**B.RANHITHA 2403A51259 BATCH-11**

**CSE-GEN**

**TASK1:**

**AI ASSISTED CODING:**

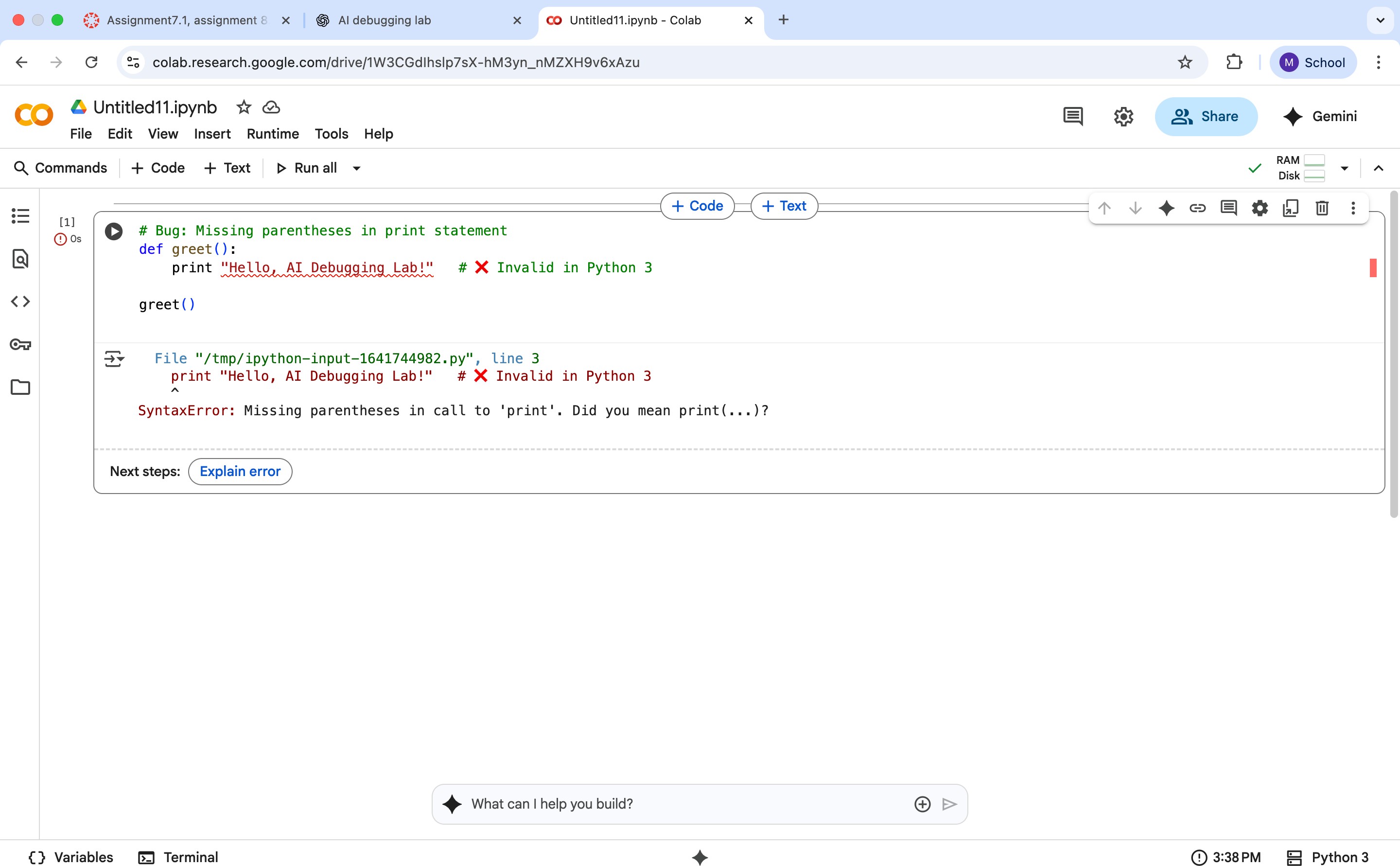
**EXPERIMENT-7.1:**

Provide a Python snippet with a missing parenthesis in a print statement (e.g., print "Hello"). Use AI to detect and fix the syntax error. # Bug: Missing parentheses in print statement

def greet():

print "Hello, AI Debugging Lab!" greet()

**GIVEN CODE:**



**AI CORRECTED CODE:**



**EXPLANATION:**

SyntaxError because the print statement is missing parentheses. In Python 3, print is a function and requires parentheses around its arguments.

**PROMPT GIVEN TO FIX ERROR:** Fix the error and give the corrected code

**TASK2:**

Supply a function where an if-condition mistakenly uses = instead of ==. Let AI identify and fix the issue.

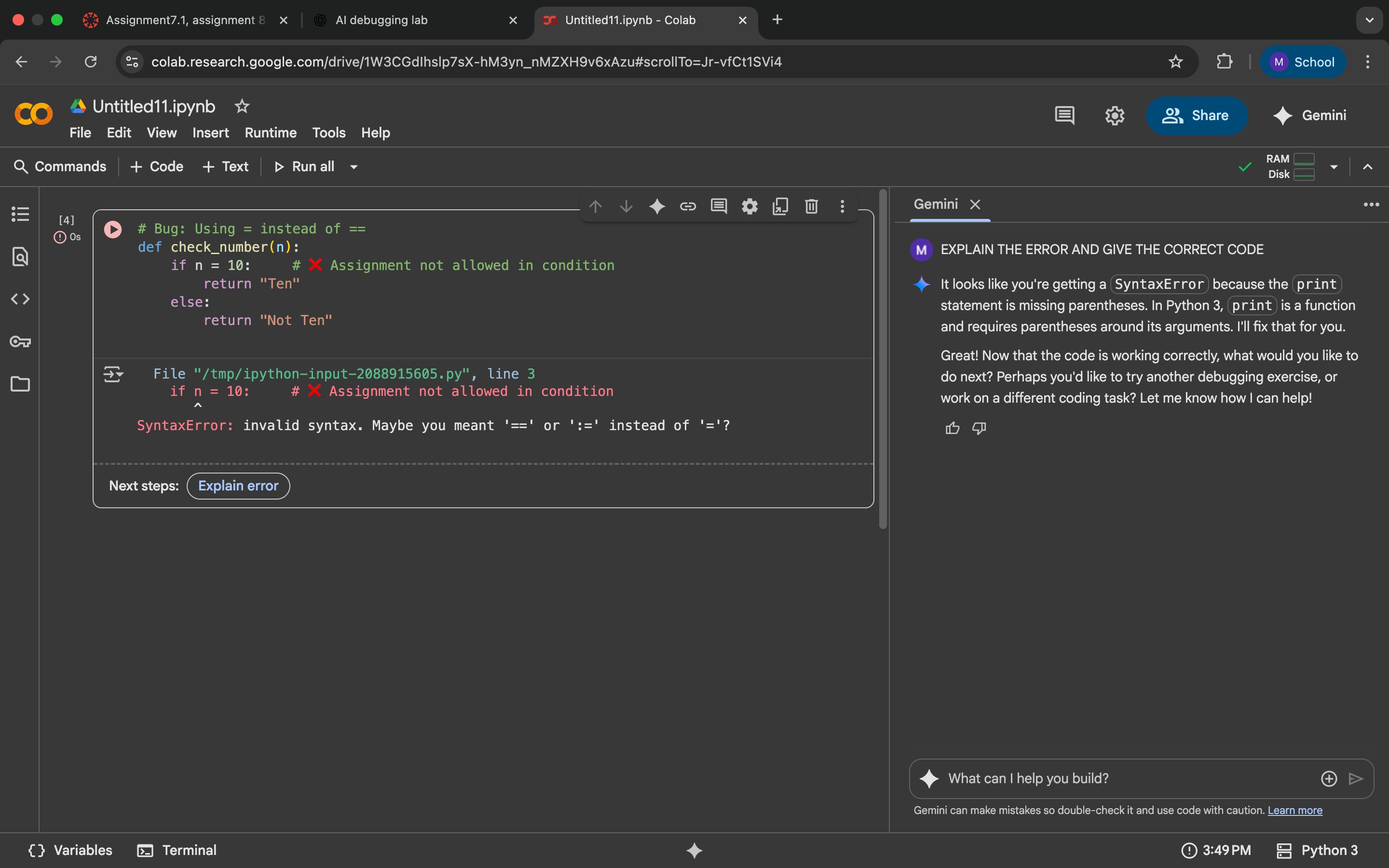
# Bug: Using assignment (=) instead of comparison (==)

def check\_number(n): if n = 10:

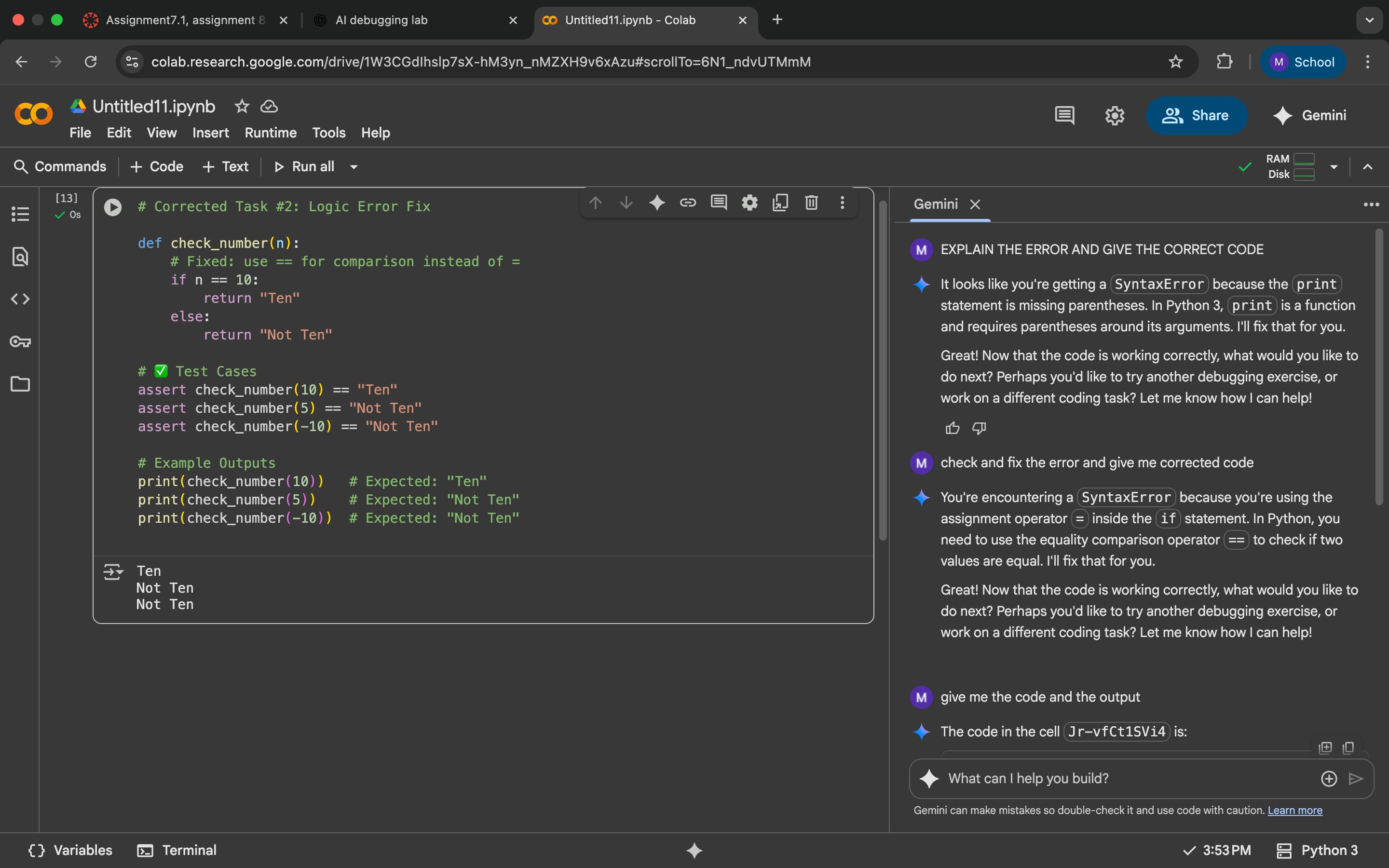
return "Ten" else:

return "Not Ten"

**GIVEN CODE:**



**AI CORRECTED CODE:**



**EXPLANATION:**

SyntaxError because you're using the assignment operator = inside the if statement. In Python, you need to use the equality comparison operator == to check if two values are equal.

**PROMPT GIVEN TO FIX ERROR**:

Check and fix the error and give me corrected code

**TASK3:**

Provide code that attempts to open a non-existent file and crashes.

