NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

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. Mohammed Ali Shaik Dr. T Sampath Kumar Mr. S Naresh Kumar Dr. V. Rajesh Dr. Brij Kishore Dr Pramoda Patro Dr. Venkataramana Dr. Ravi Chander 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.	03.08.2025 EOD

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

- Evaluate and refactor code using Cursor AI features.
- Compare AI tool behavior and code quality across different platforms.

Task Description#1

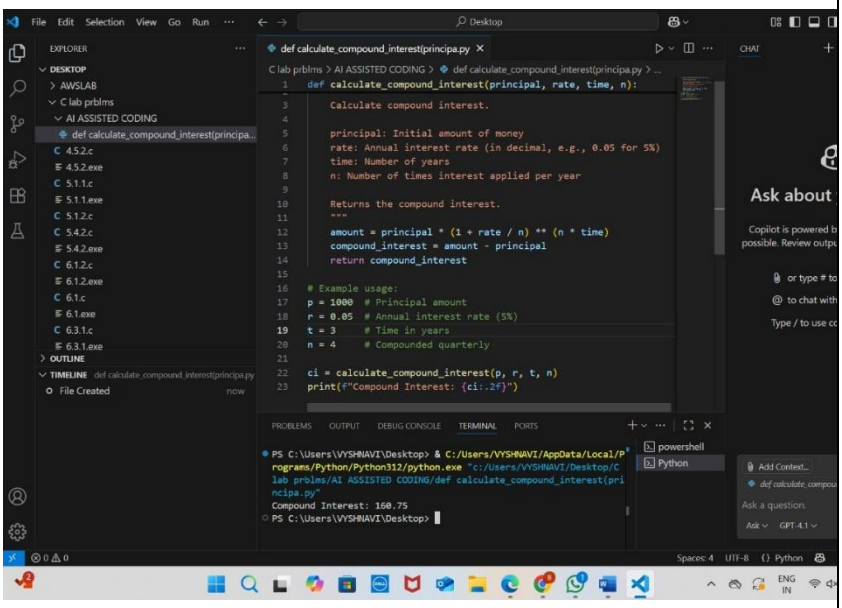
- Ask AI to write a function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example

Expected Output#1

- Comparison of AI-generated code styles

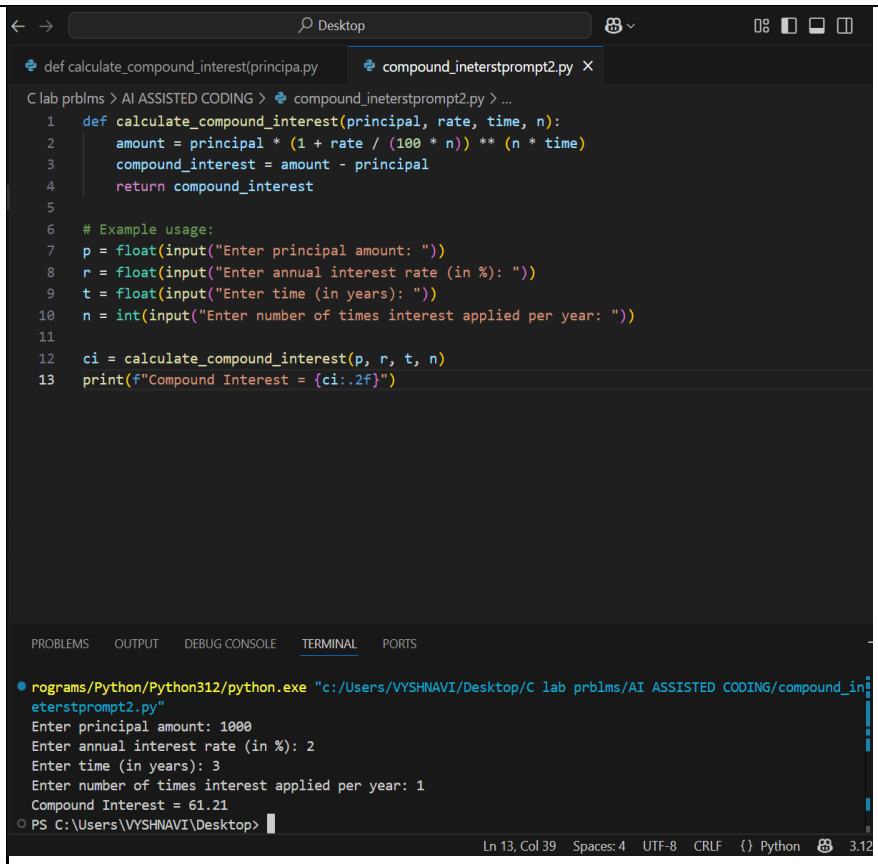
PROMPT1:

Give a python code to calculate the compound interest using functions starting with only function name.

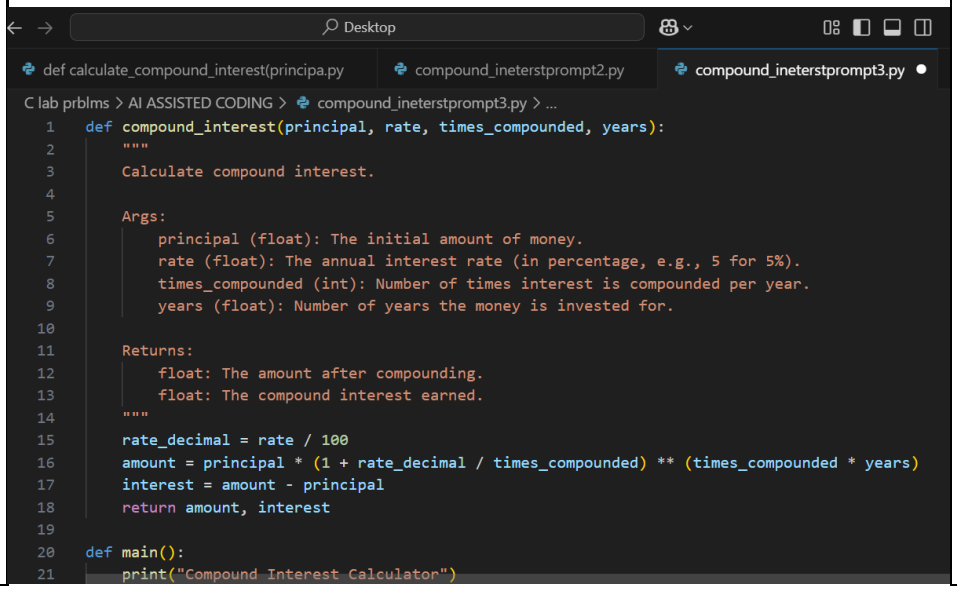


PROMPT2:
write a python code to calculate compound interest using function name

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04



PROMPT3:
write a complex code to calculate the compound interest using functions in python.



NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

```python
def main():
    print("Compound Interest Calculator")
    try:
        principal = float(input("Enter the principal amount: "))
        rate = float(input("Enter the annual interest rate (in %): "))
        times_compounded = int(input("Enter the number of times interest is compounded per year: "))
        years = float(input("Enter the number of years: "))
    except ValueError:
        print("Invalid input. Please enter numeric values.")
        return

    amount, interest = compound_interest(principal, rate, times_compounded, years)
    print(f"\nAfter {years} years, the amount will be: {amount:.2f}")
    print(f"Compound interest earned: {interest:.2f}")

if __name__ == "__main__":
    main()
```

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.

PROMPT 1:

```python
import statistics

def calculate_stats(numbers):
    average = sum(numbers) / len(numbers)
    median = statistics.median(numbers)
    mode = statistics.mode(numbers)
    return average, median, mode

# Example usage:
nums = [1, 2, 2, 3, 4]
avg, med, mod = calculate_stats(nums)
print(f"Average: {avg}")
print(f"Median: {med}")
print(f"Mode: {mod}")
```

```
Enter the number of times interest is compounded per year: 2
Enter the number of years: 4

After 4.0 years, the amount will be: 10828.57
Compound interest earned: 828.57
PS C:\Users\VYSHNAVI\Desktop> & C:/Users/VYSHNAVI/AppData/Local/Programs/Python/Python312/pyt
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/import-statistics1.py"
Average: 2.4
Median: 2
Mode: 2
```

PROMPT 2:

Write a function to calculate average, median, and mode of a list of numbers.

