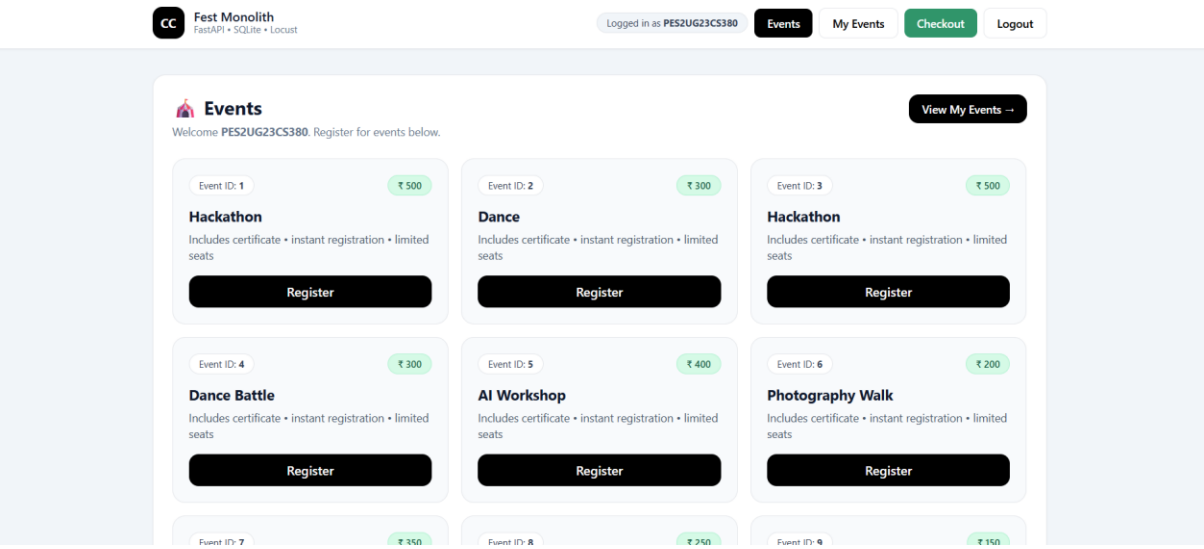
LAB 2 - MONOLITHIC ARCHITECTURE

NAME: NEHAL.G

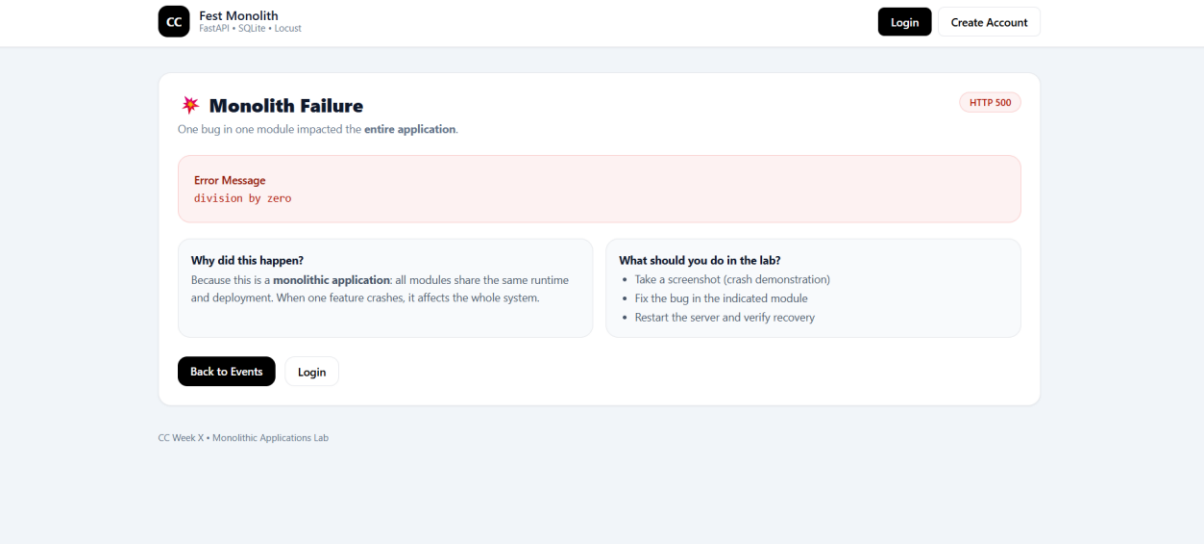
SRN: PES2UG23CS380

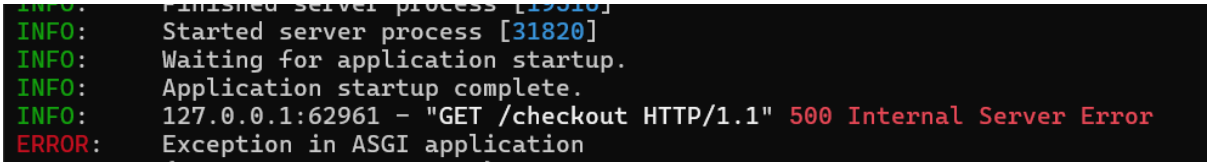
SECTION: F

SS1 (Events page loaded)

