AI-Assisted Coding

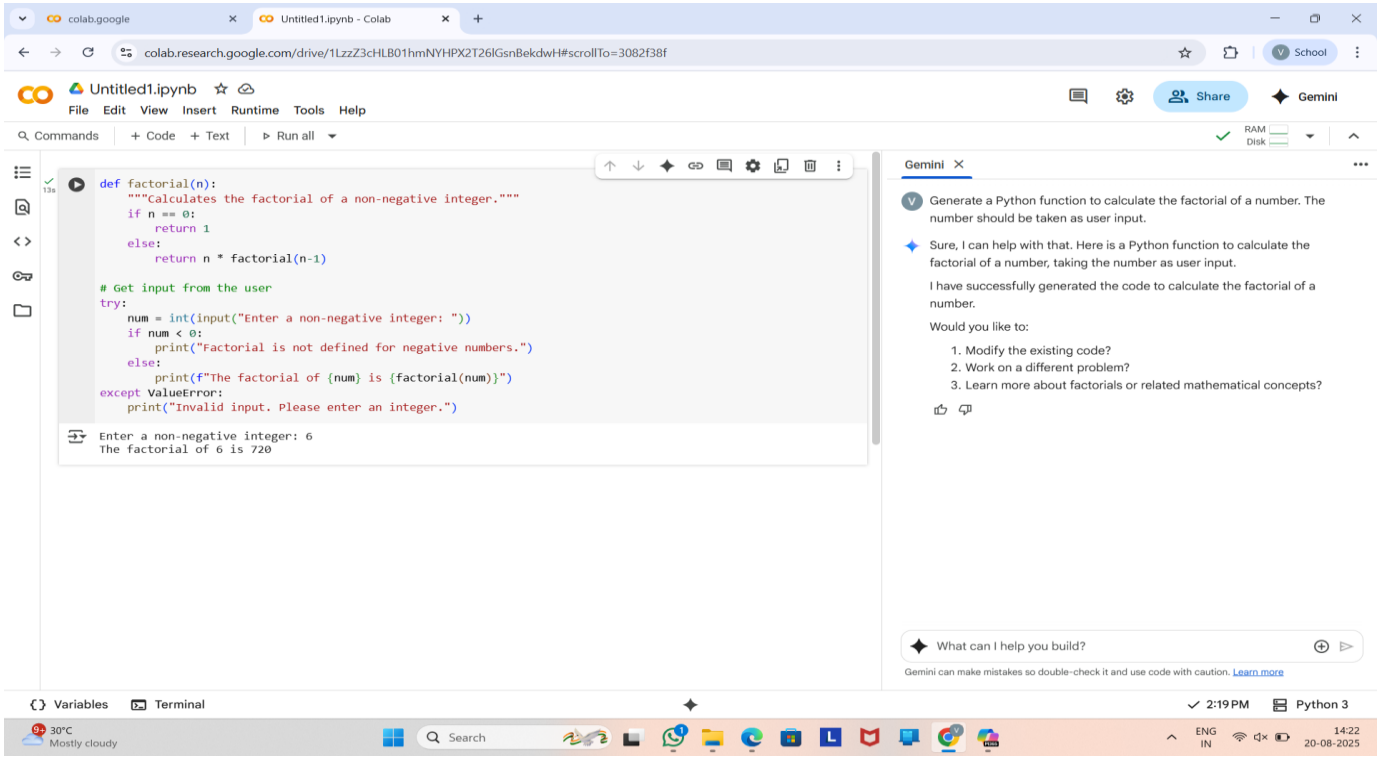
Name:Vangapandla Koushik

Htno:2403A52004

Task-1:

Prompt-1: Generate a Python function to calculate the factorial of a number. The number should be taken as user input.

