Lab 9: Documentation Generation: Automatic documentation and code comments

NAME – A.SHIVANI

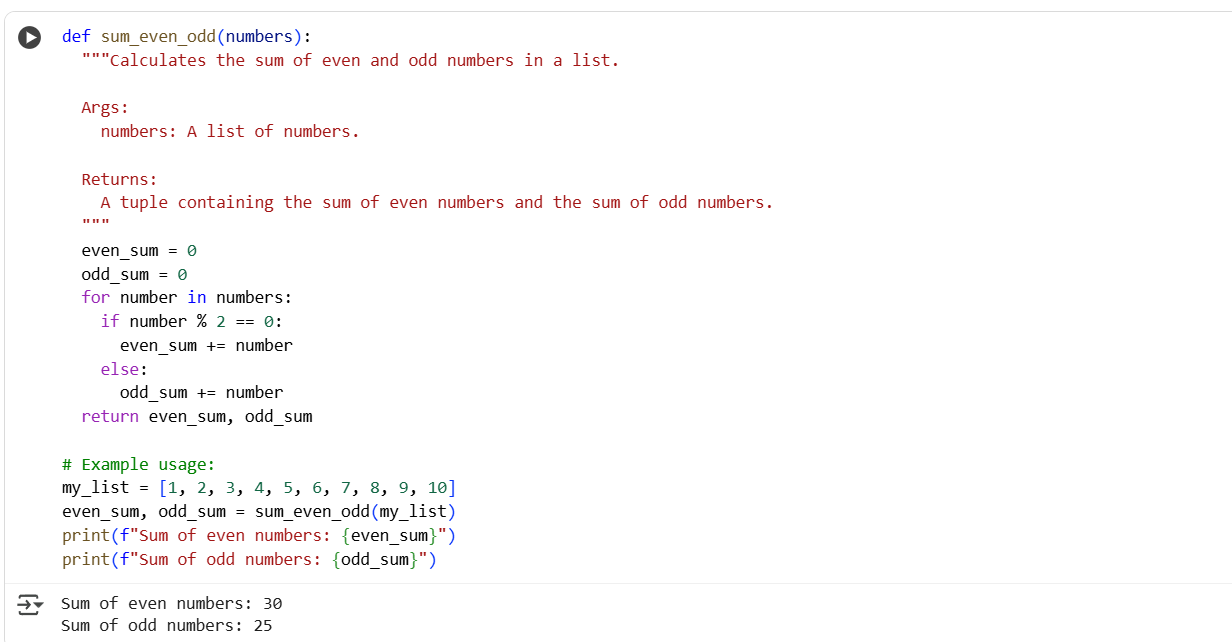
HTNO – 2403A51400

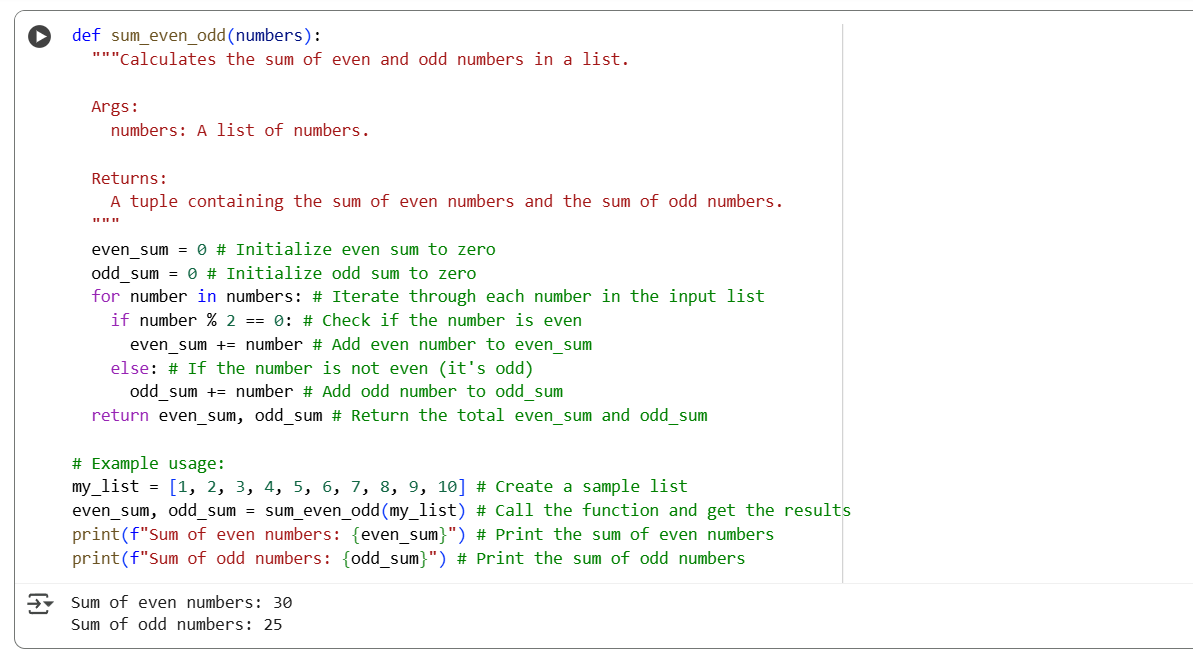
BATCH NO – 16

TASK 1 –

Write python function to return sum of even and odd numbers in the given list. • Incorporate manual docstring in code with Google Style • Use an AI-assisted tool (e.g., Gemini, Copilot, Cursor AI) to generate a docstring describing the function. • Compare the AI-generated docstring with your manually written one.