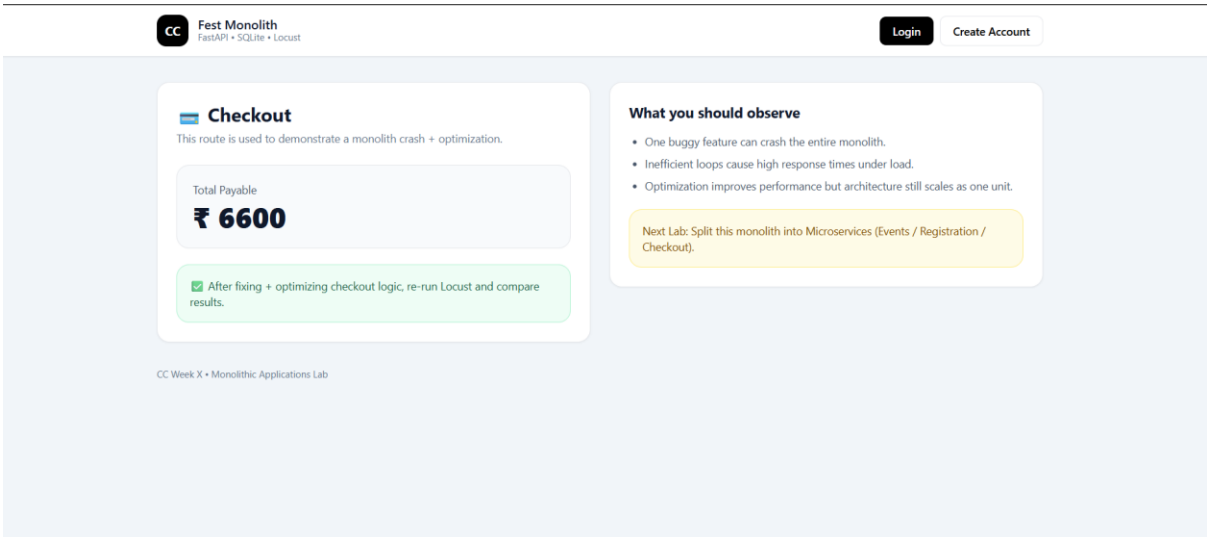


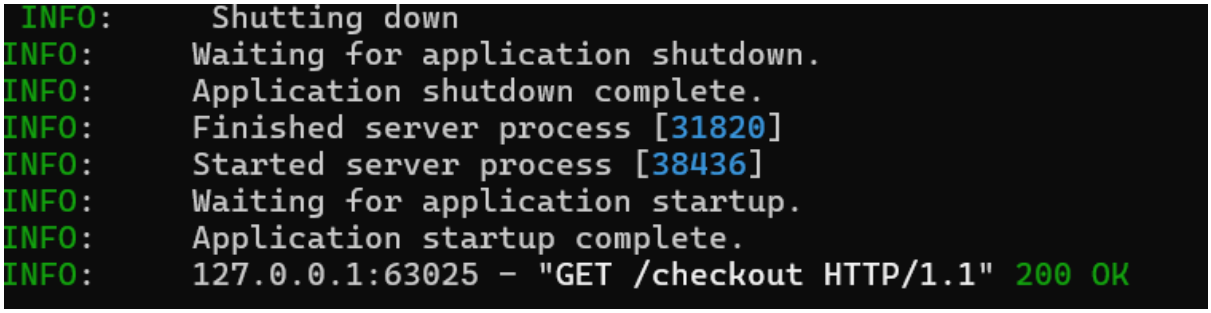
SS2 (crash)



