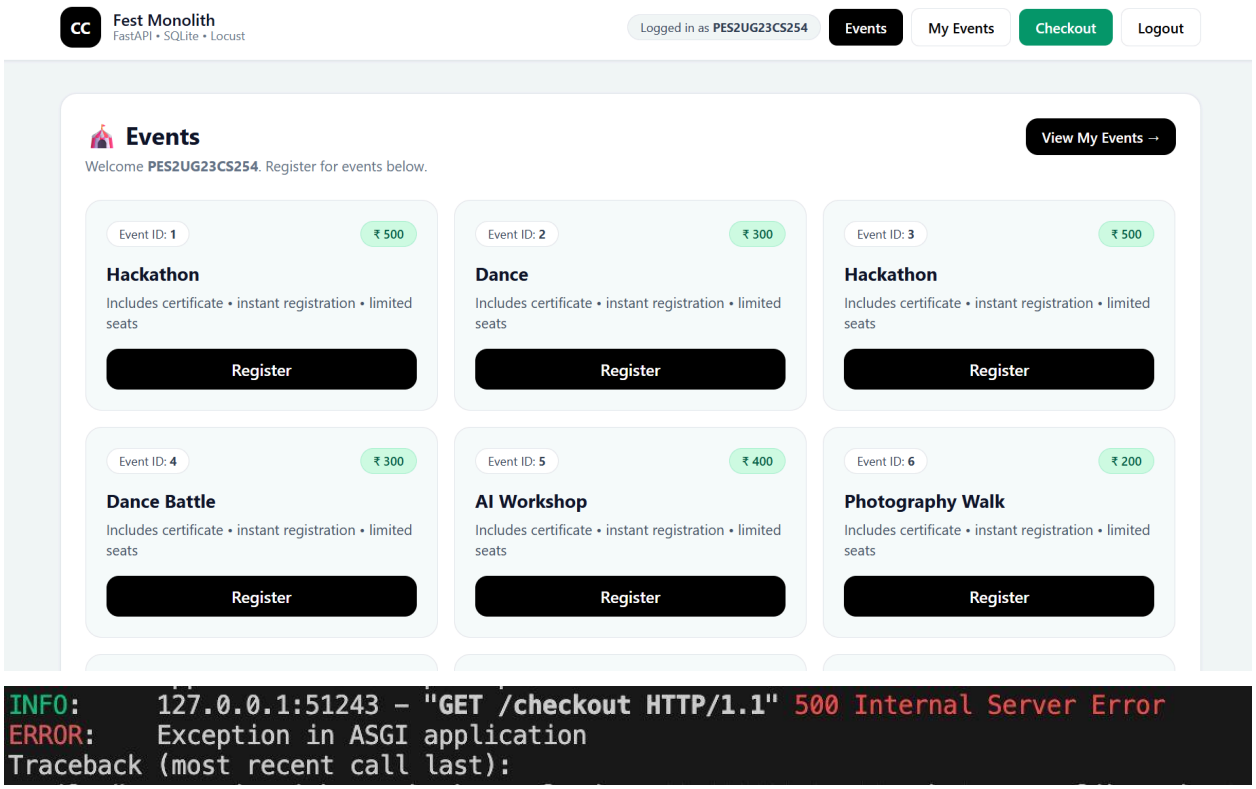
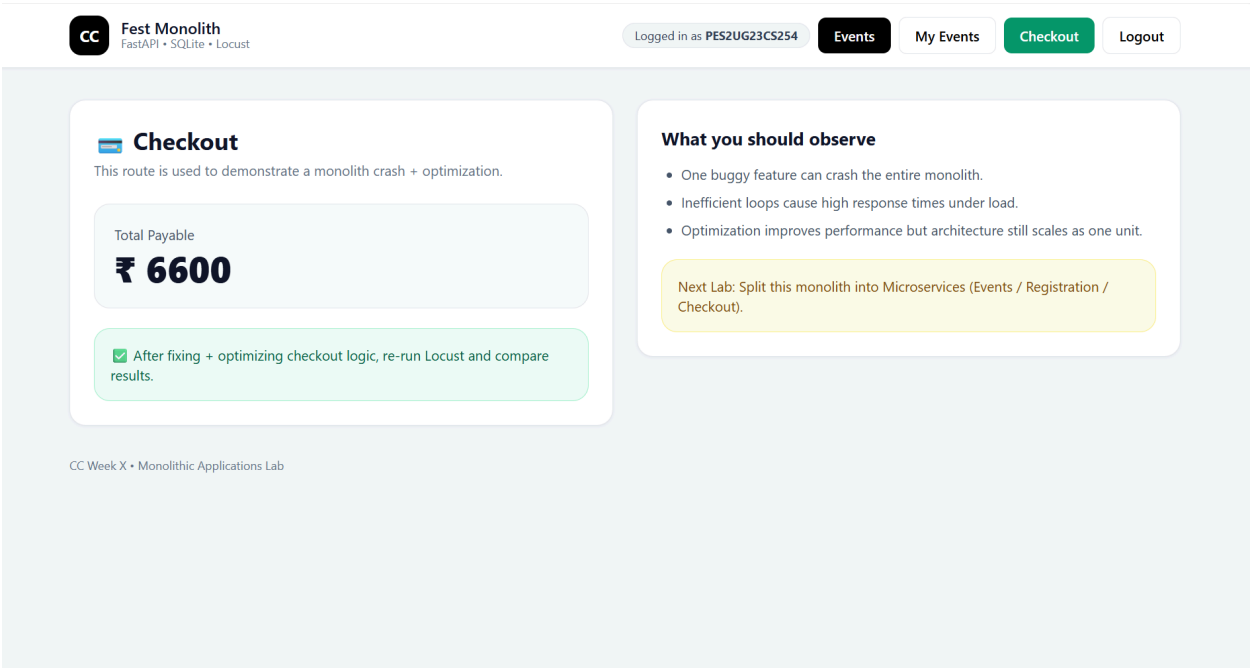
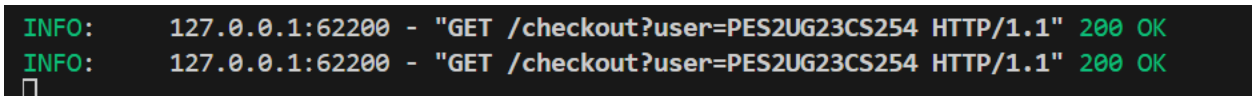
NAME : PADMAA.K.B
SRN : PES2UG23CS254
SEC : D

