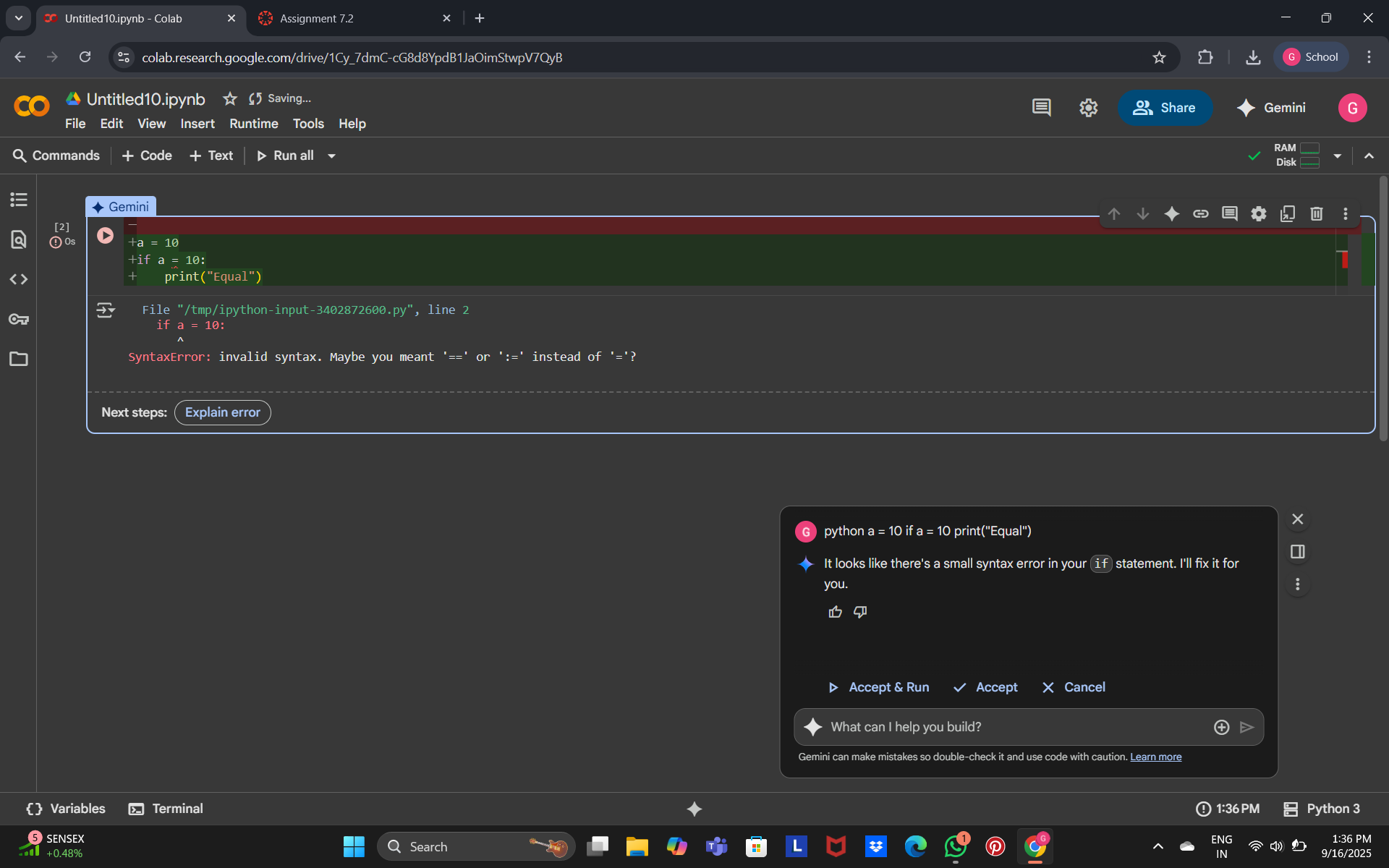
AI-ASSISTED CODING

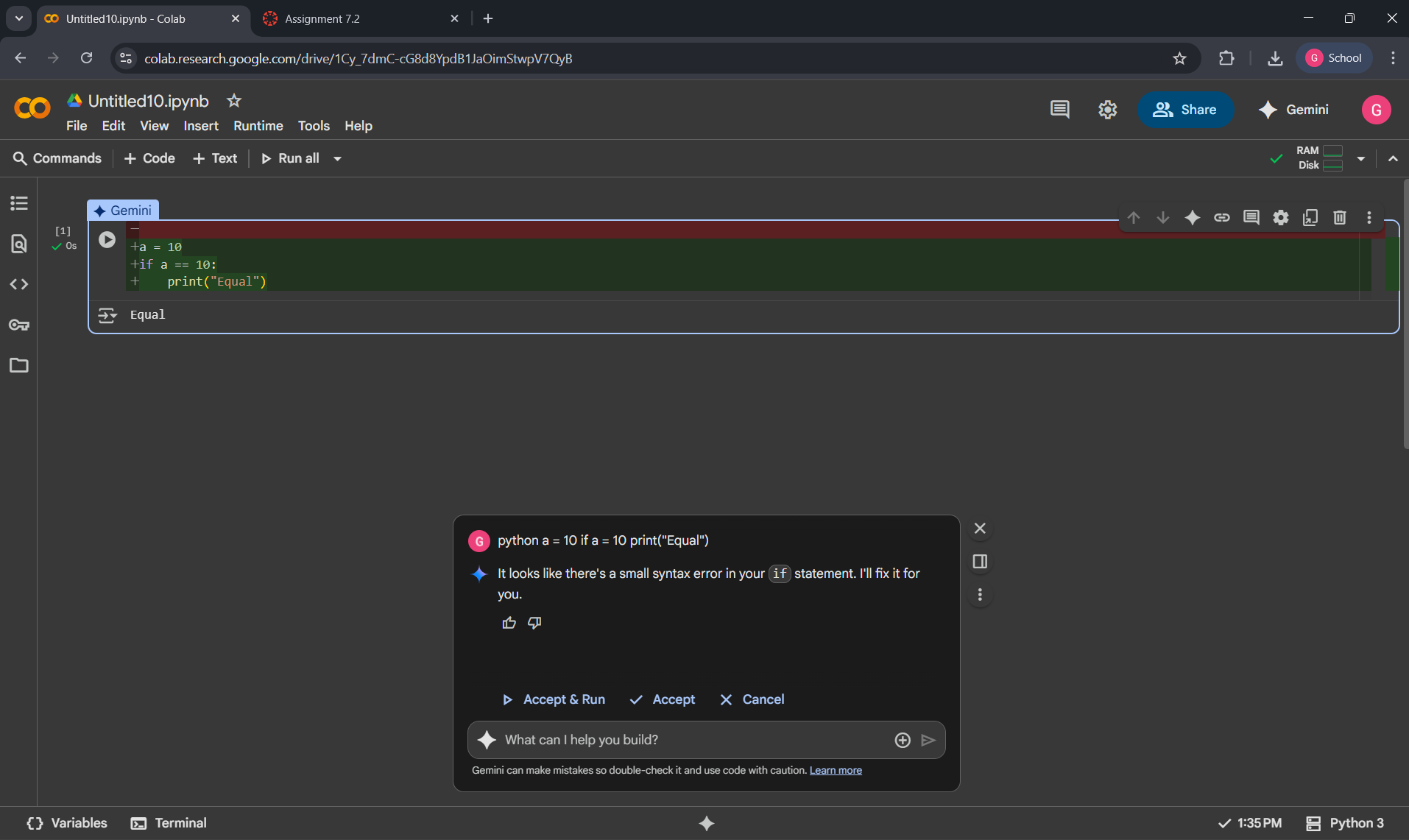
NAME : GUJJA RAJU

Ht. no.:2403A52018

ASSIGNMENT 7.2

TASK 1:

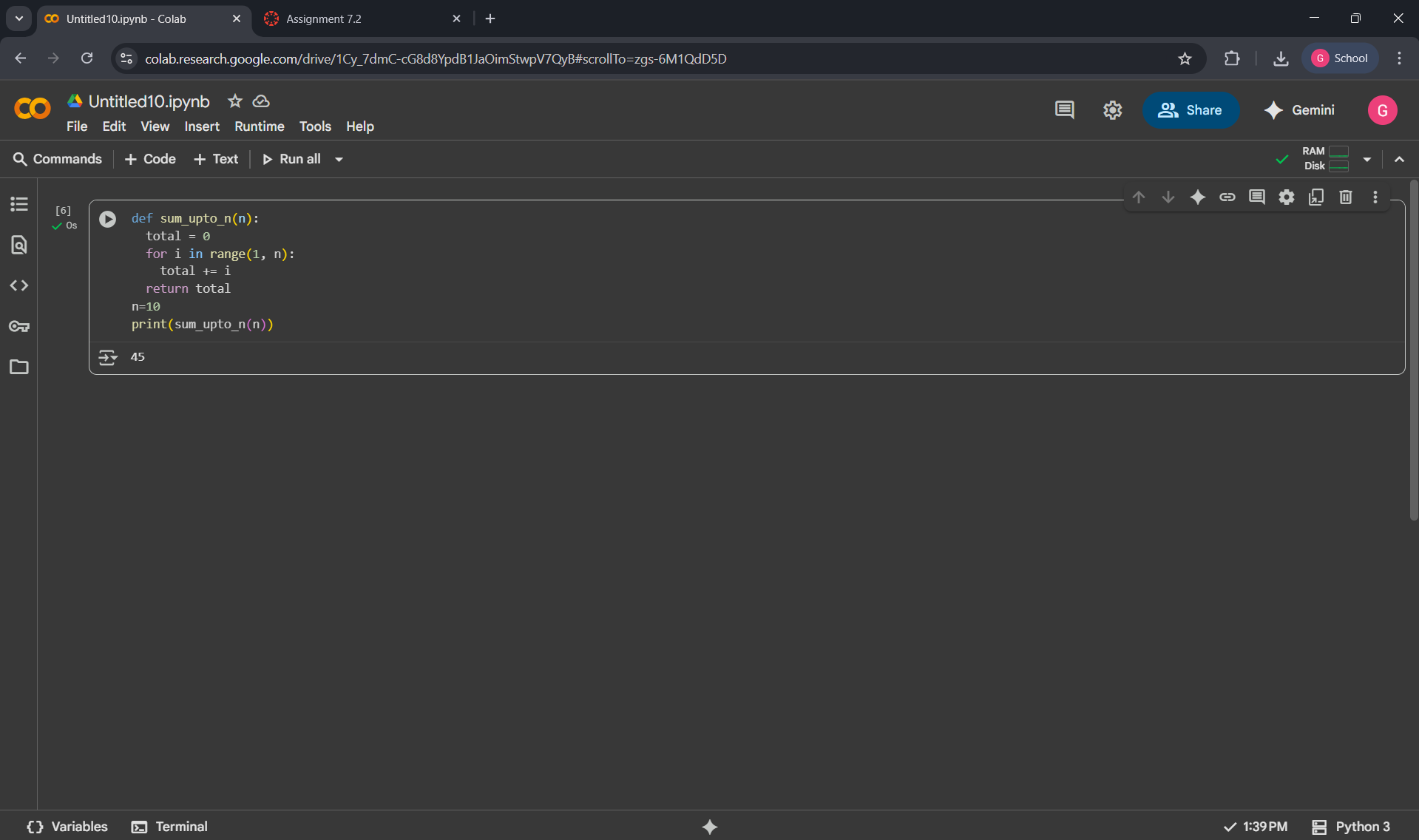


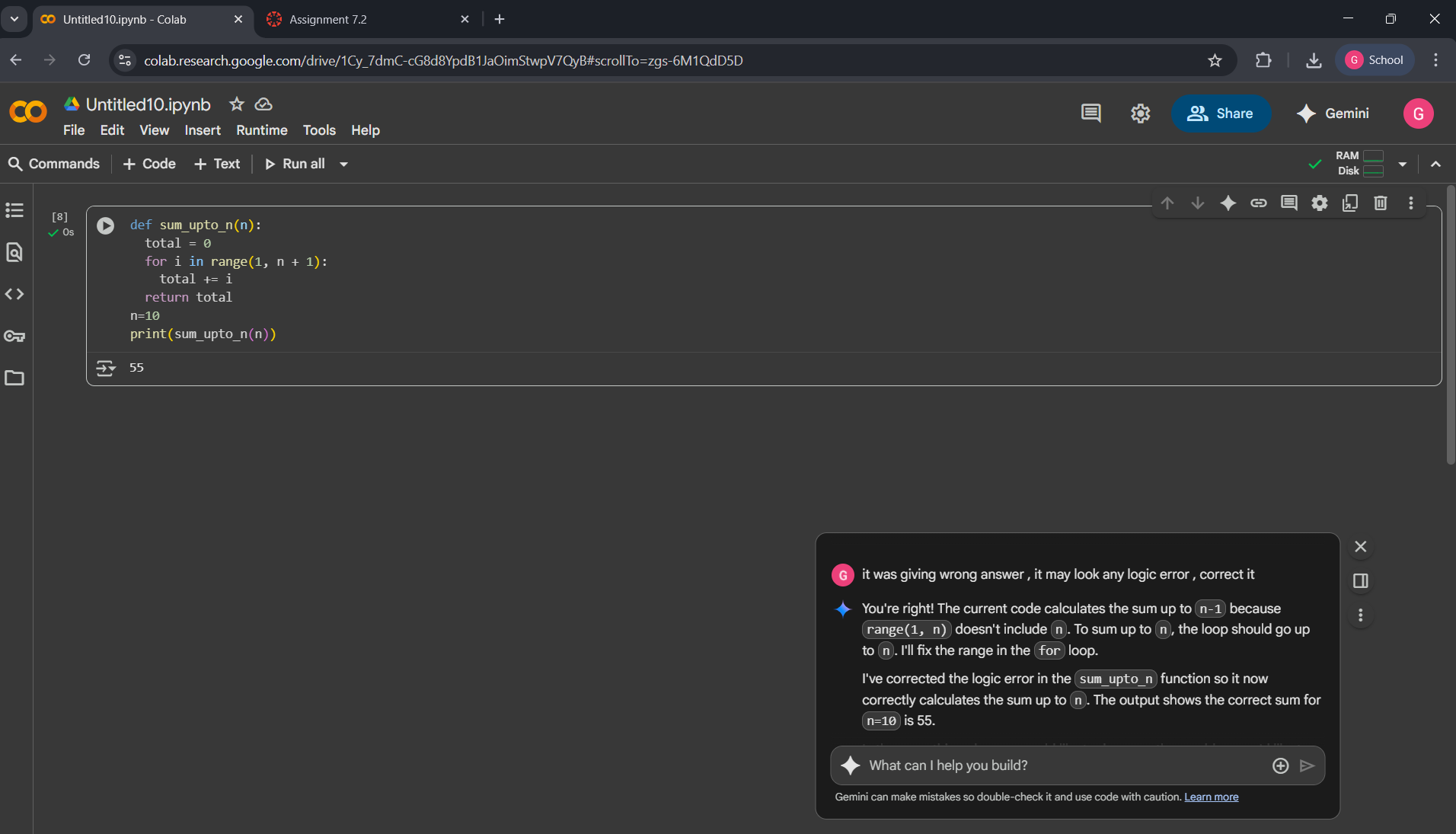


OBSERVATION:

There was small error in syntax at the if condition statement line , by giving code to GEMINI AI , the ai detected the error and corrected as shown in second image , The error happened because = (assignment) was wrongly used instead of == (comparison), after fixing both, the program runs successfully and prints *Equal*.

TASK 2:

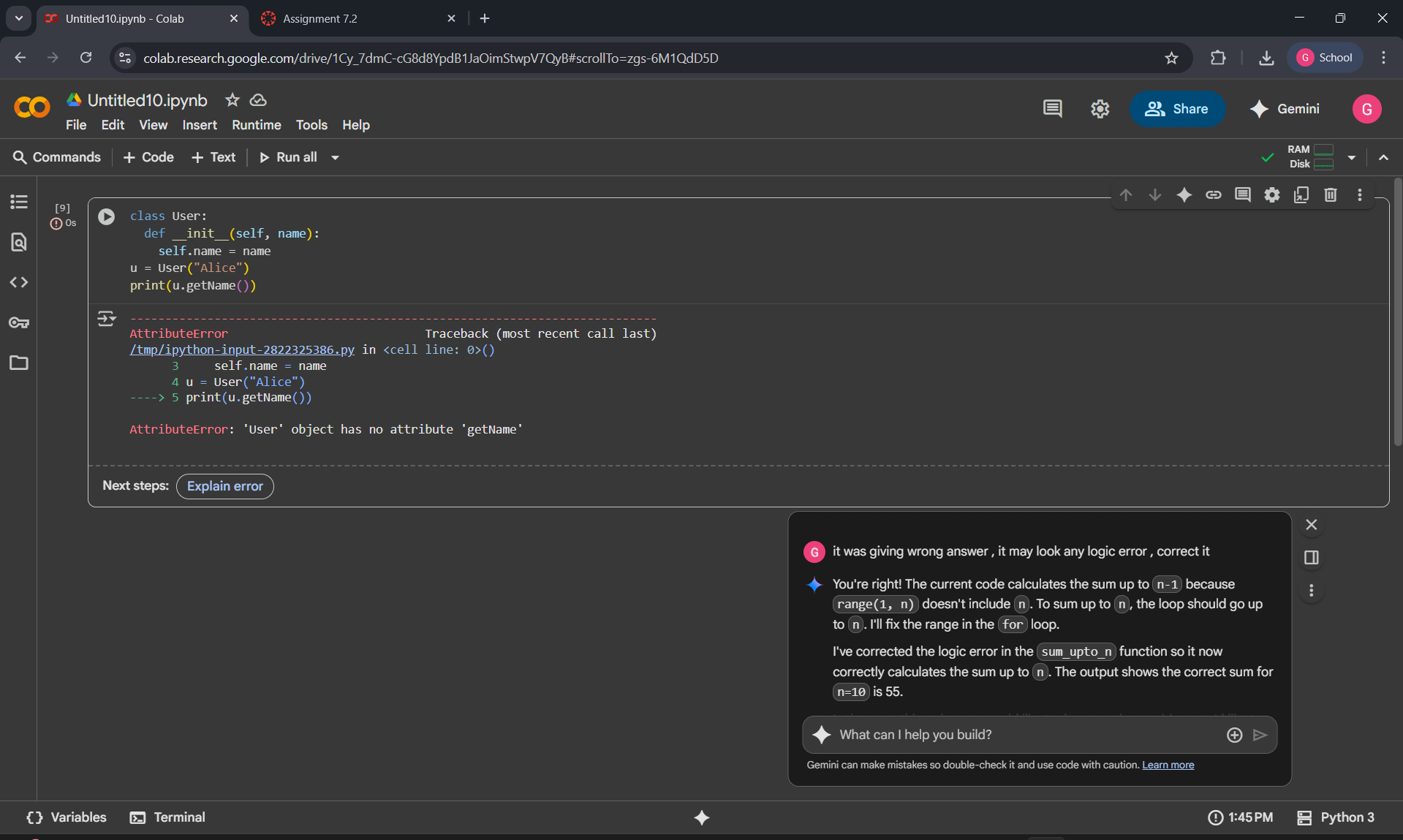


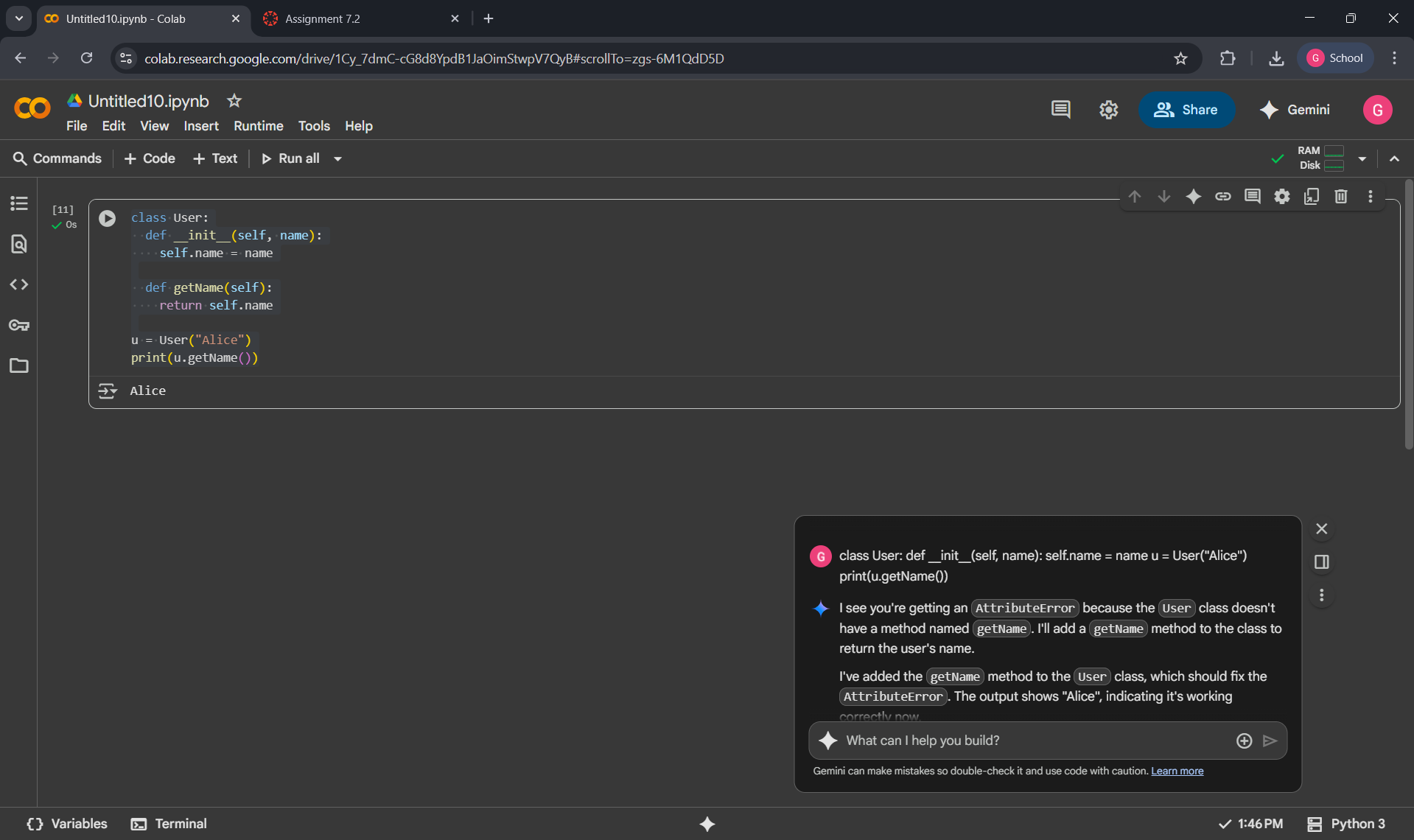


OBSERVATION :

Here , it was it have an error that the loop was closing without taking the ‘n’ number , closing at the before number only , so the output shown by given code was showing wrongly that sum upto before number only , when code given to GEMINI AI , it doesn’t captures error , as it was a simple logical error , so by giving above prompt , it was corrected.