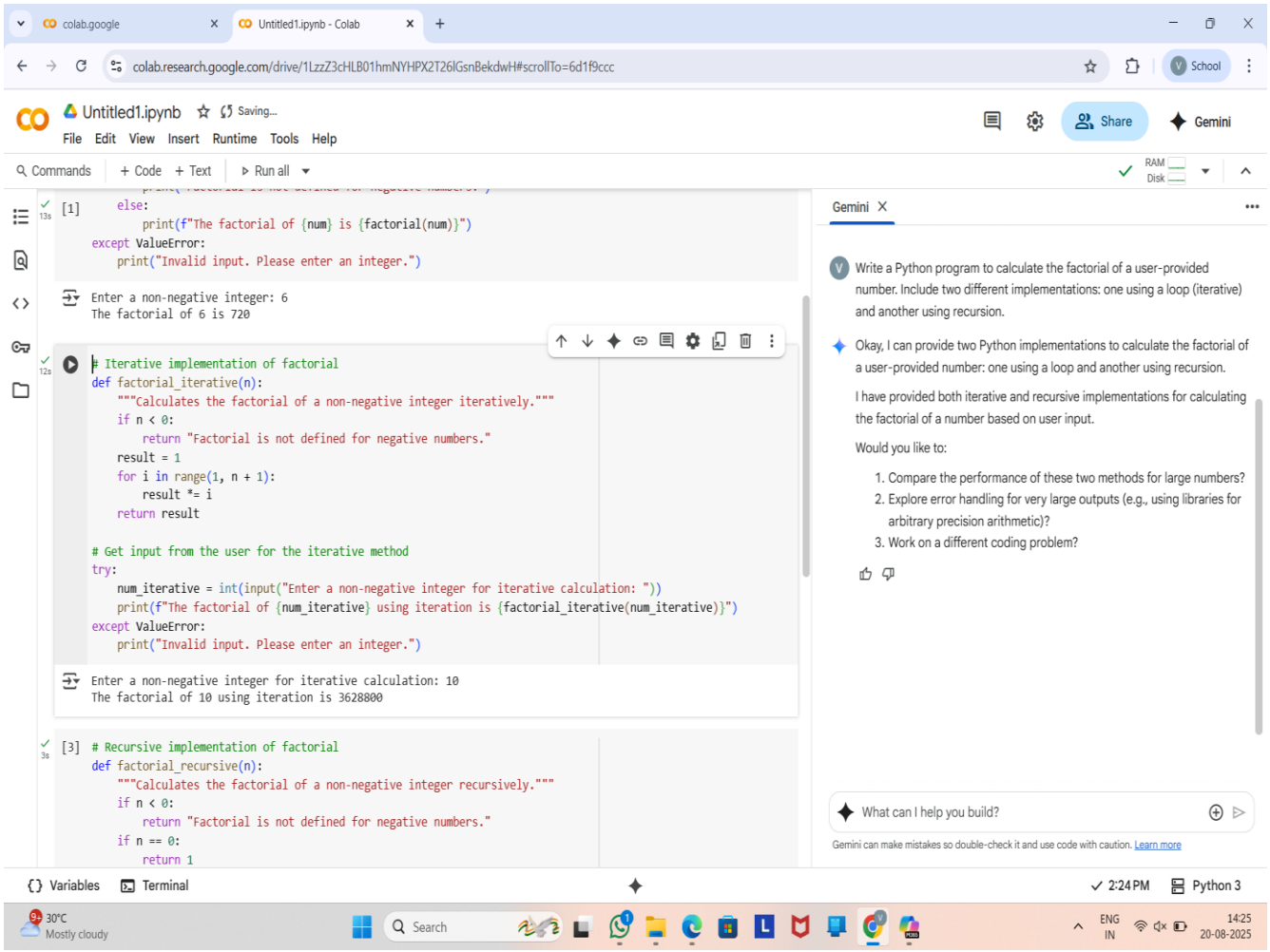
Code&Output:



Prompt-2: Write a Python program to calculate the factorial of a user-provided number. Include two

