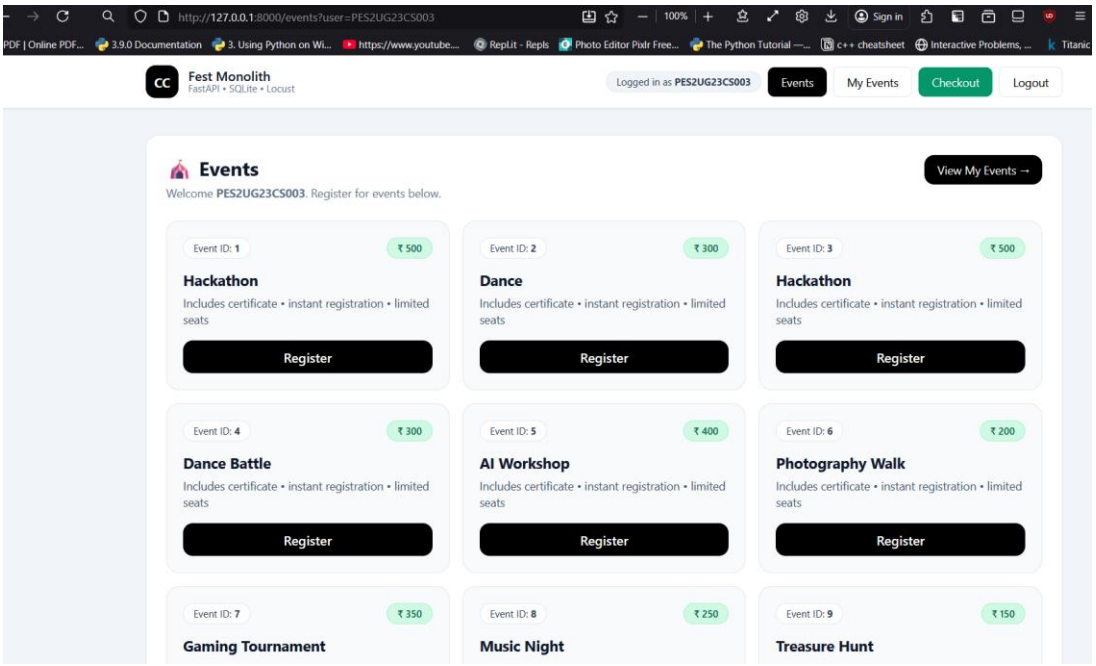
CC-Semester-6

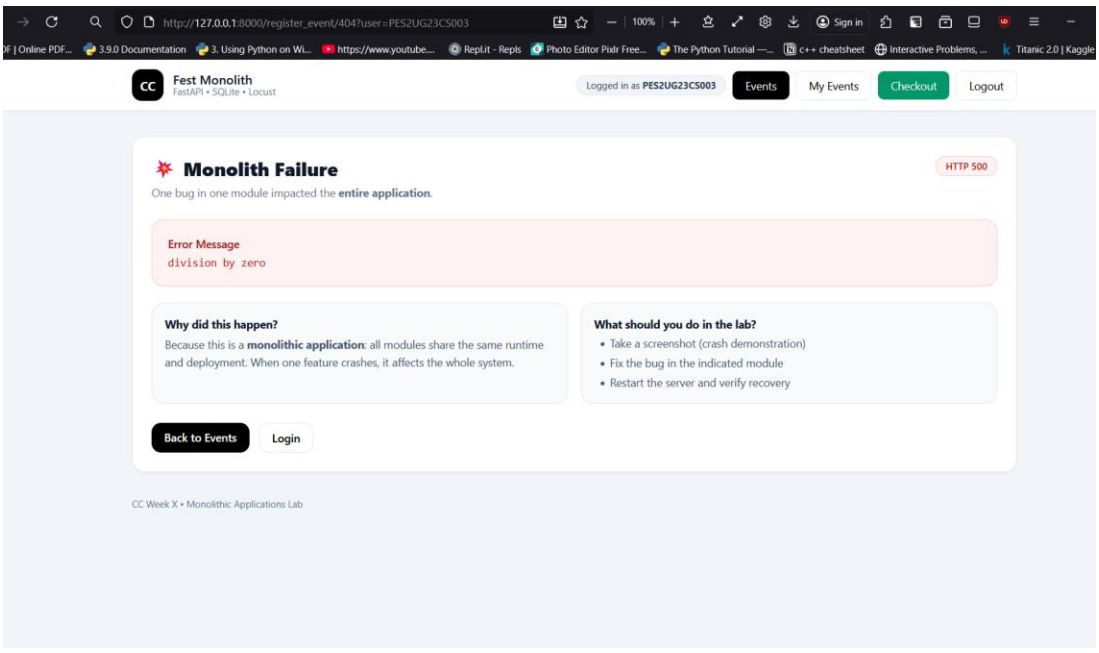
LAB-2 **Date:** 20/01/2026

Name: A Shri Karthik	**SRN:** PES2UG23CS003	**Section:** A

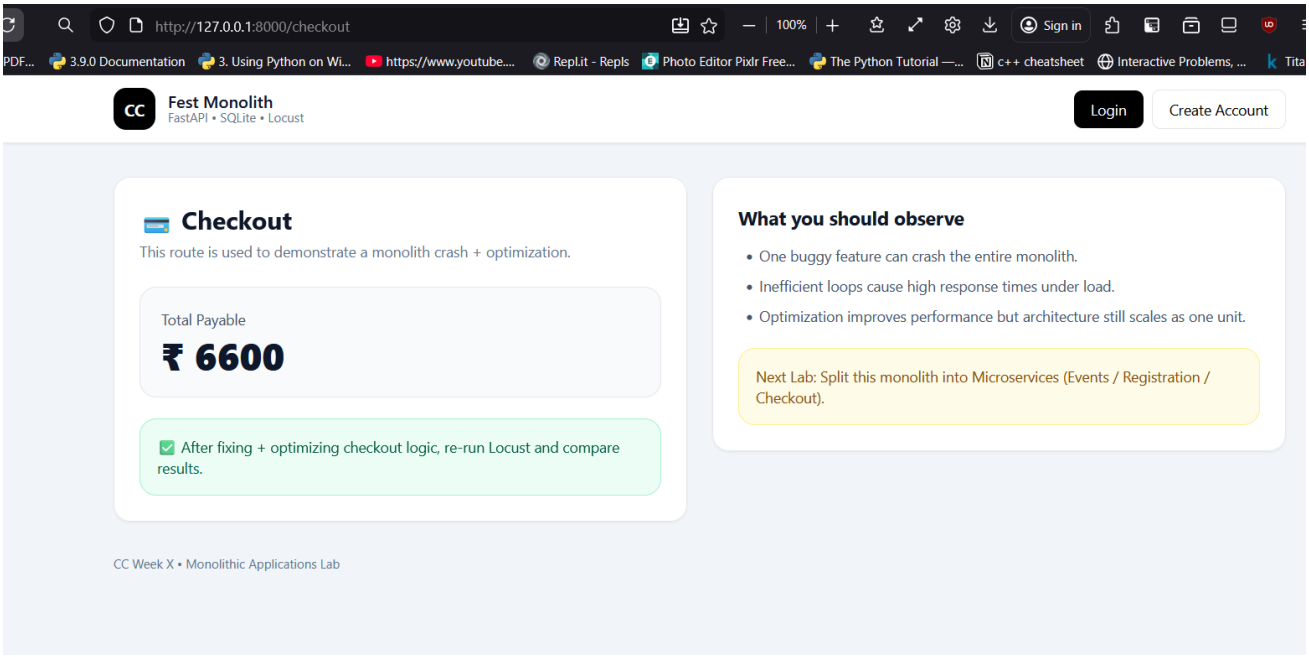
SS1



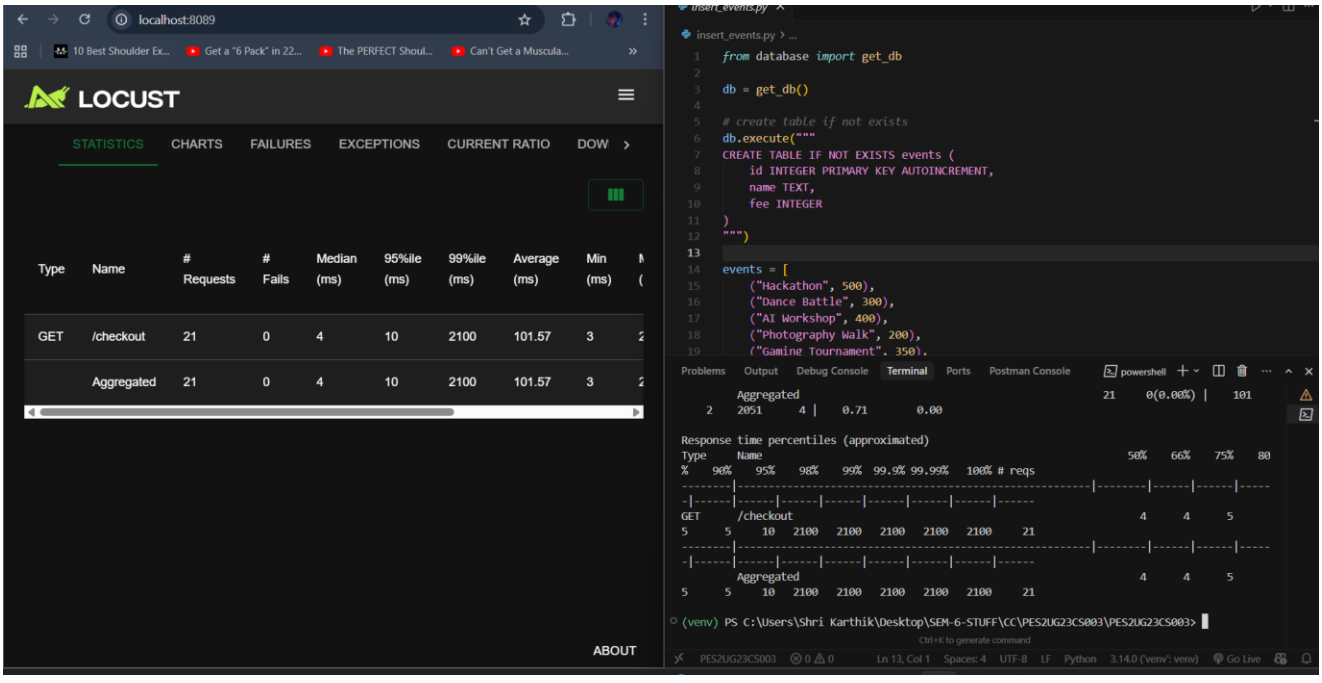
SS2



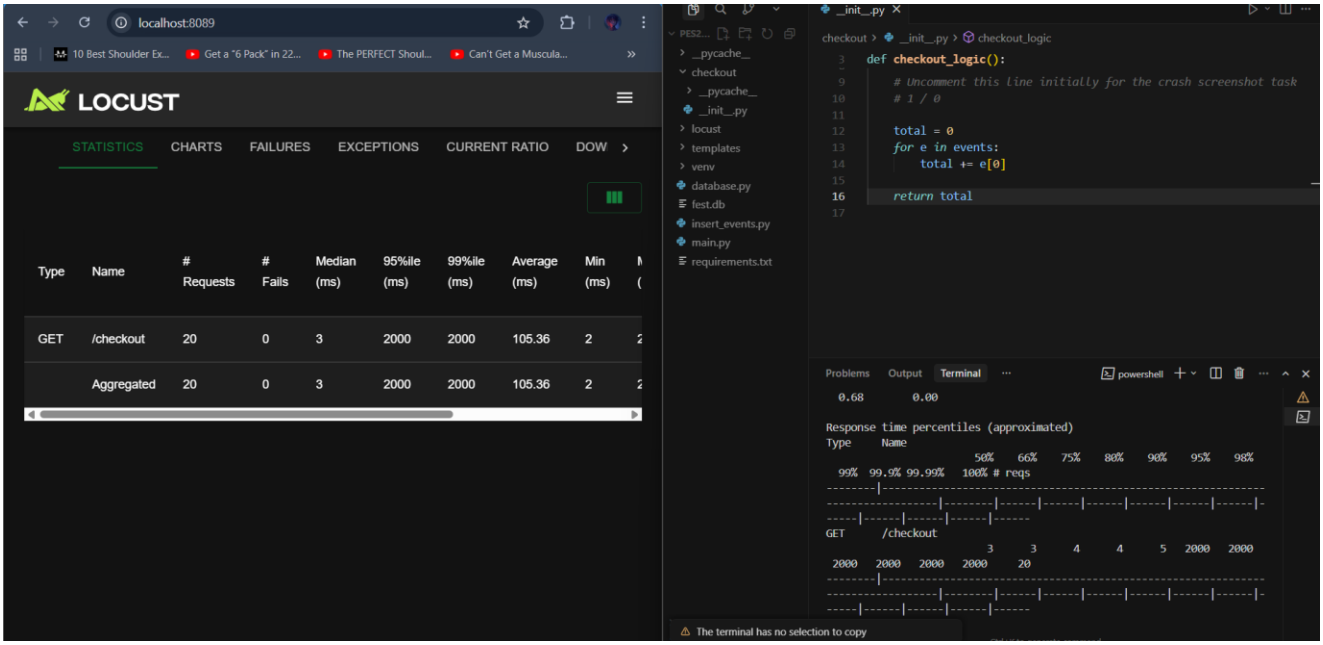
SS3


http://127.0.0.1:8000/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

