AI Assisted Coding

Week2 – Wednesday

Name: Lalu prasad Aroori

Batch: 07

Roll No.: 2303A51948

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

Week2 - Wednesday

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 different prompting strategies

Task 1: Zero-Shot Prompting – Leap Year Check

Scenario

Zero-shot prompting involves giving instructions without providing examples.

Task Description

Use zero-shot prompting to instruct an AI tool to generate a Python function that:

• Accepts a year as input

• Checks whether the given year is a leap year

• Returns an appropriate result

Note: No input-output examples should be provided in the prompt.

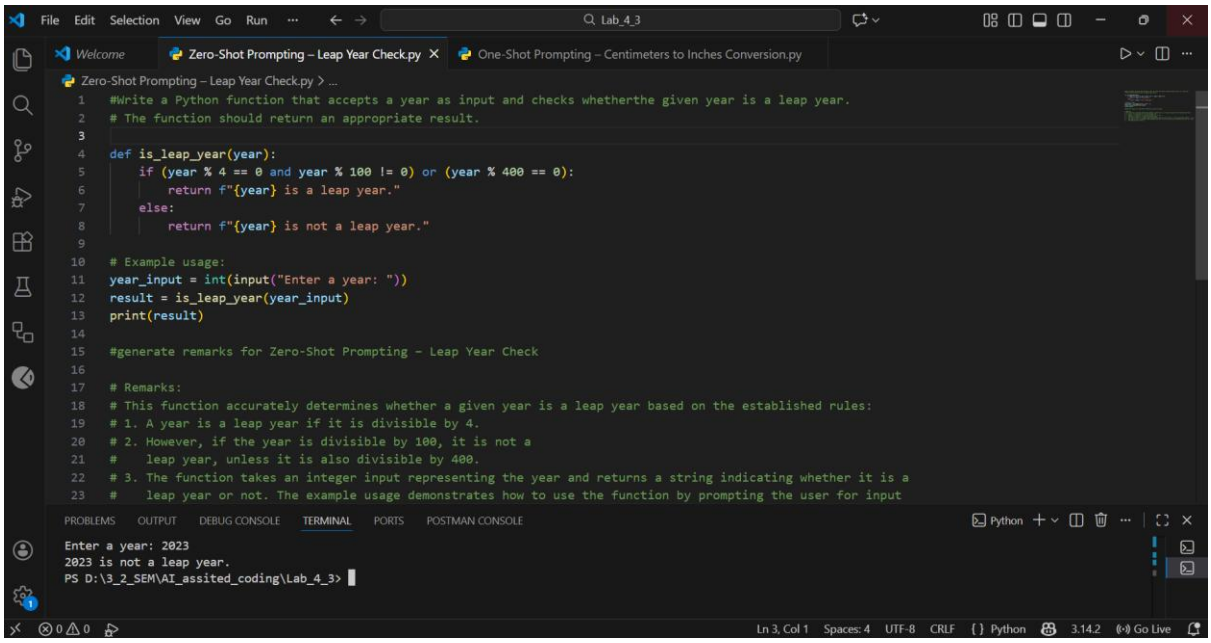
Expected Output

• AI-generated leap year checking function

• Correct logical conditions

• Sample input and output

• Screenshot of AI-generated response (if required)



Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

Use one-shot prompting by providing one input-output example to generate a Python

function that:

• Converts centimeters to inches

• Uses the correct mathematical formula

Example provided in prompt:

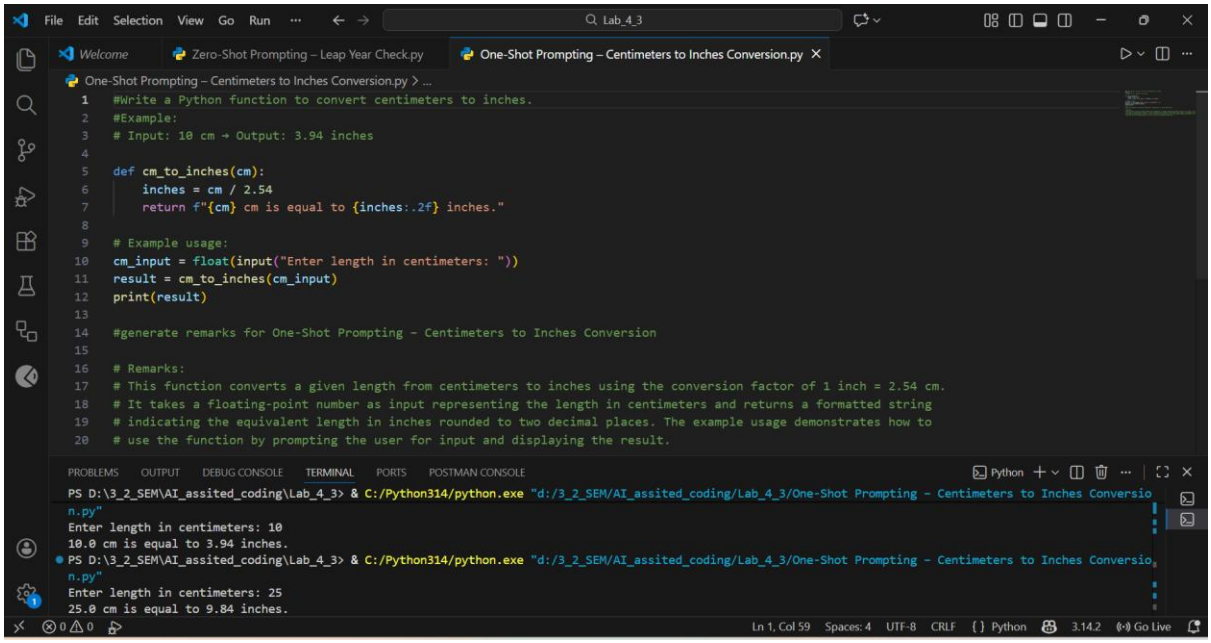
Input: 10 cm → Output: 3.94 inches

Expected Output

• Python function with correct conversion logic

• Accurate calculation

• Sample test cases and outputs



Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

• Accepts a full name as input

• Formats it as “Last, First”

Example formats:

• "John Smith" → "Smith, John"

• "Anita Rao" → "Rao, Anita"

