ASSIGNMENT – 3

NAME : M.SINDHUJA HALL TICKET NO : 2403A52060

BATCH NO : AIB03

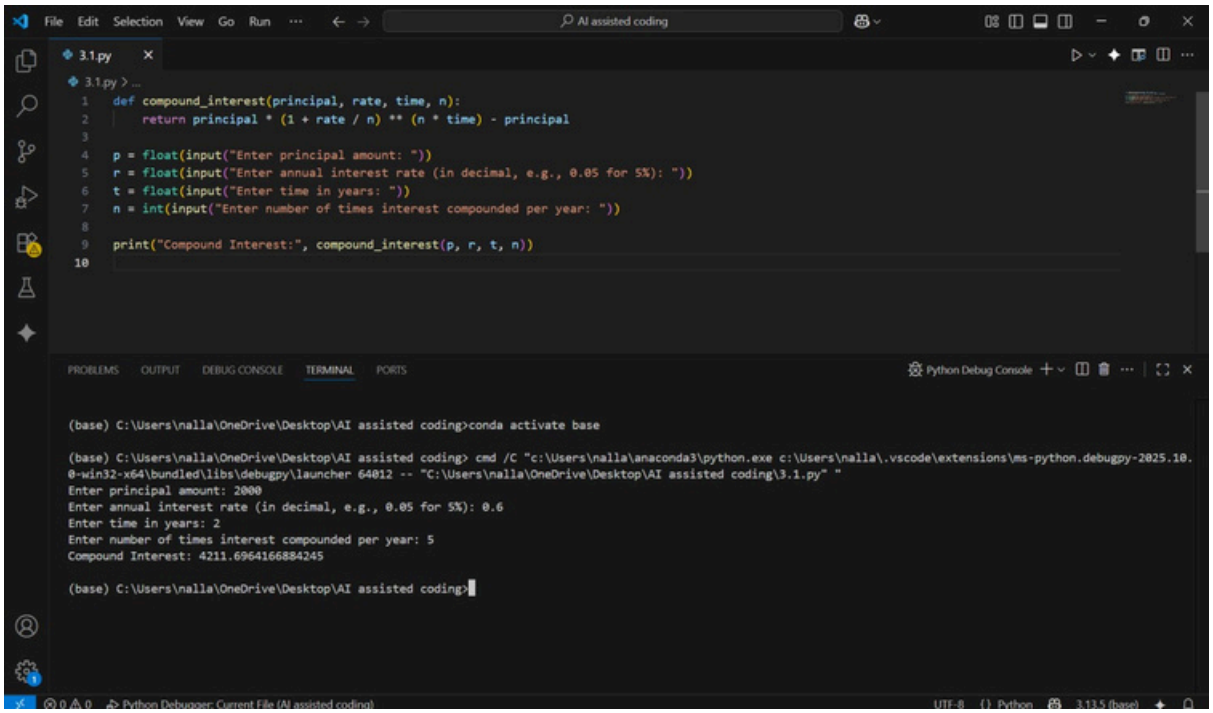
TASK 1 :

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

PROMPT :

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

CODE :



OBSERVATION :

This program calculates the compound interest for a given principal amount. It defines a function compound_interest(principal, rate, time, n)

that uses the formula CI=P×(1+r/n)n⋅t−PCI = P \times (1 + r/n)^{n \cdot t} - P, where P is the principal, r is the annual interest rate, t is the time in years, and n is the number of times interest is compounded per year. The program asks the user to enter these values and then calls the function to compute the compound interest. Finally, it prints the calculated compound interest.

