

CLOUD COMPUTING LAB

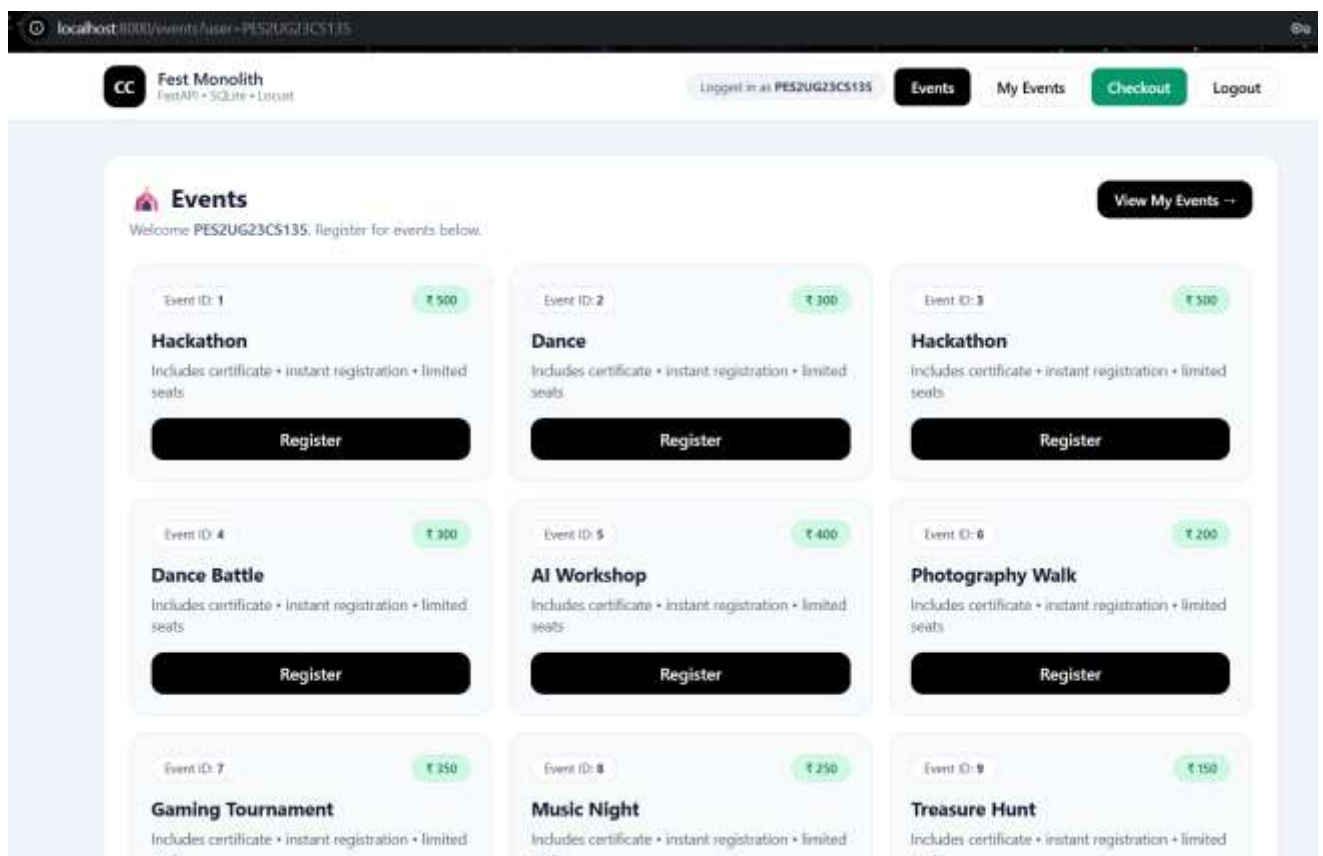
(WEEK-2: MONLITHIC)

NAME: B TEJA DEEP SAI KRISHNA

SRN: PES2UG23CS135

SEC: C

Login to events page:



Issue encountered:

←

→

↺

localhost:8000/register_event/404?user=PES2UG23CS135

☆

🔖

🌐

⋮

CC

Fest Monolith

FastAPI • SQLite • Locust

Logged in as PES2UG23CS135

Logout

Events

My Events

Checkout

🌟

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

After correcting the issue:

localhost:8000/checkout



Fest Monolith
FastAPI • SQLite • Locust



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.

CC Week X • Monolithic Applications Lab

Before optimizing checkout:

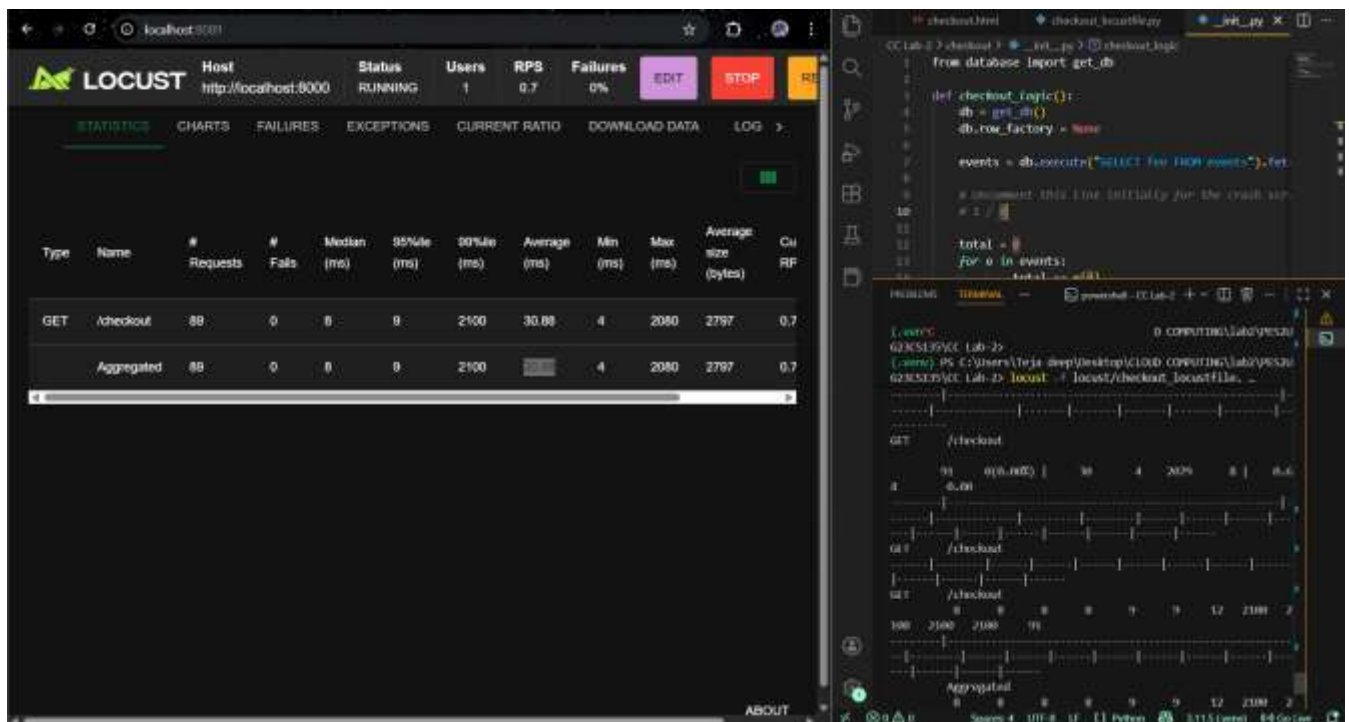
The screenshot shows the Locust web interface on the left and a terminal window on the right. The Locust interface displays statistics for the GET /checkout route, showing 19 requests, 0 failures, and an average response time of 115.23 ms. The terminal window shows the checkout logic in checkout_locustfile.py, which uses a database to fetch events and calculate the total payable. The terminal output shows the results of the checkout logic, including the total payable amount of ₹ 6600.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)
GET	/checkout	19	0	9	2000	2000	115.23	5	2044	2797
Aggregated		19	0	9	2000	2000	115.23	5	2044	2797

