

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:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists several Python files: ai.py, ai3_1.py, ai(2.5).py, ai(4.5).py, ai(7.5).py, ai(7.2).py, and ai(4.2).py. The ai(4.2).py file is open in the editor, displaying the following code:

```
# 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))
```

The terminal at the bottom shows the output of running the script:

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

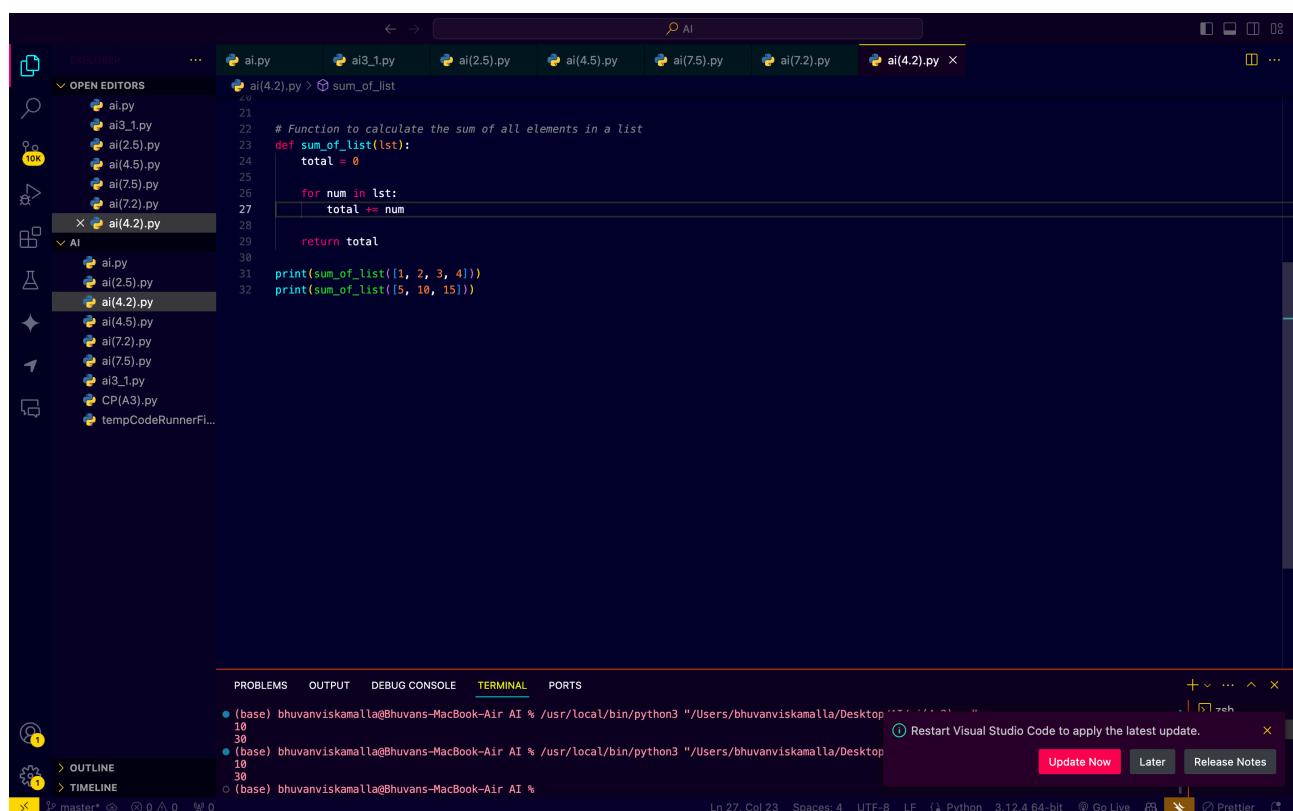
PROMPT: Write a Python function that calculates the sum of elements in a list.

Example:

Input: [1, 2, 3, 4]

Output: 10

CODE:



A screenshot of the Visual Studio Code interface. The Explorer sidebar shows several Python files in the 'OPEN EDITORS' section, including 'ai.py', 'ai(2.5).py', 'ai(4.5).py', 'ai(7.5).py', 'ai(7.2).py', and 'ai(4.2).py'. The 'ai(4.2).py' file is currently open in the main editor area. The code in the editor is as follows:

```
# 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]))
```

The terminal at the bottom shows the output of running the code:

```
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
10
30
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
10
30
```

A notification in the terminal bar says 'Restart Visual Studio Code to apply the latest update.' with buttons for 'Update Now', 'Later', and 'Release Notes'.