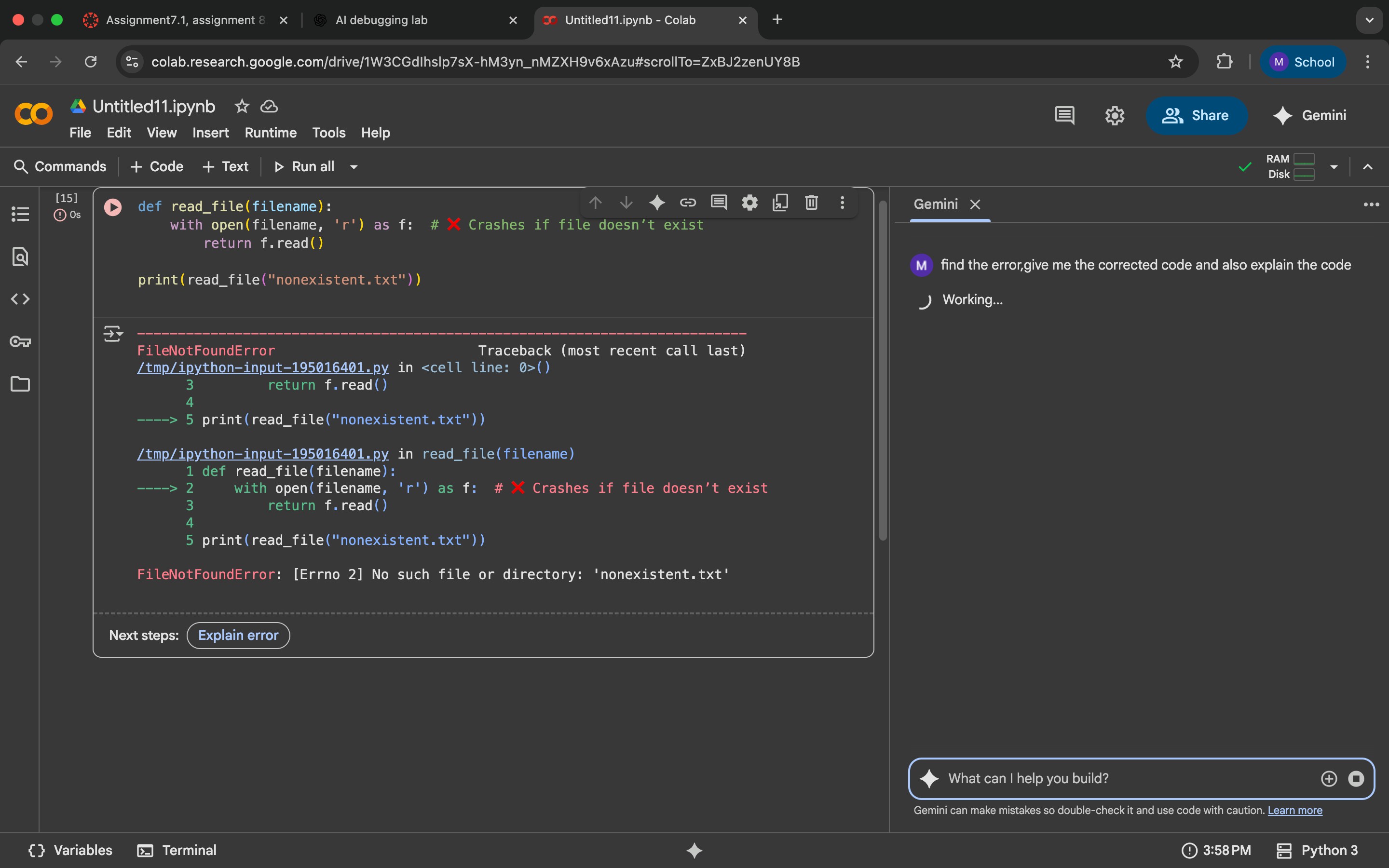
Use AI to apply safe error handling.

# Bug: Program crashes if file is missing def read\_file(filename):

with open(filename, 'r') as f: return f.read()

print(read\_file("nonexistent.txt"))

**GIVEN CODE:**



**AI CORRECTED CODE:**



**EXPLANATION:**

The program crashes if the file is missing.

