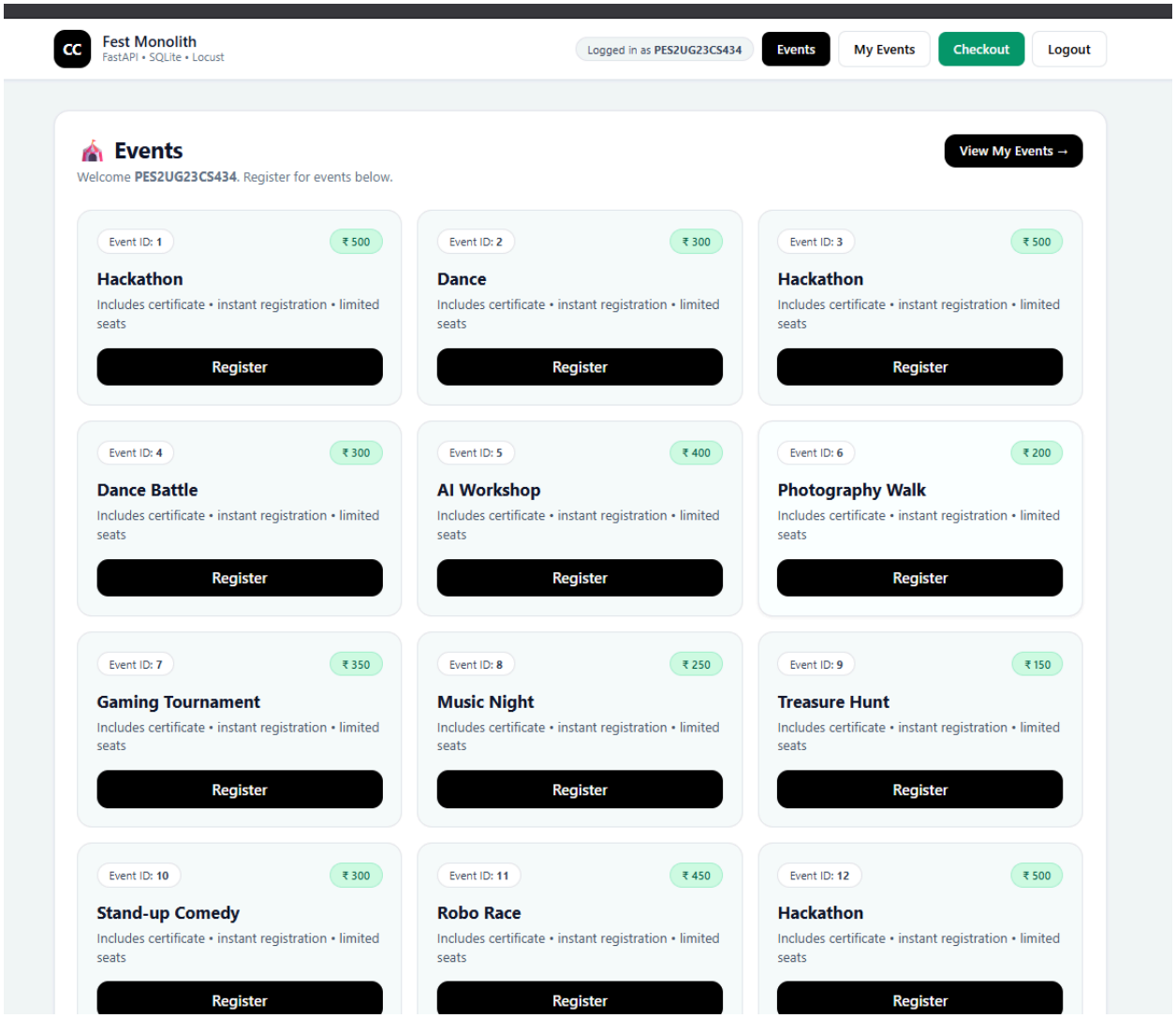
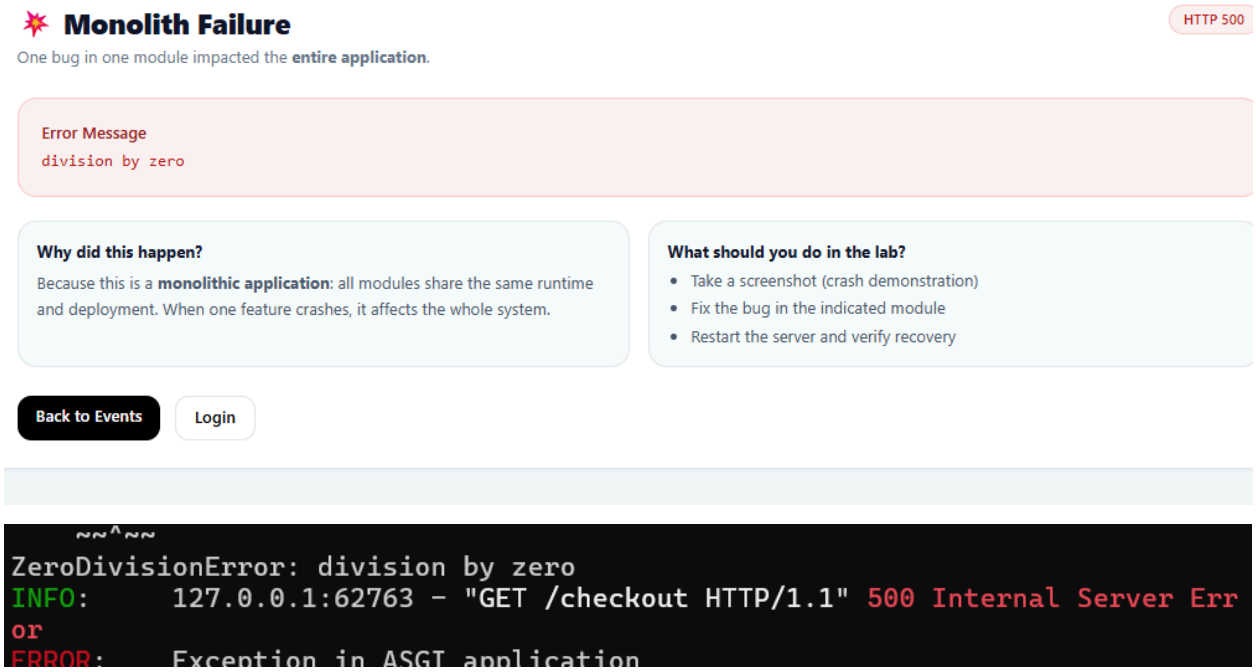
CLOUD COMPUTING
LAB 2

