
System Requirements Specification Index

For

Python Basics and NumPy, Pandas

Usecase 2

1.0

Use Case1: Inventory Management (inventory_management.py)

The dataset provided to you

Dataset: Inventory (Item, Quantity) and Prices

```
inventory_data = [  
    ("Laptop", 5),  
    ("Mouse", 20),  
    ("Keyboard", 12),  
    ("Monitor", 3),  
    ("USB Drive", 50)  
]
```

```
item_prices = {  
    "Laptop": 1000,  
    "Mouse": 25,  
    "Keyboard": 80,  
    "Monitor": 300,  
    "USB Drive": 15  
}
```

1) Write a Python program to check and return low-stock items.

- Define a function `get_low_stock_items(inventory, reorder_level)`.
- The function should:
 - Loop through the inventory dataset.
 - Identify items with stock below the reorder level.
 - Return a list of such low-stock items.

2) Write a Python program to calculate the total inventory value.

- Define a function `calculate_inventory_value(inventory, prices)`.
- The function should:
 - Multiply each item's stock by its price.
 - Sum up the total inventory value.
 - Return the total inventory value.

Use Case:2 Weather Analysis (weather_analysis.py)

Dataset temperatures = {32.5, 34.0, 31.2, 29.8, 35.5}

1) Write a Python program to analyze weather data.

- Define a function `weather_analysis(temperatures)`.
- The function should:
 - Identify the highest and lowest temperatures.
 - Detect extreme temperatures (above 34°C or below 30°C).
 - Generate and return a formatted weather analysis report.

2) Write a Python program to save the weather analysis report to a file.

- Define a function `save_report_to_file(report, filename)`.
- The function should: `weather_analysis()`
 - Append the generated weather report to a text file.
 - Display a confirmation message when the report is saved.
 - Use a sample dataset of temperature readings in Celsius.

- Detect the Highest Temperature and low temperature
- Save the generated report using `save_report_to_file()`

Use Case3: Customer Purchase Analysis (customer_purchase_analysis.py)

Customer purchase dataset provided

```
data = {
    'Customer': ['Alice', 'Bob', 'Charlie', 'Diana', 'Eve'],
    'Items_Bought': [3, 7, 5, 2, 9],
    'Amount_Spent': [150.0, 400.0, 275.0, 100.0, 600.0]
}
```

1) Write a Python program to create and display a customer purchase dataset using Pandas.

- Define a function `create_dataframe()`.
- The function should:
 - Create a Pandas DataFrame containing customer purchase details.
 - Include the following columns: Customer, Items_Bought, Amount_Spent.
 - Return the DataFrame.

2) Write a Python program to calculate the average amount spent by customers.

- Define a function `calculate_average_spending(df)`.
- The function should:
 - Compute the average amount spent across all customers.
 - Return the calculated average.

3) Write a Python program to identify high-spending customers.

- Define a function `get_top_spenders(df, average_spending)`.
- The function should:
 - Filter and return a DataFrame of customers whose spending is above the average.

4) Write a Python program to analyze and display the customer purchase data.

- Implement a `main()` function that:
 - Calls `create_dataframe()` to generate the dataset.
 - Calls `calculate_average_spending(df)` to compute the average spending.
 - Calls `get_top_spenders(df, average_spending)` to identify high-spending customers.
 - Prints the customer purchase dataset and high-spending customers.

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 customer_purchase_analysis.py

