

K. Ruthwik Bhat

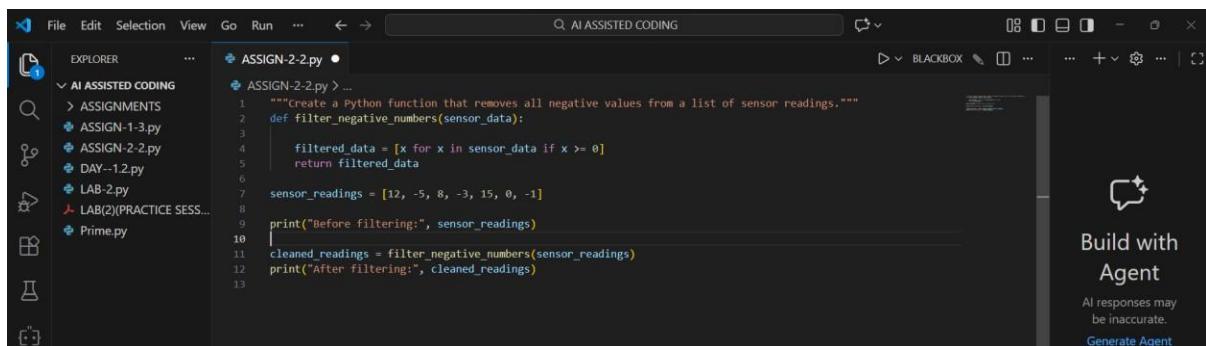
2403A51L08

B-51

ASSIGNMENT -2.2

Task 1: Cleaning Sensor Data

PROMPT: Create a Python function that removes all negative values from a list of sensor readings.

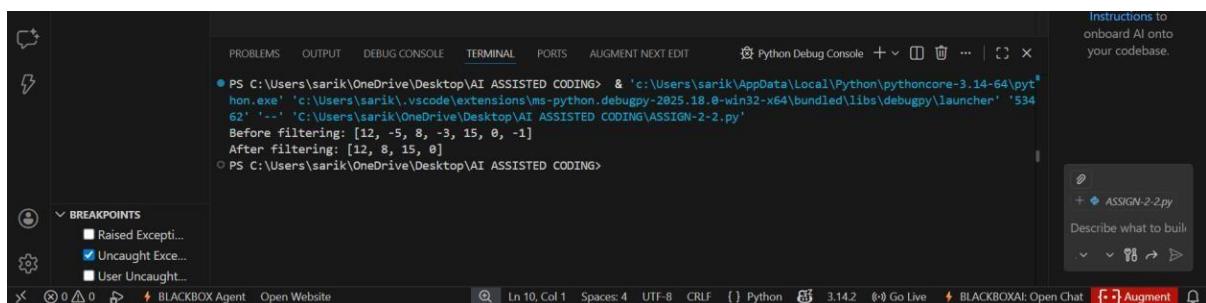


A screenshot of the Visual Studio Code interface. The left sidebar shows the file tree with 'ASSIGN-2-2.py' selected. The main editor area contains the following Python code:

```
1 """Create a Python function that removes all negative values from a list of sensor readings."""
2 def filter_negative_numbers(sensor_data):
3     filtered_data = [x for x in sensor_data if x >= 0]
4     return filtered_data
5
6 sensor_readings = [12, -5, 8, -3, 15, 0, -1]
7
8 print("Before filtering:", sensor_readings)
9 cleaned_readings = filter_negative_numbers(sensor_readings)
10
11 print("After filtering:", cleaned_readings)
12
```

The right sidebar has a 'Build with Agent' section with a message: 'AI responses may be inaccurate.' and a 'Generate Agent' button.

OUTPUT:



A screenshot of the terminal window in VS Code. The output shows the execution of the script and its results:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53462' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
Before filtering: [12, -5, 8, -3, 15, 0, -1]
After filtering: [12, 8, 15, 0]
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