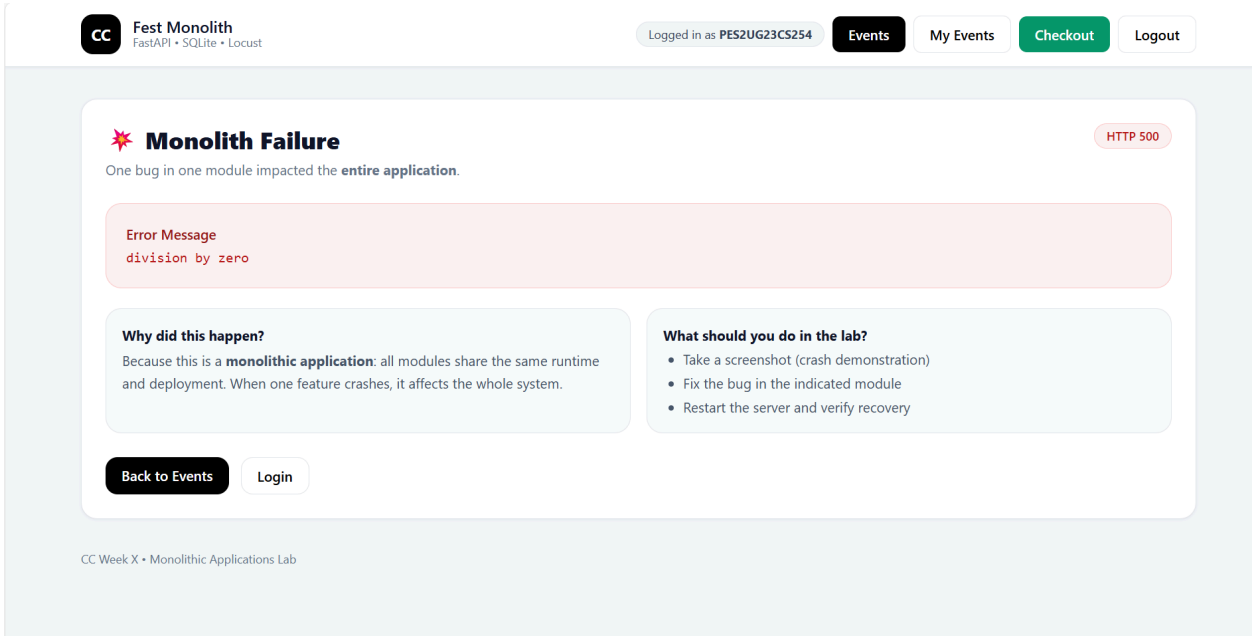
Ss1 :



