

ASSIGNMENT – 4.2

NAME : M.SAI CHARANI

BATCH : 34

HALL TICKET NUMBER : 2303A52192

Task Description-1

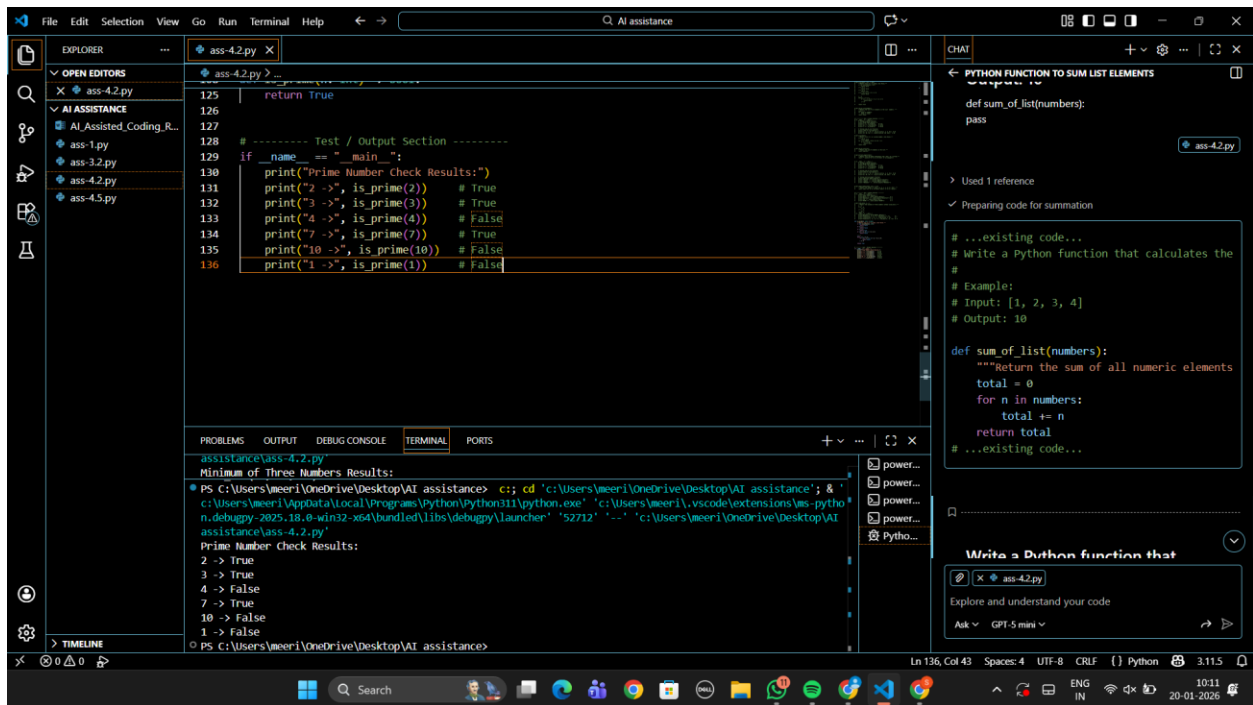
- Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

The screenshot displays a Visual Studio Code (VS Code) environment with the following components:

- EXPLORER:** Shows a project named 'AI CODING' with files: `assignment_1.2.py`, `assignment_3.2.py`, `assignment_4.2.py` (selected), `assignment_4.5.py`, `Iterative_factorial.py`, and `nb_email_classifier.pkl`.
- EDITOR:** Contains the code for `assignment_4.2.py`. The code defines a function `is_prime(n: int) -> bool:` that checks for primality using divisibility rules and a loop. It includes a test section at the bottom.

