LAB-8.3

-2303A51197

Task 1: Email Validation using TDD

Scenario: You are developing a user registration system that requires reliable email input validation.

Prompt used:

#create a python function for validating email in a registration system

#email must contain one @ symbol and at least one .

#email should not begin or end with special characters

#multiple @ symbols should not be allowed.

#return true if the is valid emil, otherwise return false.

//OUTOUT:

C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig\lab_8.3.py

```
#email must contain one @ symbol and at least one .
#email should not begin or end with special characters
#multiple @ symbols should not be allowed.
#return true if the is valid emil, otherwise return false.
#don't give comments in the code.
def validate_email(email):
    if email.count('@') != 1:
        return False
    if email.startswith('@') or email.endswith('@'):
        return False
    if email.startswith('.') or email.endswith('.'):
        return False
    if '.' not in email.split('@')[1]:
        return False
    return True
#display output
email = input("Enter your email: ")
if validate_email(email):
    print("Valid email")
else:
    print("Invalid email")
```

```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/U
Enter your email: prakashkommarajul@gmail.com
Valid email
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig> & C:\Users\PRAKASH\AppData\Local\Microsoft\WindowsApps\python3.11.exe "c:/U
Enter your email: njcsbhjdbs
Invalid email
PS C:\Users\PRAKASH\OneDrive\Desktop\Ai assisted codig>
```

Task 2: Grade Assignment using Loops

Scenario: You are building an automated grading system for an online examination platform.

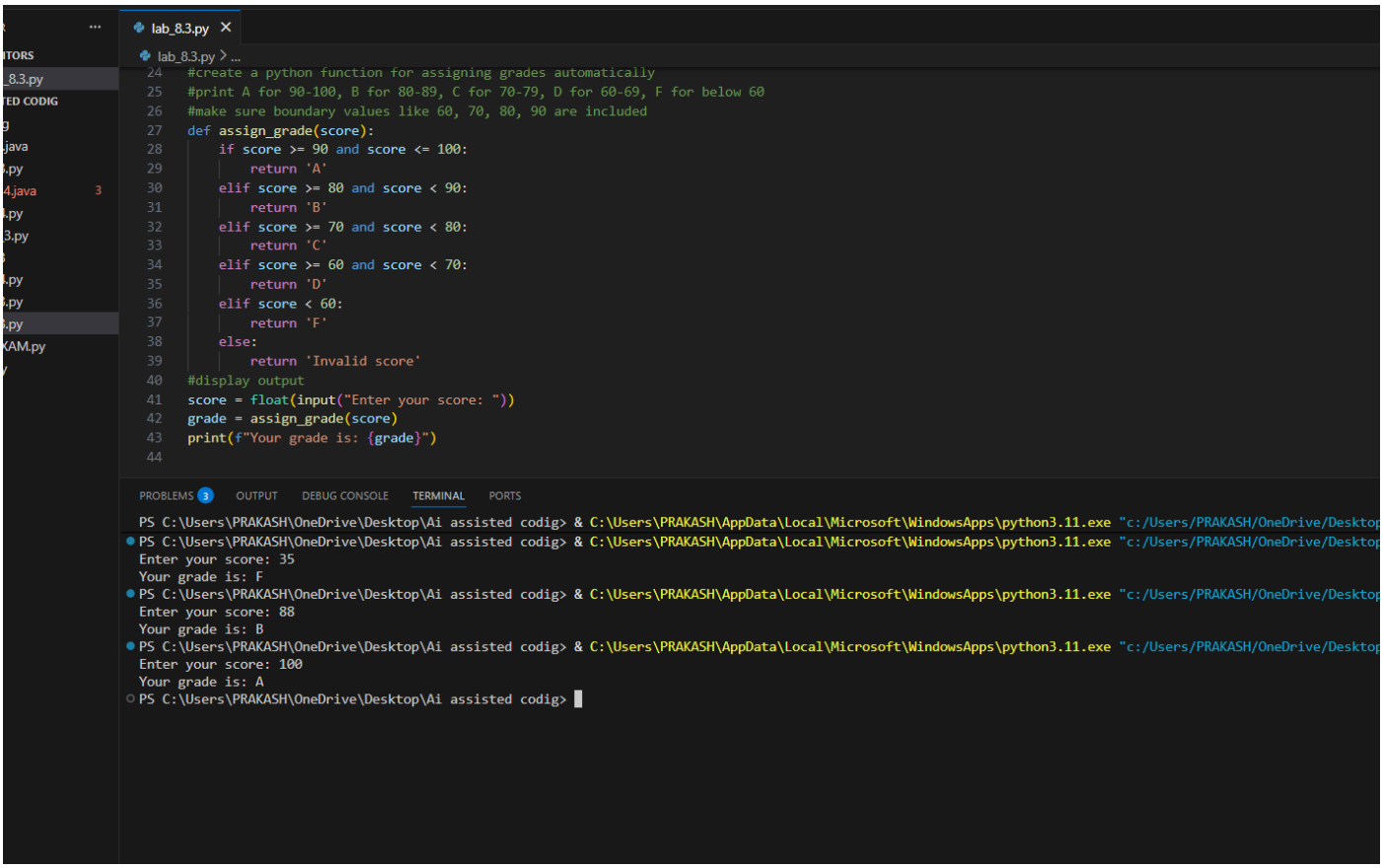
Prompt used:

#create a python function for assigning grades automatically

#print A for 90-100, B for 80-89, C for 70-79, D for 60-69, F for below 60

#make sure boundary values like 60, 70, 80, 90 are included

//OUTOUT:



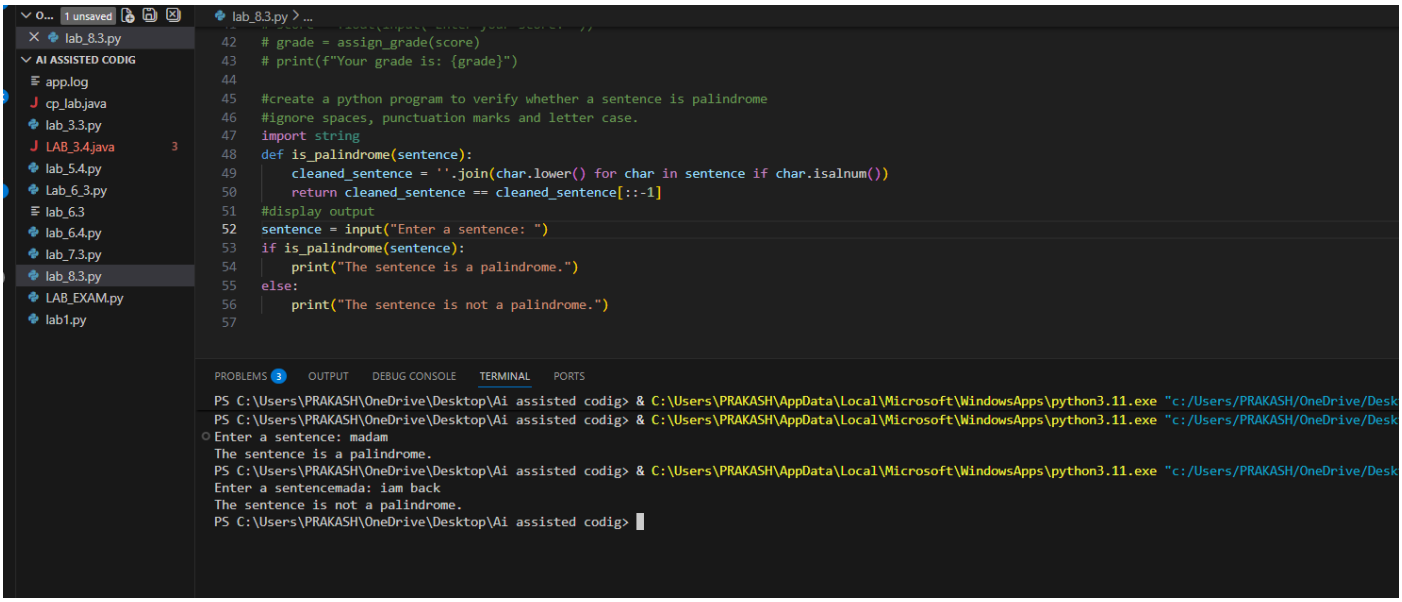
Task 3: Sentence Palindrome Checker

Scenario: You are developing a text-processing utility to analyze sentences.

Prompt used:

#create a python program to verify whether a sentence is palindrome

#ignore spaces, punctuation marks and letter case.



Task 4: ShoppingCart Class

Scenario: You are designing a basic shopping cart module for an e-commerce application.

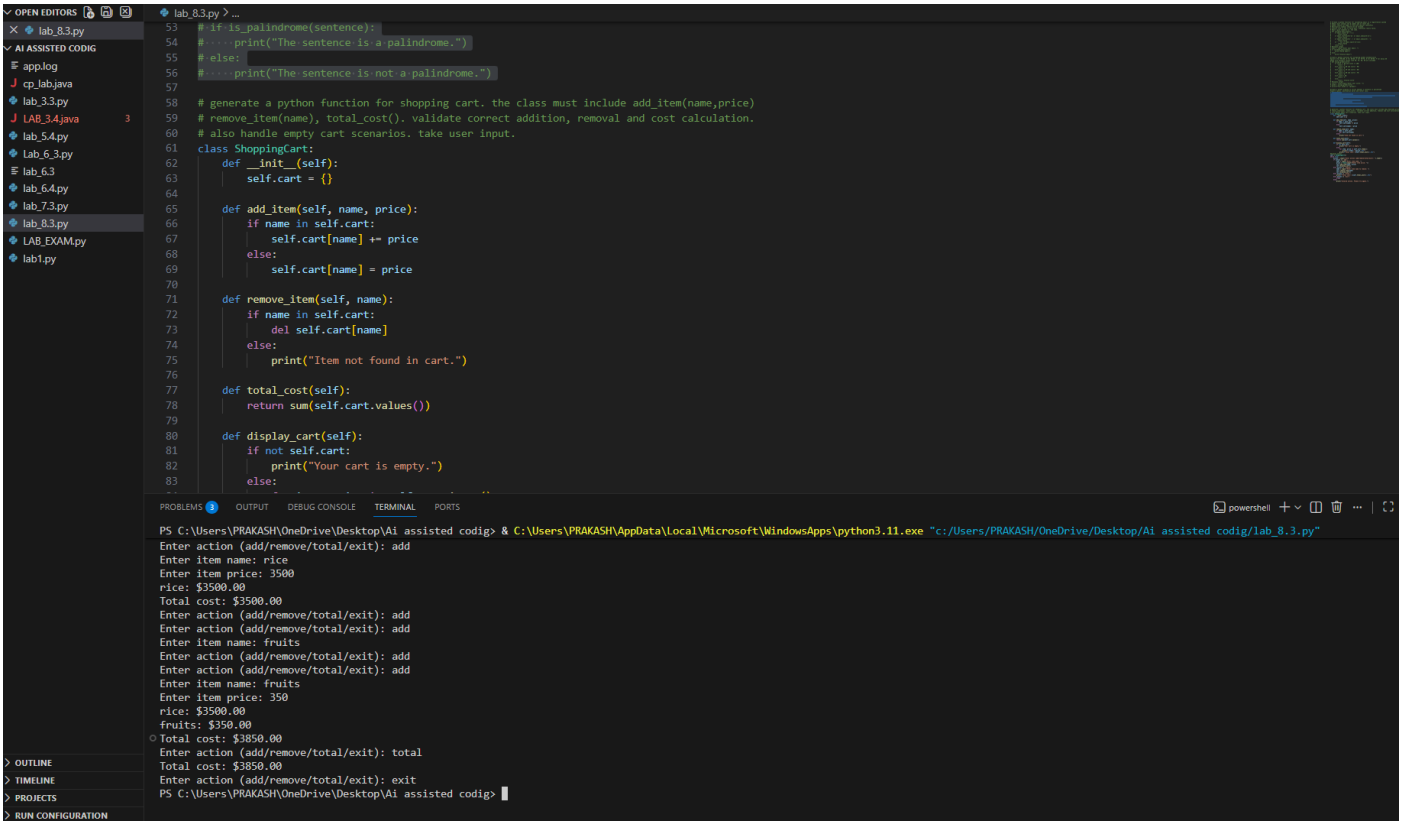
Prompt used:

 # generate a python function for shopping cart. the class must include add_item(name,price)

 # remove_item(name), total_cost(). validate correct addition, removal and cost calculation.

 # also handle empty cart scenarios. take user input

//OUTOUT:



Task 5: Date Format Conversion

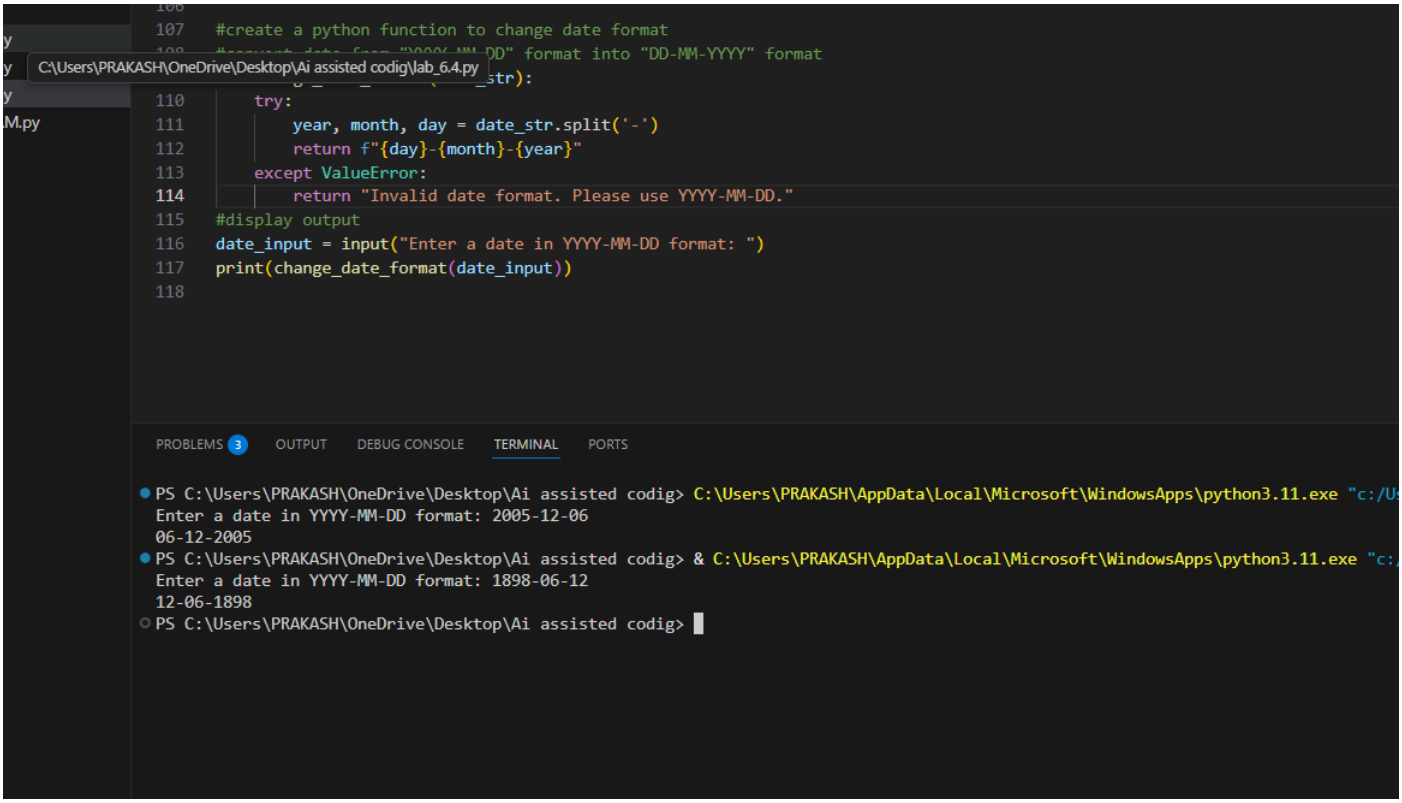
Scenario: You are creating a utility function to convert date formats for reports.

Prompt used:

#create a python function to change date format

#convert date from "YYYY-MM-DD" format into "DD-MM-YYYY" format

//OUTOUT:



Explanation:

1. The first snippet is a function to validate email addresses based on specific criteria.

2. The second snippet is a function to calculate letter grades based on numerical scores.

3. The third snippet checks if a given sentence is a palindrome, ignoring spaces and punctuation.

4. The fourth snippet defines a ShoppingCart class that allows users to add and remove items, calculate total cost, and display the cart contents.

5. The fifth snippet is a function that converts a date from "YYYY-MM-DD" format to "DD-MM-YYYY" format.