**LAB NAME : AI ASSISTED CODING**

**LAB NUMBER :02**

**ROLL NO :2503A51L24**

**BRANCH : CSE**

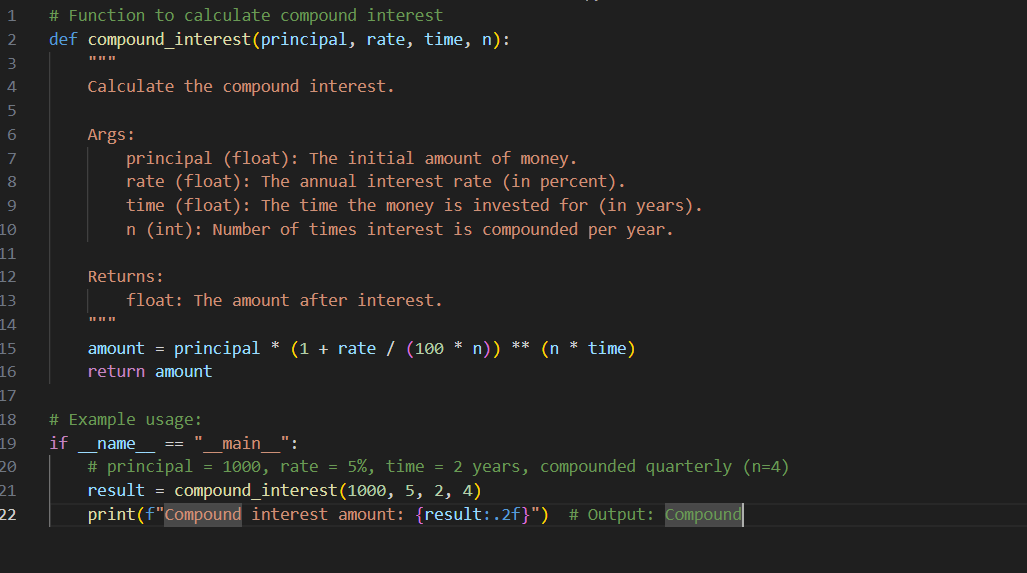
**NAME : A. Sai anvesh**

**TASK 1**

**Task Description:** 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:** Generate a Python function named calculate\_compound\_interest that takes principal, annual interest rate, time in years, and compounding frequency as parameters**.**

**CODE:**

****

**OUTPUT:**

****

**TASK 2**

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

**PROMPT:** Generate a Python function with type hints that calculates the average, median, and mode of a list of numbers, that includes a clear docstring, and handles empty lists or invalid inputs gracefully**.**

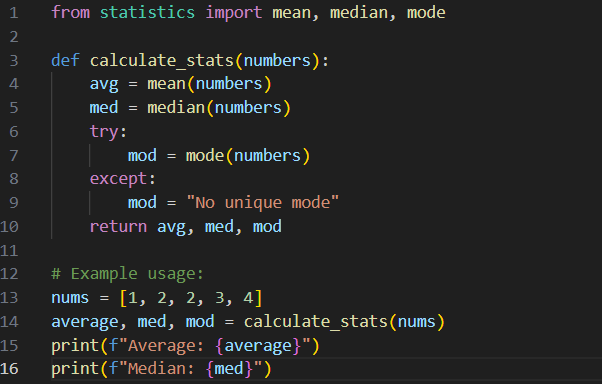
**CODE:**

**TASK 2**

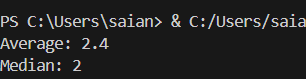
**Task Description:** Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers**.**

**PROMPT:** Generate a Python function with type hints that calculates the average, median, and mode of a list of numbers, that includes a clear docstring, and handles empty lists or invalid inputs gracefully**.**

