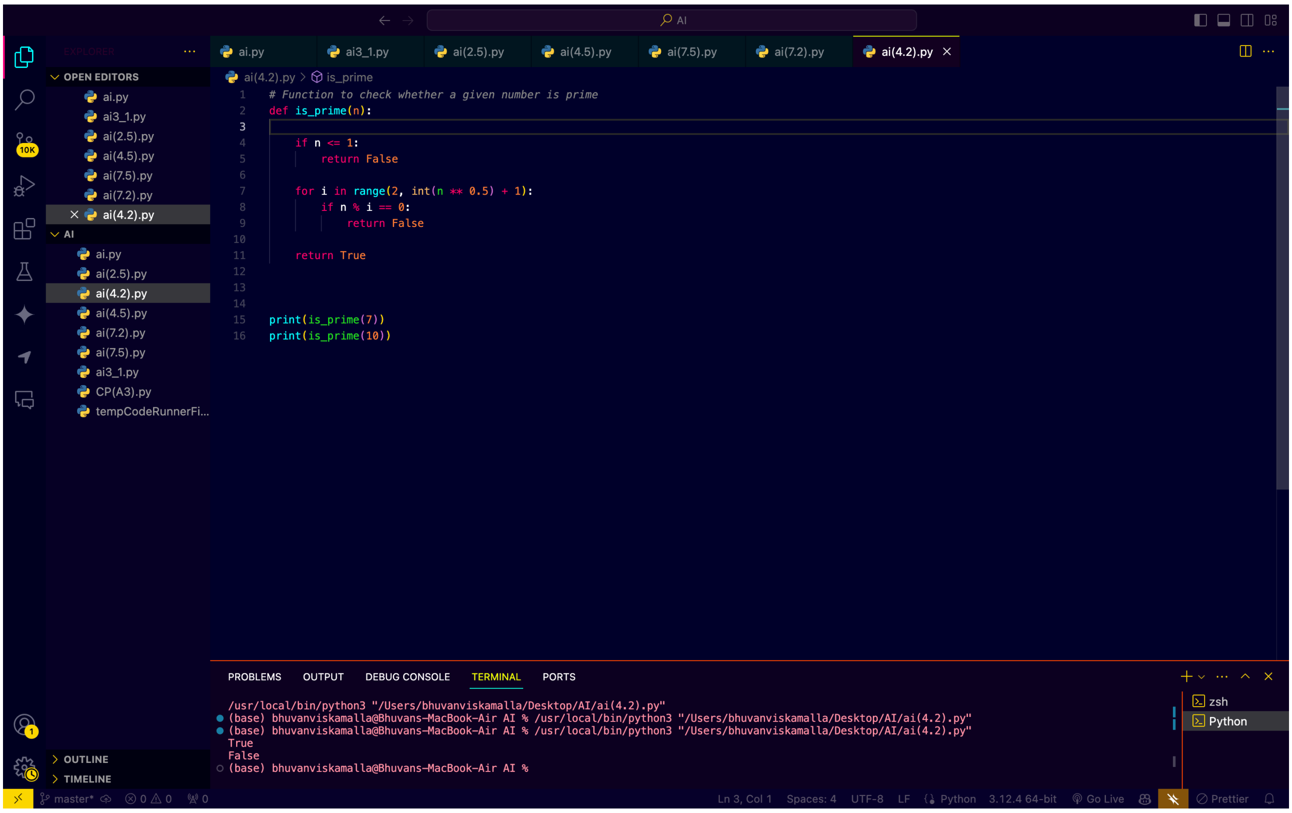
ASSIGNMENT-4.2

NAME- ARCHITHA
ROLLNO: 2306A91001

TASK-1: ZERO-SHOT PROMPTING

PROMPT: Write a Python function to determine whether a given number is prime.

CODE:



```python
# Function to check whether a given number is prime
def is_prime(n):

    if n <= 1:
        return False

    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False

    return True



print(is_prime(7))
print(is_prime(10))
```

```
/usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop/AI/ai(4.2).py"
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop/AI/ai(4.2).py"
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop/AI/ai(4.2).py"
True
False
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

OBSERVATION:

- AI model understands the concept of a prime number without being given any examples or additional guidance -It applies correct mathematical reasoning purely from the instruction -The model generates syntactically correct and logically sound Python code

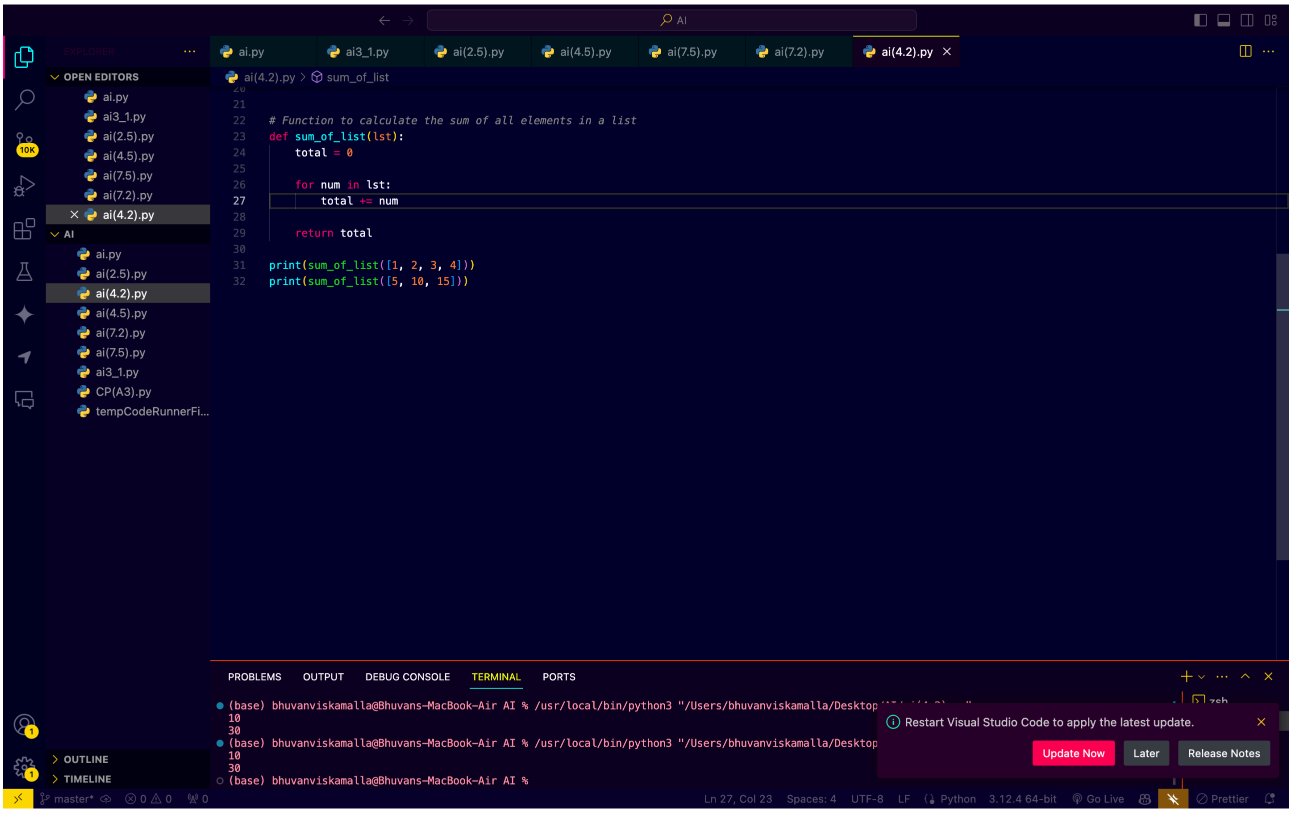
TASK-2

PROMPT: Write a Python function that calculates the sum of elements in a list.

Example:

Input: [1, 2, 3, 4]
Output: 10

CODE:



```python
# Function to calculate the sum of all elements in a list
def sum_of_list(lst):
    total = 0

    for num in lst:
        total += num

    return total

print(sum_of_list([1, 2, 3, 4]))
print(sum_of_list([5, 10, 15]))
```

```
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
10
30
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
10
30
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

Restart Visual Studio Code to apply the latest update.

