ASSIGNMENT:2.2

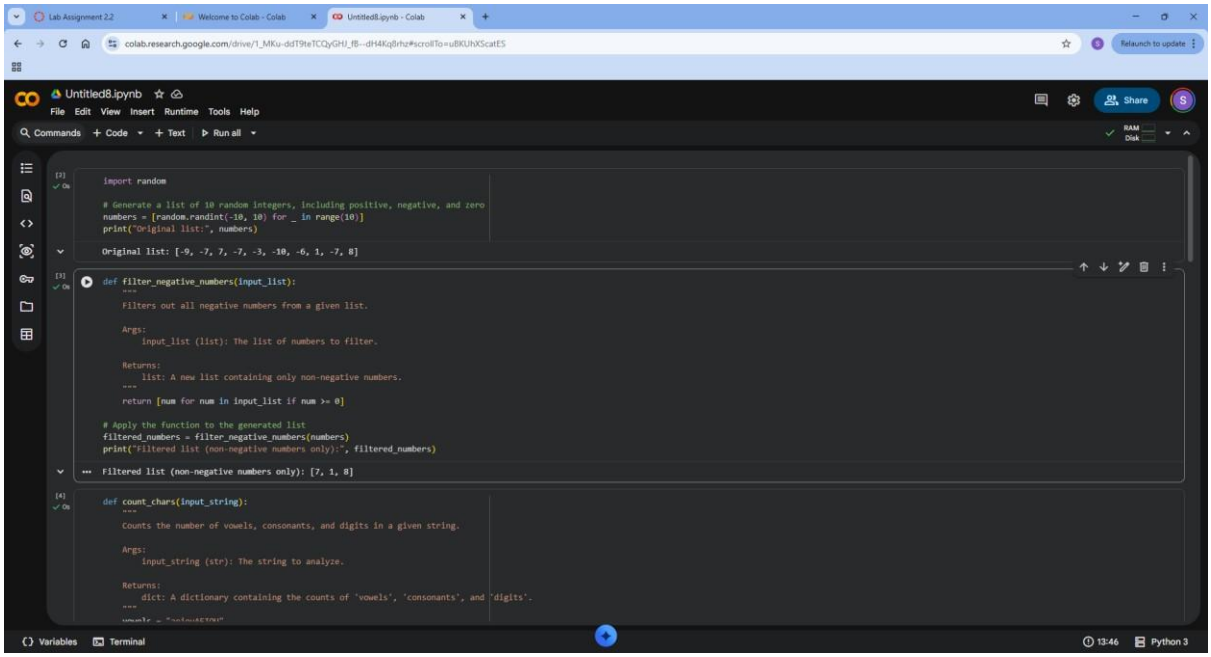
NAME-TEENESWARI

ROLL:NO:2303A51932

BATCH-30

TASK-1

PROMPT: generate a function that filters out all negative numbers from a list .(first generate list) CODE:



OBSERVATION:

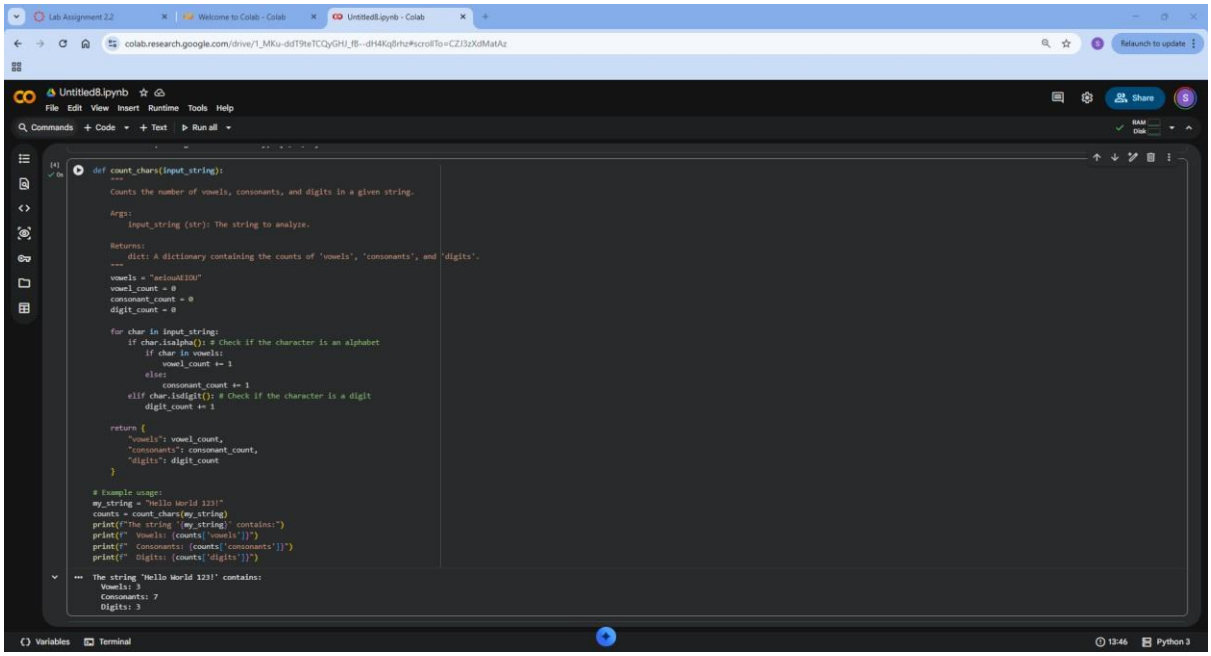
The program uses Python random module to generate a list of ten integers ranging from minus ten to ten so the list may contain negative numbers zero and positive numbers The original list is printed first showing all randomly generated values A function named filter-negative-numbers is defined to remove negative numbers from the list