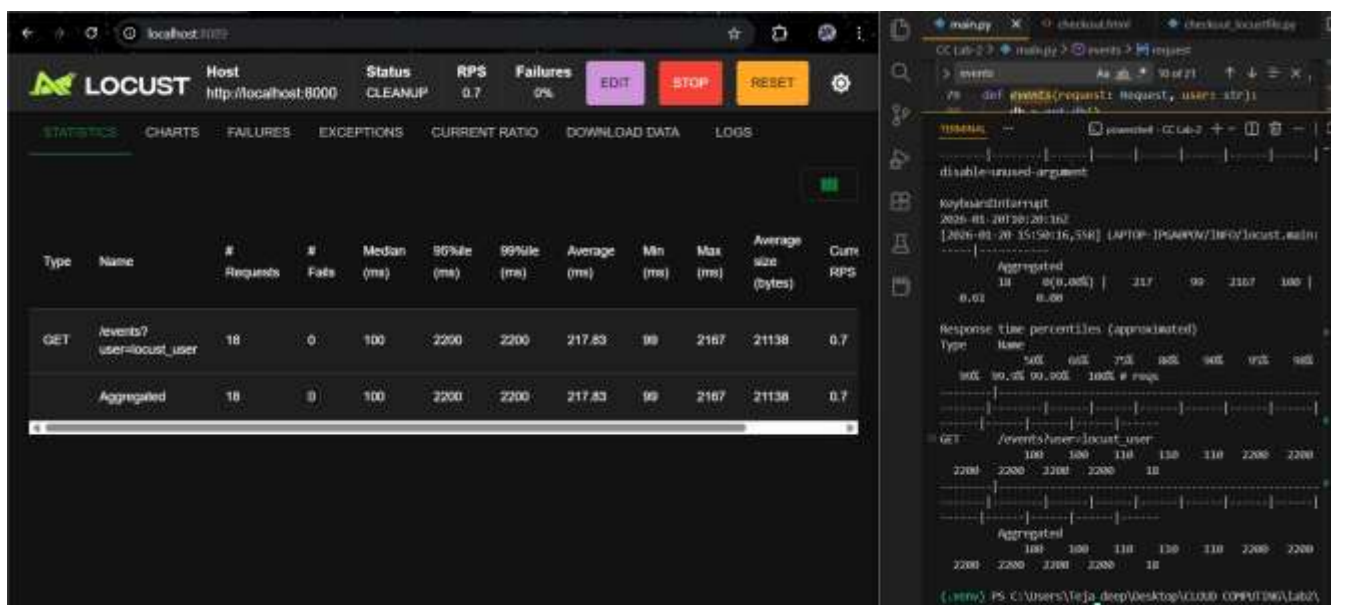
```
1 from database import get_db
2
3 def checkout_logic():
4     db = get_db()
5     db.row_factory = None
6     events = db.execute("SELECT fee FROM events").fetchall()
7
8     # ... (rest of the logic) ...
```

```
CC Lab-2 > checkout > _init_py > checkout_locustfile.py
1
2
3 def checkout_logic():
4     db = get_db()
5     db.row_factory = None
6     events = db.execute("SELECT fee FROM events").fetchall()
7
8     # ... (rest of the logic) ...
```

