System Requirements Specification Index

For

Python Basics and NumPy, Pandas

Usecase 3

1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Use Case:1 Car Inventory Management (carinventory.py)

1) Write a Python program to search for cars within a given budget.
- Define a function search_by_budget(inventory, max_price).
- The function should:
    - Filter and display cars where the price is less than or equal to max_price.
    - If no cars match the criteria, print an appropriate message.
    - Return the filtered list of cars.

2) Write a Python program to save the car inventory into a JSON file.
- Define a function save_inventory(inventory, filename).
- The function should:
    - Convert the car inventory into JSON format.
    - Save it to a file named car_inventory.json.
    - Print a success message after saving.
    - Return the filename.

3) Write a Python program to execute all inventory operations.
- Implement a main() function that:
    - Calls display_cars(car_inventory) to show all available cars.
    - Calls search_by_budget(car_inventory, 25000) to find cars under $25,000.
    - Calls save_inventory(car_inventory) to store inventory data in a JSON file.

Use Case2: Student Management System (StudentCourseManagement.py)
1) Write a Python program to store and display a list of student names.
- Define a function student_names().
- The function should:
    - Store a list of students: "John", "Emma", "Sophia", "Michael", "Daniel".
    - Append "Olivia" to the list.
    - Return the updated student list.

2) Write a Python program to store student course enrollments using a dictionary.
- Define a function student_courses().
- The function should:
    - Store student names as keys and their enrolled courses as tuple values.
    - Add a new entry for "Olivia" with courses ("Biology", "History").
    - Return the updated dictionary.

3) Write a Python program to store and display unique subjects across all students.
- Define a function unique_subjects().
- The function should:
    - Use a set to store unique subjects across all student enrollments.
    - Ensure duplicate subjects (e.g., "Math") appear only once.
    - Add "Economics" as a new subject.
    - Return the updated set of unique subjects.

Use Case3: Student Marks Analysis (StudentMarksAnalysis.py)
1) Write a Python program to compute basic statistics for student marks.
- **Define a function analyze_marks(marks).**
- **The function should:**
    - **Compute the average, maximum, and minimum marks using NumPy.**
    - **Return these three statistics.**

2) Write a Python program to classify students based on their marks.
- **Define a function classify_grades(marks).**
- **The function should:**

- **Assign grades based on the following criteria:**
    - **A: mark >= 90**
    - **B: mark >= 80**
    - **C: mark >= 70**
    - **D: mark < 70**
- **Return the list of grades.**

3) Given a list of student marks, analyze and display the results.

- **Use NumPy to store student marks.**
- **Call analyze_marks() to get statistical insights.**
- **Call classify_grades() to determine student grades.**
- **Display:**
    - **The list of marks**
    - **Average marks (rounded to 2 decimal places)**
    - **Highest and lowest marks**
    - **Corresponding grades**

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will bethrough Command Terminal.

2. To open the command terminal the test takers, need to go to Application menu(Three horizontal lines at left top) -> Terminal -> New Terminal

3. This editor Auto Saves the code

4. 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.

5. 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 sametime it was stopped from the previous logout.

6. To setup environment:

You can run the application without importing any packages

7. To launch application:

python3 carinventory.py

python3 StudentMarksAnalysis.py

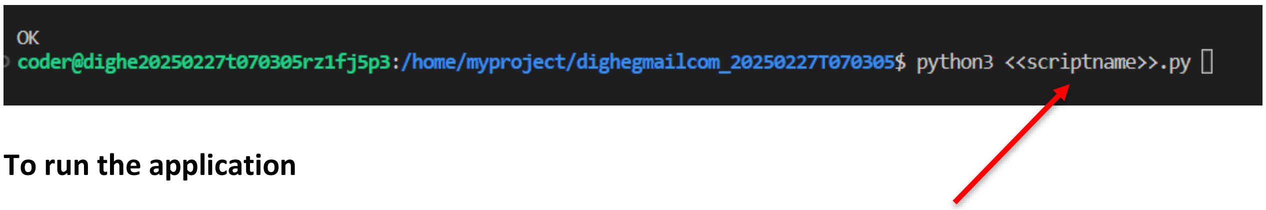
python3 StudentMarksAnalysis.py

To run Test cases:

python3 -m unittest

8. 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



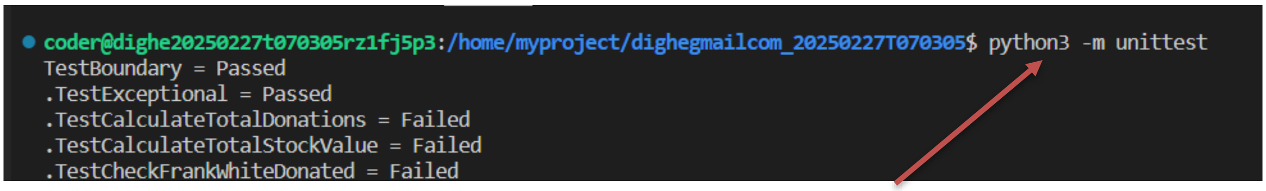
```
OK
coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 <<scriptname>>.py
```

To run the application

python3 carinventory.py

python3 StudentMarksAnalysis.py

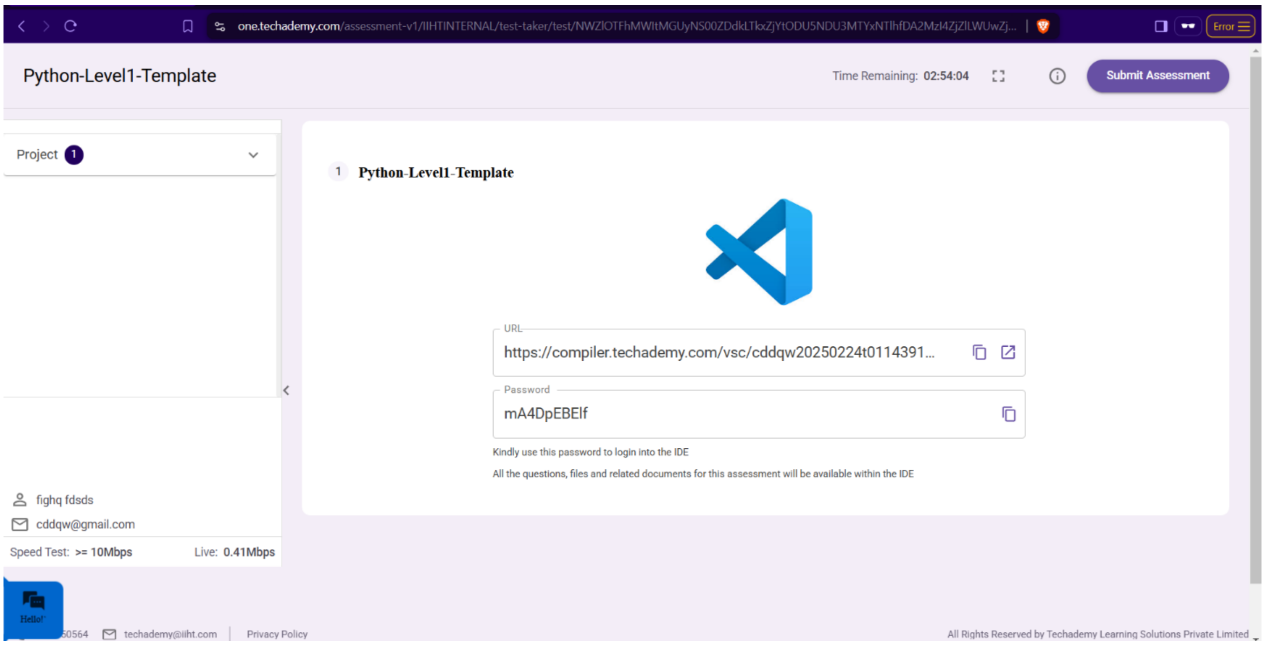
python3 StudentMarksAnalysis.py



```
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

9.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.

-----x-----