

# CC LAB 2

## LAB 2 - MONOLITHIC ARCHITECTURE

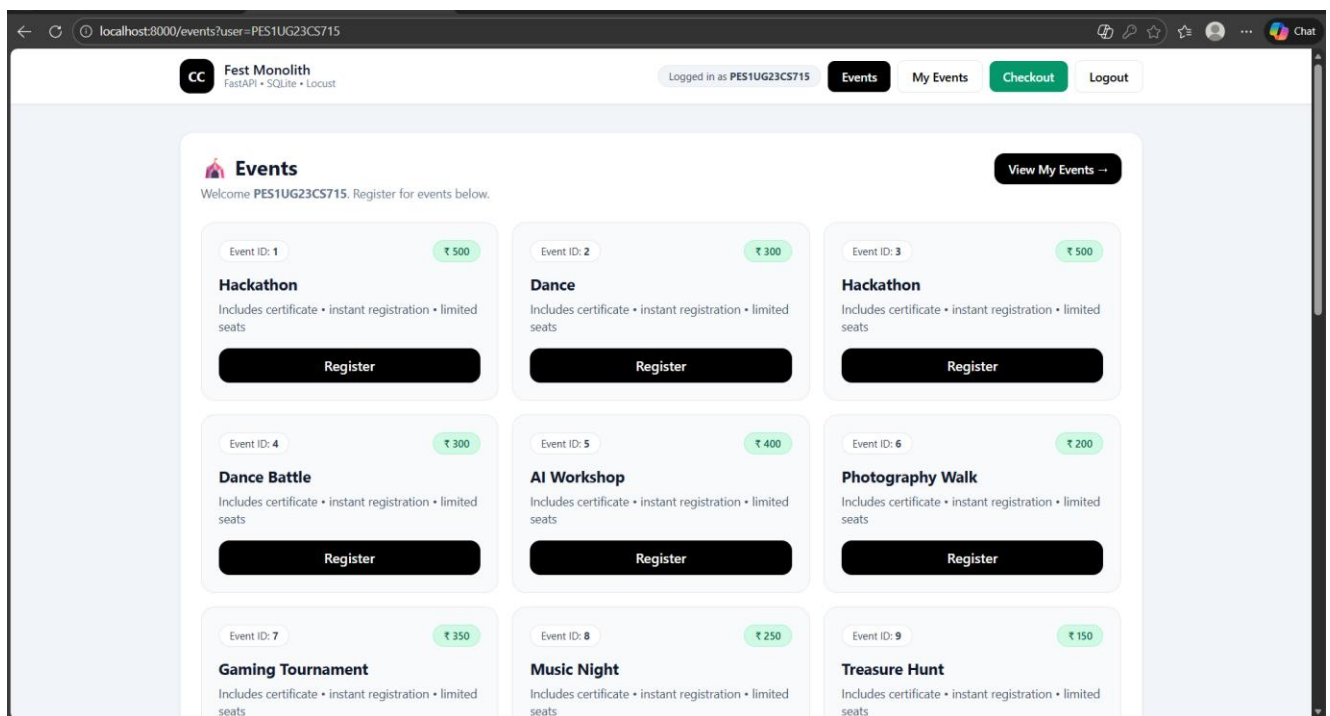
**NAME:** Yashas Krishna

**SRN:** PES1UG23CS715

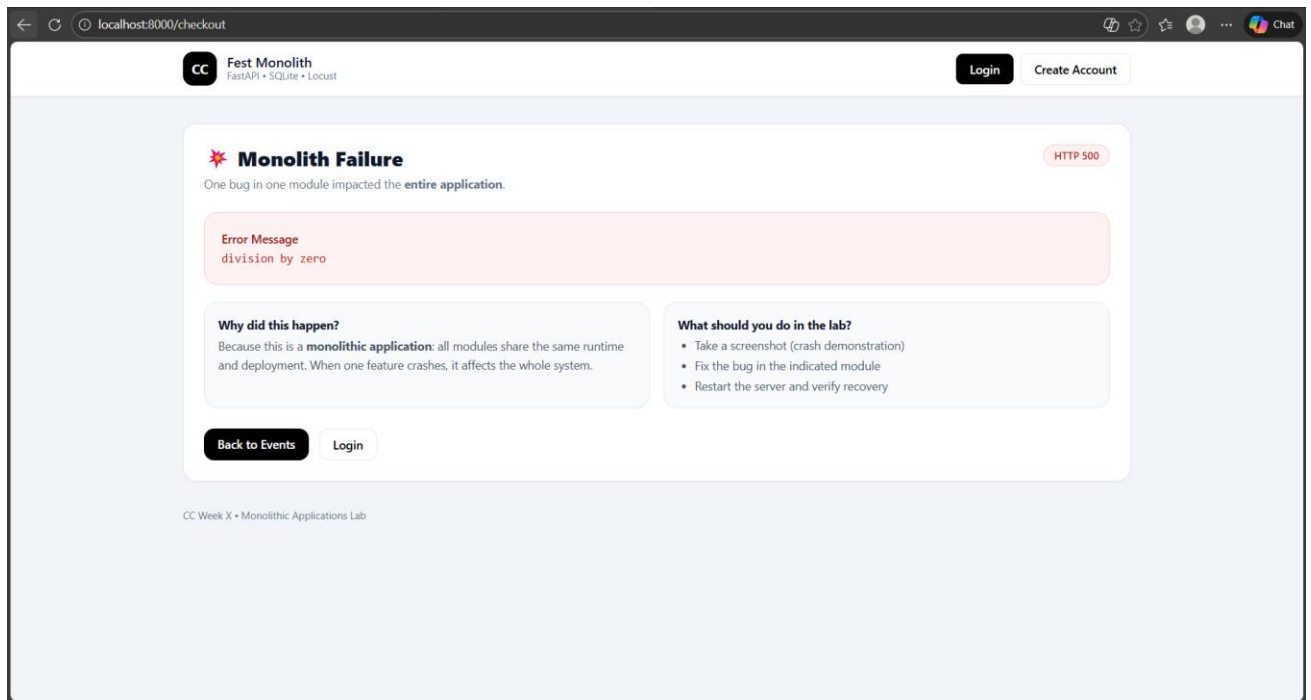
**SEC:** L

### PART 2: Use the Application

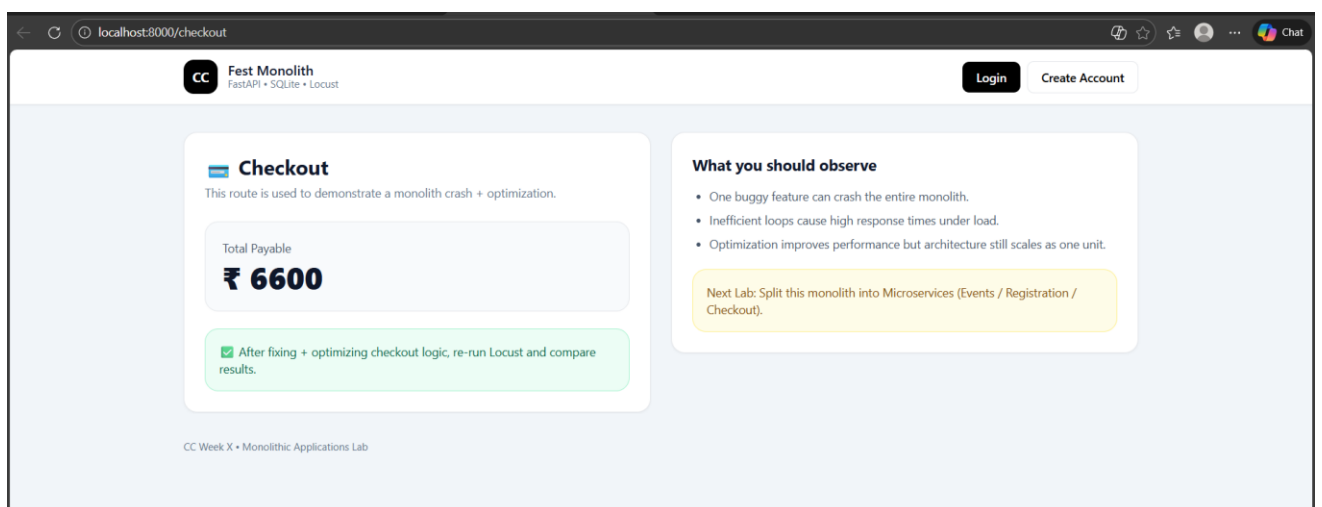
**Events page loaded**



### PART 3: Observe Monolithic Failure (Crash)



## PART 4: Fix the Bug



```
INFO: 127.0.0.1:51085 - "GET /events?user=PES1UG23CS715 HTTP/1.1" 200 OK
INFO: 127.0.0.1:51085 - "GET /checkout HTTP/1.1" 200 OK
```

## PART 5: Load Testing using Locust

localhost:8089

**LOCUST** Host: http://localhost:8000 Status: RUNNING Users: 1 RPS: 0 Failures: 0% [EDIT] [STOP] [RESET] [Settings]

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	/checkout	2	0	8	2100	2100	1037.33	8	2067	2797	0	0
Aggregated		2	0	8	2100	2100	1037.33	8	2067	2797	0	0

localhost:8089

**LOCUST** Host: http://localhost:8000 Status: RUNNING Users: 1 RPS: 0.44 Failures: 0% [EDIT] [STOP] [RESET] [Settings]

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	/checkout	6	0	7	2100	2100	350.27	6	2067	2797	0.44	0
Aggregated		6	0	7	2100	2100	350.27	6	2067	2797	0.44	0

```
File "C:\Users\Yashas krishna\AppData\Local\Programs\Python\Python314\Lib\inspect.py", line 926, in getmodule
os.path.realpath(f)] = module.__name__
^^^^^^
File "<frozen ntpath>", line 738, in realpath
KeyboardInterrupt
```