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

import-statistics1.py impo.py

C lab prblms > AI ASSISTED CODING > impo.py > calculate_stats

```python
def calculate_stats(numbers):
    from collections import Counter

    if not numbers:
        return {'average': None, 'median': None, 'mode': None}

    n = len(numbers)
    sorted_nums = sorted(numbers)

    # Average
    average = sum(numbers) / n

    # Median
    if n % 2 == 1:
        median = sorted_nums[n // 2]
    else:
        median = (sorted_nums[n // 2 - 1] + sorted_nums[n // 2]) / 2

    # Mode
    counts = Counter(sorted_nums)
    max_count = max(counts.values())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

import-statistics1.py impo.py

C lab prblms > AI ASSISTED CODING > impo.py > calculate_stats

```python
def calculate_stats(numbers):
        median = sorted_nums[n // 2]
    else:
        median = (sorted_nums[n // 2 - 1] + sorted_nums[n // 2]) / 2

    # Mode
    counts = Counter(sorted_nums)
    max_count = max(counts.values())
    mode = [num for num, count in counts.items() if count == max_count]
    if len(mode) == n:
        mode = None  # No mode if all values are unique
    elif len(mode) == 1:
        mode = mode[0]

    return {'average': average, 'median': median, 'mode': mode}
```

PROMPT 3:

write a complex and simple python code to calculate the mean,median,mode for the list of numbers

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

import-statistics1.py | impo.py | import_statistics3.py

C lab prblms > AI ASSISTED CODING > import_statistics3.py > ...

```python
# Simple version using built-in statistics module
import statistics

numbers = [1, 2, 2, 3, 4, 5, 5, 5, 6]

mean = statistics.mean(numbers)
median = statistics.median(numbers)
mode = statistics.mode(numbers)

print(f"Mean: {mean}")
print(f"Median: {median}")
print(f"Mode: {mode}")

# Complex version without using statistics module
from collections import Counter

def calculate_mean(nums):
    return sum(nums) / len(nums)

def calculate_median(nums):
    nums_sorted = sorted(nums)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\VYSHNAVI\Desktop> & C:/Users/VYSHNAVI/AppData/Local/Programs/P
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/import_statistics3.p
Mean: 3.6666666666666665
Median: 4
Mode: 5