Python3 inventory_management.py.py

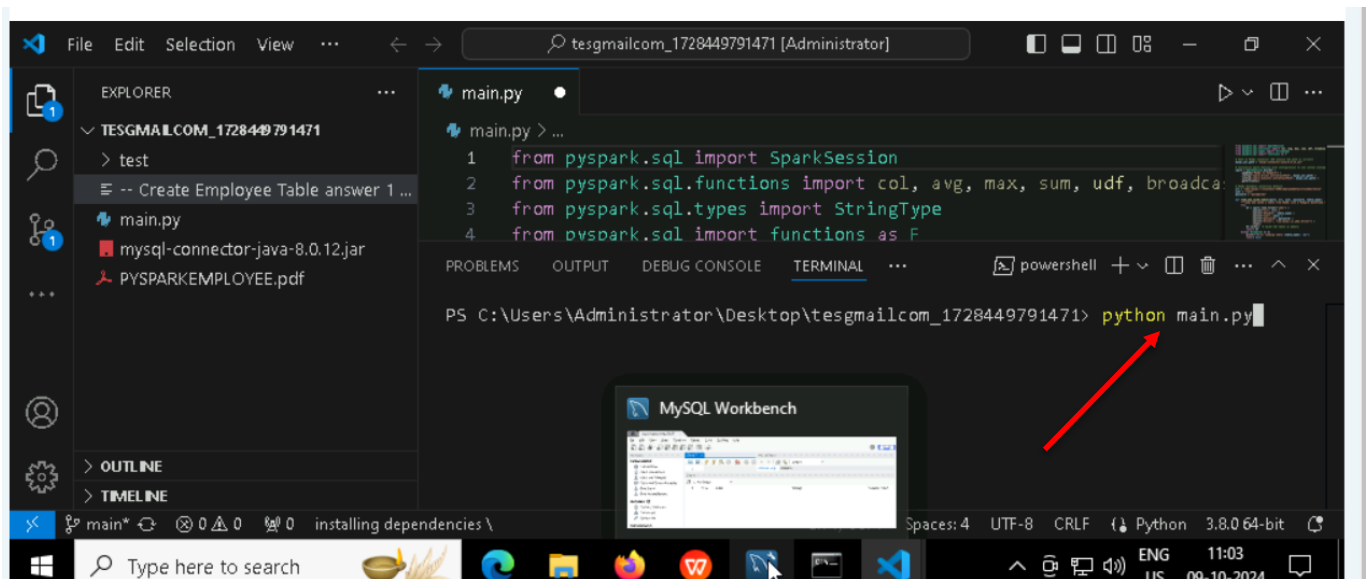
Python3 weather_analysis.py

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 internal git/repository for code

Screen shot to run the program

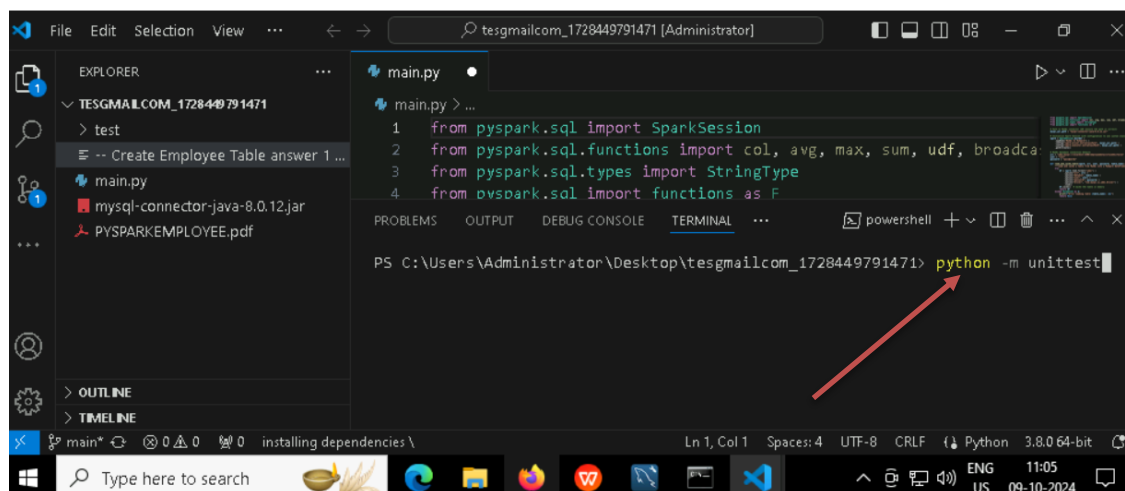


To run the application

Python3 customer_purchase_analysis.py

Python3 inventory_management.py.py

Python3 weather_analysis.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 A.**git status**

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

B **.git add .**

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git add .
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

C **.git commit -m "First commit"**

(You can provide any message every time you commit)

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git commit -m "firstcommit"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

D **.git push**

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git push
Everything up-to-date
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

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


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

0564 techademy@iit.com Privacy Policy

All Rights Reserved by Techademy Learning Solutions Private Limited

Click on the submit assessment button after you have pushed the code

-----X-----