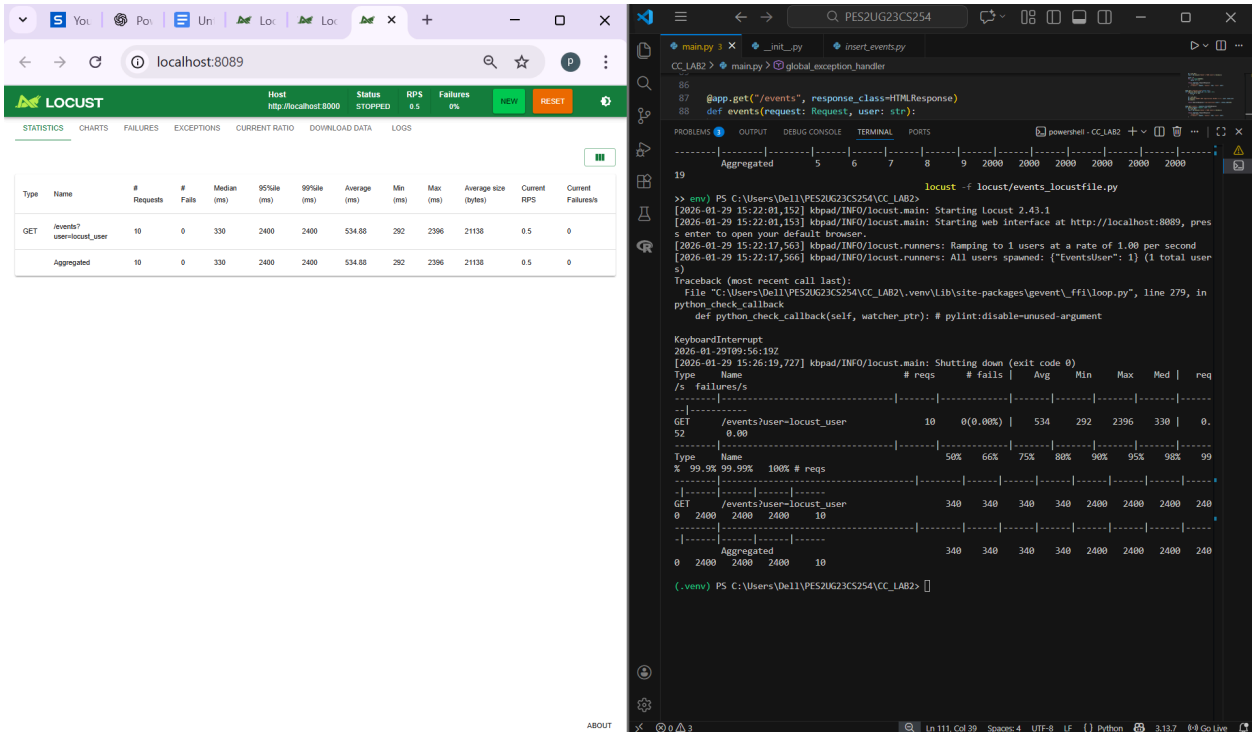
Ss2 :


CC
Fest Monolith
FastAPI • SQLite • Locust
Logged in as PES2UG23CS254
Events
My Events
Checkout
Logout
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