Complex version:
Mean: 3.6666666666666665
Median: 4
Mode: 5
PS C:\Users\VYSHNAVI\Desktop>
```

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.

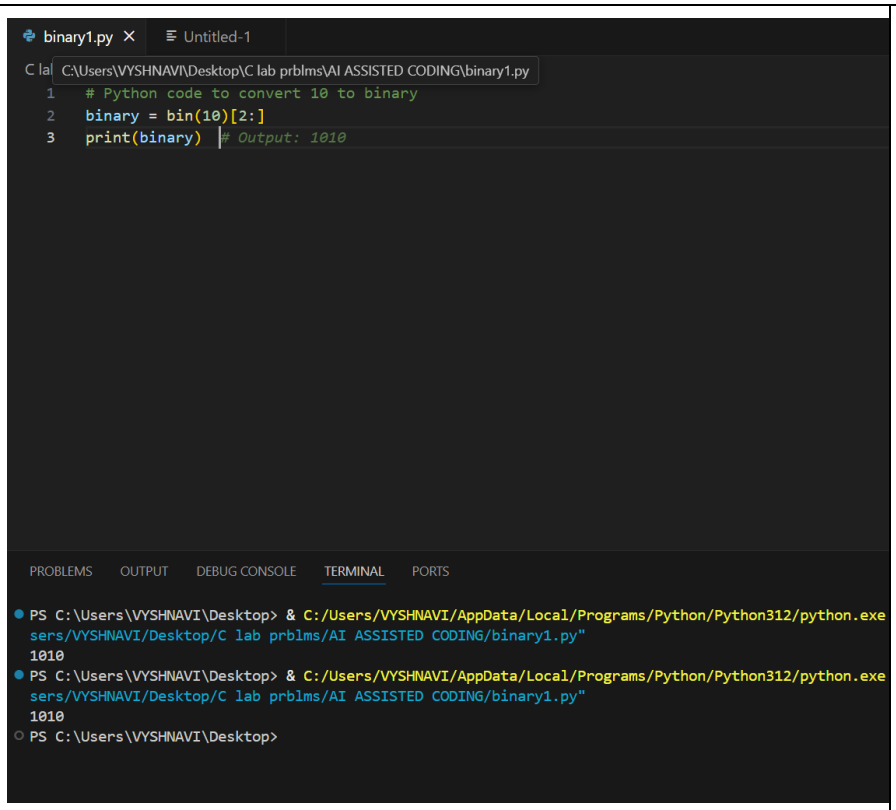
Expected Output#3

- Enhanced AI output with clearer prompts

PROMPT 1:

"Can you convert 10 to binary?"

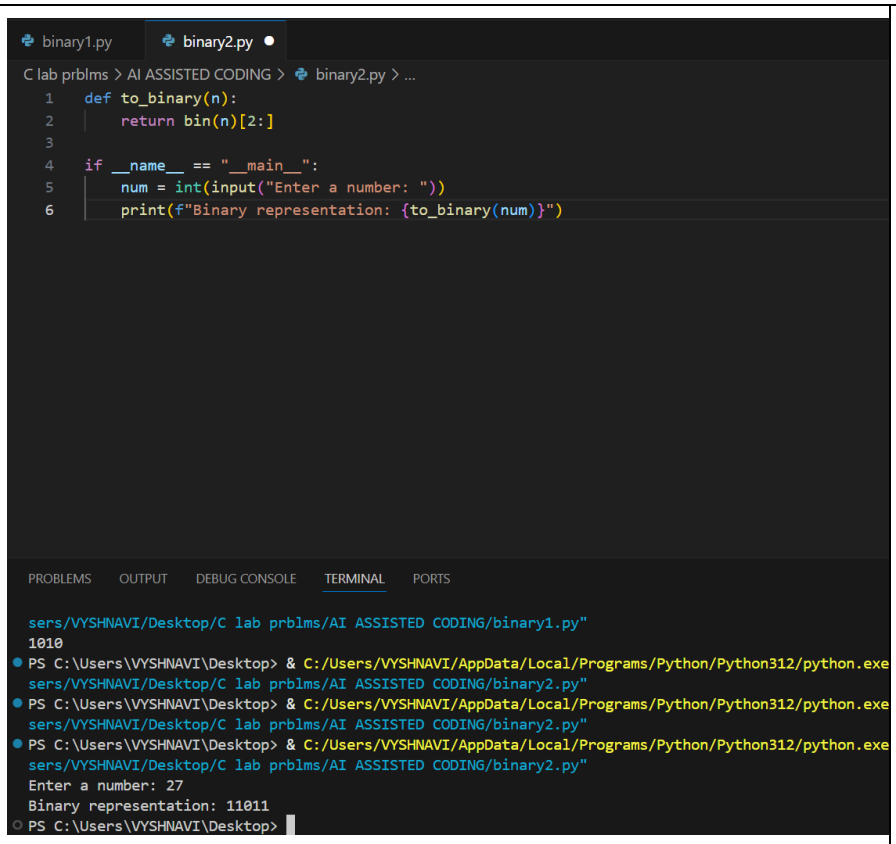
NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04



PROMPT 2:

write a python function code to convert numbers to binary

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04



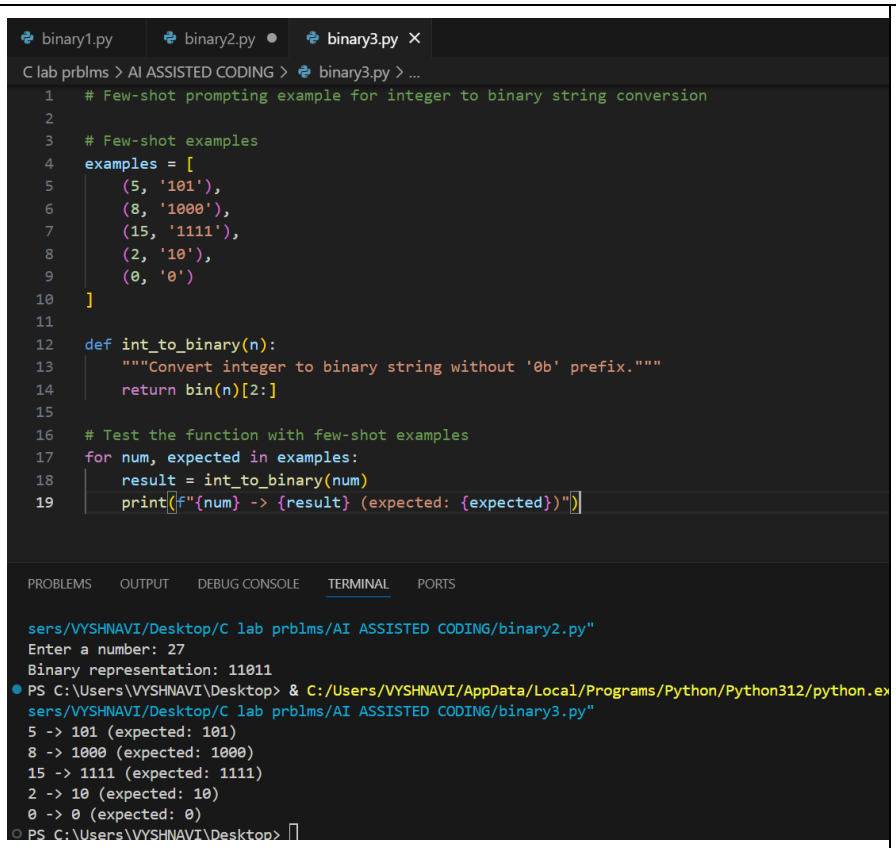
```python
def to_binary(n):
    return bin(n)[2:]