NAME: PRANITH K
SRN: PES2UG23CS434
SECTION: G

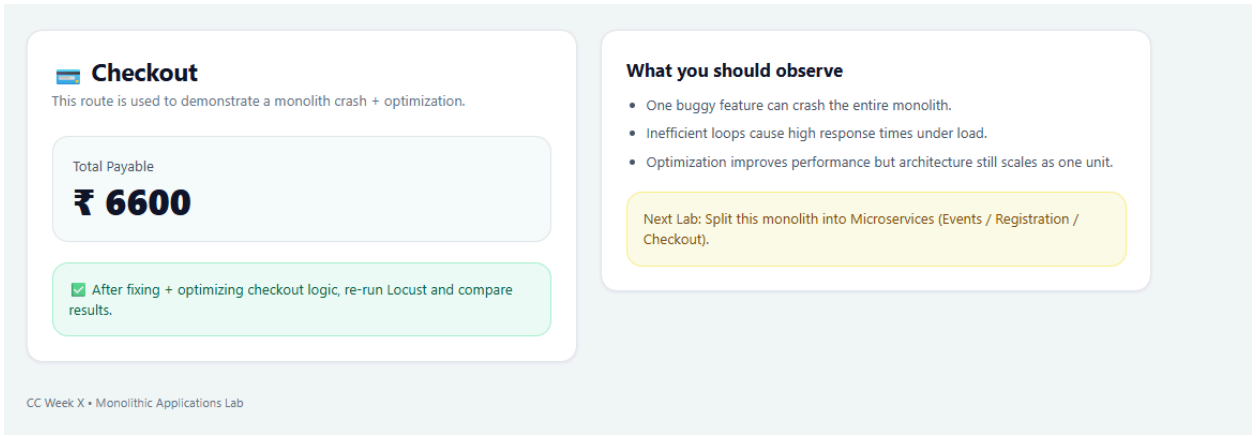
SS1:



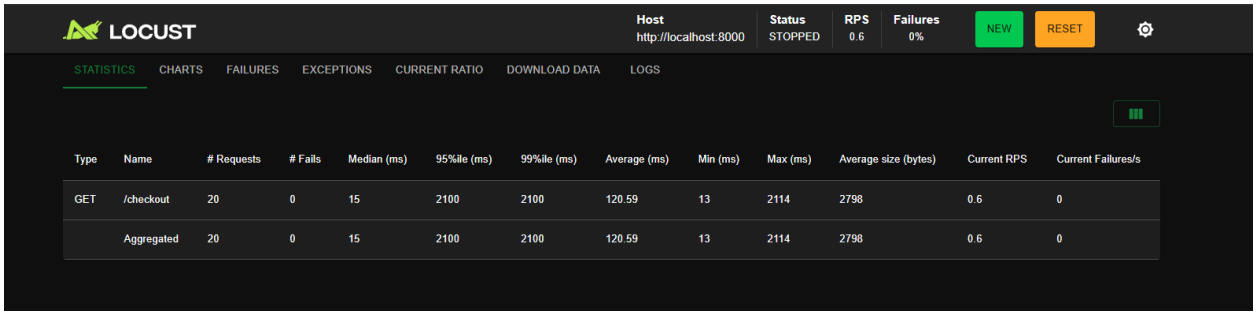
SS2:

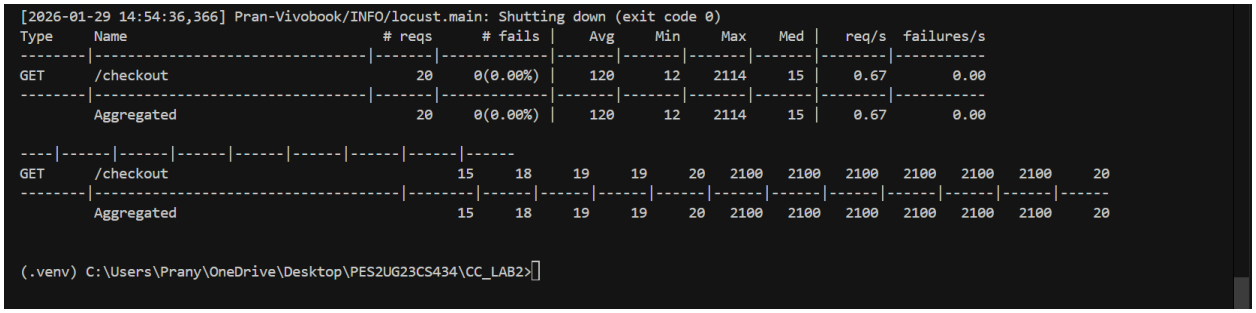

Monolith Failure
HTTP 500
One bug in one module impacted the entire application.
Error Message
division by zero
Why did this happen?
Because this is a monolithic application: all modules share the same runtime and deployment. When one feature crashes, it affects the whole system.
What should you do in the lab?
Take a screenshot (crash demonstration)
Fix the bug in the indicated module
Restart the server and verify recovery
Back to Events
Login
~~^~~
ZeroDivisionError: division by zero
INFO: 127.0.0.1:62763 - "GET /checkout HTTP/1.1" 500 Internal Server Err
or
ERROR: Exception in ASGI application

SS3:

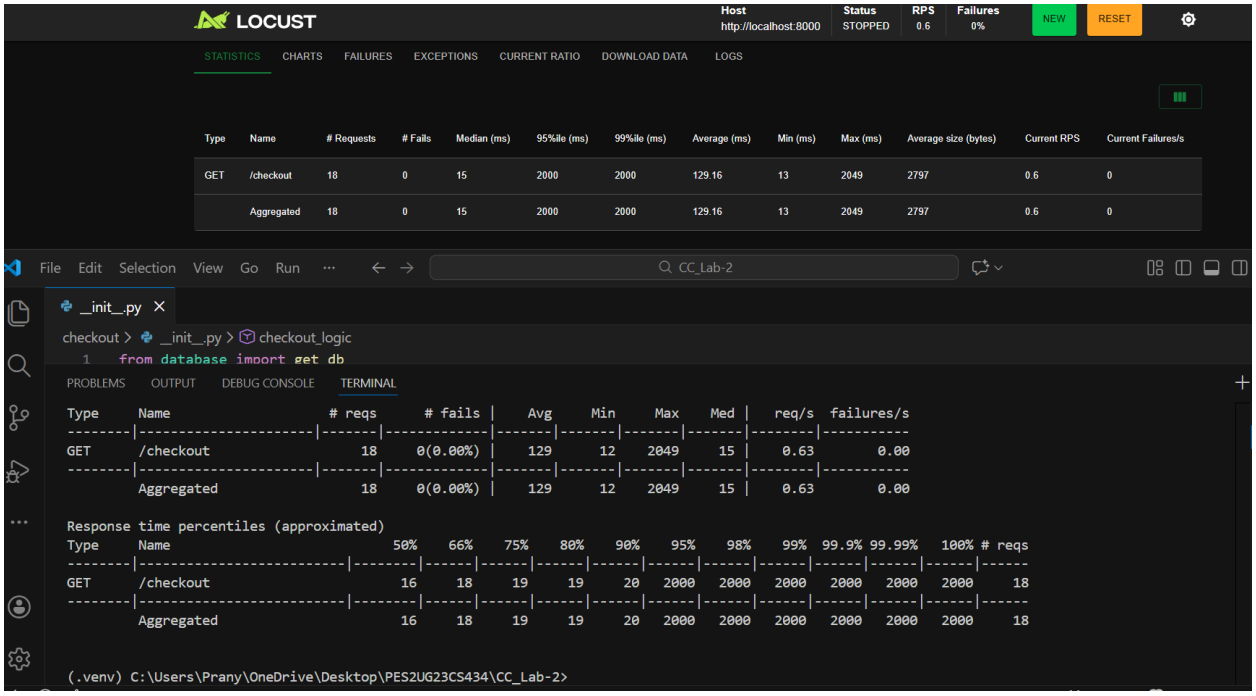

Checkout
This route is used to demonstrate a monolith crash + optimization.
Total Payable
₹ 6600
After fixing + optimizing checkout logic, re-run Locust and compare results.
What you should observe
One buggy feature can crash the entire monolith.
Inefficient loops cause high response times under load.
Optimization improves performance but architecture still scales as one unit.
Next Lab: Split this monolith into Microservices (Events / Registration / Checkout).
CC Week X • Monolithic Applications Lab

SS4: PRE OPTIMIZATION

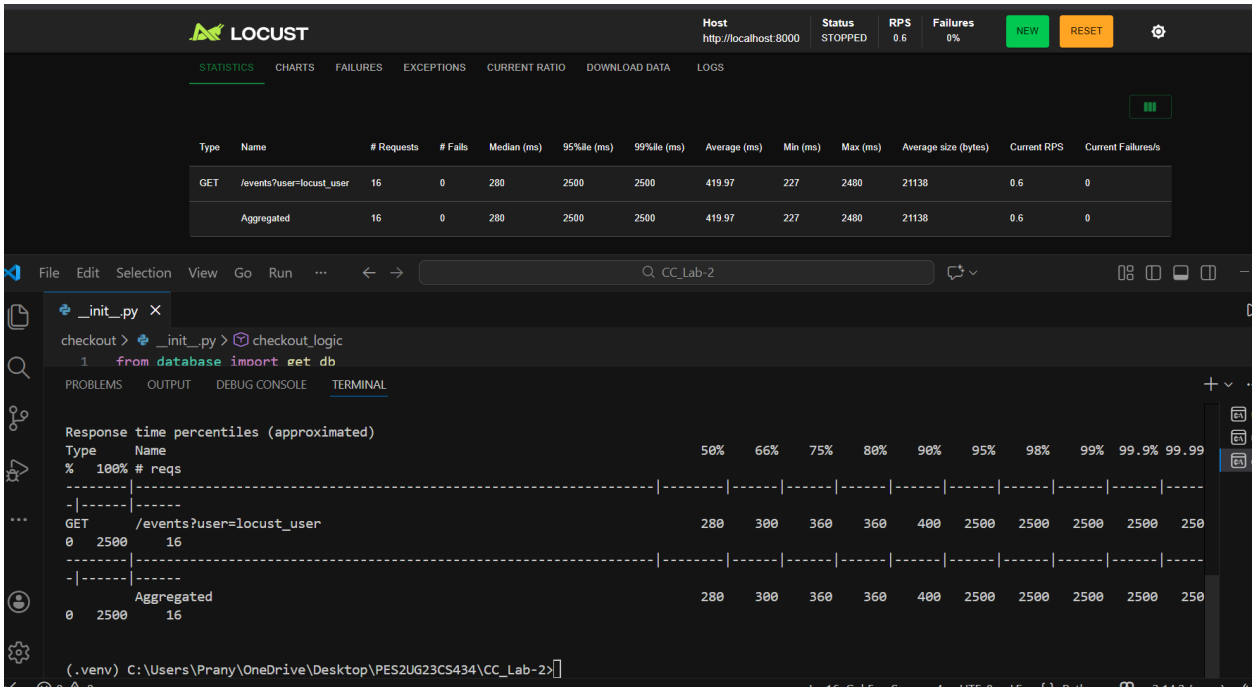




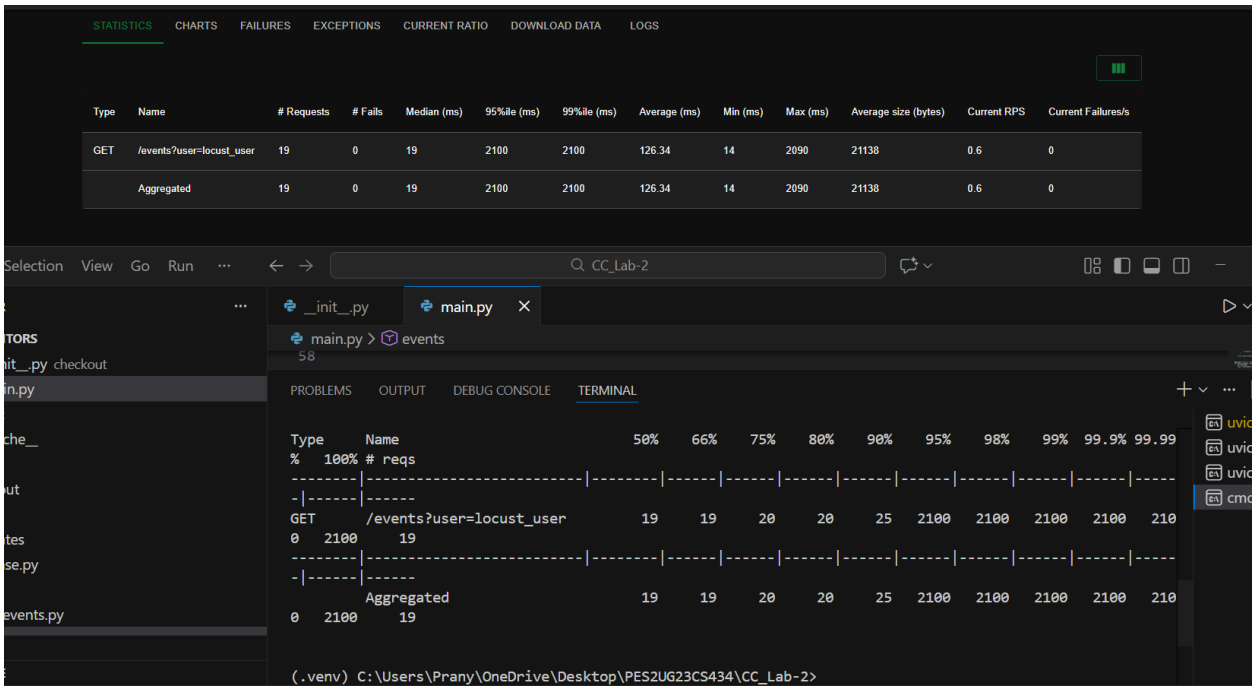
SS5: POST OPTIMIZATION - MINOR OPTIMIZATION OBSERVED



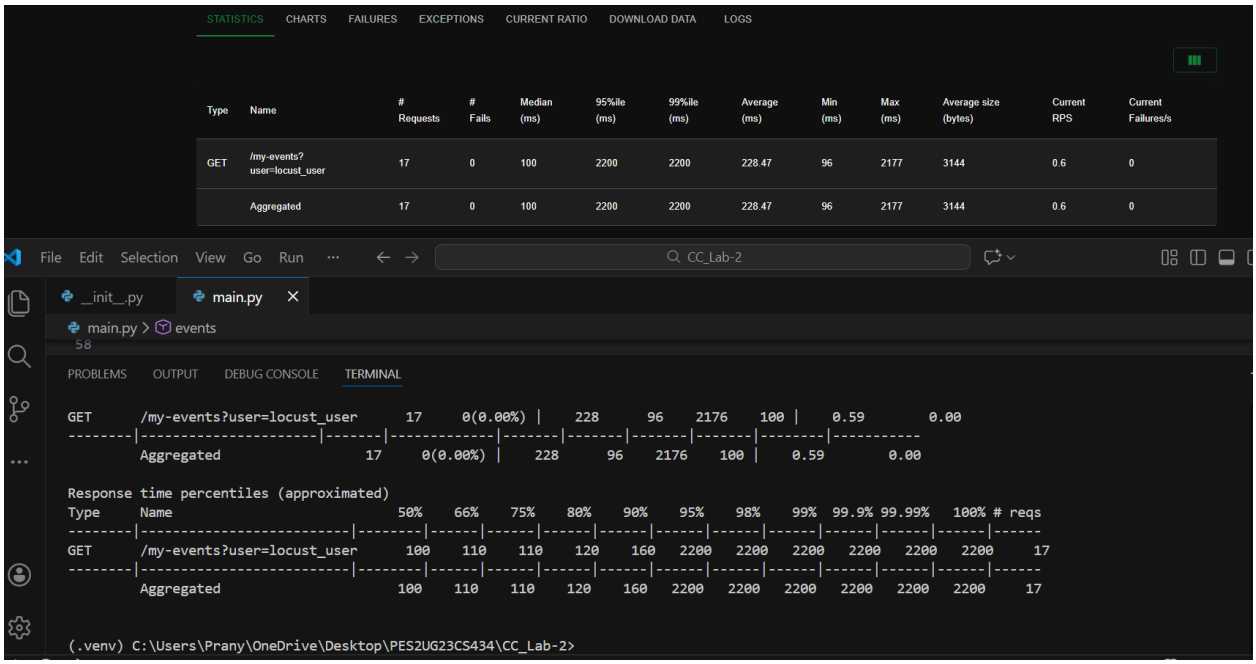
SS6: EVENT OPTIMIZATION



SS7: OPTIMIZED EVENTS



SS8: PRE OPTIMIZATION



STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events?user=locust_user	17	0	100	2200	2200	228.47	96	2177	3144	0.6	0
	Aggregated	17	0	100	2200	2200	228.47	96	2177	3144	0.6	0

