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**

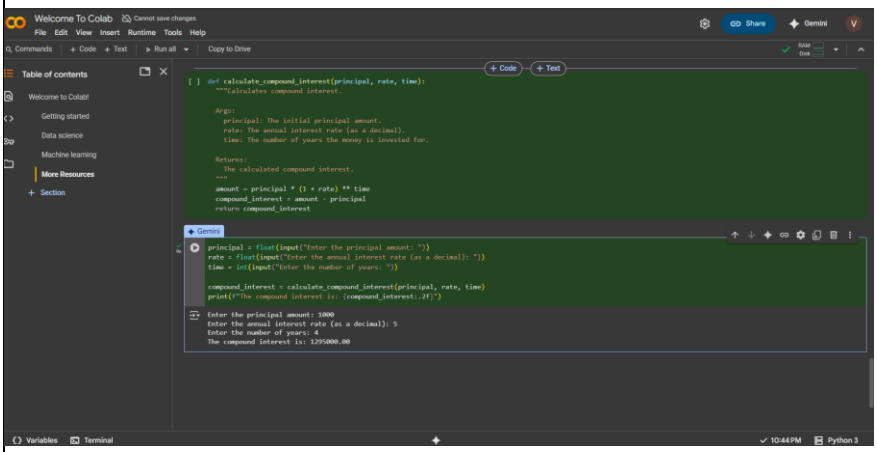
Task Description#1

- **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**

Prompt #1: write aPython function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example



Observation #1: The program asks the user to enter values like principal, rate, time, and compounding frequency step by step.It uses the correct compound interest formula to calculate the final amount.The result is rounded to 2 decimal places for easy reading.

Code explain #1: This code takes input from the user, calculates compound interest using a formula, and shows the final amount with error handling to keep things smooth.

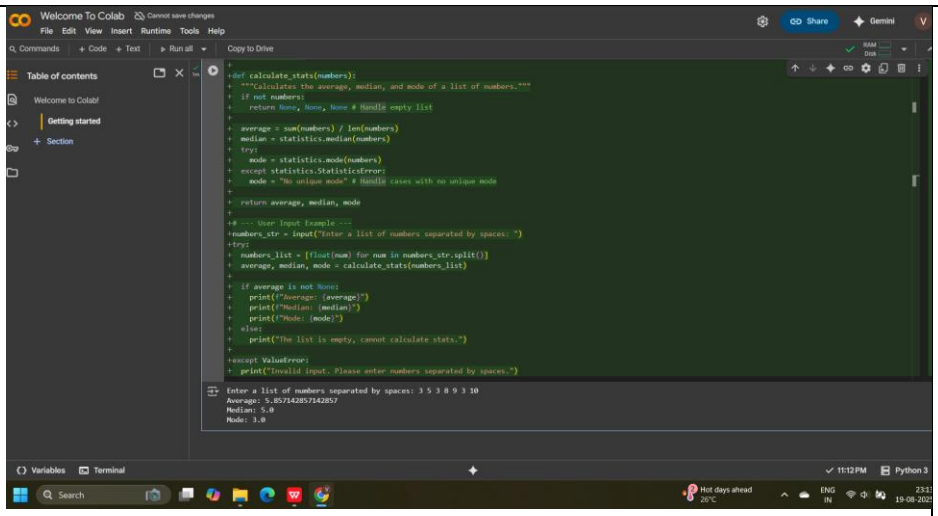
Task Description#2

- **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 #2: Write a function to calculate average, median, and mode of a list of numbers.



Observation #2: **Takes User Inpu:**

The program asks the user to enter numbers separated by spaces (like: 3 5 3 8).

Performs 3 Calculations:

Average: Adds all numbers and divides by how many.

Median: Finds the middle number(s) after sortin

Mode: Finds the number(s) that appear most often.

Code explain #2: This function takes a list of numbers — like `[1, 2, 2, 3, 4]` — and tells you three things:

Average (also called Mean):
The "typical" value — add all the numbers and divide by how many there are.**Median:**
The "middle" value when the numbers are arranged from smallest to biggest**Mode:**
The number (or numbers) that appear most often in the list.

