Enrollment No :2403A51062
Name :Siripuram Nithya Shree
Assignment :3.2

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.	03.08.2025 EOD

- 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.

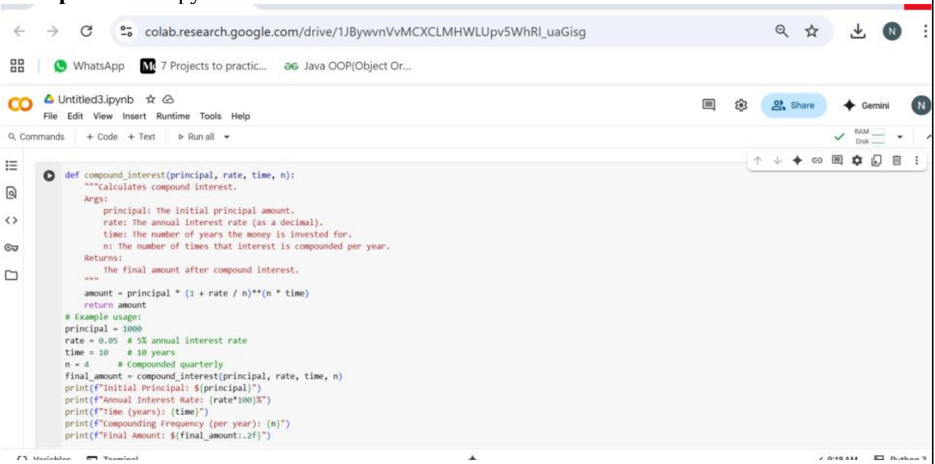
Task Description#1

- 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

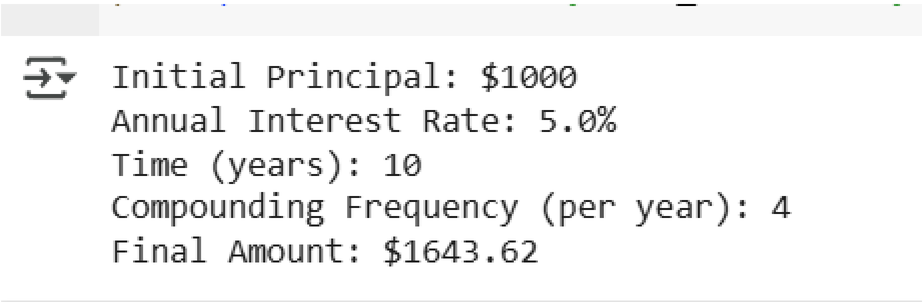
- Comparison of AI-generated code styles

Prompt1: calculate the compound interest starting with only the function name add a docstring then input-output example using function
Prompt2: Write a python code.



```
def compound_interest(principal, rate, time, n):
    """Calculates compound interest.
    Args:
        principal: The initial principal amount.
        rate: The annual interest rate (as a decimal).
        time: The number of years the money is invested for.
        n: The number of times that interest is compounded per year.
    Returns:
        The final amount after compound interest.
    """
    amount = principal * (1 + rate / n)**(n * time)
    return amount
# Example usage:
principal = 1000
rate = 0.05  # 5% annual interest rate
time = 10    # 10 years
n = 4       # Compounded quarterly
final_amount = compound_interest(principal, rate, time, n)
print(f"Initial Principal: ${principal}")
print(f"Annual Interest Rate: {rate*100}%")
print(f"Time (years): {time}")
print(f"Compounding Frequency (per year): {n}")
print(f"Final Amount: ${final_amount:.2f}")
```

OUTPUT:



```
Initial Principal: $1000
Annual Interest Rate: 5.0%
Time (years): 10
Compounding Frequency (per year): 4
Final Amount: $1643.62
```

OBSERVATION:

Given inputs of Initial Principal, Annual Interest Rate and Tine in the form of year then Compound Frequency as per the year and then finding the Final amount . By using the formula of compound Interest
amount = principal * (1 + rate / n)**(n * time) we got the

Final amount.

