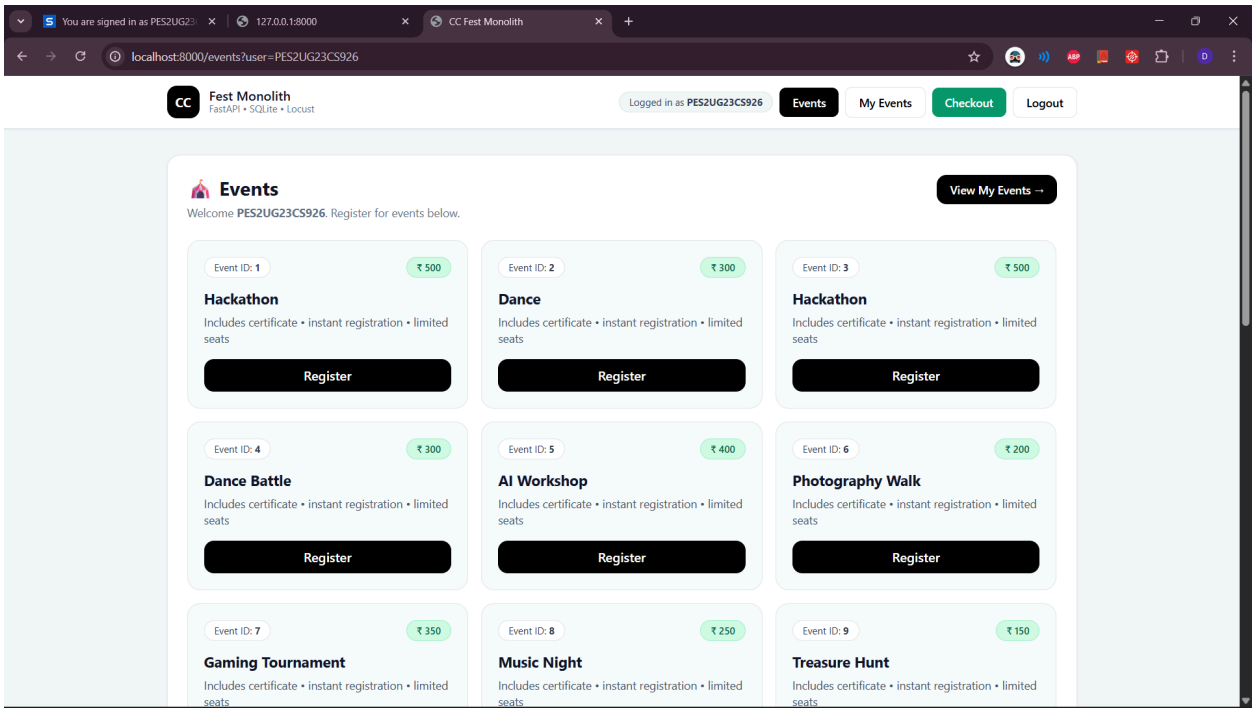
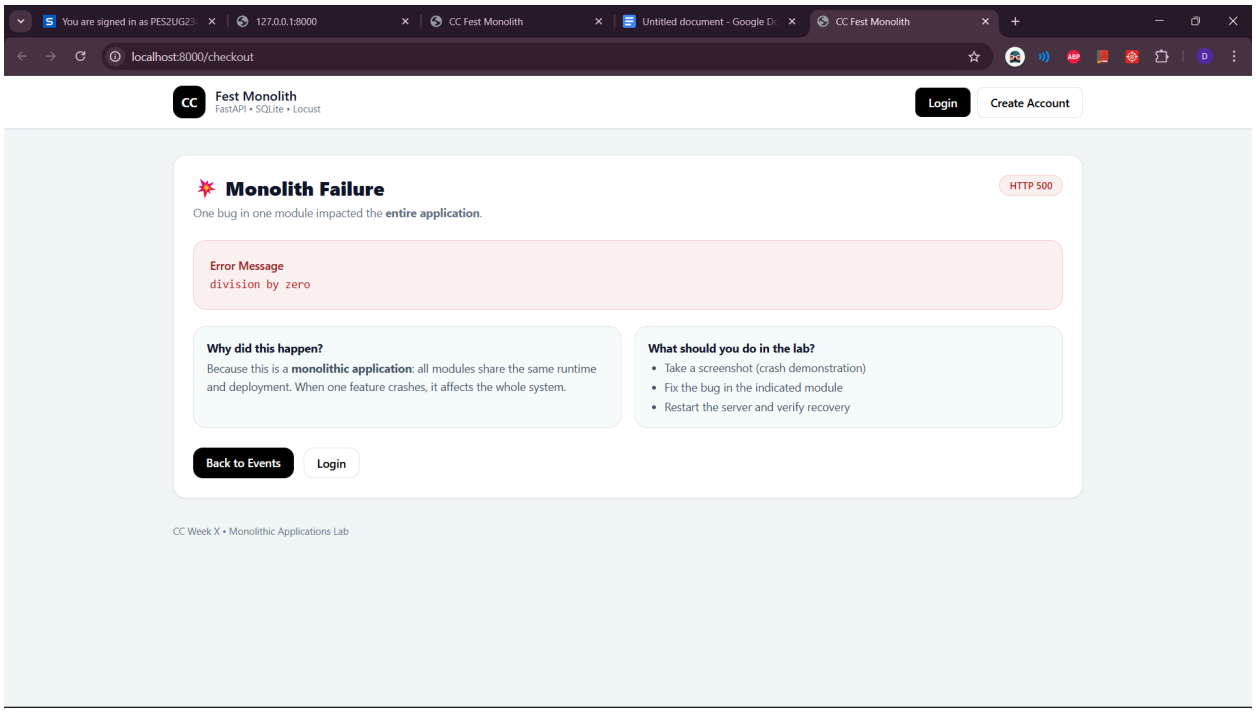
CLOUD COMPUTING LAB 2