Inside the function list comprehension is used with the condition num greater than or equal to zero which keeps only zero and positive values The function does not modify the original list instead it returns a new filtered list The filtered list output contains only non-negative numbers proving that the filtering logic works correctly

TASK-2

PROMPT: generate a Python function that counts vowels, consonants, and digits in a string

CODE:



OBSERVATION:

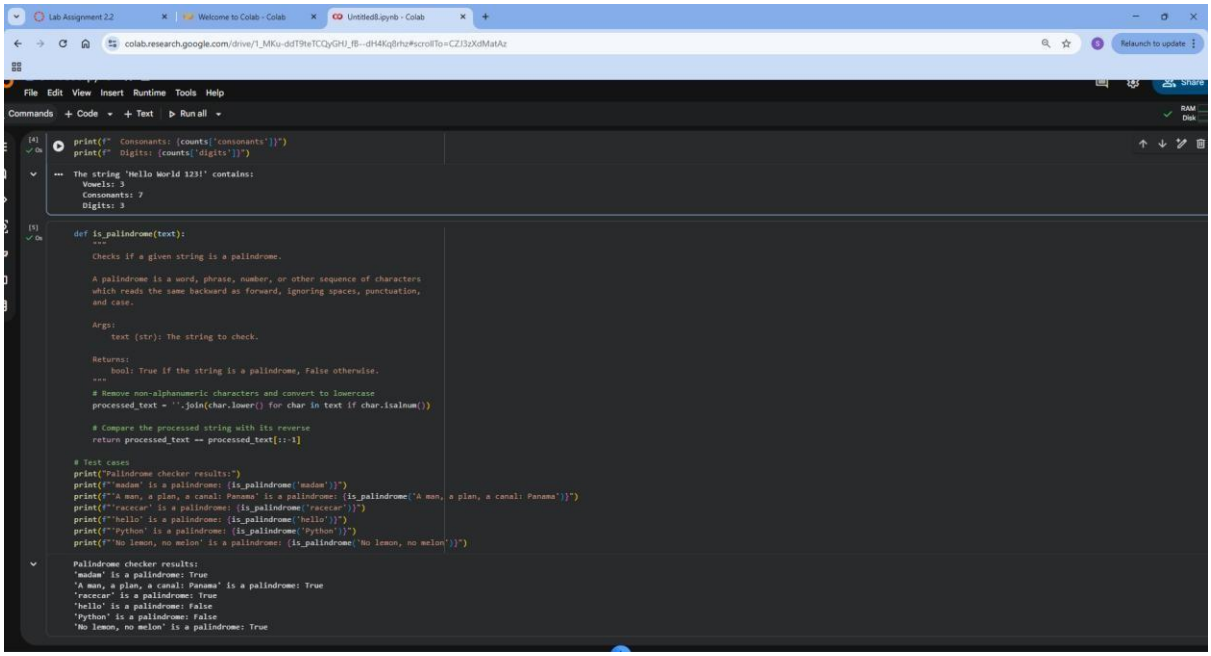
The program defines a function named count_chars that analyzes a given string It checks each character in the string one by one using a loop

If the character is an alphabet it further checks whether it is a vowel or a consonant Vowels are identified using a predefined string containing both uppercase and lowercase vowels If the character is a digit it increases the digit count Special characters spaces and symbols are ignored The function keeps separate counters for vowels consonants and digits It returns the final counts in the form of a dictionary The example string Hello World 123 is passed to the function The output correctly displays the number of vowels consonants and digits present in the string

TASK-3

PROMPT: Generate a palindrome-checking function and compare the results.