if __name__ == "__main__":
    num = int(input("Enter a number: "))
    print(f"Binary representation: {to_binary(num)}")
```

```
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/binary1.py"
1010
PS C:\Users\VYSHNAVI\Desktop> & C:/Users/VYSHNAVI/AppData/Local/Programs/Python/Python312/python.exe
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/binary2.py"
PS C:\Users\VYSHNAVI\Desktop> & C:/Users/VYSHNAVI/AppData/Local/Programs/Python/Python312/python.exe
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/binary2.py"
PS C:\Users\VYSHNAVI\Desktop> & C:/Users/VYSHNAVI/AppData/Local/Programs/Python/Python312/python.exe
sers/VYSHNAVI/Desktop/C lab prblms/AI ASSISTED CODING/binary2.py"
Enter a number: 27
Binary representation: 11011
PS C:\Users\VYSHNAVI\Desktop>
```

PROMPT 3:

"I want to train an AI model using few-shot prompting to learn how to convert integers to binary strings

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04



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

PROMPT 1:

"Can you make a simple hotel bill generator?"

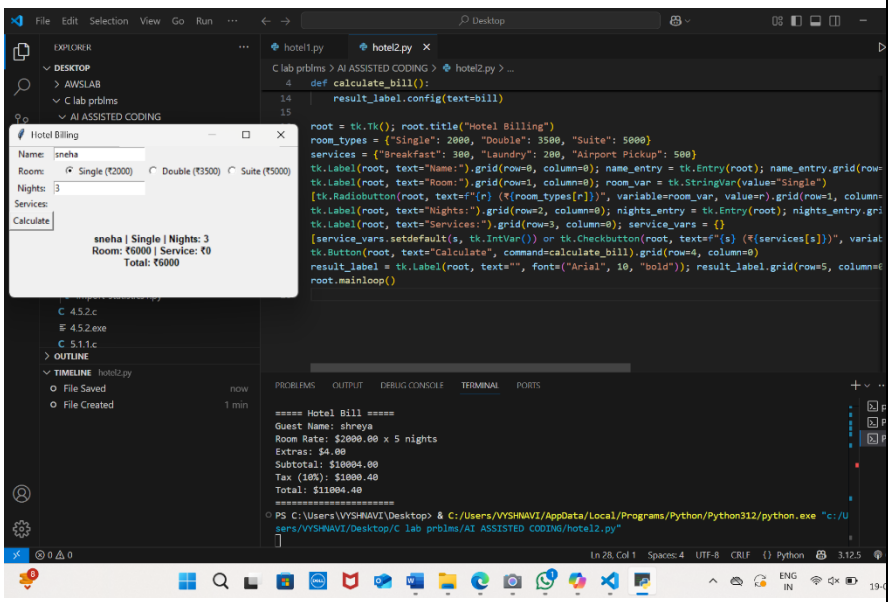
NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