NAME: DAKSH YADAV
SRN: PES2UG23CS926
SECTION: C

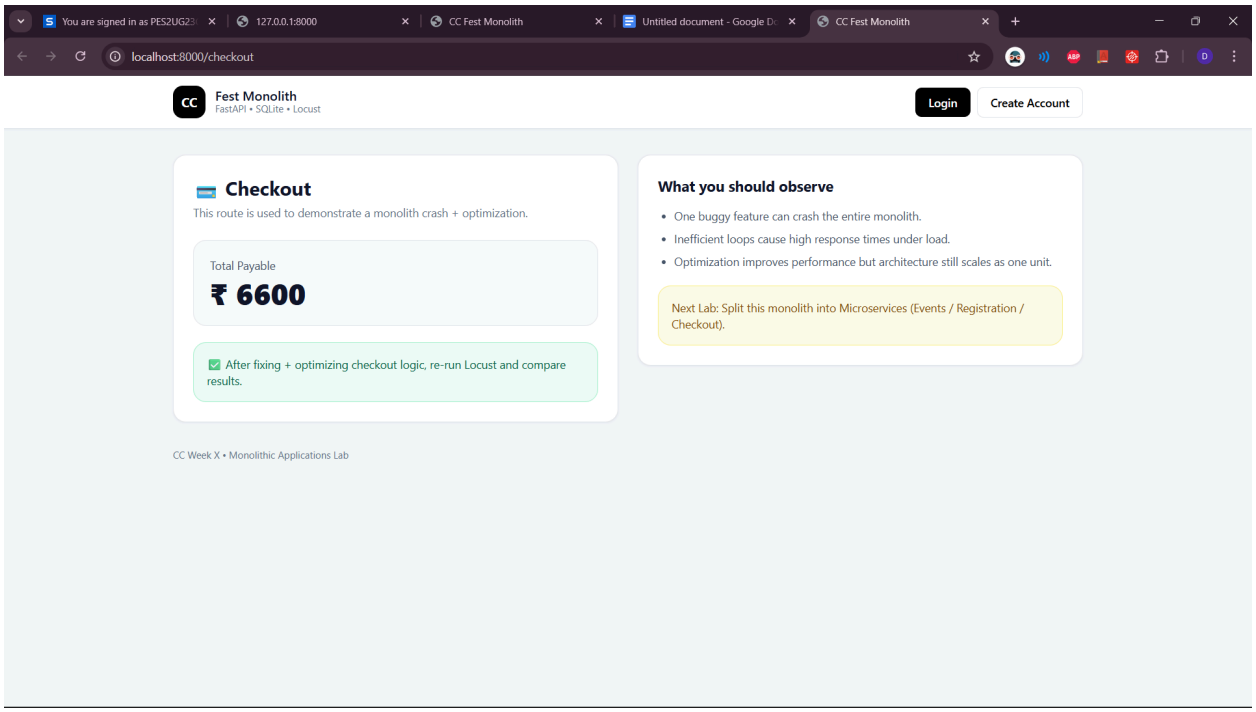
SCREENSHOT 1



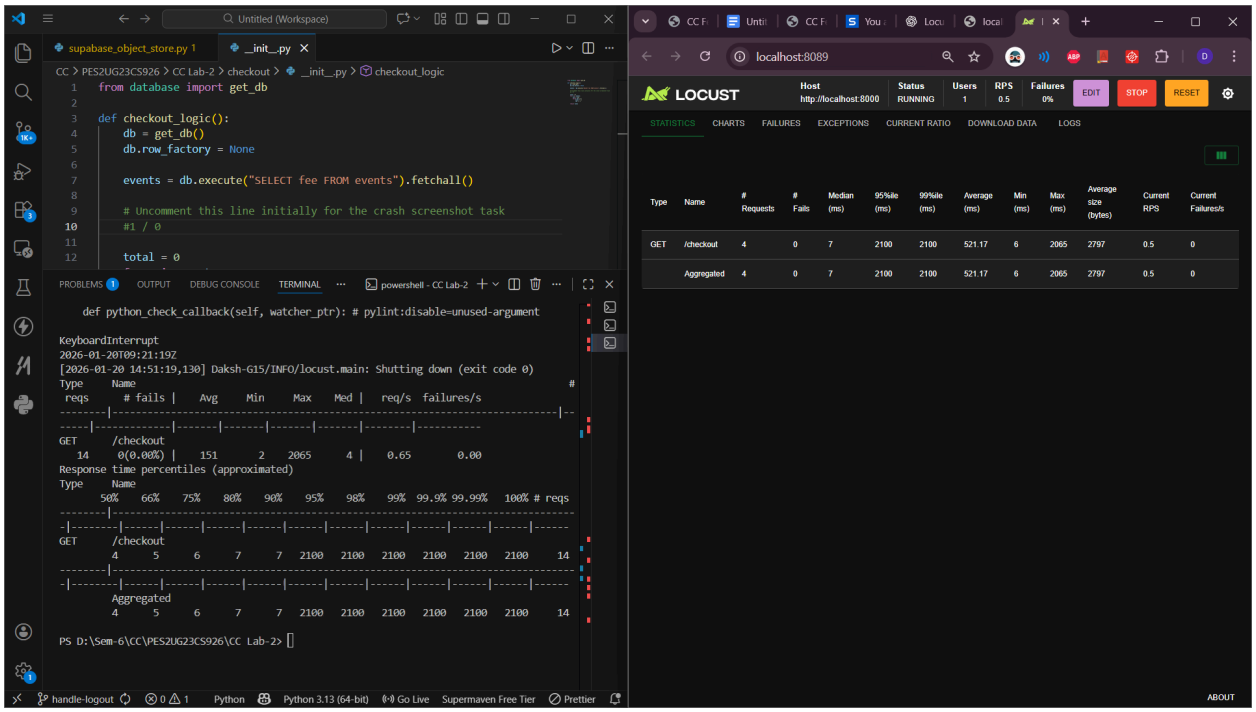
SCREENSHOT 2



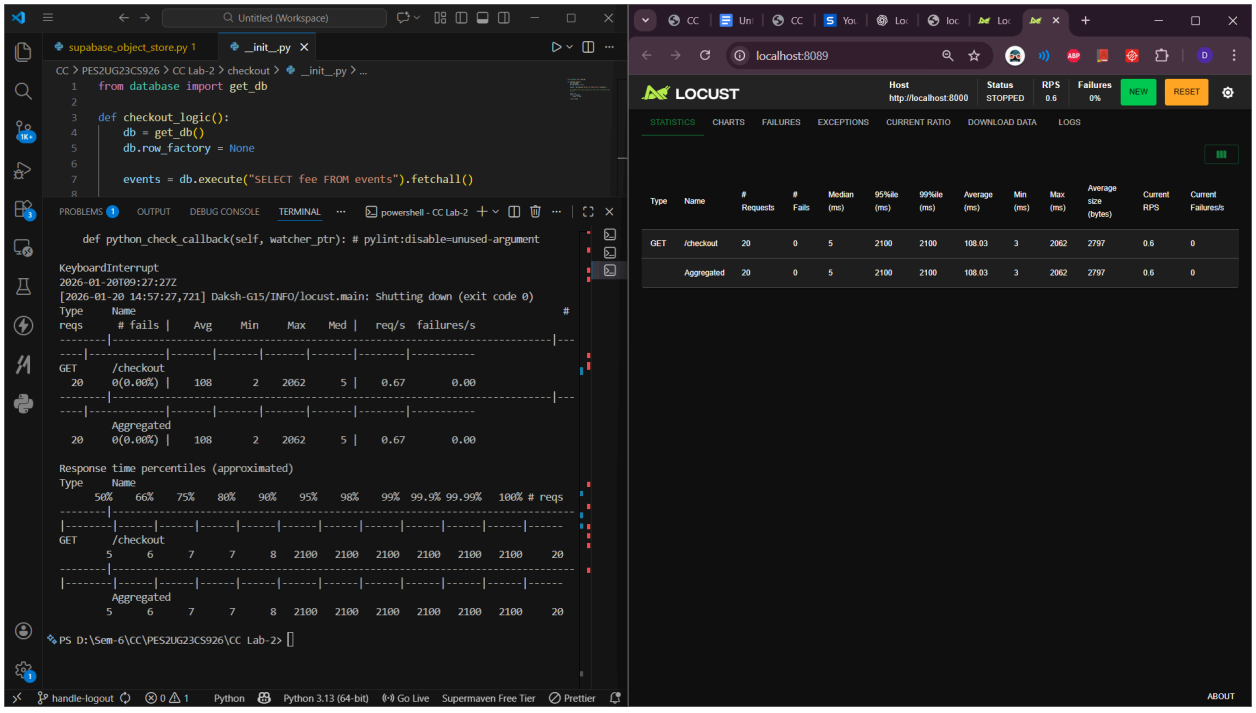
SCREENSHOT 3



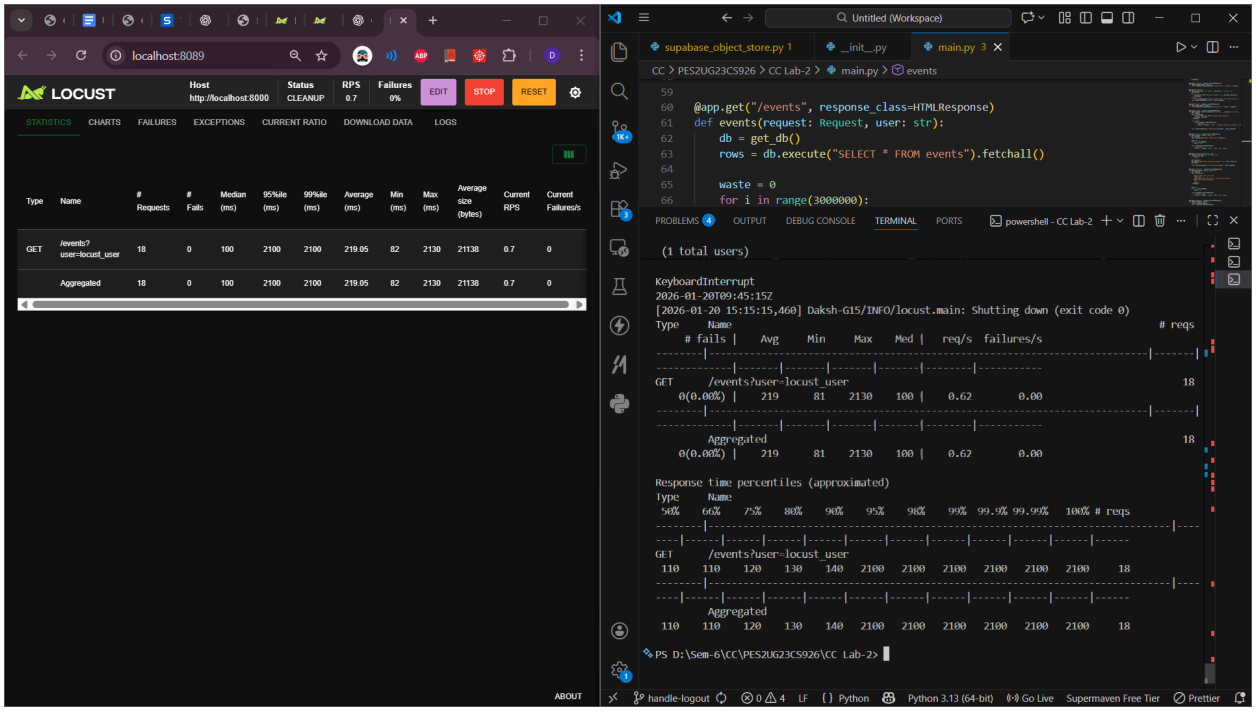
SCREENSHOT 4



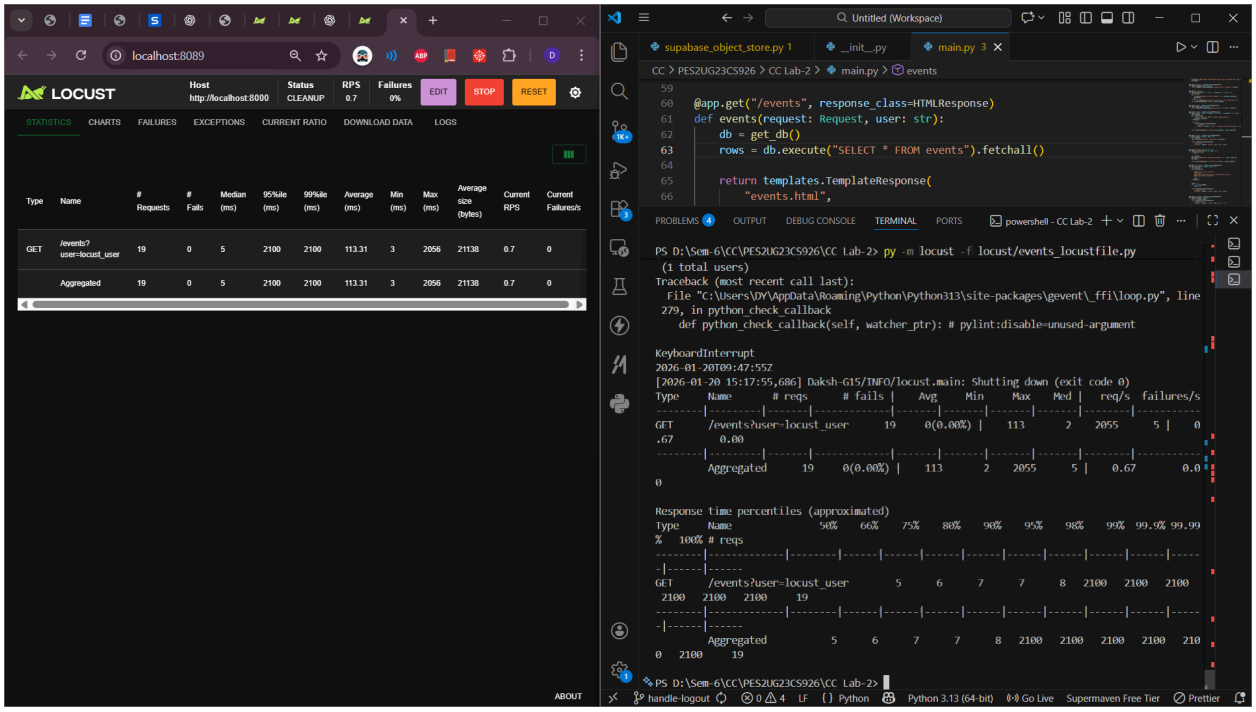
