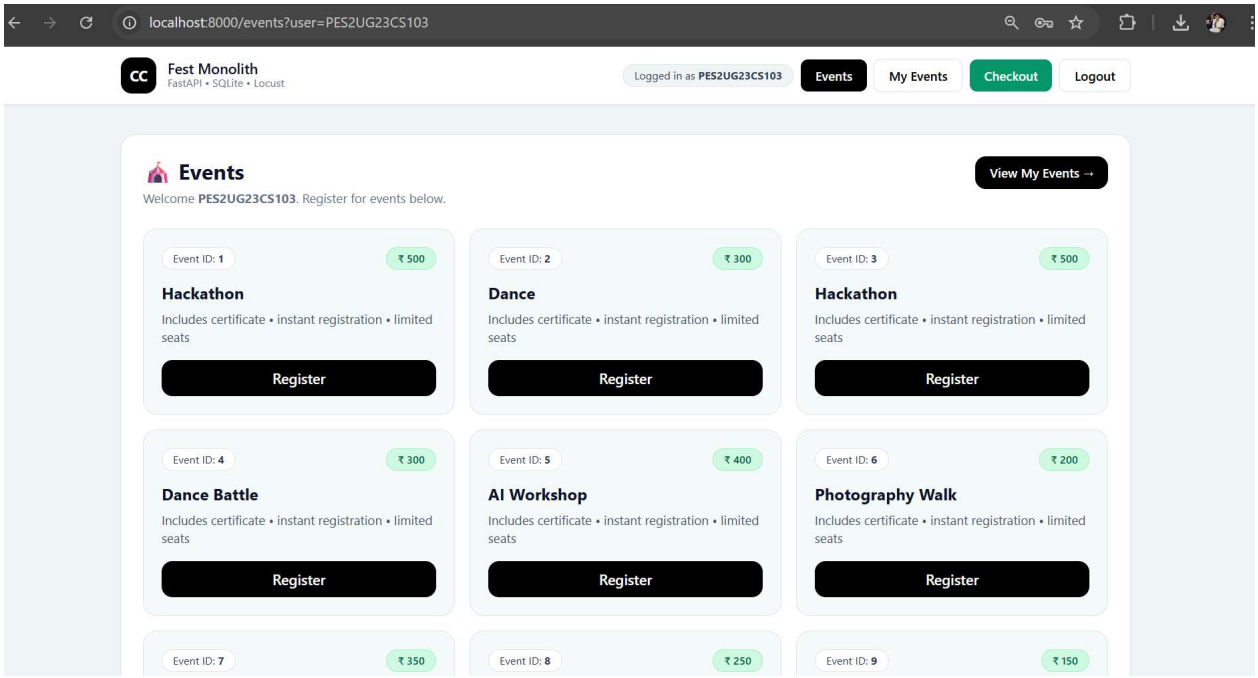
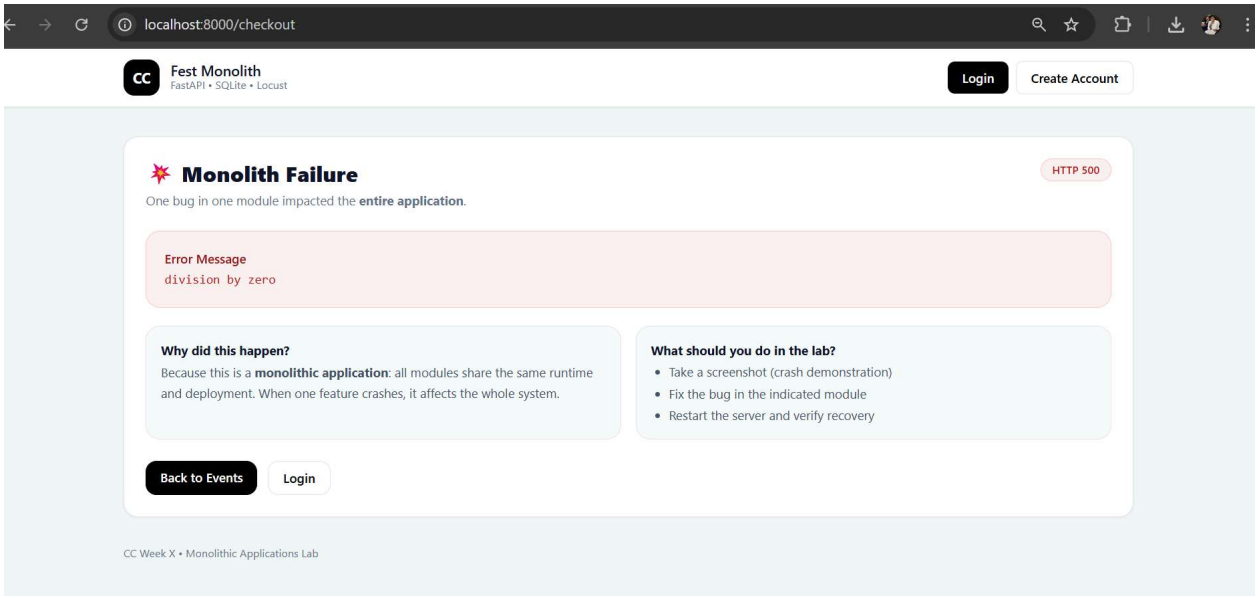