OBSERVATION:

The single example guides the AI model to understand the expected input and output relationship The model correctly generalises the pattern from the example to any list of Numbers

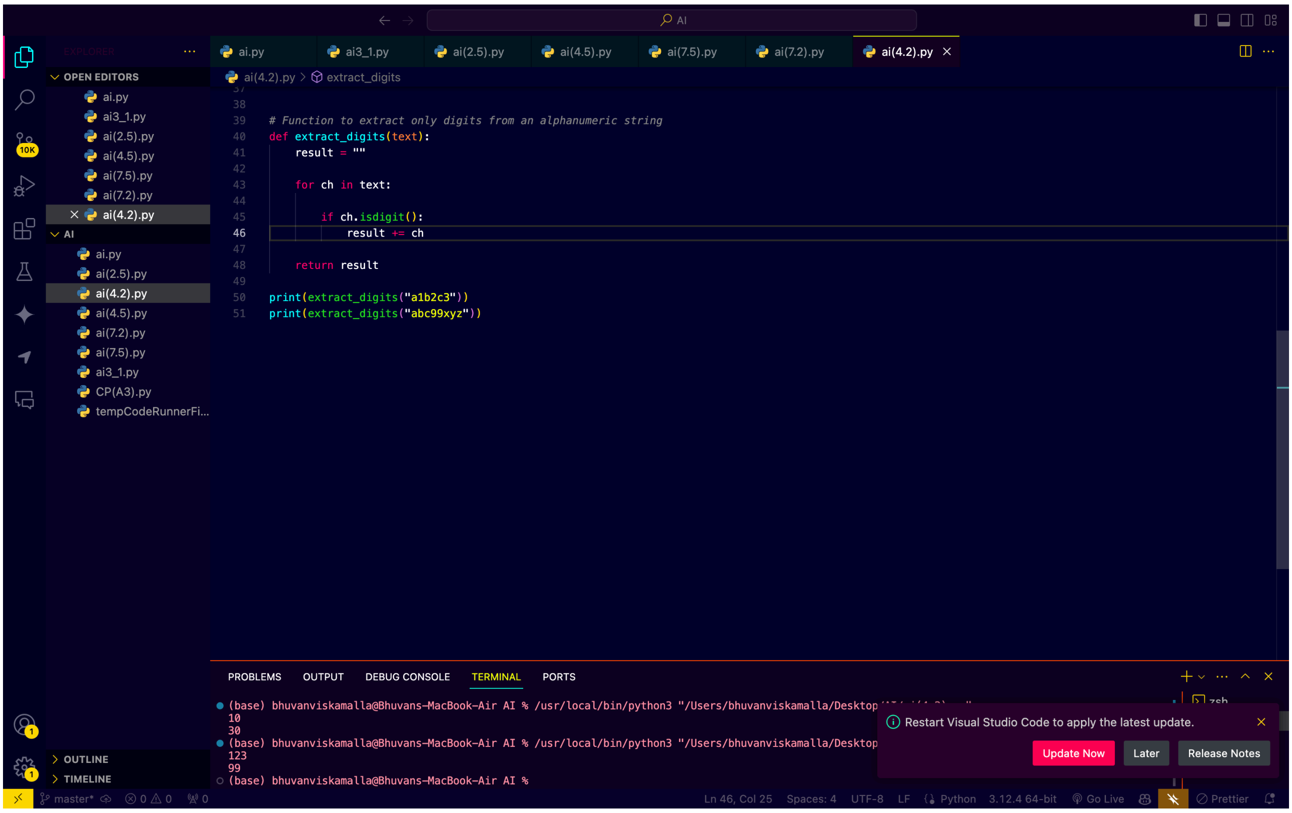
TASK-3

PROMPT: Write a Python function that extracts only digits from an alphanumeric string.

Examples:

Input: "a1b2c3"
Output: "123"
Input: "x9y8z7"
Output: "987"
Input: "abc123def"
Output: “123"