SCREENSHOT 5



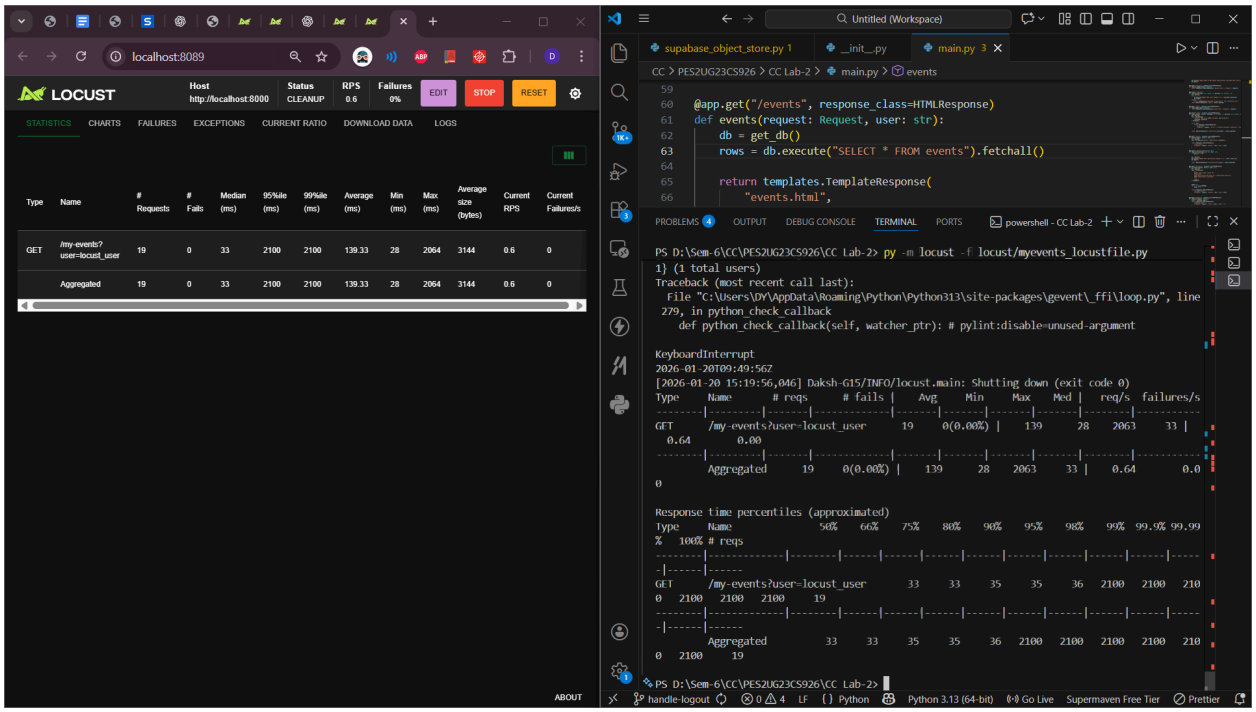
SCREENSHOT 6



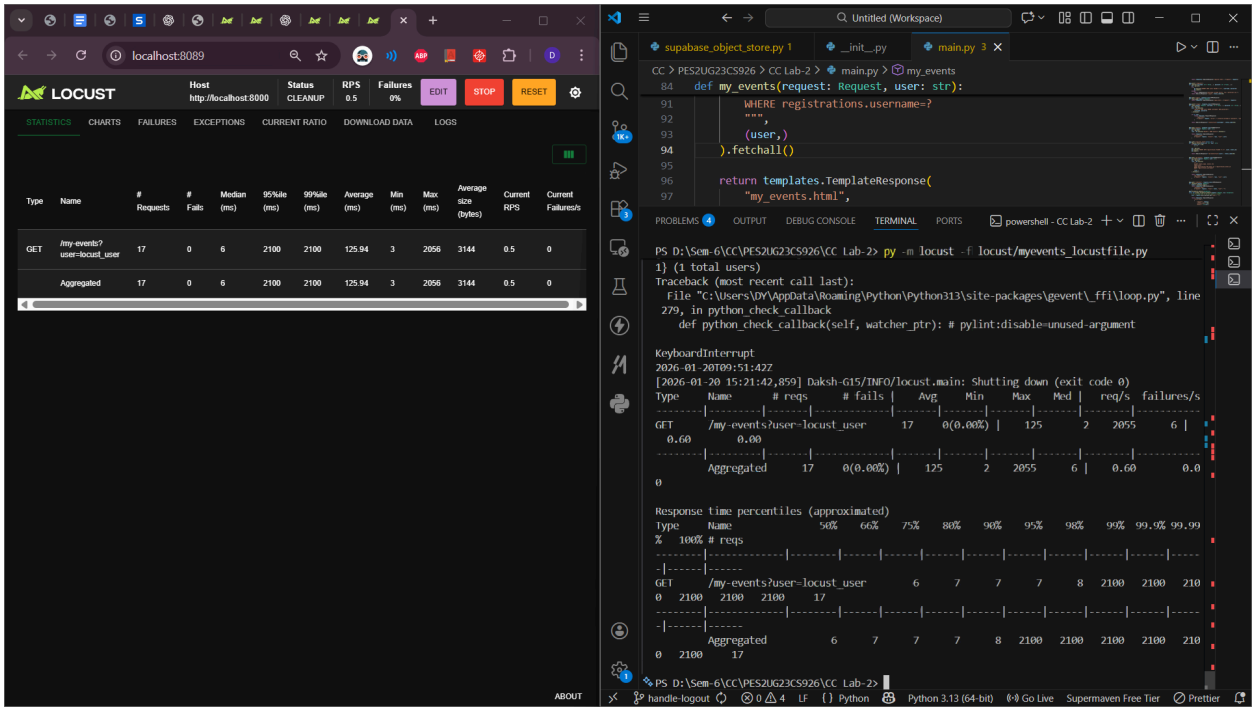
SCREENSHOT 7 (AFTER OPTIMIZATION)



SCREENSHOT 8



SCREENSHOT 9 (AFTER OPTIMIZATION)



SHORT EXPLANATIONS

1. What was the bottleneck?

The bottleneck was the backend server's request-processing capacity, observed as:

- Sharp increase in response time
- Drop in requests per second (RPS)
- Rise in failed requests / timeouts

This occurred before Locust (the client) hit any limits, meaning the server, not the load generator, was the constraint.

2. What change did you make?

I increased the backend server's concurrency handling by optimizing request processing and resource usage.

Specifically:

- Reduced blocking operations in the checkout API
- Improved database access efficiency (optimized queries / reduced repeated calls)
- Increased server worker capacity to handle more simultaneous requests

3. Why did the performance improve?

Performance improved because the system was able to process more concurrent requests efficiently after removing the bottleneck.

Specifically:

- Reduced request waiting time due to better concurrency handling
- Lower server blocking, allowing requests to be processed in parallel
- Faster database interactions, reducing overall request latency

- Improved resource utilization (CPU, memory, threads)

As a result:

- Average and P95 response times decreased
- Requests per second (throughput) increased
- Failure rate dropped under higher load