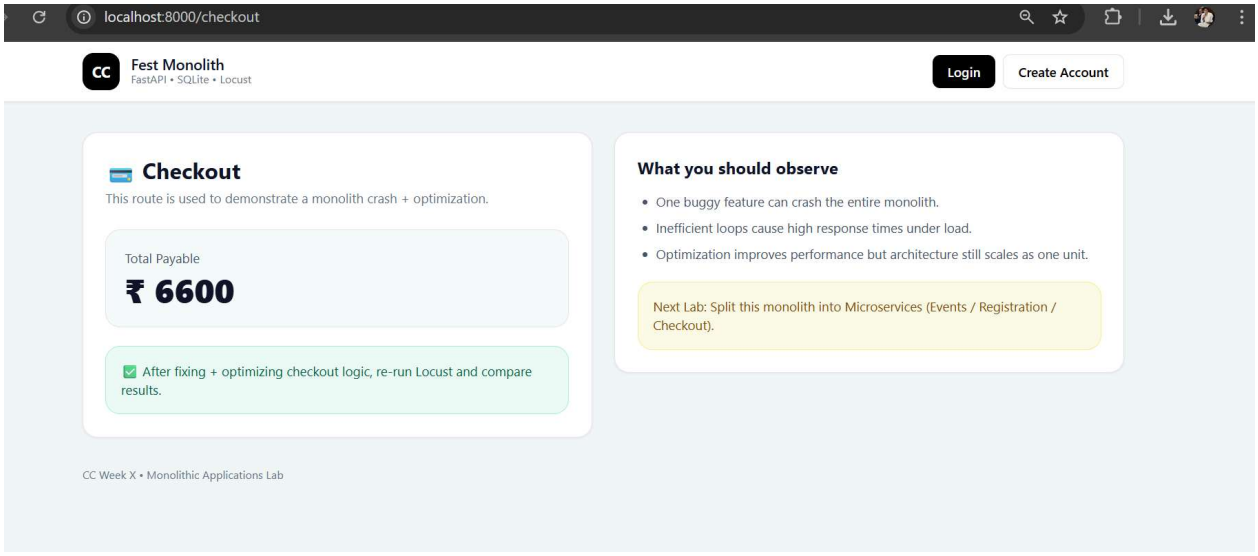
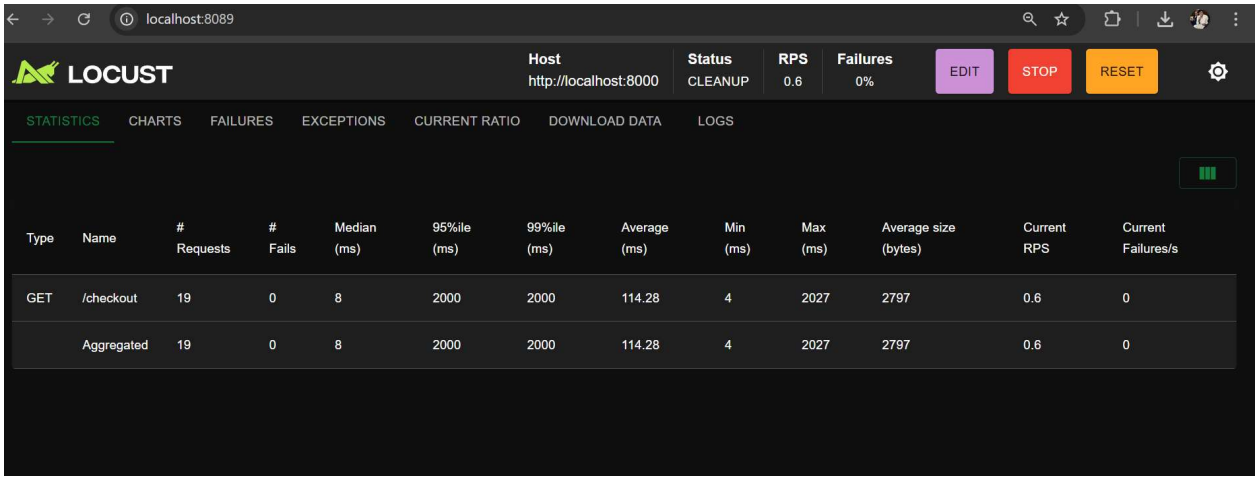

