

CC LAB-2

Name: Nikitha Devaraj

SRN: PES2UG23CS388

Date : 20.01.2026

Github repository link :

https://github.com/PES2UG23CS388/CC-Lab-2_PES2UG23CS388

The screenshot shows a web application interface for 'Fest Monolith' at localhost:8000/events?user=PES2UG23CS388. The user is logged in as 'PES2UG23CS388'. The interface displays a grid of nine event cards:

- Event ID: 1** (Hackathon): ₹ 500. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 2** (Dance): ₹ 300. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 3** (Hackathon): ₹ 500. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 4** (Dance Battle): ₹ 300. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 5** (AI Workshop): ₹ 400. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 6** (Photography Walk): ₹ 200. Includes certificate • instant registration • limited seats. **Register**
- Event ID: 7** (Gaming Tournament): ₹ 350.
- Event ID: 8** (Music Night): ₹ 250.
- Treasure Hunt**: ₹ 150.

Navigation and account information are visible at the top right, including 'Events', 'My Events', 'Checkout', and 'Logout'.

Monolith Failure

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

Checkout

This route is used to demonstrate a monolith crash + optimization.

Total Payable
₹ 6600

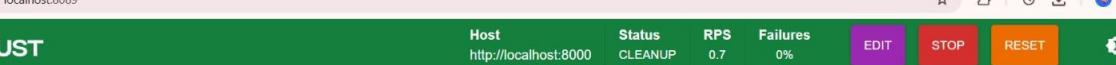
After fixing + optimizing checkout logic, re-run Locust and compare results.

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



A screenshot of the Locust web interface. At the top, there's a header bar with browser controls (back, forward, search) and a URL field showing "localhost:8089". To the right are icons for star, copy, refresh, download, and help. Below the header is a green navigation bar with the Locust logo and tabs for "STATISTICS", "CHARTS", "FAILURES", "EXCEPTIONS", "CURRENT RATIO", "DOWNLOAD DATA", and "LOGS". The "STATISTICS" tab is currently selected. On the right side of the main content area, there are buttons for "EDIT", "STOP", and "RESET". The main content area displays performance metrics: Host (http://localhost:8000), Status (CLEANUP), RPS (0.7), and Failures (0%). Below these metrics is a table with two rows. The first row has columns for Type, Name, # Requests, # Fails, Median (ms), 95%ile (ms), 99%ile (ms), Average (ms), Min (ms), Max (ms), Average size (bytes), Current RPS, and Current Failures/s. The second row shows data for a GET request to "/checkout" with 18 requests, 0 fails, and a median of 7 ms. The third row is labeled "Aggregated" with the same values as the first row.

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	18	0	7	2000	2000	120.44	6	2037	2797	0.7	0
	Aggregated	18	0	7	2000	2000	120.44	6	2037	2797	0.7	0

```
Command Prompt - locust -f x + v - x

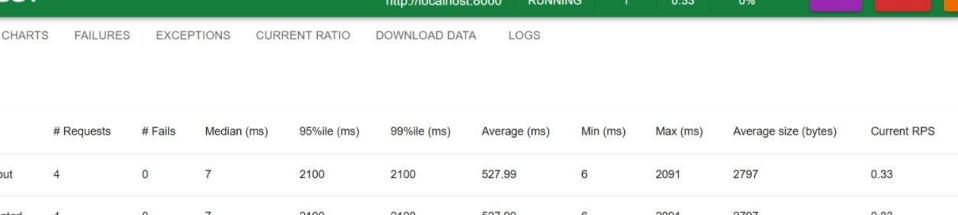
The system cannot find the path specified.

C:\Users\Nikitha\Desktop\PES2UG23CS388>cd CC Lab-2

C:\Users\Nikitha\Desktop\PES2UG23CS388>CC Lab-2>.\venv\Scripts\activate

(.venv) C:\Users\Nikitha\Desktop\PES2UG23CS388>CC Lab-2>locust -f locust/checkout_locustfile.py
[2026-01-20 15:28:15,895] Nikitha-D/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:28:15,895] Nikitha-D/INFO/locust.main: Starting web interface at http://localhost:8089, pres
s enter to open your default browser.
[2026-01-20 15:29:35,811] Nikitha-D/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:29:35,811] Nikitha-D/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total us
ers)
```





The Locust performance testing dashboard shows the following details:

- Host:** http://localhost:8000
- Status:** RUNNING
- Users:** 1
- RPS:** 0.33
- Failures:** 0%

Navigation tabs include: STATISTICS, CHARTS, FAILURES, EXCEPTIONS, CURRENT RATIO, DOWNLOAD DATA, and LOGS.