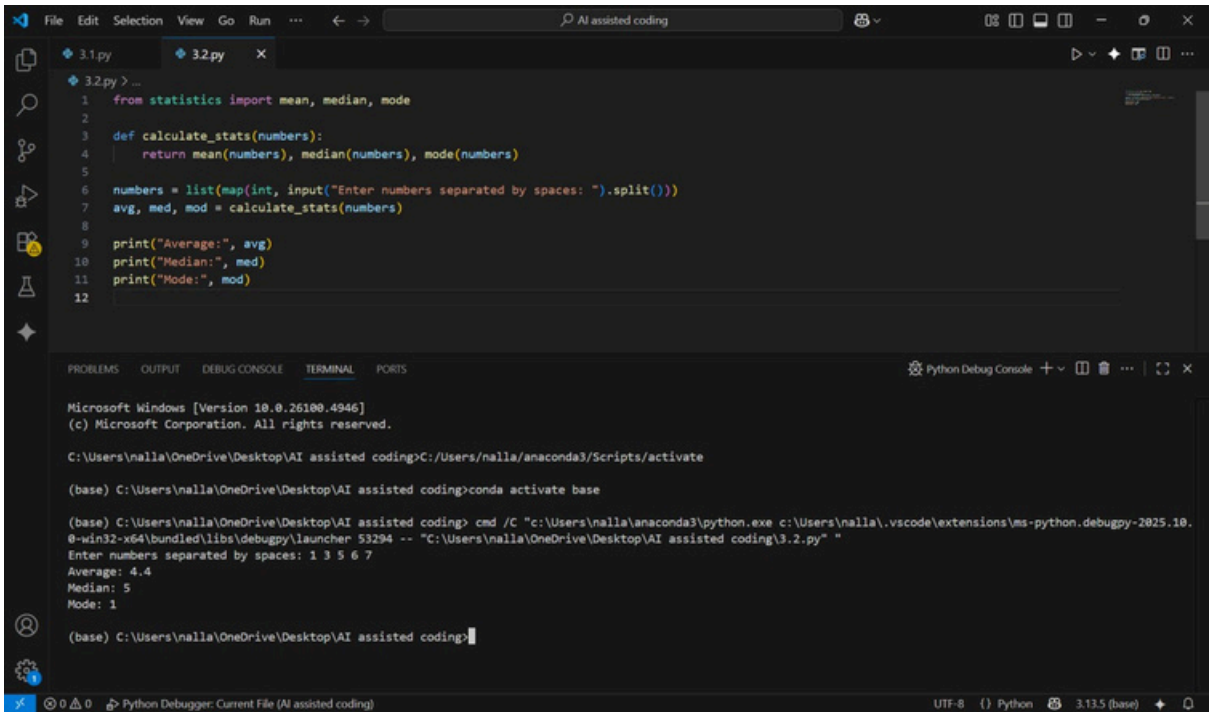
TASK 2:

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

PROMPT :

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

CODE :



OBSERVATION :

This Python code calculates basic statistics—average, median, and mode—of a list of numbers entered by the user. It first imports mean,

median, and mode from the statistics module. The function calculate_stats takes a list of numbers and returns the mean, median, and mode. The program prompts the user to enter numbers separated by spaces, converts them into a list of integers, and passes the list to calculate_stats. It then prints the average, median, and mode of the entered numbers.