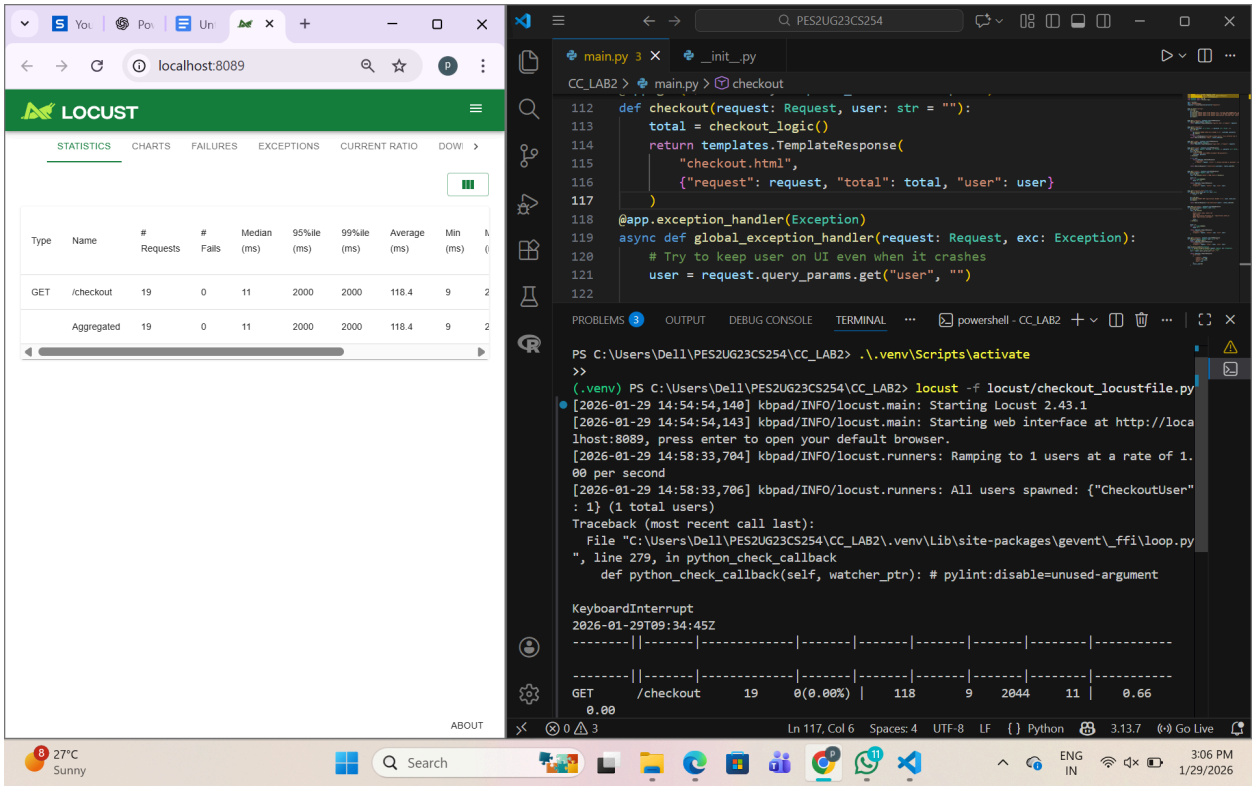

INFO: 127.0.0.1:62200 - "GET /checkout?user=PES2UG23CS254 HTTP/1.1" 200 OK
INFO: 127.0.0.1:62200 - "GET /checkout?user=PES2UG23CS254 HTTP/1.1" 200 OK


CC
Fest Monolith
FastAPI • SQLite • Locust
Logged in as PES2UG23CS254
Events
My Events
Checkout
Logout
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
CC Week X • Monolithic Applications Lab

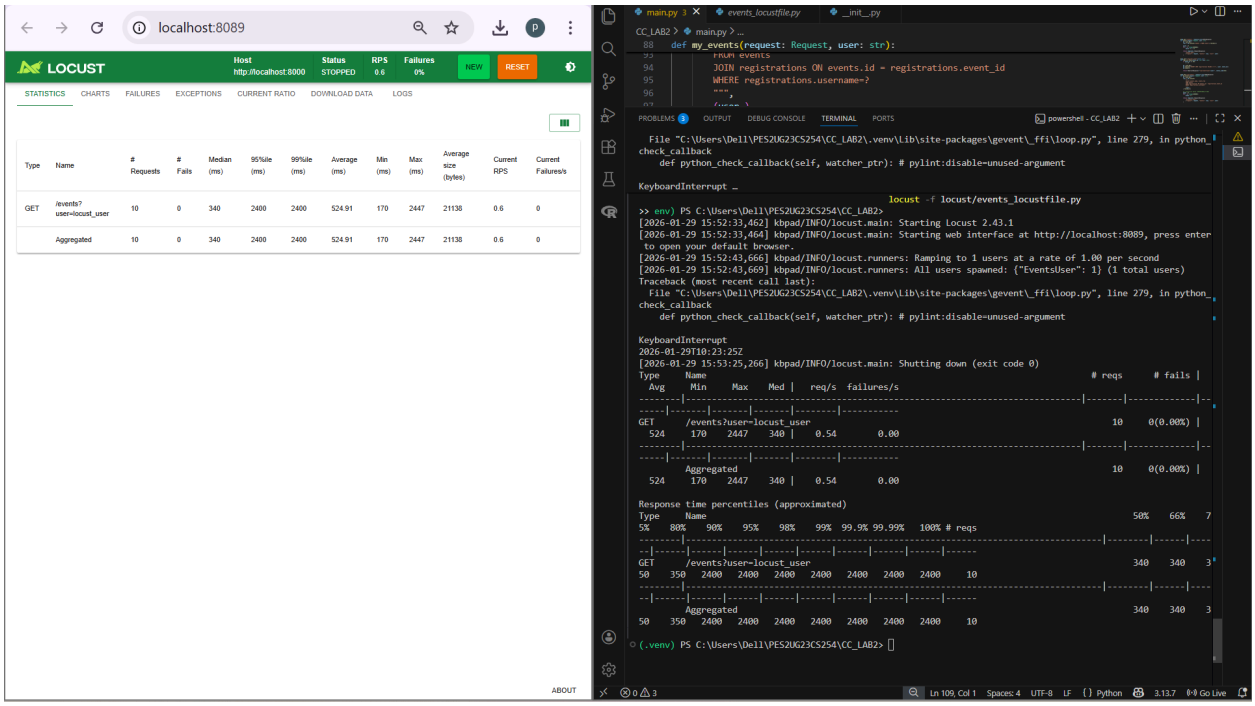
Ss4 :


