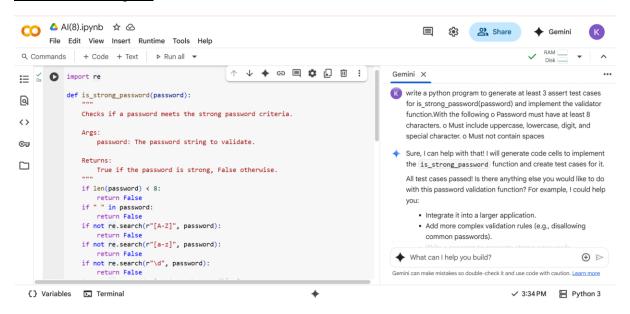
Assignment-8.1

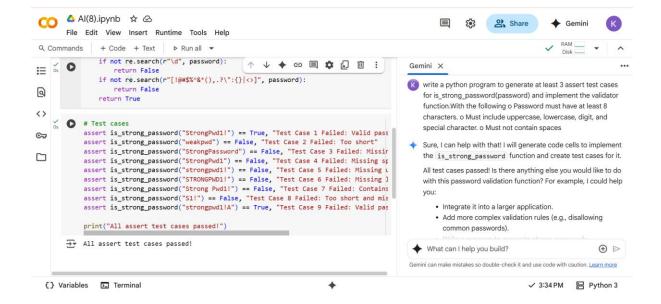
2403A52118

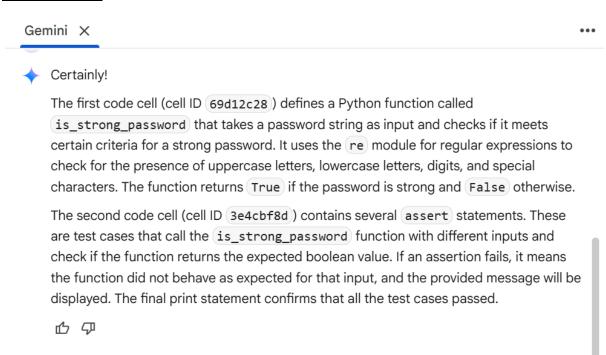
Task 1:

Write a python program to generate at least 3 assert test cases for is_strong_password(password) and implement the validator function. With the following

- >Password must have at least 8 characters.
- >Must include uppercase, lowercase, digit, and special character.
- >Must not contain spaces.







⊕ (+)

Gemini can make mistakes so double-check it and use code with caution. Learn more

Task 2:

Write a python program to generate at least 3 assert test cases for a classify_number(n) function. Implement using loops.with the following conditions

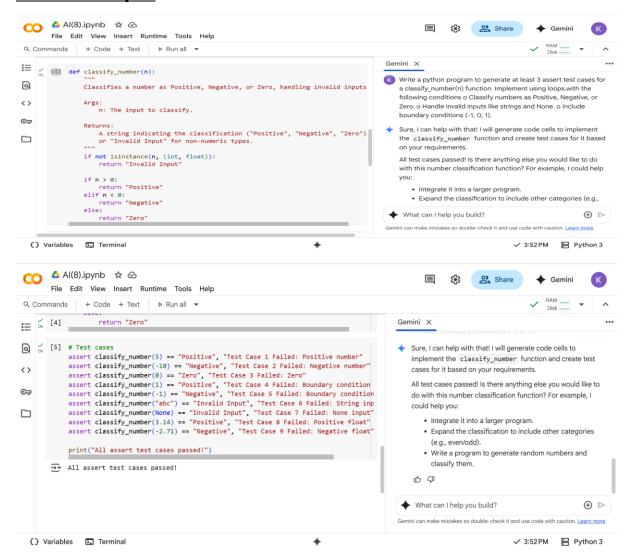
>Classify numbers as Positive, Negative, or Zero.

What can I help you build?

