ASSIGNMENT-8.5

Name: G.Vignesh

HT.No:2303A52359

Batch:36

Task Description #1 (Username Validator – Apply AI in Authentication Context)

- Task: Use AI to generate at least 3 assert test cases for a function is_valid_username(username) and then implement the function using Test-Driven Development principles.
- Requirements:

o Username length must be between 5 and 15 characters. o Must contain only alphabets and digits. o Must not start with a digit.

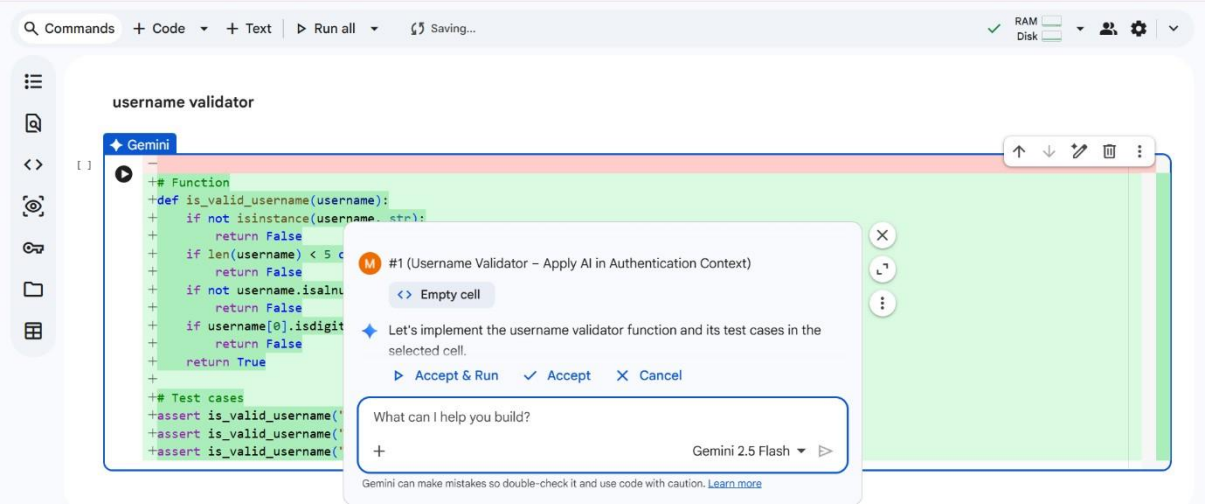
o No spaces allowed. Example Assert Test Cases:

```
assert is_valid_username("User123") == True
assert is_valid_username("12User") == False
assert is_valid_username("Us er") == False
```

Expected Output #1:

- Username validation logic successfully passing all AI-generated test cases.

Output:



Task Description #2 (Even–Odd & Type Classification – Apply AI for Robust Input Handling)

- Task: Use AI to generate at least 3 assert test cases for a function classify_value(x) and implement it using conditional logic and loops.
- Requirements:

o If input is an integer, classify as "Even" or "Odd".

o If input is 0, return "Zero".

o If input is non-numeric, return "Invalid Input".

Example Assert Test Cases:

assert classify_value(8) == "Even" assert

classify_value(7) == "Odd" assert

classify_value("abc") == "Invalid Input"

Expected Output #2:

- Function correctly classifying values and passing all test cases.

Output:

Task Description #3 (Palindrome Checker – Apply AI for String Normalization)

- Task: Use AI to generate at least 3 assert test cases for a function is_palindrome(text) and implement the function.
- Requirements:

o Ignore case, spaces, and punctuation.

o Handle edge cases such as empty strings and single characters.

Example Assert Test Cases:

assert is_palindrome("Madam") == True

assert is_palindrome("A man a plan a canal Panama") ==

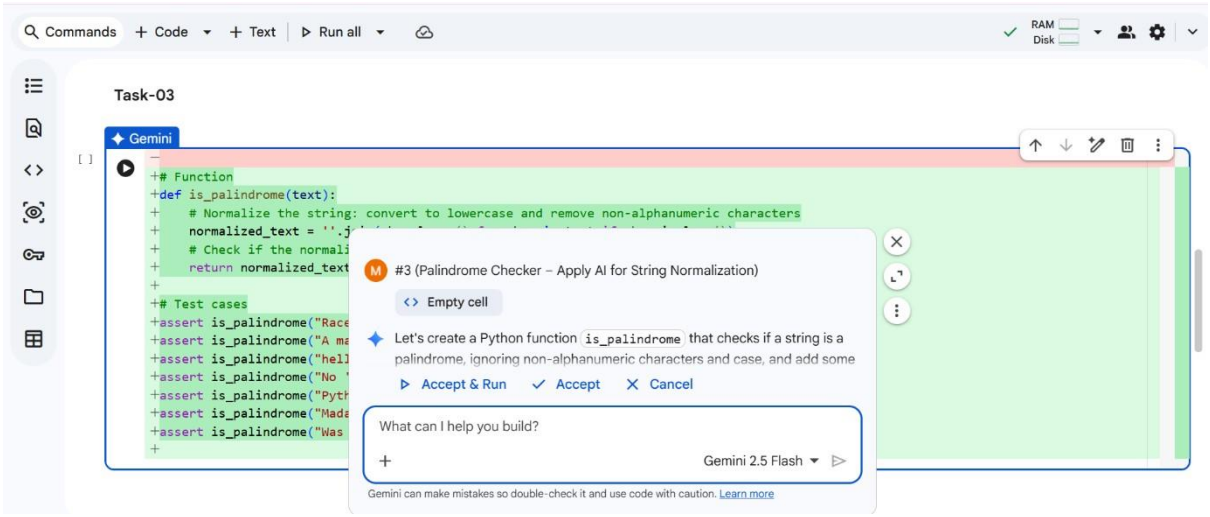
True

assert is_palindrome("Python") == False Expected

Output #3:

- Function correctly identifying palindromes and passing all AI-generated tests.

Output:



Task Description #4 (BankAccount Class – Apply AI for

Object-Oriented Test-Driven Development)

- Task: Ask AI to generate at least 3 assert-based test cases for a BankAccount class and then implement the class.
- Methods:

o deposit(amount) o

withdraw(amount)

o get_balance()

Example Assert Test Cases:

acc = BankAccount(1000)

acc.deposit(500)

assert acc.get_balance() == 1500

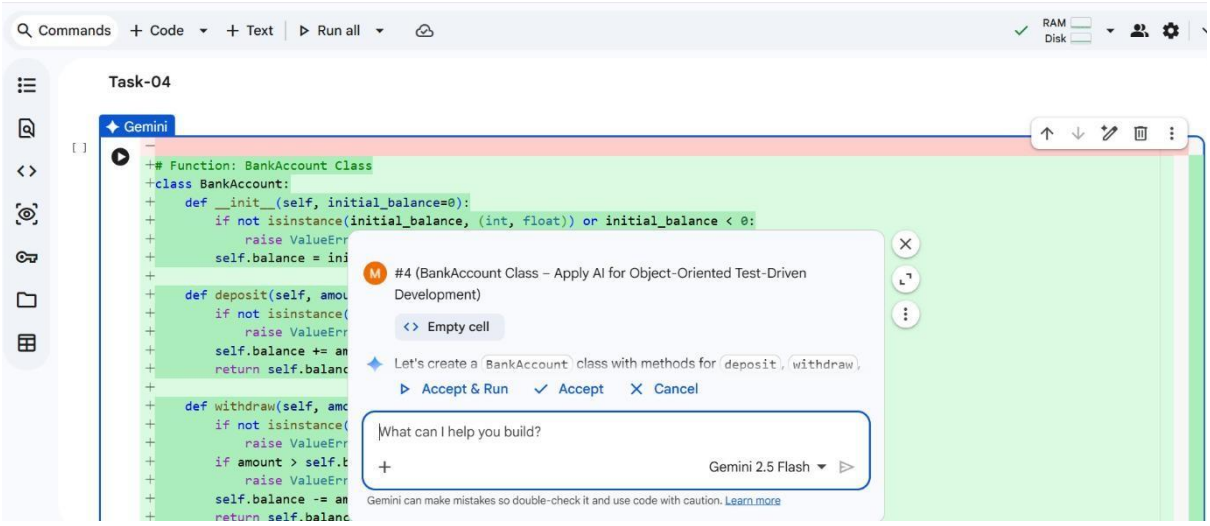
acc.withdraw(300)