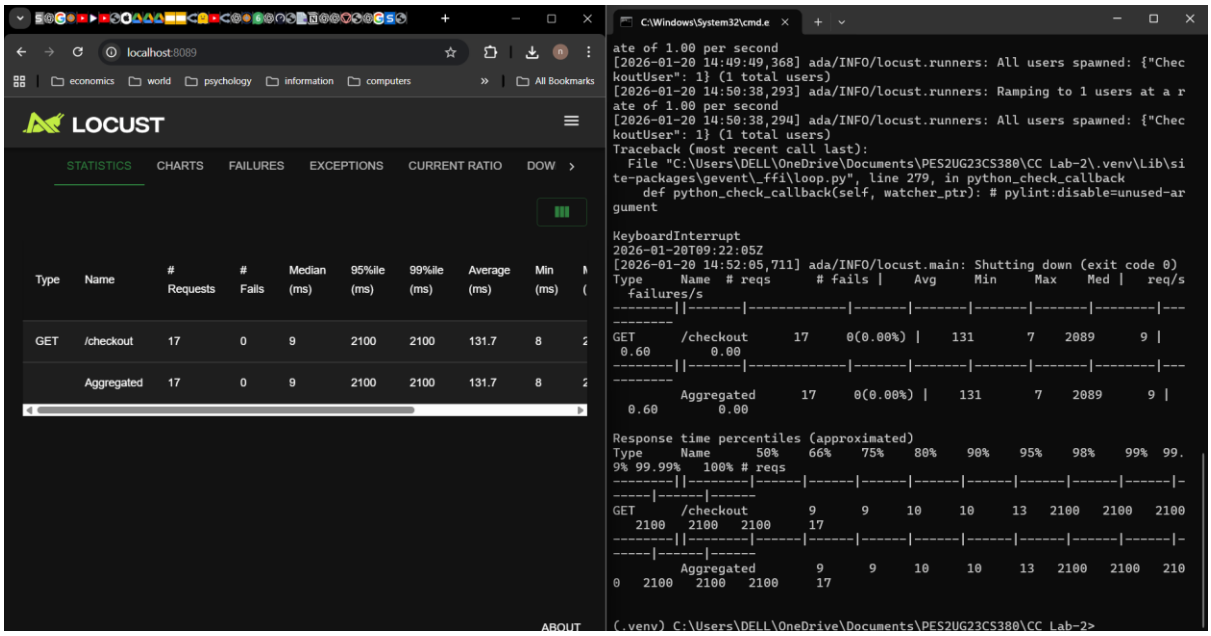


SS3

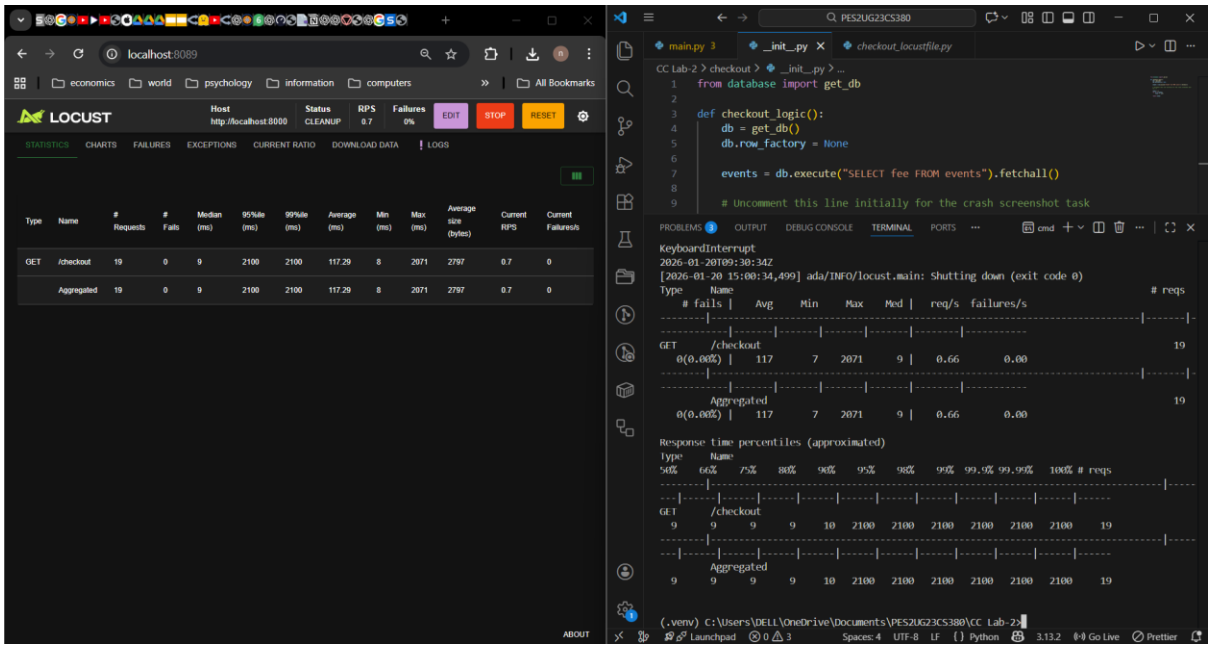

Fest Monolith
FastAPI • SQLite • Locust
Login
Create Account
Checkout
This route is used to demonstrate a monolith crash + optimization.
Total Payable
₹ 6600
After fixing + optimizing checkout logic, re-run Locust and compare results.
What you should observe
One buggy feature can crash the entire monolith.
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: Shutting down
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [31820]
INFO: Started server process [38436]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:63025 - "GET /checkout HTTP/1.1" 200 OK

