

Assignment – 4.3

Name – Rounak Raj

Hall-ticket – 2303A54043

AI Assistant Coding

Task 1: Zero-Shot Prompting – Leap Year Check

Prompt : Generate a Python function that takes a year as input, determines whether it is a leap year using correct logic, returns the result, and include sample input and output

The screenshot shows a Python file named Untitled-1.py in VS Code. The code defines a function is_leap_year that checks if a given year is a leap year according to the rules: divisible by 4, except centuries must be divisible by 400. It includes type checking and sample input/output. The terminal below shows the execution of the code with four sample years (1900, 2000, 2004, 2001) and their corresponding outputs (False, True, True, False).

```
Untitled-1.py > ...
1  def is_leap_year(year: int) -> bool:
2      """Return True if year is a leap year, otherwise False.
3          Rule: divisible by 4, except centuries must be divisible by 400.
4      """
5      if not isinstance(year, int):
6          raise TypeError("year must be an integer")
7      return (year % 4 == 0) and (year % 100 != 0 or year % 400 == 0)
8
9
10 if __name__ == "__main__":
11     # Sample input and output
12     samples = [1900, 2000, 2004, 2001]
13     for y in samples:
14         print(f"Input: {y} -> Output: {is_leap_year(y)}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python Debug Console + ⌂ ⌂ ... | [] ×

```
PS C:\Users\ayans\Downloads\ai assited code\web development\Restro Website Pro> & 'C:\Python314\python.exe' 'c:\Users\ayans\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundle\libs\debugpy\launcher' '29794' '--' 'C:\Users\ayans\Downloads\ai assited code\web development\Restro Website Pro\Untitled-1.py'
Input: 1900 -> Output: False
Input: 2000 -> Output: True
Input: 2004 -> Output: True
Input: 2001 -> Output: False
PS C:\Users\ayans\Downloads\ai assited code\web development\Restro Website Pro>
```

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

Prompt: Generate a Python function that converts centimeters to inches using the correct formula. Example: Input: 10 cm → Output: 3.94 inches. Include sample test cases and outputs.

```
Untitled2.py > ...
1  def cm_to_inches(cm: float, ndigits: int = 2) -> float:
2      """Convert centimeters to inches. Returns value rounded to ndigits decimal places."""
3      return round(cm / 2.54, ndigits)
4
5
6  if __name__ == "__main__":
7      # Sample test cases and outputs
8      examples = [10, 2.54, 0, -5]
9      for cm in examples:
10         inches = cm_to_inches(cm)
11         print(f"Input: {cm} cm → Output: {inches} inches")
12
13     # Expected outputs:
14     # Input: 10 cm → Output: 3.94 inches
15     # Input: 2.54 cm → Output: 1.0 inches
16     # Input: 0 cm → Output: 0.0 inches

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    Python Debug Console + ▾ ⏹ ⏹ ... | [] ×

Input: -5 cm → Output: -1.97 inches
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro> ^C
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro>
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro> c;; cd 'c:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro'; & 'C:\Python314\python.exe' 'c:\Users\ayans\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '38033' '--' 'C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro\Untitled2.py'
Input: 10 cm → Output: 3.94 inches
Input: 2.54 cm → Output: 1.0 inches
Input: 0 cm → Output: 0.0 inches
Input: -5 cm → Output: -1.97 inches
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro>
```

#Task 3: Few-Shot Prompting – Name Formatting

Prompt: Generate a Python function that takes a full name as input and formats it as ‘Last, First’. Examples: ‘John Smith’ → ‘Smith, John’; ‘Anita Rao’ → ‘Rao, Anita’. Include sample inputs and outputs

```
UNTITLED-11.PY > ...
1 def format_name(full_name: str) -> str:
2     """
3         Format a full name as 'Last, First'.
4     Examples:
5         'John Smith' -> 'Smith, John'
6         'Anita Rao' -> 'Rao, Anita'
7         For multi-part names, uses first and last tokens. Returns empty string if
8         """
9     if not full_name or not full_name.strip():
10         return ""
11     parts = full_name.strip().split()
12     if len(parts) == 1:
13         return parts[0]
14     first, last = parts[0], parts[-1]
15     return f"{last}, {first}"
```