localhost:8089
LOCUST
Host
http://localhost:8000
Status
STOPPED
RPS
0.5
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
10
0
330
2400
2400
534.88
292
2396
21138
0.5
0
Aggregated
10
0
330
2400
2400
534.88
292
2396
21138
0.5
0
ABOUT
PES2UG23CS254
main.py
__init__.py
insert_events.py
CC_LAB2 > main.py > global_exception_handler
@app.get("/events", response_class=HTMLResponse)
def events(request: Request, user: str):
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell - CC_LAB2
Aggregated 5 6 7 8 9 2000 2000 2000 2000 2000 2000
19
locust -f locust/events_locustfile.py
>> env) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2>
[2026-01-29 15:22:01,152] kbpad/INFO/locust.main: Starting Locust 2.43.1
[2026-01-29 15:22:01,153] kbpad/INFO/locust.main: Starting web interface at http://localhost:8089, pres
s enter to open your default browser.
[2026-01-29 15:22:17,563] kbpad/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 15:22:17,566] kbpad/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total user
s)
Traceback (most recent call last):
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
2026-01-29T09:56:19Z
[2026-01-29 15:26:19,727] kbpad/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 10 0(0.00%) | 534 292 2396 330 | 0.
52 0.00
Type Name 50% 66% 75% 80% 90% 95% 98% 99
% 99.9% 99.99% 100% # reqs
GET /events?user=locust_user 340 340 340 340 2400 2400 2400 240
0 2400 2400 2400 10
Aggregated 340 340 340 340 2400 2400 2400 240
0 2400 2400 2400 10
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2>
Ln 111, Col 39
Spaces: 4
UTF-8
LF
Python
3.13.7
Go Live