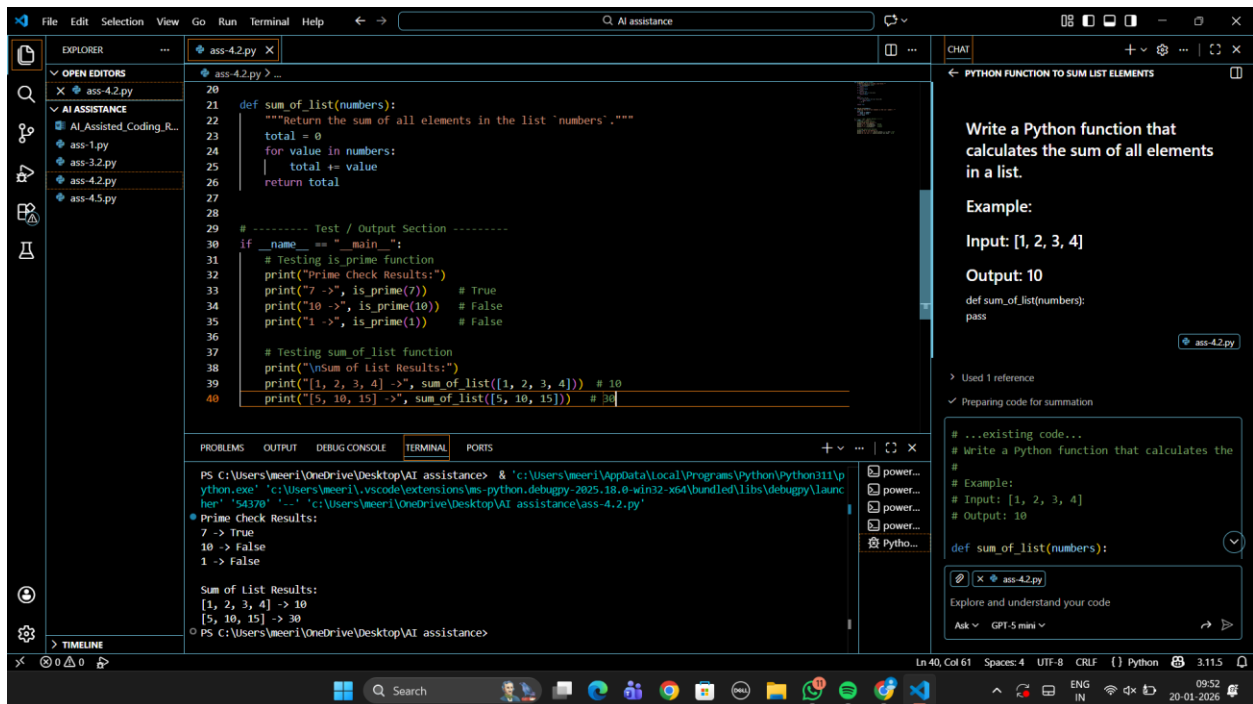
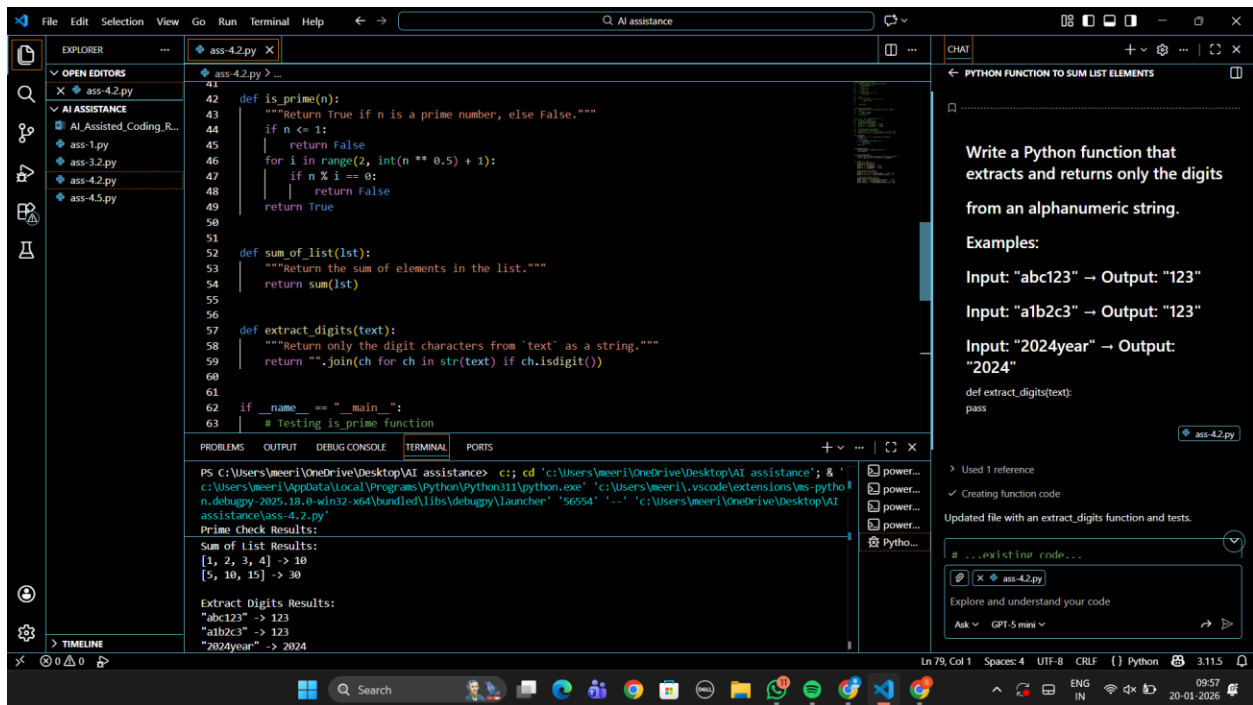
```
1 def is_prime(n: int) -> bool:
2     """Return True if n is prime, else False."""
3     if not isinstance(n, int):
4         return False
5     if n <= 1:
6         return False
7     if n <= 3:
8         return True
9     if n % 2 == 0 or n % 3 == 0:
10        return False
11
12    i = 5
13    while i * i <= n:
14        if n % i == 0 or n % (i + 2) == 0:
15            return False
16        i += 6
17
18    return True
19
20
21 # ----- Test / Output Section -----
22 if __name__ == "__main__":
23     print("Prime Number Check Results:")
24     print("2 ->", is_prime(2))    # True
```
- TERMINAL:** Shows the execution of the script using `python` and `hon.exe`. The output is:

```
PS C:\Users\LENDKO\OneDrive\Desktop\New folder\ai_coding> python .\assignment_4.2.py
hon.exe "C:\Users\LENDKO\OneDrive\Desktop\New folder\ai_coding\assignment_4.2.py"
PS C:\Users\LENDKO\OneDrive\Desktop\New folder\ai_coding\assignment_4.2.py
Prime Number Check Results:
2 -> True
3 -> True
4 -> False
7 -> True
10 -> False
1 -> False
```
- CHAT:** A sidebar chat window titled 'PRIME NUMBER DETERMINATION IN P...' shows a conversation. The user prompt is 'otherwise.' and the AI response is a code snippet for `is_prime(n)`. Below the chat, there is a section 'Used 1 reference' showing a code block with the same `is_prime` function definition as in the editor.