TASK 3:

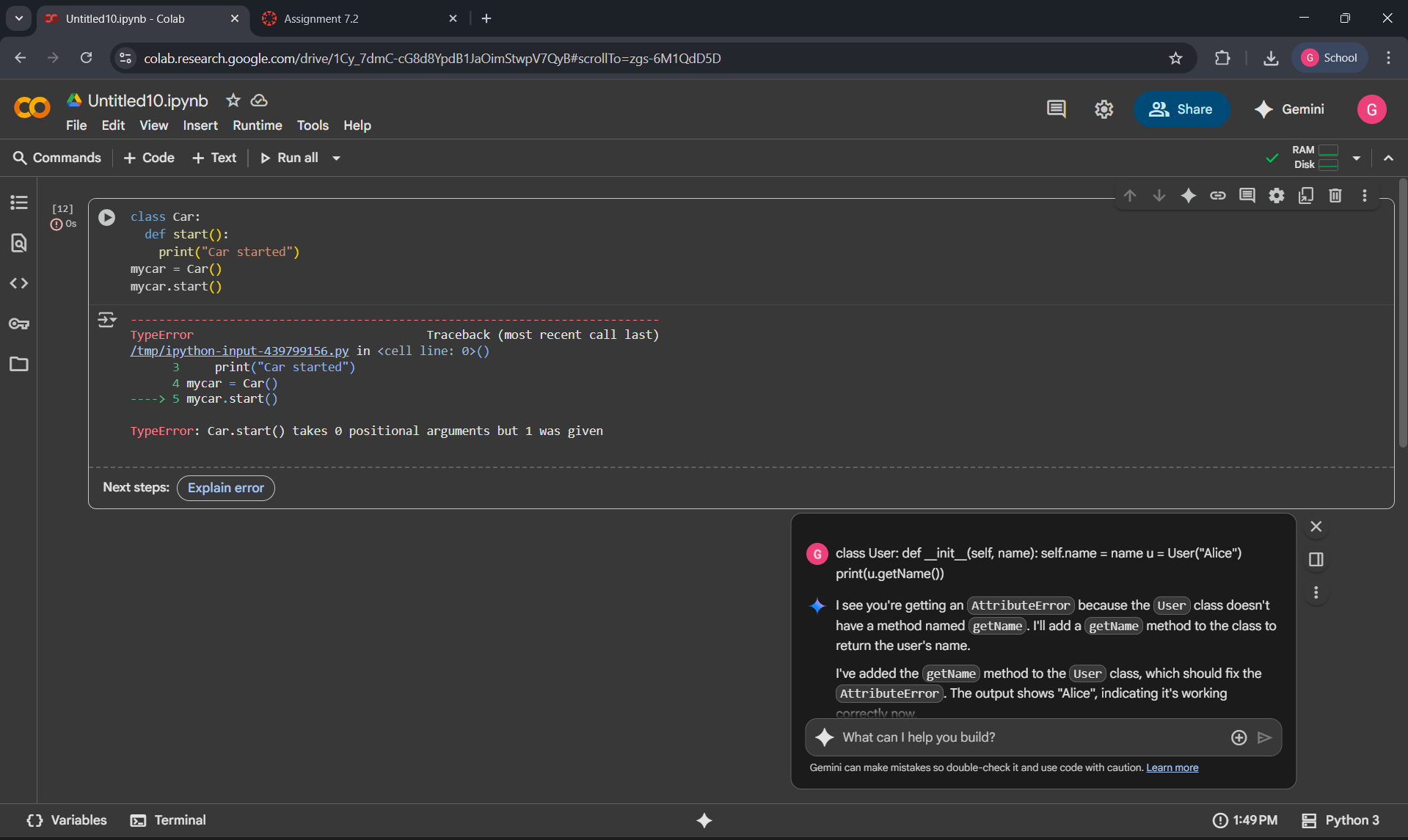


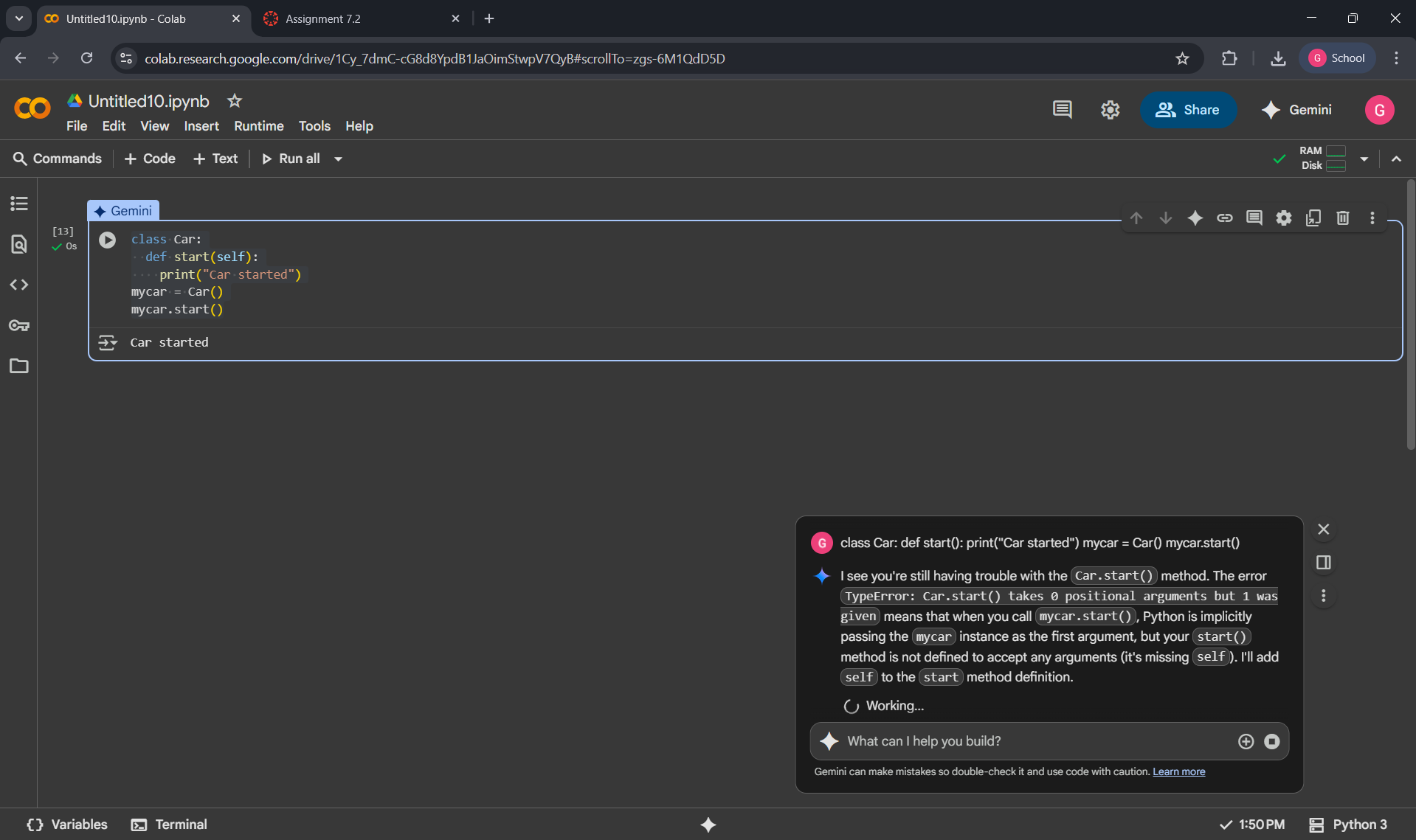


OBSERVATION :

The **first code** produced an AttributeError because the User class was missing the **getname()** method. The program was instructed to call a method that hadn't been defined, causing the error. The **second