localhost:8000/events?user=PES2UG23CS103
Fest Monolith
FastAPI • SQLite • Locust
Logged in as PES2UG23CS103
Events
My Events
Checkout
Logout
Events
View My Events →
Welcome PES2UG23CS103. Register for events below.
Event ID: 1
₹ 500
Hackathon
Includes certificate • instant registration • limited seats
Register
Event ID: 2
₹ 300
Dance
Includes certificate • instant registration • limited seats
Register
Event ID: 3
₹ 500
Hackathon
Includes certificate • instant registration • limited seats
Register
Event ID: 4
₹ 300
Dance Battle
Includes certificate • instant registration • limited seats
Register
Event ID: 5
₹ 400
AI Workshop
Includes certificate • instant registration • limited seats
Register
Event ID: 6
₹ 200
Photography Walk
Includes certificate • instant registration • limited seats
Register
Event ID: 7
₹ 350
Event ID: 8
₹ 250
Event ID: 9
₹ 150

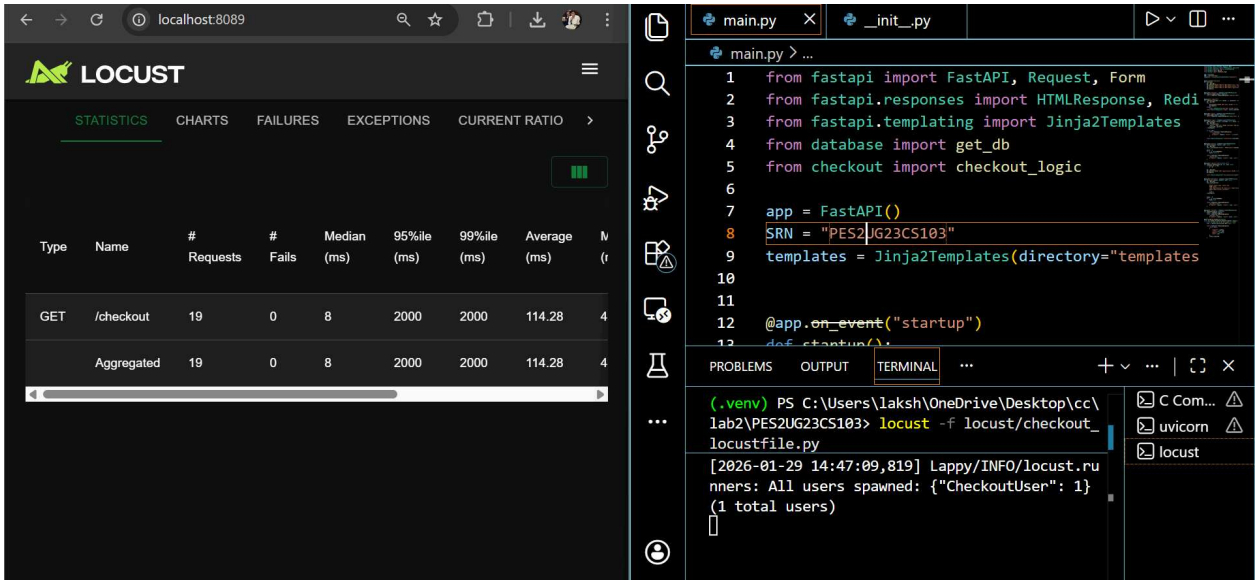

localhost:8000/checkout
Fest Monolith
FastAPI • SQLite • Locust
Login
Create Account
Monolith Failure
HTTP 500
One bug in one module impacted the entire application.
Error Message
division by zero
Why did this happen?
Because this is a monolithic application: all modules share the same runtime and deployment. 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

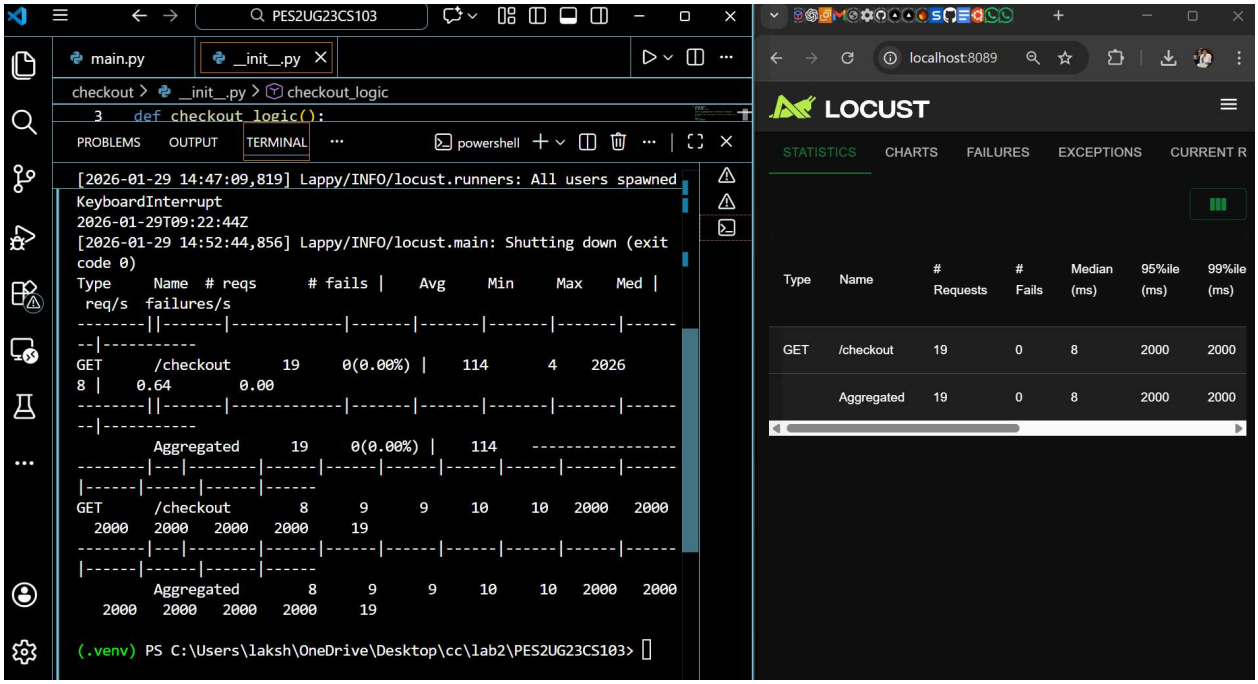

INFO: 127.0.0.1:55907 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application