Task Description-2

- One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.



Task Description-3

- Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.

```

41
42 def is_prime(n):
43     """Return True if n is a prime number, else False."""
44     if n <= 1:
45         return False
46     for i in range(2, int(n ** 0.5) + 1):
47         if n % i == 0:
48             return False
49     return True
50
51
52 def sum_of_list(list):
53     """Return the sum of elements in the list."""
54     return sum(list)
55
56
57 def extract_digits(text):
58     """Return only the digit characters from 'text' as a string."""
59     return "".join(ch for ch in str(text) if ch.isdigit())
60
61
62 if __name__ == "__main__":
63     # Testing is_prime function

```

PS C:\Users\meeri\OneDrive\Desktop\AI assistance> c:\cd 'c:\Users\meeri\OneDrive\Desktop\AI assistance'; & 'c:\Users\meeri\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\meeri\.vscode\extensions\ms-python-debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '56554' '-' 'c:\Users\meeri\OneDrive\Desktop\AI assistance\ass-4.2.py'

Prime Check Results:

```

7 -> True
10 -> False
1 -> False

```

Sum of List Results:

```

[1, 2, 3, 4] -> 10
[5, 10, 15] -> 30

```

Extract Digits Results:

```

"abc123" -> 123
"a1b2c3" -> 123
"2024year" -> 2024

```

Write a Python function that extracts and returns only the digits from an alphanumeric string.

Examples:

Input: "abc123" → Output: "123"

Input: "a1b2c3" → Output: "123"

Input: "2024year" → Output: "2024"

def extract_digits(text):
pass

```

62 if __name__ == "__main__":
63     # Testing is_prime function
64     print("Prime Check Results:")
65     print("7 -> ", is_prime(7)) # True
66     print("10 -> ", is_prime(10)) # False
67     print("1 -> ", is_prime(1)) # False
68
69     # Testing sum_of_list function
70     print("\nSum of List Results:")
71     print("[1, 2, 3, 4] -> ", sum_of_list([1, 2, 3, 4])) # 10
72     print("[5, 10, 15] -> ", sum_of_list([5, 10, 15])) # 30
73
74     # Testing extract_digits function
75     print("\nExtract Digits Results:")
76     print("abc123 -> ", extract_digits("abc123")) # "123"
77     print("a1b2c3 -> ", extract_digits("a1b2c3")) # "123"
78     print("2024year -> ", extract_digits("2024year")) # "2024"
79