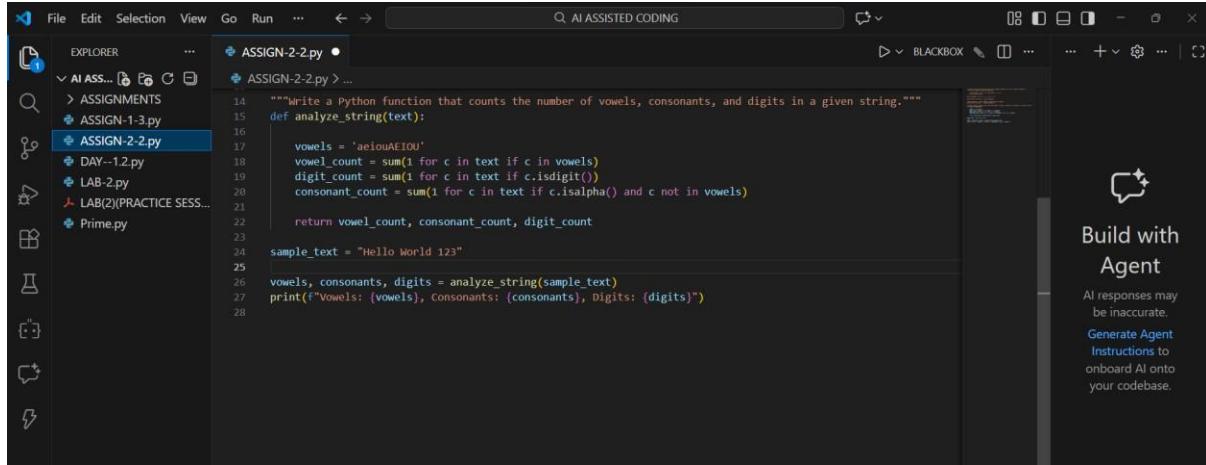
EXPLANATION:

This function removes invalid negative sensor values using list comprehension.

Only values greater than or equal to zero are retained, ensuring clean IoT sensor data.

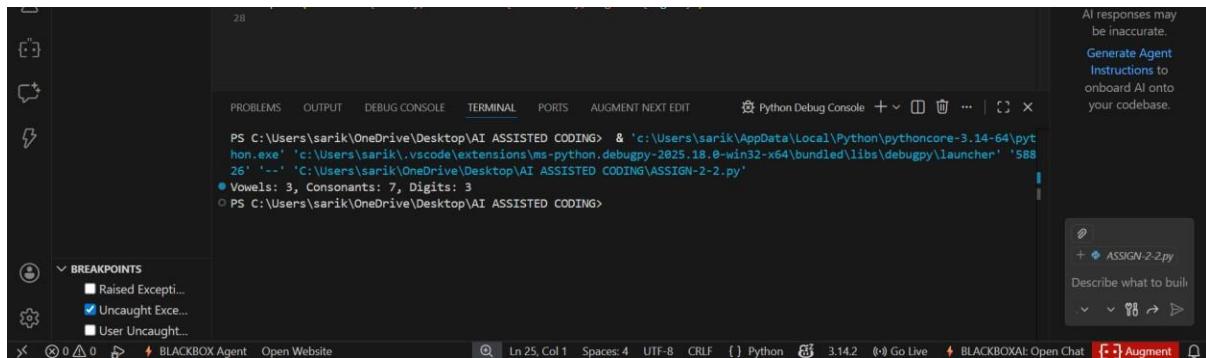
Task 2: String Character Analysis

PROMPT: Write a Python function that counts the number of vowels, consonants, and digits in a given string.



```
14 """Write a Python function that counts the number of vowels, consonants, and digits in a given string."""
15 def analyze_string(text):
16     vowels = 'aeiouAEIOU'
17     vowel_count = sum(1 for c in text if c in vowels)
18     digit_count = sum(1 for c in text if c.isdigit())
19     consonant_count = sum(1 for c in text if c.isalpha() and c not in vowels)
20
21     return vowel_count, consonant_count, digit_count
22
23 sample_text = "Hello World 123"
24
25 vowels, consonants, digits = analyze_string(sample_text)
26 print(f"Vowels: {vowels}, Consonants: {consonants}, Digits: {digits}")
27
28
```

OUTPUT:



```
PS C:\Users\sarik\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58826' '--' 'C:\Users\sarik\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
● Vowels: 3, Consonants: 7, Digits: 3
○ PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

EXPLANATION:

The function iterates through each character and classifies it as a vowel, consonant, or digit.

Python string methods like `isalpha()` and `isdigit()` improve accuracy and readability.

Task 3: Palindrome Check - Tool Comparison

Gemini Prompt: Write a Python function to check if a string is a palindrome. Ignore spaces and capitalization.

File Edit Selection View Go Run ... ↺ ↻

AI ASSISTED CODING

RUN AND DEBUG ...

RUN

Run and Debug

To customize Run and Debug create a launch.json file.

Debug using a terminal command or in an interactive chat.

ASSIGN-2-2.py

```
11
12     return vowel_count, consonant_count, digit_count
13
14     sample_text = "Hello World 123"
15
16     vowels, consonants, digits = analyze_string(sample_text)
17     print(f"vowels: {vowels}, consonants: {consonants}, Digits: {digits}"")
18
19
20     #Gemini Prompt:
21     "Write a python function to check if a string is a palindrome. Ignore spaces and capitalization."
22     def is_palindrome_gemini(s):
23         s = s.replace(" ", "").lower()
24         return s == s[::-1]
25
26     print(is_palindrome_gemini("Racecar"))
```

OUTPUT:

The screenshot shows a Python code editor interface. The code in the editor is:

```
32 |     s = s[::-1]
33 |
34 |     return s == s[::-1]
35 |
36 print(is_palindrome_gemini("racecar"))
37
```

The terminal output shows the execution of the code in a Python debugger:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58783' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
True
```

The interface includes a 'CALL STACK' section, a 'PROBLEMS' tab, and a 'BREAKPOINTS' section where 'Uncaught Exceptions' are listed.

