**7.Write a Python decorator that measures the execution time of a function and logs it. Apply this decorator to a function that performs a computationally expensive task.**

import time

# Create the Decorator

def execution\_time\_logger(func):

def wrapper(\*args, \*\*kwargs):

start\_time = time.time() # Record the start time

result = func(\*args, \*\*kwargs) # Execute the function

end\_time = time.time() # Record the end time

execution\_time = end\_time - start\_time # Calculate the execution time

print(f"Execution time of {func.\_\_name\_\_}: {execution\_time:.4f} seconds")

return result

return wrapper

# Apply the Decorator

@execution\_time\_logger

def sum\_of\_squares(n):

return sum(i \* i for i in range(n))

# Test the decorated function

result = sum\_of\_squares(10\*\*6)

print(f"Result: {result}")

**OUTPUT**:

