

UE23CS351B - CLOUD COMPUTING LAB 2 -> MONOLITHIC ARCHITECTURE

Name: Neha Rajkumar Patil

SRN: PES2UG23CS379

Sec: F

The image shows two screenshots of a monolithic application interface. The top screenshot displays a list of events with registration buttons. The bottom screenshot shows a failure page with error logs at the bottom.

Events Page Screenshot:

- Event ID: 1 - Hackathon (₹ 500), Register button
- Event ID: 2 - Dance (₹ 300), Register button
- Event ID: 3 - Hackathon (₹ 500), Register button
- Event ID: 4 - Dance Battle (₹ 300), Register button
- Event ID: 5 - AI Workshop (₹ 400), Register button
- Event ID: 6 - Photography Walk (₹ 200), Register button
- Event ID: 7 - Gaming Tournament (₹ 350), Register button
- Event ID: 8 - Music Night (₹ 250), Register button
- Event ID: 9 - Treasure Hunt (₹ 150), Register button

Failure Page Screenshot:

Monolith Failure
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**

CC Week X • Monolithic Applications Lab

```
INFO: 127.0.0.1:54268 - "GET /events?user=PES2UG23CS379 HTTP/1.1" 200 OK
INFO: 127.0.0.1:54268 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
```

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

```
[INFO]: Application startup complete.  
[INFO]: 127.0.0.1:63483 - "GET /checkout HTTP/1.1" 200 OK
```

Load Testing using Locust

Start new load test

Number of users (peak concurrency) *

1

Ramp up (users started/second) *

1

Host

http://localhost:8000

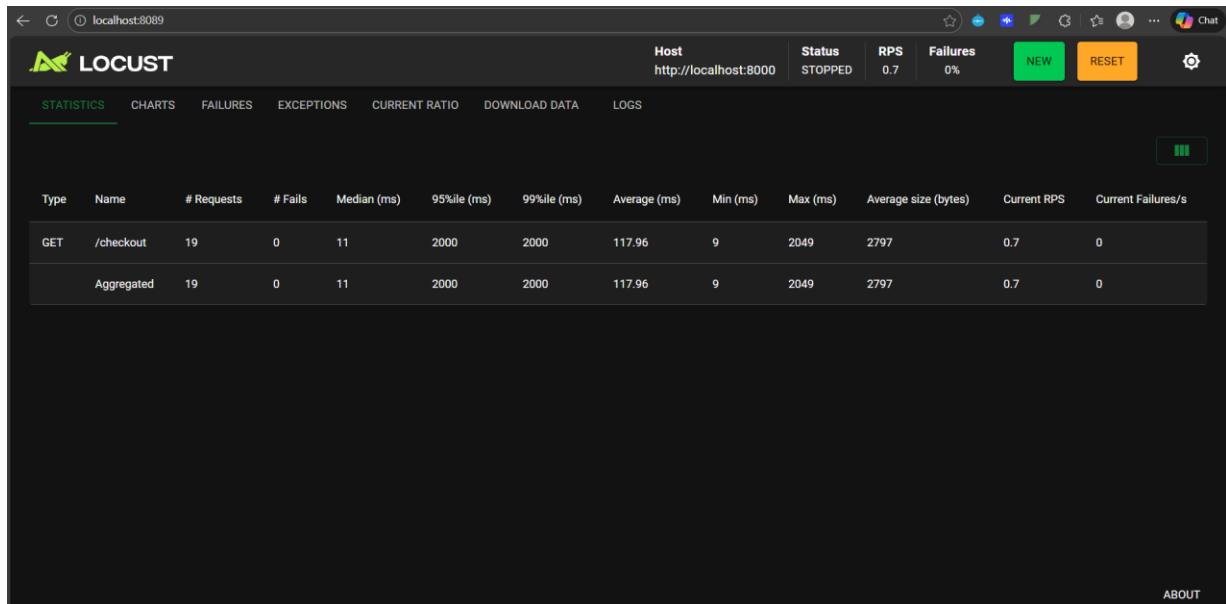
Advanced options

Run time (e.g. 20, 20s, 3m, 2h, 1h20m, 3h30m10s, etc.)

