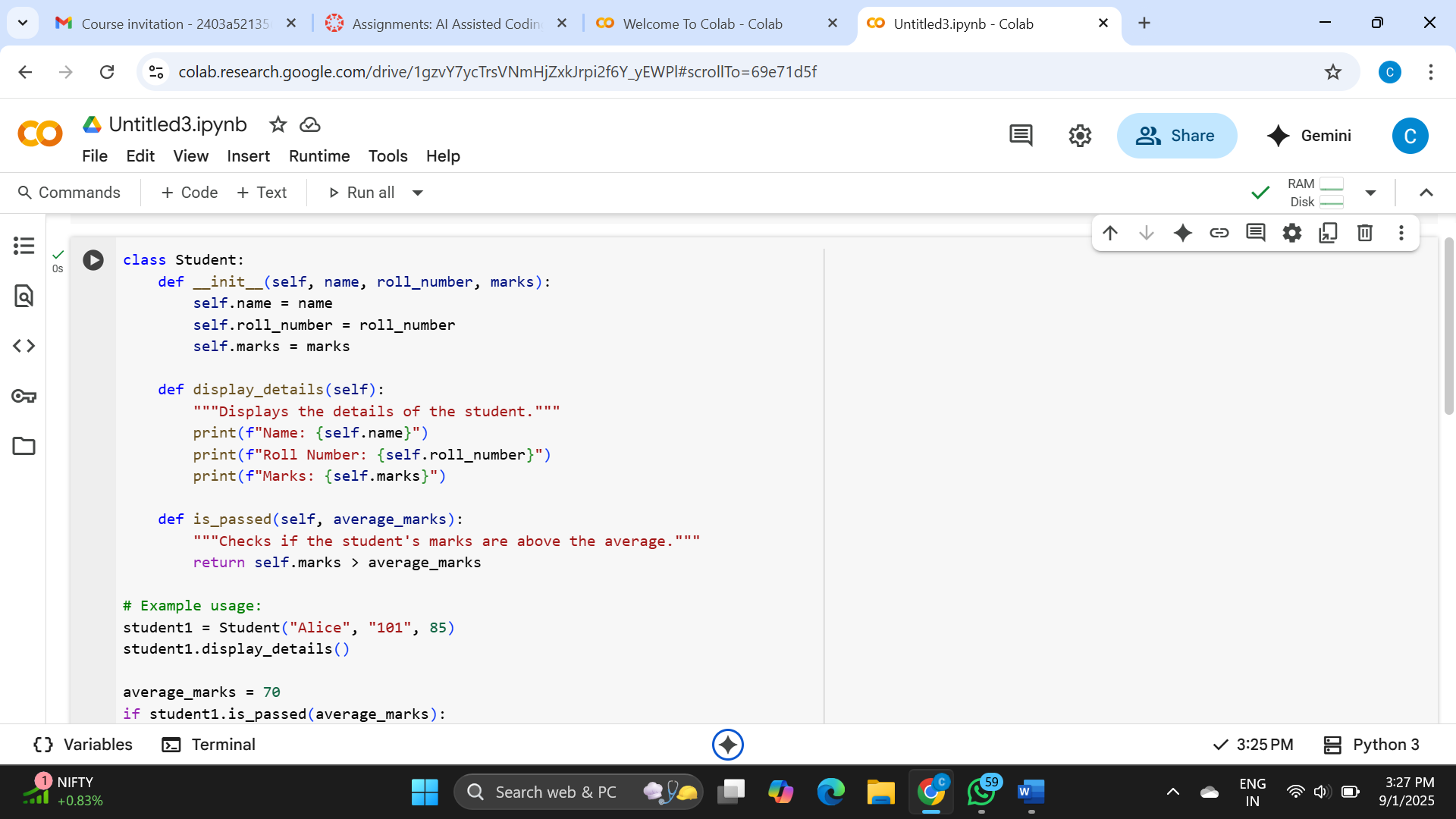
Assignment-6

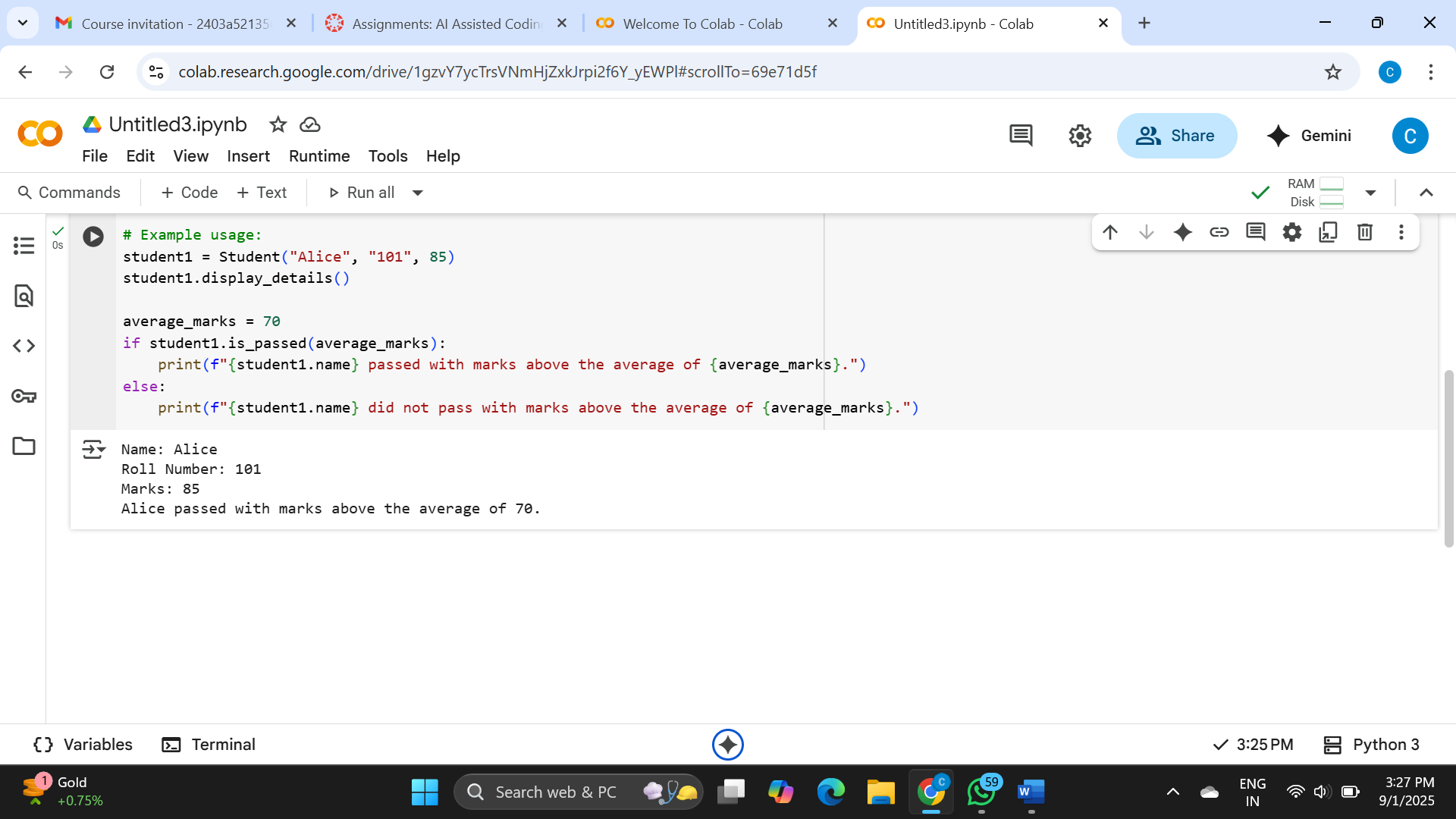
Task-1

Start a Python class named Student with attributes name, roll\_number, and marks. Prompt GitHub Copilot to complete methods for displaying details and checking if marks are above average.