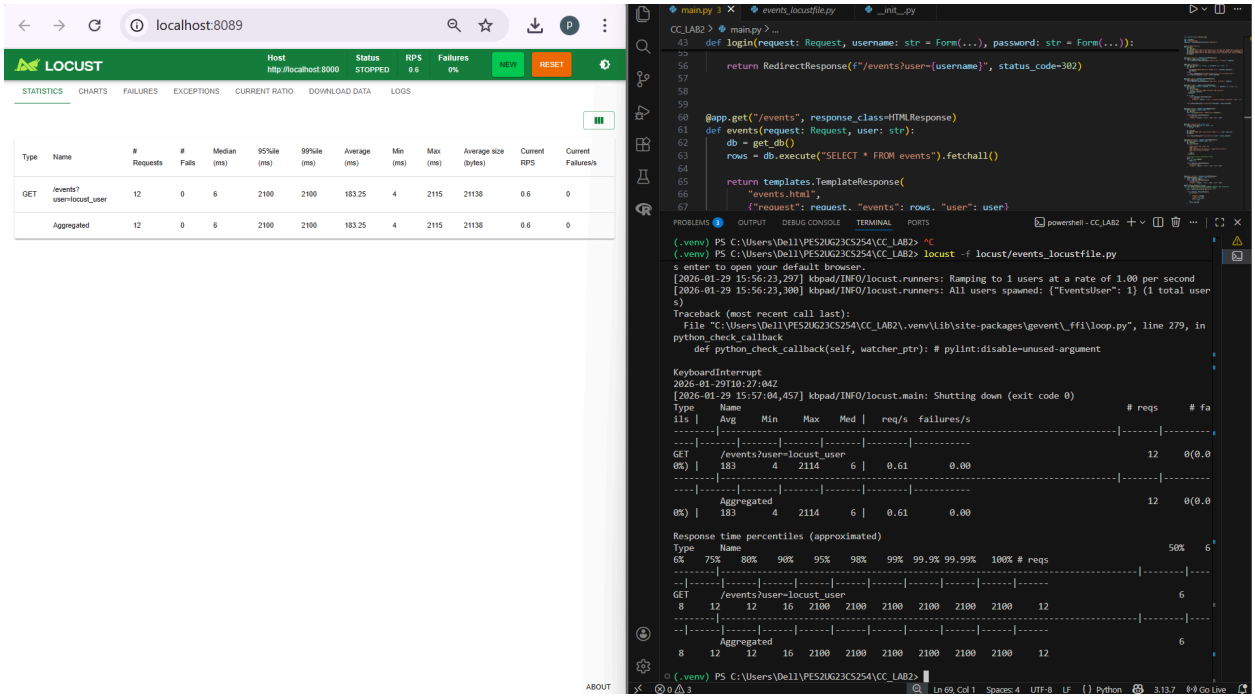
Ss5 :


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
19
0
11
2000
2000
118.4
9
Aggregated
19
0
11
2000
2000
118.4
9
ABOUT
PES2UG23CS254
main.py 3
__init__.py
CC_LAB2 > main.py > checkout
112 def checkout(request: Request, user: str = ""):
113 total = checkout_logic()
114 return templates.TemplateResponse(
115 "checkout.html",
116 {"request": request, "total": total, "user": user}
117)
118 @app.exception_handler(Exception)
119 async def global_exception_handler(request: Request, exc: Exception):
120 # Try to keep user on UI even when it crashes
121 user = request.query_params.get("user", "")
122
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
powershell - CC_LAB2
PS C:\Users\Dell\PES2UG23CS254\CC_LAB2> .\.venv\Scripts\activate
>>
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2> locust -f locust/checkout_locustfile.py
[2026-01-29 14:54:54,140] kbpad/INFO/locust.main: Starting Locust 2.43.1
[2026-01-29 14:54:54,143] kbpad/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-29 14:58:33,704] kbpad/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 14:58:33,706] kbpad/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
2026-01-29T09:34:45Z
GET /checkout 19 0(0.00%) | 118 9 2044 11 | 0.66 0.00
Ln 117, Col 6
Spaces: 4
UTF-8
LF
Python
3.13.7
Go Live
27°C
Sunny
Search
ENG
IN
3:06 PM
1/29/2026

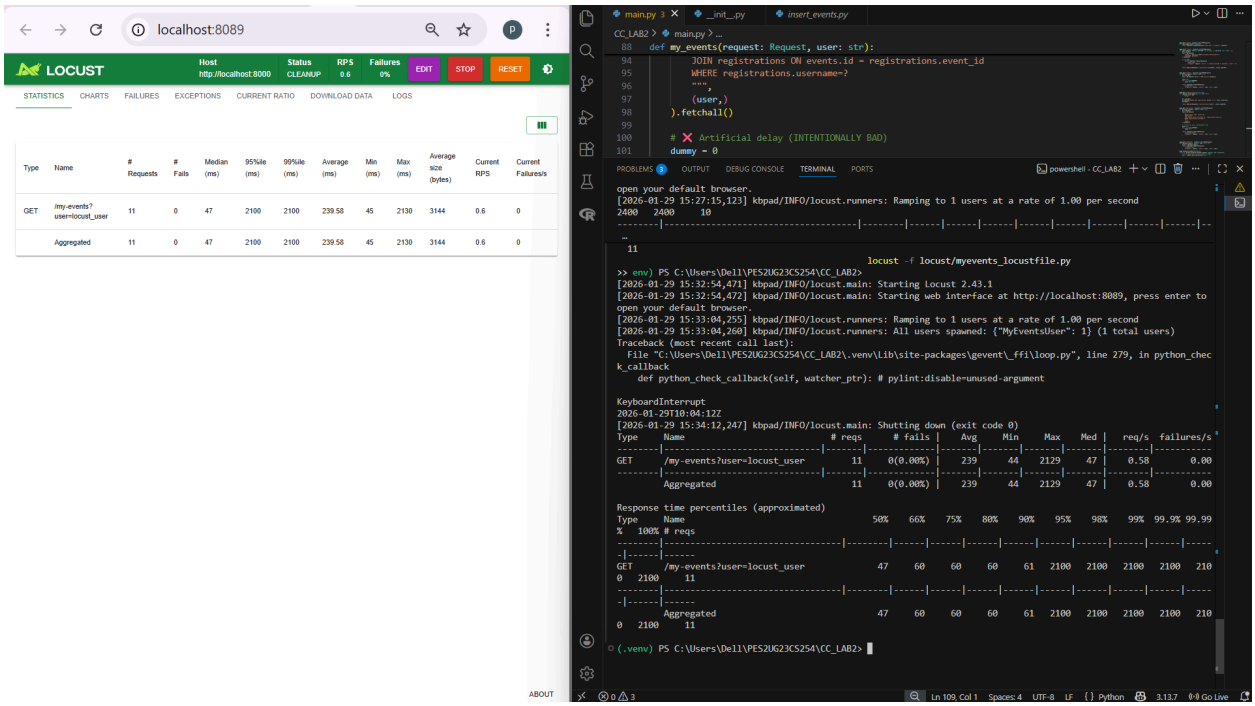
Ss6 :