Table of results:

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	4	0	7	2100	2100	527.99	6	2091	2797	0.33	0
	Aggregated	4	0	7	2100	2100	527.99	6	2091	2797	0.33	0

Command Prompt window showing Locust command execution and logs:

```
Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.

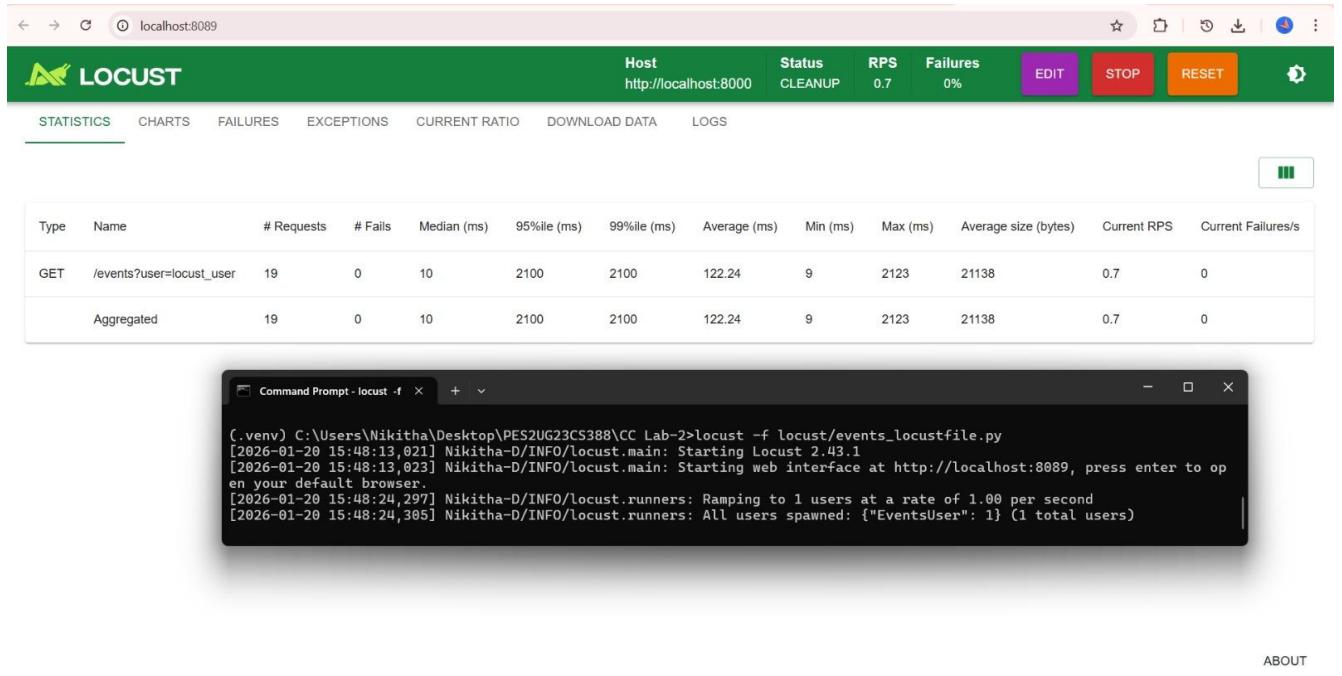
C:\Users\Nikitha>cd Desktop
C:\Users\Nikitha\Desktop>cd PES2UG23CS388
C:\Users\Nikitha\Desktop\PES2UG23CS388>cd CC Lab-2
C:\Users\Nikitha\Desktop\PES2UG23CS388\CC Lab-2>.\.venv\Scripts\activate

(.venv) C:\Users\Nikitha\Desktop\PES2UG23CS388\CC Lab-2>locust -f locust/checkout_locustfile.py
[2026-01-20 15:38:40,295] Nikitha-D/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:38:40,295] Nikitha-D/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-20 15:39:09,866] Nikitha-D/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:39:09,869] Nikitha-D/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
```

The screenshot shows the Locust web interface at <http://localhost:8089>. The top navigation bar includes links for STATISTICS, CHARTS, FAILURES, EXCEPTIONS, CURRENT RATIO, DOWNLOAD DATA, and LOGS. A green 'NEW' button and an orange 'RESET' button are also present. The main content area displays a table of performance metrics for a 'GET' request to '/events?user=locust_user'. Below this is an aggregated summary. To the right, a command prompt window titled 'Command Prompt - locust -f' shows Locust starting and spawning users. The bottom of the screen features a footer with links for 'ABOUT', 'CONTACT', 'REPORT BUG', and 'CONTRIBUTORS'.

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	15	0	340	2400	2400	474.1	277	2396	21138	0.5	0
Aggregated		15	0	340	2400	2400	474.1	277	2396	21138	0.5	0

```
(.venv) C:\Users\Nikitha\Desktop\PES2UG23CS388\CC_Lab-2>locust -f locust/events_locustfile.py
[2026-01-20 15:42:48,280] Nikitha-D/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:42:48,280] Nikitha-D/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-20 15:42:55,825] Nikitha-D/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:42:55,827] Nikitha-D/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
```



ABOUT

Checkout Route

Earlier, the checkout code was slow because it used unnecessary loops to calculate the total amount. I simplified the logic by adding all the event fees using a single loop. This reduced extra processing and made the checkout faster.

Events Route

The events page was slow because the data was processed multiple times. I optimized the code by removing repeated loops and handling the data only once. This reduced the response time and improved performance.

My-Events Route

The my-events page was slow due to inefficient handling of user data. I simplified the logic by avoiding repeated processing and using a direct approach. This made the page load faster.