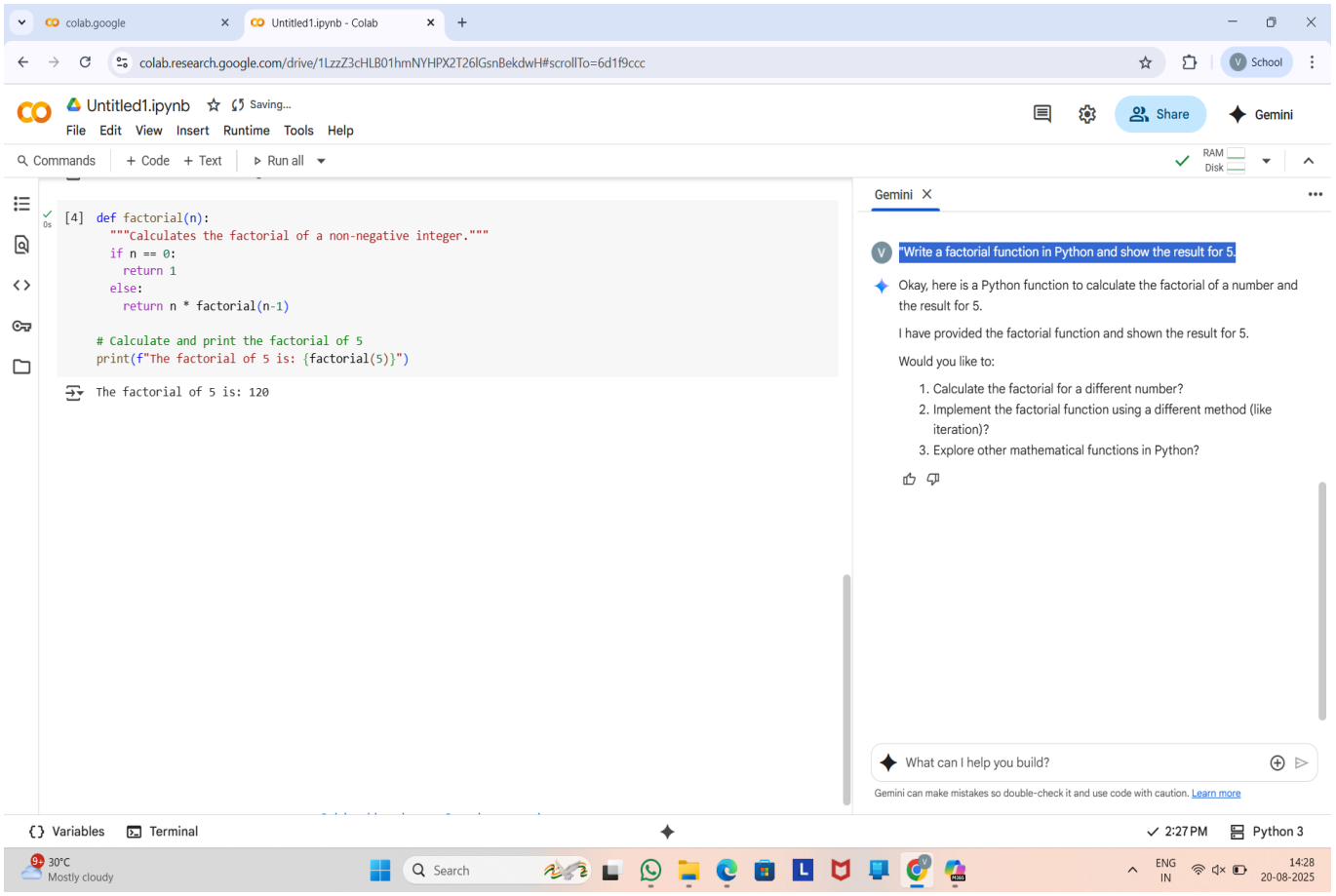
different implementations: one using a loop (iterative) and another using recursion.

Code&Output:



Prompt-3: Write a factorial function in Python and show the result for 5.

Code&Output:



Observation: All three methods give the same result for calculating factorial.

- The recursive method is great for learning, but not ideal for very large numbers due to stack limits.

