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 return the dataset of cars where the price is less than or equal to max_price.
  - If no cars match the criteria, return the empty dataset.

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.
  - Return the filename.

Use Case2: Student Management System (StudentCourseManagement.py)

1) Write a Python program to add a new student in student list.

- Define a function student_names().
- The function should:
  - Template code already has a list of students.
  - Append "Olivia" to the list.
  - Return the updated student list.

2) Write a Python program to store student course enrolments using a dictionary.

- Define a function student_courses().
- The function should:
  - Template code already has an existing dictionary where 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:
  - Template code already has a list of subjects (some are duplicate)
  - Add "Economics" as a new subject.
  - Remove all the duplicate subjects
  - Return the updated list 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. (rounded to 2 decimal places)
  - Return these three statistics in following order as tuple : avg, max, min

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

  Put these grades in a list in same orders as the marks
- Return the list of grades.

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through 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 same time 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

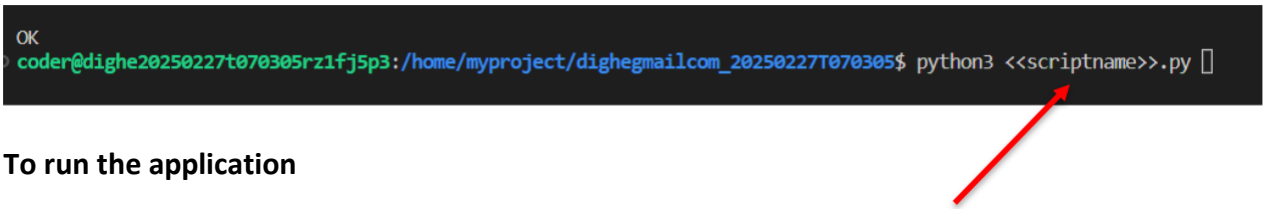
python3 StudentMarksAnalysis.py

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

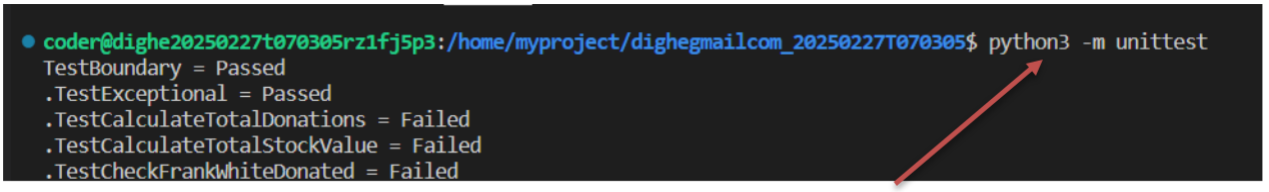


To run the application

python3 carinventory.py

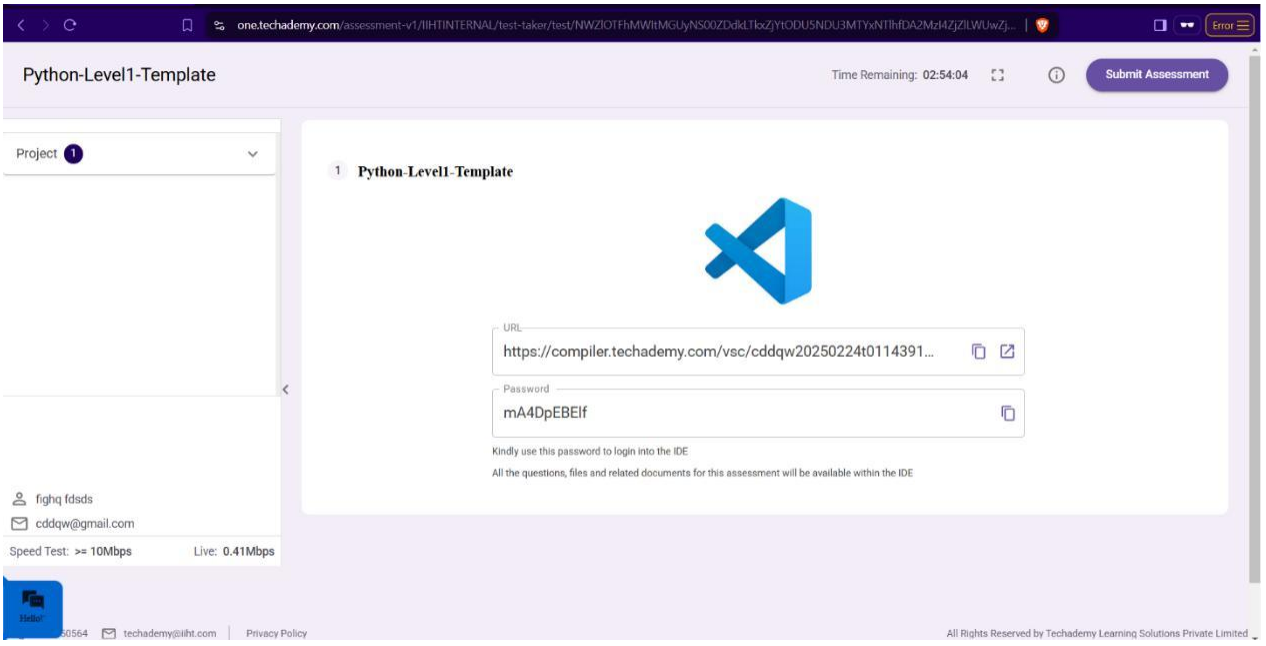
python3 StudentMarksAnalysis.py

python3 StudentMarksAnalysis.py



To run the testcase

- **python3 -m unittest**



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-----