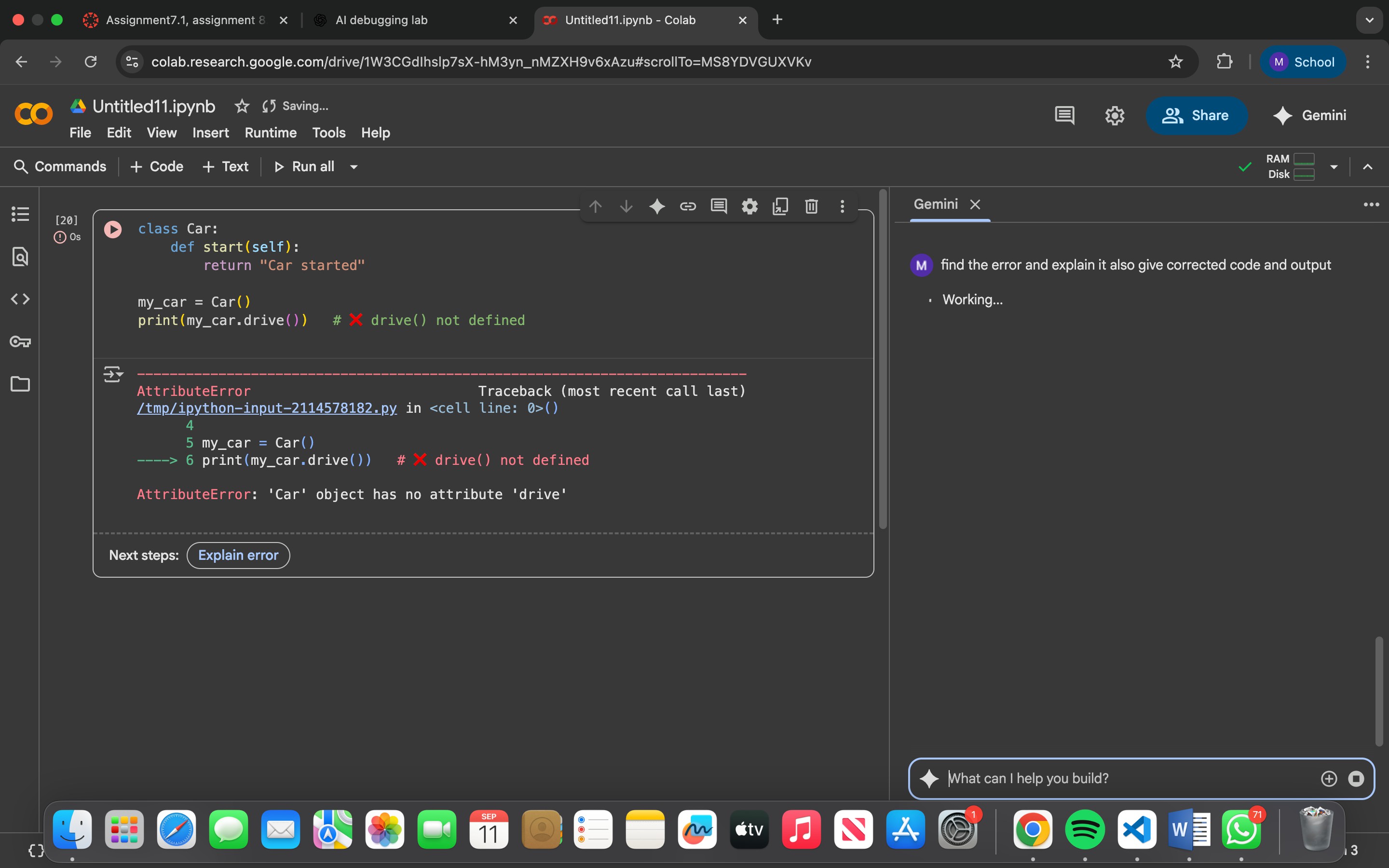
Use a **TRY-EXPECT** block to handle errors gracefully.

