System Requirements Specification Index

For

Python Basics and NumPy, Pandas

Usecase No 4

1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Use Case: Blood Bank Management (BloodBankManagementSystem.py)

The dataset to be used in the format is

blood_groups=np.array(["A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"])
units_available=np.array ([10, 5, 8, 4, 15, 7, 6, 3])

1. **Write a Python program to manage blood bank inventory.**
    - Define a function **get_blood_bank_data()**.

2. **Write a Python program to add a new blood group to the inventory.**
    - Define a function **add_new_blood_group(blood_bank, blood_group, units)**.
    - The function should:
        - Accept a **DataFrame** (blood bank inventory), a **new blood group**, and its **unit count**.
        - Append the new blood group entry to the DataFrame.
        - Return the updated inventory.
    - Add a **new blood group (P+ with 10 units)** using **add_new_blood_group()**.

3. **Write a Python program to calculate the total number of blood units available.**
    - Define a function **get_total_units(blood_bank)**.
    - The function should:
        - Compute and return the **sum of all blood units** in the inventory.

Use Case: Food Delivery Management (OnlineFoodDeliverySystem.py)

1. **Write a Python program to manage a food menu and prices.**

Dataset provides are {
"Pizza": 8.50,
"Burger": 5.00,
"Pasta": 7.25,
"Fries": 3.50,
"Cola": 2.00
}

    - Define a function get_menu().
    - The function should:
        - Return a **dictionary** containing food items as **keys** and their prices as **values**.

2. **Write a Python program to calculate the total bill for a food order.**
    - Define a function **calculate_bill(orders, menu)**.
    - The function should:
        - Accept a **list of tuples** containing food items and quantities.
        - Multiply each item's **price by its quantity** to compute the cost.
        - Compute the **total bill**.
        - Return both the **total bill** and an **order summary**.

3. **Write a Python program to save order details to a file.**
    - Define a function **save_order(order_summary, total_bill, filename="food_orders.txt")**.
    - The function should:
        - Write the **order summary** and **total bill** to a text file.
        - Append new orders without overwriting existing data.
        - Return the filename after successful save.

Use Case 3: Employee Leave Management (EmployeeLeaveManagementSystem.py)

Dataset to be used should be {

"E001": {"name": "John Doe", "leave_balance": 12},
"E002": {"name": "Alice Smith", "leave_balance": 10},
"E003": {"name": "Bob Johnson", "leave_balance": 8},
"E004": {"name": "Emma Davis", "leave_balance": 15},
"E005": {"name": "Michael Brown", "leave_balance": 5},

Leave request { ("E001", 3), -> leave request count
("E003", 2),
("E005", 4),
("E002", 1),
("E004", 5),
("E999", 3),
("E003", -2) }

1. **Write a Python program to manage employee leave balances.**
    - Define a function **get_employee_data()**.
    - The function should:
        - Return a **dictionary** containing employee IDs as **keys**.
        - Each employee ID maps to a dictionary with **name** and **leave balance**.

2. **Write a Python program to process leave requests.**
    - Define a function **process_leave_requests(employees, leave_requests)**.
    - The function should:
        - Accept a **dictionary** of employees and a **list of leave requests** (employee ID, leave days).
        - Validate if the **employee ID exists**.
        - Reject **negative leave requests**.
        - Approve leave if the **balance is sufficient** and update the leave balance.
        - Deny leave if the **balance is insufficient**.
        - Return a **list of messages** summarizing the leave request outcomes.

3. **Process and display the leave request summary.**
    - Display the **summary of leave approvals and rejections**.

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 & Exitoption 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 whilelogging 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 BloodBankManagementSystem.py