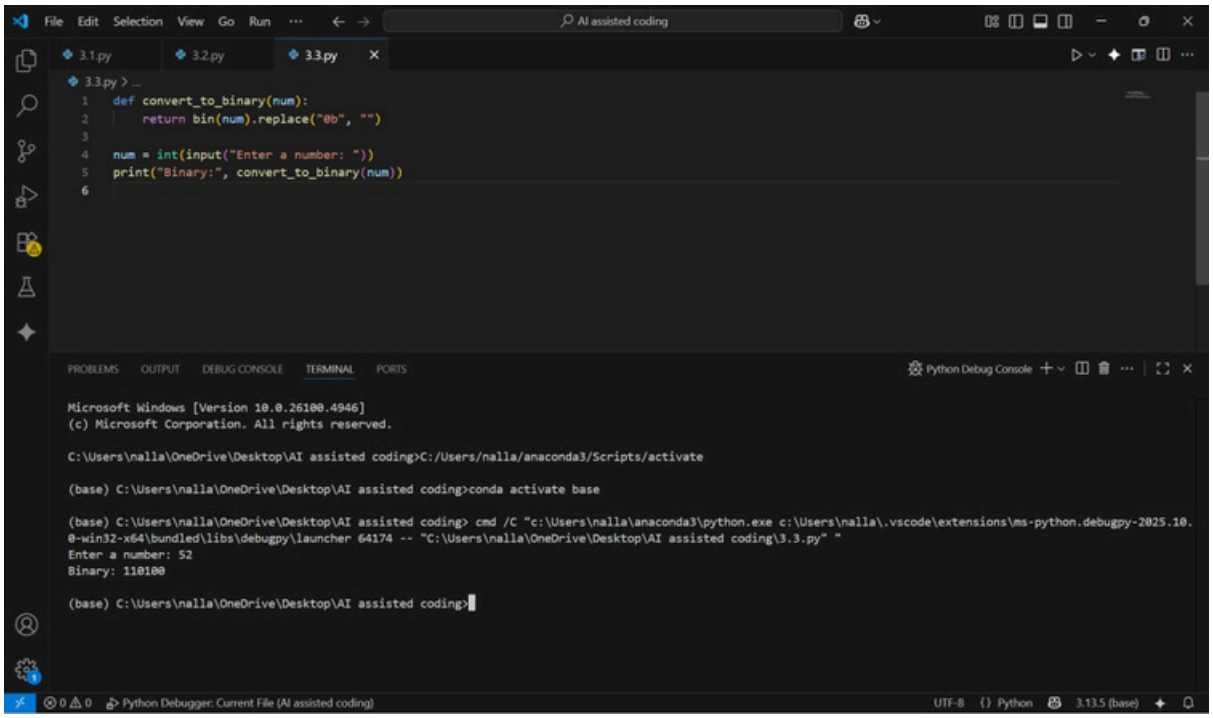
TASK 3 :

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

PROMPT :

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

CODE :



OBSERVATION :

This Python code converts a decimal number entered by the user into its binary representation. It defines a function convert_to_binary that takes an integer num and uses the built-in bin() function to convert it to a binary string. The bin() function returns a string starting with "0b", so .replace("0b", "") removes this prefix. The program prompts the user to enter a number, calls convert_to_binary with this input, and prints the resulting binary value.

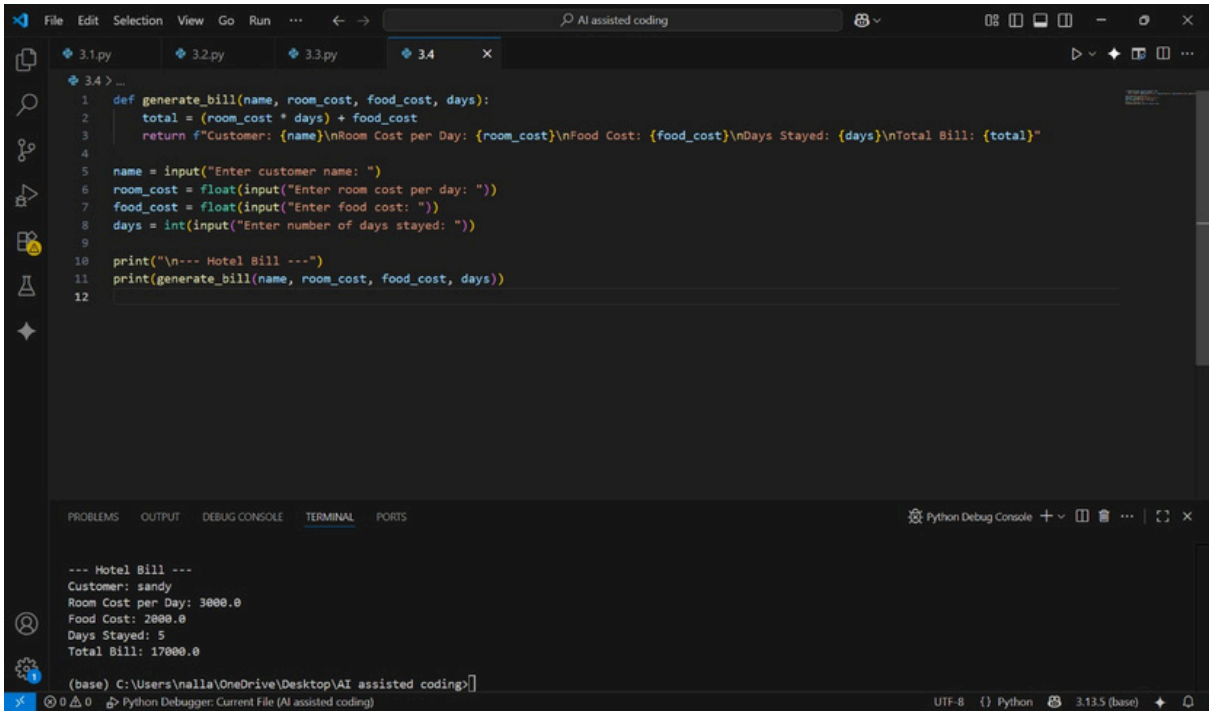
TASK 4 :

