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		1. Dr. Mohammed Ali Shaik 2. Dr. T Sampath Kumar 3. Mr. S Naresh Kumar 4. Dr. V. Rajesh 5. Dr. Brij Kishore 6. Dr Pramoda Patro 7. Dr. Venkataramana 8. Dr. Ravi Chander 9. Dr. Jagjeeth Singh	
CourseCode	24CS002PC215	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week2-Tuesday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	24CSBTB01 To 24CSBTB39
AssignmentNumber:3.2(Present assignment number)/**24**(Total number of assignments)			

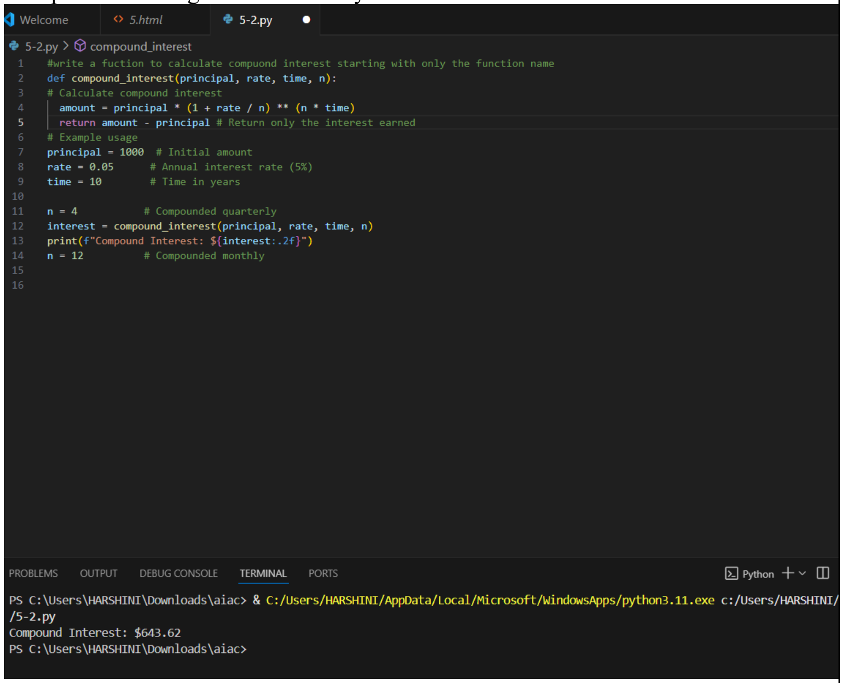
Q.No.	**Question**	***Expected Time to complete***
1	Lab 3: Prompt Engineering – Improving Prompts and Context Management **Lab Objectives:** ● To understand how prompt structure and wording influence AI-generated code. ● To explore how context (like comments and function names) helps AI generate relevant output. ● To evaluate the quality and accuracy of code based on prompt clarity. ● To develop effective prompting strategies for AI-assisted programming. **Lab Outcomes (LOs):** After completing this lab, students will be able to: ● Generate Python code using Google Gemini in Google Colab. ● Analyze the effectiveness of code explanations and suggestions by Gemini. ● Set up and use Cursor AI for AI-powered coding assistance. ● Evaluate and refactor code using Cursor AI features. ● Compare AI tool behavior and code quality across different platforms.	03.08.2025 EOD

Task Description#1

- Ask AI to write a function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example

Expected Output#1

- Comparison of AI-generated code styles



```python
#write a fuction to calculate compound interest starting with only the function name
def compound_interest(principal, rate, time, n):
# Calculate compound interest
    amount = principal * (1 + rate / n) ** (n * time)
    return amount - principal # Return only the interest earned
# Example usage
principal = 1000  # Initial amount
rate = 0.05      # Annual interest rate (5%)
time = 10        # Time in years

n = 4           # Compounded quarterly
interest = compound_interest(principal, rate, time, n)
print(f"Compound Interest: ${interest:.2f}")
n = 12          # Compounded monthly
```