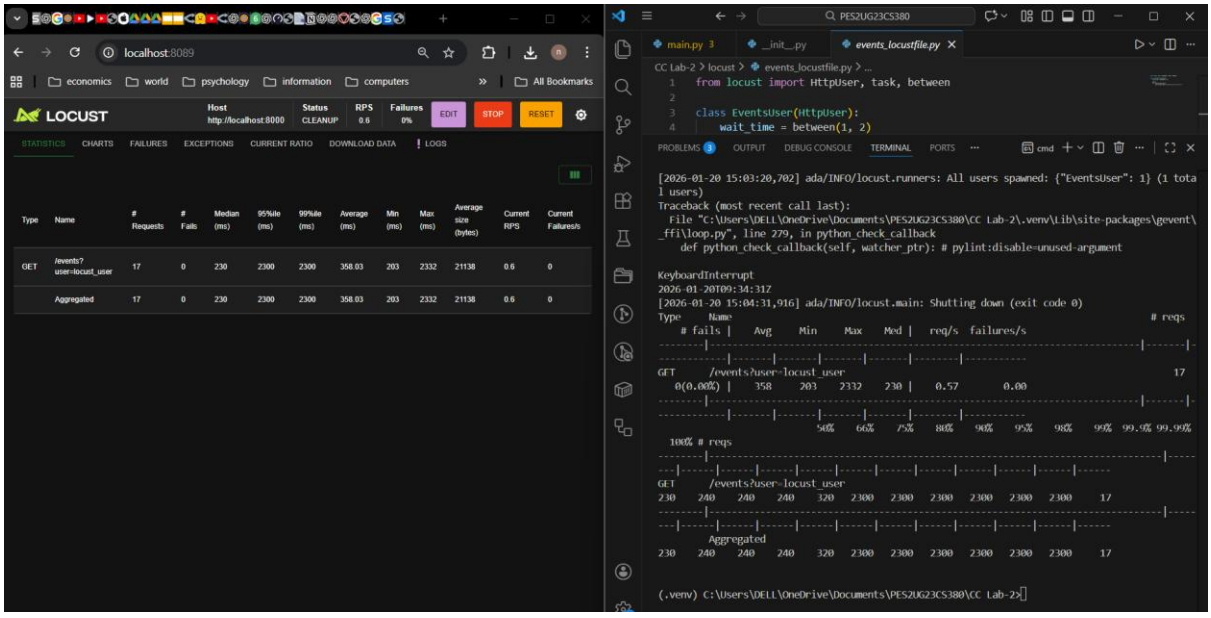
SS4


localhost:8089
LOCUST
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms) Min (ms)
GET /checkout 17 0 9 2100 2100 131.7 8
Aggregated 17 0 9 2100 2100 131.7 8
ABOUT
C:\Windows\System32\cmd.e
ate of 1.00 per second
[2026-01-20 14:49:49,368] ada/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
[2026-01-20 14:50:38,293] ada/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 14:50:38,294] ada/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2\.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
2026-01-20T09:22:05Z
[2026-01-20 14:52:05,711] ada/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs # fails | Avg Min Max Med | req/s failures/s
GET /checkout 17 0(0.00%) | 131 7 2089 9 | 0.60 0.00
Aggregated 17 0(0.00%) | 131 7 2089 9 | 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 /checkout 9 9 10 10 13 2100 2100 2100 2100 2100 2100 17
Aggregated 9 9 10 10 13 2100 2100 2100 2100 2100 2100 17
(.venv) C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2>