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
10
0
340
2400
2400
524.91
170
2447
21138
0.6
0
Aggregated
10
0
340
2400
2400
524.91
170
2447
21138
0.6
0
ABOUT
main.py 3
events_locustfile.py
__init__.py
CC_LAB2 > main.py > ...
88 def my_events(request: Request, user: str):
94 JOIN registrations ON events.id = registrations.event_id
95 WHERE registrations.username=?
96 """,
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell - CC_LAB2
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2> locust -f locust/events_locustfile.py
[2026-01-29 15:52:33,462] kbpad/INFO/locust.main: Starting Locust 2.43.1
[2026-01-29 15:52:33,464] kbpad/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-29 15:52:43,666] kbpad/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 15:52:43,669] kbpad/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
2026-01-29T10:23:25Z
[2026-01-29 15:53:25,266] kbpad/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 10 0(0.00%) | 524 170 2447 340 | 0.54 0.00
Aggregated 10 0(0.00%) | 524 170 2447 340 | 0.54 0.00
Response time percentiles (approximated)
Type Name 50% 66% 7 5% 80% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
GET /events?user=locust_user 340 340 3 50 350 2400 2400 2400 2400 2400 2400 2400 10
Aggregated 340 340 3 50 350 2400 2400 2400 2400 2400 2400 2400 10
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2>
Ln 109, Col 1
Spaces: 4
UTF-8
LF
Python
3.13.7
Go Live

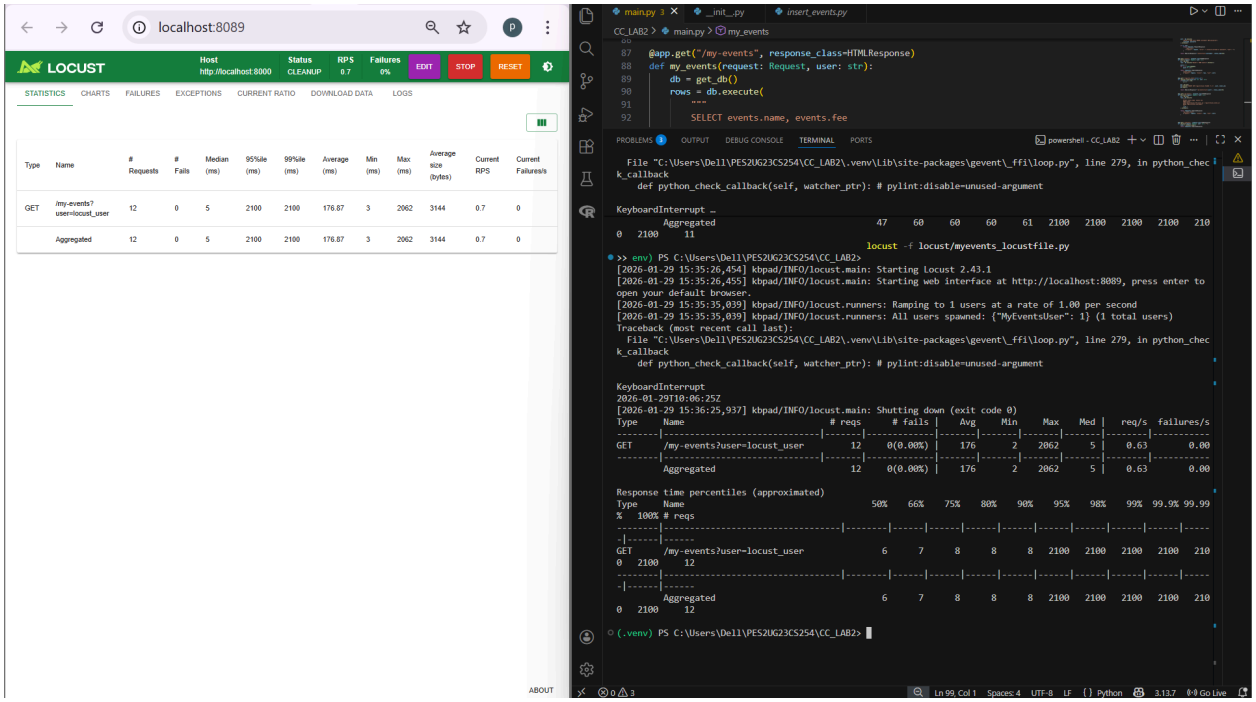
Ss7 :


