System Requirements Specification Index

For

NumPy Array Operations for Sales Data Analysis

(Topic: Mathematical and Statistical Operations)

Version 1.0

Sales Data Analysis Console

Project Abstract

The Sales Data Analysis Console is a Python application designed to perform various statistical calculations on sales data for two products. This system analyzes daily sales data for two products, computes the maximum, minimum, and total sales, and provides the results in an easy-to-understand format. The console application uses NumPy to handle and process the sales data, ensuring efficient and fast computations. The goal is to offer insightful sales performance metrics for decision-making purposes in retail environments.

Business Requirements:

The system should perform the following tasks:

- **Calculate the maximum sales value across both products.**
- **Calculate the minimum sales value across both products.**
- **Calculate the sum of the total sales of both products.**
- **Handle potential empty datasets gracefully by raising appropriate error messages.**

Constraints

Input Requirements:

- **Product 1 Sales:**
  - **Must be provided as a list of integers representing daily sales.**
  - **Example:** `[20, 30, 50, 60]`
- **Product 2 Sales:**
  - **Must be provided as a list of integers representing daily sales.**
  - **Example:** `[10, 40, 25, 35]`

Sales Data Format:

- **The sales data must be passed as NumPy arrays of type** `int32`**.**

Conversion Constraints:

- **Max Sales Calculation:**
  - **Calculate the maximum sales value from the combined sales of both products.**
  - **Use** `np.max()` **for this calculation.**

    - Ensure that no empty sales data is passed to avoid errors.
- **Min Sales Calculation:**
    - Calculate the minimum sales value from the combined sales of both products.
    - Use `np.min()` for this calculation.
    - Ensure that no empty sales data is passed to avoid errors.
- **Total Sales Calculation:**
    - Calculate the sum of the sales values for both products.
    - Use `np.sum()` for this calculation.
    - Ensure that no empty sales data is passed to avoid errors.

Output Constraints:

1. **Display Format:**
    - Show the calculated maximum, minimum, and total sales for the provided sales data.
    - Handle empty sales data gracefully and output appropriate error messages.

Required Output Format: - "Maximum Sales: {value}" - "Minimum Sales: {value}" - "Total Sales: {value}"

Template Code Structure:

1. **Initialization:**
    - Accept sales data for two products.
    - Store the data as NumPy arrays for efficient computation.
2. **Sales Analysis Functions:**
    - `find_max()` — Find the maximum sales value.
    - `find_min()` — Find the minimum sales value.
    - `find_sum()` — Find the total sales.
3. **Error Handling:**
    - Raise a `ValueError` if the sales data is empty.
4. **Output Section:**
    - Display the maximum, minimum, and total sales.

<u>**Execution Steps to Follow:**</u>

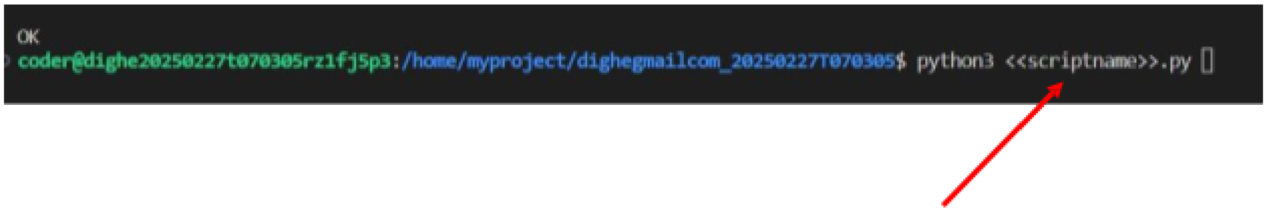
- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want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

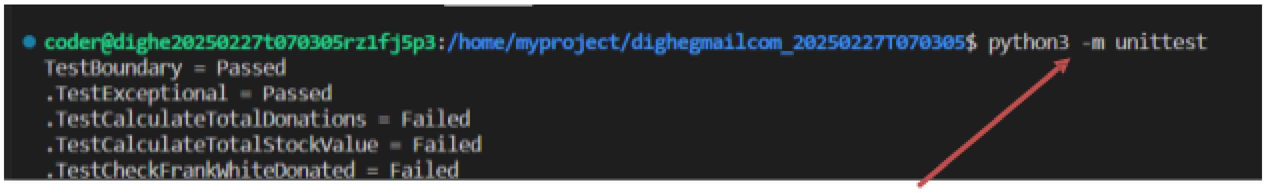
<u>Screen shot to run the program</u>

To run the application



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

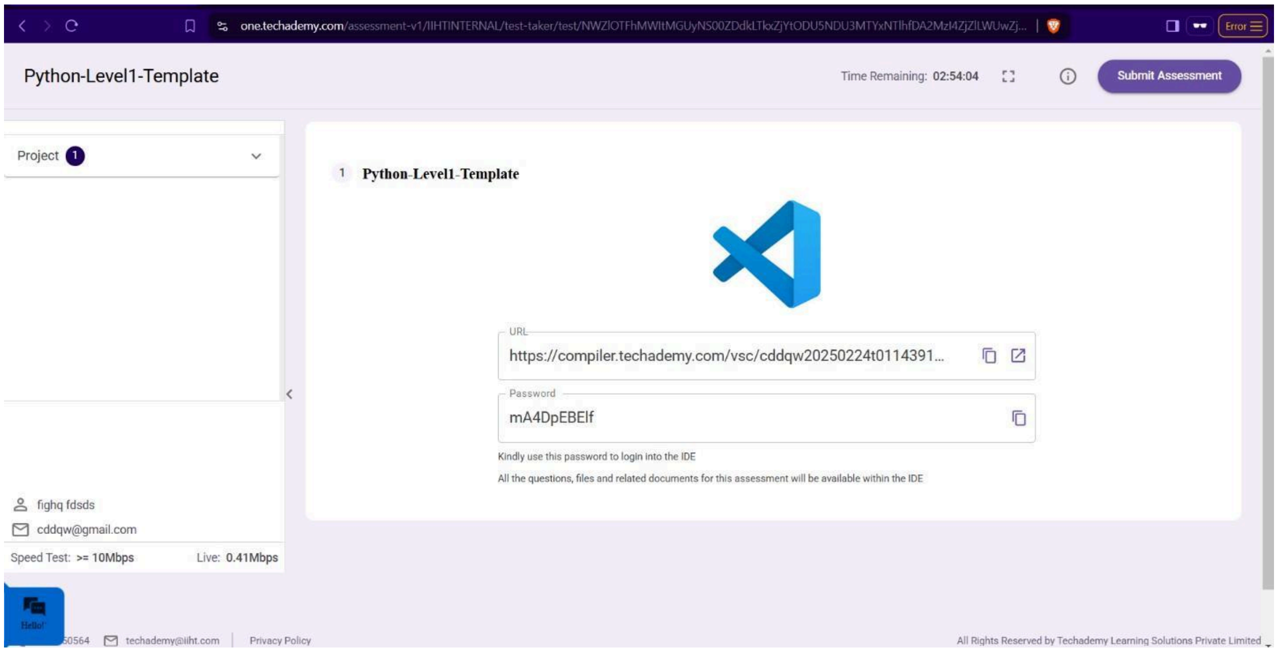
python3 mainclass.py python3



```
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.**