Ai Assistant Coding

Week:4.3

Ht.No: 2303A52492

Batch:50

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)

Prompt:-

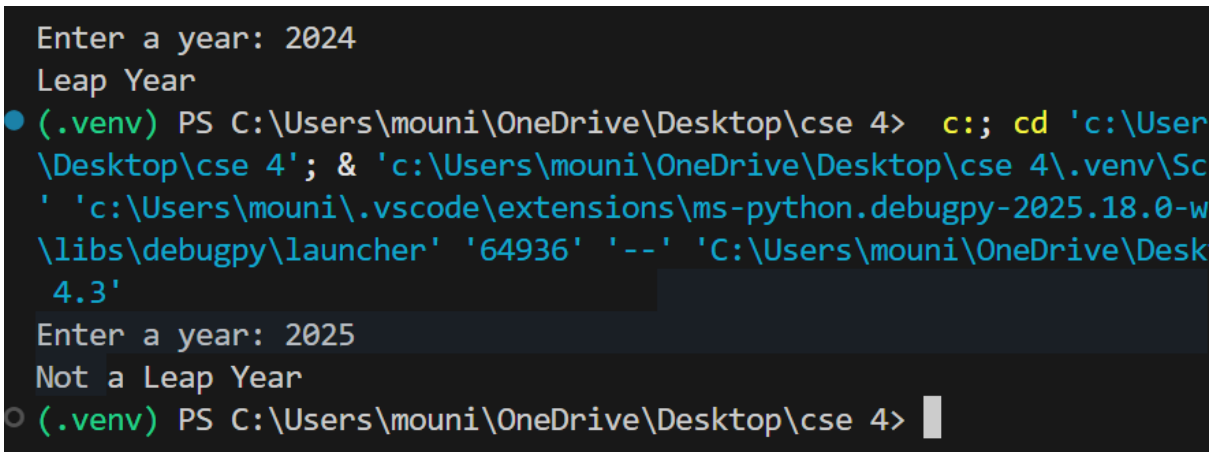
Write a Python program where you need to enter an input as year and should check whether it is a leap year or non leap year.

Code:-

```
def is_leap_year(year):
```

```
    if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
        return "Leap Year"
    else:
        return "Not a Leap Year"
year = int(input("Enter a year: "))
result = is_leap_year(year)
print(result)
```

Output:-



Approach:-

In this program we can learn that how zero short prompting works in leap year program without giving an example.

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.

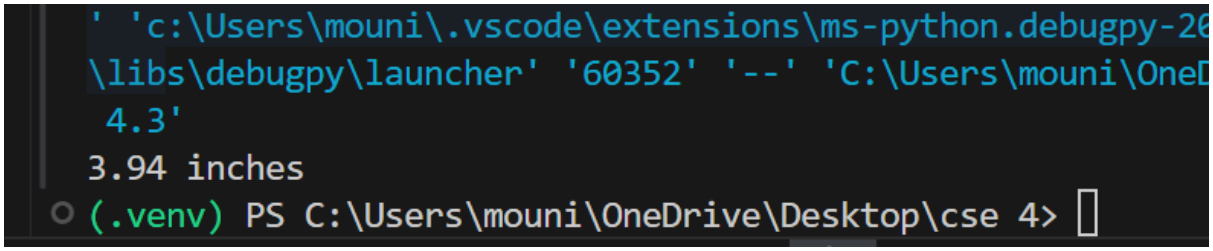
Prompt:-

Write a Python program that converts centimeters to inches.
Example:input should be 10 cm and output should be 3.94 inches.

Code:-

```
cm = 10

inches = cm / 2.54

print(f"{inches:.2f} inches")
```

Output:-



Approach:-

In this python program we can learn that how one Short promting involves by giving an example in the prompt.

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.

Prompt:-

Write a python program for accepting a full name as input and formats it as “Last, First”.Example formats:

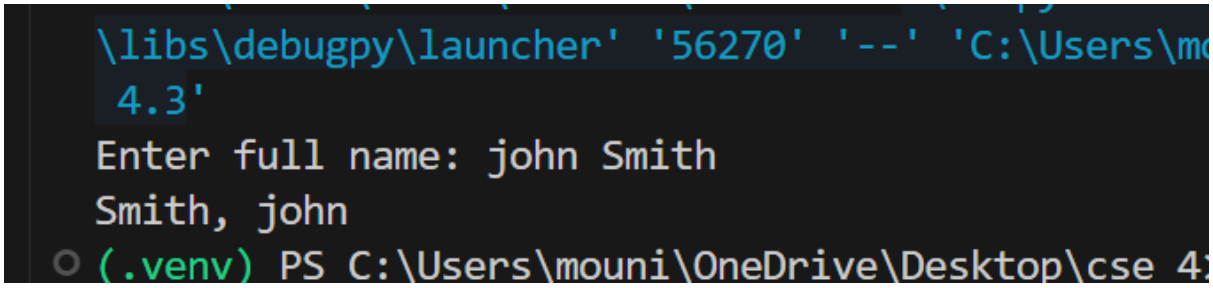
• "John Smith" → "Smith, John"

• "Anita Rao" → "Rao, Anita"

Code:-

```
full_name = input("Enter full name: ")

parts = full_name.split()

first_name = parts[0]

last_name = parts[-1]

formatted_name = f"{last_name}, {first_name}"

print(formatted_name)
```

Output:-



Approach:-

In this python program we can learn how one short prompt works by giving multiple examples.