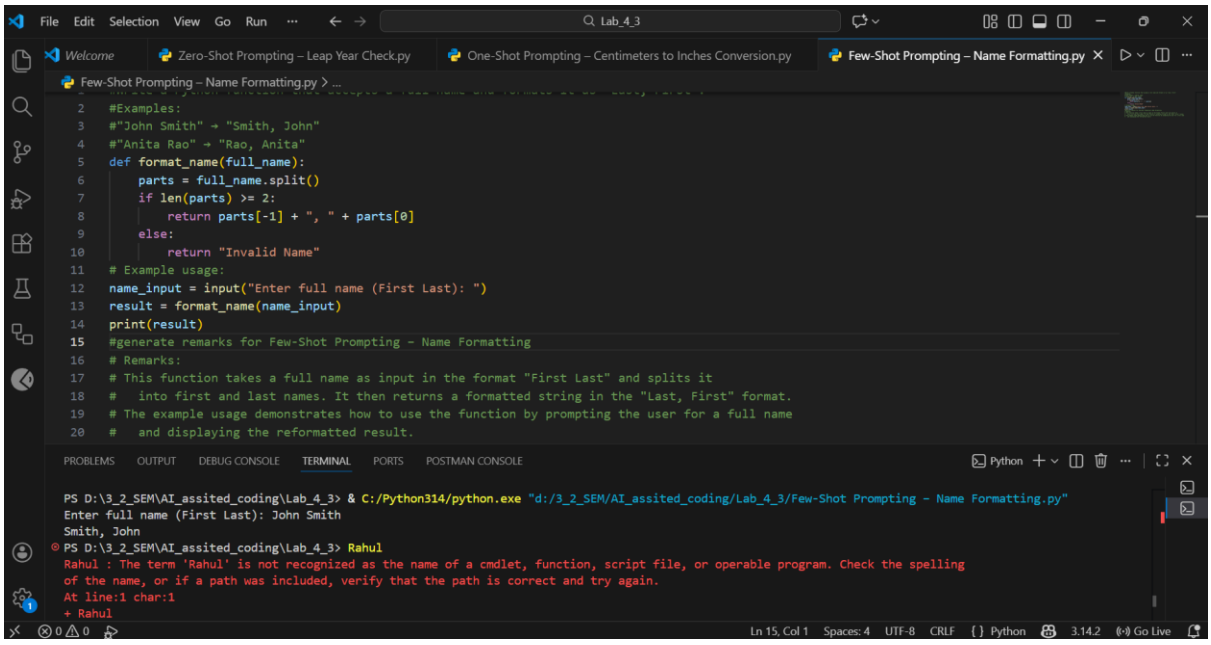
Expected Output

• Well-structured Python function

• Output strictly following example patterns

• Correct handling of names

• Sample inputs and outputs



Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

Scenario

Different prompt strategies may produce different code quality.

Task Description

• Use zero-shot prompting to generate a function that counts vowels in a string

• Use few-shot prompting for the same problem

• Compare both outputs based on:

o Accuracy

o Readability

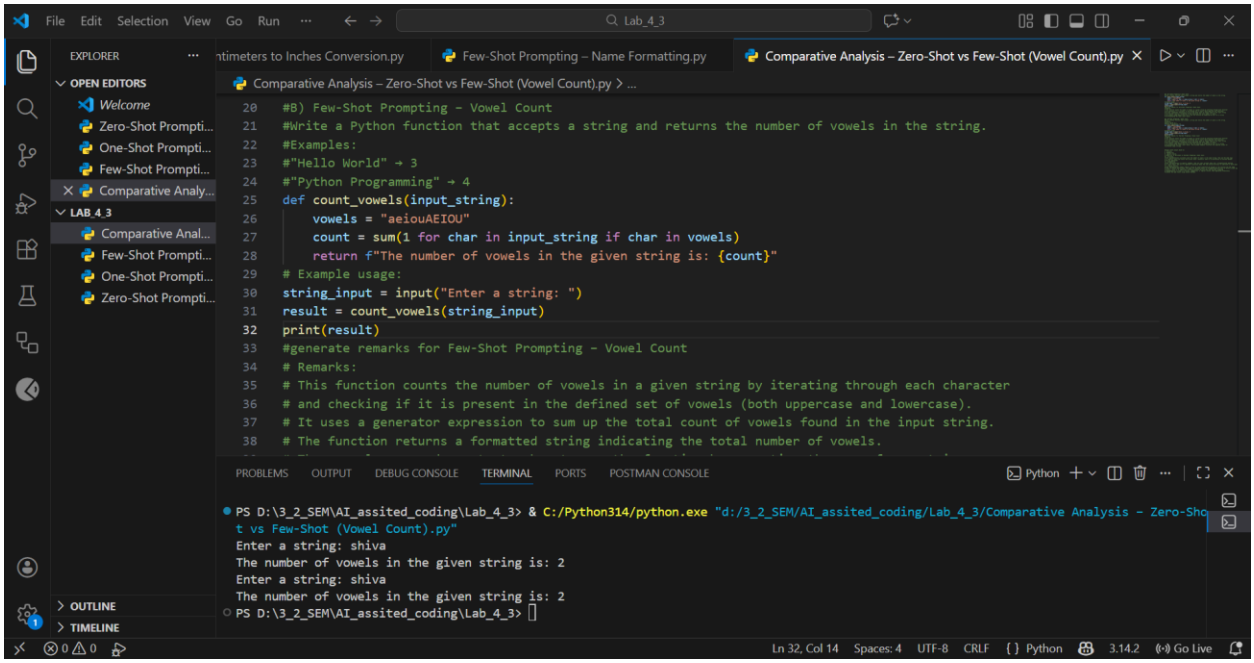
o Logical clarity

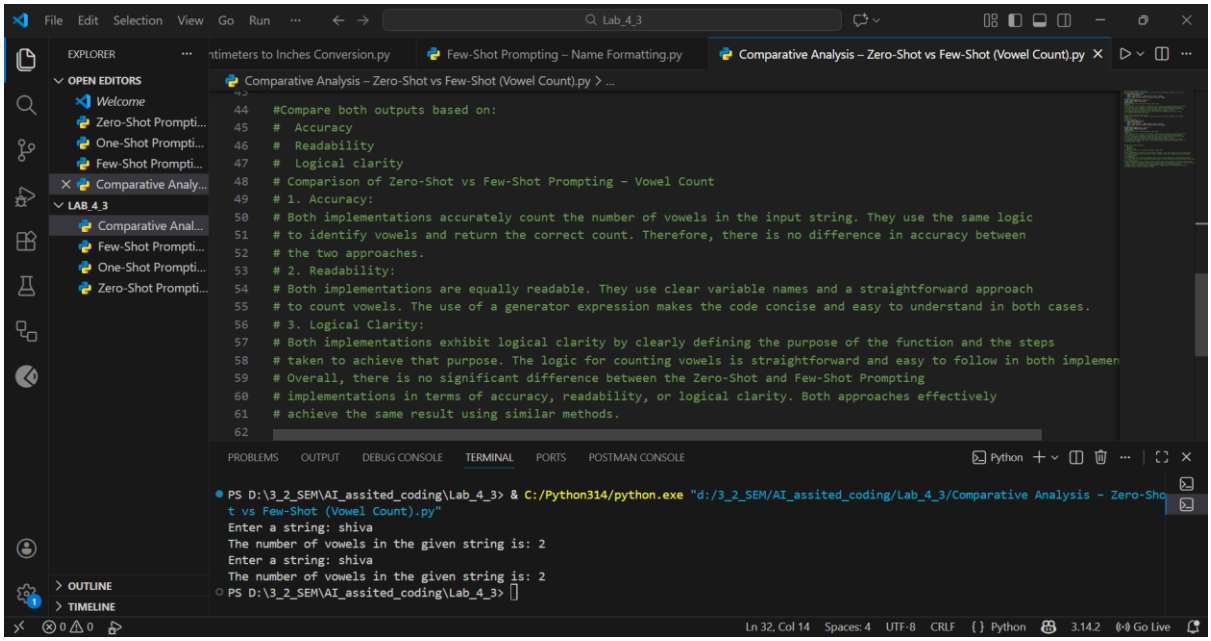
Expected Output

• Two vowel-counting functions

• Comparison table or short reflection paragraph

• Conclusion on prompt effectiveness





Task 5: Few-Shot Prompting – File Handling

Scenario

File processing requires clear logical understanding.