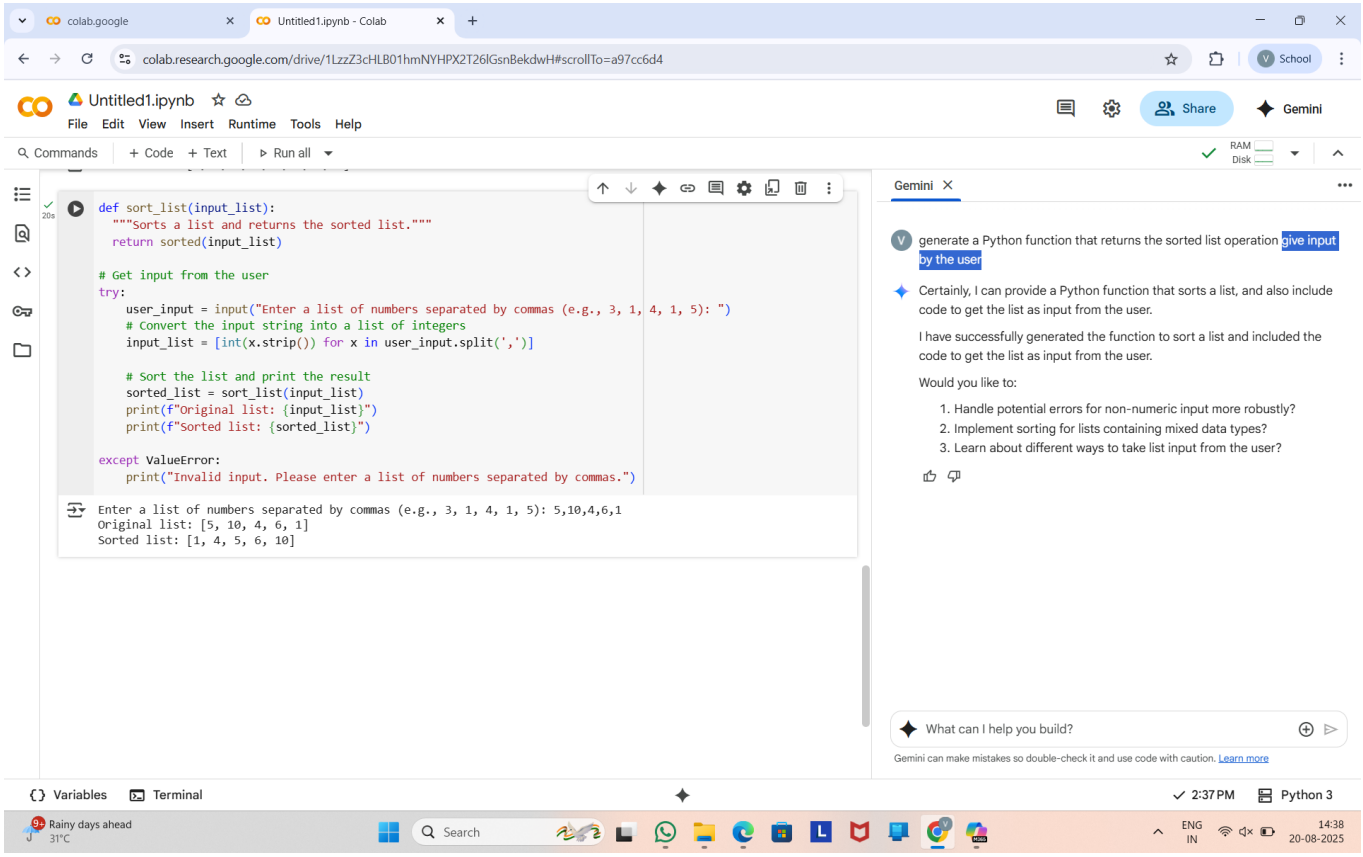
-The iterative method is practical and commonly used.

-The math module is best for performance and simplicity—just plug and play!

