SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName:B. Tech		**Assignment Type: Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
Instructor(s)Name		Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)	
CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week2 - Wednesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	
AssignmentNumber:4.3(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*ExpectedTime 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	Week2 - Wednesday

generation.

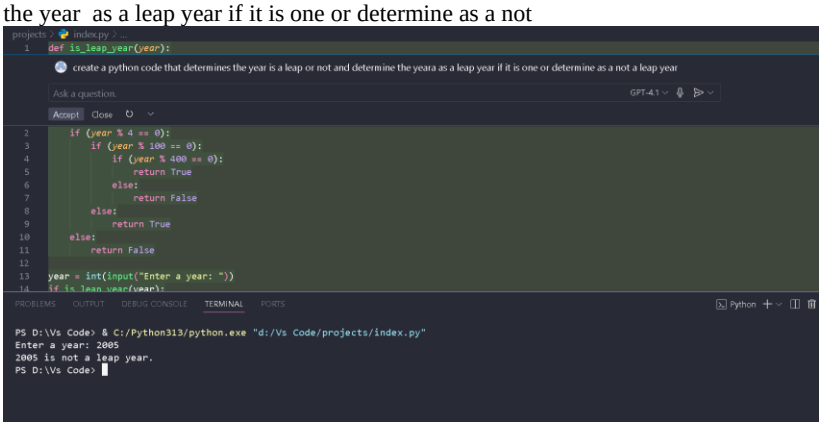
- To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality.
- To evaluate the impact of context richness and example quantity on AI performance.
- To build awareness of prompt strategy effectiveness for different problem types.

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
- **Prompt**.: create a python code that determines the year is a leap or not and determine the year as a leap year if it is one or determine as a not



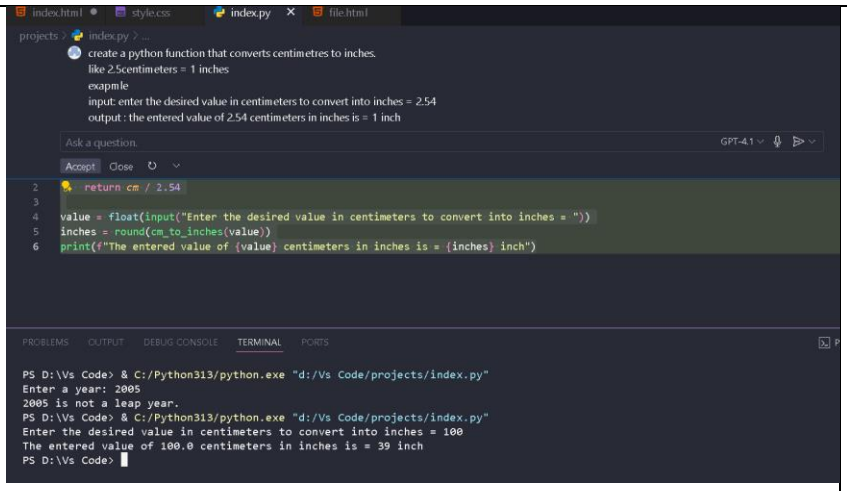
- **EXPLAINATION:** This code checks if a year is a leap year:
- It defines a function is_leap_year(year) that returns True if the year is a leap year, otherwise False.
- It asks the user to enter a year.
- It prints whether the entered year is a leap year or not.
- A leap year is:
- Divisible by 4,
- But if it's divisible by 100, it must also be divisible by 400.

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.
- **PROMPT**: create a python function that converts centimetres to inches.
  like 2.5centimeters = 1 inches
  exapmle
  input: enter the desired value in centimeters to convert into inches = 2.54
  output : the entered value of 2.54 centimeters in inches is = 1 inch
-
-



-

Explaination: This code converts centimeters to inches:

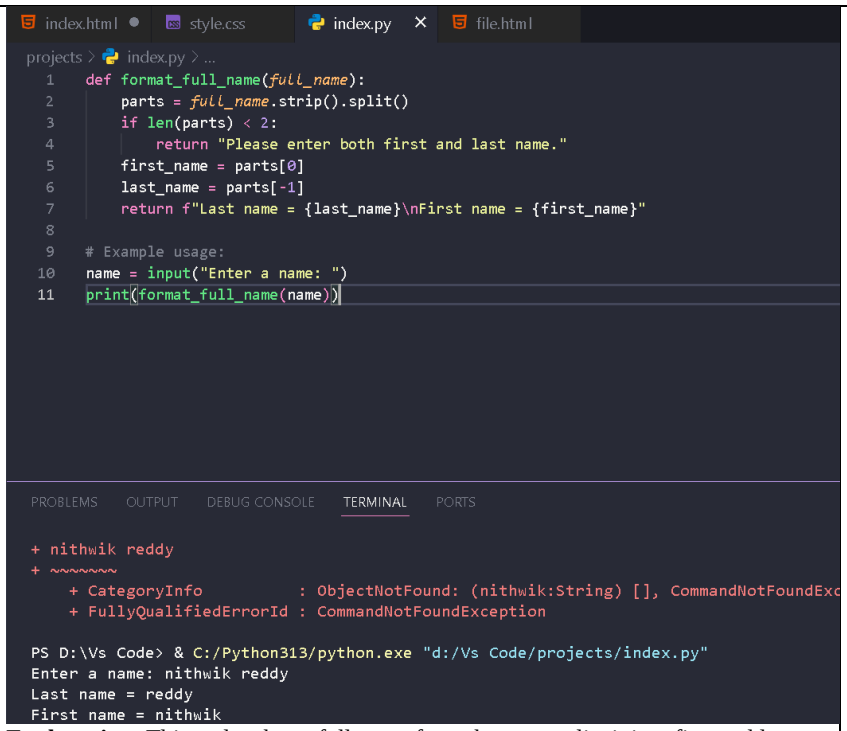
- It defines a function to change centimeters into inches.
- It asks the user to enter a value in centimeters.
- It converts that value to inches and rounds it.
- It prints the result.

Expected Output#2

- Function with correct conversion logic

Task Description#3

- Few-shot: Provide 2–3 examples to generate a function that formats full names as "Last, First".
- **Prompt:** generate a function that formats full names as "Last, First".
- example 1
- input: enter a name: Rithvik Gona
- output: last name = Gona
- First name = Rithvik
- example 2
- input: enter a name: Shiva Nalumachu
- output: Last name = Nalumachu
- First name = Shiva
- **CODE:**
-



- **Explanation:** This code takes a full name from the user, splits it into first and last names, and prints them separately. If the user doesn't enter both names, it asks for both.

Expected Output#3

- Well-structured function respecting the examples

Task Description#4

- Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string.
- **Prompt:** generate a python function to count the number of the vowels in the input string Code
-

```
projects > index.py > ...
1   def count_vowels(input_string):
2       vowels = "aeiouAEIOU"
3       count = 0
4       for char in input_string:
5           if char in vowels:
6               count += 1
7       return count
8
9   # Example usage:
10  text = input("Enter a string: ")
11  print(f"Number of vowels in the input string: {count_vowels(text)}")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

First name = nithwik
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
  File "d:\Vs Code\projects\index.py", line 11
    print(f"Number of vowels in the input string:
          ^
SyntaxError: unterminated f-string literal (detected at line 11)
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Enter a string: hero
Number of vowels in the input string: 2
```

- **Explaination:** This code counts how many vowels are in a given string and prints the total.

Expected Output#4

- Functional output and comparative reflection

Task Description#5

- Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.
- **Prompt: :** generate a Python function that reads a .txt file and returns the number of lines
  Example 1
  Input: A text file named example.txt with the following content:
  Line 1
  Line 2
  Line 3
  Expected Output:3
  Example 2
  Input: A non-existent text file named nonexistent.txt
  Expected Output:
  0 (with an appropriate error message indicating the file was not found)
  Code:

projects > index.py > ...

```python
def count_lines_in_file(filename):
    try:
        with open(filename, 'r', encoding='utf-8') as file:
            return sum(1 for _ in file)
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return 0

# Example usage:
filename = input("Enter the filename: ")
lines = count_lines_in_file(filename)
print(f"Number of lines in the file: {lines}")
```

Explaination: This code counts the number of lines in a text file.
If the file doesn't exist, it prints an error and returns 0.
It asks the user for a filename and shows the line count.

Expected Output#5

- Working file-processing function with AI-guided logic

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)	0.5
One Shot (Task#2)	0.5
Few Shot (Task#3 & Task #5)	1.0
Comparison (Task#4)	0.5
Total	**2.5 Marks**