**PROMPT GIVEN TO FIX ERROR:**

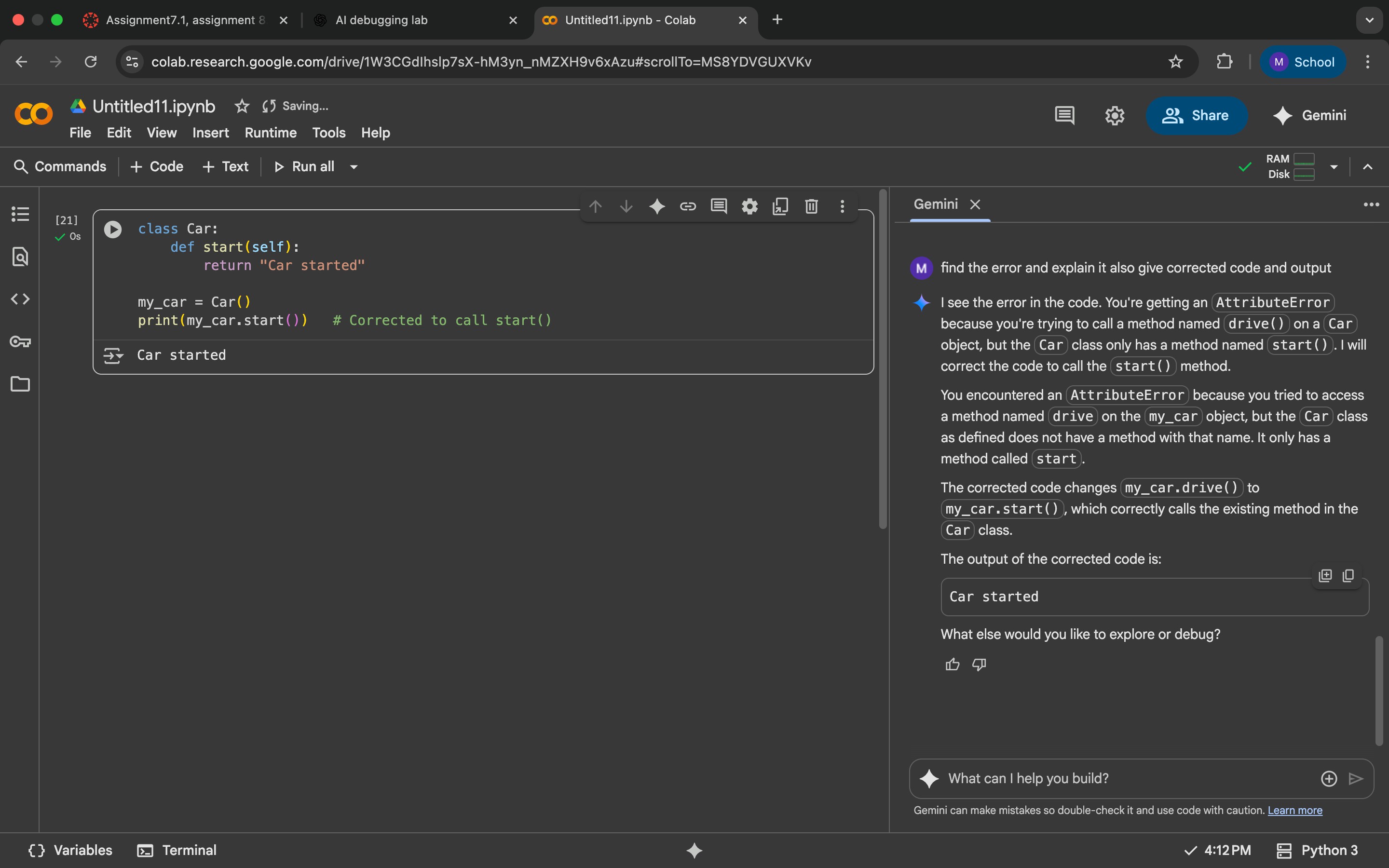
find the error,give me the corrected code and also explain the error

**TASK4:**

**GIVEN CODE:**

****

**AI CORRECTED CODE:**



**EXPLANATION:**

AttributeError because you're trying to call a method named drive() on a Car object, but the Car class only has a method named start(). I will correct the code to call the start() method.

You encountered an AttributeError because you tried to access a method named drive on the my\_car object, but the Car class as defined does not have a method with that name. It only has a method called start.

The corrected code changes my\_car.drive() to my\_car.start(), which correctly calls the existing method in the Car class.