Task Description#2

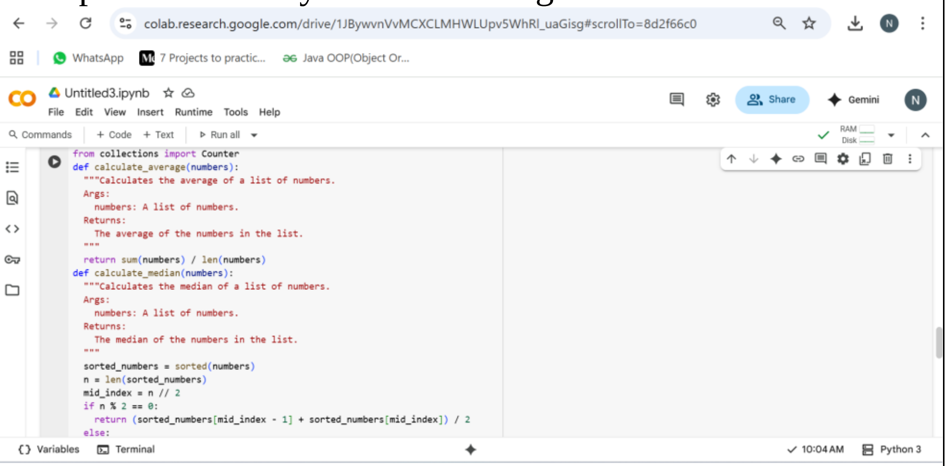
- Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers.

Expected Output#2

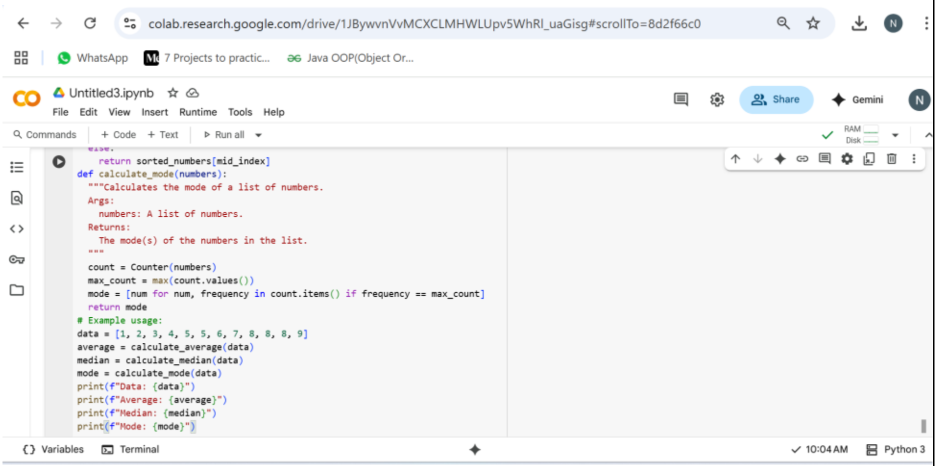
- AI-generated function evolves from unclear to accurate multi-statistical operation.

Prompt1:Calculate the Average ,median and mode of a list of numbers

Prompt2:write the Python code using the Function.

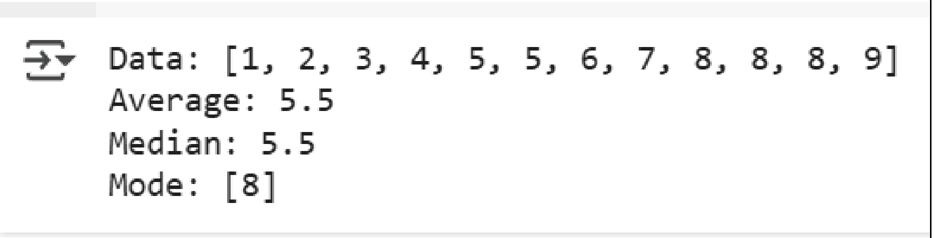


```python
from collections import Counter
def calculate_average(numbers):
  """Calculates the average of a list of numbers.
  Args:
    numbers: A list of numbers.
  Returns:
    The average of the numbers in the list.
  """
  return sum(numbers) / len(numbers)
def calculate_median(numbers):
  """Calculates the median of a list of numbers.
  Args:
    numbers: A list of numbers.
  Returns:
    The median of the numbers in the list.
  """
  sorted_numbers = sorted(numbers)
  n = len(sorted_numbers)
  mid_index = n // 2
  if n % 2 == 0:
    return (sorted_numbers[mid_index - 1] + sorted_numbers[mid_index]) / 2
  else:
```



```python
  else:
    return sorted_numbers[mid_index]
def calculate_mode(numbers):
  """Calculates the mode of a list of numbers.
  Args:
    numbers: A list of numbers.
  Returns:
    The mode(s) of the numbers in the list.
  """
  count = Counter(numbers)
  max_count = max(count.values())
  mode = [num for num, frequency in count.items() if frequency == max_count]
  return mode
# Example usage:
data = [1, 2, 3, 4, 5, 5, 6, 7, 8, 8, 8, 9]
average = calculate_average(data)
median = calculate_median(data)
mode = calculate_mode(data)
print(f"Data: {data}")
print(f"Average: {average}")
print(f"Median: {median}")
print(f"Mode: {mode}")
```