**Expected Outcome #1:**

**•** Completed class with Copilot-generated methods like display\_details() and is\_passed(), demonstrating use of if-else conditions





Explanation:

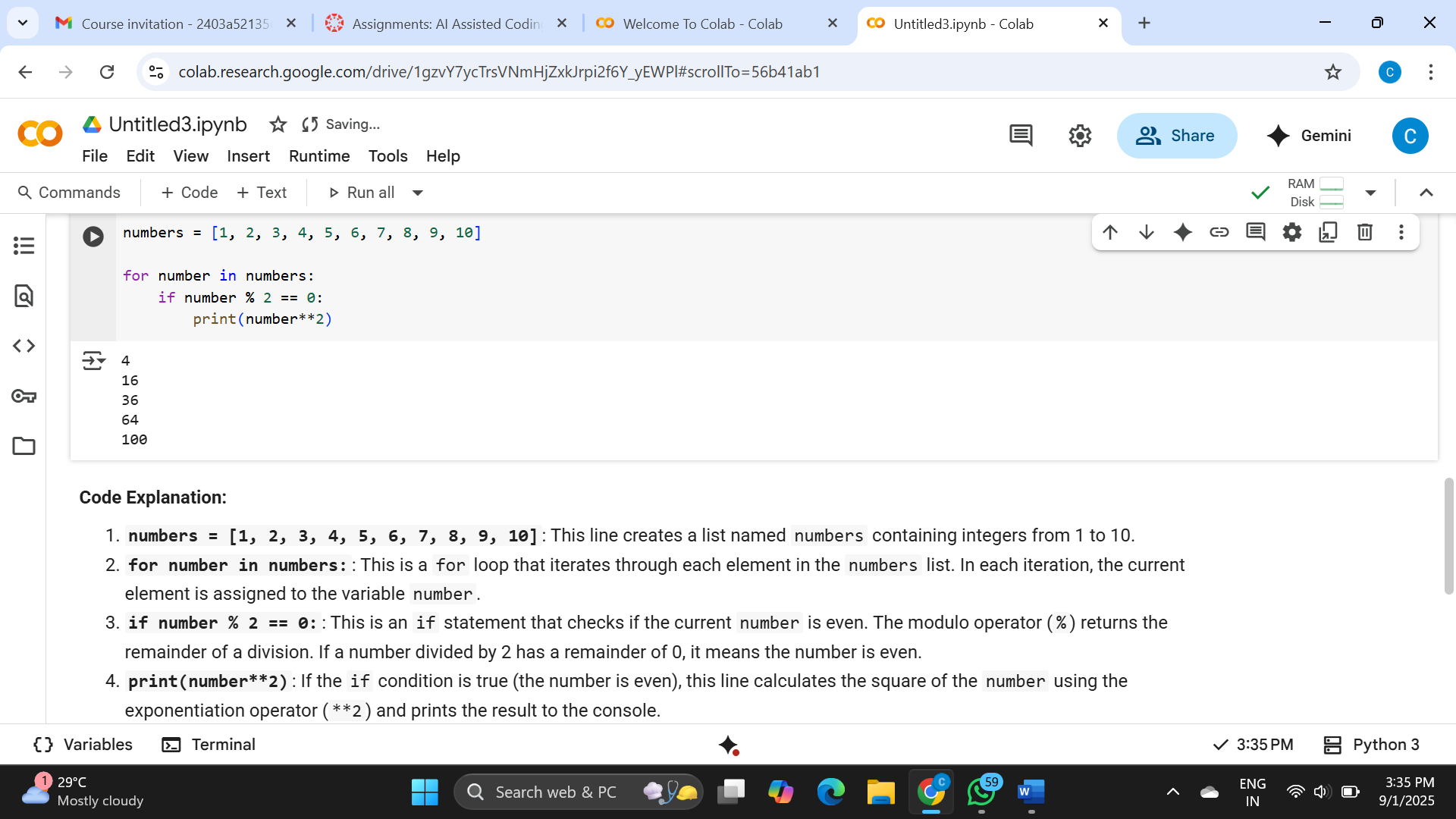
* **class Student:**: This line starts the definition of a new class named Student. Think of a class as a template for creating objects.
* **\_\_init\_\_(self, name, roll\_number, marks):**: This is a special method called the constructor. It's run whenever you create a new Student object. It sets up the object with the name, roll\_number, and marks you provide. self refers to the object being created.
* **self.name = name**, **self.roll\_number = roll\_number**, **self.marks = marks**: These lines store the name, roll\_number, and marks that were passed in as arguments into the object's own attributes (characteristics).
* **display\_details(self):**: This method is defined to print out the details (name, roll number, and marks) of a specific student object.
* **is\_passed(self, average\_marks):**: This method checks if a student's marks are greater than a given average\_marks. It returns True if they are, and False otherwise.
* **student1 = Student("Alice", "101", 85)**: This line creates a new Student object named student1 with the name "Alice", roll number "101", and marks 85.
* **student1.display\_details()**: This calls the display\_details method on the student1 object, printing its information.
* **average\_marks = 70**: This sets a variable average\_marks to 70.
* **if student1.is\_passed(average\_marks):**: This calls the is\_passed method on student1 with the average\_marks (70). The if statement then checks if the result is True.
* **print(...)**: Based on whether is\_passed returned True or False, one of these lines will print a message indicating if Alice passed or not.

Task-2

Write the first two lines of a for loop to iterate through a list of numbers. Use a comment prompt to let Copilot suggest how to calculate and print the square of even numbers only.

**Expected Outcome #2:**

**•** A complete loop generated by Copilot with conditional logic (if number % 2 == 0) and appropriate output.

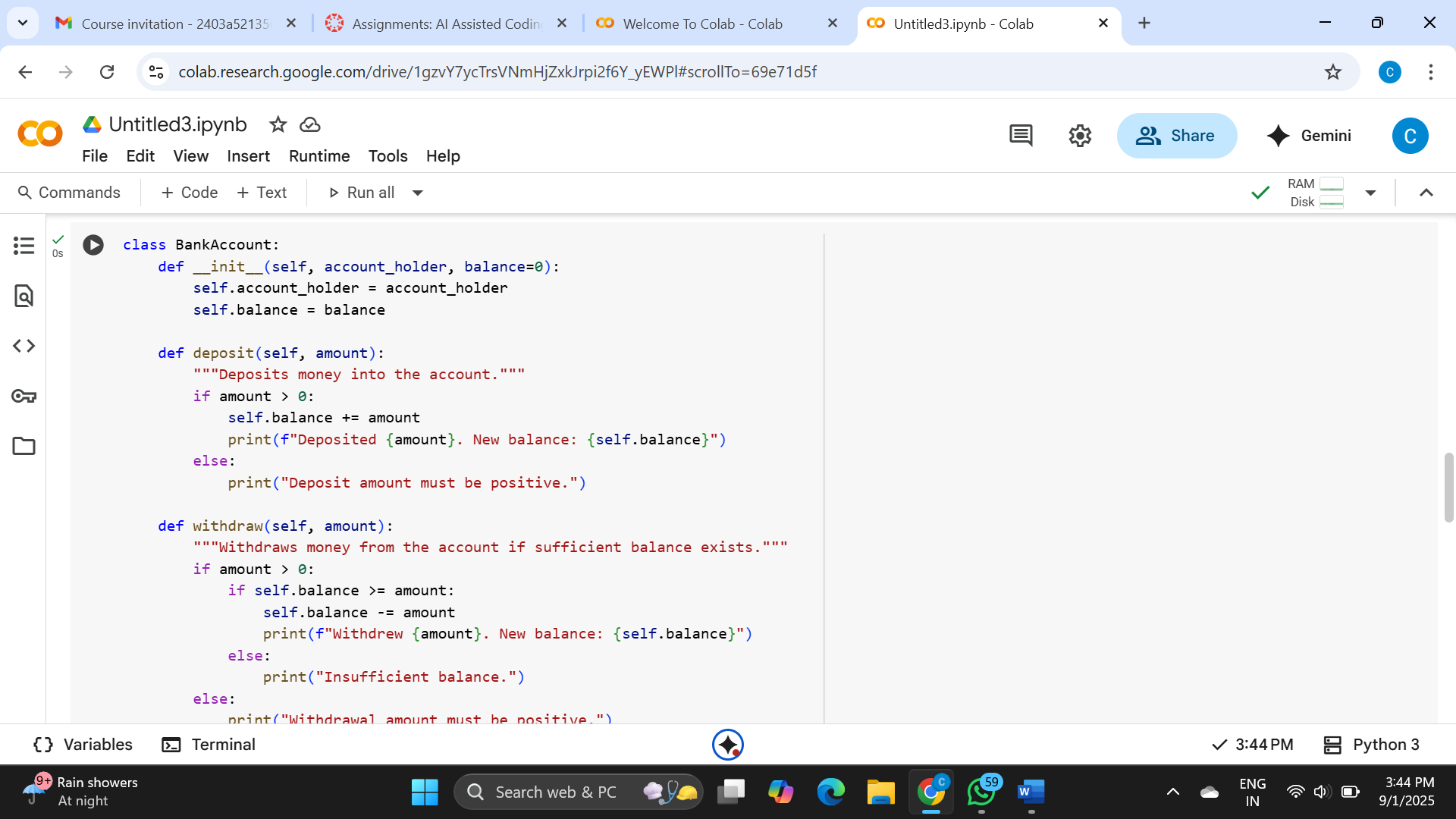


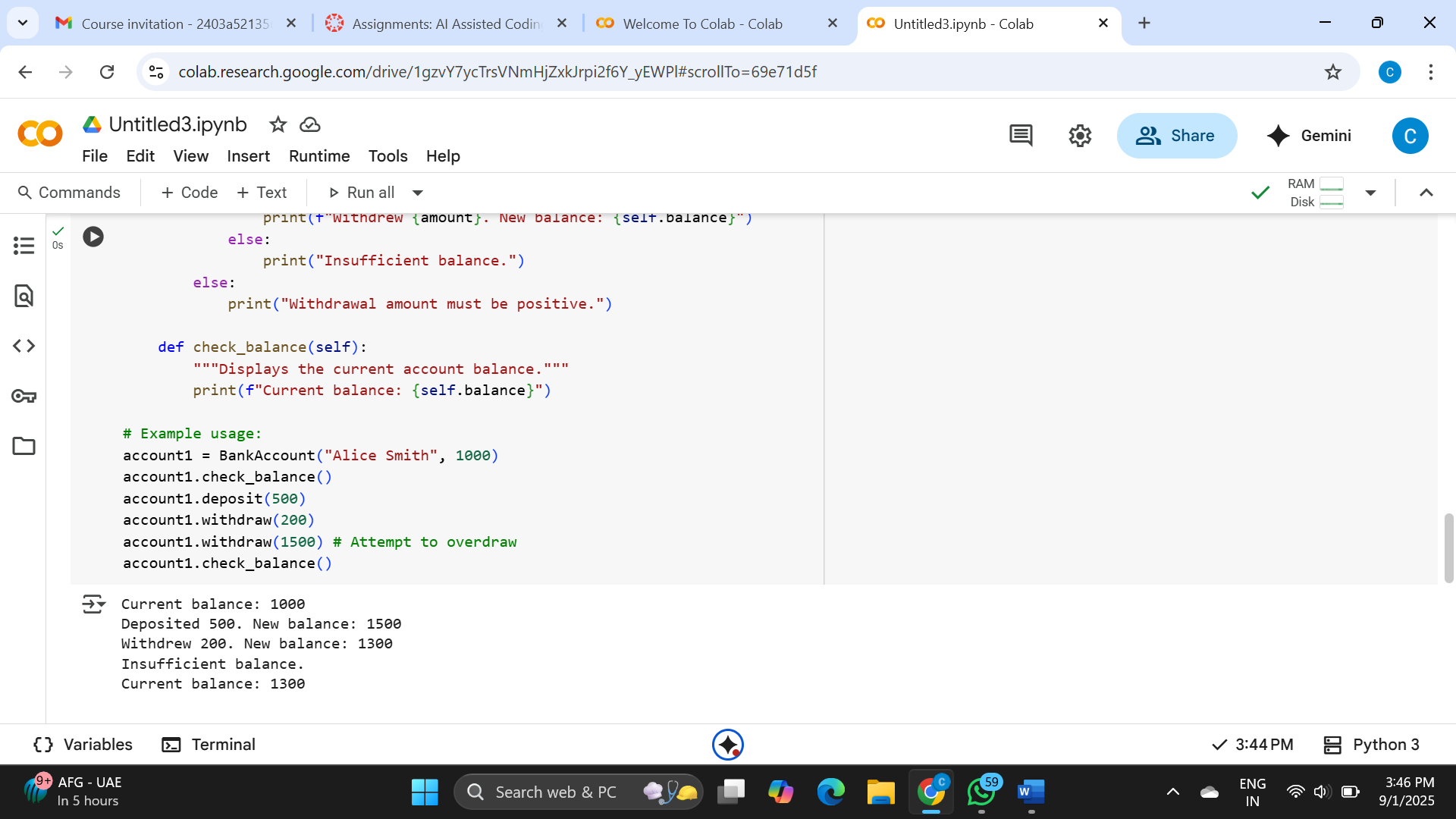
Task-3