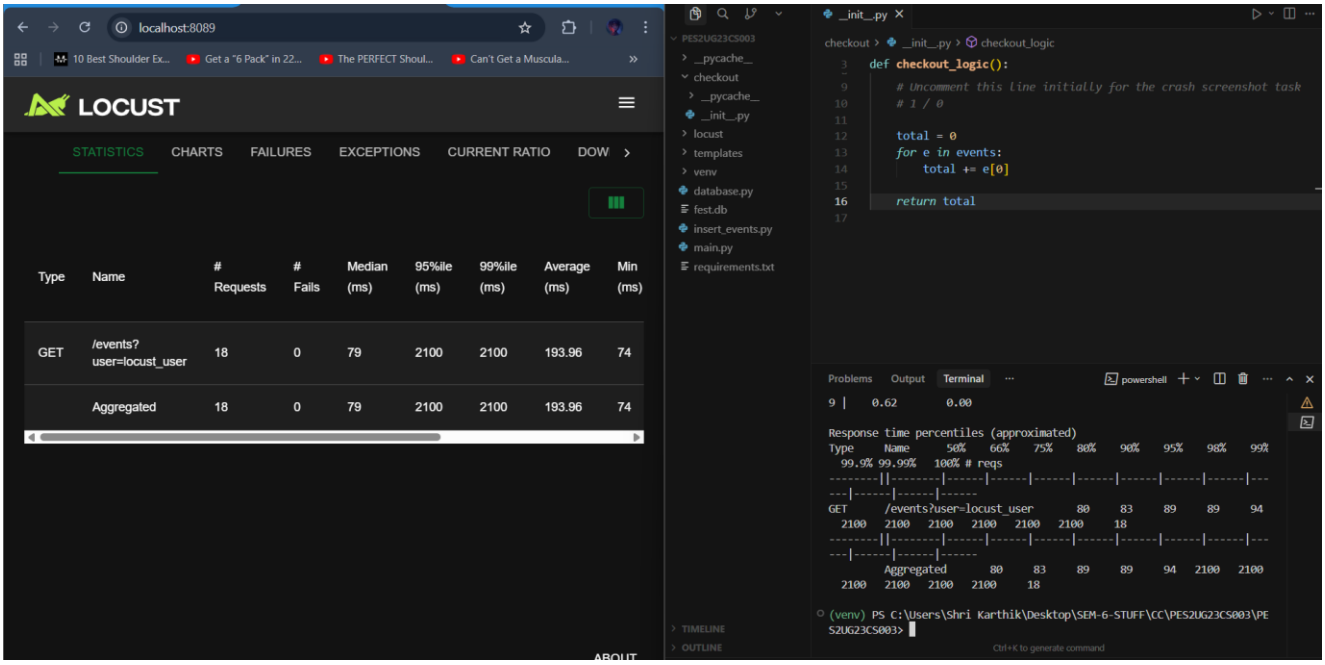
SS4


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
21
0
4
10
2100
101.57
3
Aggregated
21
0
4
10
2100
101.57
3
ABOUT
insert_events.py
from database import get_db
db = get_db()
# create table if not exists
db.execute("""
CREATE TABLE IF NOT EXISTS events (
id INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT,
fee INTEGER
)
""")
events = [
("Hackathon", 500),
("Dance Battle", 300),
("AI Workshop", 400),
("Photography Walk", 200),
("Gaming Tournament", 350),
Problems
Output
Debug Console
Terminal
Ports
Postman Console
powershell
Aggregated 21 0(0.00%) | 101
2 2051 4 | 0.71 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80
% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
GET /checkout 4 4 5
5 5 10 2100 2100 2100 2100 2100 21
Aggregated 4 4 5
5 5 10 2100 2100 2100 2100 2100 21
(venv) PS C:\Users\Shri Karthik\Desktop\SEM-6-STUFF\CC\PES2UG23CS003\PES2UG23CS003>

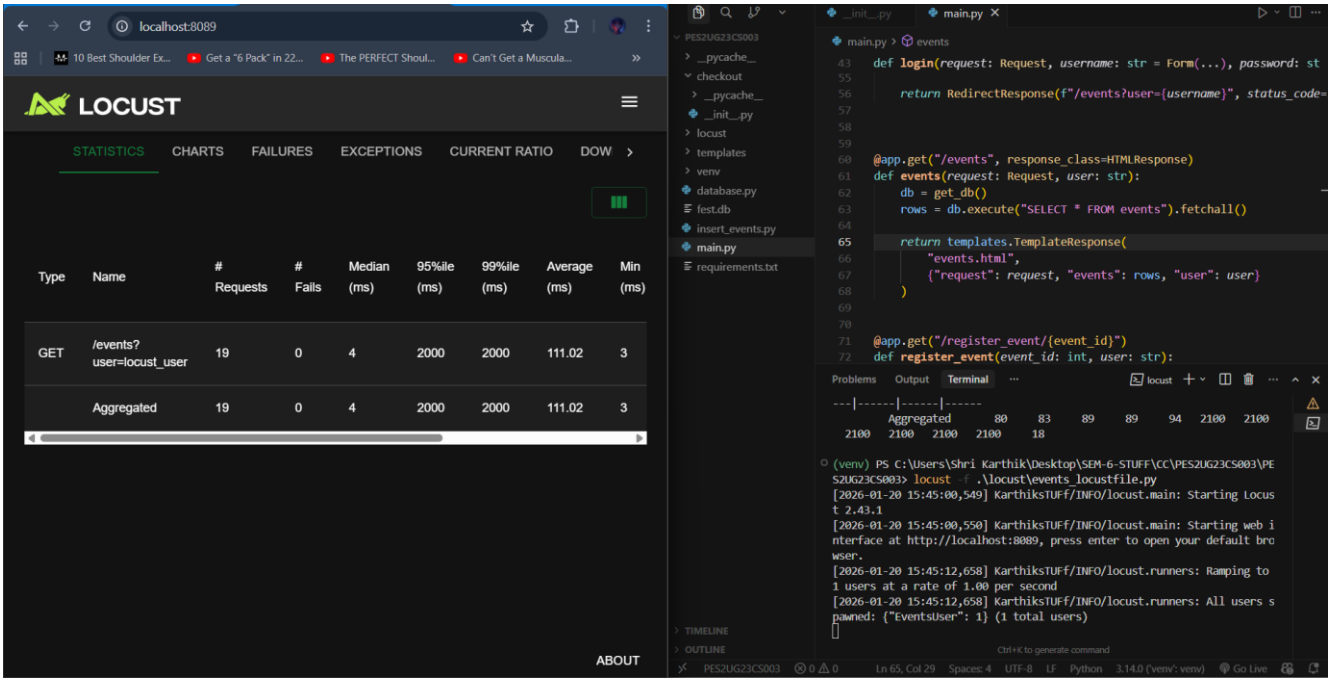
SS5


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 20 0 3 2000 2000 105.36 2
Aggregated 20 0 3 2000 2000 105.36 2
def checkout_logic():
# Uncomment this line initially for the crash screenshot task
# 1 / 0
total = 0
for e in events:
total += e[0]
return total
Problems Output Terminal
powershell
0.68 0.00
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
GET /checkout 3 3 4 4 5 2000 2000 2000 2000 2000 2000 20
The terminal has no selection to copy