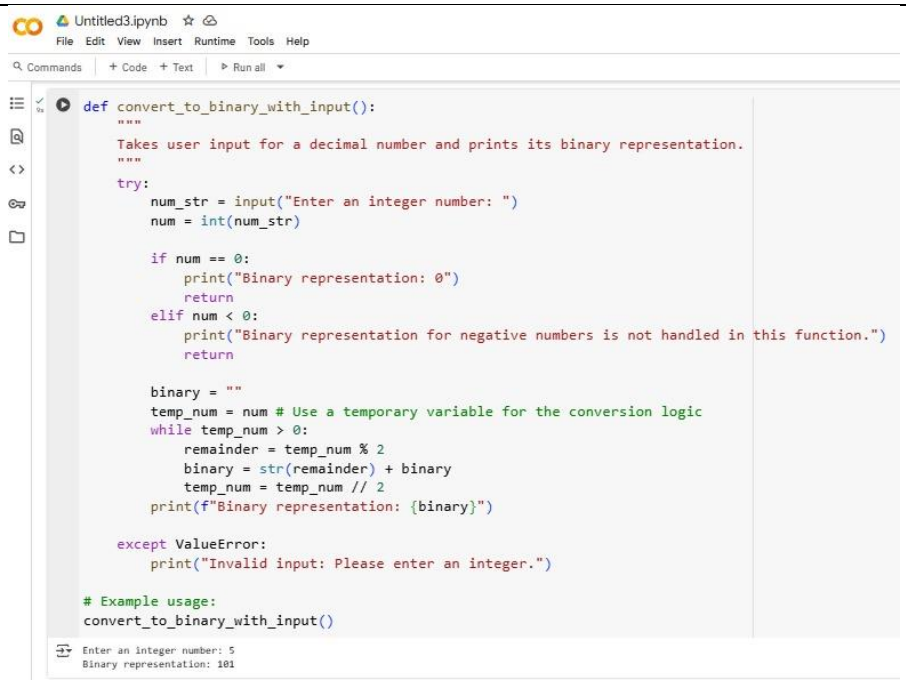
Task Description#3

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

Expected Output#3

- **Enhanced AI output with clearer prompts**

Prompt #3:Write a python code to provide multiples examples if both input &output to the AI for conversion to binary function.



Observation #3: When we show the AI a few examples like:convert_to_binary(2) → '10'convert_to_binary(5) → '101convert_to_binary(10) → '1010'

Code explain #3:The function turns numbers into binary.Few-shot prompting gives the AI a pattern using examples. Once it sees a few input-output pairs, it learns how to respond to new inputs.