Task-2:

Prompt: Generate a Python function that returns the sorted list operation give input by the user

Code&Output:

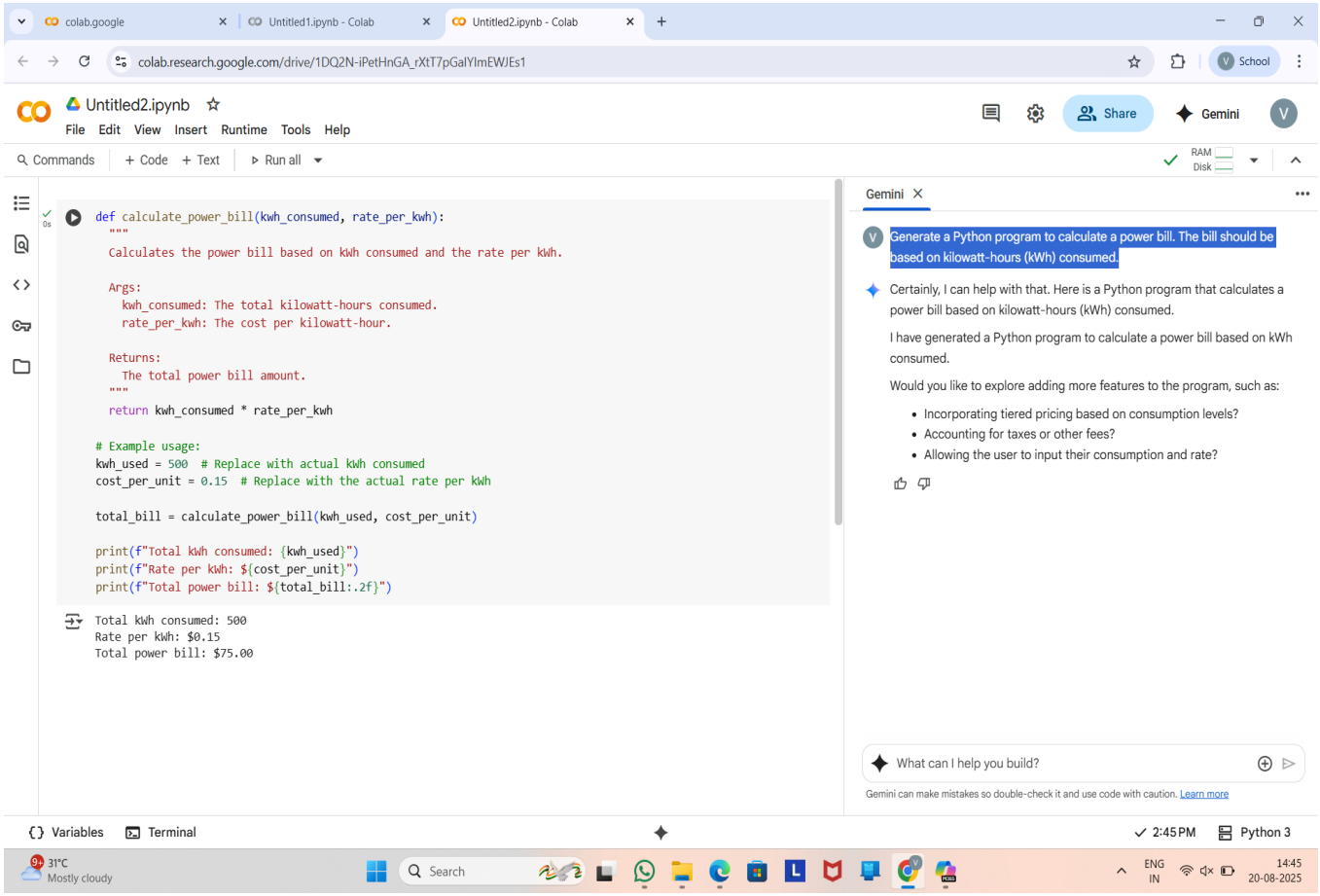


Observation: This Python code uses the Bubble Sort algorithm to sort a list of numbers in ascending order. It compares each pair of adjacent elements and swaps them if they're out of order. This process repeats until the entire list is sorted.

Task-3:

Prompt: Generate a Python program to calculate a power bill. The bill should be based on kilowatt-hours (kWh) consumed.

Code&Output:



Observation: -Prompt now specifies exact billing logic.

-Covers all necessary components of a realistic bill.

