

# **System Requirements**

## **Specification Index**

**For**

### **NumPy Array Operations - Fancy Indexing and Boolean Indexing**

**(Topic: Advanced NumPy)**

**Version 1.0**

# Student Test Score Analysis Console

## Project Abstract

The Student Test Score Analysis system is a Python-based console application designed to analyze and process student test scores. This system takes in student IDs and their corresponding test scores, and provides various functionalities for analyzing the data, including fancy indexing, boolean indexing, and calculating statistical measures such as mean, median, and standard deviation of the scores. The application aims to provide an efficient and accurate way to assess student performance and generate relevant insights based on the data.

## Business Requirements:

1. **Student IDs:** A list of student IDs associated with their test scores.
  2. **Test Scores:** A list of test scores corresponding to each student.
  3. **Data Analysis Functions:**
    - **Fancy Indexing:** Select test scores at even indices.
    - **Boolean Indexing:** Select test scores greater than 50.
    - **Statistical Measures:** Calculate mean, median, and standard deviation of the scores.
  4. **Output:**
    - Display selected scores and calculated statistics for analysis.
  5. **Efficiency:**
    - Use NumPy arrays for efficient storage and manipulation of large datasets.
- 

## Constraints

### Input Requirements:

- **Student IDs:**
  - Must be stored as a list of integers.
  - Example: [1, 2, 3, 4]
- **Test Scores:**
  - Must be stored as a list of floats.
  - Must contain non-negative values.
  - Example: [90.5, 80.0, 55.0, 68.0]

### Data Analysis Functions:

1. Fancy Indexing:
    - Select elements at even indices in the test score array.
    - Example: For the array [90.5, 80.0, 55.0, 68.0], select scores at indices 0 and 2: [90.5, 55.0].
  2. Boolean Indexing:
    - Select elements from the test score array where the score is greater than 50.
    - Example: For the array [90.5, 80.0, 55.0, 68.0], return [90.5, 80.0, 55.0, 68.0].
  3. Statistical Measures:
    - Mean: Calculate the average score.
    - Median: Calculate the middle score when the scores are sorted.
    - Standard Deviation: Measure the spread of the scores.
- 

## Output Constraints

### 1. Display Format:

- Show selected test scores using fancy and boolean indexing.
- Show calculated statistics (mean, median, and standard deviation).

### Required Output Format:

- Fancy Indexing: Display the test scores at even indices. Example: "Selected Scores (Even Indices): [90.5, 55.0]"
  - Boolean Indexing: Display the test scores greater than 50. Example: "Selected Scores (Greater than 50): [90.5, 80.0, 55.0, 68.0]"
  - Statistical Data:
    - "Mean Score: {value}"
    - "Median Score: {value}"
    - "Standard Deviation: {value}"
- 

## Template Code Structure:

### 1. Class and Method Definitions:

1. TestScoreAnalysis Class:
  - Attributes: student\_ids (NumPy array of student IDs), test\_scores (NumPy array of test scores).
  - Methods:

- `fancy_indexing()`: Select elements at even indices.
  - `boolean_indexing()`: Select elements greater than 50.
  - `mean_score()`: Calculate the mean of the test scores.
  - `median_score()`: Calculate the median of the test scores.
  - `standard_deviation()`: Calculate the standard deviation of the test scores.
2. **Input Section:**
    - Input student IDs and test scores as lists.
  3. **Data Analysis Section:**
    - Call the methods for fancy indexing, boolean indexing, and calculating the statistical measures.
  4. **Output Section:**
    - Display the results for the selected scores and statistics.
- 

#### **Example Code Flow:**

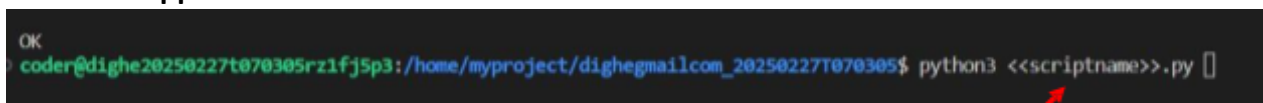
1. **Initialize Test Data:**
  - Input lists: `student_ids = [1, 2, 3, 4]` and `test_scores = [90.5, 80.0, 55.0, 68.0]`.
2. **Data Analysis:**
  - Apply fancy indexing to retrieve test scores at even indices.
  - Apply boolean indexing to retrieve test scores greater than 50.
  - Calculate mean, median, and standard deviation of the test scores.
3. **Display Output:**
  - Show the selected scores and calculated statistics.

### 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
```

A red arrow points from the bottom right towards the terminal window.

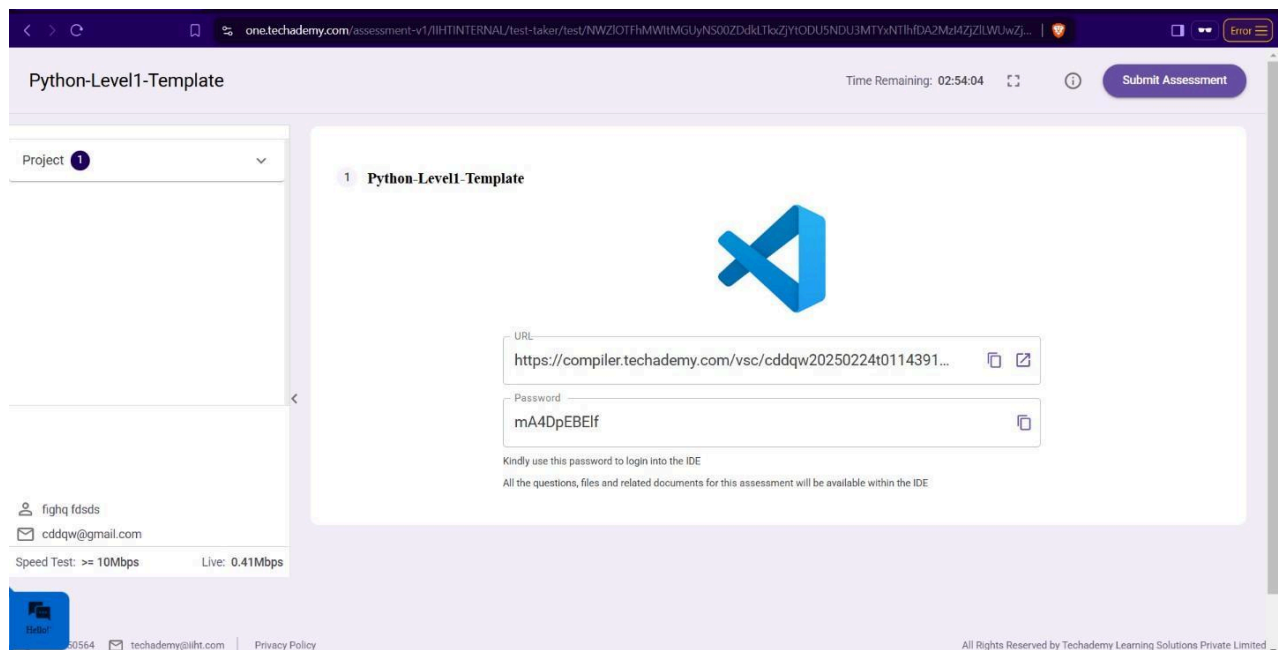
python3 mainclass.py python3

```
• 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 unittest7



- 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.