CODE:



```python
# Function to extract only digits from an alphanumeric string
def extract_digits(text):
    result = ""

    for ch in text:

        if ch.isdigit():
            result += ch

    return result

print(extract_digits("a1b2c3"))
print(extract_digits("abc99xyz"))
```

```
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
10
30
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
123
99
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

Restart Visual Studio Code to apply the latest update.

Update Now
Later
Release Notes
Ln 46, Col 25 Spaces: 4 UTF-8 LF Python 3.12.4 64-bit Go Live Prettier

OBSERVATION:

-Multiple examples help the AI model clearly identify the pattern to be learned -The

model focuses only on digit characters and ignores alphabetic content

-The AI demonstrates improved confidence and reduced ambiguity compared to zero
shot and one shot prompting

TASK-4

PROMPT: ZERO-SHOT: Write a Python function that counts the number of vowels in a
string.

FEW-SHOT: Write a Python function that counts the number of vowels in a string.

Examples:

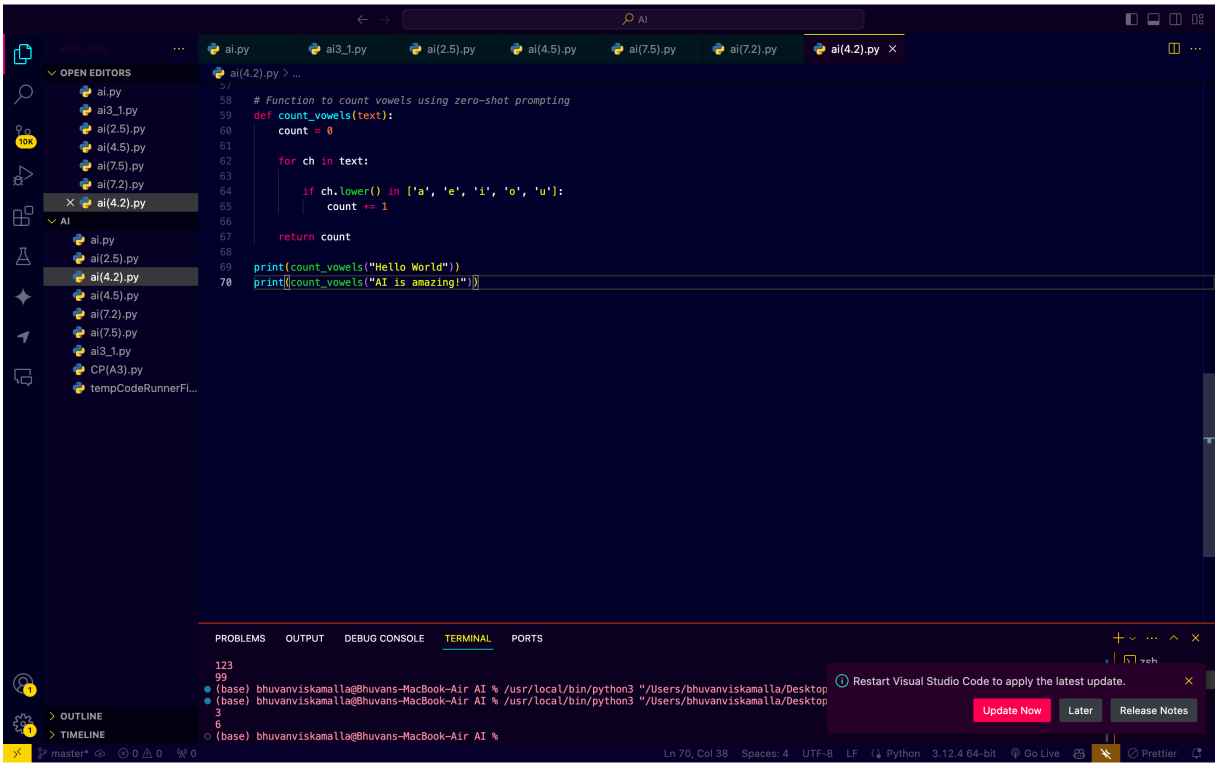
Input: "hello"
Output: 2

Input: "AEIOU"
Output: 5

Input: "chatgpt"
Output: 2

CODE:

ZERO-SHOT:

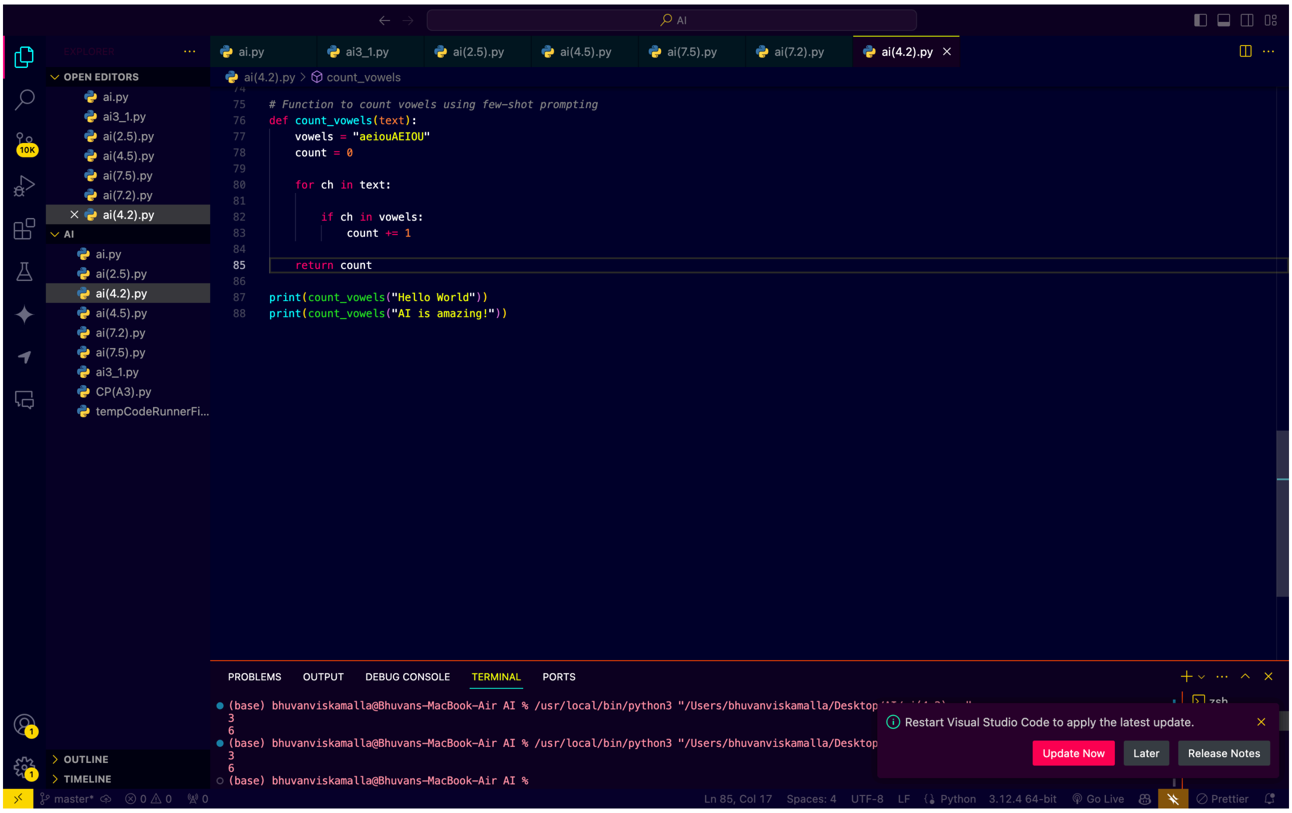