```python
def generate_bill(guest_name, room_rate, nights, extras):
    print(f"Guest Name: {guest_name}")
    print(f"Room Rate: ${room_rate:.2f} x {nights} nights")
    print(f"Extras: ${extras:.2f}")
    print(f"Subtotal: ${subtotal:.2f}")
    print(f"Tax (10%): ${tax:.2f}")
    print(f"Total: ${total:.2f}")
    print("====================")

# Example usage:
if __name__ == "__main__":
    name = input("Enter guest name: ")
    rate = float(input("Enter room rate per night: "))
    nights = int(input("Enter number of nights: "))
    extras = float(input("Enter extras (e.g., minibar, laundry): "))
    generate_bill(name, rate, nights, extras)
```

```
Enter guest name: shreya
Enter room rate per night: 2000
Enter number of nights: 5
Enter extras (e.g., minibar, laundry): 4
===== Hotel Bill =====
Guest Name: shreya
Room Rate: $2000.00 x 5 nights
Extras: $4.00
Subtotal: $10004.00
Tax (10%): $1000.40
Total: $11004.40
```

PROMPT 2:

"I need a user interface for a hotel billing system that calculates charges based on customer preferences like room type and services used. Can you help?"

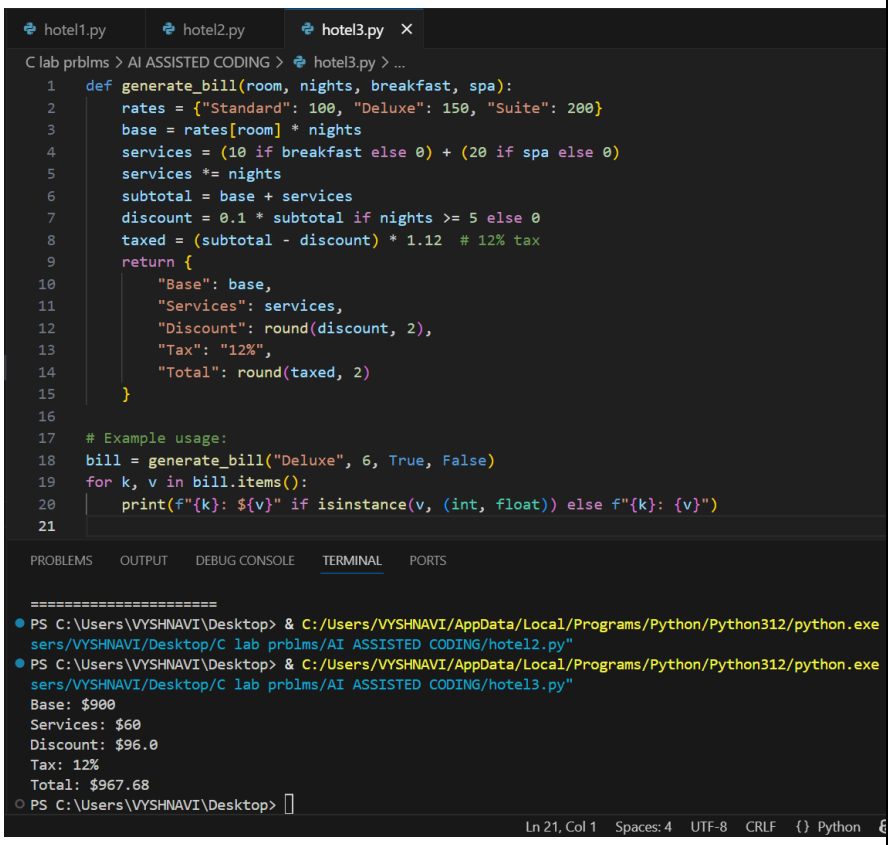


PROMPT 3:

"Design a responsive front-end interface for a hotel management system that generates itemized bills based on customer inputs such as room category, duration of stay, and optional services. Include modular components for discounts, taxes, and final billing

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

summary."



Task Description#5

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

Expected Output#5

- Code quality difference analysis for various prompts

PROMPT 1:

write a code temperature converstion function with clear instruction by using python.

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