**CODE:**

****

**OUTPUT:**

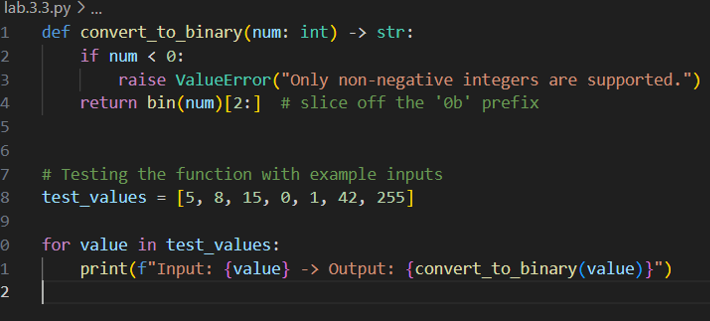


**TASK 3**

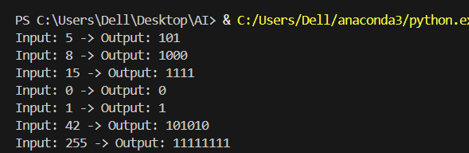
**Task Description:** Provide multiple examples of input-output to the AI for convert\_to\_binary(num) function. Observe how AI uses few-shot prompting to generalize**.**

**PROMPT:** Provide multiple examples of input-output for the convert\_to\_binary(num) function to help the AI generalize using few-shot prompting**.**