CODE –





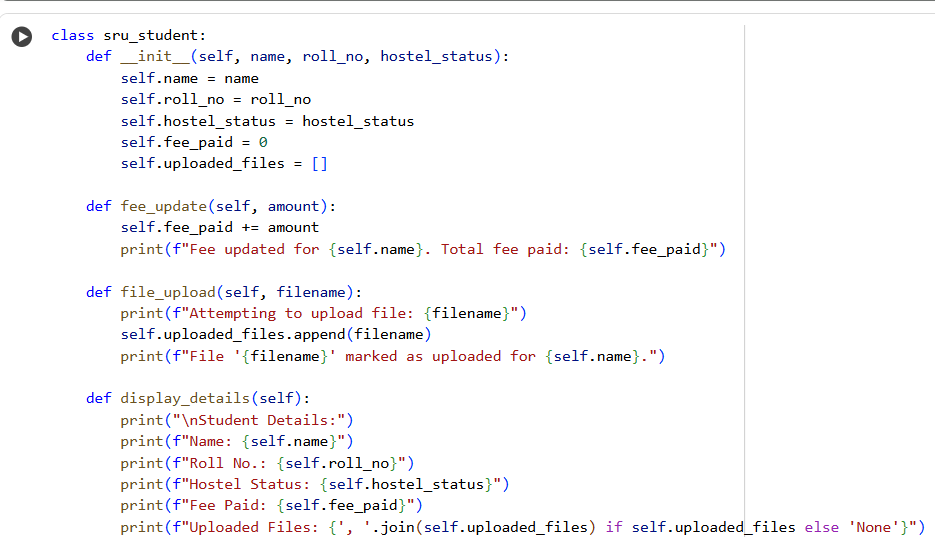
EXPLANATION –

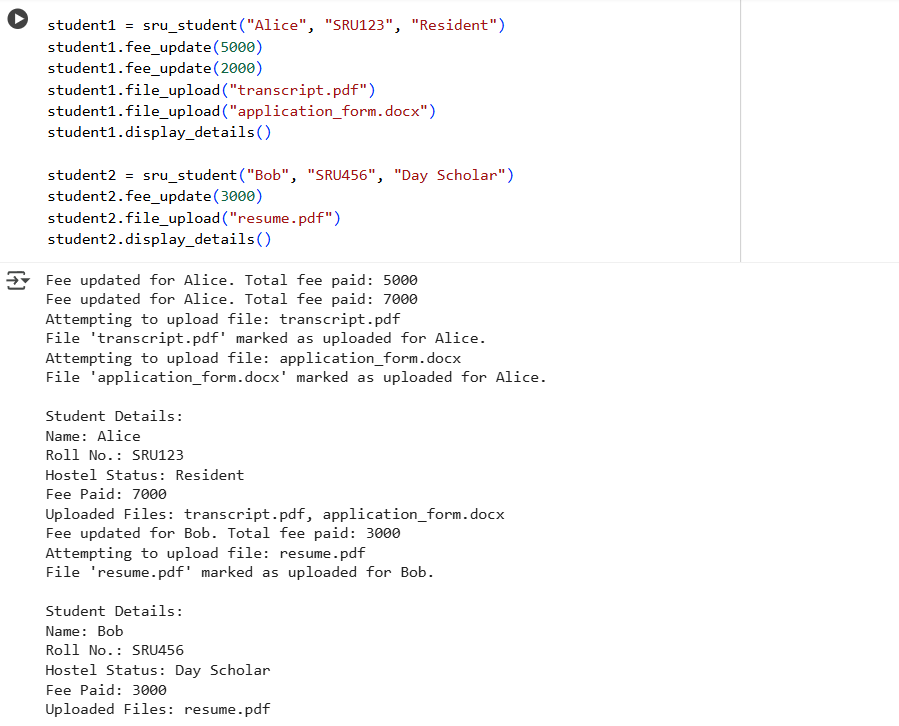
The Python code defines a function sum\_even\_odd that takes a list of numbers as input. It initializes two variables, even\_sum and odd\_sum, to zero. It then iterates through each number in the input list. Inside the loop, it checks if the current number is even using the modulo operator (%). If the number is even, it is added to even\_sum; otherwise (if it's odd), it is added to odd\_sum. After processing all numbers in the list, the function returns a tuple containing the total even\_sum and odd\_sum. The code then demonstrates