code** fixed this by adding the getname() method to the User class. This allowed the program to successfully and print the user's name, demonstrating the correct implementation of a class method.The gemini was corrected by giving directly as it was basic attribute error.

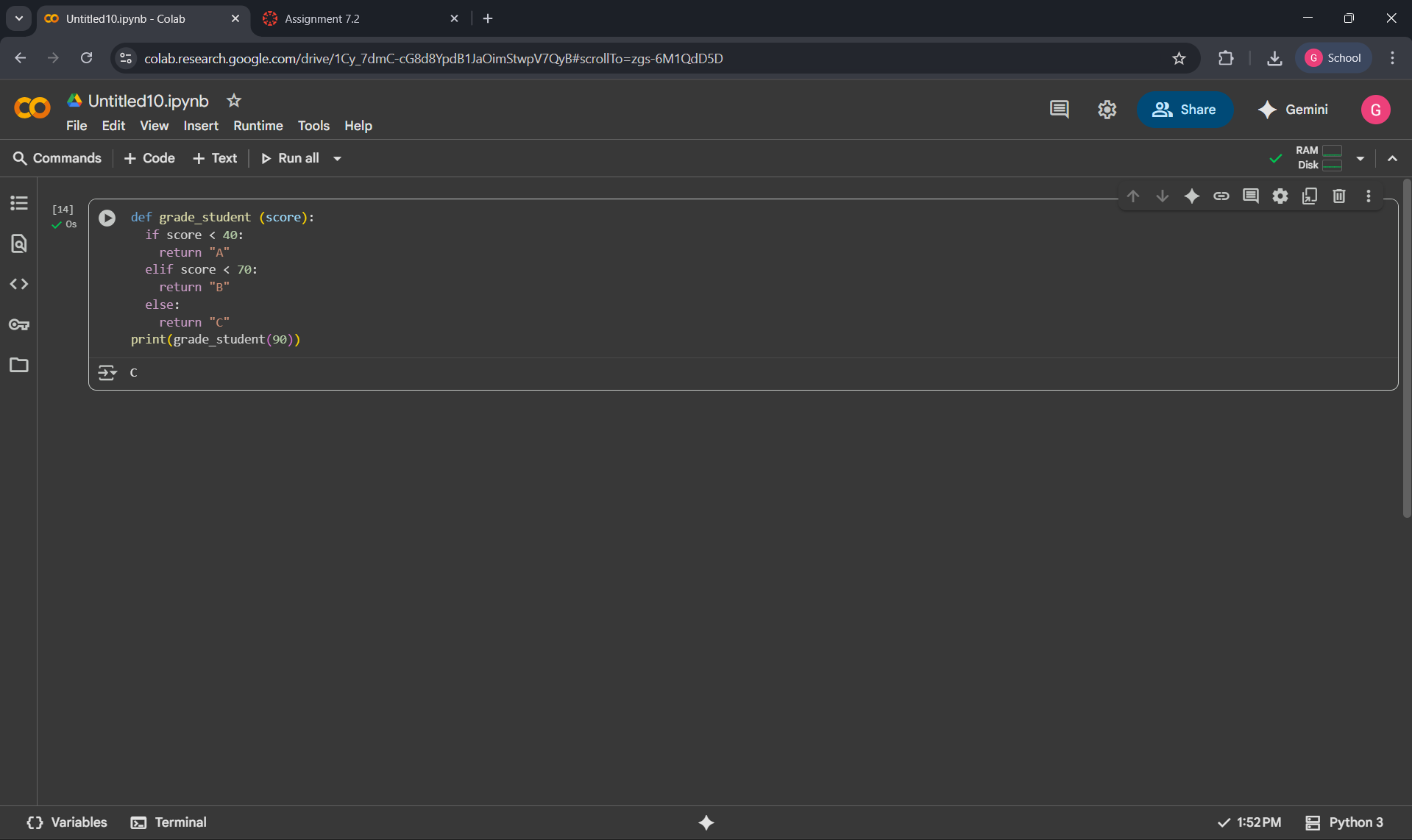
TASK 4:

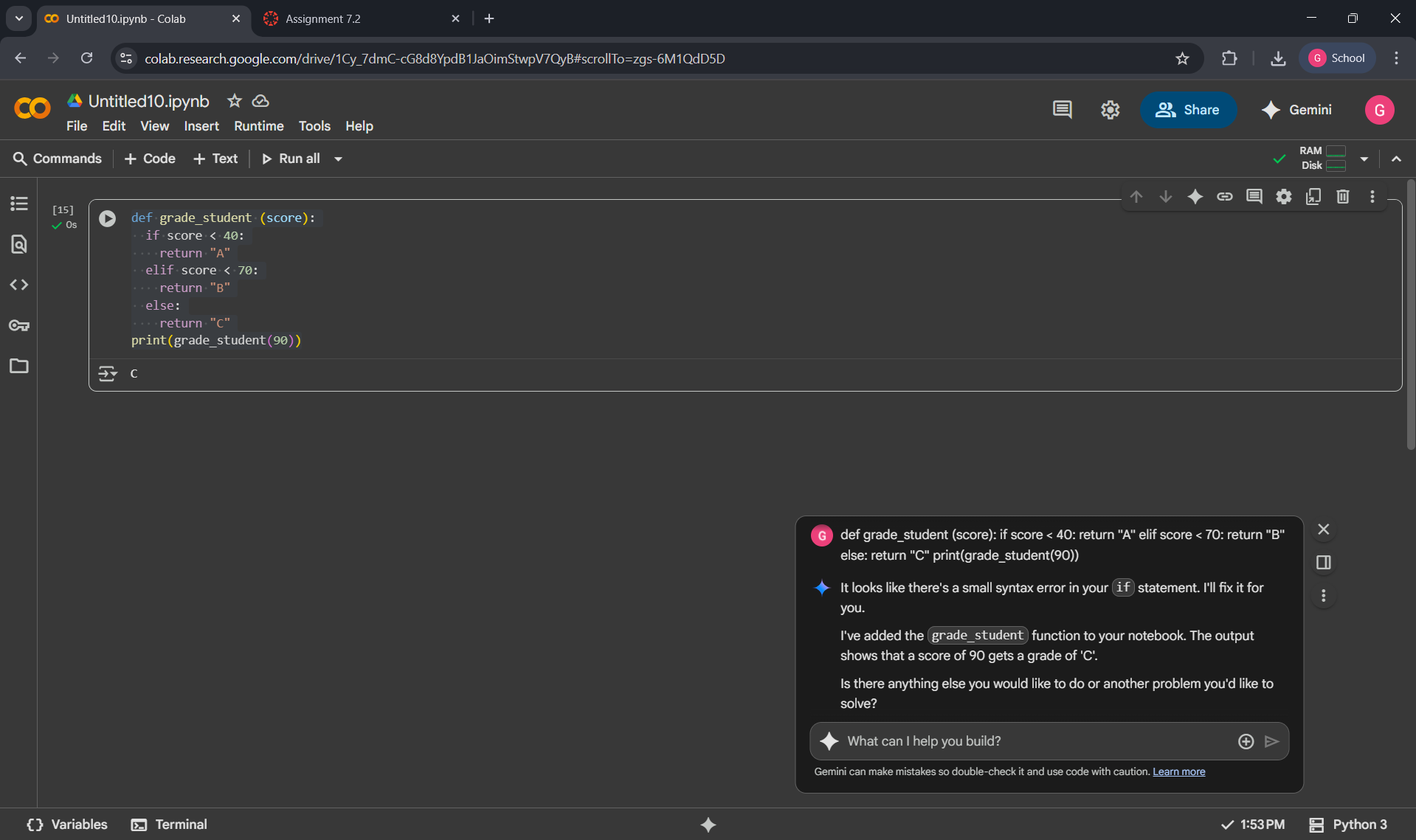


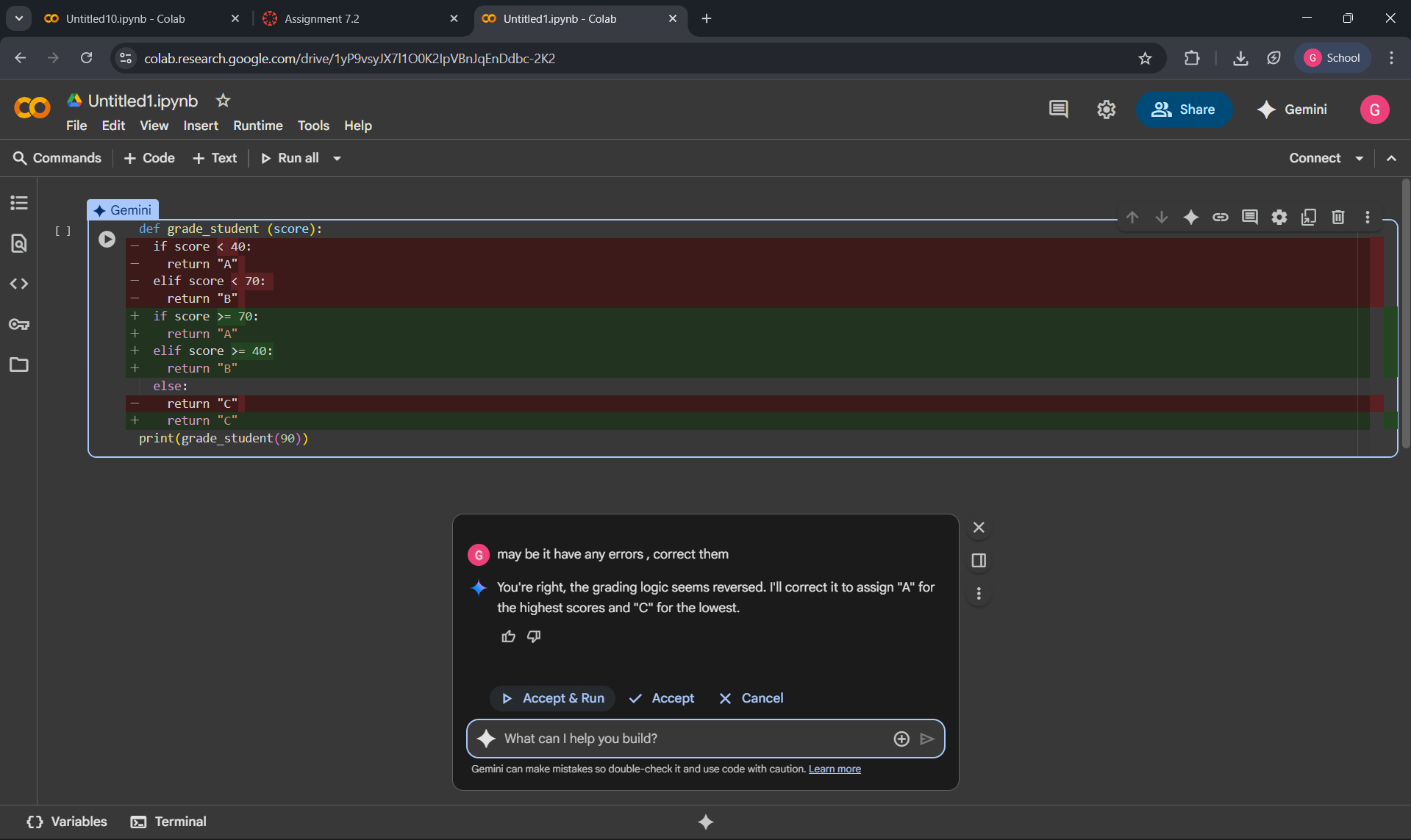
OBSERVATION:

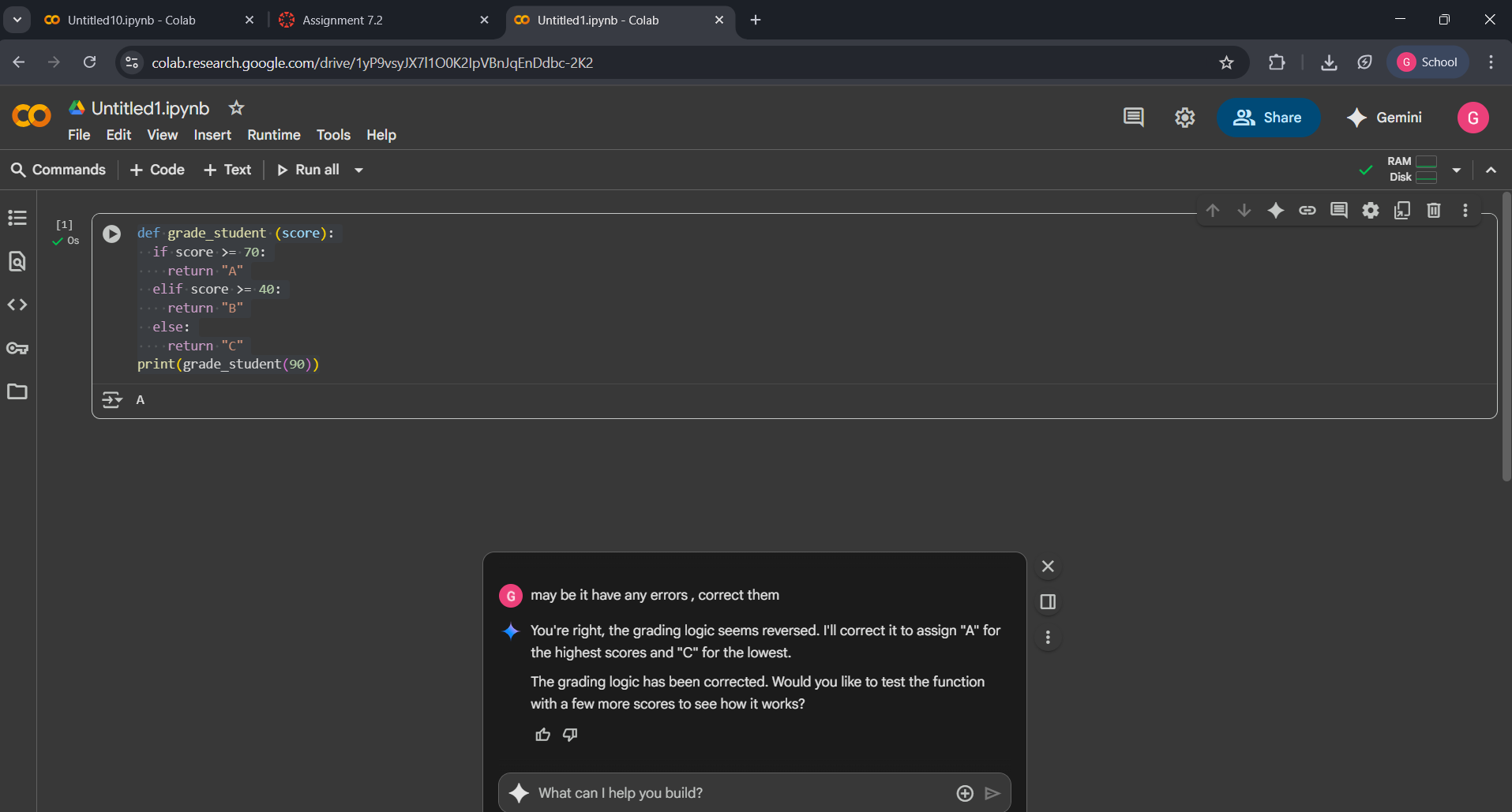
In the first code, the start() method raises a **TypeError** because it lacks the required self parameter. In Python, instance methods must include self to refer to the object. After updating the method to def start(self):, it runs correctly, and calling mycar.start() outputs **"Car started"** without errors.The error in given code was the basic attribute initialization , so the AI was corrected it directly by giving code to it.

TASK 5:









OBSERVATION:

Gemini does not initially capture the illogical grading error without a prompt. The first two codes show the code running as intended based on its current logic, giving a 'C' for a score of 90. It's only in the third image, after the us explicitly asks to check for errors, that Gemini identifies and corrects the reversed grading logic. The correction involves changing the conditions to award an 'A' for scores above 70, a 'B' for scores above 40, and a 'C' for anything else, fixing the illogical assignment.Whereas the gemini doesn’t captures illogical grading error the other AI platfroms like chatgpt captures error directly without prompt.