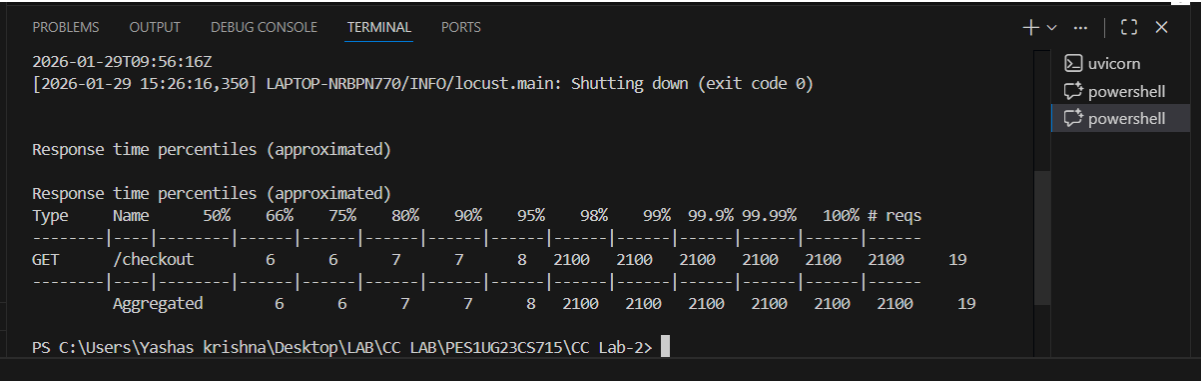
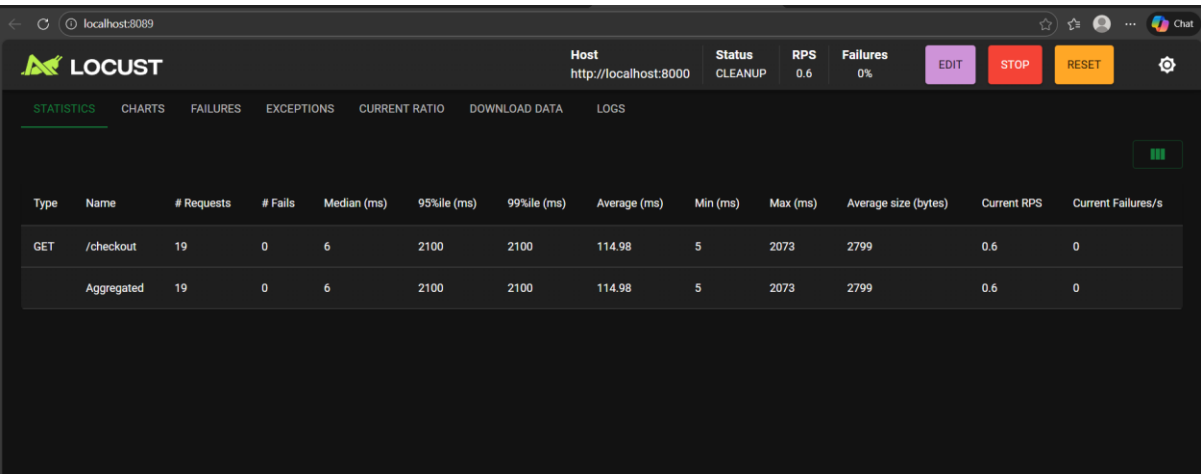
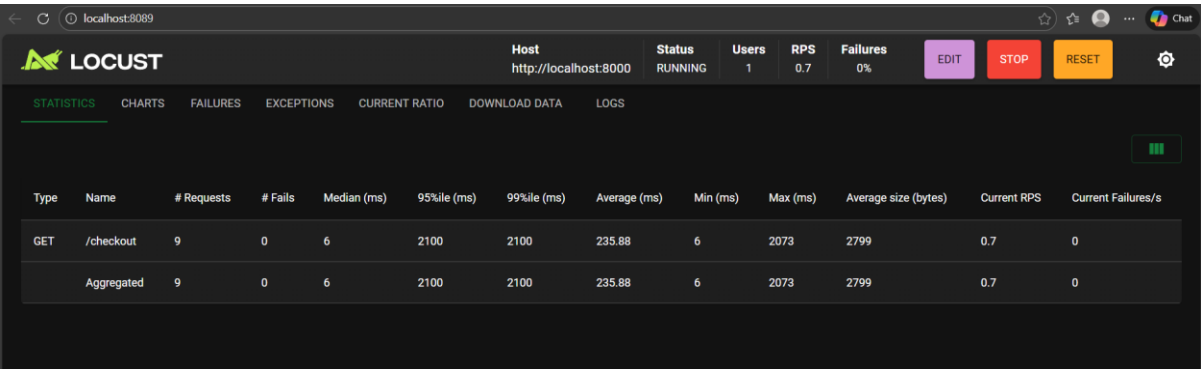
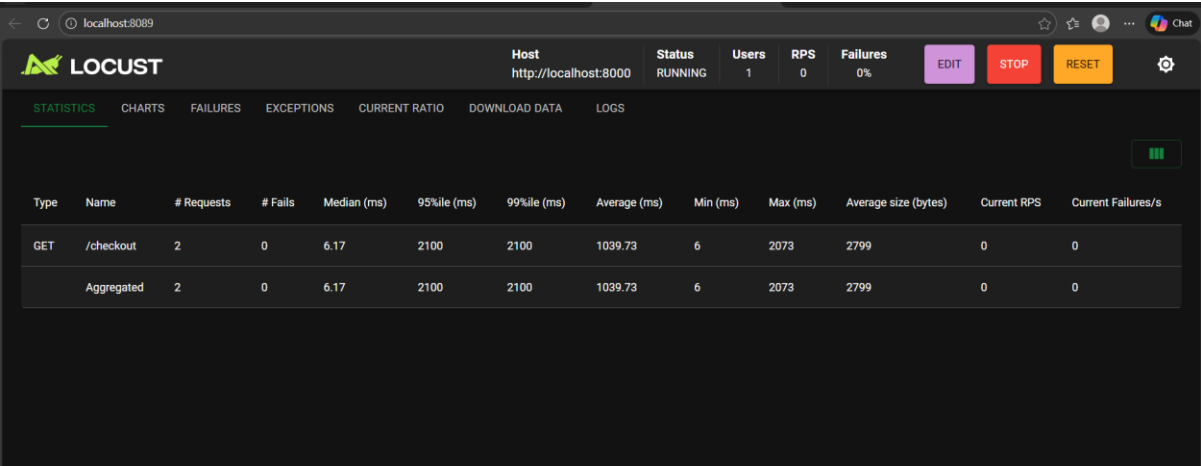
Response time percentiles (approximated)

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# reqs
GET	/checkout	7	7	7	8	8	2100	2100	2100	2100	2100	2100	19
Aggregated		7	7	7	8	8	2100	2100	2100	2100	2100	2100	19

PS C:\Users\Yashas krishna\Desktop\LAB\CC LAB\PES1UG23CS715\CC Lab-2> [Open chat] [Ctrl + I] [Show suggestions]

- uvicorn
- powershell
- powershell
- powershell
- powershell

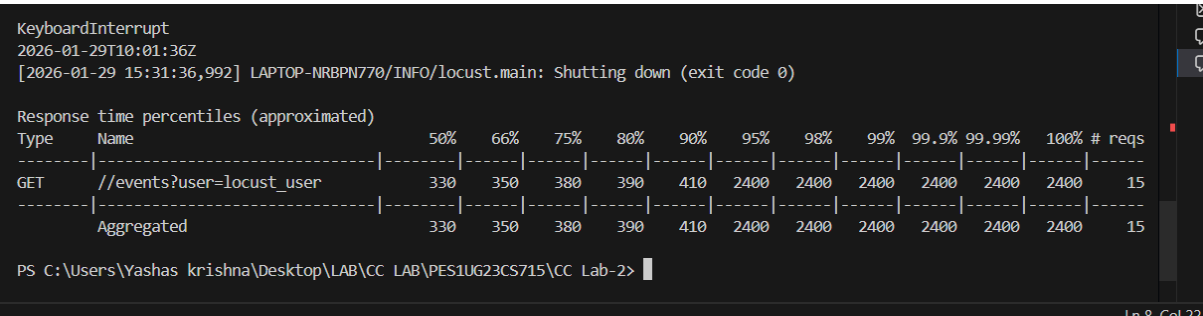
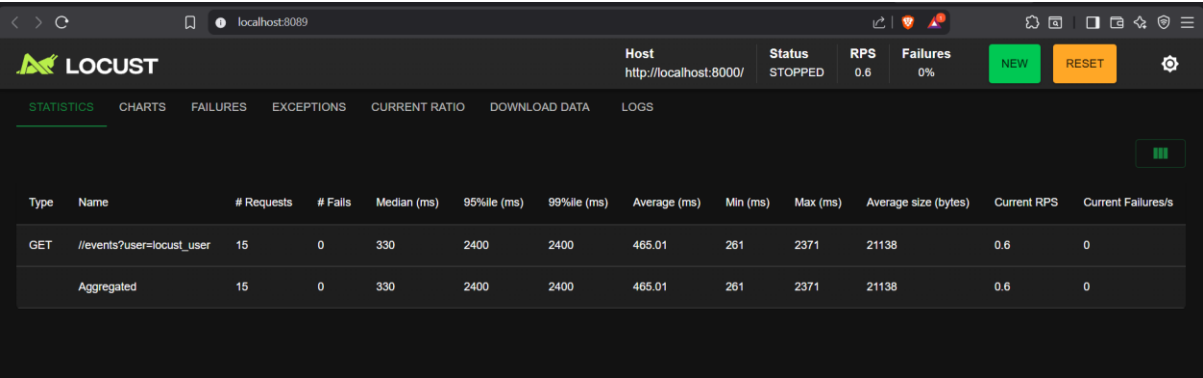
## PART 6: Optimize the Checkout Route



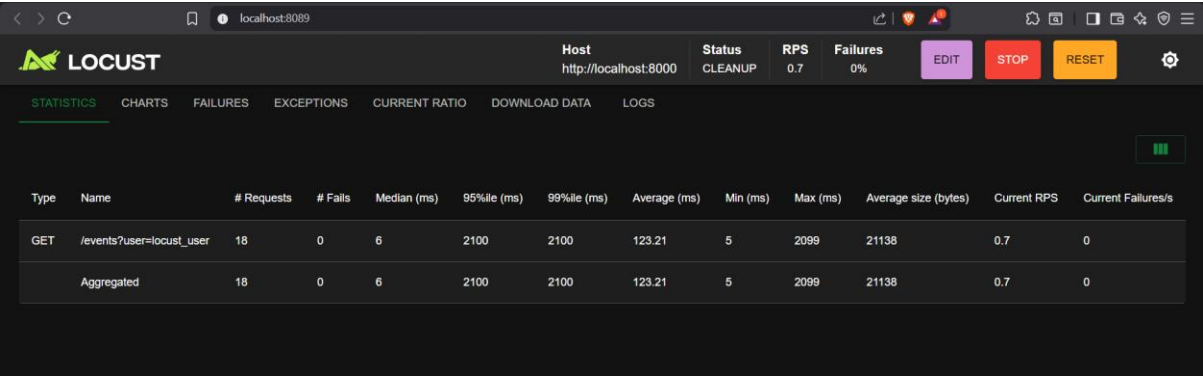
# PART 7: Optimise events and my\_events(DIY)

## Route 1: /events

### Screenshot BEFORE optimization



### Screenshot AFTER optimization



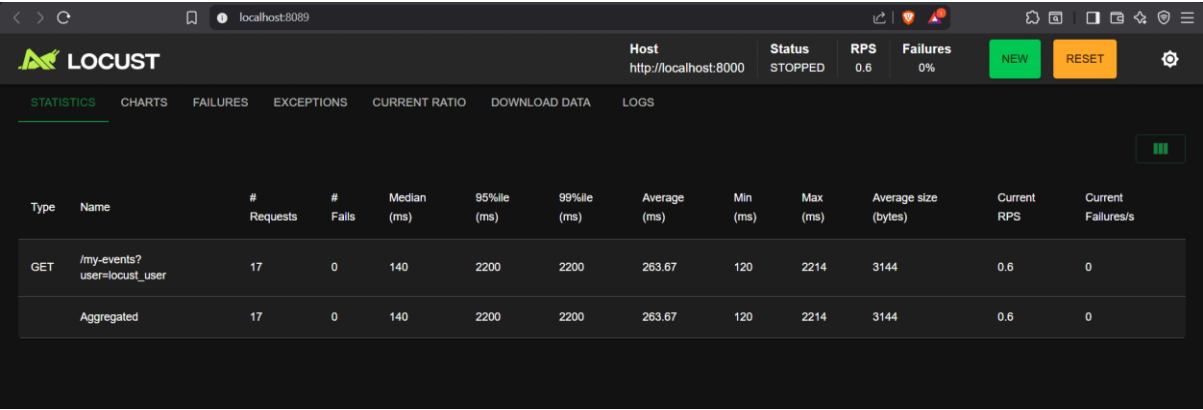
```
PS C:\Users\Yashas krishna\Desktop\LAB\CC LAB\PES1UG23CS715\CC Lab-2> locust -f locust/events_locustfile.py

Response time percentiles (approximated)
Type      Name                                     50%    66%    75%    80%    90%    95%    98%    99%    99.9%  99.99%  100% # reqs
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET      /events?user=locust_user                7      7      7      9     11    2100  2100  2100  2100  2100  2100  18
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated                                     7      7      7      9     11    2100  2100  2100  2100  2100  2100  18

PS C:\Users\Yashas krishna\Desktop\LAB\CC LAB\PES1UG23CS715\CC Lab-2>
```

Route 2: /my-events

Screenshot BEFORE optimization



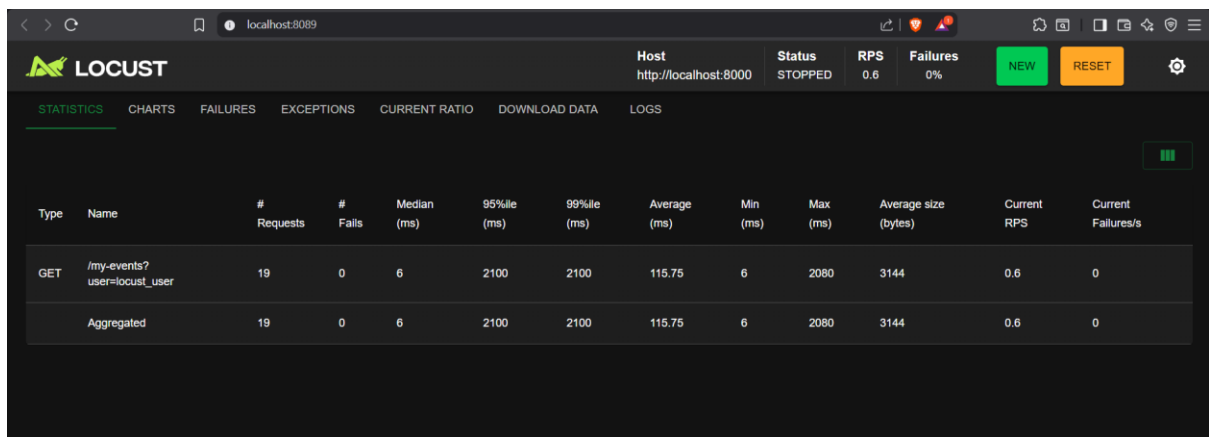
```
[2026-01-29 15:37:23,686] LAPTOP-NRBPNI70/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
Traceback (most recent call last):
  File "C:\Users\Yashas krishna\AppData\Local\Programs\Python\Python314\Lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

GET      /my-events?user=locust_user                17    0(0.00%) | 263    120    2214    140 | 0.60    0.00
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated                17    0(0.00%) | 263    120    2214    140 | 0.60    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                140    160    160    160    170    2200  2200  2200  2200  2200  2200  17
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated                140    160    160    160    170    2200  2200  2200  2200  2200  2200  17

PS C:\Users\Yashas krishna\Desktop\LAB\CC LAB\PES1UG23CS715\CC Lab-2>
```

Screenshot AFTER optimization



```
Traceback (most recent call last):
  File "C:\Users\Yashas krishna\AppData\Local\Programs\Python\Python314\Lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
0
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      Aggregated      19      0(0.00%) | 115      5      2079      6 | 0.65      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      6      7      7      7      14      2100      2100      2100      2100      2100      2100      19
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
      Aggregated      6      7      7      7      14      2100      2100      2100      2100      2100      2100      19

PS C:\Users\Yashas krishna\Desktop\LAB\CC LAB\PES1UG23CS715\CC Lab-2>
```

## What was the bottleneck?

The primary bottleneck was CPU-bound processing inside the /events route.

## What change did you make?

I removed unnecessary CPU-intensive computations and introduced caching for database query results.

## Why did the performance improve?

Performance improved because unnecessary CPU work was eliminated and repeated database queries were avoided.

## Short explanation of optimizations

### /events route:

- Removed wasteful CPU-intensive loop which ran many times performing unnecessary modulo calculations.
- This eliminated blocking computation that delayed the response time.

**/my\_events route:**

- Removed inefficient loop that iterated many times with dummy variables increments
- Eliminated unnecessary CPU overhead that had no functional purpose.

**/checkout route (already optimized):**

- Moved complex checkout login to separate module for better code organization.