System RequirementsSpecification Index

For

Python Basics and NumPy, Pandas
Usecase No 7
1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Use Case1: Event Management System (event.py)

Dataset

```
events = [
   ("Tech Conference", "2025-03-15", "2025-03-17", "New York"),
   ("Music Festival", "2025-04-20", "2025-04-22", "Los Angeles"),
   ("Art Exhibition", "2025-05-10", "2025-05-15", "Paris"),
   ("Startup Meetup", "2025-06-05", "2025-06-06", "London"),
   ("AI Summit", "2025-07-18", "2025-07-20", "Tokyo")
]
```

1.)Write a Python function to display upcoming events.
Define display_upcoming_events()
The function should:

- Iterate through the list of events.
- Extract event details including name, start date, end date, and location.
- Print all upcoming events in a formatted manner.

2.)Write a Python function to count the total number of events.
Define count_total_events()
The function should:

- Count the number of events in the dataset.
- Print the total count of events.

3.)Write a Python function to find the event with the longest duration.
Define longest_event_duration()
The function should:

- Calculate the duration of each event (difference between start and end dates).
- Identify the event with the maximum duration.
- Print the event name along with its duration in days.

Use Case2: Gym Membership Management System (gym.py)

Dataset

```
membership_data = """John, Gold, 2024-01-15, 2025-01-15, Paid
Alice, Silver, 2024-02-01, 2025-02-01, Unpaid
Bob, Platinum, 2024-03-10, 2025-03-10, Paid
Emma, Gold, 2024-04-20, 2025-04-20, Paid
Mike, Silver, 2024-05-05, 2025-05-05, Unpaid"""
```

Note: you need to create membership_data.txt

1)Write a Python function to create a analyse membership file.
Define check_membership_status():
The function should:

- Create a text file named "membership_data.txt".
- Complete the create_membership_file():
- Populate it with sample membership details (name, plan type, start date, end date, and payment status).
- Store active members in a dictionary with details (Plan, End Date).
- Extract and filter members with a "Paid" status.

- Display the count list of active members

2) Write a Python function to def count_members_by_plan():

- Read from the membership_data.txt
- Count the number member by each plan (gold ,sliver, Platinum)
- Return the value

Usecase 3: Hotel Room & Booking Management System (hotel.py)

Dataset

def get_room_data()

```
{
  "R101": {"type": "Standard", "price": 100.00, "capacity": 2, "amenities": ["TV", "WiFi"]},
  "R102": {"type": "Deluxe", "price": 150.00, "capacity": 2, "amenities": ["TV", "WiFi", "Mini Bar"]},
  "R103": {"type": "Suite", "price": 250.00, "capacity": 4, "amenities": ["TV", "WiFi", "Mini Bar", "Jacuzzi"]},
  "R104": {"type": "Standard", "price": 100.00, "capacity": 2, "amenities": ["TV", "WiFi"]},
  "R105": {"type": "Deluxe", "price": 150.00, "capacity": 2, "amenities": ["TV", "WiFi", "Mini Bar"]}
}
```

get_booking_data():

```
[
  {"booking_id": "B001", "room_id": "R102", "guest_name": "John Smith", "check_in": today - timedelta(days=2), "check_out": today + timedelta(days=3), "guests": 2},
  {"booking_id": "B002", "room_id": "R105", "guest_name": "Alice Johnson", "check_in": today + timedelta(days=5), "check_out": today + timedelta(days=10), "guests": 2},
  {"booking_id": "B003", "room_id": "R103", "guest_name": "Bob Williams", "check_in": today + timedelta(days=15), "check_out": today + timedelta(days=20), "guests": 3},
  {"booking_id": "B004", "room_id": "R101", "guest_name": "Emma Davis", "check_in": today + timedelta(days=7), "check_out": today + timedelta(days=9), "guests": 1},
  {"booking_id": "B005", "room_id": "R104", "guest_name": "Michael Brown", "check_in": today + timedelta(days=12), "check_out": today + timedelta(days=14), "guests": 2}
]
```

Feature 1: Get the Price of the Room with a Jacuzzi
Function: jacuzzi_room_price()

- Retrieves room data.
- Checks if any room has a "Jacuzzi" in its amenities.
- Returns the room ID and price if found; otherwise, returns None.

Feature 2: Find the Most Expensive Room
Function: most_expensive_room()

- Retrieves room data.

- Finds the room with the highest price.
- Returns the room ID and price.