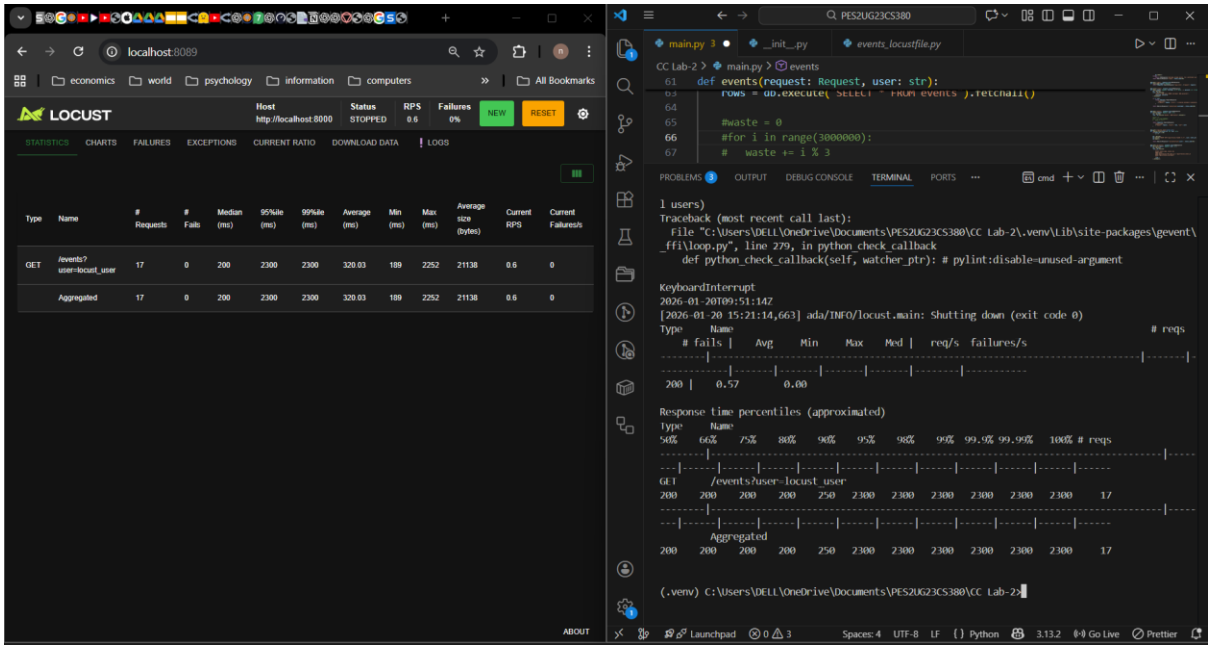
SS5


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) 99%ile (ms) Average (ms) Min (ms) Max (ms) Average size (bytes) Current RPS Current Failures/s
GET /checkout 19 0 9 2100 2100 117.29 8 2071 2797 0.7 0
Aggregated 19 0 9 2100 2100 117.29 8 2071 2797 0.7 0
ABOUT
PES2UG23CS380
main.py 3
__init__.py
checkout_locustfile.py
CC Lab-2 > checkout > __init__.py > ...
from database import get_db
def checkout_logic():
db = get_db()
db.row_factory = None
events = db.execute("SELECT fee FROM events").fetchall()
# Uncomment this line initially for the crash screenshot task
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
KeyboardInterrupt
2026-01-20T09:30:34Z
[2026-01-20 15:00:34,499] ada/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 7 2071 9 | 0.66 0.00
Aggregated 19
0(0.00%) | 117 7 2071 9 | 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
9 9 9 9 10 2100 2100 2100 2100 2100 2100 19
Aggregated
9 9 9 9 10 2100 2100 2100 2100 2100 2100 19
(.venv) C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2>