```python
# Function to count vowels using zero-shot prompting
def count_vowels(text):
    count = 0

    for ch in text:

        if ch.lower() in ['a', 'e', 'i', 'o', 'u']:
            count += 1

    return count

print(count_vowels("Hello World"))
print(count_vowels("AI is amazing!"))
```

```
123
99
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
3
6
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

Restart Visual Studio Code to apply the latest update.

FEW-SHOT:



```python
# Function to count vowels using few-shot prompting
def count_vowels(text):
    vowels = "aeiouAEIOU"
    count = 0

    for ch in text:

        if ch in vowels:
            count += 1

    return count

print(count_vowels("Hello World"))
print(count_vowels("AI is amazing!"))
```

```
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
3
6
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
3
6
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

Restart Visual Studio Code to apply the latest update.

```
def count_vowels(text):
vowels = "aeiouAEIOU"
count = 0

for ch in text:
if ch in vowels:
count += 1
return count
```

OBSERVATION:

FEW-SHOT OBSERVATION
The provided examples clearly define what characters should be counted as vowels The
model confidently includes both uppercase and lowercase vowels due to examples

ZERO SHOT:
zero shot prompting the AI guesses the intent based on general knowledge which may
vary for ambiguous tasks

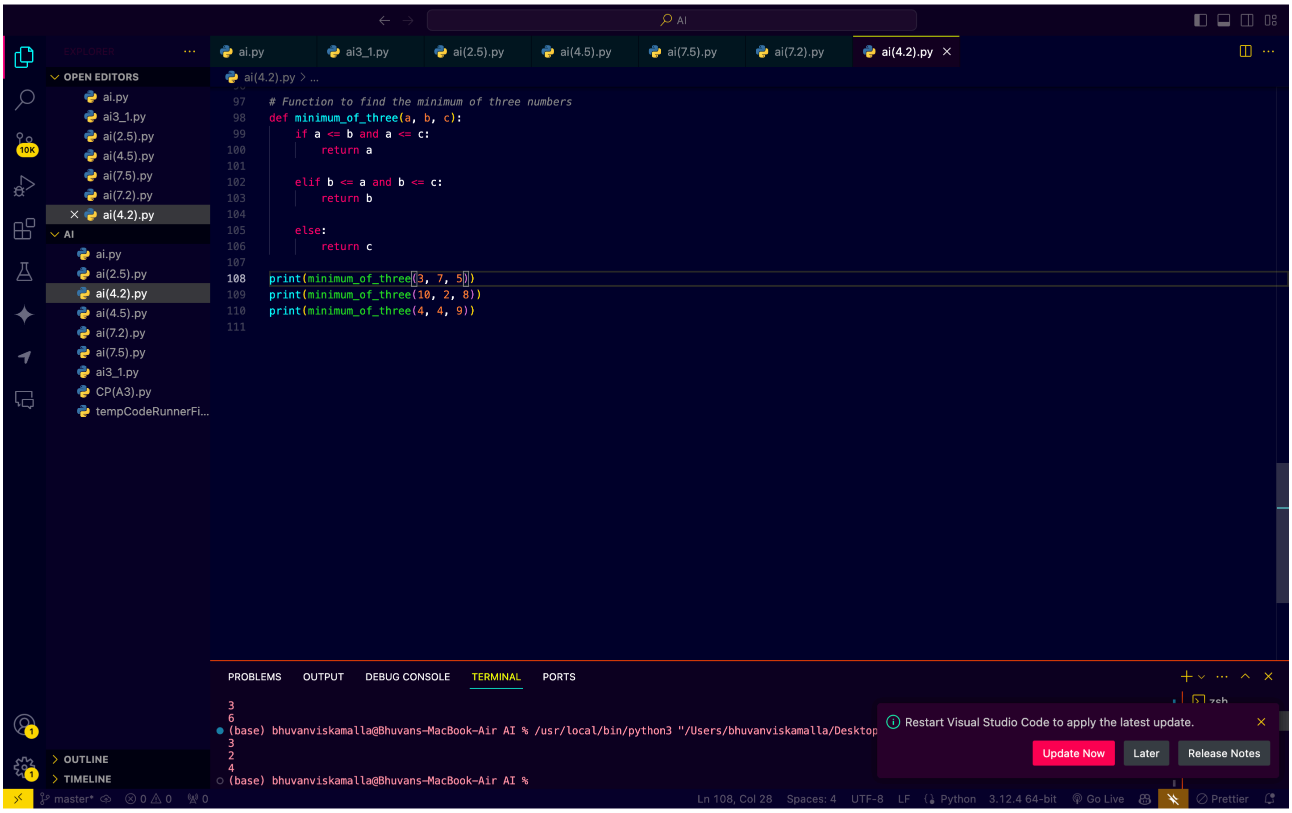
TASK-5

PROMPT:

Write a Python function that determines the minimum of three numbers without using
the built-in min() function.

Examples:
Input: 3, 7, 5
Output: 3
Input: 10, 2, 8
Output: 2
Input: 4, 4, 9
Output: 4

CODE:



ai(4.2).py > ...

```python
# Function to find the minimum of three numbers
def minimum_of_three(a, b, c):
    if a <= b and a <= c:
        return a

    elif b <= a and b <= c:
        return b

    else:
        return c

print(minimum_of_three(3, 7, 5))
print(minimum_of_three(10, 2, 8))
print(minimum_of_three(4, 4, 9))
```

```
3
6
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvanviskamalla/Desktop
3
2
4
(base) bhuvanviskamalla@Bhuvans-MacBook-Air AI %
```

Restart Visual Studio Code to apply the latest update.

OBSERVATION:

The examples clearly establish the comparison pattern needed to identify the smallest value The AI model infers the requirement to handle equality cases correctly

Conditional logic is generated without relying on built-in functions