Code is ready to run and easy to adapt.

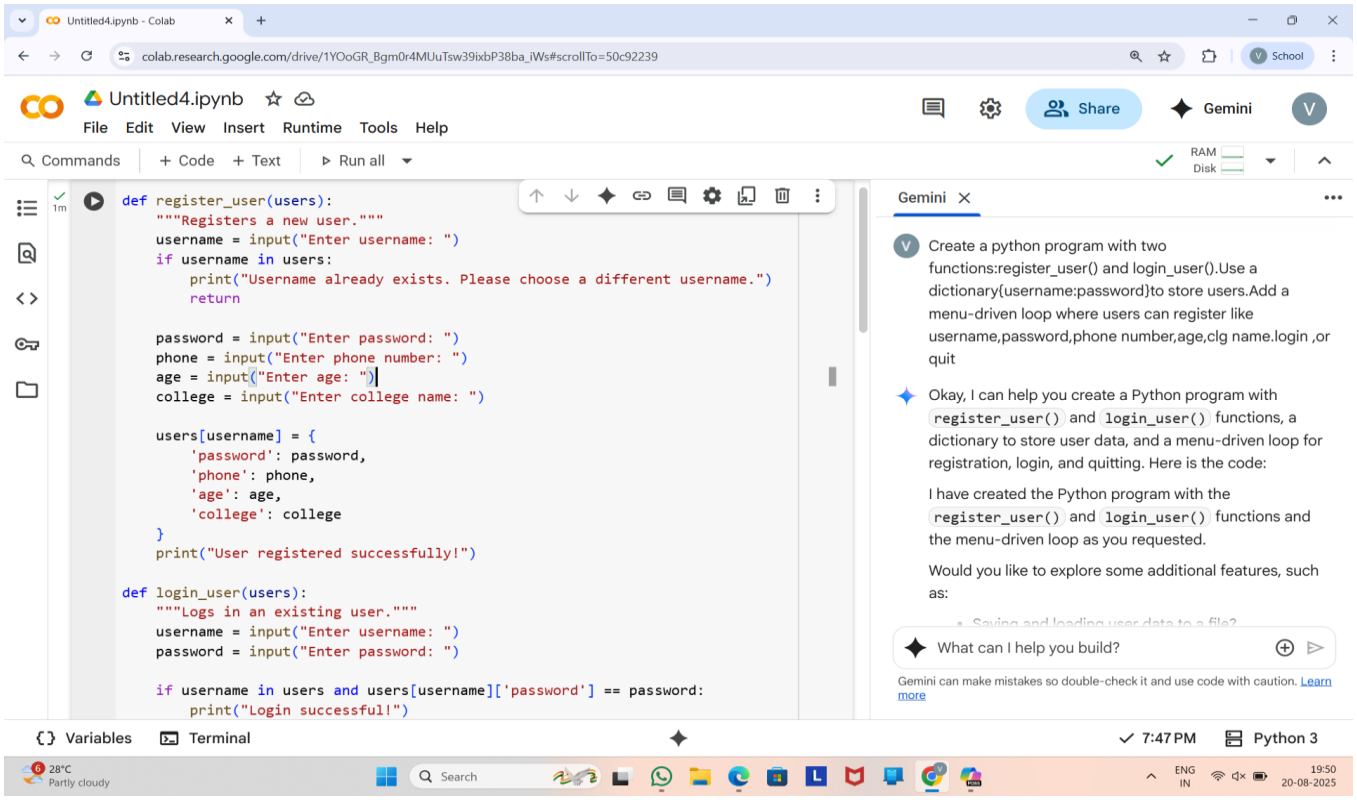
This Python function calculates a basic electricity bill by multiplying the number of