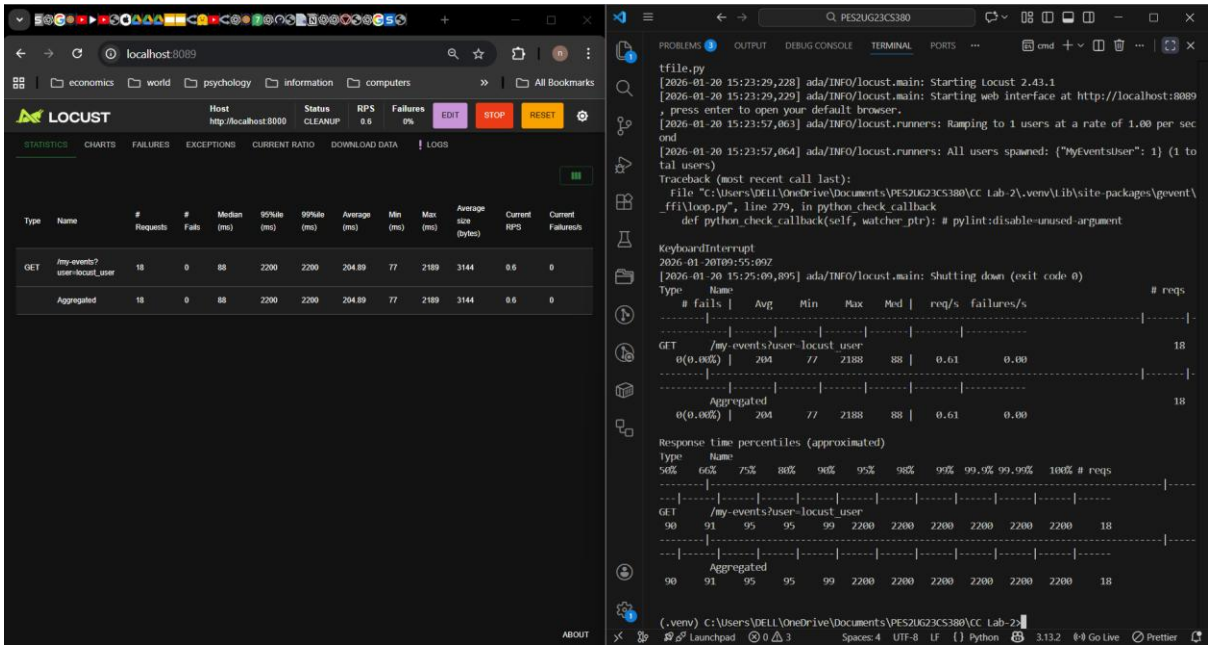
SS6


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) 99%ile (ms) Average (ms) Min (ms) Max (ms) Average size (bytes) Current RPS Current Failures/s
GET /events?user=locust_user 17 0 230 2300 2300 358.03 203 2332 21138 0.6 0
Aggregated 17 0 230 2300 2300 358.03 203 2332 21138 0.6 0
PES2UG23CS380
main.py 3
__init__.py
events_locustfile.py
CC Lab-2 > locust > events_locustfile.py > ...
from locust import HttpUser, task, between
class EventsUser(HttpUser):
wait_time = between(1, 2)
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
[2026-01-20 15:03:20,702] ada/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2\.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
2026-01-20T09:34:31Z
[2026-01-20 15:04:31,916] ada/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 17
0(0.00%) | 358 203 2332 230 | 0.57 0.00
50% 66% 75% 80% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
GET /events?user=locust_user
230 240 240 240 320 2300 2300 2300 2300 2300 2300 17
Aggregated
230 240 240 240 320 2300 2300 2300 2300 2300 2300 17
(.venv) C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2>