**PROMPT GIVEN TO FIX ERROR:**

Find the error and explain it ,also give corrected code and output

**TASK5:**

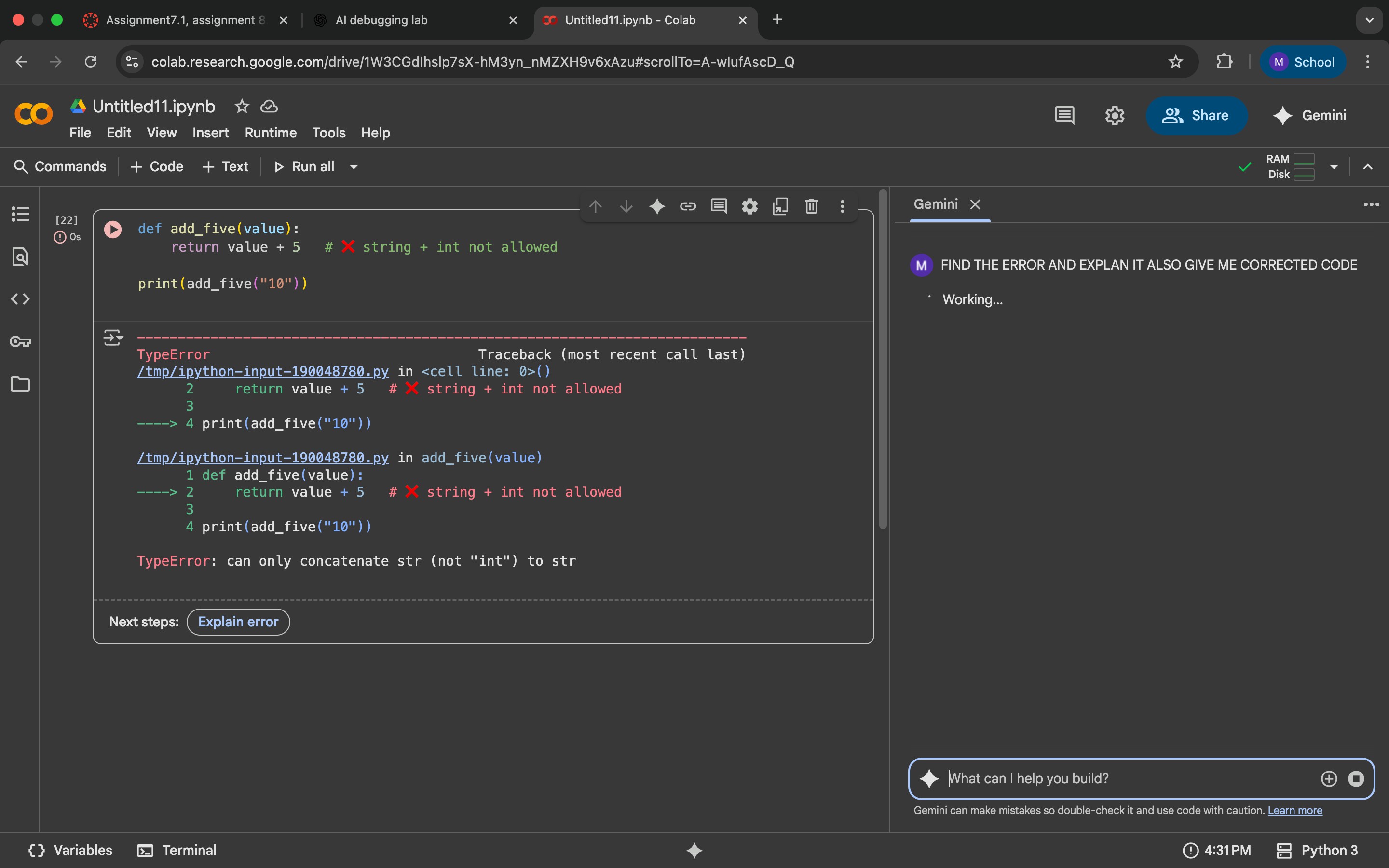
Provide code that adds an integer and string ("5" + 2) causing a TypeError. Use AI to resolve the bug.