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:

```
# 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) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/.../ai(4.2).py
10
30
123
99
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI %
```

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:

The screenshot shows the Visual Studio Code interface with the following details:

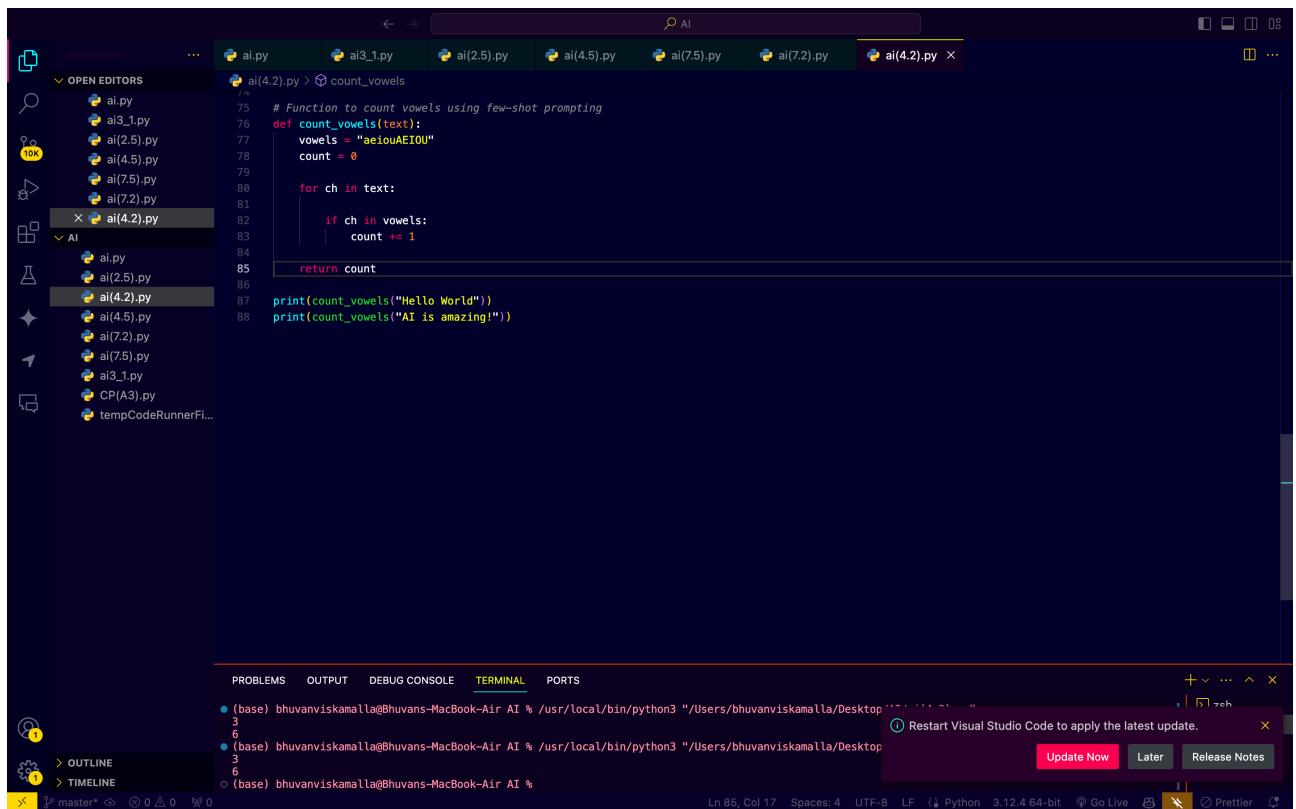
- Explorer View:** Shows multiple Python files in the workspace, including `ai.py`, `ai3_1.py`, `ai(2.5).py`, `ai(4.5).py`, `ai(7.5).py`, `ai(7.2).py`, and `ai(4.2).py`. The file `ai(4.2).py` is currently open in the editor.
- Editor View:** Displays the following Python code:

```
# 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!"))
```
- Terminal View:** Shows the output of running the code:

```
123
99
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
6
6
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI %
```
- Status Bar:** Shows the current file is `ai(4.2).py`, the line is 123, and the column is 99. It also indicates the code is run in Python 3.12.4 64-bit environment.

FEW-SHOT:



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists several Python files in the 'AI' folder. The 'ai(4.2).py' file is currently open in the editor. The code defines a function 'count_vowels' that counts the number of vowels in a given text. The terminal at the bottom shows the output of running the script with two test cases.

```
# 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!"))
```

TERMINAL OUTPUT:

```
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
3
6
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/ai(4.2).py"
3
6
(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI %
```

```
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:

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder structure under "OPEN EDITORS" named "AI". Inside "AI", there are several files: ai.py, ai(2.5).py, ai(4.5).py, ai(7.5).py, ai(7.2).py, and ai(4.2).py. The file "ai(4.2).py" is currently selected.
- Code Editor:** Displays the content of "ai(4.2).py". The code defines a function "minimum_of_three" that takes three parameters (a, b, c) and returns the smallest value. It uses conditional logic to compare the values. The code also includes some test prints at the bottom.
- Terminal:** Shows a command-line session with two entries:
 - 3
 - 6Following these entries is a message from the AI model: "(base) bhuvaniskamalla@Bhuvans-MacBook-Air AI % /usr/local/bin/python3 "/Users/bhuvaniskamalla/Desktop/".
- Bottom Status Bar:** Provides information about the current file: Line 108, Column 28, Spaces: 4, UTF-8, LF, Python 3.12.4 64-bit, Go Live, and Prettier.

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.