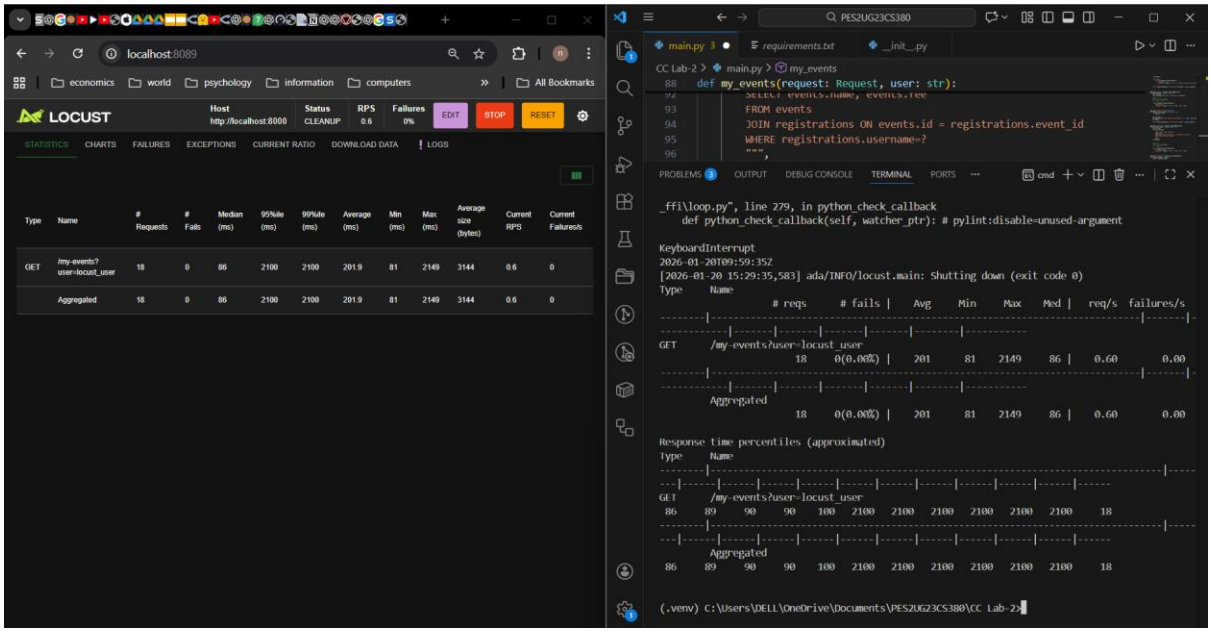
SS7


localhost:8089
LOCUST
Host
http://localhost:8000
Status
STOPPED
RPS
0.6
Failures
0%
NEW
RESET
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
/events?user=locust_user
17
0
200
2300
2300
320.03
189
2252
21138
0.6
0
Aggregated
17
0
200
2300
2300
320.03
189
2252
21138
0.6
0
ABOUT
PES2UG23CS380
main.py
__init__.py
events_locustfile.py
def events(request: Request, user: str):
#waste = 0
#for i in range(3000000):
waste += i % 3
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
1 users)
Traceback (most recent call last):
File "C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2\.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
2026-01-20T09:51:14Z
[2026-01-20 15:21:14,663] ada/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs
fails | Avg Min Max Med | req/s failures/s
200 | 0.57 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
200 200 200 200 250 2300 2300 2300 2300 2300 2300 17
Aggregated
200 200 200 200 250 2300 2300 2300 2300 2300 2300 17
(.venv) C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2>
Launchpad
Spaces: 4
UTF-8
LF
Python
3.13.2
Go Live
Prettier

SS8


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)
99%ile (ms)
Average (ms)
Min (ms)
Max (ms)
Average size (bytes)
Current RPS
Current Failures/s
GET
/my-events?user=locust_user
18
0
88
2200
2200
204.89
77
2189
3144
0.6
0
Aggregated
18
0
88
2200
2200
204.89
77
2189
3144
0.6
0
ABOUT
PES2UG23CS380
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
tfile.py
[2026-01-20 15:23:29,228] ada/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:23:29,229] ada/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-20 15:23:57,063] ada/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:23:57,064] ada/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2\.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