Return Functions

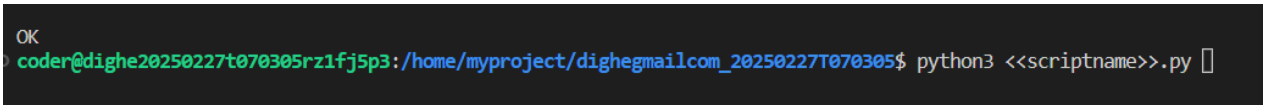
1. get_room_data() – Returns room details (type, price, capacity, amenities).
2. get_booking_data() – Returns sample booking records with check-in/check-out dates.

Execution Steps to Follow:

- All actions like build, compile, running application, running test cases will be through the 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 hotel.py
  python3 Event.py
  python3 Gym.py
- To run Test cases:
  python3 -m unittest

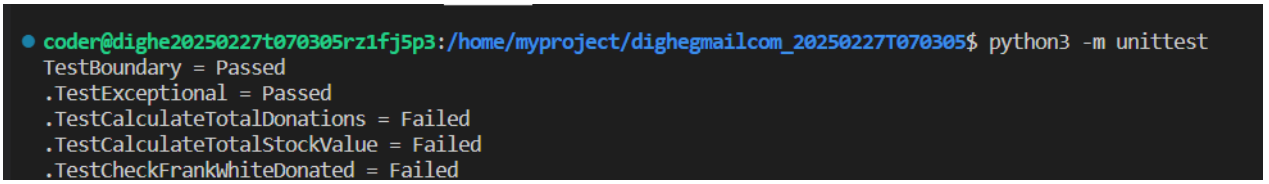
- Before Final Submission also, you need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

Screen shot to run the program



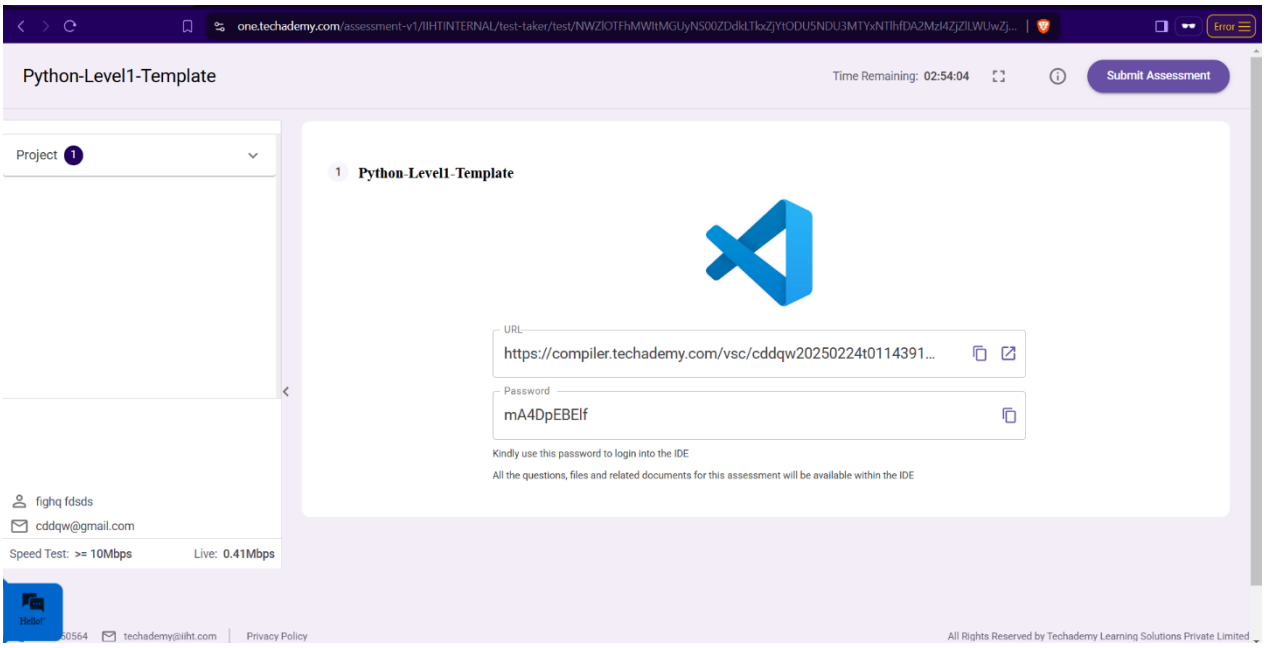
To run the application

python3 hotel.py
python3 Event.py
python3 Gym.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.