System Requirements Specification Index

For

NumPy Array Operations for Student Test Scores Analysis

(Topic: Mathematical and Statistical Operations)

Version 1.0

Student Test Scores Analysis Console

Project Abstract

The Student Test Scores Analysis Console is a Python application that processes and analyzes student test scores to provide statistical insights such as the mean, median, and mode. This system is designed to support educators and analysts in evaluating test performance, identifying trends, and making data-driven decisions for improving student outcomes. It utilizes NumPy and SciPy to compute statistical measures, ensuring quick and accurate analysis of student performance data. The system is essential for generating reports and visualizations that aid in academic performance assessments.

Business Requirements

The system should be able to handle a list of student test scores and provide the following statistical analysis:

- **Calculation of the mean of the test scores.**
- **Calculation of the median score.**
- **Calculation of the mode score.**

Constraints

Input Requirements

Test Scores:

- **Must be provided as a list of integers (e.g., [70, 85, 90, 75, 80, 65, 95]).**
- **Cannot be an empty list.**
- **All values must represent valid test scores (e.g., no negative numbers).**

Methods and Constraints

Mean Score Calculation:

- **Use NumPy's `mean()` function to calculate the average of the test scores.**
- **Must return a single floating-point value.**

Median Score Calculation:

- **Use NumPy's `median()` function to calculate the middle value in the sorted test scores.**
- **Must return a single integer value.**

Mode Score Calculation:

- **Use SciPy's `stats.mode()` to calculate the most frequent score in the test data.**
- **Must return a single integer value representing the mode.**

Output Requirements

Required Output Format:

1. **The system should display the following for each analysis:**
   - **Mean Score: The average of all test scores.**
   - **Median Score: The middle score after sorting.**
   - **Mode Score: The most frequent score in the list.**

Template Code Structure:

1. **Initialization Functions:**
   - `__init__(self, scores)`
2. **Statistical Analysis Functions:**
   - `mean_score()`
   - `median_score()`
   - `mode_score()`
3. **Input Section:**
   - **Receive test scores as input (a list of integers).**
4. **Calculation Section:**
   - **Calculate mean, median, and mode for the provided scores.**
5. **Output Section:**
   - **Display the calculated mean, median, and mode in the required format.**

Execution Steps to Follow:

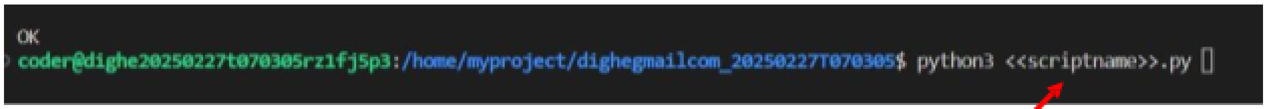
- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want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

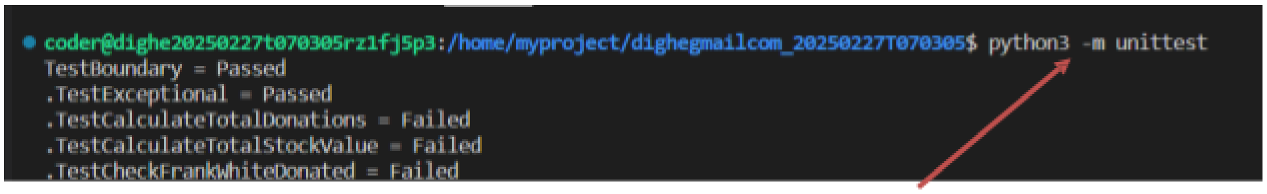
Screen shot to run the program

To run the application



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

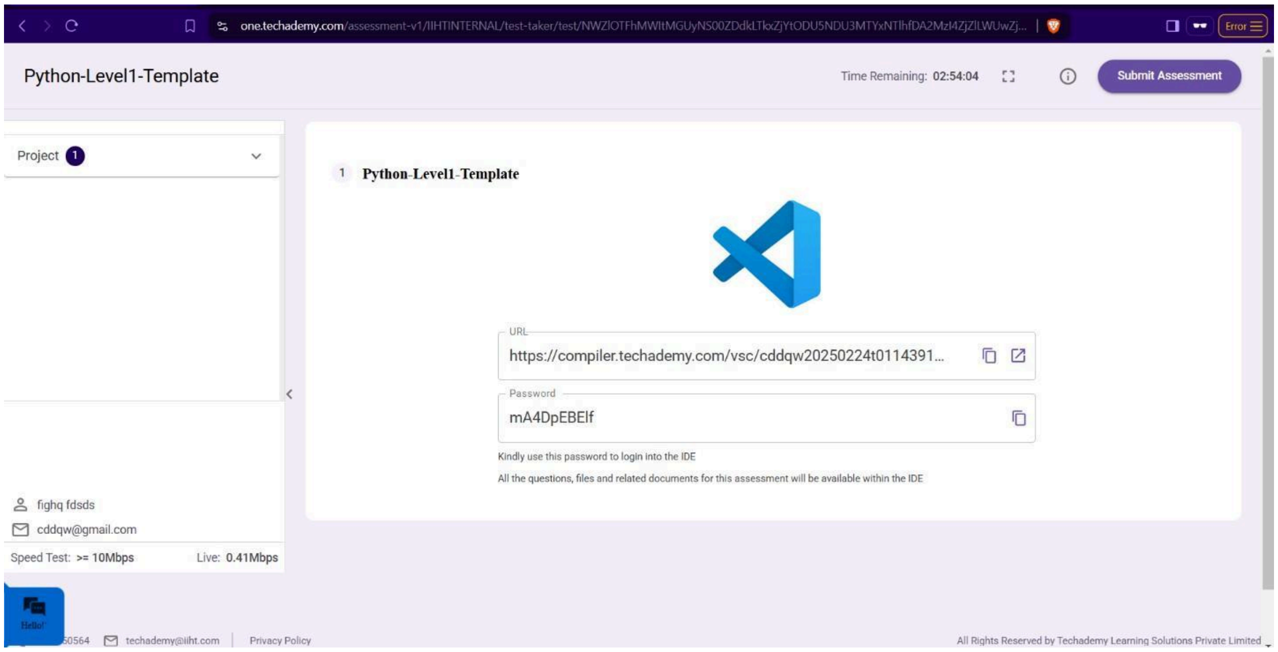
python3 mainclass.py python3



```
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 unittest



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

- **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.**