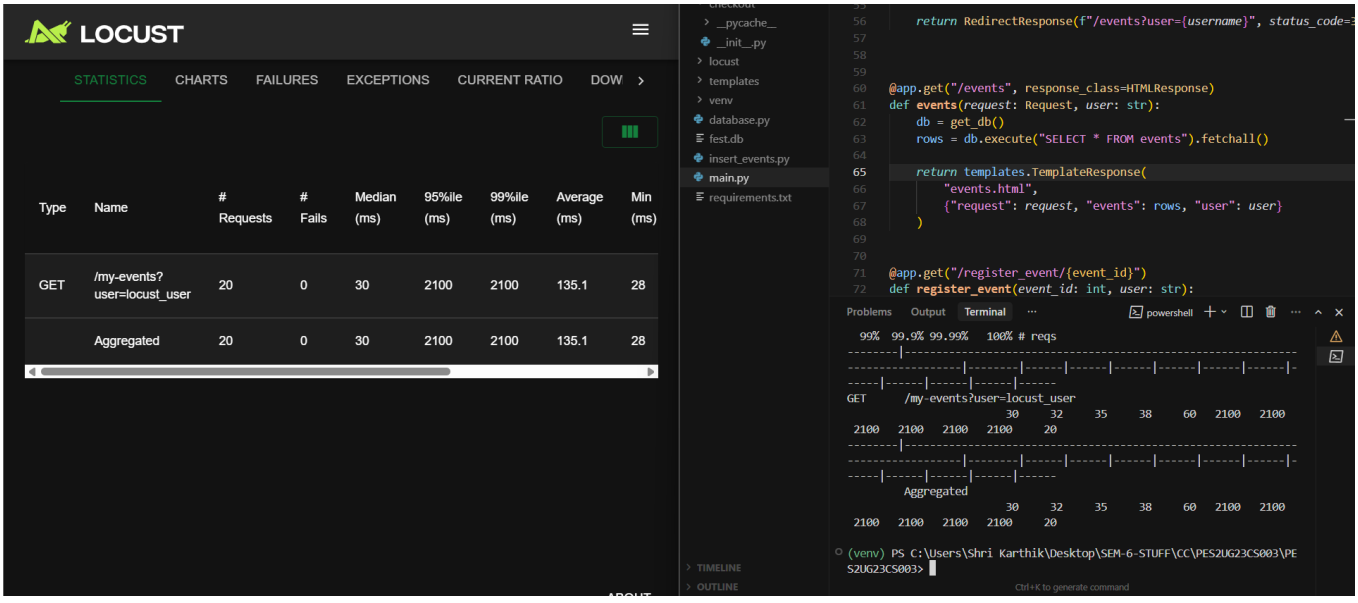
SS6


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 /events?user=locust_user 18 0 79 2100 2100 193.96 74
Aggregated 18 0 79 2100 2100 193.96 74
def checkout_logic():
# Uncomment this line initially for the crash screenshot task
# 1 / 0
total = 0
for e in events:
total += e[0]
return total
Problems Output Terminal
powershell
9 | 0.62 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 80 83 89 89 94 2100 2100 2100 2100 2100 2100 18
Aggregated 80 83 89 89 94 2100 2100 2100 2100 2100 2100 18
(venv) PS C:\Users\Shri Karthik\Desktop\SEM-6-STUFF\CC\PES2UG23CS003\PES2UG23CS003>