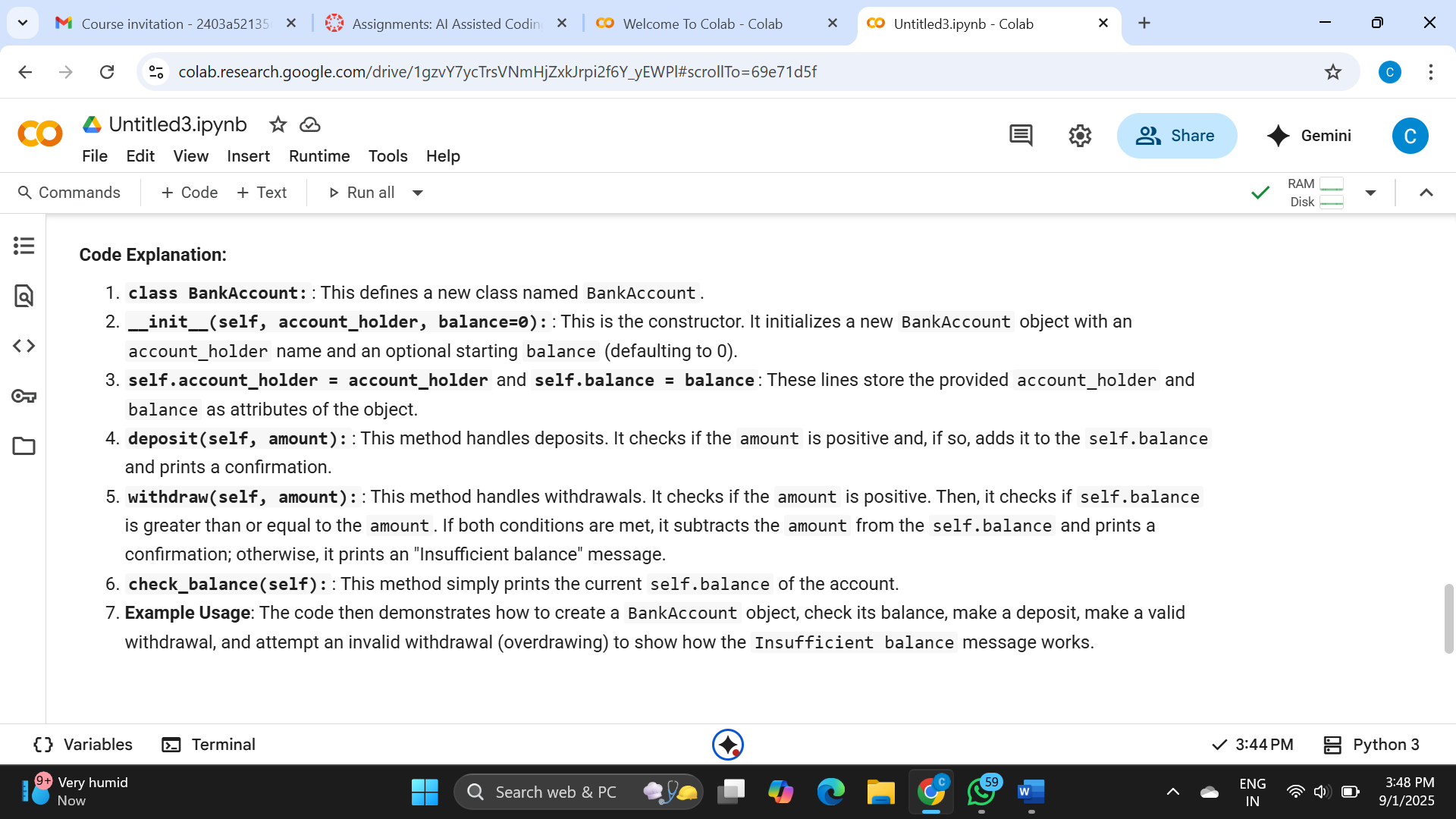
Create a class called BankAccount with attributes account\_holder and balance. Use Copilot to complete methods for deposit(), withdraw(), and check for insufficient balance.