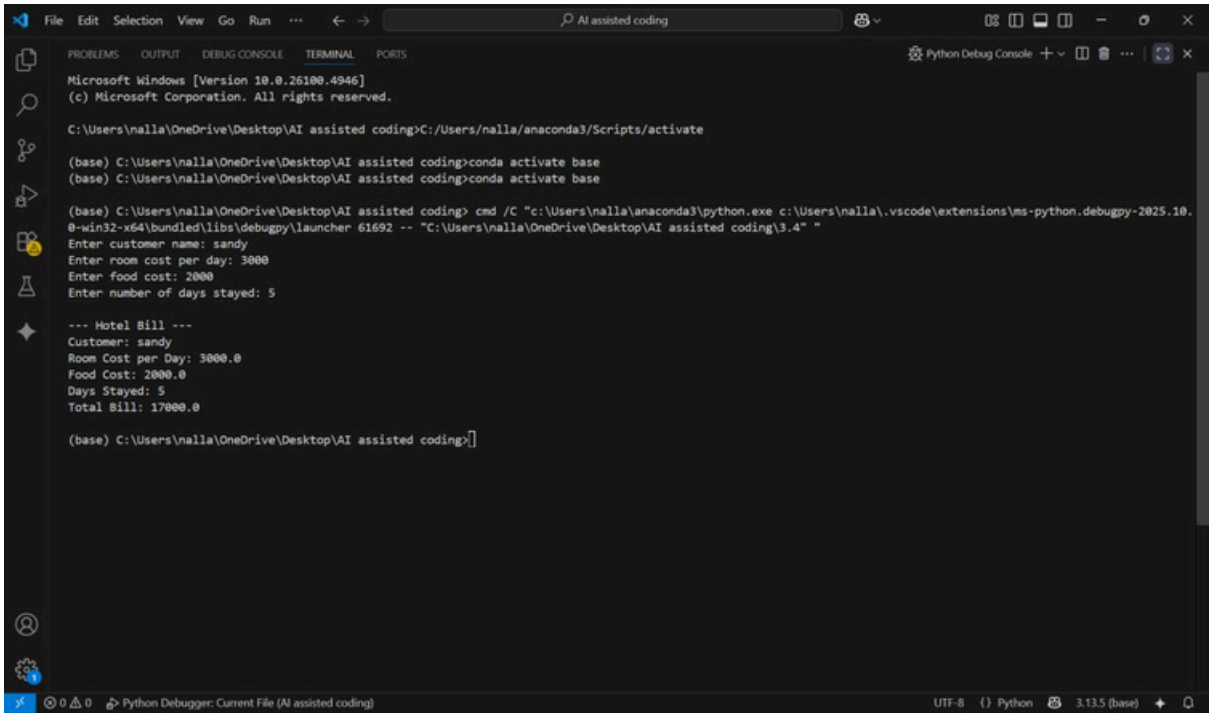
Create an user interface for an hotel to generate bill based on customer requirements

PROMPT :

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

CODE :





OBSERVATION :

This Python code generates a hotel bill for a customer. It defines a function generate_bill that takes the customer's name, room cost per day, food cost, and number of days stayed. Inside the function, it calculates the total bill by multiplying the room cost by the number of days and adding the food cost. It then returns a formatted string showing the customer's name, room cost, food cost, days stayed, and total bill. The program prompts the user to enter these details, calls generate_bill with the inputs, and prints the formatted hotel bill.