OUTPUT:



```
Data: [1, 2, 3, 4, 5, 5, 6, 7, 8, 8, 8, 9]
Average: 5.5
Median: 5.5
Mode: [8]
```

OBSERVATION:

Mean: The sum of all values divided by the number of values in a list.

Median: The middle value when the data is arranged in ascending or Descending order

If total number of values is odd then use total_values/2

Else total number of values is even then use (total_values/2)+1

Mode: The values that appear most frequently in the list (or) we can also say the most frequently used numbers.

So,Appling all these Mean,Median and Mode we get the ouput correctly.

Task Description#3

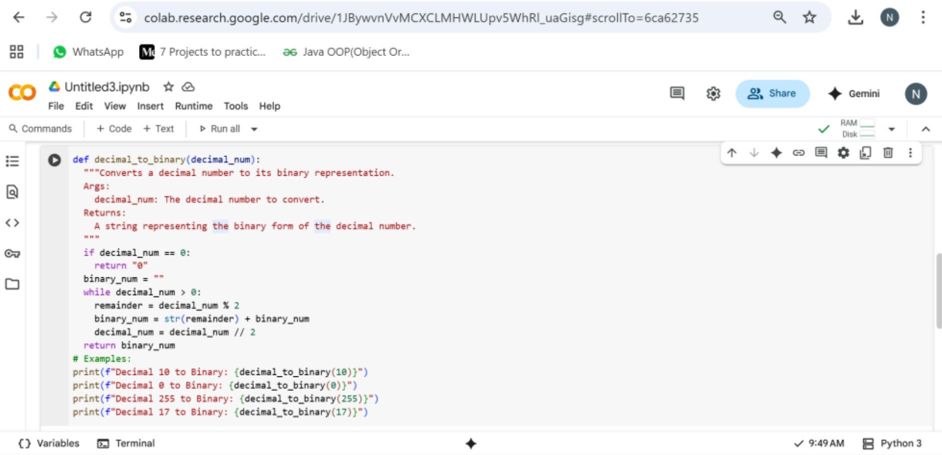
- Provide multiple examples of input-output to the AI for convert_to_binary(num) function. Observe how AI uses few-shot prompting to generalize.

Expected Output#3

- Enhanced AI output with clearer prompts

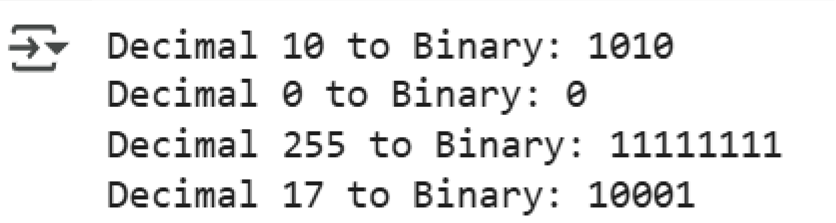
Prompt1: write a python code to convert to binary using function

Prompt2:and Take multiple example



```
def decimal_to_binary(decimal_num):
  """Converts a decimal number to its binary representation.
  Args:
    decimal_num: The decimal number to convert.
  Returns:
    A string representing the binary form of the decimal number.
  """
  if decimal_num == 0:
    return "0"
  binary_num = ""
  while decimal_num > 0:
    remainder = decimal_num % 2
    binary_num = str(remainder) + binary_num
    decimal_num = decimal_num // 2
  return binary_num
# Examples:
print(f"Decimal 10 to Binary: {decimal_to_binary(10)}")
print(f"Decimal 0 to Binary: {decimal_to_binary(0)}")
print(f"Decimal 255 to Binary: {decimal_to_binary(255)}")
print(f"Decimal 17 to Binary: {decimal_to_binary(17)}")
```

OUTPUT:



```
Decimal 10 to Binary: 1010
Decimal 0 to Binary: 0
Decimal 255 to Binary: 11111111
Decimal 17 to Binary: 10001
```

OBSERVATION:

Binary means nothing but it has 2 numbers that is 0 and 1. Firstly they taken the input as 10 then divide the 10 with 2 and write the reaminders the left side.Continue until with the number not divisible with 2.Then,whatever the remainders we wrote in the left side write from last to top then that will be our binary number .Also,same for 255,17 and every number.