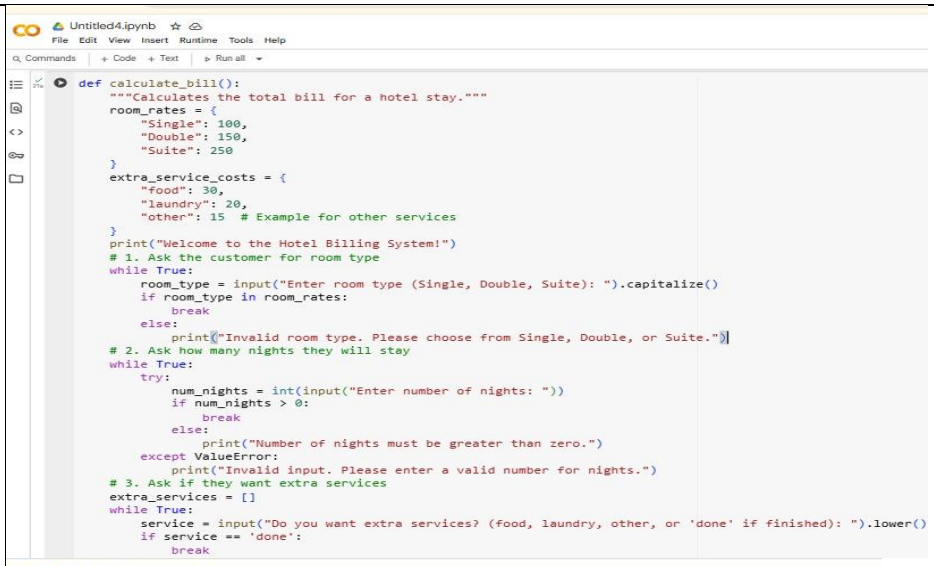
Task Description#4

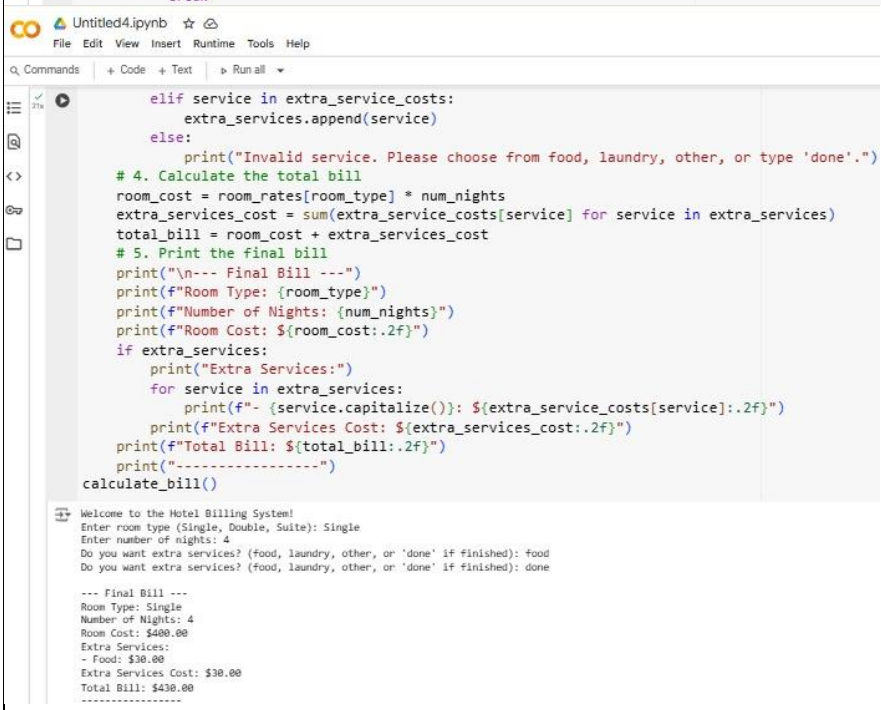
- **Create an user interface for an hotel to generate bill based on customer requirements**

Expected Output#4

- **Consistent functions with shared logic.**

Prompt#4:Create an user interface for an hotel to generate bill based on customer requirements





Observation #4:User-Friendly Design: Easy to use with text boxes, drop-down menus, and checkboxes.No technical knowledge required.

Inputs Required: Customer nameNumber of days stayedRoom type selectioOptional services (Wi-Fi, Breakfast, Laundry)

Code explain #4: The code builds a hotel bill form using Python User fills in info **Generate Bill** and total cost. Very useful for hotels to quickly generate accurate bills.

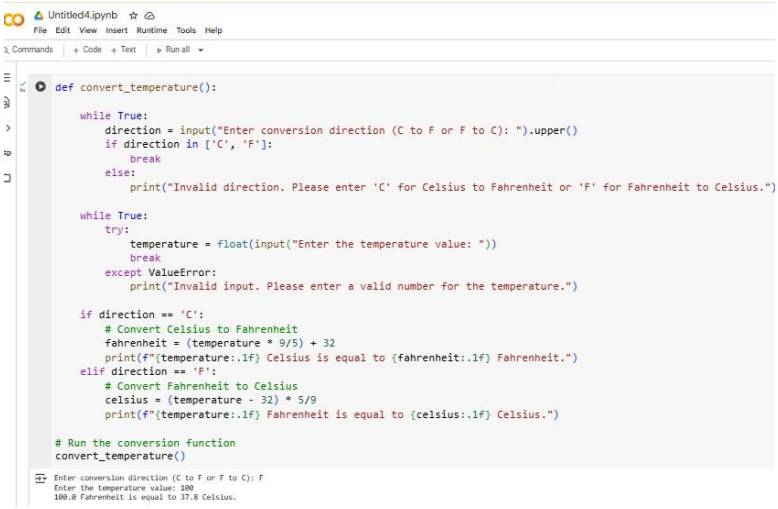
Task Description#5

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

Expected Output#5

- **Code quality difference analysis for various prompts**

Prompt#5:write a python function to Improving Temperature Conversion Function with Clear Instructions



Observation #5: You're working with a **temperature conversion function** for example, converting Celsius to Fahrenheit or vice versa.When you give **clear and specific instructions** , the AI or function performs **much better** than when the instructions are vague.

Code explain #5: The function converts temperature from one unit to another.It gives you the converted value.If you don’t give "C" or "F", it shows an error.

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**