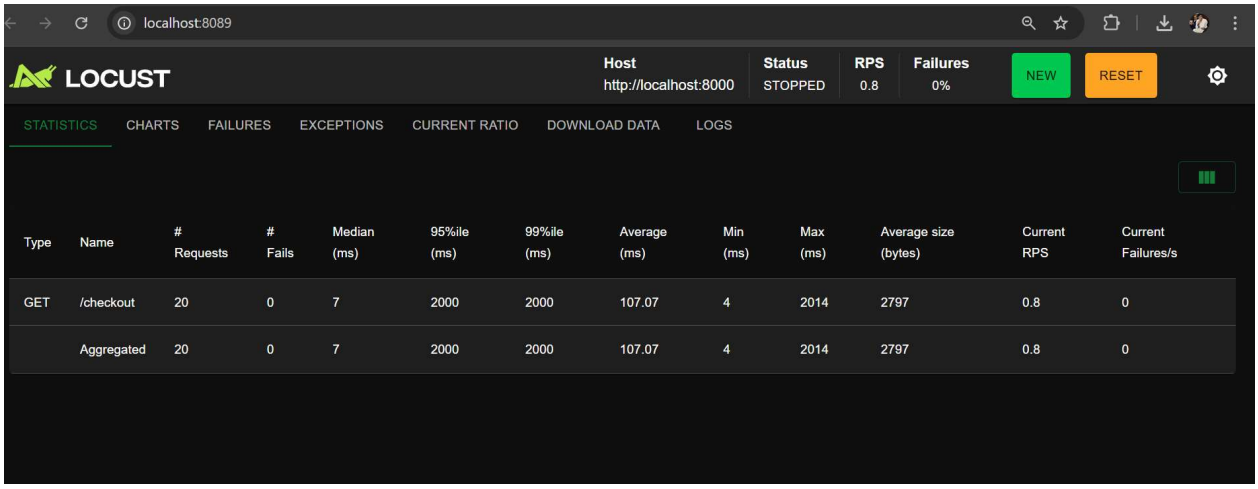

localhost:8000/checkout
Fest Monolith
FastAPI • SQLite • Locust
Login
Create Account
Checkout
This route is used to demonstrate a monolith crash + optimization.
Total Payable
₹ 6600
After fixing + optimizing checkout logic, re-run Locust and compare results.
What you should observe
One buggy feature 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