File Edit Selection View Go Run ··· CC_Lab-2

__init__.py main.py

main.py > events

58

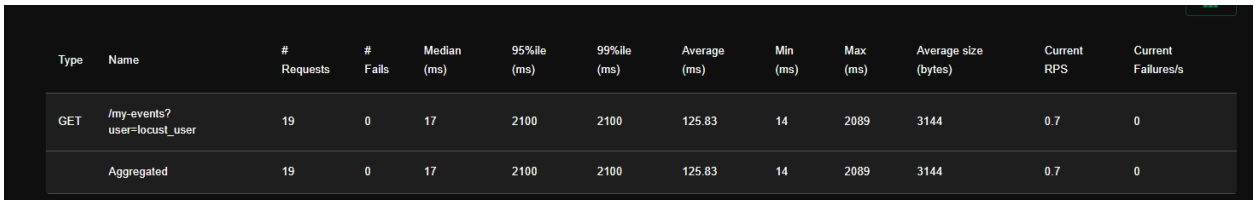
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
GET      /my-events?user=locust_user      17     0(0.00%) |    228      96    2176    100 |    0.59        0.00
--------|----------------------------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated                      17     0(0.00%) |    228      96    2176    100 |    0.59        0.00

Response time percentiles (approximated)
Type     Name                            50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99%   100% # reqs
--------|----------------------------|--------|------|------|------|------|------|------|------|------|------|------|------
GET      /my-events?user=locust_user      100    110    110    120    160   2200   2200   2200   2200   2200   2200     17
--------|----------------------------|--------|------|------|------|------|------|------|------|------|------|------|------
         Aggregated                      100    110    110    120    160   2200   2200   2200   2200   2200   2200     17


(.venv) C:\Users\Prany\OneDrive\Desktop\PES2UG23CS434\CC_Lab-2>
```

SS9: POST OPTIMIZATION



Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events?user=locust_user	19	0	17	2100	2100	125.83	14	2089	3144	0.7	0
	Aggregated	19	0	17	2100	2100	125.83	14	2089	3144	0.7	0

```
2026-01-29T10:10:39Z
[2026-01-29 15:40:39,399] Pran-Vivobook/INFO/locust.main: Shutting down (exit code 0)
Type     Name                                                          # reqs      # fails |    Avg     Min     Max    Med |   req/s  failures/s
--------|-------------------------------------------------------------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /my-events?user=locust_user                                      19     0(0.00%) |    125      13    2089     17 |    0.64        0.00
--------|-------------------------------------------------------------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated                                                       19     0(0.00%) |    125      13    2089     17 |    0.64        0.00

Response time percentiles (approximated)
Type     Name                                                                50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|-------------------------------------------------------------|--------|------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /my-events?user=locust_user                                          17     18     19     19     21   2100   2100   2100   2100   210
0   2100     19
--------|-------------------------------------------------------------|--------|------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated                                                           17     18     19     19     21   2100   2100   2100   2100   210
0   2100     19


(.venv) C:\Users\Prany\OneDrive\Desktop\PES2UG23CS434\CC_Lab-2>
(.venv) C:\Users\Prany\OneDrive\Desktop\PES2UG23CS434\CC_Lab-2>
```

QNA:

Short Explanation for Both Routes

Route: /events

What was the bottleneck?

The events route had extra processing logic that was executed on every request, which increased the response time.

What change did you make?

I removed the unnecessary processing while keeping the database access and page output the same.

Why did the performance improve?

By reducing the amount of work done per request, the server was able to handle the request more efficiently.

Route: /my-events

What was the bottleneck?

The route included redundant operations that caused additional CPU usage during each request.

What change did you make?

I removed the redundant code without changing the functionality of the route.

Why did the performance improve?

Since fewer computations were performed, the response time improved during load testing.