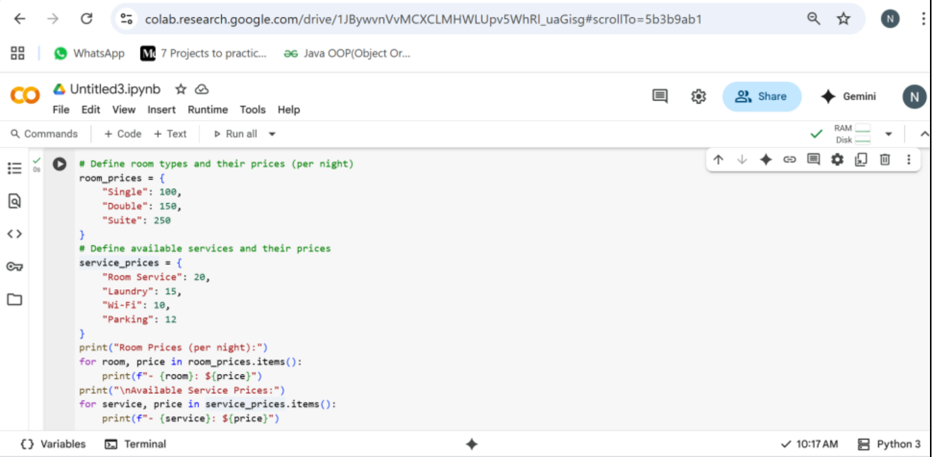
Task Description#4

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

Expected Output#4

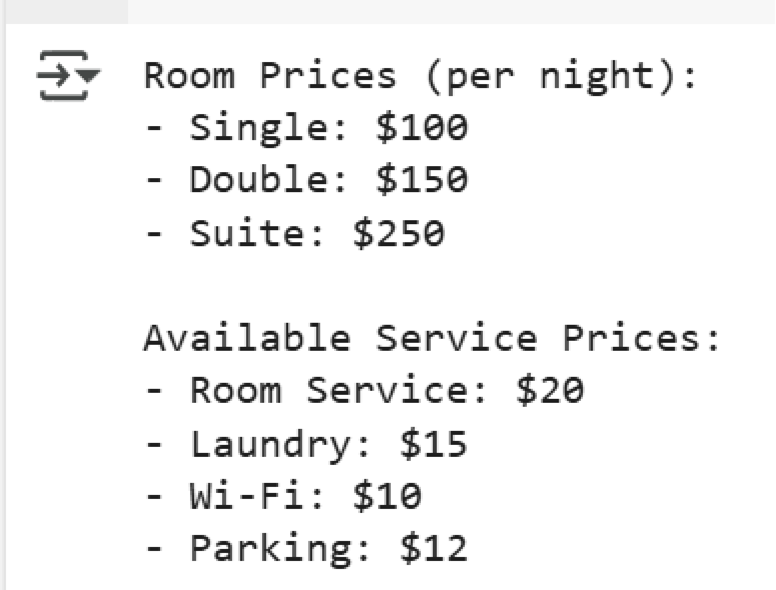
- Consistent functions with shared logic

Prompt: Write a python code to create an user interface for an hotel to generate bill based on customer requirements



```
# Define room types and their prices (per night)
room_prices = {
    "Single": 100,
    "Double": 150,
    "Suite": 250
}
# Define available services and their prices
service_prices = {
    "Room Service": 20,
    "Laundry": 15,
    "Wi-Fi": 10,
    "Parking": 12
}
print("Room Prices (per night):")
for room, price in room_prices.items():
    print(f"- {room}: ${price}")
print("\nAvailable Service Prices:")
for service, price in service_prices.items():
    print(f"- {service}: ${price}")
```

OUTPUT:



```
Room Prices (per night):
- Single: $100
- Double: $150
- Suite: $250

Available Service Prices:
- Room Service: $20
- Laundry: $15
- Wi-Fi: $10
- Parking: $12
```

OBSERVATION:

This program defines and displays the pricing structure for different types of hotel rooms and additional services offered. The code is organized into two main sections that is Room price and Available service prices.