```

PS C:\Users\meeri\OneDrive\Desktop\AI assistance> c:\cd 'c:\Users\meeri\OneDrive\Desktop\AI assistance'; & 'c:\Users\meeri\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\meeri\.vscode\extensions\ms-python-debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '56554' '-' 'c:\Users\meeri\OneDrive\Desktop\AI assistance\ass-4.2.py'

Prime Check Results:

```

7 -> True
10 -> False
1 -> False

```

Sum of List Results:

```

[1, 2, 3, 4] -> 10
[5, 10, 15] -> 30

```

Extract Digits Results:

```

"abc123" -> 123
"a1b2c3" -> 123
"2024year" -> 2024

```

Write a Python function that extracts and returns only the digits from an alphanumeric string.

Examples:

Input: "abc123" → Output: "123"

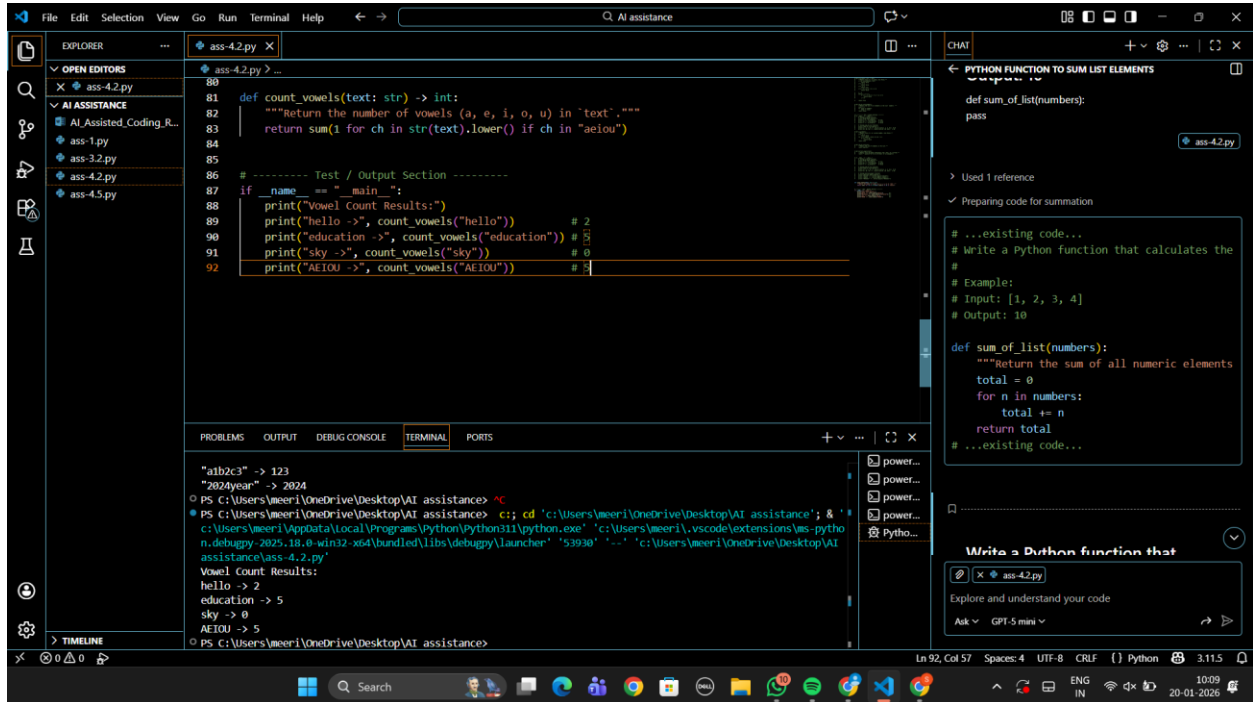
Input: "a1b2c3" → Output: "123"

Input: "2024year" → Output: "2024"

def extract_digits(text):
pass

Task Description-4

- Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.



Task Description-5

- Use few-shot prompting with 3 sample inputs to generate a function that determines the minimum of three numbers without using the built-in `min()` function.