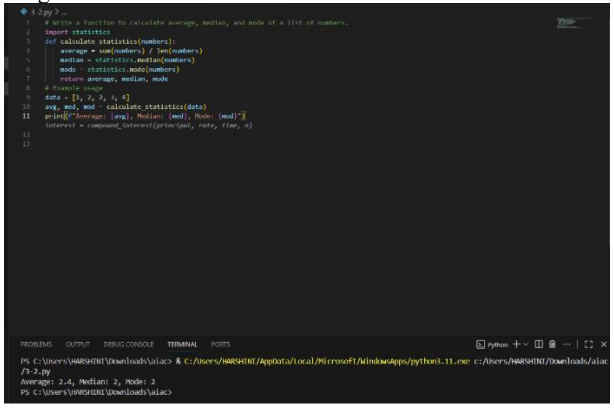
```
PS C:\Users\HARSHINI\Downloads\aiac> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/
/5-2.py
Compound Interest: $643.62
PS C:\Users\HARSHINI\Downloads\aiac>
```

Task Description#2

- Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers.

Expected Output#2

- AI-generated function evolves from unclear to accurate multi-statistical operation.



```python
# Write a function to calculate average, median, and mode of a list of numbers.
import statistics
def calculate_statistics(numbers):
    average = sum(numbers) / len(numbers)
    median = statistics.median(numbers)
    mode = statistics.mode(numbers)
    return average, median, mode
# Example usage
data = [1, 2, 2, 3, 4]
avg, med, mod = calculate_statistics(data)
print(f"Average: {avg}, Median: {med}, Mode: {mod}")
interest = compound_interest(principal, rate, time, n)
```

```
PS C:\Users\HARSHINI\Downloads\aiac> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiac
/3-2.py
Average: 2.4, Median: 2, Mode: 2
PS C:\Users\HARSHINI\Downloads\aiac>
```

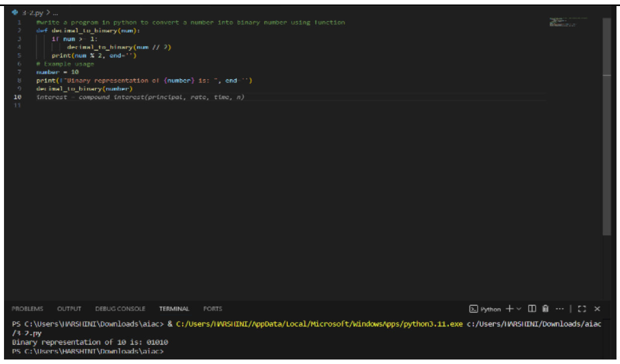
-
-

Task Description#3

- Provide multiple examples of input-output to the AI for convert_to_binary(num) function. Observe how AI uses few-shot prompting to generalize.

Expected Output#3

- Enhanced AI output with clearer prompts



3-2.py > ...

```python
#write a program in python to convert a number into binary number using function
def decimal_to_binary(num):
    if num >= 1:
        decimal_to_binary(num // 2)
    print(num % 2, end='')
# Example usage
number = 10
print(f"Binary representation of {number} is: ", end='')
decimal_to_binary(number)
interest = compound_interest(principal, rate, time, n)
```

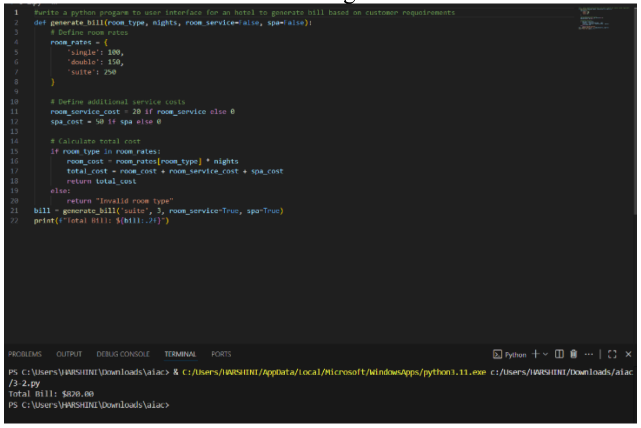
```
PS C:\Users\HARSHINI\Downloads\aiac> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiac
/3-2.py
Binary representation of 10 is: 01010
PS C:\Users\HARSHINI\Downloads\aiac>
```