**CODE:**

****

**OUTPUT:**

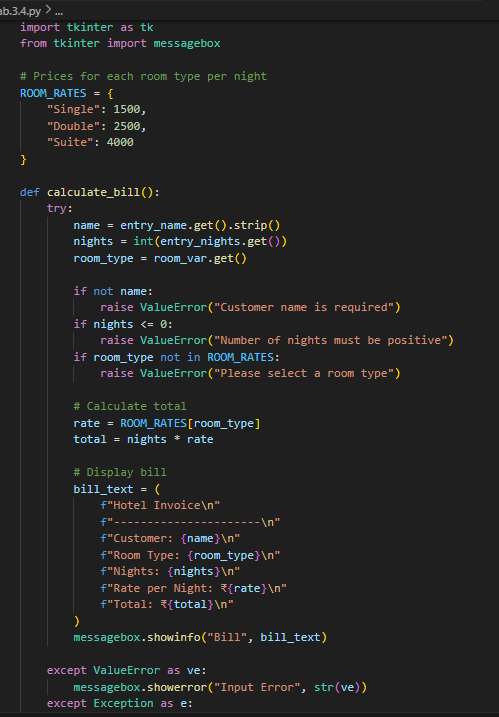


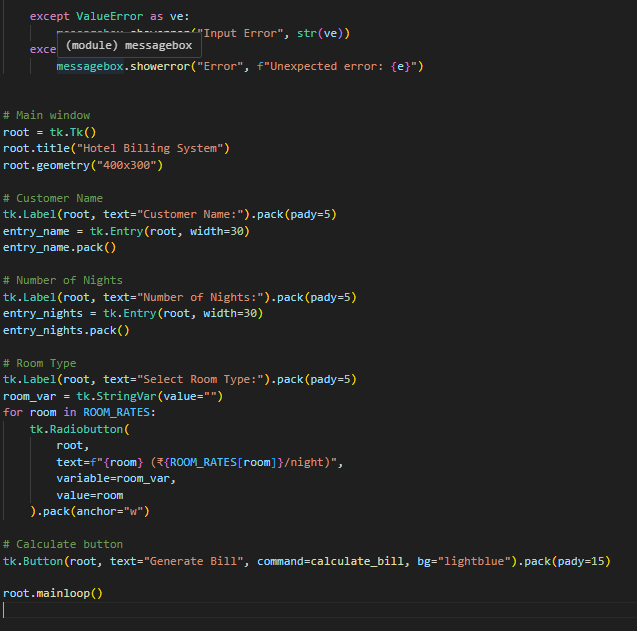
**TASK 4**

**Task Description:** Create a user interface for a hotel to generate bill based on customer requirements.

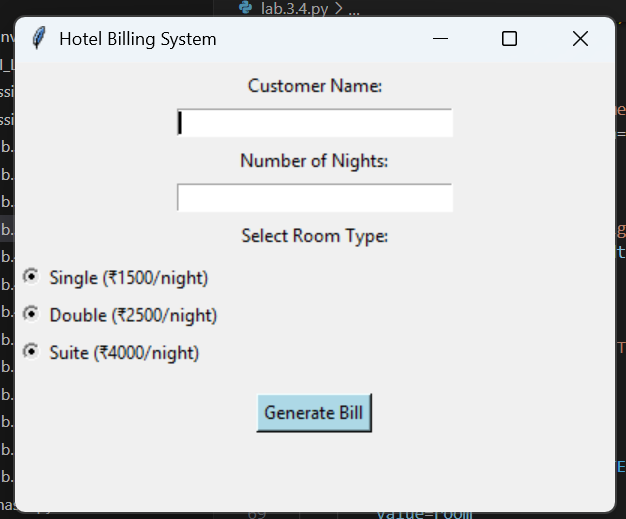
**PROMPT**: Create a user interface for a hotel billing system that generates a bill based on customer requirements**.**