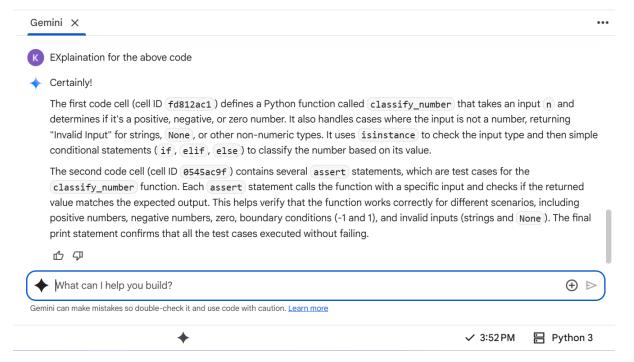
>Handle invalid inputs like strings and None.

>Include boundary conditions (-1, 0, 1).

Code and Output:



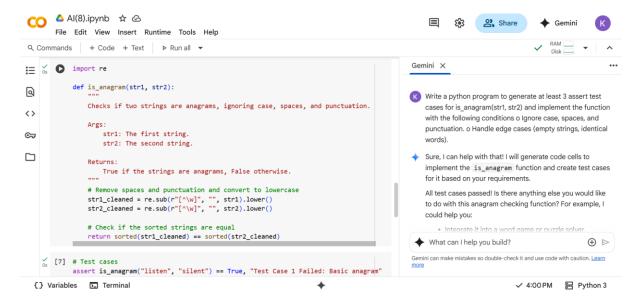
Explanation:



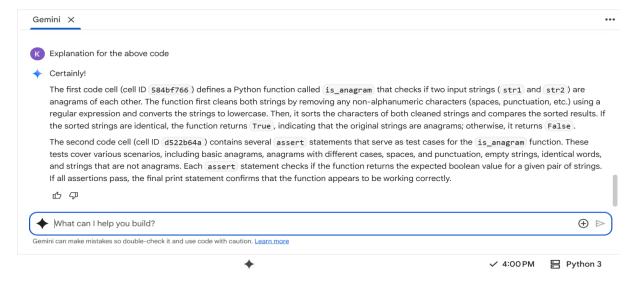
Task 3:

Write a python program to generate at least 3 assert test cases for is_anagram(str1, str2) and implement the function with the following conditions

- > Ignore case, spaces, and punctuation.
- >Handle edge cases (empty strings, identical words).







Task 4:

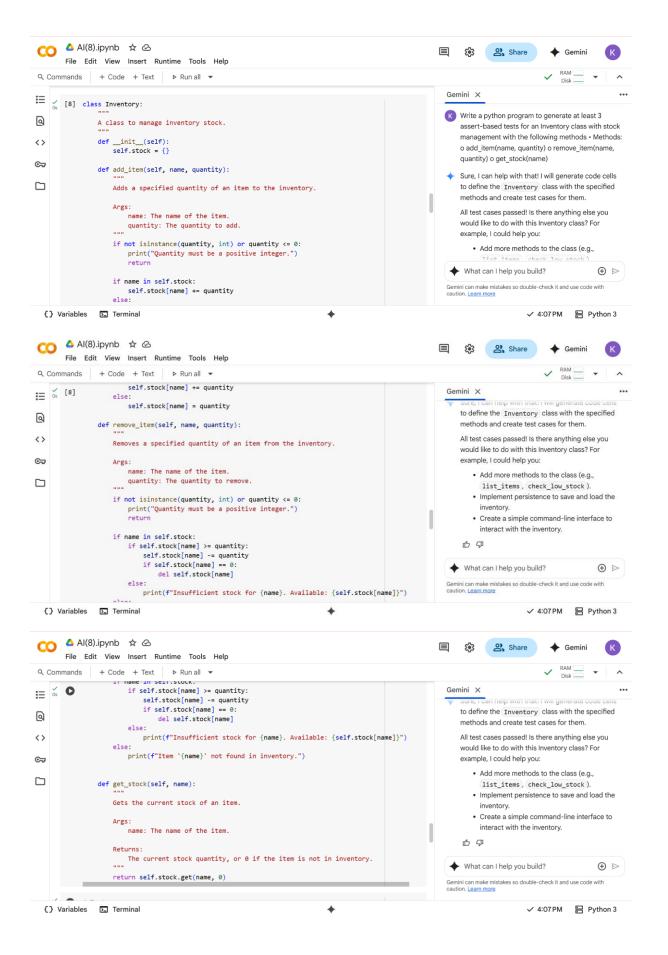
Write a python program to generate at least 3 assert-based tests for an Inventory class with stock management with the following methods

