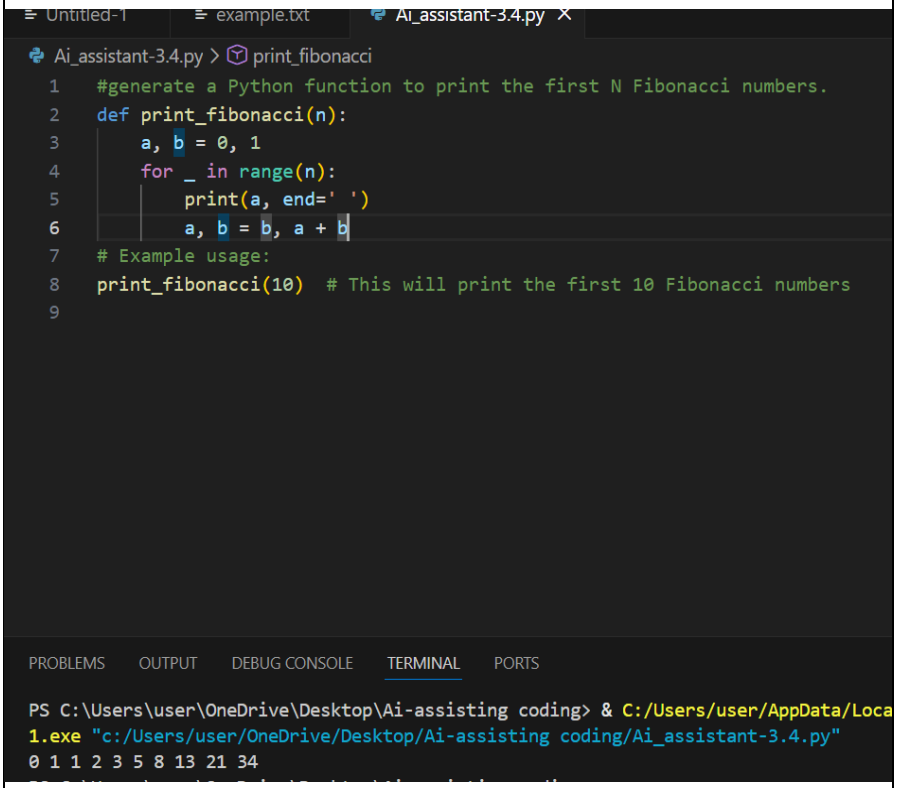
NAME:P.RAKESH
HT.NO:2303A51742

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name	Dr. Rishabh Mittal		
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju		
CourseCode	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/II	**Regulation**	R23
Date and Day of Assignment	**Week2**	**Time(s)**	23CSBTB01 To 23CSBTB52
Duration	2 Hours	**Applicable to Batches**	All batches
Assignment Number: 3.4 (Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*Expected Time to complete*
1	Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and	Week2

	Few-shot Techniques **Task 1: Zero-shot Prompt – Fibonacci Series Generator** **Task Description #1** • Without giving an example, write a single comment prompt asking GitHub Copilot to generate a Python function to print the first N Fibonacci numbers. **Expected Output #1** • A complete Python function generated by Copilot without any example provided. • Correct output for sample input N = 7 → 0 1 1 2 3 5 8 • Observation on how Copilot understood the instruction with zero context.	



Task 2: One-shot Prompt – List Reversal Function

Task Description #2

• Write a comment prompt to reverse a list and provide one example below the comment to guide Copilot.

Expected Output #2

• Copilot-generated function to reverse a list using slicing or loop.

• Output: [3, 2, 1] for input [1, 2, 3]

• Observation on how adding a single example improved Copilot's accuracy.

Ai_assistant-3.4.py > ...

```
#generate a Python code to reverse a list.
def reverse_list(input_list):
    """This function takes a list as input and returns a new list with the elements in reverse order
    return input_list[::-1]
# Example usage:
my_list = [1, 2, 3]
reversed_list = reverse_list(my_list)
print(reversed_list)  # Output: [3, 2, 1]
```

ROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
S C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/pythc
.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
3, 2, 1]
```

Task 3: Few-shot Prompt – String Pattern Matching

Task Description #3

• Write a comment with 2–3 examples to help Copilot understand how to check if a string starts with a capital letter and ends with a period.

Expected Output #3

• A function is_valid() that checks the pattern.

• Output: True or False based on input.

• Students reflect on how multiple examples guide Copilot to generate more accurate code.

Ai_assistant-3.4.py > ...