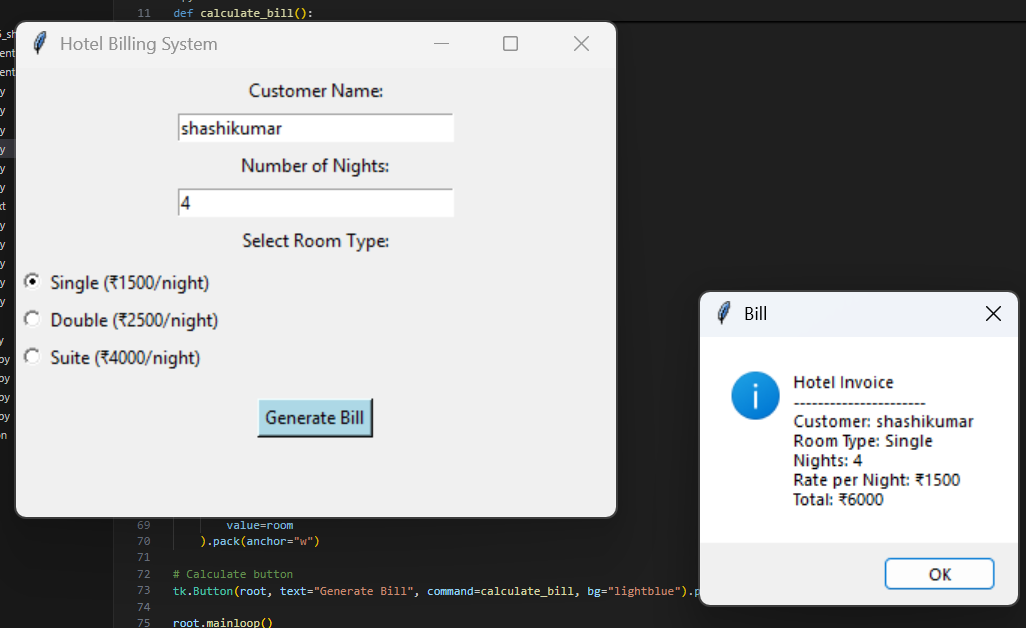
**CODE:**

****

****

**OUTPUT:**

****

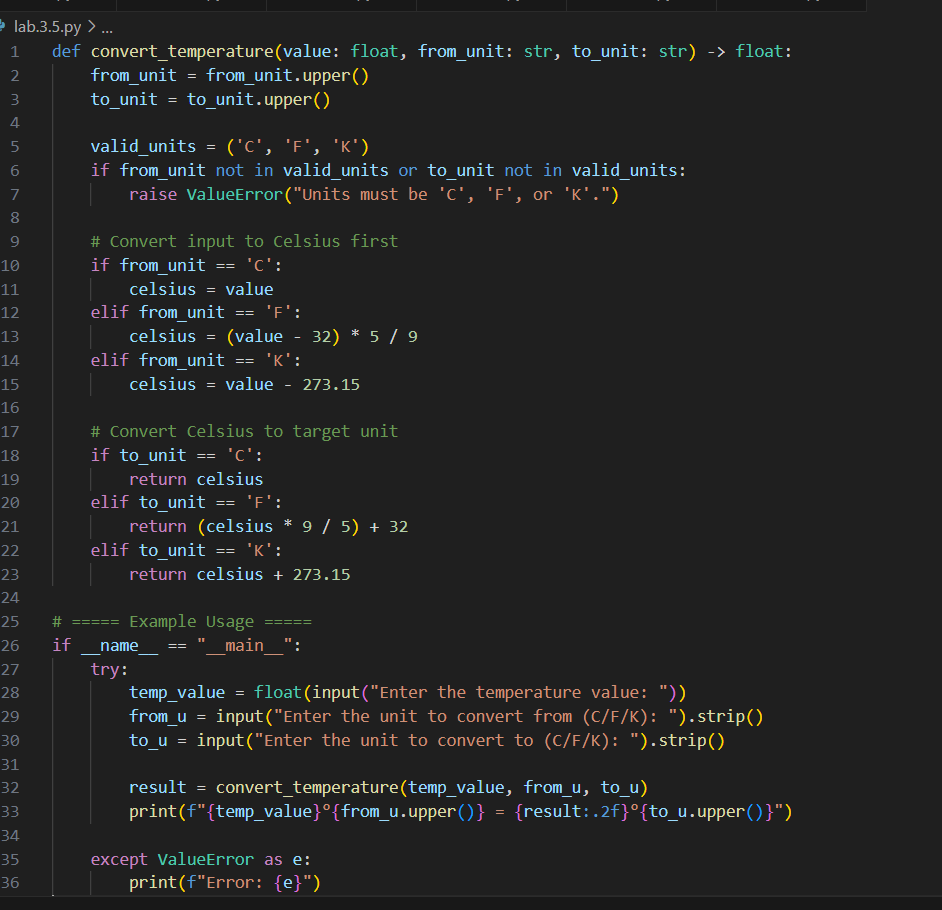
****

**TASK 5**

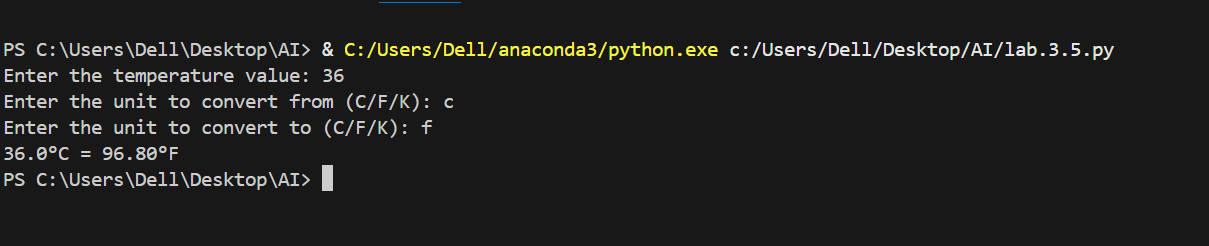
**Task Description:** Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions

**PROMPT:** Give a Python Program to convert temperatures between Celsius, Fahrenheit, and Kelvin.

**CODE:**

****

**OUTPUT:**

****

**OBSERVATION:** I observed that GitHub copilot can quickly generate working code for tasks such as login systems, loan approvals, Fibonacci functions, and job applicant scoring. However, the generated code sometimes contains issues like hardcoded values, lack of encryption, or biased decision logic. This shows that AI tools are helpful for faster coding but require human review for security, fairness, and correctness. GitHub Copilot is a fascinating tool to observe—especially in how it transforms the developer experience. Here's a breakdown of key observations across its functionality, impact, and adoption