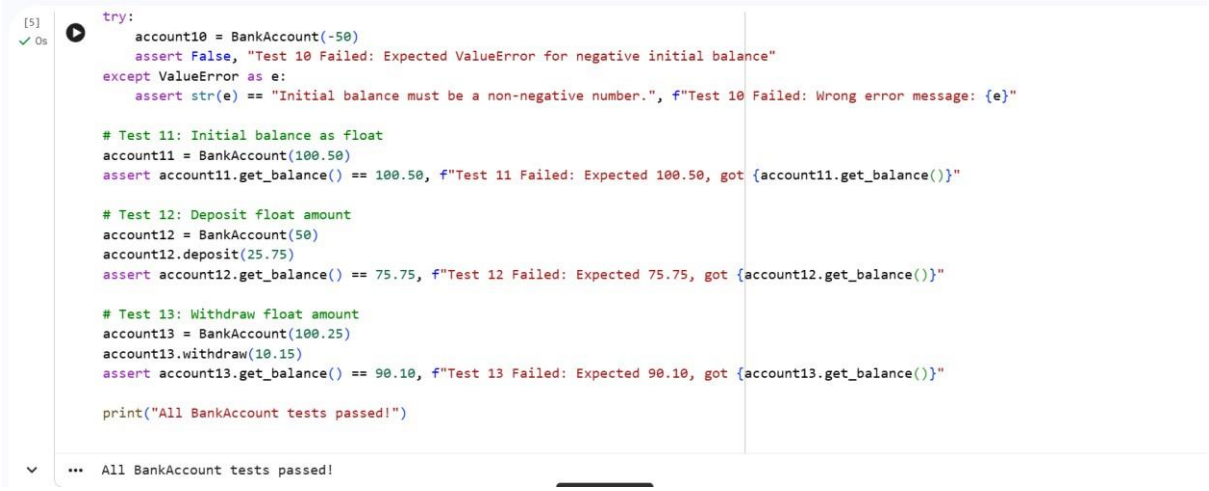
assert acc.get_balance() == 1200 Expected

Output #4:

- Fully functional class that passes all AI-generated assertions.

Output:





Task Description #5 (Email ID Validation – Apply AI for Data Validation)

- Task: Use AI to generate at least 3 assert test cases for a function validate_email(email) and implement the function.
- Requirements:

o Must contain @ and . o Must not start or end with special characters. o Should handle invalid formats gracefully.

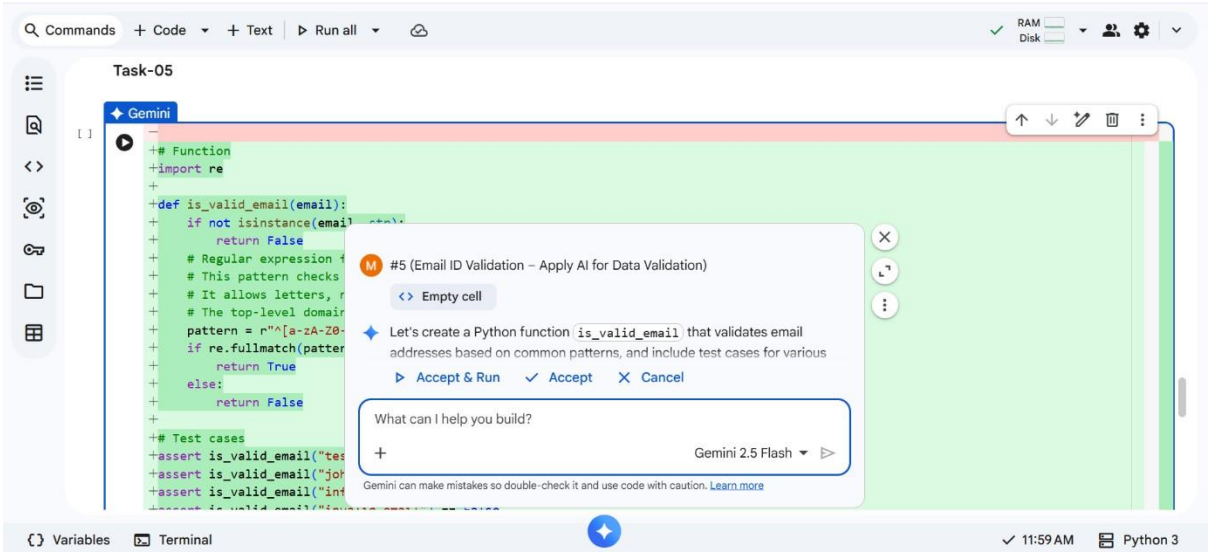
Example Assert Test Cases:

```
assert validate_email("user@example.com") == True
assert validate_email("userexample.com") == False
assert validate_email("@gmail.com") == False
```

Expected Output #5:

- Email validation function passing all AI-generated test cases and handling edge cases correctly.

Output:


Commands
Code
Text
Run all
RAM
Disk
Task-05
Gemini
# Function
import re
def is_valid_email(email):
if not isinstance(email
return False
# Regular expression
# This pattern checks
# It allows letters,
# The top-level domain
pattern = r"^[a-zA-Z0-
if re.fullmatch(patter
return True
else:
return False
Test cases
assert is_valid_email("tes
assert is_valid_email("joh
assert is_valid_email("inv
#5 (Email ID Validation – Apply AI for Data Validation)
Empty cell
Let's create a Python function is_valid_email that validates email addresses based on common patterns, and include test cases for various
Accept & Run
Accept
Cancel
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more
Variables
Terminal
11:59 AM
Python 3