CODE:



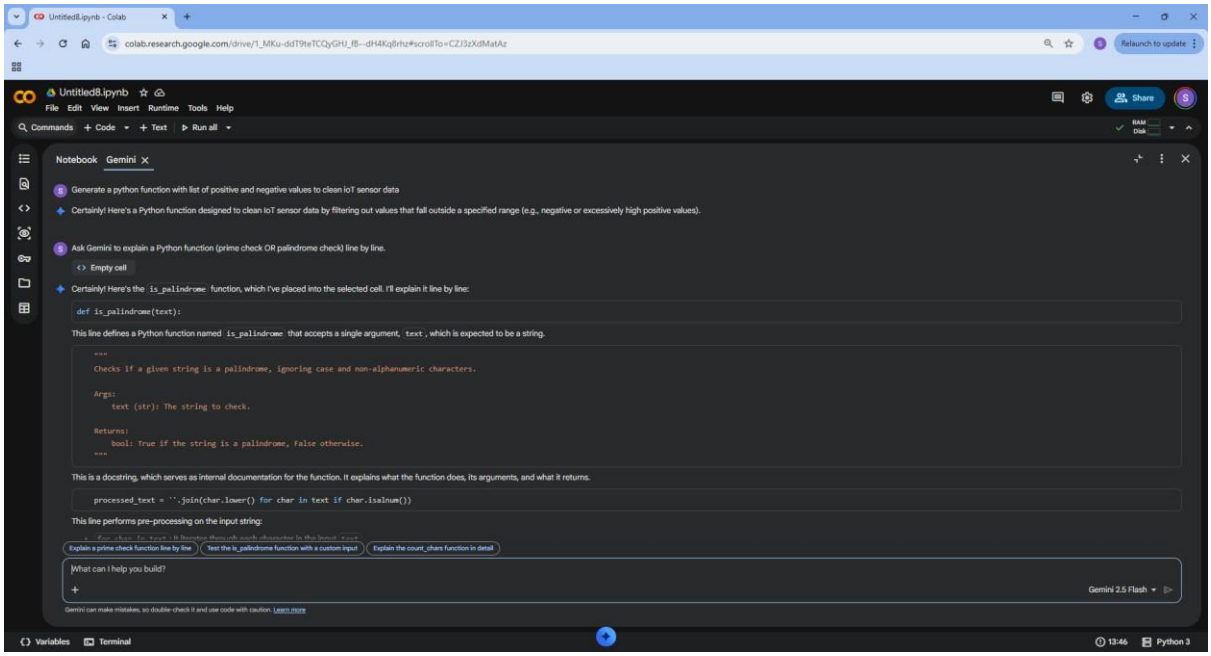
OBSERVATION:

The program defines a function named is_ palindrome that checks whether a given string is a palindrome It first removes all non-alphanumeric characters such as spaces punctuation and symbols All remaining characters are converted to lowercase to ensure case insensitive comparison The cleaned string is then compared with its reverse using slicing If both strings are the same the function returns true otherwise it returns false This approach allows phrases and sentences to be checked correctly not just single words

TASK-4

PROMPT: explain a Python function (prime check OR palindrome check) line by line.

CODE:



OBSERVATION:

Explanation:

def is_palindrome(text):

This line defines a function named is_palindrome that takes one argument: text (which is expected to be a string).

"""..."""

This is a docstring, which provides a brief description of what the function does, its arguments (Args), and what it returns (Returns). It's good practice for documenting code.

processed_ text = ''. join(char. lower() for char in text if char.isalnum()) This is

the core of the pre-processing step:

for char in text: It iterates through each character in the input text string.

if char. isalnum(): It checks if the character is alphanumeric (i.e., a letter or a number). This effectively filters out spaces, punctuation, and other symbols.

char. lower(): For each alphanumeric character, it converts it to lowercase. This ensures that case doesn't affect the palindrome check (e.g., 'Racecar' should be considered a palindrome).

''.join(...): It concatenates all the filtered and lowercased characters back into a single string. The result is stored in the processed_ text variable.

return processed_ text == processed_ text[::-1]

This line performs the actual palindrome check and returns the result:

processed_ text[::-1]: This creates a reversed version of the processed_ text string using Python's slicing notation. [::-1] means start from the end, go to the beginning, with a step of -1.

processed_ text == processed_ text[::-1]: It compares the processed_ text with its reversed version. If they are identical, the expression evaluates to True, indicating it's a palindrome. Otherwise, it evaluates to False. return: The Boolean result (True or False) is returned by the function.