2026-01-20T09:55:09Z
[2026-01-20 15:25:09,895] ada/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 18
0(0.00%) | 204 77 2188 88 | 0.61 0.00
Aggregated 18
0(0.00%) | 204 77 2188 88 | 0.61 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
90 91 95 95 99 2200 2200 2200 2200 2200 2200 18
Aggregated
90 91 95 95 99 2200 2200 2200 2200 2200 2200 18
(.venv) C:\Users\DELL\OneDrive\Documents\PES2UG23CS380\CC Lab-2>
Launchpad
Spaces: 4
UTF-8
LF
Python
3.13.2
Go Live
Prettier

SS9



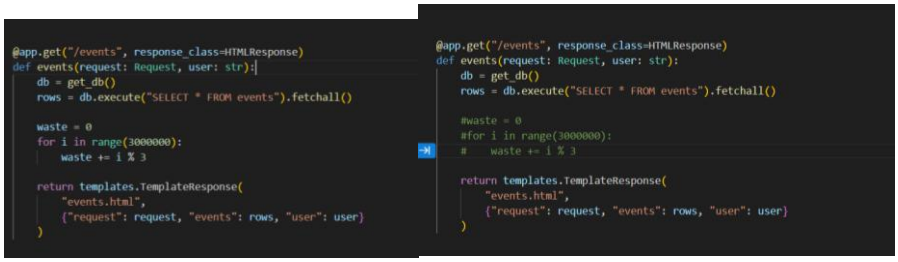
Route: /events

What was the bottleneck?

The route contained a loop that executed on every request, blocking the FastAPI event loop and preventing the server from handling other users concurrently.

What change did you make?

I commented out the lines which contain the unnecessary loop.



Why did the performance improve?

Removing blocking lines freed the event loop, allowing FastAPI to handle multiple requests concurrently and significantly reducing response time.

Route: /my-events

What was the bottleneck?

The route included a large blocking loop, causing slow query execution and request blocking.

What change did you make?

I commented the lines

```
99
100
101     dummy = 0
102     for _ in range(1500000):
103         dummy += 1
104
105     return templates.TemplateResponse(
106         "my_events.html",
107         {"request": request, "events": rows, "user": user}
108     )
```

```
    #dummy = 0
    #for _ in range(1500000):
    #    dummy += 1

    return templates.TemplateResponse(
        "my_events.html",
        {"request": request, "events": rows, "user": user}
    )
```

Why did the performance improve?

eliminating blocking code allowed faster query execution and improved concurrency.