```python
#generate a Python code ,to check if a string starts with a capital letter and ends with a period.
def check_string(s):
    if s[0].isupper() and s.endswith('.'):
        return True
    return False
# Example usage
test_string = "Hello world."
result = check_string(test_string)
print(f"The string '{test_string}' starts with a capital letter and ends with a period: {result}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
The string 'Hello world.' starts with a capital letter and ends with a period: True
```

Task 4: Zero-shot vs Few-shot – Email Validator

Task Description #4

• First, prompt Copilot to write an email validation function using zero-shot (just the task in comment).

• Then, rewrite the prompt using few-shot examples.

Expected Output #4

• Compare both outputs:

Zero-shot may result in basic or generic validation.

Few-shot gives detailed and specific logic (e.g., @ and domain checking).

• Submit both code versions and note how few-shot improves reliability.

```
Ai_assistant-3.4.py > ...
1   #generate a python code ,to write an email validation function using zero-shot (just the task in comm
2   def is_valid_email(email):
3       if email.count("@") != 1:
4           return False
5
6       local, domain = email.split("@")
7       if not local or "." not in domain:
8           return False
9
10      return True
11  # Test the is_valid_email function
12  test_emails = ["user@example.com", "userexample.com", "user@com"]
13  for email in test_emails:
ROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   Python
S C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/python3
.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
he string 'Hello world.' starts with a capital letter and ends with a period: True
S C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/python3
.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
s_valid_email("user@example.com") → True
s_valid_email("userexample.com") → False
s_valid_email("user@com") → False
```

Task 5: Prompt Tuning – Summing Digits of a Number

Task Description #5

• Experiment with 2 different prompt styles to generate a function that returns the sum of digits of a number.

Style 1: Generic task prompt

Style 2: Task + Input/Output example

Expected Output #5

• Two versions of the sum_of_digits() function.

• Example Output: sum_of_digits(123) → 6

• Short analysis: which prompt produced cleaner or more optimized code and why?

Ai_assistant-3.4.py > ...

```python
#to generate a function that returns the sum of digits of a number with generic task.
def sum_of_digits(number):
    # Convert the number to string to iterate over each digit
    digit_str = str(abs(number))  # Use abs to handle negative numbers
    digit_sum = 0

    # Iterate over each character in the string representation of the number
    for digit in digit_str:
        digit_sum += int(digit)  # Convert character back to integer and add to sum

    return digit_sum
# Example usage:
result = sum_of_digits(-12345)
print("The sum of digits is:", result)  # Output: The sum of digits is: 15

# to generate a function that returns the sum of digits of a number with task input +output.
def sum_of_digits_with_io(number):
    """
    This function takes an integer input and returns the sum of its digits.
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/p
.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
The sum of digits is: 15
```

Ai_assistant-3.4.py > ...

```python
# to generate a function that returns the sum of digits of a number with task input +output.
def sum_of_digits_with_io(number):
    """
    This function takes an integer input and returns the sum of its digits.

    Input:
    number (int): The number whose digits will be summed.

    Output:
    int: The sum of the
    """
    digit_str = str(abs(number))  # Use abs to handle negative numbers
    digit_sum = sum(int(digit) for digit in digit_str)  # Sum the digits using a generator expressio

    return digit_sum
# Example usage:
result = sum_of_digits_with_io(6789)
print("The sum of digits is:", result)  # Output: The sum of digits is: 30
```

(parameter) number: Any
number (int): The number whose digits will be summed.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/pytho
1.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
The sum of digits is: 15
PS C:\Users\user\OneDrive\Desktop\Ai-assisting coding> & C:/Users/user/AppData/Local/Microsoft/WindowsApps/pytho
1.exe "c:/Users/user/OneDrive/Desktop/Ai-assisting coding/Ai_assistant-3.4.py"
The sum of digits is: 15
The sum of digits is: 30
```

	Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots	