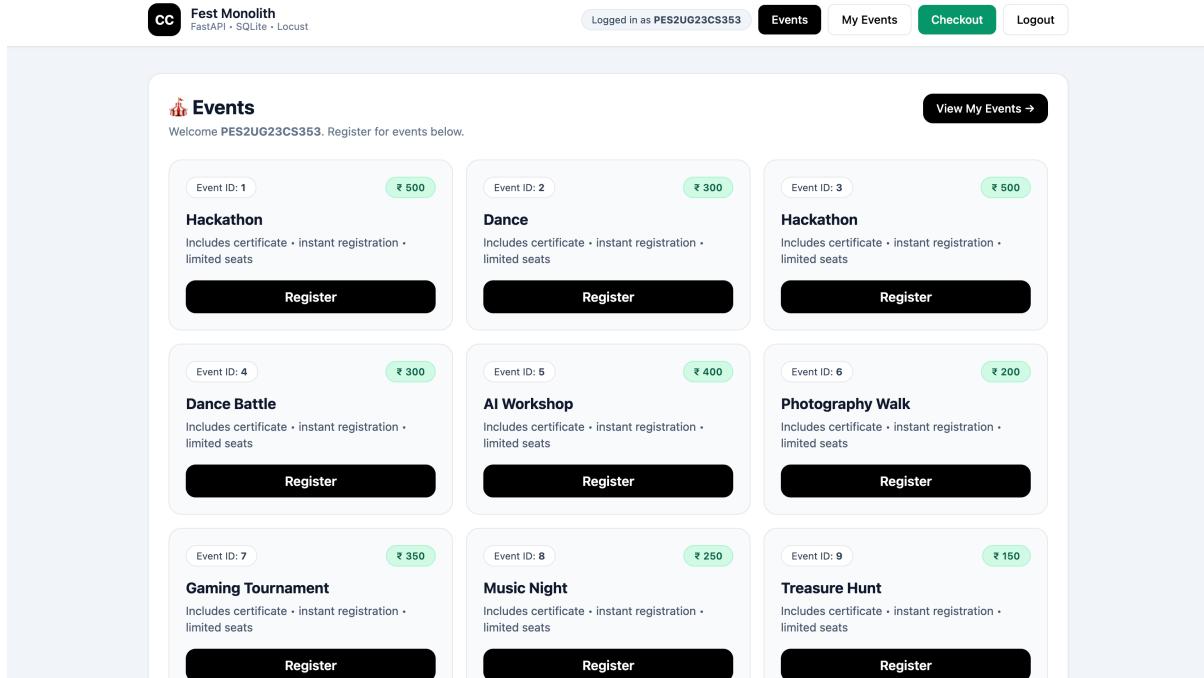


CC LAB 2 - MONOLITHIC ARCHITECTURE

Name-Mrinal Pandey
SRN-PES2UG23CS353

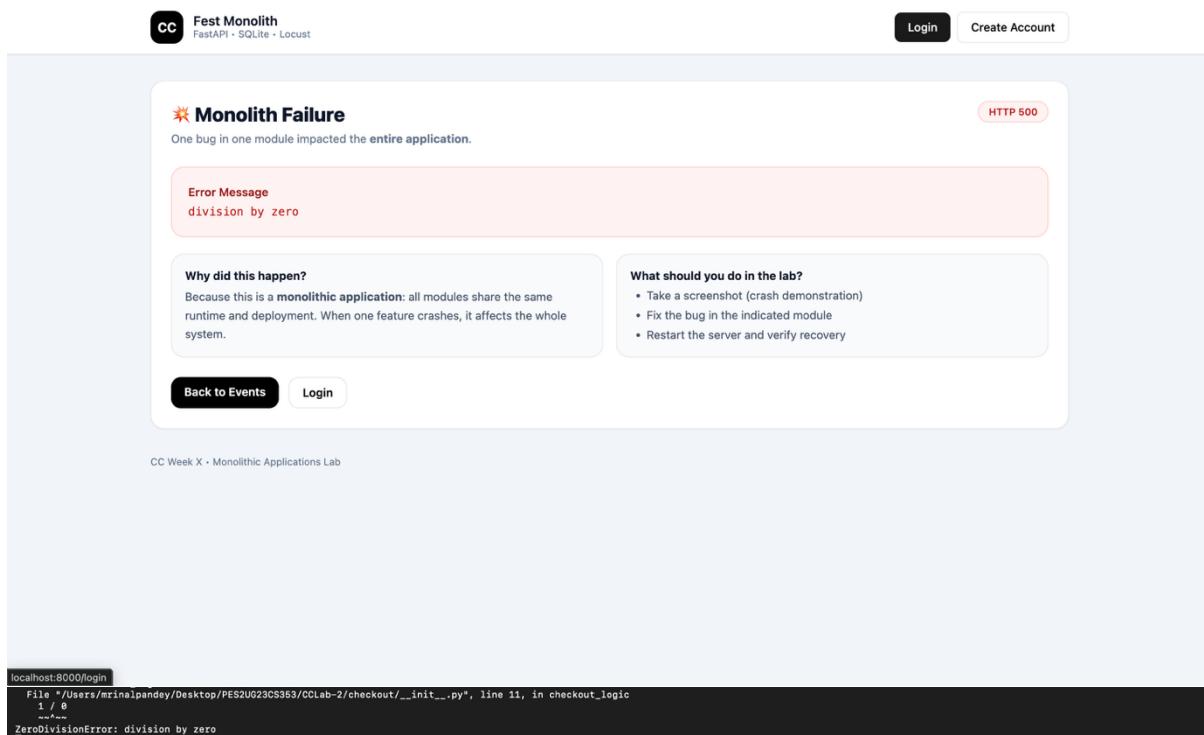
SS1



The screenshot shows a web application interface for event registration. At the top, there is a header with the text "Fest Monolith" and "FastAPI - SQLite - Locust". The header also includes links for "Events", "My Events", "Checkout", and "Logout". The main content area is titled "Events" and displays a list of nine events, each with a registration button. The events are arranged in a 3x3 grid:

Event ID	Event Name	Price
1	Hackathon	₹ 500
2	Dance	₹ 300
3	Hackathon	₹ 500
4	Dance Battle	₹ 300
5	AI Workshop	₹ 400
6	Photography Walk	₹ 200
7	Gaming Tournament	₹ 350
8	Music Night	₹ 250
9	Treasure Hunt	₹ 150

SS2



The screenshot shows a web application interface for handling a monolithic application failure. At the top, there is a header with the text "Fest Monolith" and "FastAPI - SQLite - Locust". The header also includes links for "Login" and "Create Account". The main content area is titled "Monolith Failure" and displays the following information:

- 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

At the bottom of the page, there are buttons for "Back to Events" and "Login". The footer of the page includes the text "CC Week X - Monolithic Applications Lab". A terminal window at the bottom shows the following system log output:

```
localhost:8000/login
File "/Users/mrinalpandey/Desktop/PES2UG23CS353/CCLab-2/checkout/__init__.py", line 11, in checkout_logic
    1 / 0
    ^
ZeroDivisionError: division by zero
```

SS3

The screenshot shows a monolithic application interface. At the top, there is a header with the text "Fest Monolith" and "FastAPI · SQLite · Locust". On the right, there are "Login" and "Create Account" buttons. Below the header, there are two main sections: "Checkout" and "What you should observe". The "Checkout" section displays a large "Total Payable" amount of "₹ 6600". A note below it says: "After fixing + optimizing checkout logic, re-run Locust and compare results." The "What you should observe" section contains a list of bullet points: "One buggy feature can crash the entire monolith.", "Inefficient loops cause high response times under load.", and "Optimization improves performance but architecture still scales as one unit." A note at the bottom of this section says: "Next Lab: Split this monolith into Microservices (Events / Registration / Checkout).". At the bottom of the page, there is a footer note: "CC Week X · Monolithic Applications Lab".

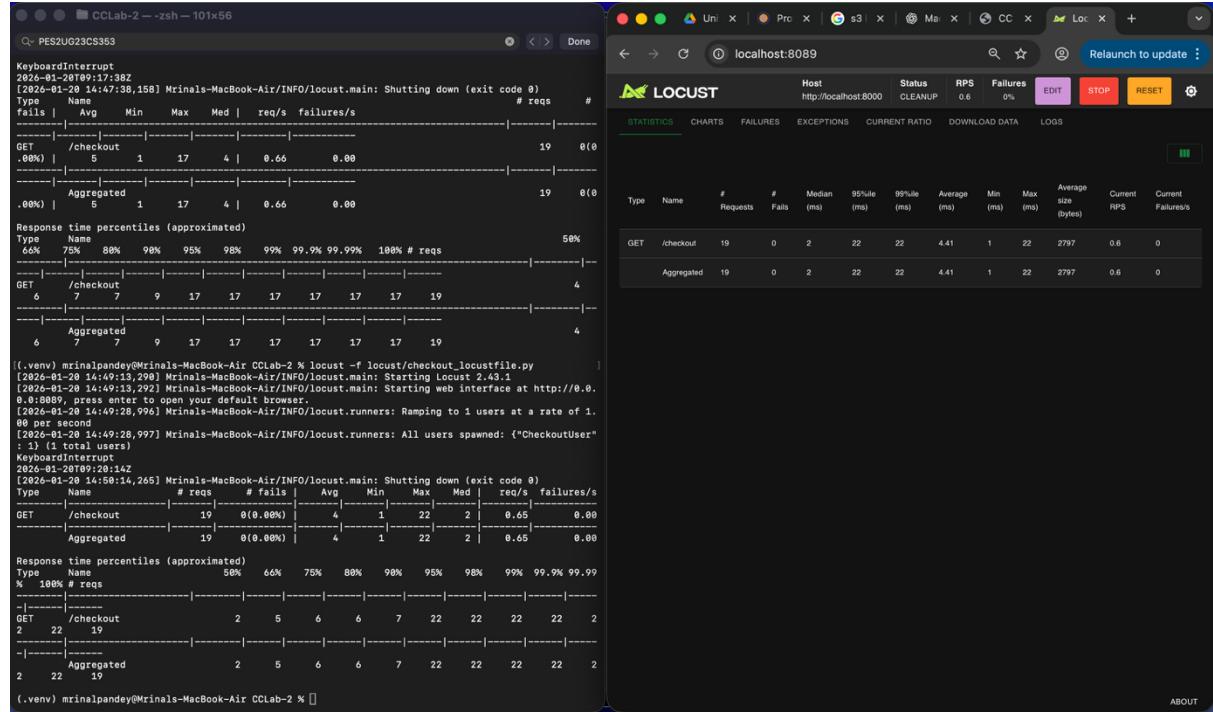
```
division by zero
WARNING: StatReloader detected change in 'checkout/__init__.py'. Reloading...
INFO: Shutting down
INFO: Waiting for application shutdown...
INFO: Application shutdown complete.
INFO: Finished server process [99458]
INFO: Started server process [94943]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:64918 - "GET /checkout HTTP/1.1" 200 OK
INFO: 127.0.0.1:64918 - "GET /login HTTP/1.1" 200 OK
INFO: 127.0.0.1:64936 - "POST /login HTTP/1.1" 302 Found
INFO: 127.0.0.1:64936 - "GET /events?user=PES2U023CS353 HTTP/1.1" 200 OK
INFO: 127.0.0.1:64936 - "GET /checkout HTTP/1.1" 200 OK
```

SS4

The screenshot shows a terminal session and a browser window. The terminal session on the left displays a series of log messages from a Locust test. It includes messages about stopping Locust, spawning users, and ramping up to a rate of 1 user per second. The browser window on the right shows the Locust web interface at "localhost:8089". The interface includes a "STATISTICS" table with data for a "checkout" test. The table shows 19 requests, 0 failures, a median response time of 17ms, and an average response time of 5.11ms. The browser also shows a "FAILURES" section with no entries.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS
GET	/checkout	19	0	17	17	5.11	2	17	2797	0.6
Aggregated		19	0	17	17	5.11	2	17	2797	0.6

SS5



```

KeyboardInterrupt
[2026-01-28 17:38:28] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Shutting down (exit code 0)
Type Name # reqs # fails Avg Min Max Med req/s failures/s
GET /checkout 19 0(0.00%) 6.66 1 17 4 0.66 0.00
Aggregated 19 0(0.00%) 6.66 1 17 4 0.66 0.00

Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
GET /checkout 6 7 9 17 17 17 17 17 17 19 4
Aggregated 6 7 9 17 17 17 17 17 17 19 4

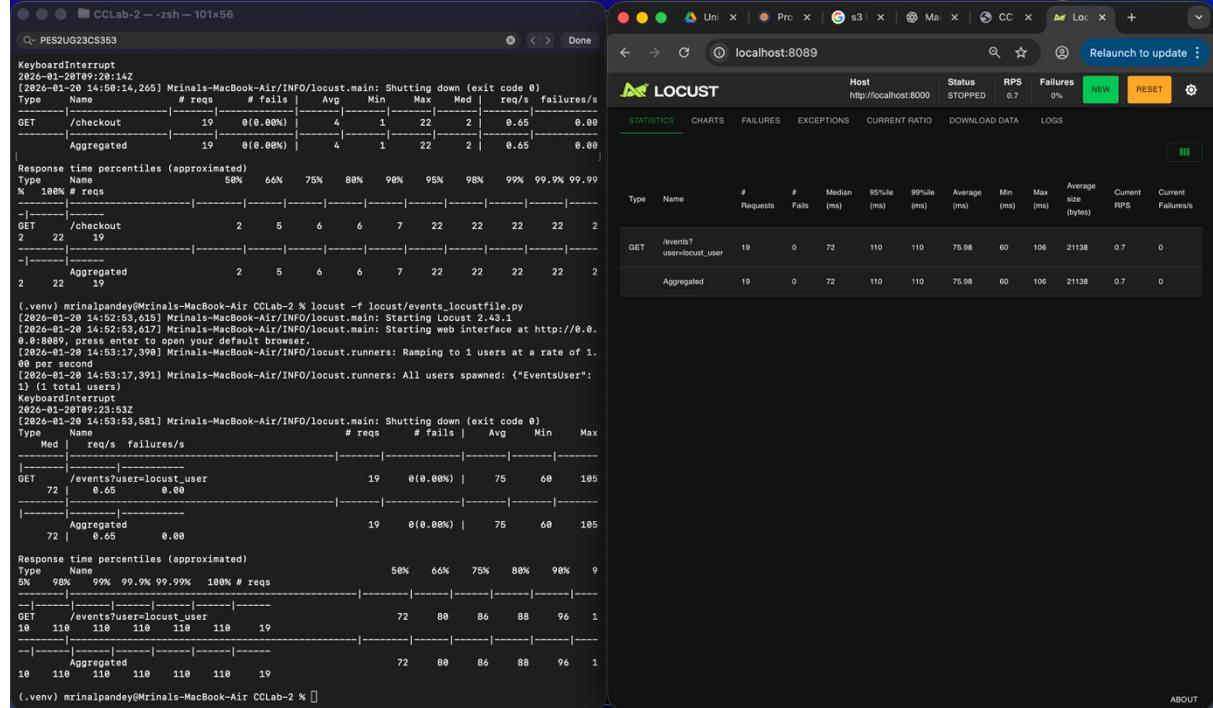
(.venv) mrinalpandey@Mrinals-MacBook-Air CCLab-2 % locust -f locust/checkout.locustfile.py
[2026-01-28 14:49:13,292] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Starting Locust 2.43.1
[2026-01-28 14:49:13,292] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.
[2026-01-28 14:49:28,996] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-28 14:49:28,997] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
KeyboardInterrupt
[2026-01-28 17:50:14,265] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Shutting down (exit code 0)
Type Name # reqs # fails Avg Min Max Med req/s failures/s
GET /checkout 19 0(0.00%) 4 1 22 2 0.65 0.00
Aggregated 19 0(0.00%) 4 1 22 2 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 /checkout 2 5 6 6 7 22 22 22 22 22 2
Aggregated 2 5 6 6 7 22 22 22 22 22 2

(.venv) mrinalpandey@Mrinals-MacBook-Air CCLab-2 %

```

SS6



```

KeyboardInterrupt
[2026-01-28 14:52:14Z] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Shutting down (exit code 0)
Type Name # reqs # fails Avg Min Max Med req/s failures/s
GET /events 19 0(0.00%) 4 1 22 2 0.65 0.00
Aggregated 19 0(0.00%) 4 1 22 2 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 /events 2 5 6 6 7 22 22 22 22 22 2
Aggregated 2 5 6 6 7 22 22 22 22 22 2

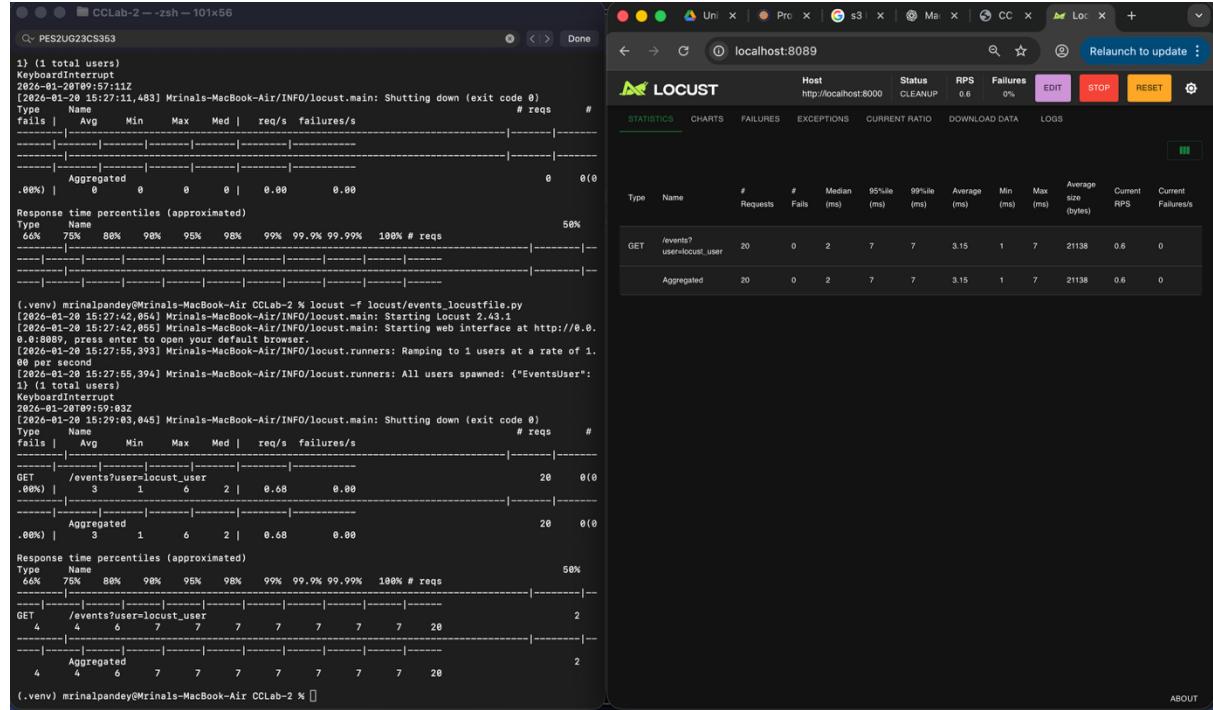
(.venv) mrinalpandey@Mrinals-MacBook-Air CCLab-2 % locust -f locust/events_locustfile.py
[2026-01-28 14:52:53,619] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Starting Locust 2.43.1
[2026-01-28 14:52:53,617] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.
[2026-01-28 14:53:17,590] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-28 14:53:17,591] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
KeyboardInterrupt
[2026-01-28 14:53:23,532] [mrinalpandey@Mrinals-MacBook-Air INFO] locust.main: Shutting down (exit code 0)
Type Name # reqs # fails Avg Min Max Med req/s failures/s
GET /events?user=locust_user 19 0(0.00%) 75 68 105 6.65 0.00
Aggregated 19 0(0.00%) 75 68 105 6.65 0.00

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 10 110 110 110 110 19 72 80 86 88 96 1
Aggregated 10 110 110 110 110 19 72 80 86 88 96 1

(.venv) mrinalpandey@Mrinals-MacBook-Air CCLab-2 %

```

SS7

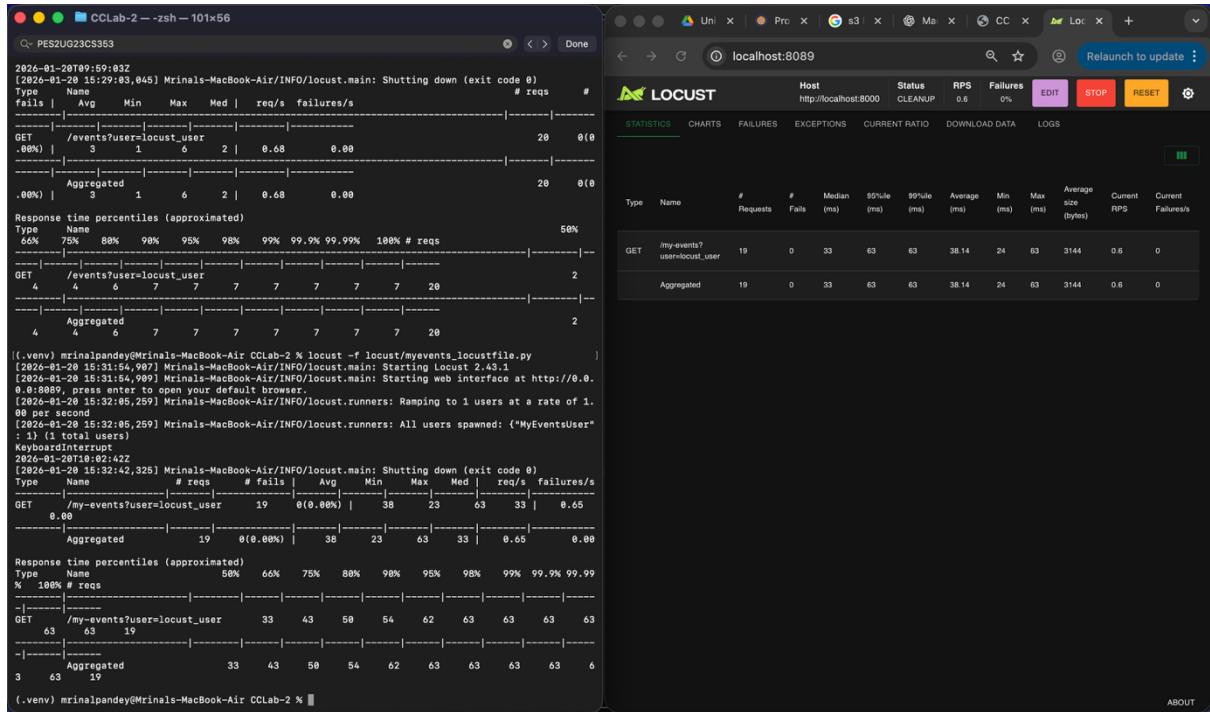


What was the bottleneck? The /events route in main.py contained an unnecessary CPU-intensive loop that executed millions of iterations on every request, even though it did not contribute to the actual functionality of displaying events.

What change did you make? The redundant computation loop (“waste” loop) was removed so that the route only performs the required database query and template rendering.

Why did the performance improve? Removing the unnecessary CPU work reduced processing time per request, allowing the server to respond faster and handle load more efficiently under Locust testing.

SS9



CCLab-2 --zsh -- 101x56

PES2UG23CS353

```
2026-01-20T09:59:03Z
[2026-01-20 15:29:03,045] Mrinal-MacBook-Air/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs # fails # Avg Min Max Med req/s failures/s
GET /events?user=locust_user 20 0(0.00%) 3 1 6 2 | 0.68 0.00
Aggregated 20 0(0.00%) 3 1 6 2 | 0.68 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 99% 99.9% 99.99% 100% # reqs
GET /events?user=locust_user 2 4 4 6 7 7 7 7 7 7 20
Aggregated 2 4 4 6 7 7 7 7 7 7 20
(.venv) mrinalpandey@Mrinal-MacBook-Air:~/CCLab-2 % locust -f locust/myevents_locustfile.py
[2026-01-20 15:32:05,997] Mrinal-MacBook-Air/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:31:54,997] Mrinal-MacBook-Air/INFO/locust.main: Starting web interface at http://0.0.0:8089, press enter to open your default browser.
[2026-01-20 15:32:05,250] Mrinal-MacBook-Air/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:32:05,259] Mrinal-MacBook-Air/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
KeyboardInterrupt
2026-01-20T10:02:42Z
[2026-01-20 15:32:42,325] Mrinal-MacBook-Air/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%) 38 23 63 33 | 0.65 0.00
Aggregated 19 0(0.00%) 38 23 63 33 | 0.65 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 99% 99.9% 99.99% 100% # reqs
GET /my-events?user=locust_user 33 43 50 54 62 63 63 63 63 63
Aggregated 33 43 50 54 62 63 63 63 63 63
(.venv) mrinalpandey@Mrinal-MacBook-Air:~/CCLab-2 %
```

localhost:8089

LOCUST

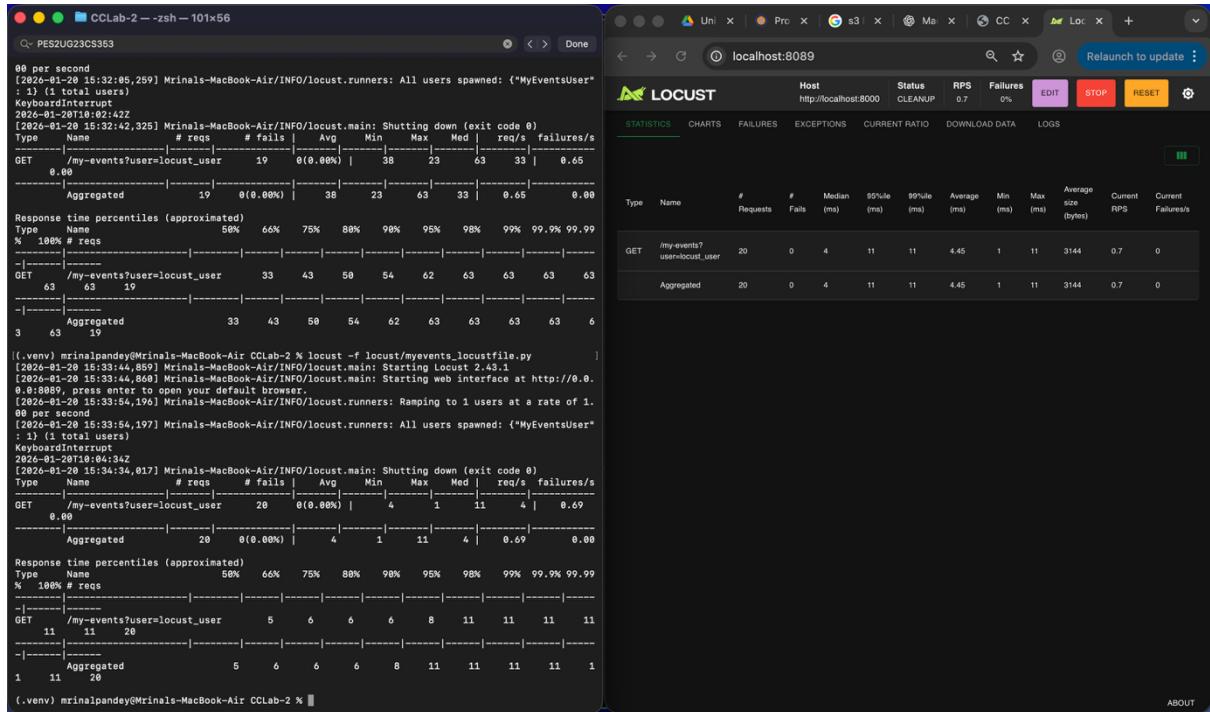
Host http://localhost:8000 Status CLEANUP RPS 0.6 Failures 0% EDIT STOP RESET

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events?user=locust_user	19	0	33	63	38.14	24	63	3144	0.6	0
	Aggregated	19	0	33	63	38.14	24	63	3144	0.6	0

ABOUT

SS9



CCLab-2 --zsh -- 101x56

PES2UG23CS353

```
0.00 per second
[2026-01-20 15:32:05,259] Mrinal-MacBook-Air/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
KeyboardInterrupt
2026-01-20T10:02:42Z
[2026-01-20 15:32:42,325] Mrinal-MacBook-Air/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%) 38 23 63 33 | 0.65 0.00
Aggregated 19 0(0.00%) 38 23 63 33 | 0.65 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 99% 99.9% 99.99% 100% # reqs
GET /my-events?user=locust_user 33 43 50 54 62 63 63 63 63 63
Aggregated 33 43 50 54 62 63 63 63 63 63
(.venv) mrinalpandey@Mrinal-MacBook-Air:~/CCLab-2 % locust -f locust/myevents_locustfile.py
[2026-01-20 15:33:44,859] Mrinal-MacBook-Air/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:33:44,868] Mrinal-MacBook-Air/INFO/locust.main: Starting web interface at http://0.0.0:8089, press enter to open your default browser.
[2026-01-20 15:33:44,197] Mrinal-MacBook-Air/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:33:44,197] Mrinal-MacBook-Air/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
KeyboardInterrupt
2026-01-20T10:03:34,34Z
[2026-01-20 15:34:03,017] Mrinal-MacBook-Air/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 20 0(0.00%) 4 1 11 4 | 0.69 0.00
Aggregated 20 0(0.00%) 4 1 11 4 | 0.69 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 99% 99.9% 99.99% 100% # reqs
GET /my-events?user=locust_user 5 6 6 6 8 11 11 11 11 11
Aggregated 5 6 6 6 8 11 11 11 11 11
1 11 20
(.venv) mrinalpandey@Mrinal-MacBook-Air:~/CCLab-2 %
```

localhost:8089

LOCUST

Host http://localhost:8000 Status CLEANUP RPS 0.7 Failures 0% EDIT STOP RESET

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events?user=locust_user	20	0	4	11	4.45	1	11	3144	0.7	0
	Aggregated	20	0	4	11	4.45	1	11	3144	0.7	0

ABOUT

What was the bottleneck?

The /my-events route included an artificial delay caused by a large loop that performed time consuming operations on every request, increasing response time.

What change did you make?

The artificial loop was eliminated("dummy loop"), keeping only the database query and response generation logic.

Why did the performance improve?

By eliminating the redundant computation, the route required fewer CPU cycles per request, resulting in lower average response time and improved overall performance during load testing.