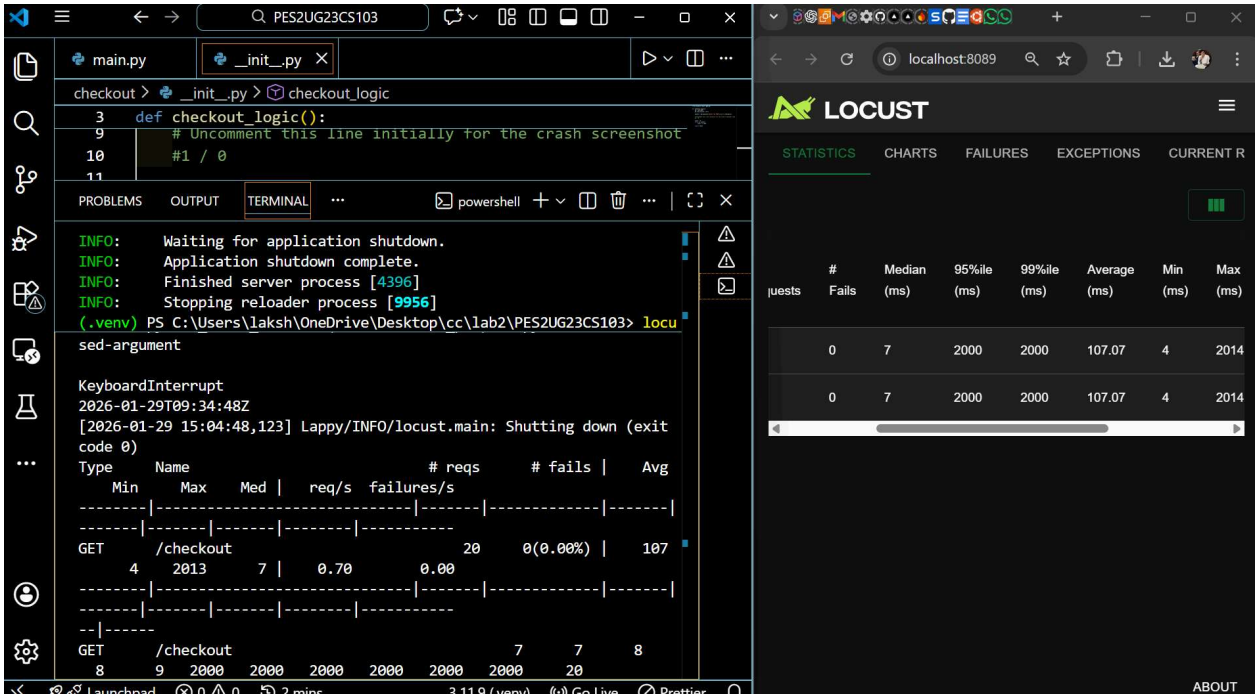

INFO: 127.0.0.1:57379 - "GET /checkout HTTP/1.1" 200 OK

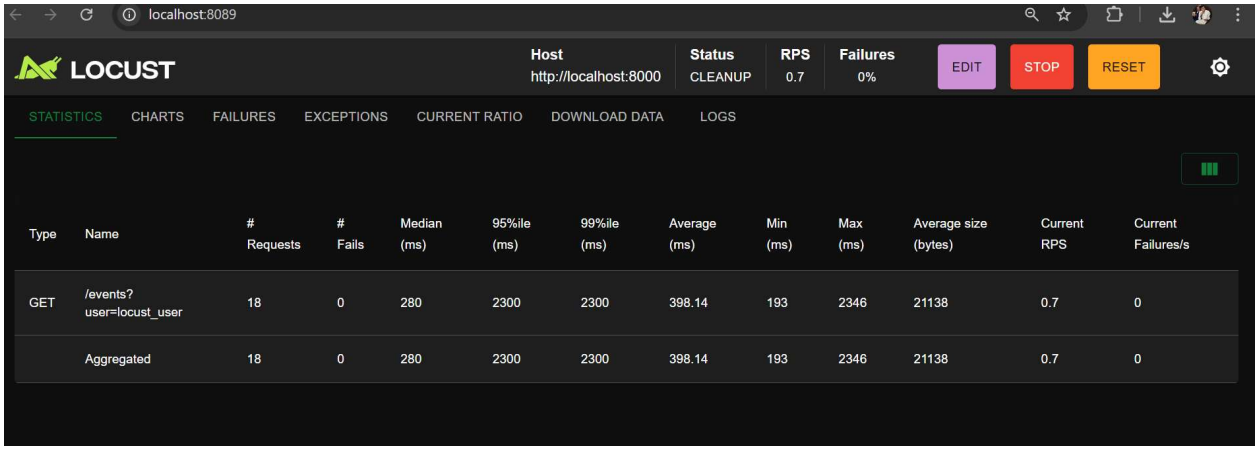

localhost:8089
LOCUST
Host
http://localhost:8000
Status
CLEANUP
RPS
0.6
Failures
0%
EDIT
STOP
RESET
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
DOWNLOAD DATA
LOGS
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms) Min (ms) Max (ms) Average size (bytes) Current RPS Current Failures/s
GET /checkout 19 0 8 2000 2000 114.28 4 2027 2797 0.6 0
Aggregated 19 0 8 2000 2000 114.28 4 2027 2797 0.6 0

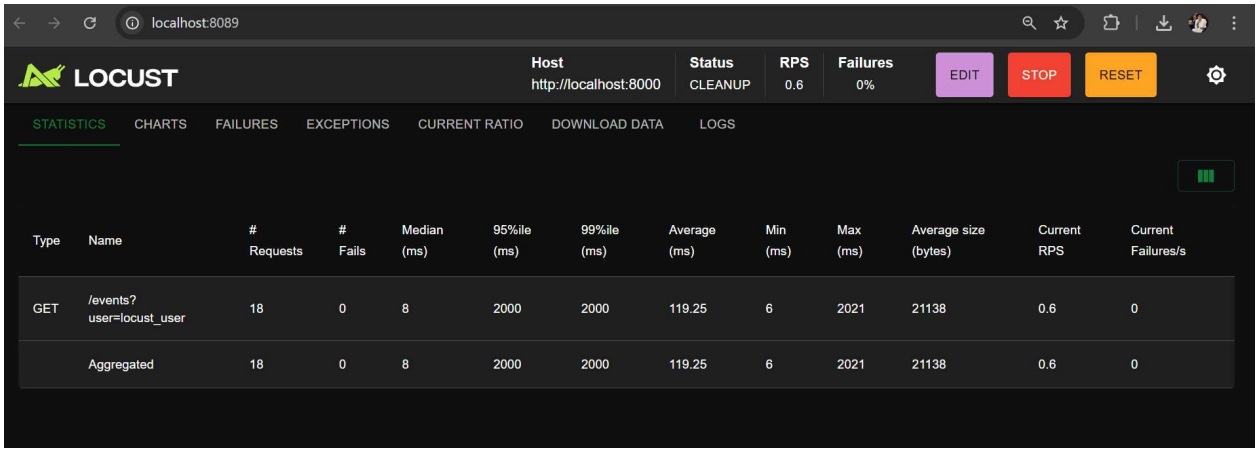

localhost:8089
LOCUST
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms)
GET /checkout 19 0 8 2000 2000 114.28
Aggregated 19 0 8 2000 2000 114.28
main.py
__init__.py
main.py > ...
from fastapi import FastAPI, Request, Form
from fastapi.responses import HTMLResponse, Redi
from fastapi.templating import Jinja2Templates
from database import get_db
from checkout import checkout_logic
app = FastAPI()
SRN = "PES2UG23CS103"
templates = Jinja2Templates(directory="templates
@app.on_event("startup")
PROBLEMS
OUTPUT
TERMINAL
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2UG23CS103> locust -f locust/checkout_locustfile.py
[2026-01-29 14:47:09,819] Lappy/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
C Com...
uvicorn
locust

