

Assignment-4.3

Name-B.Devendar

Hall.ticketno-2303A53036

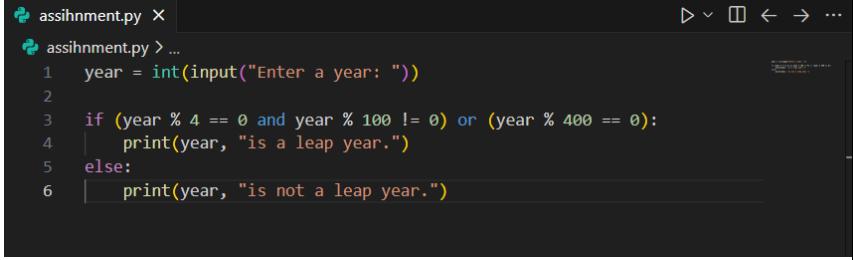
| SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE | | DEPARTMENT OF COMPUTER SCIENCE ENGINEERING | |
|--|--------------------|---|------------------------|
| Program Name: B. Tech | | Assignment Type: Lab | |
| Course Coordinator Name | | Dr. Rishabh Mittal | |
| Instructor(s) Name | | Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju | |
| Course Code | 23CS002PC304 | Course Title | AI Assisted Coding |
| Year/Sem | III/I | Regulation | R23 |
| Date and Day of Assignment | Week 2 - Wednesday | Time(s) | 23CSBTB01 To 23CSBTB52 |
| Duration | 2 Hours | Applicable to Batches | All batches |
| Assignment Number: 3.3(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 <ul style="list-style-type: none"> 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 | Week2 - Wednesday |

Assignment-4.3

Name-B.Devendar

Hall.ticketno-2303A53036

| | | |
|--|---|--|
| | <p>types</p> <p>Lab Outcomes (LOs)</p> <p>After completing this lab, students will be able to:</p> <ul style="list-style-type: none">• 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 <hr/> <p>Task 1: Zero-Shot Prompting – Leap Year Check</p> <p>Scenario</p> <p>Zero-shot prompting involves giving instructions without providing examples.</p> <p>Task Description</p> <p>Use zero-shot prompting to instruct an AI tool to generate a Python function that:</p> <ul style="list-style-type: none">• Accepts a year as input• Checks whether the given year is a leap year• Returns an appropriate result <p>Note: No input-output examples should be provided in the prompt.</p> <p>Expected Output</p> <ul style="list-style-type: none">• AI-generated leap year checking function• Correct logical conditions• Sample input and output• Screenshot of AI-generated response (if required)  <pre>assignment.py > ... assignment.py > ... 1 year = int(input("Enter a year: ")) 2 3 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0): 4 print(year, "is a leap year.") 5 else: 6 print(year, "is not a leap year.") PS C:\Users\Devendar\OneDrive\Music\sqlpractical> & C:/Users/Devendar/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Devendar/OneDrive/Music/sqlpractical/assignment.py Enter a year: 2024 2024 is a leap year. PS C:\Users\Devendar\OneDrive\Music\sqlpractical></pre> | |
|--|---|--|

Assignment-4.3

Name-B.Devendar

Hall.ticketno-2303A53036

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:

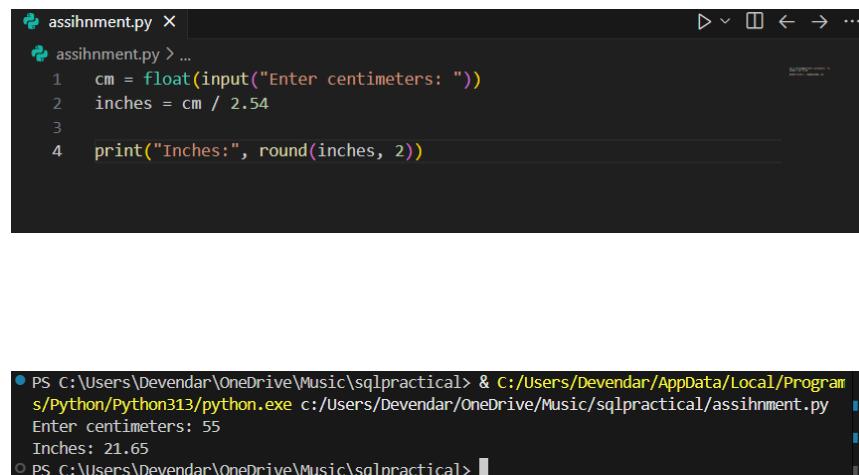
- Converts centimeters to inches
- Uses the correct mathematical formula

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



```
assignment.py X
assignment.py > ...
1   cm = float(input("Enter centimeters: "))
2   inches = cm / 2.54
3
4   print("Inches:", round(inches, 2))

PS C:\Users\Devendar\OneDrive\Music\sqlpractical> & C:/Users/Devendar/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Devendar/OneDrive/Music/sqlpractical/assihmment.py
Enter centimeters: 55
Inches: 21.65
PS C:\Users\Devendar\OneDrive\Music\sqlpractical>
```

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:

Assignment-4.3

Name-B.Devendar

Hall.ticketno-2303A53036

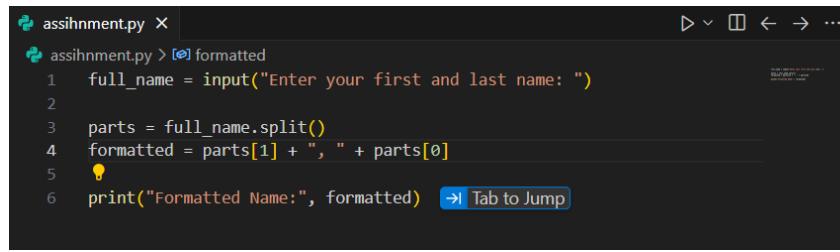
- Accepts a full name as input
- Formats it as "Last, First"

Example formats:

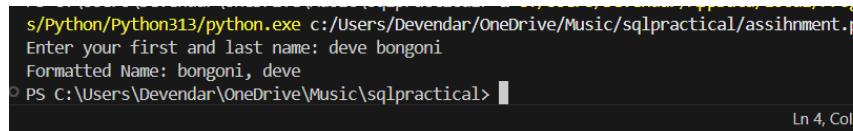
- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output

- Well-structured Python function
- Output strictly following example patterns
- Correct handling of names
- Sample inputs and outputs



```
assihnment.py x
assihnment.py > [e] formatted
1 full_name = input("Enter your first and last name: ")
2
3 parts = full_name.split()
4 formatted = parts[1] + ", " + parts[0]
5
6 print("Formatted Name:", formatted) ➔ Tab to Jump
```



```
s/Python/Python313/python.exe c:/Users/Devendar/OneDrive/Music/sqlpractical/assihnment.py
Enter your first and last name: deve bongoni
Formatted Name: bongoni, deve
PS C:\Users\Devendar\OneDrive\Music\sqlpractical>
```

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:
 - Accuracy
 - Readability
 - Logical clarity

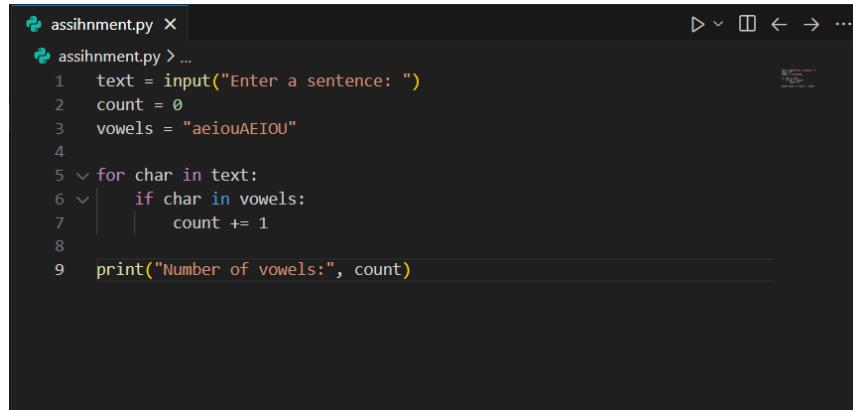
Expected Output

Assignment-4.3

Name-B.Devendar

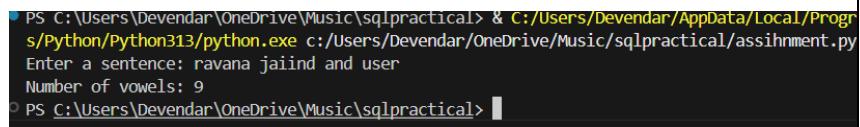
Hall.ticketno-2303A53036

- Two vowel-counting functions
- Comparison table or short reflection paragraph
- Conclusion on prompt effectiveness



A screenshot of a code editor window titled "assihment.py". The code is a Python script that prompts the user for a sentence, counts the vowels in it, and prints the total count. The code uses a for loop to iterate through each character in the input sentence, an if statement to check if the character is a vowel, and a += operator to increment the count variable.

```
assihment.py X
assihment.py > ...
1 text = input("Enter a sentence: ")
2 count = 0
3 vowels = "aeiouAEIOU"
4
5 for char in text:
6     if char in vowels:
7         count += 1
8
9 print("Number of vowels:", count)
```



A screenshot of a terminal window showing the execution of the "assihment.py" script. The user runs the command "python assihment.py" from the directory "C:/Users/Devendar/OneDrive/Music/sqlpractical". The script prompts the user to enter a sentence ("Enter a sentence: ravana jaiind and user"), and then prints the number of vowels ("Number of vowels: 9").

```
PS C:\Users\Devendar\OneDrive\Music\sqlpractical> & C:/Users/Devendar/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Devendar/OneDrive/Music/sqlpractical/assihment.py
Enter a sentence: ravana jaiind and user
Number of vowels: 9
PS C:\Users\Devendar\OneDrive\Music\sqlpractical>
```

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

Assignment-4.3

Name-B.Devendar

Hall.ticketno-2303A53036

```
assihnment.py > [?] filename
1   filename = input("Enter the filename (e.g., data.txt): ")
2   file = open(filename, "r")
3   lines = file.readlines()
4   print("Total lines:", len(lines))
5   file.close()
```

```
s/Python/Python313/python.exe c:/Users/Devendar/OneDrive/Music/sqlpractical/assihnment.py
Enter the filename (e.g., data.txt): C:\Users\Devendar\OneDrive\Music\sqlpractical\data.txt
Total lines: 5
PS C:\Users\Devendar\OneDrive\Music\sqlpractical>
```

Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.