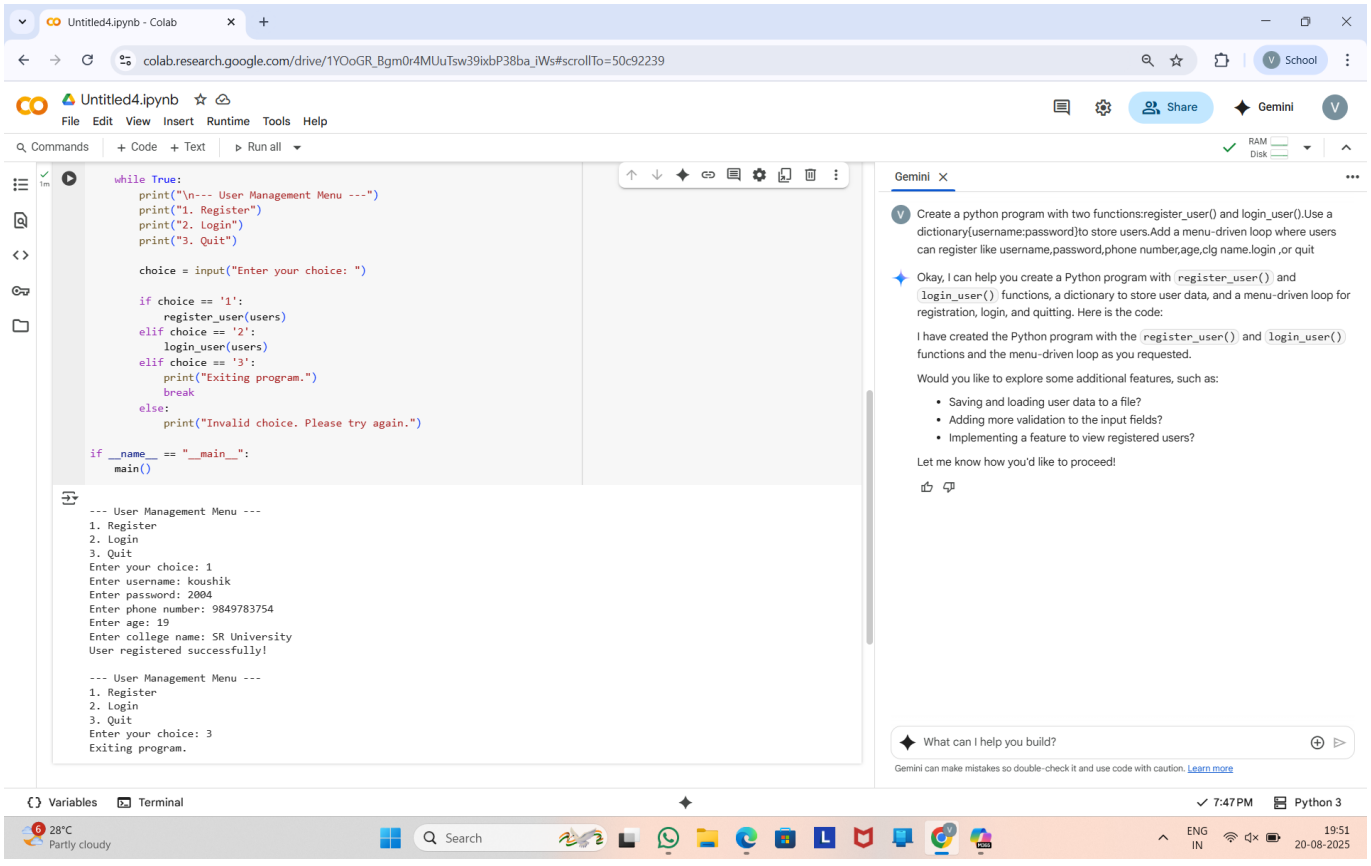
kilowatt-hours consumed (consumption_kwh) with the cost per kilowatt-hour (rate_per_kwh). It includes a clear docstring explaining its purpose and usage. The example shows a consumption of 500 kWh at a rate of $0.15 per kWh, resulting in a total bill of $75.00.

