**SR UNIVERSITY**

**LAB ASSIGNMENT-8**

**NAME:POMMEDI.BHAVANA**

**ROLL – 2403A54069**

**Batch – 03(DS)**

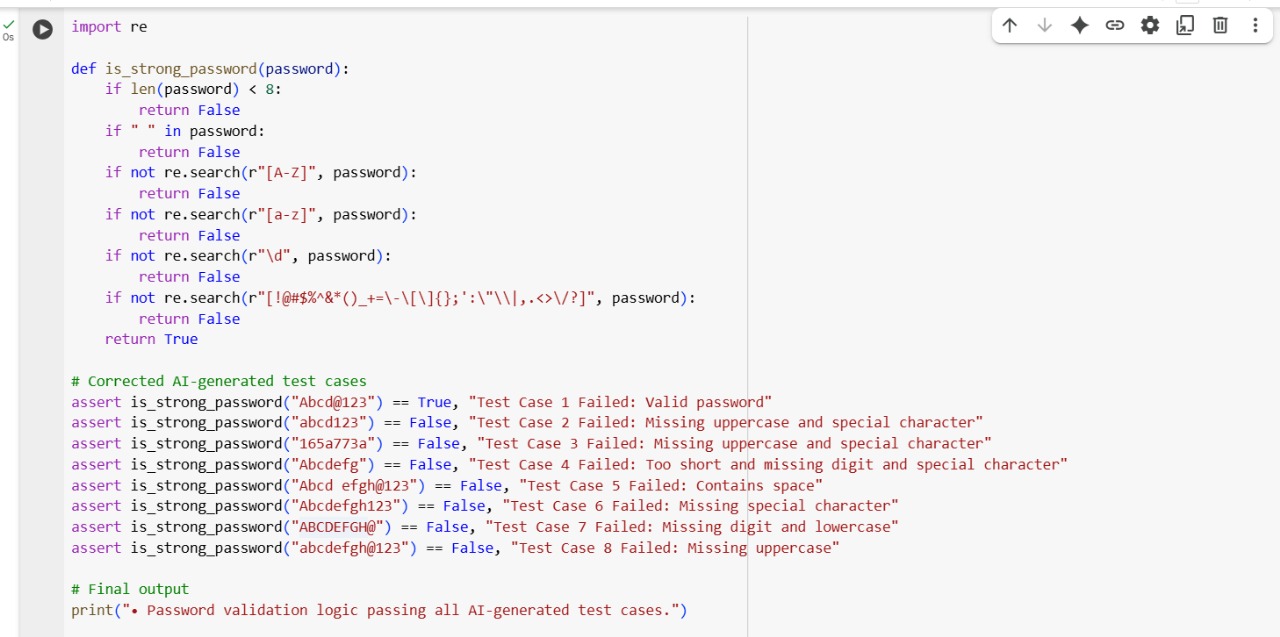
Task Description #1 (Password Strength Validator – Apply AI in  
Security Context)

Prompt:

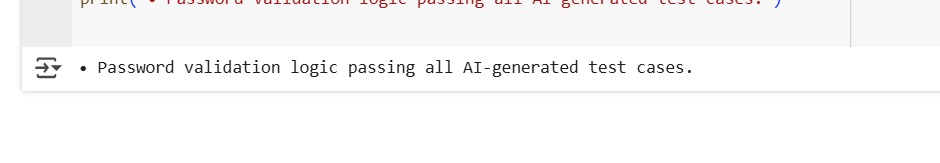
Prompt-genarate at least 5 assert test cases of a function is\_strong\_password(password) that checks password strength .

requirment:at least 8 characters,,must uppercase,lowercase,digit and special character,and must not conatain spaces

Code:



Output:



Explanation:

A screenshot of a computer program

AI-generated content may be incorrect.

Task Description #2 (Number Classification with Loops – Apply AI for  
Edge Case Handling)

Prompt:

Prompt-for students:”generate at least 5 aasert test cases for a function classify\_number(n)that returns ‘positive’,’negative’,’zero’,or ‘invalid input’:include boundary condition -1,0,1and some invalid inputs”.

Code:

A screenshot of a computer program

AI-generated content may be incorrect.

Output:

A close up of a word

AI-generated content may be incorrect.

Explantion:

A screenshot of a computer

AI-generated content may be incorrect.

Task Description #3 (Anagram Checker – Apply AI for String Analysis)

Prompt:

**Prompt**:  
Implement a Python function is\_anagram(str1, str2) that checks whether two strings are anagrams of each other.

**Requirements**:

* Ignore **case**, **spaces**, and **punctuation**.
* Handle **edge cases** such as empty strings and identical words.

Also, generate at least **3 assert test cases** to verify the correctness of the implementation.

Code:

A screenshot of a computer code

AI-generated content may be incorrect.

Output:



Explanation:

A screenshot of a computer

AI-generated content may be incorrect.

Task Description #4 (Inventory Class – Apply AI to Simulate Real-  
World Inventory System)

Prompt:

**Prompt**:

Create a Python class called Inventory to simulate a real-world stock management system. The class should support these methods:

* add\_item(name, quantity) — Adds the specified quantity of an item to inventory.
* remove\_item(name, quantity) — Removes the specified quantity. If quantity to remove exceeds available stock, remove all remaining.
* get\_stock(name) — Returns the current stock for the specified item (default to 0 if item not in inventory).

Then, generate **at least 3 assert-based test cases** to verify correct behavior of stock management.

Code:

A screenshot of a computer

AI-generated content may be incorrect.

Output:

A screenshot of a phone

AI-generated content may be incorrect.

Explanation:

A white text box with black text

AI-generated content may be incorrect.

Task Description #5 (Date Validation & Formatting – Apply AI for  
Data Validation)

Prompt:

**Prompt**:  
Create a Python function named validate\_and\_format\_date(date\_str) that performs the following:

1. Accepts a string input in the format "MM/DD/YYYY".
2. Validates whether the input is a correct and valid date.
   * Rejects non-existent dates (e.g., "02/30/2023").
   * Rejects incorrect formats (e.g., "2023/01/01", "15-10-2023").
3. If the date is valid, return the formatted version as "YYYY-MM-DD".
4. If invalid, return the string "Invalid Date".

Then, generate **at least 3 assert-based test cases** to verify:

* A valid date conversion
* An invalid date
* An edge case (e.g., leap year or incorrect format)

Code:

A screenshot of a computer

AI-generated content may be incorrect.

Output:

A close up of a sign

AI-generated content may be incorrect.

Explanation:

A text on a white background

AI-generated content may be incorrect.