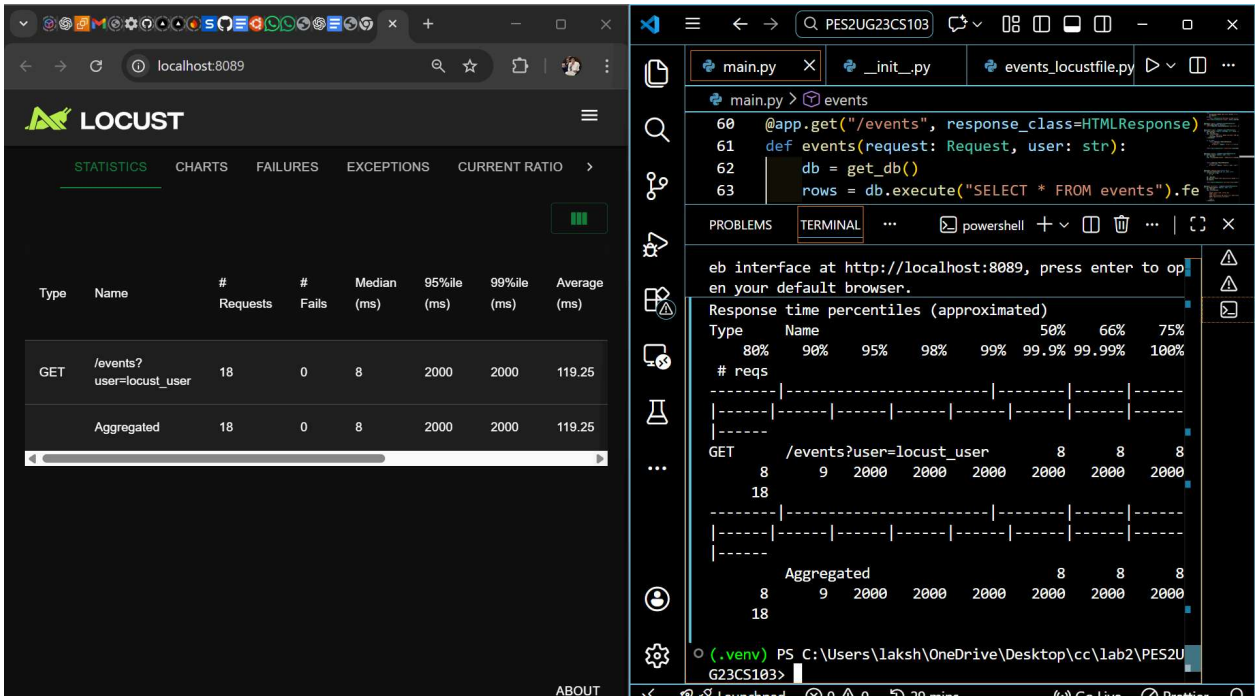

PES2UG23CS103
main.py
__init__.py
checkout > __init__.py > checkout_logic
3 def checkout_logic():
PROBLEMS
OUTPUT
TERMINAL
powershell
[2026-01-29 14:47:09,819] Lappy/INFO/locust.runners: All users spawned
KeyboardInterrupt
2026-01-29T09:22:44Z
[2026-01-29 14:52:44,856] Lappy/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs # fails | Avg Min Max Med | req/s failures/s
GET /checkout 19 0(0.00%) | 114 4 2026 8 | 0.64 0.00
Aggregated 19 0(0.00%) | 114
GET /checkout 8 9 9 10 10 2000 2000 2000 2000 2000 2000 19
Aggregated 8 9 9 10 10 2000 2000 2000 2000 2000 2000 19
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2UG23CS103>
localhost:8089
LOCUST
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT R
Type
Name
Requests
Fails
Median (ms)
95%ile (ms)
99%ile (ms)
GET
/checkout
19
0
8
2000
2000
Aggregated
19
0
8
2000
2000

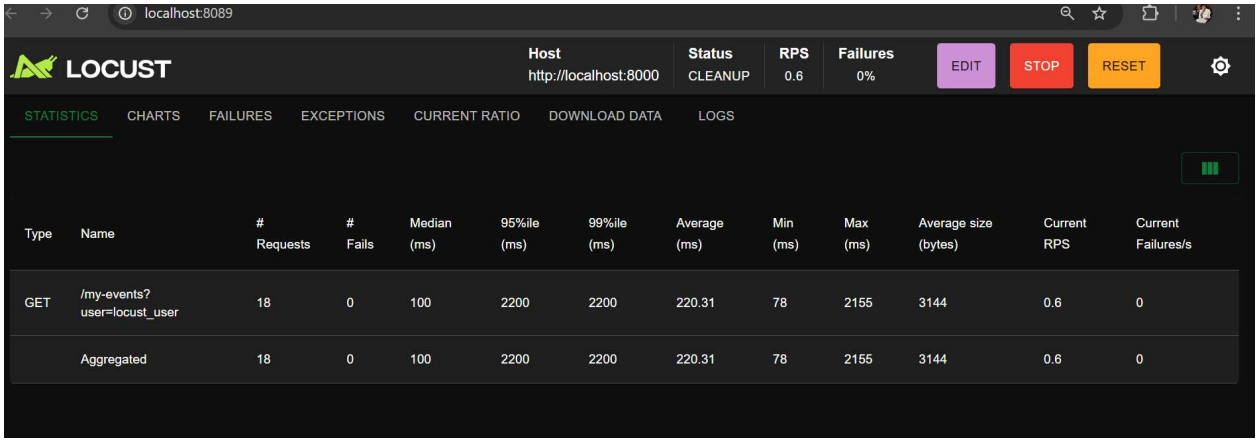

localhost:8089
LOCUST
Host
http://localhost:8000
Status
STOPPED
RPS
0.8
Failures
0%
NEW
RESET
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
DOWNLOAD DATA
LOGS
Type
Name
Requests
Fails
Median (ms)
95%ile (ms)
99%ile (ms)
Average (ms)
Min (ms)
Max (ms)
Average size (bytes)
Current RPS
Current Failures/s
GET
/checkout
20
0
7
2000
2000
107.07
4
2014
2797
0.8
0
Aggregated
20
0
7
2000
2000
107.07
4
2014
2797
0.8
0