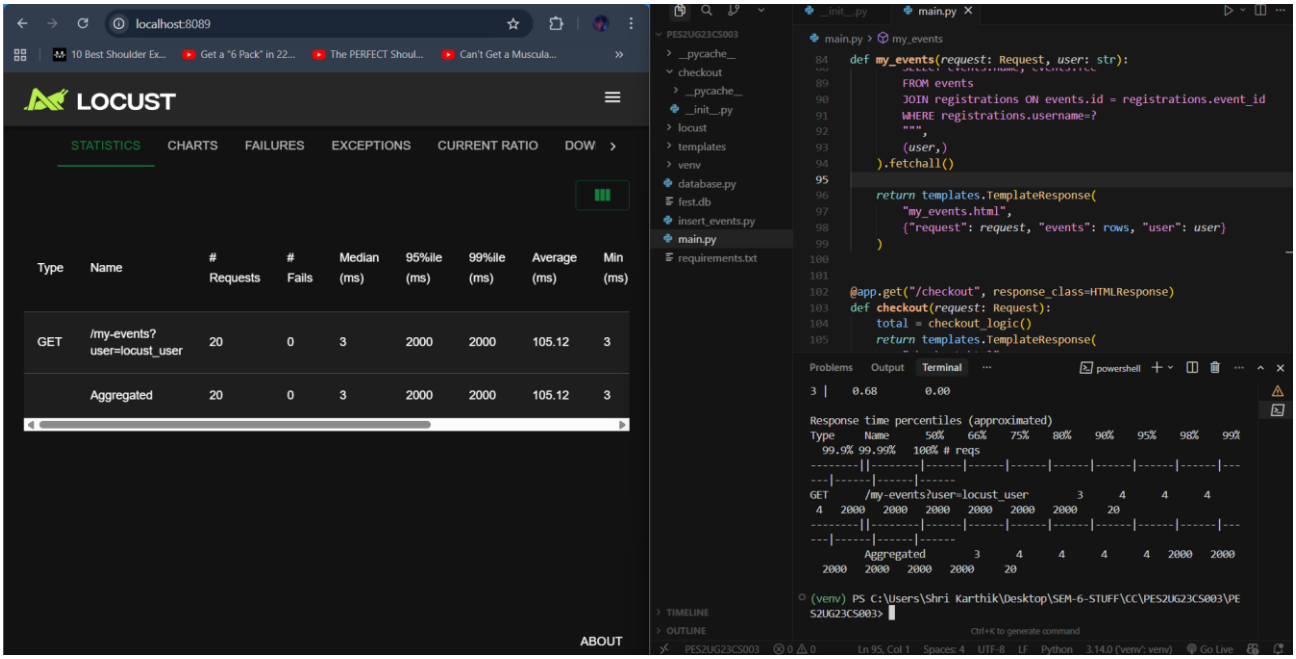
SS7


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
/events? user=locust_user
19
0
4
2000
2000
111.02
3
Aggregated
19
0
4
2000
2000
111.02
3
ABOUT
def login(request: Request, username: str = Form(...), password: st
return RedirectResponse(f"/events?user={username}", status_code=
@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}
)
@app.get("/register_event/{event_id}")
def register_event(event_id: int, user: str):
Problems
Output
Terminal
(venv) PS C:\Users\Shri Karthik\Desktop\SEM-6-STUFF\CC\PES2UG23CS003\PES2UG23CS003> locust -f .\locust\events_locustfile.py
[2026-01-20 15:45:00,549] KarthiksTUFf/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:45:00,550] KarthiksTUFf/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-20 15:45:12,658] KarthiksTUFf/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:45:12,658] KarthiksTUFf/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)

SS8


LOCUST
STATISTICS
CHARTS
FAILURES
EXCEPTIONS
CURRENT RATIO
Type
Name
Requests
Fails
Median (ms)
95%ile (ms)
99%ile (ms)
Average (ms)
Min (ms)
GET
/my-events? user=locust_user
20
0
30
2100
2100
135.1
28
Aggregated
20
0
30
2100
2100
135.1
28
@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}
)
@app.get("/register_event/{event_id}")
def register_event(event_id: int, user: str):
Problems
Output
Terminal
powershell
99% 99.9% 99.99% 100% # reqs
GET /my-events?user=locust_user
30 32 35 38 60 2100 2100
2100 2100 2100 2100 20
Aggregated
30 32 35 38 60 2100 2100
2100 2100 2100 2100 20
(venv) PS C:\Users\Shri Karthik\Desktop\SEM-6-STUFF\CC\PES2UG23CS003\PES2UG23CS003>

SS9



2nd Case

I) In this case, the loop was wasting time and increasing the response time for each request.
II) We just remove the waste cycle or comment it.
III) The loop is removed and the time to send the response for each request decreases thus improving the performance

3rd Case

I) In this also, the loop was wasting time and increasing the response time for each request.
II) We just remove the waste cycle or comment it.
III) The loop is removed and the time to send the response for each request decreases thus improving the performance

Git Repo Link - https://github.com/Kart8ik/Cloud-Computing-Lab-PES2UG23CS003