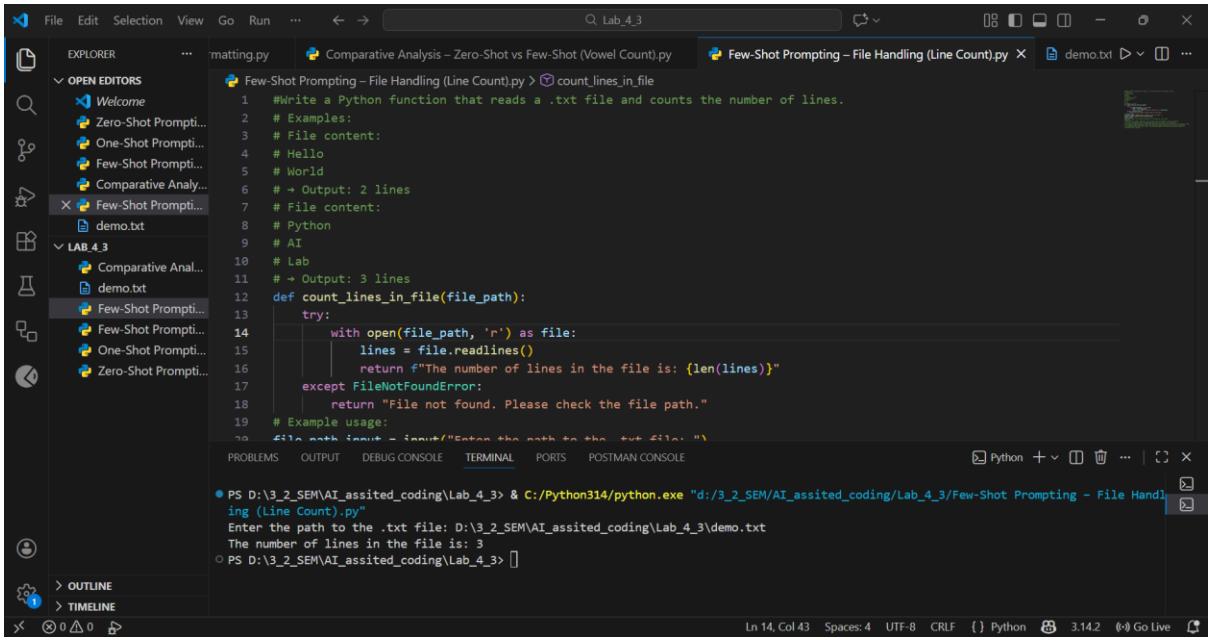
Task Description

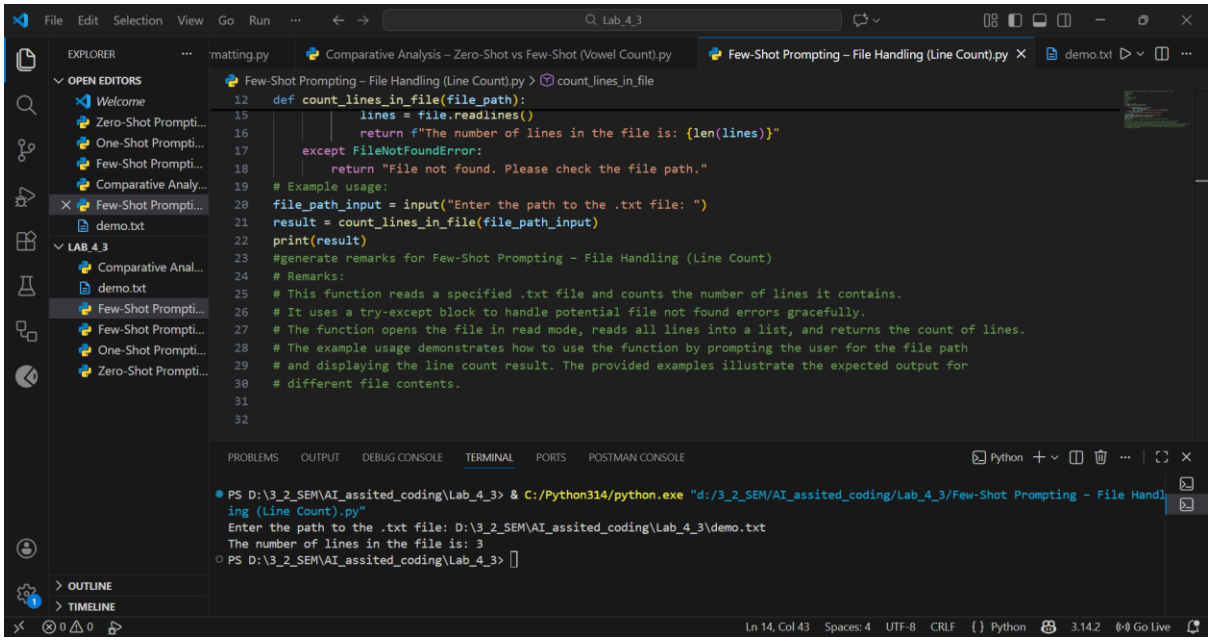
Use few-shot prompting to generate a Python function that:

- Reads a .txt file
- Counts the number of lines in the file
- Returns the line count

Expected Output

- Working Python file-processing function
- Correct line count
- Sample .txt input and output
- AI-assisted logic explanation


File Edit Selection View Go Run
Lab_4_3
EXPLORER
matting.py
Comparative Analysis – Zero-Shot vs Few-Shot (Vowel Count).py
Few-Shot Prompting – File Handling (Line Count).py
demo.txt
OPEN EDITORS
Welcome
Zero-Shot Prompti...
One-Shot Prompti...
Few-Shot Prompti...
Comparative Analy...
Few-Shot Prompti...
demo.txt
LAB_4_3
Comparative Anal...
demo.txt
Few-Shot Prompti...
Few-Shot Prompti...
One-Shot Prompti...
Zero-Shot Prompti...
Few-Shot Prompting – File Handling (Line Count).py > count_lines_in_file
#Write a Python function that reads a .txt file and counts the number of lines.
# Examples:
# File content:
# Hello
# World
# → Output: 2 lines
# File content:
# Python
# AI
# Lab
# → Output: 3 lines
def count_lines_in_file(file_path):
try:
with open(file_path, 'r') as file:
lines = file.readlines()
return f"The number of lines in the file is: {len(lines)}"
except FileNotFoundError:
return "File not found. Please check the file path."
Example usage:
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
Python
PS D:\3_2_SEM\AI_assited_coding\Lab_4_3> & C:/Python314/python.exe "d:/3_2_SEM/AI_assited_coding/Lab_4_3/Few-Shot Prompting – File Handling (Line Count).py"
Enter the path to the .txt file: D:\3_2_SEM\AI_assited_coding\Lab_4_3\demo.txt
The number of lines in the file is: 3
PS D:\3_2_SEM\AI_assited_coding\Lab_4_3>
OUTLINE
TIMELINE
Ln 14, Col 43 Spaces: 4 UTF-8 CRLF { } Python 3.14.2 Go Live


File Edit Selection View Go Run
Lab_4_3
EXPLORER
matting.py
Comparative Analysis – Zero-Shot vs Few-Shot (Vowel Count).py
Few-Shot Prompting – File Handling (Line Count).py
demo.txt
OPEN EDITORS
Welcome
Zero-Shot Prompti...
One-Shot Prompti...
Few-Shot Prompti...
Comparative Analy...
Few-Shot Prompti...
demo.txt
LAB_4_3
Comparative Anal...
demo.txt
Few-Shot Prompti...
Few-Shot Prompti...
One-Shot Prompti...
Zero-Shot Prompti...
Few-Shot Prompting – File Handling (Line Count).py > count_lines_in_file
def count_lines_in_file(file_path):
lines = file.readlines()
return f"The number of lines in the file is: {len(lines)}"
except FileNotFoundError:
return "File not found. Please check the file path."
# Example usage:
file_path_input = input("Enter the path to the .txt file: ")
result = count_lines_in_file(file_path_input)
print(result)
#generate remarks for Few-Shot Prompting – File Handling (Line Count)
Remarks:
This function reads a specified .txt file and counts the number of lines it contains.
It uses a try-except block to handle potential file not found errors gracefully.
The function opens the file in read mode, reads all lines into a list, and returns the count of lines.
The example usage demonstrates how to use the function by prompting the user for the file path
and displaying the line count result. The provided examples illustrate the expected output for
different file contents.
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
Python
PS D:\3_2_SEM\AI_assited_coding\Lab_4_3> & C:/Python314/python.exe "d:/3_2_SEM/AI_assited_coding/Lab_4_3/Few-Shot Prompting – File Handling (Line Count).py"
Enter the path to the .txt file: D:\3_2_SEM\AI_assited_coding\Lab_4_3\demo.txt
The number of lines in the file is: 3
PS D:\3_2_SEM\AI_assited_coding\Lab_4_3>
OUTLINE
TIMELINE
Ln 14, Col 43 Spaces: 4 UTF-8 CRLF { } Python 3.14.2 Go Live