Task Description#4

- Create an user interface for an hotel to generate bill based on customer requirements

Expected Output#4

- Consistent functions with shared logic



```python
#write a python progarm to user interface for an hotel to generate bill based on customer reqouirements
def generate_bill(room_type, nights, room_service=False, spa=False):
    # Define room rates
    room_rates = {
        'single': 100,
        'double': 150,
        'suite': 250
    }

    # Define additional service costs
    room_service_cost = 20 if room_service else 0
    spa_cost = 50 if spa else 0

    # Calculate total cost
    if room_type in room_rates:
        room_cost = room_rates[room_type] * nights
        total_cost = room_cost + room_service_cost + spa_cost
        return total_cost
    else:
        return "Invalid room type"
bill = generate_bill('suite', 3, room_service=True, spa=True)
print(f"Total Bill: ${bill:.2f}")
```

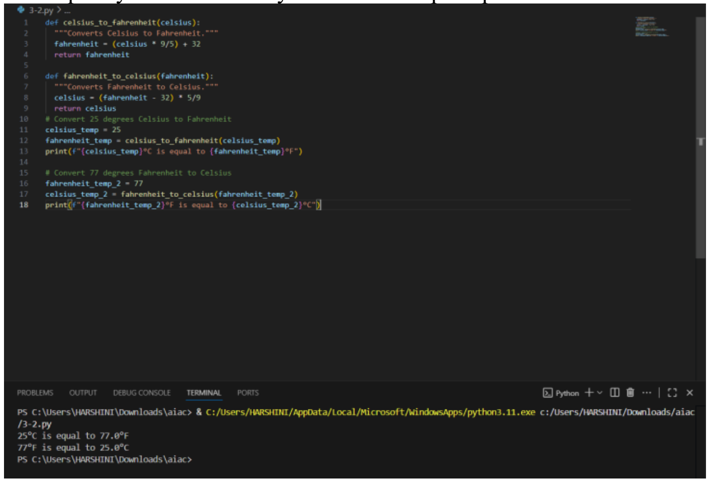
```
PS C:\Users\HARSHINI\Downloads\aiac> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiac
/3-2.py
Total Bill: $820.00
PS C:\Users\HARSHINI\Downloads\aiac>
```

Task Description#5

- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Expected Output#5

- Code quality difference analysis for various prompts



3-2.py > ...

```python
def celsius_to_fahrenheit(celsius):
    """Converts Celsius to Fahrenheit."""
    fahrenheit = (celsius * 9/5) + 32
    return fahrenheit

def fahrenheit_to_celsius(fahrenheit):
    """Converts Fahrenheit to Celsius."""
    celsius = (fahrenheit - 32) * 5/9
    return celsius
# Convert 25 degrees Celsius to Fahrenheit
celsius_temp = 25
fahrenheit_temp = celsius_to_fahrenheit(celsius_temp)
print(f"{celsius_temp}°C is equal to {fahrenheit_temp}°F")

# Convert 77 degrees Fahrenheit to Celsius
fahrenheit_temp_2 = 77
celsius_temp_2 = fahrenheit_to_celsius(fahrenheit_temp_2)
print(f"{fahrenheit_temp_2}°F is equal to {celsius_temp_2}°C")
```

```
PS C:\Users\HARSHINI\Downloads\aiac> & C:/Users/HARSHINI/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/HARSHINI/Downloads/aiac
/3-2.py
25°C is equal to 77.0°F
77°F is equal to 25.0°C
PS C:\Users\HARSHINI\Downloads\aiac>
```

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
Task#1	0.5
Task#2	0.5
Task #3	0.5
Task #4	0.5
Task #5	0.5
Total	**2.5 Marks**