```
UNTITLED-11.PY > ...
17
18 # Sample inputs and outputs
19 if __name__ == "__main__":
20     examples = [
21         "John Smith",
22         "Anita Rao",
23         "John Ronald Reuel Tolkien",
24         "Madonna",
25         " Alice Johnson ",
26         ""
27     ]
28     for name in examples:
29         print(f"{name!r} -> {format_name(name)!r}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python Debug Console + ⌂ ⌄ ... | ⌋ X

```
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro>
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro> c;; cd 'c:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro'; & 'C:\Python314\python.exe' 'c:\Users\ayans\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '15703' '--' 'C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro\Untitled-11.py'
'John Smith' -> 'Smith, John'
'Anita Rao' -> 'Rao, Anita'
'John Ronald Reuel Tolkien' -> 'Tolkien, John'
'Madonna' -> 'Madonna'
' Alice Johnson ' -> 'Johnson, Alice'
```

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

Prompt: Using zero-shot prompting, generate a Python function that counts vowels in a string. Then, using few-shot prompting, generate the same function. Compare both implementations based on accuracy, readability, and logical clarity, and conclude which prompting method is more effective

```
from collections import Counter.py > ...
1  from collections import Counter
2  from typing import Dict
3
4  # File: vowel_counter.py
5  # Two implementations of "count vowels in a string":
6  # - zero-shot style: straightforward imperative loop
7  # - few-shot style: uses Python stdlib for concision
8
9
10 def count_vowels_zero_shot(s: str) -> Dict[str, int]:
11     """
12         Count vowels (a,e,i,o,u) in s. Returns dict with counts for each vowel
13         and a 'total' key.
14         Imperative, explicit implementation (zero-shot style).
15     """
16
17     vowels = 'aeiou'
18     counts = {v: 0 for v in vowels}
19     for ch in s:
20         cl = ch.lower()
21         if cl in vowels:
22             counts[cl] += 1
23     counts['total'] = sum(counts.values())
24     return counts
25
26 def count_vowels_few_shot(s: str) -> Dict[str, int]:
27     """
28         Count vowels (a,e,i,o,u) in s. Returns dict with counts for each vowel
```

```

28     and a 'total' key.
29     Concise implementation using collections.Counter (few-shot style).
30     """
31     s_lower = s.lower()
32     c = Counter(ch for ch in s_lower if ch in 'aeiou')
33     counts = {v: c.get(v, 0) for v in 'aeiou'}
34     counts['total'] = sum(counts.values())
35     return counts
36
37 if __name__ == "__main__":
38     samples = [
39         "",
40         "Hello, World!",
41         "AEIOU aeiou",

```

```

from collections import Counter
samples = [
    "",
    "Hello, World!",
    "AEIOU aeiou",
    "Python programming"
]
for s in samples:
    z = count_vowels_zero_shot(s)
    f = count_vowels_few_shot(s)
    assert z == f, f"Mismatch for '{s}': {z} vs {f}"
print("All tests passed.")

n.exe' 'c:\Users\ayans\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64636' '--' 'C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro\from collections import Counter.py'
All tests passed.
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro> ^C
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro>
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro> cd 'c:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro'; & 'C:\Python314\python.exe' 'c:\Users\ayans\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64655' '--' 'C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro\from collections import Counter.py'
All tests passed.
PS C:\Users\ayans\Downloads\ai assisted code\web development\Restro Website Pro>

```

#Task 5: Few-Shot Prompting – File Handling

Prompt: Generate a Python function that reads a .txt file and returns the number of lines.

Example: Input file with 3 lines → Output: 3. Include sample input/output and briefly explain the logic.

```
cp # count_lines.py > ...
1  # count_lines.py
2  # Reads a .txt file and returns the number of lines.
3  # Example:
4  # - Create "example.txt" with 3 lines -> count_lines("example.txt") returns
5  # Logic: open the file and sum 1 for each line (memory-efficient).
6
7  def count_lines(file_path):
8      """Return the number of lines in the given text file."""
9      with open(file_path, 'r', encoding='utf-8') as f:
10          return sum(1 for _ in f)
11
12 if __name__ == "__main__":
13     # Sample input: create a file with 3 lines
14     sample = "example.txt"
```

```
# count_lines.py > ...
7  def count_lines(file_path):
8      with open(file_path, 'r', encoding='utf-8') as f:
9          return sum(1 for _ in f)
10
11
12 if __name__ == "__main__":
13     # Sample input: create a file with 3 lines
14     sample = "example.txt"
15     with open(sample, "w", encoding="utf-8") as f:
16         f.write("First line\nSecond line\nThird line\n")
17     # Sample output: prints 3
18     print(count_lines(sample))
```