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 12 0 6 2100 2100 183.25 4 2115 21138 0.6 0
Aggregated 12 0 6 2100 2100 183.25 4 2115 21138 0.6 0
def login(request: Request, username: str = Form(...), password: str = Form(...)):
return RedirectResponse(f"/events?user={username}", status_code=302)
@app.get("/events", response_class=HTMLResponse)
def events(request: Request, user: str):
db = get_db()
rows = db.execute("SELECT * FROM events").fetchall()
return templates.TemplateResponse(
"events.html",
{"request": request, "events": rows, "user": user}
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2> locust -f locust/events_locustfile.py
[2026-01-29 15:56:23,297] kbpad/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 15:56:23,300] kbpad/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
2026-01-29T10:27:04Z
[2026-01-29 15:57:04,457] kbpad/INFO/locust.main: Shutting down (exit code 0)
Response time percentiles (approximated)
GET /events?user=locust_user 183 4 2114 6 | 0.61 0.00
Aggregated 183 4 2114 6 | 0.61 0.00

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 11 0 47 2100 2100 239.58 45 2130 3144 0.6 0
Aggregated 11 0 47 2100 2100 239.58 45 2130 3144 0.6 0
def my_events(request: Request, user: str):
JOIN registrations ON events.id = registrations.event_id
WHERE registrations.username=?
""",
(user,)
).fetchall()
# ❌ Artificial delay (INTENTIONALLY BAD)
dummy = 0
(.venv) PS C:\Users\Dell\PES2UG23CS254\CC_LAB2> locust -f locust/myevents_locustfile.py
[2026-01-29 15:32:54,471] kbpad/INFO/locust.main: Starting Locust 2.43.1
[2026-01-29 15:32:54,472] kbpad/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-29 15:33:04,255] kbpad/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 15:33:04,260] kbpad/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
Traceback (most recent call last):
File "C:\Users\Dell\PES2UG23CS254\CC_LAB2\.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
2026-01-29T10:04:12Z
[2026-01-29 15:34:12,247] kbpad/INFO/locust.main: Shutting down (exit code 0)
Response time percentiles (approximated)
GET /my-events?user=locust_user 47 60 60 60 61 2100 2100 2100 2100 2100
Aggregated 47 60 60 60 61 2100 2100 2100 2100 2100

Ss9 :



ANSWERS :

Route: /events

What was the bottleneck?

The `/events` route contained an artificial delay implemented using a loop that executed three million iterations. This loop did not contribute to the application logic and unnecessarily increased response time.

What change did you make?

The redundant loop was removed, allowing the route to directly fetch event data from the database and return the response.

Why did the performance improve?

Removing the unnecessary computation reduced CPU usage and execution time, resulting in a faster response and lower average latency as observed in Locust results.

Route: /my-events

What was the bottleneck?

The `/my-events` route had an artificial delay caused by a dummy loop that executed 1.5 million iterations, which significantly increased response time without affecting functionality.

What change did you make?

The dummy loop was removed so that the route only performs the required database query and renders the response.

Why did the performance improve?

Eliminating the redundant loop reduced processing overhead, leading to quicker execution and improved response times under load, as verified using Locust.