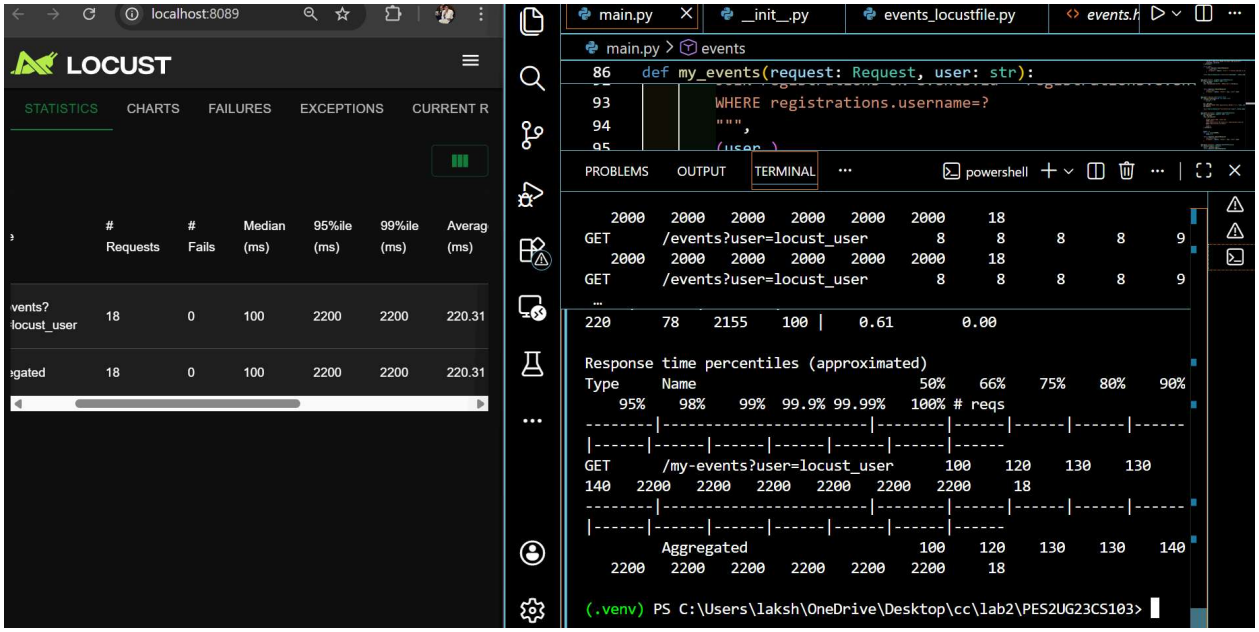

PES2UG23CS103
main.py
__init__.py
checkout > __init__.py > checkout_logic
3 def checkout_logic():
9 # Uncomment this line initially for the crash screenshot
10 #1 / 0
PROBLEMS OUTPUT TERMINAL powershell
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [4396]
INFO: Stopping reloader process [9956]
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2UG23CS103> locu
sed-argument
KeyboardInterrupt
2026-01-29T09:34:48Z
[2026-01-29 15:04:48,123] Lappy/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs # fails | Avg
Min Max Med | req/s failures/s
GET /checkout 20 0(0.00%) | 107
4 2013 7 | 0.70 0.00
GET /checkout 7 7 8
8 9 2000 2000 2000 2000 2000 2000 20
localhost:8089
LOCUST
STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT R
Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms) Min (ms) Max (ms)
0 7 2000 2000 107.07 4 2014
0 7 2000 2000 107.07 4 2014
ABOUT

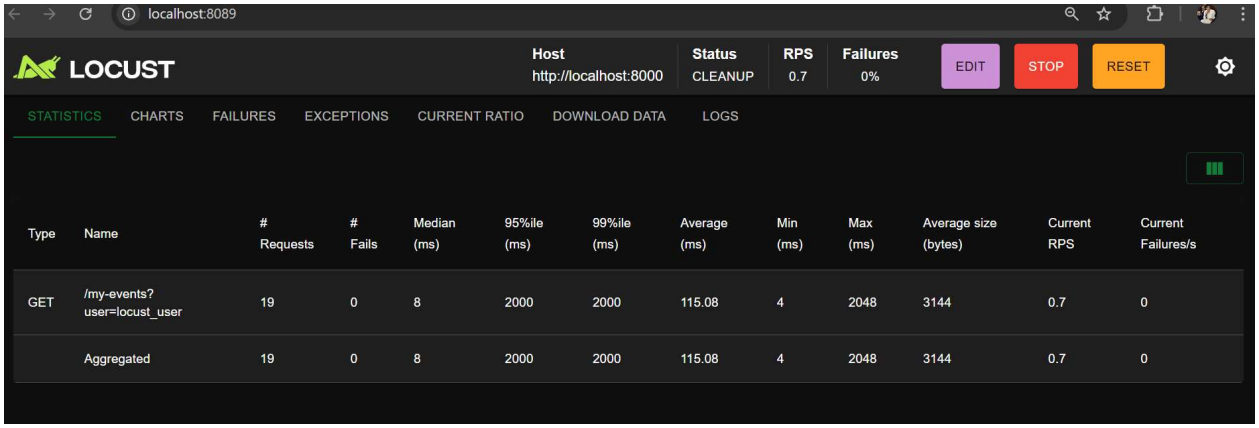

localhost:8089
LOCUST
Host http://localhost:8000
Status CLEANUP
RPS 0.7
Failures 0%
EDIT STOP RESET
STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms) Min (ms) Max (ms) Average size (bytes) Current RPS Current Failures/s
GET /events? user=locust_user 18 0 280 2300 2300 398.14 193 2346 21138 0.7 0
Aggregated 18 0 280 2300 2300 398.14 193 2346 21138 0.7 0