Python3 OnlineFoodDeliverySystem.py

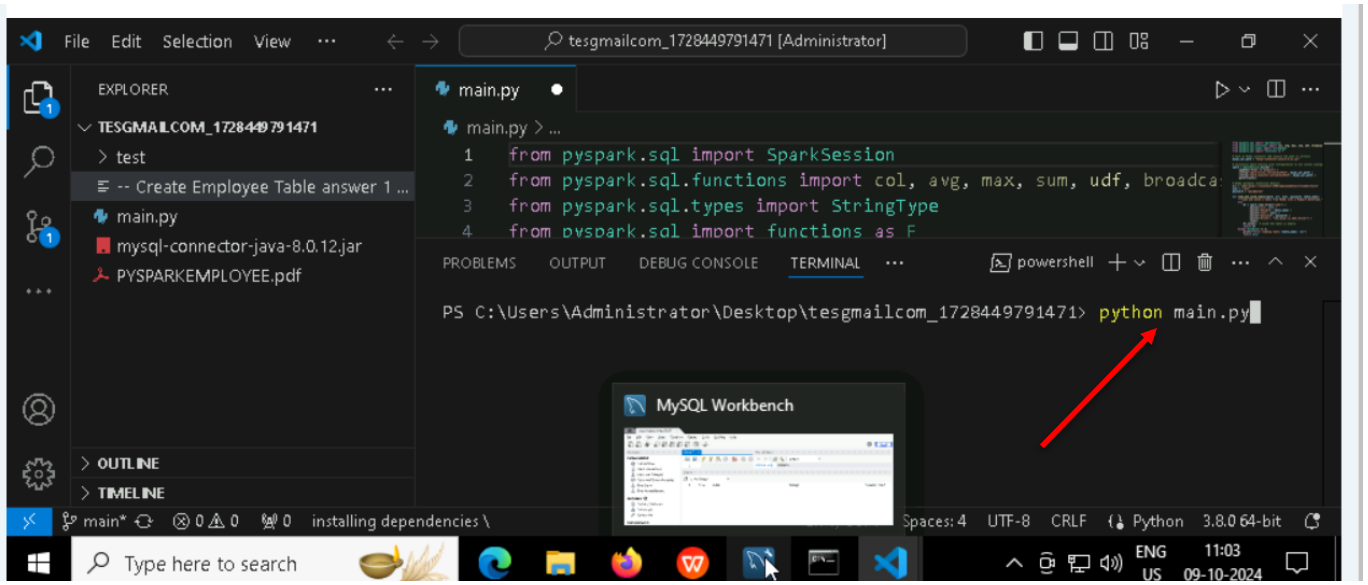
Python3 EmployeeLeaveManagementSystem.py

To run Test cases:

Python3 -m unittest

Before Final Submission also, you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internalgit/repository for code

Screen shot to run the program

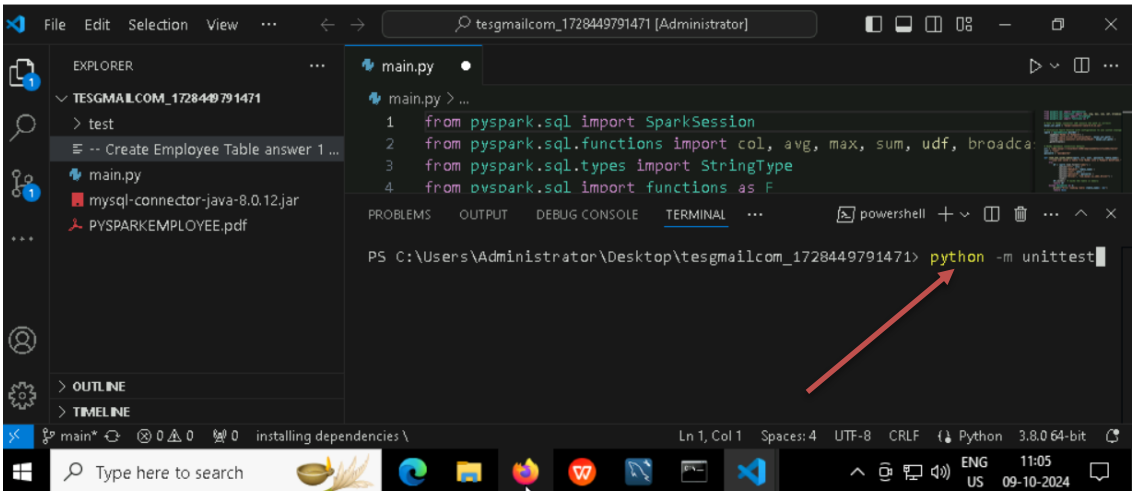


To run the application

Python3 BloodBankManagementSystem.py

Python3 OnlineFoodDeliverySystem.py

Python3 EmployeeLeaveManagementSystem.py



To run the testcase

- **Python3 -m unittest**

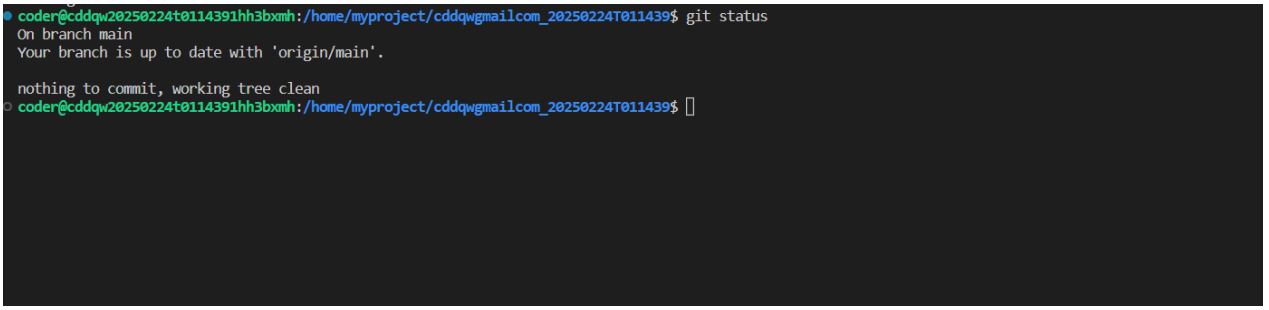
Screenshot to push the application to github