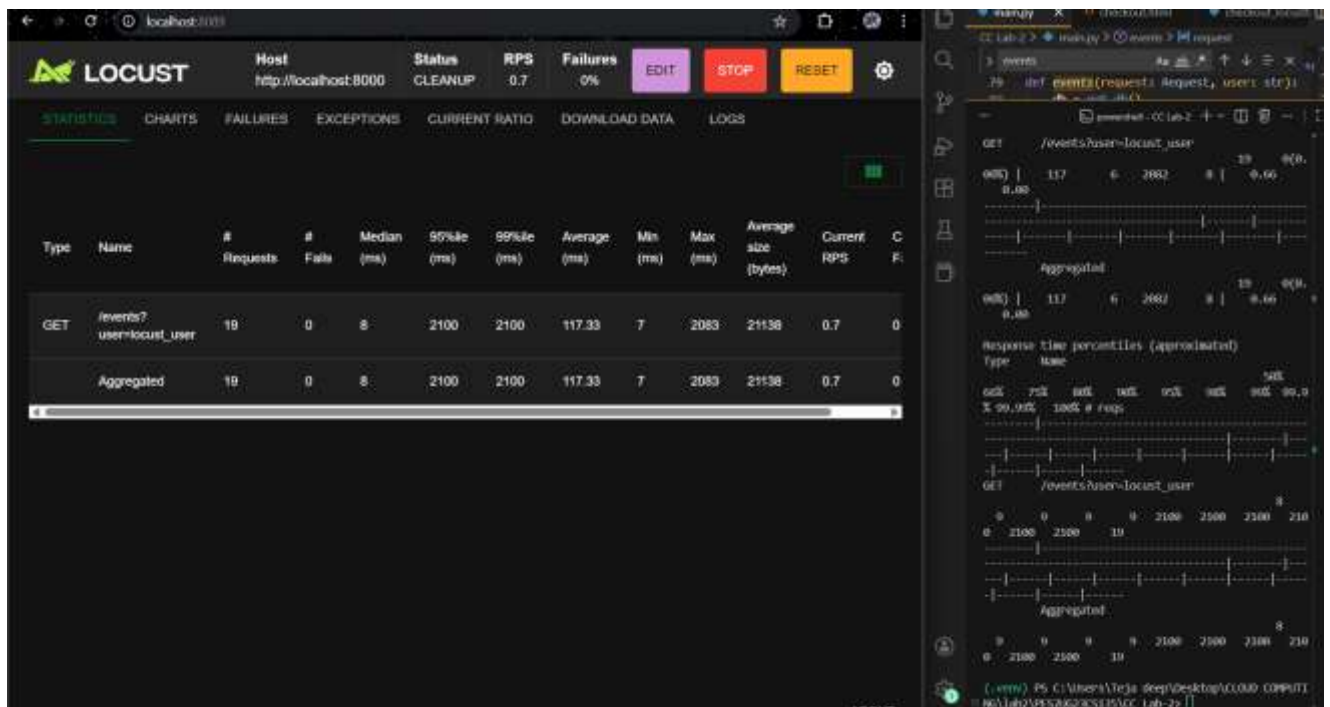
After optimization of checkout:



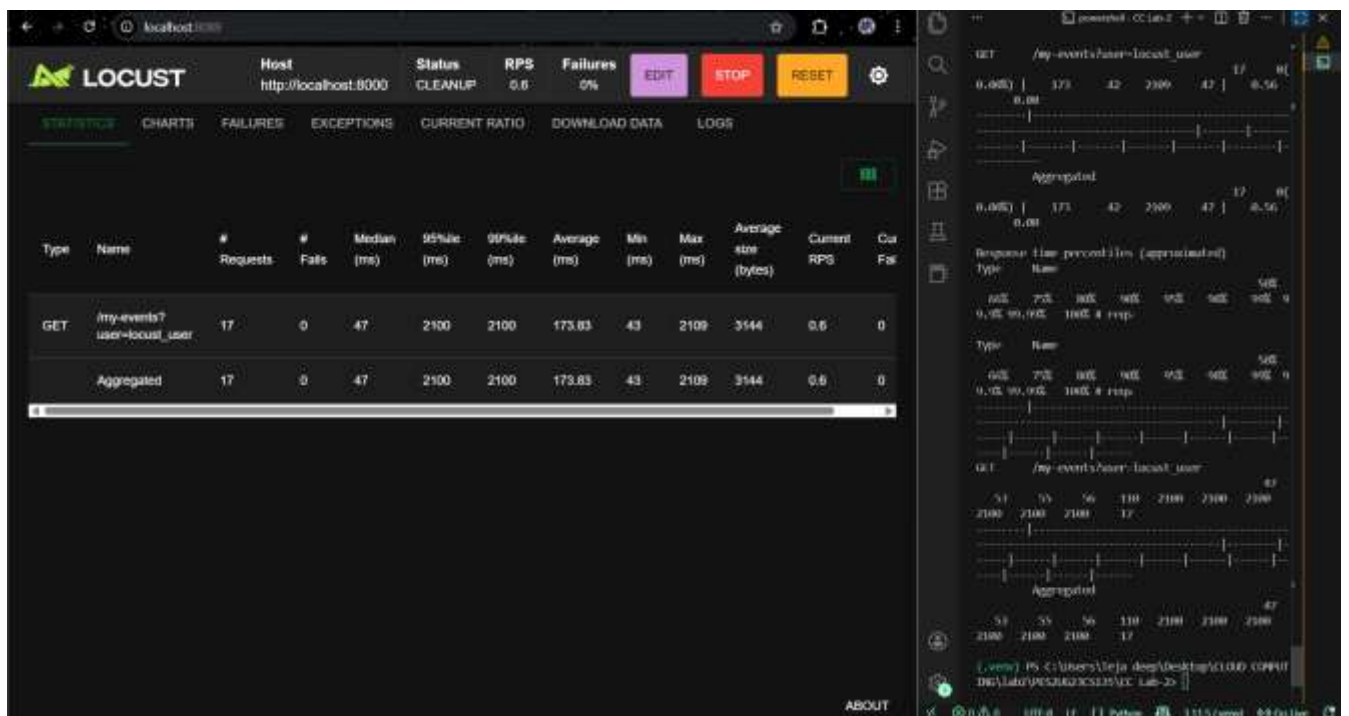
Before events optimization:



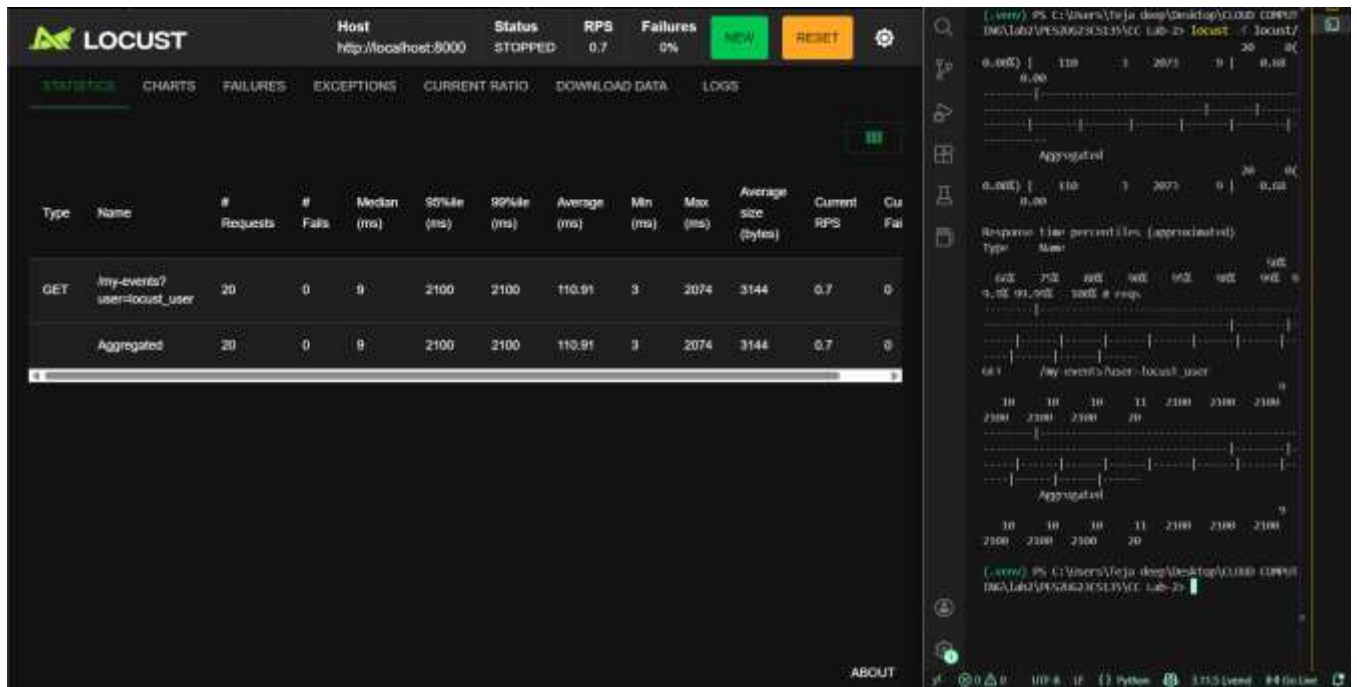
After events optimization:



Before myevents optimization:



After myevents optimization:



GITHUB REPO:

<https://github.com/TejaDeep11/CLOUD-COMPUTING-LAB/tree/main/CC%20LAB%202>