Task-4:

Prompt: Create a python program with two functions:register_user() and login_user().Use a dictionary{username:password}to store users.Add a menu-driven loop where users can register like username,password,phone number,age,clg name.login ,or quit

Code&Output:





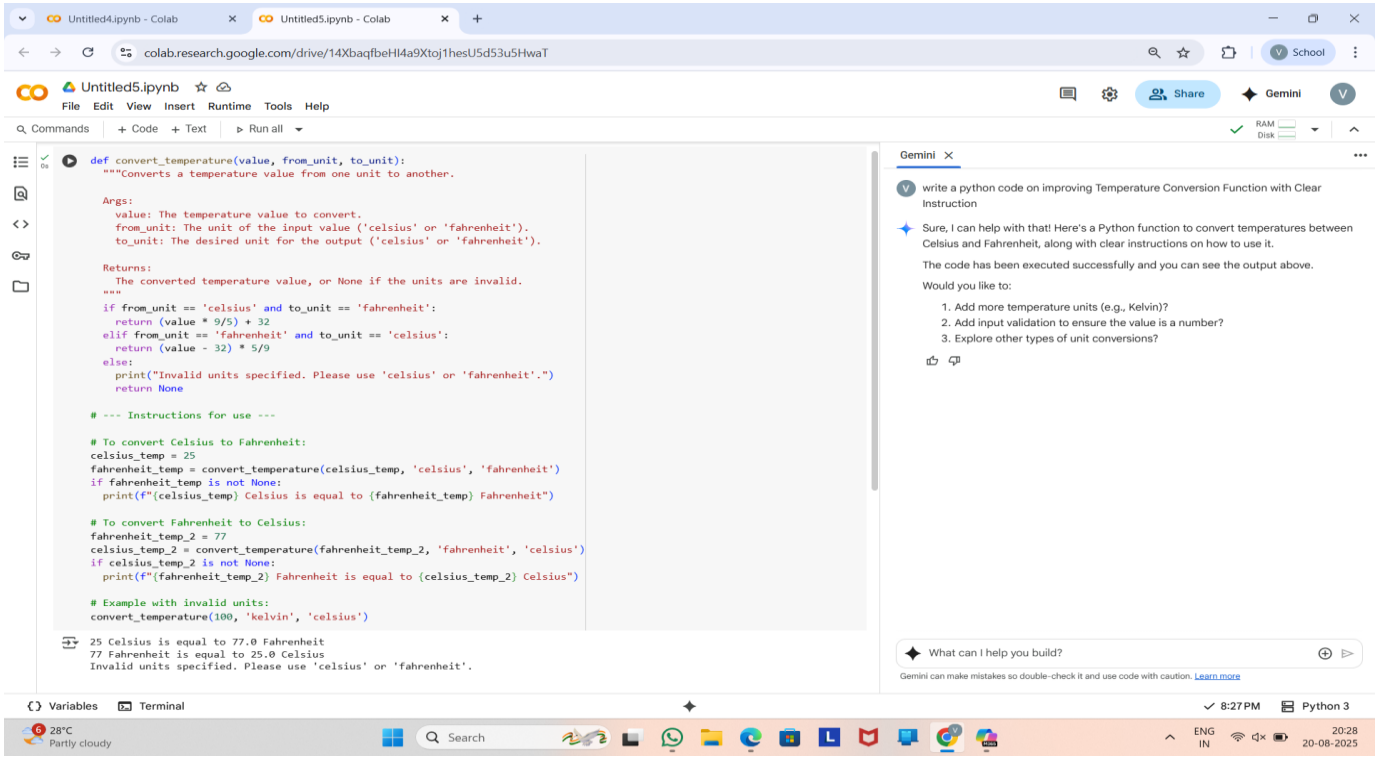
Observation: The code provides a simple command-line user management system with

registration and login functionality. It uses a dictionary (`users`) to store user data in memory. The structure is clear and functional for basic scenarios, but lacks features like data persistence (saving to a file), password hashing for security, and input validation (e.g., checking phone number format or age type).

Task-5:

Prompt: Write a python code on improving Temperature Conversion Function with Clear Instruction

Code&Output:



Observation: The `convert_temperature` function correctly converts temperatures between Celsius and Fahrenheit. It handles two valid conversion paths and returns `None` with an error message if the input units are invalid. Example usages are clear and demonstrate functionality, including an invalid case for robustness.