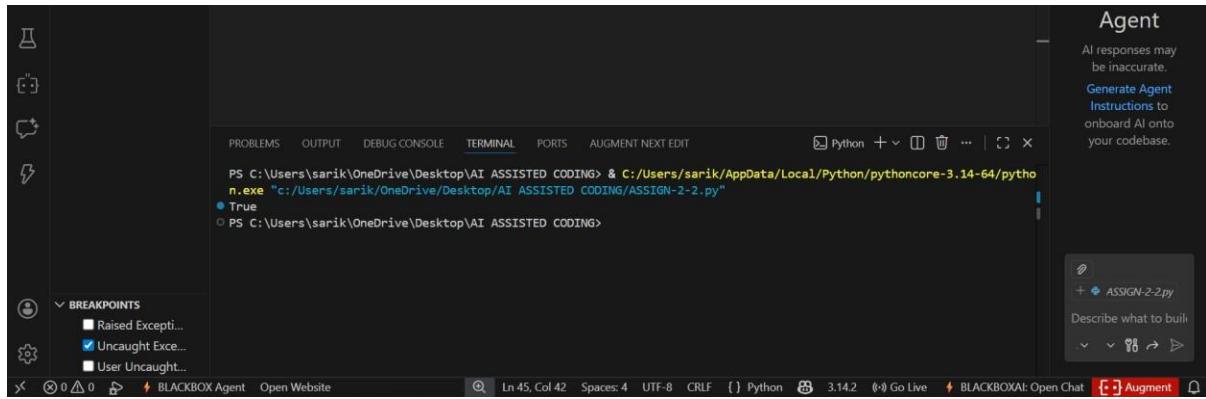
Copilot Prompt: Write a Python function to check palindrome. Consider only letters and ignore case.

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and other standard menu items. A search bar contains the text 'Q. AI ASSISTED CODING'. The left sidebar has icons for file operations like 'RUN AND DEBUG' and 'RUN'. A prominent blue button labeled 'Run and Debug' is highlighted. Below it, a message says 'To customize Run and Debug create a launch.json file.' Another message below that says 'Debug using a terminal command or in an interactive chat.' The main editor area displays Python code for a palindrome checker:

```
37:     #Copilot Prompt:
38:     """
39:     Write a Python function to check palindrome. consider only letters and ignore case.
40:     def is_palindrome_copilot(text):
41:         cleaned_text = ''.join(c.lower() for c in text if c.isalnum())
42:         return cleaned_text == cleaned_text[::-1]
43:
44:
45: print(is_palindrome_copilot("Racecar"))
```

To the right of the editor, there's a sidebar titled 'BLACKBOX' with a list of items: 'is_palindrome_copilot', 'is_anagram_copilot', and 'is_pangram_copilot'. On the far right, there's a 'Build with Agent' section featuring a speech bubble icon and the text 'Build with Agent' and 'AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.'

OUTPUT:



Comparison Table:

Feature	Gemini	Copilot
Clarity	Simple, minimal code	Slightly longer, more robust
Handling spaces/case	Ignores spaces, converts to lowercase	Ignores spaces and punctuation, lowercase
Readability	Very clear	Clear, slightly more detailed
Efficiency	Uses string slicing	Uses string comprehension

EXPLANATION:

Gemini provides concise and easy-to-read logic, making it beginner-friendly.

Copilot generates more robust code that handles punctuation and special characters.

Task 4: Code Explanation Using AI

Step 1 - Code Snippet:

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Run. The main area displays a Python script named 'ASSIGN-2-2.py'. The code contains the following function:

```
47
48     ##Step 1 - Code Snippet(Code Explanation):
49     def is_palindrome(text):
50         text = text.replace(" ", "").lower() # Remove spaces and lowercase
51         return text == text[::-1] # Compare string with its reverse
52
53
```

To the right of the code, there is an AI-assisted coding panel titled 'AI ASSISTED CODING' with a progress bar at 100%. A message bubble icon is present, along with a 'Build with Agent' button and a note about AI responses being inaccurate. Below this is a link to 'Generate Agent Instructions to onboard AI onto your codebase.'

Step 2 - AI Explanation:

1. `text.replace(" ", "").lower()` → Removes spaces and converts letters to lowercase.
2. `text == text[::-1]` → Checks if the string is equal to its reverse.

EXPLANATION:

The function normalizes the string to avoid case and space mismatches. It then compares the string with its reverse to verify palindrome logic.