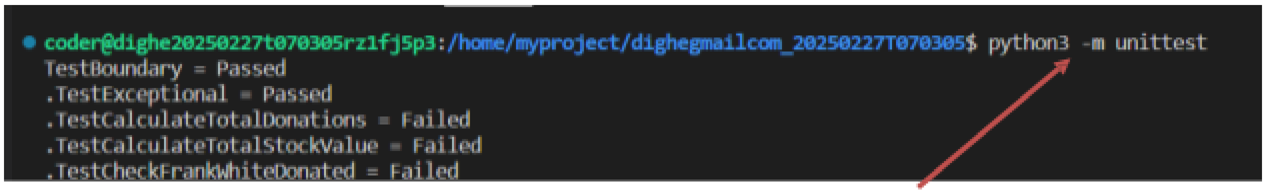
Screen shot to run the program

To run the application



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

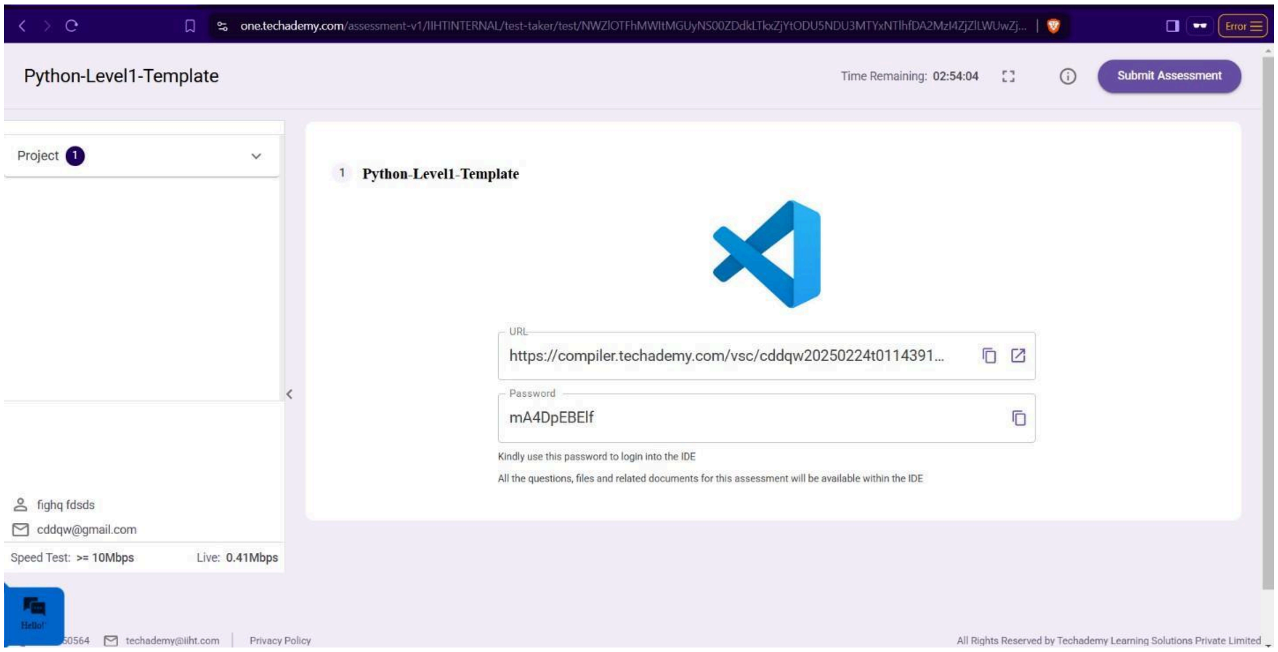
python3 mainclass.py python3



```
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 unittest6


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
fighq fdsds
cddqw@gmail.com
Speed Test: >= 10Mbps
Live: 0.41Mbps
techademy@iiht.com
Privacy Policy
All Rights Reserved by Techademy Learning Solutions Private Limited

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

- **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.**