System Requirements Specification Index

For

Python Basics and NumPy, Pandas

Usecase No 5

1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Use Case:1 Car Inventory Management (car.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 Case: Smart Waste Management System (wastetreatment.py)

Dataset {

("ZoneA","Organic",120),
("ZoneB","Plastic",80),
("ZoneC","Electronic",45),
("ZoneD","Metal",60),
("ZoneE","Organic",95)
}

1. **Write a Python function to calculate total waste by type.**
   - Define **calculate_total_waste_by_type(data)**.
   - The function should:
     - Iterate through the waste data list.
     - Calculate the **total weight** for each waste type.
     - Return a dictionary with **waste type** as the key and **total weight** as the value.

2. **Write a Python function to identify unique waste zones.**
   - Define **unique_waste_zones(data)**.
   - The function should:
     - Extract **unique locations** from the dataset.
     - Return a **set** of unique zones.

3. **Write a Python function to find the location with the heaviest waste.**
   - Define **find_heaviest_location(data)**.
   - The function should:

- Identify the **zone with the highest waste weight**.
- Return the **zone name and its corresponding weight**.

Use Case: Railway Reservation System (trainseatallocation.py)
Functions Required

Dataset

```
seat_numbers = np.array([101, 102, 103, 104, 105])
ticket_prices = np.array([250, 300, 400, 350, 500])
availability = np.array([True, False, True, False, True])
passenger_names = ["John", "Alice", "Bob", "Emma", "David"]
```

 # Availability status (True means available)

1 **.Find the highest ticket price.**
   - Define **highest_ticket_price(df)**
   - The function should:
     - Extract the **maximum ticket price**.
     - Identify the **seat number** associated with the highest price.
     - Return both **highest price and corresponding seat number**.

2 **Find the number of available seats.**
   - Define **available_seats(df)**
   - The function should:
     - Count the **number of available seats** (Availability == True).
     - Return the count of **available seats**.

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 sametime it was stopped from the previous logout.
- To setup environment:
  You can run the application without importing any packages
- To launch application:
  python3 car.py
  python3 trainseatallocation.py.py
  python3 wastetreatment.py

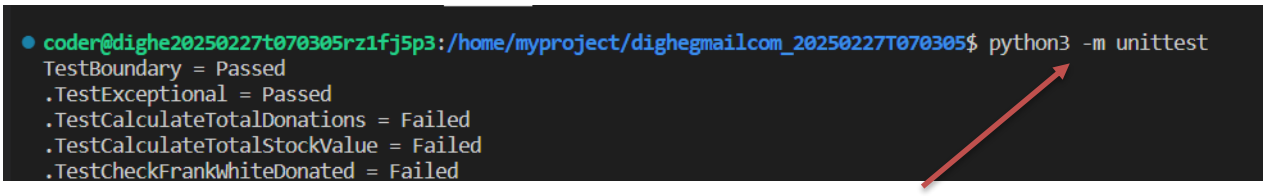
- To run Test cases:
  python3 -m unittest

- 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

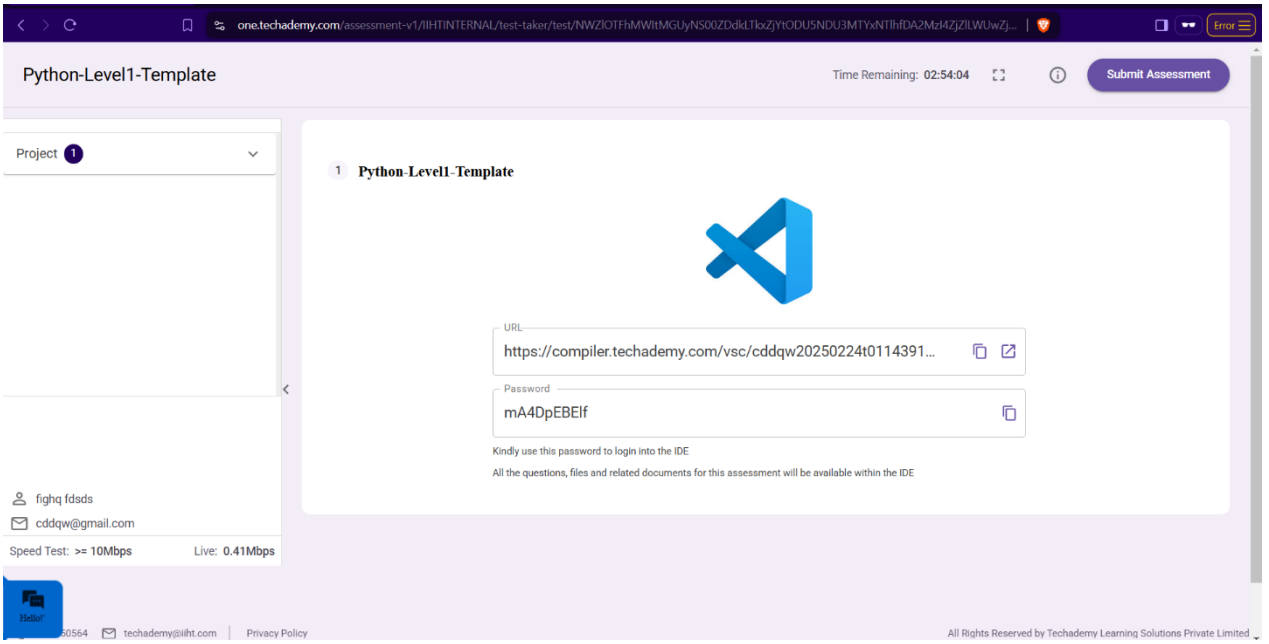
To run the application

python3 car.py
python3 trainseatallocation.py.py
python3 wastetreatment.py



To run the testcase

python3 -m unittest



- **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-----