Task Description#5

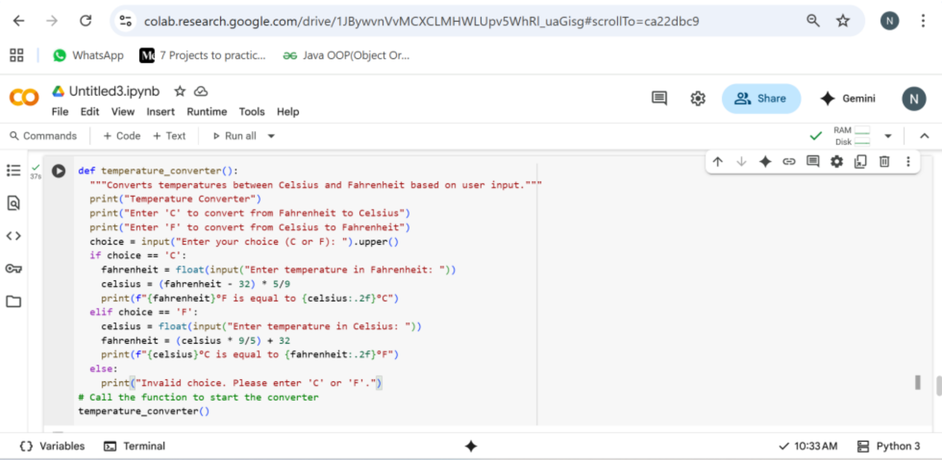
- Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

Expected Output#5

- Code quality difference analysis for various prompts

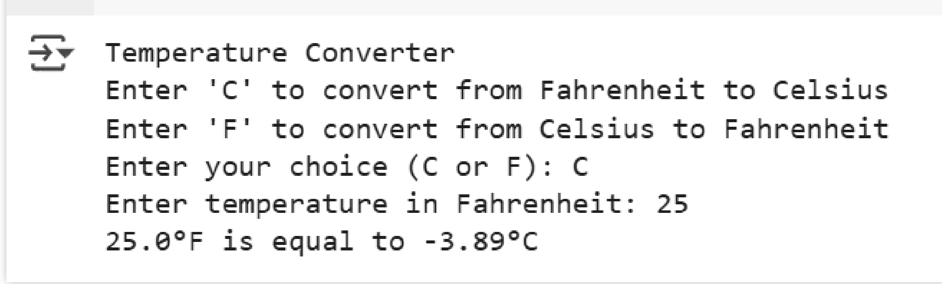
Prompt1: Write a Python Code function that converts temperatures between Celsius and Fahrenheit.

Prompt2: take inputs from the user



```python
def temperature_converter():
  """Converts temperatures between Celsius and Fahrenheit based on user input."""
  print("Temperature Converter")
  print("Enter 'C' to convert from Fahrenheit to Celsius")
  print("Enter 'F' to convert from Celsius to Fahrenheit")
  choice = input("Enter your choice (C or F): ").upper()
  if choice == 'C':
    fahrenheit = float(input("Enter temperature in Fahrenheit: "))
    celsius = (fahrenheit - 32) * 5/9
    print(f"{fahrenheit}°F is equal to {celsius:.2f}°C")
  elif choice == 'F':
    celsius = float(input("Enter temperature in Celsius: "))
    fahrenheit = (celsius * 9/5) + 32
    print(f"{celsius}°C is equal to {fahrenheit:.2f}°F")
  else:
    print("Invalid choice. Please enter 'C' or 'F'.")
# Call the function to start the converter
temperature_converter()
```

OUTPUT:



```
Temperature Converter
Enter 'C' to convert from Fahrenheit to Celsius
Enter 'F' to convert from Celsius to Fahrenheit
Enter your choice (C or F): C
Enter temperature in Fahrenheit: 25
25.0°F is equal to -3.89°C
```

OBSERVATION:

In the above code we given a number that will ask to choose F or C then will convert Celsius to Fahrenheiet and aslo from Fahrenheit to Celsius

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**