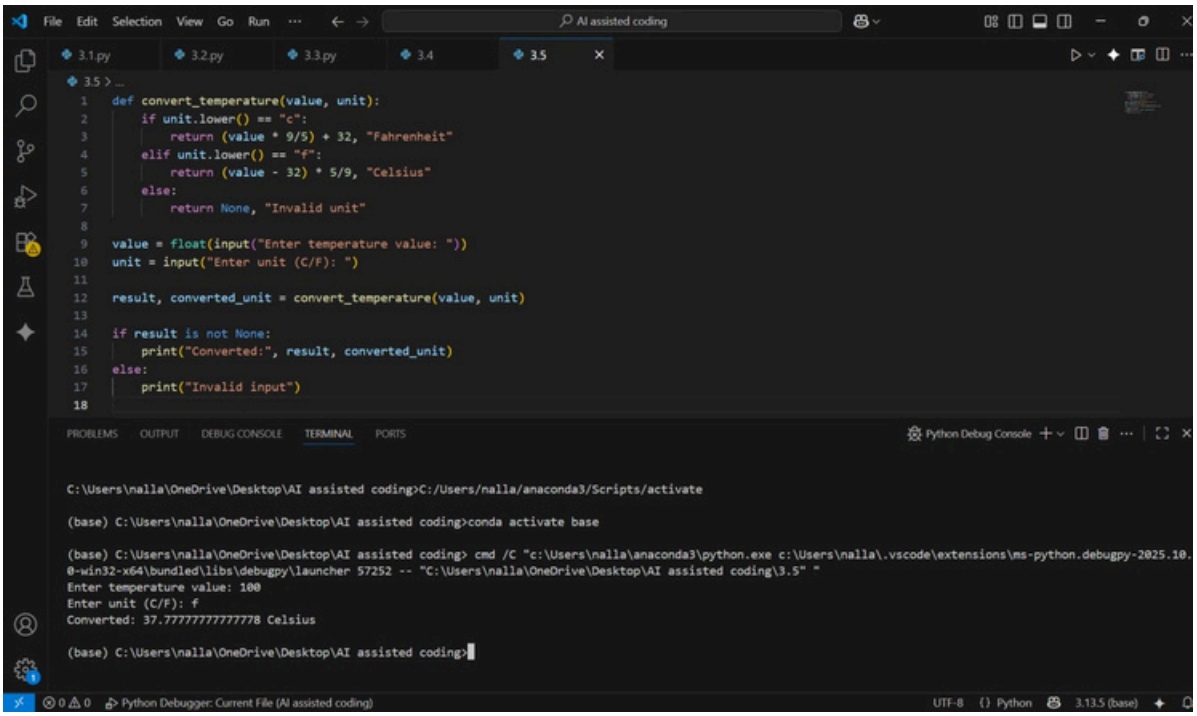
TASK 5 :

Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

PROMPT :

Improving Temperature Conversion Function with
Clear Instructions

CODE :



OBSERVATION:

This Python code converts a temperature between Celsius and Fahrenheit. It defines a function convert_temperature that takes a numeric value and a unit ("C" for Celsius or "F" for Fahrenheit). If the unit is "C", it converts the value to Fahrenheit using the formula (C × 9/5) + 32; if the unit is "F", it converts to Celsius using (F − 32) × 5/9. If the unit is invalid, it returns None and marks the unit as invalid. The program prompts the user to enter a temperature value and its unit, calls convert_temperature, and then prints the converted temperature with the corresponding unit, or an error message if the input was invalid.