SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: M.Tech. and MCA		**Assignment Type: Lab**	**AcademicYear:**2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Course Code		**Course Title**	AI Assisted Problem Solving Using Python
Year/Sem	I/I	**Regulation**	R24
Date and Day of Assignment	Week3 - Monday	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	M.Tech. and MCA
AssignmentNumber:4.3(Present assignment number)/**24**(Total number of assignments)			

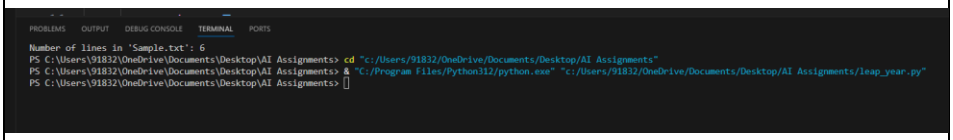
Q.No.	Question	*Expected Time to complete*
1	Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques **Lab Objectives:** ● To explore and apply different levels of prompt examples in AI-assisted code generation. ● To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality. ● To evaluate the impact of context richness and example quantity on AI performance. ● To build awareness of prompt strategy effectiveness for different problem types. **Lab Outcomes (LOs):** After completing this lab, students will be able to: ● Use zero-shot prompting to instruct AI with minimal context. ● Use one-shot prompting with a single example to guide AI code generation. ● Apply few-shot prompting using multiple examples to improve AI responses. ● Compare AI outputs across the three prompting strategies. **Task Description#1** ● Zero-shot: Prompt AI to write a function that checks whether a given year is a leap year.	Week3 - Monday

leap_year.py > is_leap_year

```python
# Prompt: Write a Python function that checks whether a given year is a leap year.
def is_leap_year(year: int) -> bool:
    """Return True if the given year is a leap year, False otherwise."""
    if year % 4 != 0:
        return False
    elif year % 100 != 0:
        return True
    elif year % 400 != 0:
        return False
    else:
        return True
```

Expected Output#1

- AI-generated function with no examples provided



Task Description#2

- One-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches.

one_shot_cm_to_inches.py > ...

```python
# Prompt: Write a Python function that converts centimeters to inches.
# Example: 10 cm → 3.937 inches.
def cm_to_inches(cm: float) -> float:
    """Convert centimeters to inches."""
    return cm / 2.54
# Ask user for input
cm_value = float(input("Enter length in centimeters: "))
inches_value = cm_to_inches(cm_value)
# Call the function and display the result
inches = cm_to_inches(cm_value)
print(f"{cm_value} cm = {inches:.3f} inches")


```

Expected Output#2

- Function with correct conversion logic

```
one_shot_cm_to_inches.py"
Enter length in centimeters: 25
25.0 cm = 9.843 inches
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments>
```

Task Description#3

- Few-shot: Provide 2–3 examples to generate a function that formats full names as "Last, First".

Task-4.3.py > ...

```python
# Write a Python function that formats full names as "Last,
def format_name(first: str, last: str) -> str:
    """Format the full name as 'Last, First'."""
    return f"{last}, {first}"
# Examples:
# Input: "John Smith" → Output: "Smith, John"
# Input: "Jane Doe" → Output: "Doe, Jane"
# Test the function with example inputs
first_name = "John"
last_name = "Smith"
formatted_name = format_name(first_name, last_name)
print(formatted_name)  # Output: "Smith, John"
first_name = "Jane"
last_name = "Doe"
formatted_name = format_name(first_name, last_name)
print(formatted_name)  # Output: "Doe, Jane"
#ask user for input
first_name = input("Enter first name: ")
last_name = input("Enter last name: ")
formatted_name = format_name(first_name, last_name)
print("Formatted name:", formatted_name)

```

Expected Output#3

- Well-structured function respecting the examples

```
e/Documents/Desktop/AI Assignments/Task-4.3.py"
Smith, John
Doe, Jane
Enter first name: viquar
Enter last name: fathima
Formatted name: fathima, viquar
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments>
```

Task Description#4

- Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string.

Task-4.4.py > ...

```python
# -------------------- ZERO-SHOT PROMPT --------------------
# Prompt: Write a Python function that counts the number of vowels in a string.
def count_vowels(s: str) -> int:
    """Return the number of vowels in the input string."""
    vowels = "aeiouAEIOU"
    return sum(1 for char in s if char in vowels)
# Example usage:
input_string = "Hello, World!"
vowel_count = count_vowels(input_string)
print(f"Number of vowels in '{input_string}': {vowel_count}")
# -------------------- END OF ZERO-SHOT PROMPT --------------------
# -------------------- FEW-SHOT PROMPT --------------------
# Prompt: Write a Python function that counts the number of vowels in a string.
def count_vowels(s: str) -> int:
    """Return the number of vowels in the input string."""
    vowels = "aeiouAEIOU"
    return sum(1 for char in s if char in vowels)
# Example usage:
input_string ="pythonic"
vowel_count = count_vowels(input_string)
print(f"Number of vowels in '{input_string}': {vowel_count}")
# -------------------- END OF FEW-SHOT PROMPT --------------------
```

Expected Output#4

- Functional output and comparative reflection

```
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments> & "C:/Program Files/Python312/python.exe" "c:/Users/91832/OneDrive/Documents/AI Assignments/Task-4.4.py"
Number of vowels in 'Hello, World!': 3
Number of vowels in 'pythonic': 2
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments>
```

Task Description#5

- Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.

Task-4.5.py > ...

```python
# Write a Python function that reads a .txt file and returns the number o
def count_lines_in_file(file_path: str) -> int:
    """Return the number of lines in the specified text file."""
    with open(file_path, 'r') as file:
        lines = file.readlines()
    return len(lines)
# Example usage:
file_path = 'Sample.txt'
line_count = count_lines_in_file(file_path)
print(f"Number of lines in '{file_path}': {line_count}")

```

Sample.txt

```
1 Hello world
2 This is a test file.
3 This is viquar fathima
4 I am pursuing Mtech in SR UNIVERSITY
5
6
```

Expected Output#5

- Working file-processing function with AI-guided logic

```
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments> cd "c:/Users/91832/OneDrive/Documents/Desktop/AI Assignments"
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments> & "C:/Program Files/Python312/python.exe" "c:/Users/91832/OneDrive/Documents/Desktop/AI Assignm
Number of lines in 'Sample.txt': 5
PS C:\Users\91832\OneDrive\Documents\Desktop\AI Assignments>
```

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

Evaluation Criteria:

Criteria	Max Marks
Zero Shot (Task #1)	2.5
One Shot (Task#2)	2.5
Few Shot (Task#3 & Task #5)	2.5
Comparison (Task#4)	2.5
Total	**10 Marks**