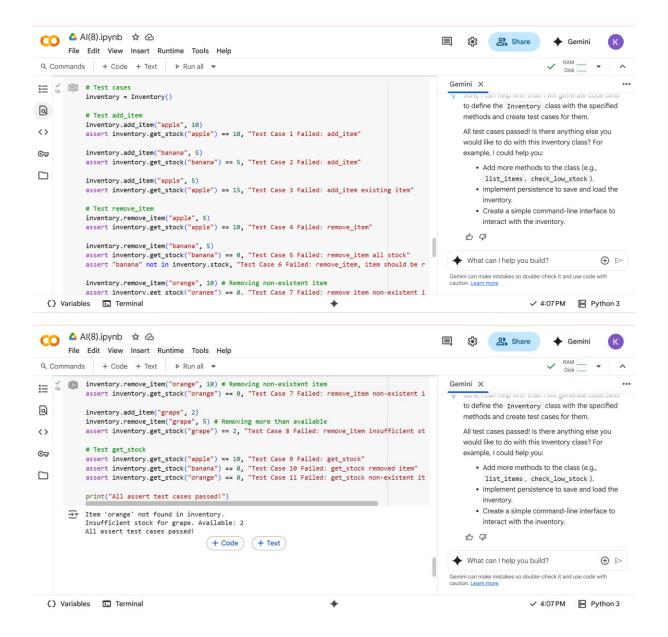
Methods:

>add_item(name, quantity).

> remove_item(name, quantity).

>get_stock(name).



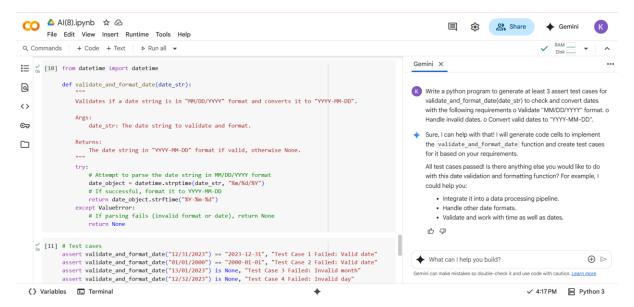


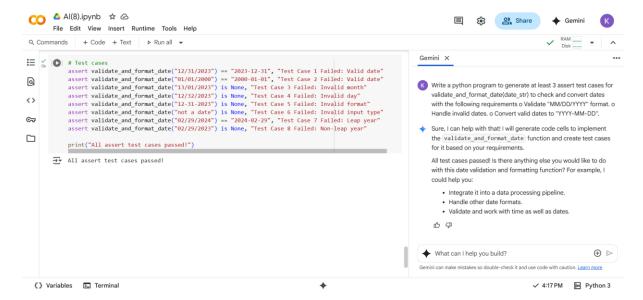


Task 5:

Write a python program to generate at least 3 assert test cases for validate_and_format_date(date_str) to check and convert dates with the following requirements

- > Validate "MM/DD/YYYY" format.
- >Handle invalid dates.
- > Convert valid dates to "YYYY-MM-DD".





Gemini X

K Explanation for the above code

Certainly!

The first code cell (cell ID 642f0218) defines a Python function called validate_and_format_date. This function takes a string date_str as input and attempts to validate if it represents a date in the "MM/DD/YYYY" format. If the format is valid, it converts the date to the "YYYY-MM-DD" format and returns the new string. If the input string is not in the expected format or represents an invalid date (like a non-existent day or month), the function catches the ValueError that occurs during parsing and returns None. It utilizes the datetime module's strptime method for parsing and strftime for formatting.

The second code cell (cell ID 777d526d) contains several assert statements which serve as test cases for the validate_and_format_date function. These tests cover valid dates that should be formatted correctly, as well as invalid inputs like incorrect months, days, formats, and non-date strings, which should result in None being returned. It also includes test cases for leap years to ensure accurate validation. The final print statement indicates that all the provided test cases have passed.