30

Profile

START



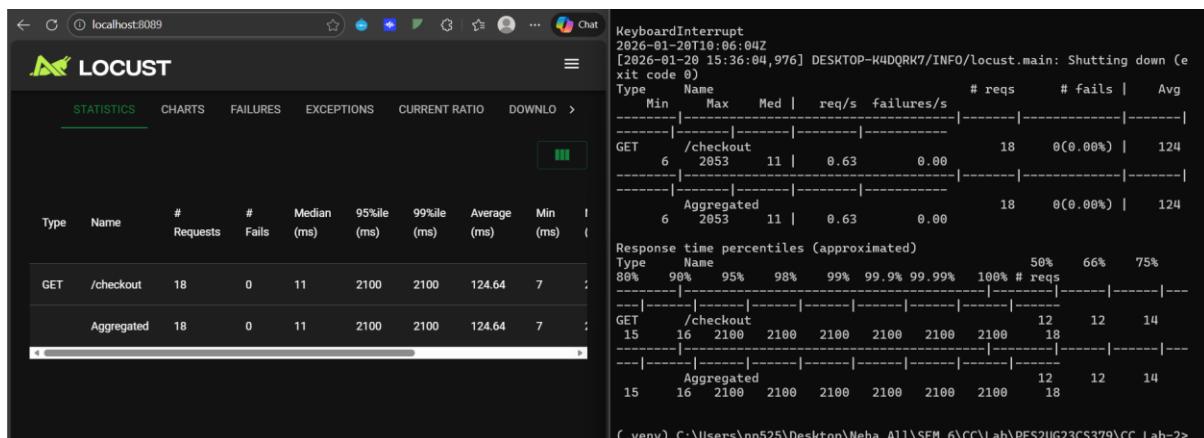
Optimize the Checkout Route

```
Traceback (most recent call last):
  File "C:\Users\np525\Desktop\Neha_All\SEM_6\CC\Lab\PES2UG23CS379\.venv\lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

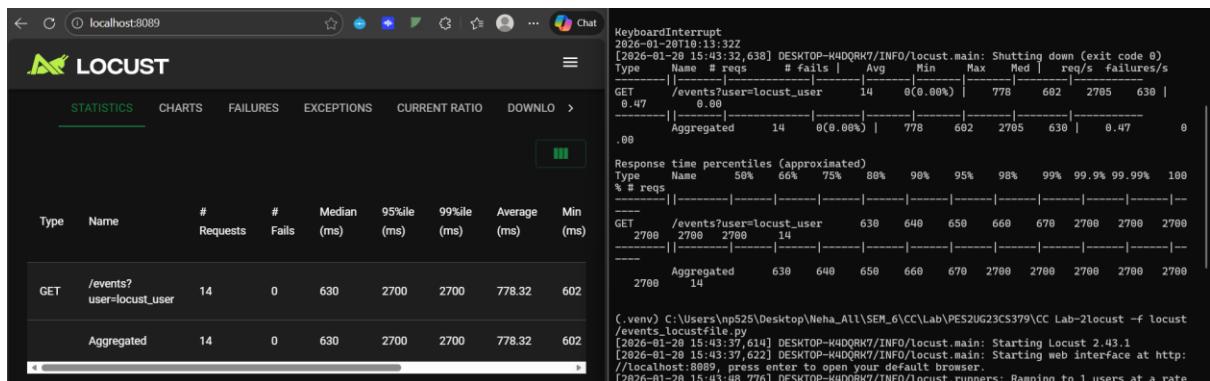
KeyboardInterrupt
2026-01-20T09:39:36Z
[2026-01-20 15:09:36,701] DESKTOP-K4DQRK7/INFO/locust.main: Shutting down (exit code 0)
Type      Name          # reqs   # fails   Avg   Min   Max   Med   req/s failures/s
GET      /checkout      19       0(0.00%) 117.96 9     2049   11    0.65   0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET      Aggregated    19       0(0.00%) 117.96 9     2049   11    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      11     11     11     11     12     2000  2000  2000  2000  2000  2000  19
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET      Aggregated    11     11     11     11     12     2000  2000  2000  2000  2000  2000  19

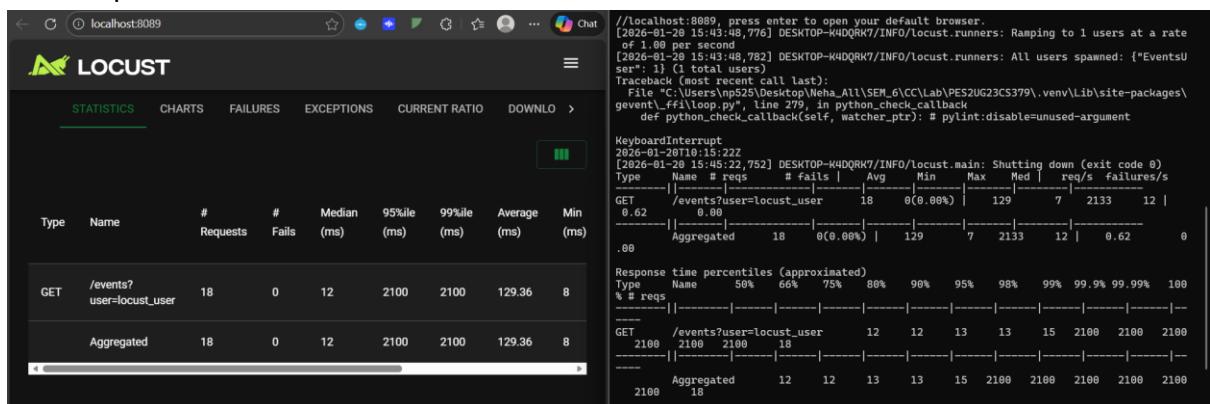
(.venv) C:\Users\np525\Desktop\Neha_All\SEM_6\CC\Lab\PES2UG23CS379\CC_Lab-2>
```



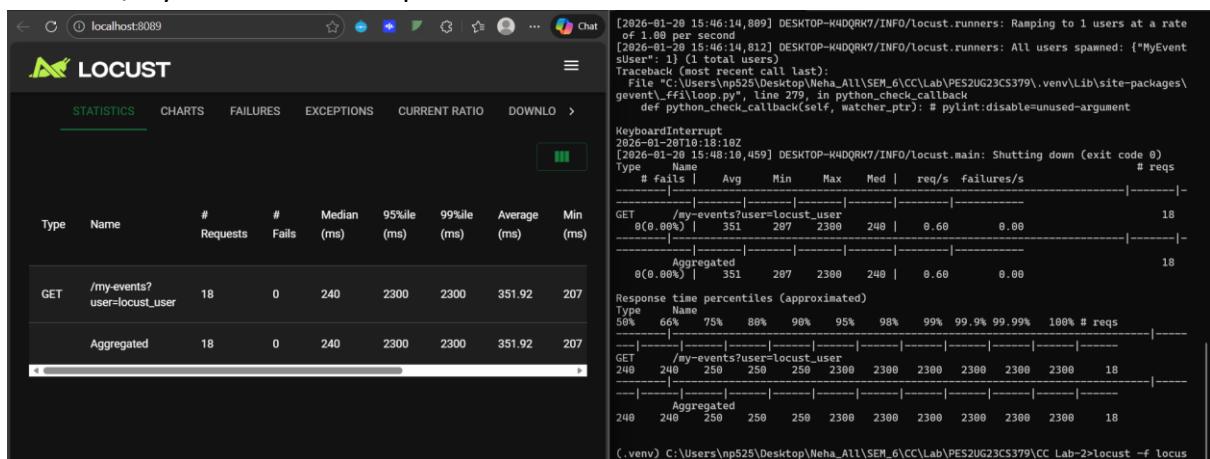
Route 1: /events BEFORE optimization



AFTER optimization



Route 2: /my-events BEFORE optimization



AFTER optimization