**Expected Outcome #3:**

**•** Functional class with complete method definitions using if conditions and self attributes. Code should prevent overdrawing.





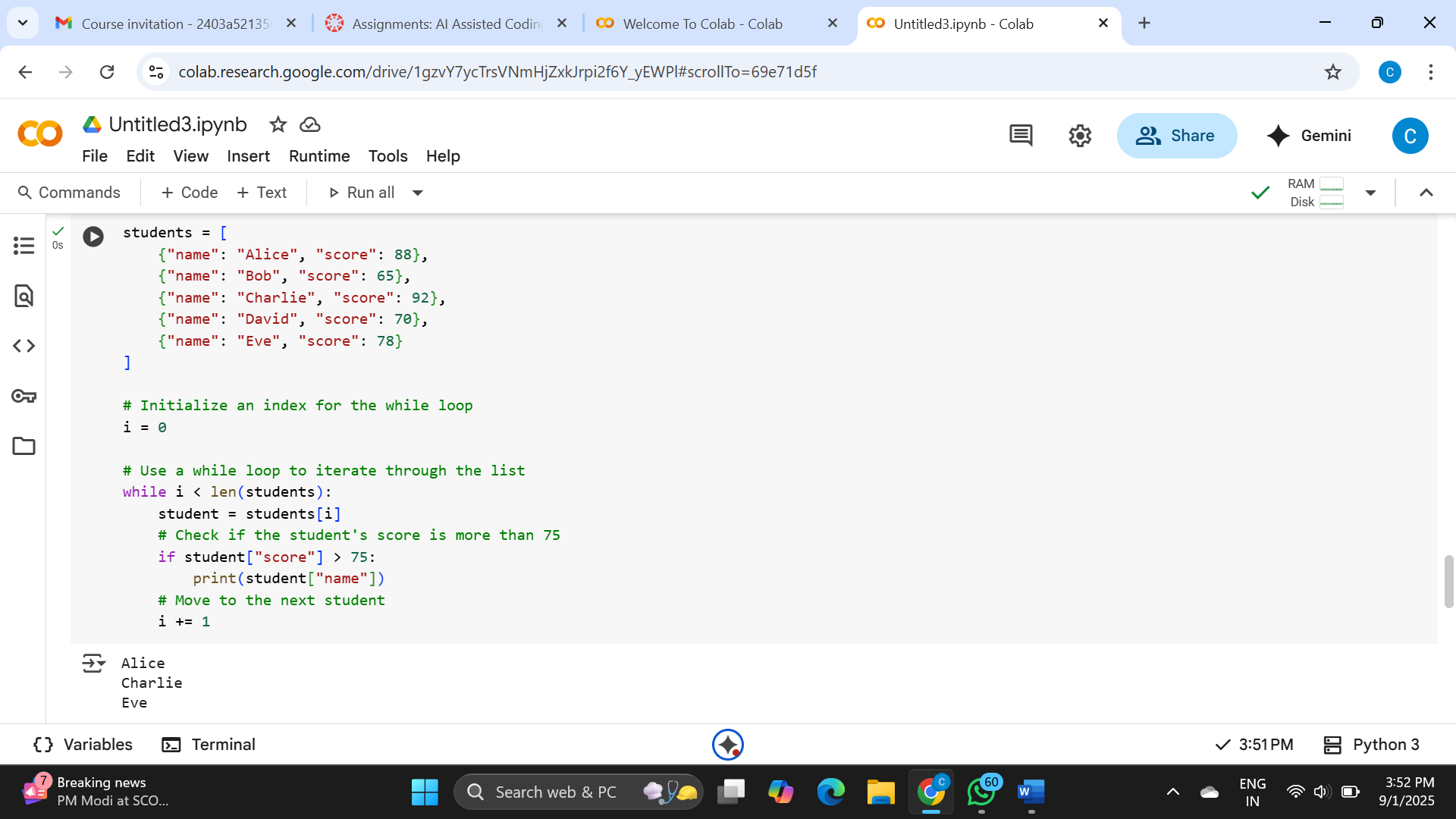


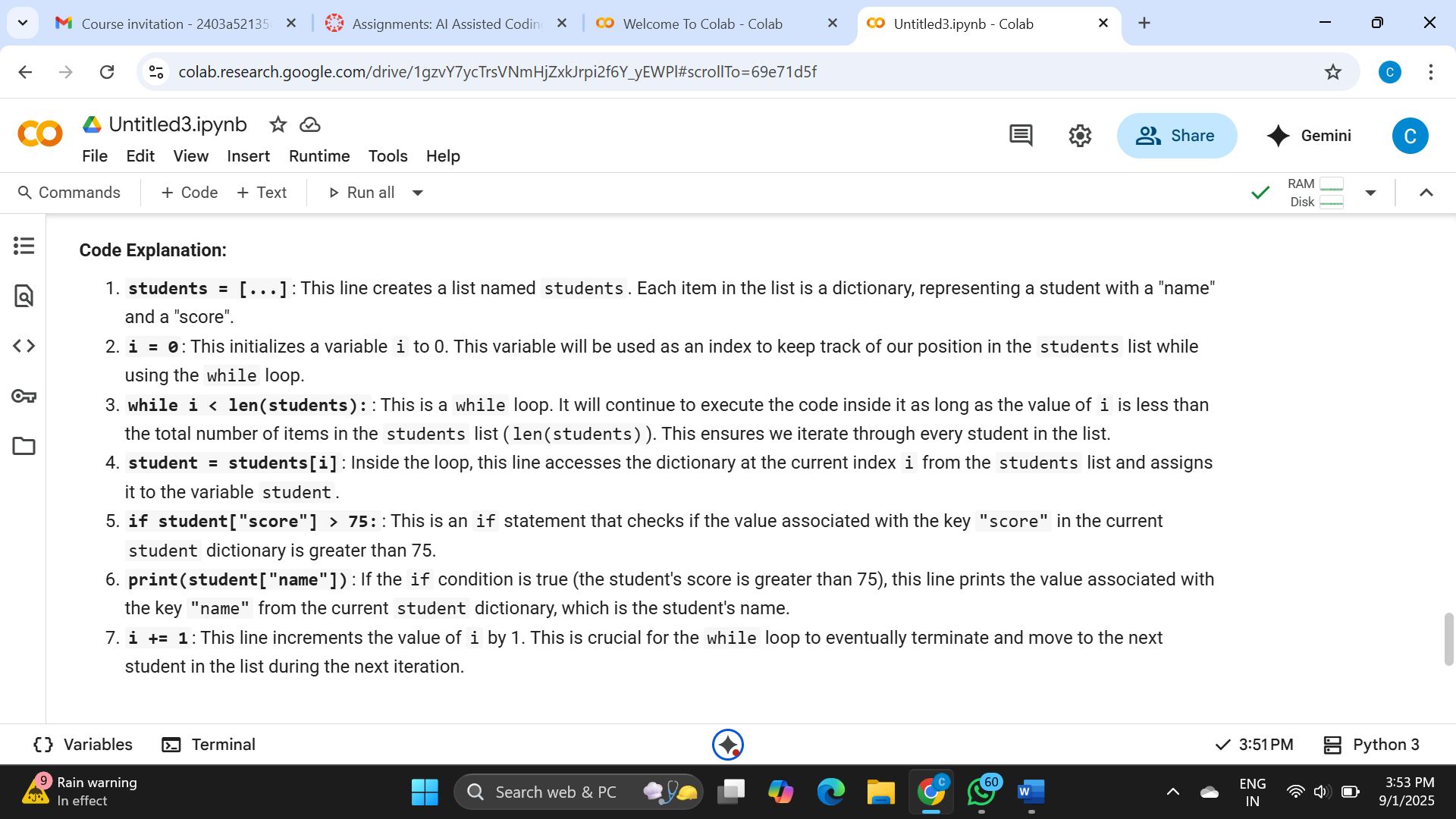
Task-4

Define a list of student dictionaries with keys name and score. Ask Copilot to write a while loop to print the names of students who scored more than 75.

**Expected Outcome #4:**

**•** A complete while loop generated by Copilot with proper condition checks and formatted output.

****

****

Task-5

Begin writing a class ShoppingCart with an empty items list. Prompt Copilot to generate methods to add\_item, remove\_item, and use a loop to calculate the total bill using conditional discounts.

**Expected Outcome #5:**

**•** A fully implemented ShoppingCart class with Copilot-generated loops and if-else statements handling item management and discount logic.