```python
def temperature_conversion():
    """
    Temperature Conversion Tool

    Instructions:
    1. Choose the conversion type:
        a. Celsius to Fahrenheit
        b. Fahrenheit to Celsius
    2. Enter the temperature value.
    3. The converted temperature will be displayed.

    Example:
        Enter 1 for Celsius to Fahrenheit, 2 for Fahrenheit to Celsius: 1
        Enter temperature: 25
        25°C is 77.0°F
    """
    print("Temperature Conversion Tool")
    print("1: Celsius to Fahrenheit")
    print("2: Fahrenheit to Celsius")
    choice = input("Enter 1 or 2: ")

    if choice == '1':
        celsius = float(input("Enter temperature in Celsius: "))
        fahrenheit = (celsius * 9/5) + 32
        print(f"{celsius}°C is {fahrenheit}°F")
    elif choice == '2':
        fahrenheit = float(input("Enter temperature in Fahrenheit: "))
        celsius = (fahrenheit - 32) * 5/9
        print(f"{fahrenheit}°F is {celsius}°C")
    else:
        print("Invalid choice. Please enter 1 or 2.")

if __name__ == "__main__":
    temperature_conversion()
```

PROMPT 2:
give a python code to convert the improving temperature conversion funtion with clear instruction.

```python
def convert_temperature(value, from_unit, to_unit):
    """
    Convert temperature between Celsius, Fahrenheit, and Kelvin.

    Parameters:
        value (float): The temperature value to convert.
        from_unit (str): The unit of the input value ('C', 'F', or 'K').
        to_unit (str): The unit to convert to ('C', 'F', or 'K').

    Returns:
        float: The converted temperature value.

    Raises:
        ValueError: If an invalid unit is provided.
    """
    from_unit = from_unit.upper()
    to_unit = to_unit.upper()
    # Convert input to Celsius
    if from_unit == 'C':
        celsius = value
    elif from_unit == 'F':
        celsius = (value - 32) * 5/9
    elif from_unit == 'K':
        celsius = value - 273.15
    else:
        raise ValueError("Invalid from_unit. Use 'C', 'F', or 'K'.")
    # Convert Celsius to target unit
    if to_unit == 'C':
        return celsius
    elif to_unit == 'F':
        return celsius * 9/5 + 32
    elif to_unit == 'K':
        return celsius + 273.15
    else:
        raise ValueError("Invalid to_unit. Use 'C', 'F', or 'K'.")
# To convert 100 Celsius to Fahrenheit:
result = convert_temperature(100, 'C', 'F')
print("100°C is", result, "°F")
# To convert 32 Fahrenheit to Kelvin:
result = convert_temperature(32, 'F', 'K')
print("32°F is", result, "K")
```

PROMPT 3: to improve the temperature write a python code to create a temperatue conventor

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04

```python
def celsius_to_fahrenheit(celsius):
    return (celsius * 9/5) + 32

def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5/9

def main():
    print("Temperature Converter")
    print("1. Celsius to Fahrenheit")
    print("2. Fahrenheit to Celsius")
    choice = input("Choose conversion (1 or 2): ")

    if choice == '1':
        c = float(input("Enter temperature in Celsius: "))
        f = celsius_to_fahrenheit(c)
        print(f"{c}°C = {f:.2f}°F")
    elif choice == '2':
        f = float(input("Enter temperature in Fahrenheit: "))
        c = fahrenheit_to_celsius(f)
        print(f"{f}°F = {c:.2f}°C")
    else:
        print("Invalid choice.")

if __name__ == "__main__":
    main()2
```

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**

NAME: B.VYSHNAVI
ROLL NO.: 2403A510C8
BATCH : 04