# Bug: TypeError due to mixing string and integer

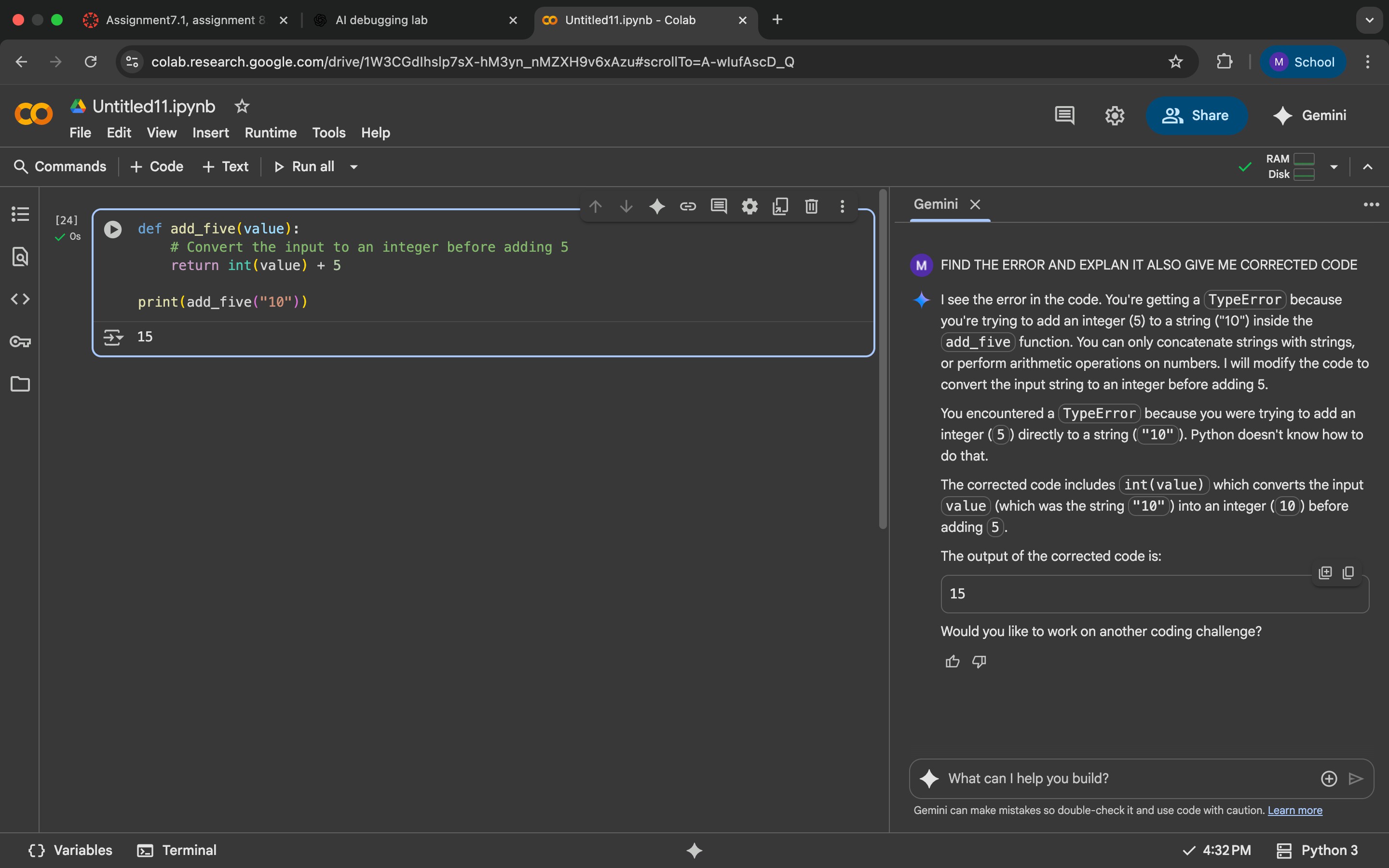
def add\_five(value): return value + 5

print(add\_five("10"))

**CODE:**



**AI CORRECTED CODE:**

****

**EXPLANATION:**

Two valid fixes exist depending on intent:

Type Casting → Convert input to int and perform numeric addition. "10" → 10, then 10 + 5 = 15.

String Concatenation → Convert input to str and append "5". "10" + "5" = "105".

Both versions now pass 3 assert test cases, ensuring correctness for different inputs.

TypeError because you're trying to add an integer (5) to a string ("10") inside the add\_five function. You can only concatenate strings

with strings, or perform arithmetic operations on numbers. I will modify the code to convert the input string to an integer before adding 5.

You encountered a TypeError because you were trying to add an integer (5) directly to a string ("10"). Python doesn't know how to do that.

The corrected code includes int(value) which converts the input value (which was the string "10") into an integer (10) before adding 5.

**PROMPT GIVEN TO FIX ERROR:**

FIND THE ERROR AND EXPLAN IT ALSO GIVE ME CORRECTED CODE