-----x-----

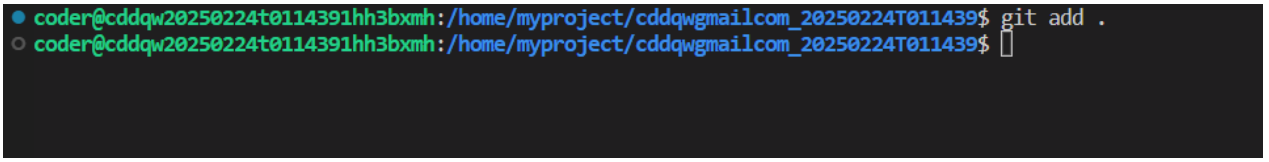
You can run test cases as many numbers of times and at any stage of Development, to check howmany test cases are passed/failed and accordingly refactor your code.

1. Make sure before final submission you commit all changes to git. For that

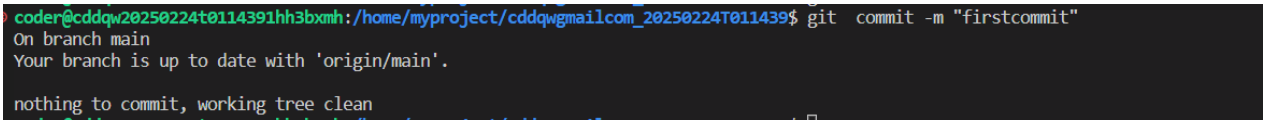
In the terminal use the command **git status**



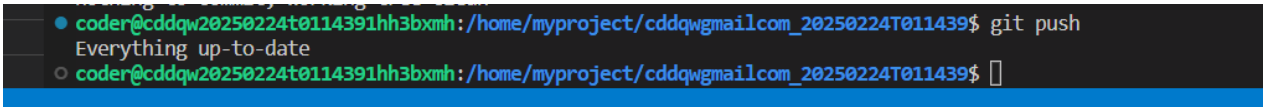
git add .



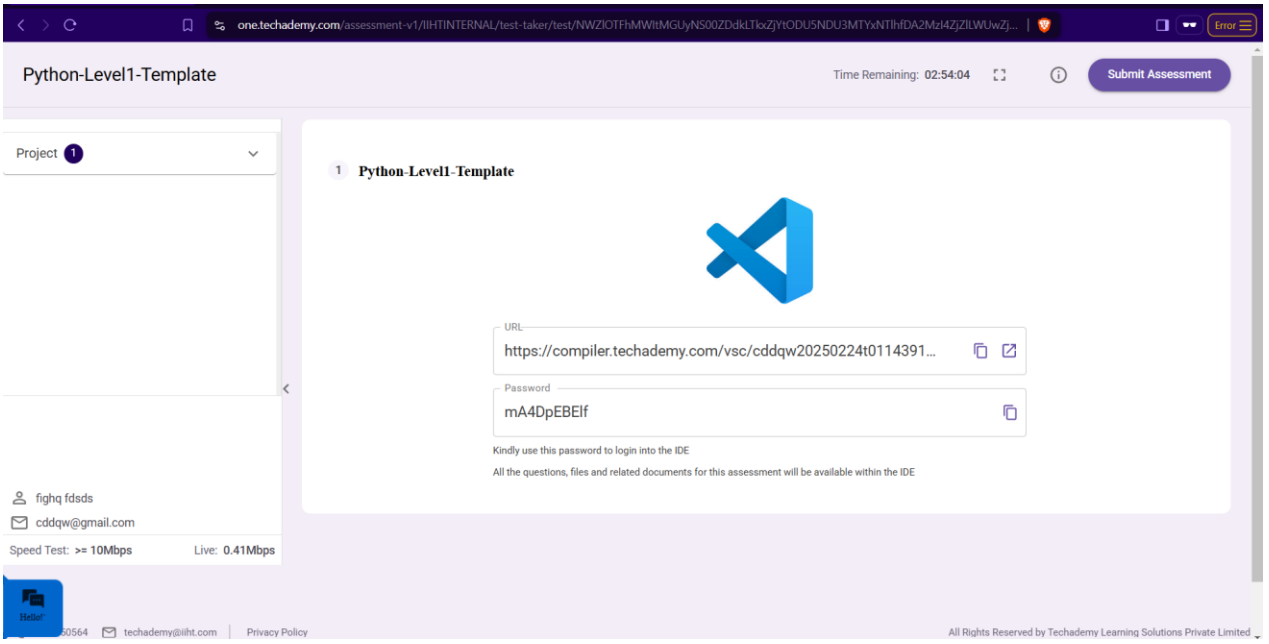
git commit -m “First commit”
(You can provide any message every time you commit)



git push



After you have pushed your code Finally click on the final submission button



Click on the submit assessment buttonafter you have pushed the code

-----x-----