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

Prompt:-

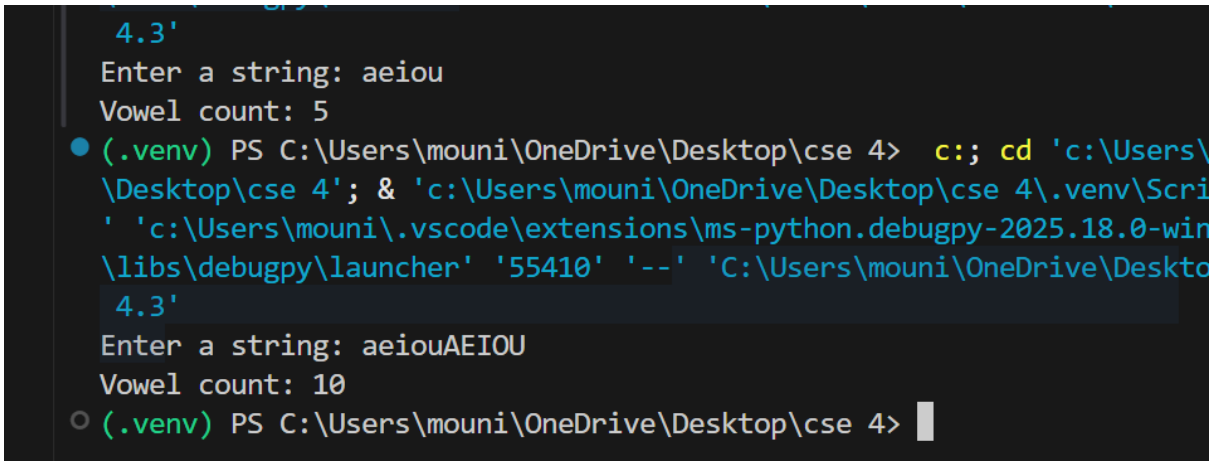
Write a python program that counts vowels in a string and compare both outputs based on Accuracy,Redability and logical clarity.

Code:-

```
text = input("Enter a string: ")

vowels = "aeiouAEIOU"

count = 0


for ch in text:
```

```
    if ch in vowels:
        count += 1
print("Vowel count:", count)
```

Output:-



Approach:-

In this code we can know ,that how to conver vowels to string by using Zero short prompting means without giving any example.

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

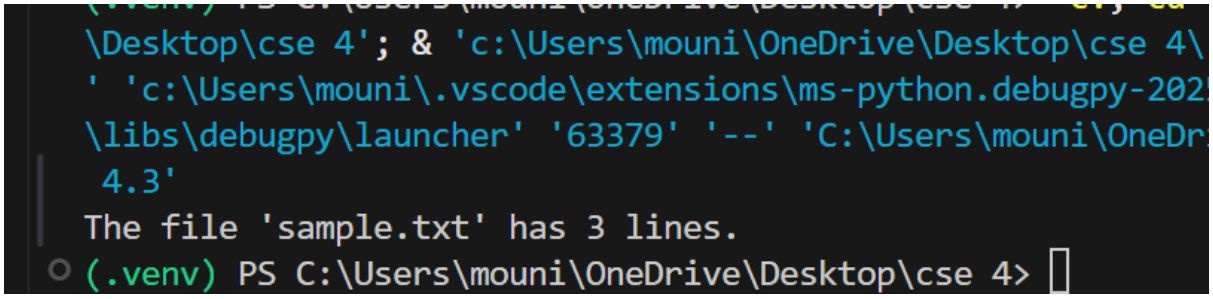
Prompt:-

Write a python program by using few short prompting which Reads a .txt file ,counts the number of lines in the file and returns the line count.

Code:-

```
def count_lines_in_file(file_path):
if __name__ == "__main__":
    sample_file = 'sample.txt'
    with open(sample_file, 'w') as f:
        f.write("Hello, World!\n")
        f.write("This is a sample file.\n")
        f.write("It has multiple lines.\n")
    line_count = count_lines_in_file(sample_file)
    print(f"The file '{sample_file}' has {line_count} lines.")
```

Output:-



Approach:

In this code we can learn how the file is reading by using few short prompting.