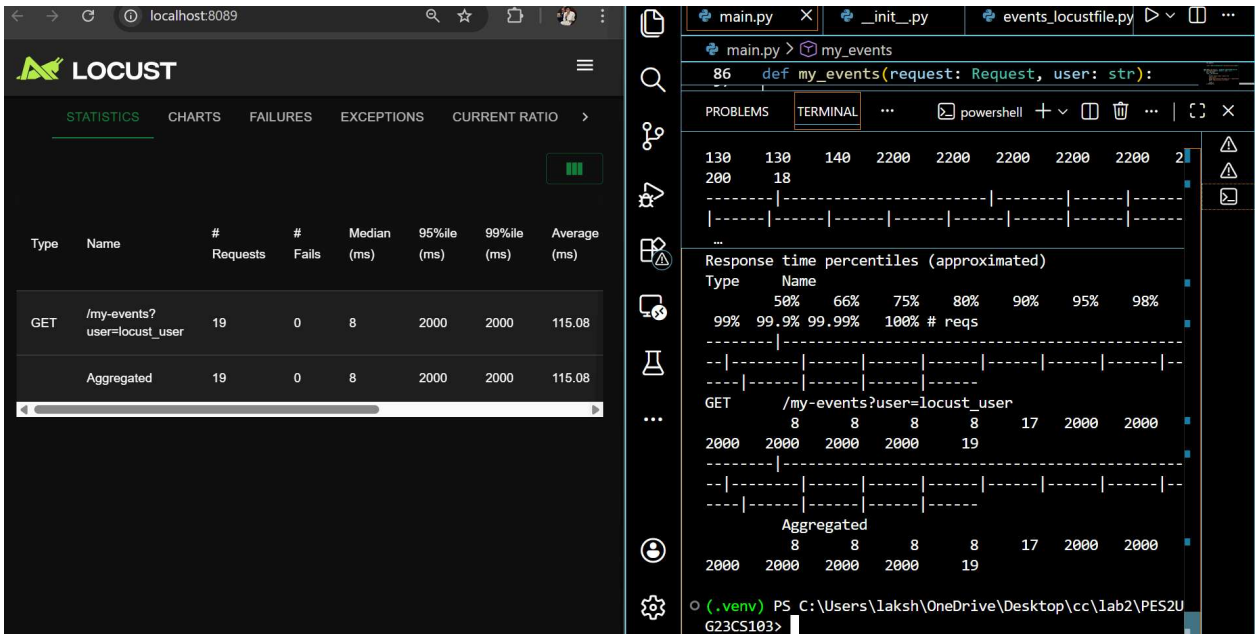

localhost:8089
LOCUST
Host http://localhost:8000
Status CLEANUP
RPS 0.6
Failures 0%
EDIT STOP RESET
STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOWNLOAD DATA LOGS
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms) Min (ms) Max (ms) Average size (bytes) Current RPS Current Failures/s
GET /events? user=locust_user 18 0 8 2000 2000 119.25 6 2021 21138 0.6 0
Aggregated 18 0 8 2000 2000 119.25 6 2021 21138 0.6 0


localhost:8089
LOCUST
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
Type
Name
Requests
Fails
Median (ms)
95%ile (ms)
99%ile (ms)
Average (ms)
GET
/events?user=locust_user
18
0
8
2000
2000
119.25
Aggregated
18
0
8
2000
2000
119.25
ABOUT
PES2UG23CS103
main.py
__init__.py
events_locustfile.py
main.py > events
@app.get("/events", response_class=HTMLResponse)
def events(request: Request, user: str):
db = get_db()
rows = db.execute("SELECT * FROM events").fe
PROBLEMS
TERMINAL
powershell
eb interface at http://localhost:8089, press enter to op
en your default browser.
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 8 8 8
8 9 2000 2000 2000 2000 2000 2000
18
Aggregated 8 8 8
8 9 2000 2000 2000 2000 2000 2000
18
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2U
G23CS103>


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
100
2200
2200
220.31
78
2155
3144
0.6
0
Aggregated
18
0
100
2200
2200
220.31
78
2155
3144
0.6
0


localhost:8089
LOCUST
STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT R
Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Averag (ms)
vents? locust_user 18 0 100 2200 2200 220.31
gated 18 0 100 2200 2200 220.31
main.py __init__.py events_locustfile.py events.h
main.py > events
86 def my_events(request: Request, user: str):
93 WHERE registrations.username=?
94 """,
PROBLEMS OUTPUT TERMINAL powershell
2000 2000 2000 2000 2000 2000 18
GET /events?user=locust_user 8 8 8 8 9
2000 2000 2000 2000 2000 2000 18
GET /events?user=locust_user 8 8 8 8 9
220 78 2155 100 | 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 100 120 130 130
140 2200 2200 2200 2200 2200 2200 18
Aggregated 100 120 130 130 140
2200 2200 2200 2200 2200 2200 18
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2UG23CS103>


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 /my-events? user=locust_user 19 0 8 2000 2000 115.08 4 2048 3144 0.7 0
Aggregated 19 0 8 2000 2000 115.08 4 2048 3144 0.7 0


localhost:8089
LOCUST
STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO
Type Name # Requests # Fails Median (ms) 95%ile (ms) 99%ile (ms) Average (ms)
GET /my-events? user=locust_user 19 0 8 2000 2000 115.08
Aggregated 19 0 8 2000 2000 115.08
main.py __init__.py events_locustfile.py
main.py > my_events
86 def my_events(request: Request, user: str):
PROBLEMS TERMINAL powershell
130 130 140 2200 2200 2200 2200 2200 2
200 18
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
8 8 8 8 17 2000 2000
2000 2000 2000 2000 19
Aggregated
8 8 8 8 17 2000 2000
2000 2000 2000 2000 19
(.venv) PS C:\Users\laksh\OneDrive\Desktop\cc\lab2\PES2U
G23CS103>

Optimized Routes

Route 1: `/events`

1. What was the bottleneck? The `/events` route contained an intentional CPU-intensive loop that performed millions of unnecessary computations, causing artificial delay in every request.

2. What change did you make? Removed the wasteful computation loop from the route logic and kept only the required database query and template rendering.

3. Why did the performance improve? Because unnecessary CPU processing was eliminated, the server spent less time per request, reducing response time and improving overall performance.

Route 2: `/my-events`

1. What was the bottleneck? The `/my-events` route also contained an artificial delay loop that performed unnecessary iterations, increasing execution time for each request.

2. What change did you make? Removed the dummy computation loop and retained only the database join query and template rendering logic.

3. Why did the performance improve? Eliminating redundant computations reduced CPU load, resulting in faster request processing and better response times.

Route 3: `/checkout` (from Part 8)

1. What was the bottleneck? The checkout logic used inefficient iterative computation to calculate total fee, increasing processing time.

2. What change did you make? Replaced inefficient looping logic with optimized aggregation logic.

3. Why did the performance improve? Optimized computation reduced algorithmic complexity and execution time, improving response speed.