the use of this function by creating a sample list my\_list, calling the sum\_even\_odd function with this list, and printing the calculated sums of even and odd numbers. This function effectively separates and sums the even and odd numbers within a given list.

TASK 2 –

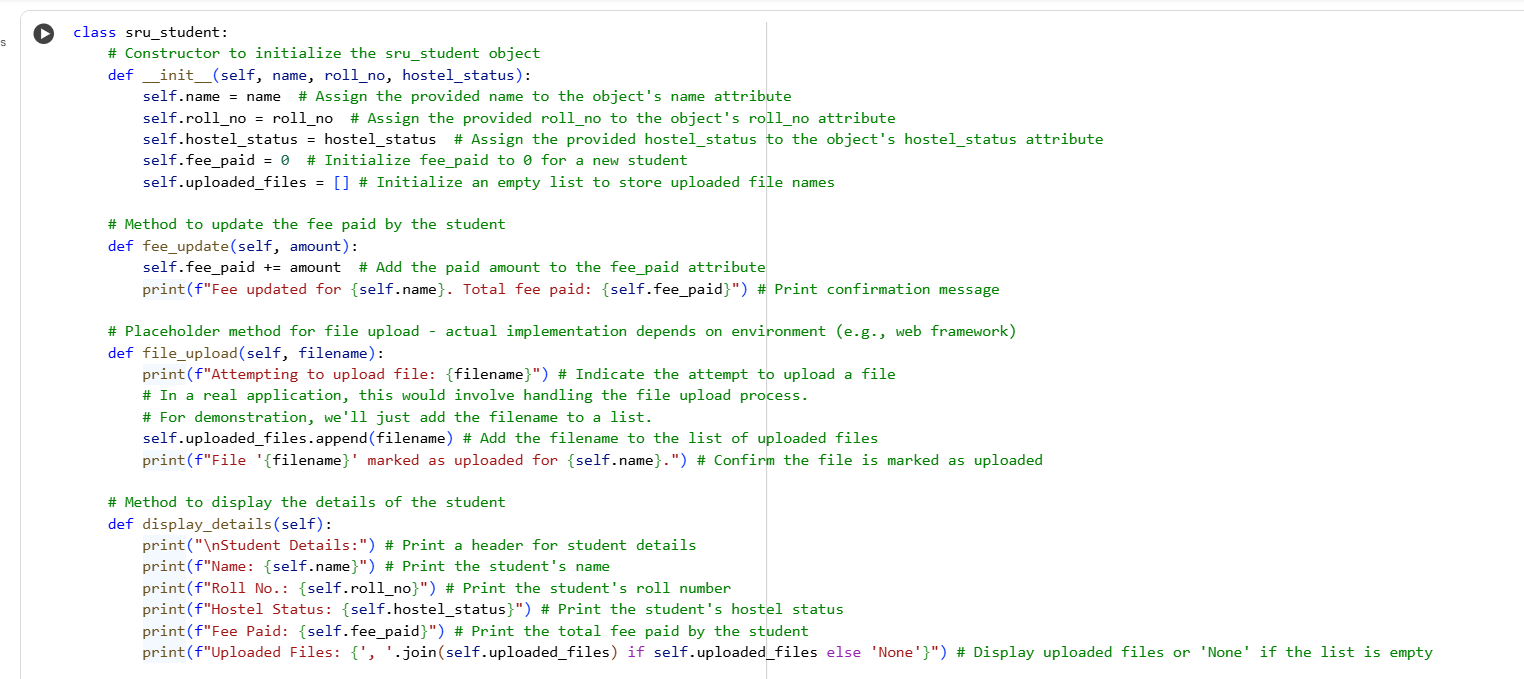
Write python program for sru\_student class with attributes like name, roll no., hostel\_status and fee\_update method and display\_details method. • Write comments manually for each line/code block • Ask an AI tool to add inline comments explaining each line/step. • Compare the AI-generated comments with your manually written one.

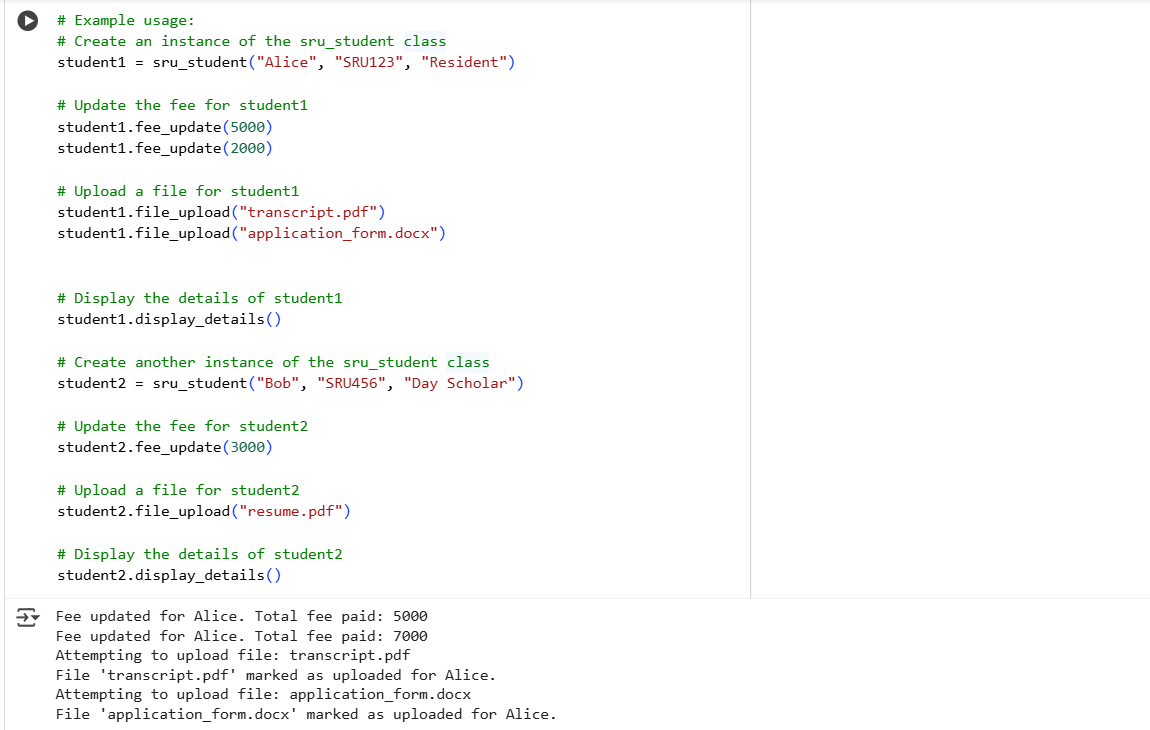
CODE –

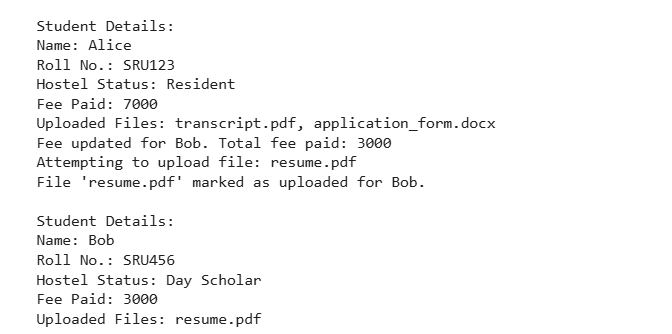




INLINE COMMANDS –







EXPLANATION –

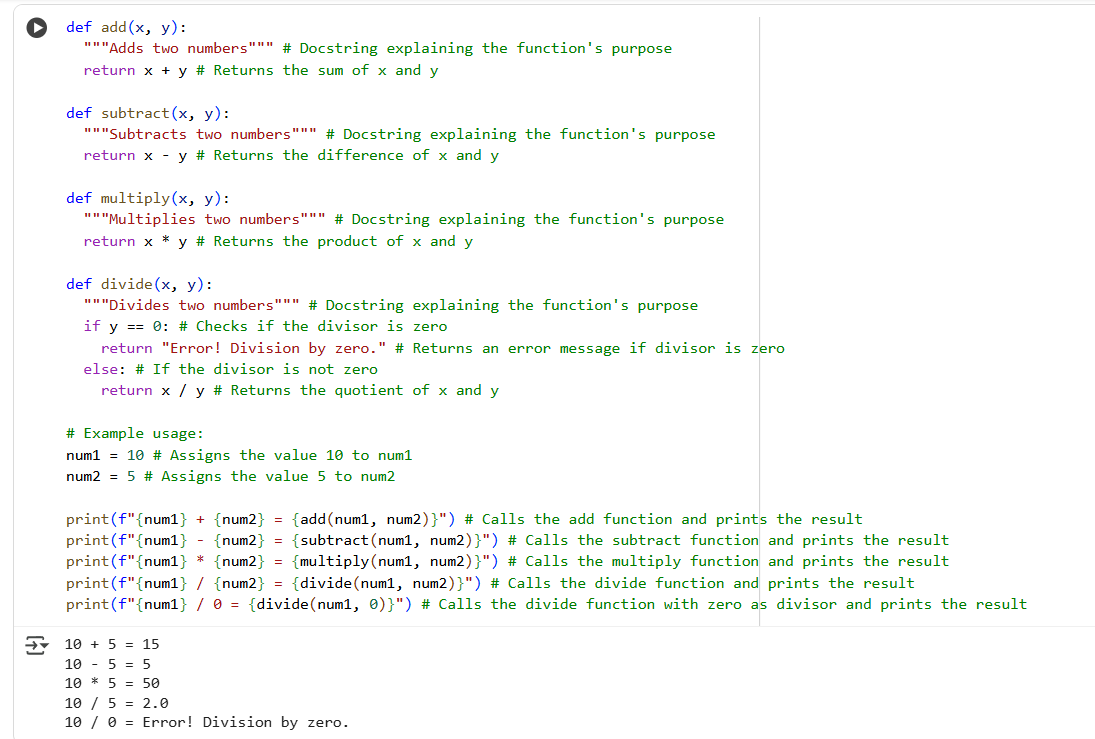
The provided Python code defines a class named sru\_student to represent student information. The \_\_init\_\_ method is the constructor, which initializes a new student object with a name, roll\_no, and hostel\_status, and sets the initial fee\_paid to 0 and uploaded\_files to an empty list. The fee\_update method allows updating the fee paid by adding an amount to the fee\_paid attribute and prints a confirmation. The file\_upload method simulates uploading a file by adding the filename to the uploaded\_files list and printing a message. The display\_details method prints a formatted summary of the student's name, roll number, hostel status, total fee paid, and the list of uploaded files. The example usage demonstrates creating two sru\_student objects, updating their fees, simulating file uploads, and then displaying their details.

TASK 3 –

Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide). Incorporate manual docstring in code with NumPy Style .Use AI assistance to generate a module-level docstring + individual function docstrings. Compare the AI-generated docstring with your manually written one

CODE –





EXPLANATION –

Python code defines four basic arithmetic functions: add, subtract, multiply, and divide. Each function takes two arguments and performs the corresponding operation, returning the result. The divide function includes a check to prevent division by zero, returning an error message in that case. The code then demonstrates how to use these functions by assigning values to two variables, num1 and num2, and calling each function with these variables as arguments. The results of each operation, including an example of division by zero, are printed to the console using f-strings for clear output. This script provides a simple command-line calculator functionality.