



Department of Computer Science and Engineering
UE23CS351B Cloud Computing
Lab 2 Submission

Name	Sudiksha Chindula
SRN	PES1UG23CS902
Section	L
Date	29 January 2026

Screenshot SS1

The screenshot shows a web browser window for the URL `localhost:8000/events?user=PES1UG23CS902`. The page title is "Events". At the top right, there are buttons for "Events", "My Events", "Checkout", and "Logout". A message "Logged in as PES1UG23CS902" is displayed. The main content area is titled "Events" and contains a welcome message: "Welcome PES1UG23CS902. Register for events below." Below this, there is a grid of nine event cards:

- Event ID: 1, ₹ 500, Hackathon, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 2, ₹ 300, Dance, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 3, ₹ 500, Hackathon, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 4, ₹ 300, Dance Battle, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 5, ₹ 400, AI Workshop, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 6, ₹ 200, Photography Walk, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 7, ₹ 350, Gaming Tournament, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 8, ₹ 250, Music Night, Includes certificate • instant registration • limited seats, Register button.
- Event ID: 9, ₹ 150, Treasure Hunt, Includes certificate • instant registration • limited seats, Register button.

Screenshot SS2

The screenshot shows a web browser window for the URL `localhost:8000/checkout`. The page title is "Monolith Failure". At the top right, there are buttons for "Login" and "Create Account". A message "One bug in one module impacted the entire application." is displayed. The main content area is titled "Monolith Failure" and contains the following sections:

- 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

At the bottom, there are "Back to Events" and "Login" buttons. The footer includes the text "CC Week X • Monolithic Applications Lab".

Below the page, a terminal window shows the following log output:

```
INFO: 127.0.0.1:49593 - "GET /events?user=PES1UG23CS902 HTTP/1.1" 200 OK
INFO: 127.0.0.1:49656 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
Traceback (most recent call last):
```

Screenshot SS3

The screenshot shows a web browser window with the URL `localhost:8000/checkout`. The page title is "Fest Monolith" and it includes links for "Login" and "Create Account". The main content area is titled "Checkout" and contains the following information:

- Total Payable**: ₹ 9500
- A green callout box with a checkmark icon says: "After fixing + optimizing checkout logic, re-run Locust and compare results."
- A sidebar titled "What you should observe" lists:
 - 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.
- A yellow callout box at the bottom right says: "Next Lab: Split this monolith into Microservices (Events / Registration / Checkout)."

At the bottom left, there is a footer note: "CC Week X • Monolithic Applications Lab".

Screenshot SS4

The screenshot shows a Locust test run interface. The top bar indicates the host is `http://localhost:8000`, status is "CLEANUP", RPS is 0.6, and Failures are 0%. The interface has tabs for STATISTICS, CHARTS, FAILURES, EXCEPTIONS, CURRENT RATIO, DOWNLOAD DATA, and LOG.

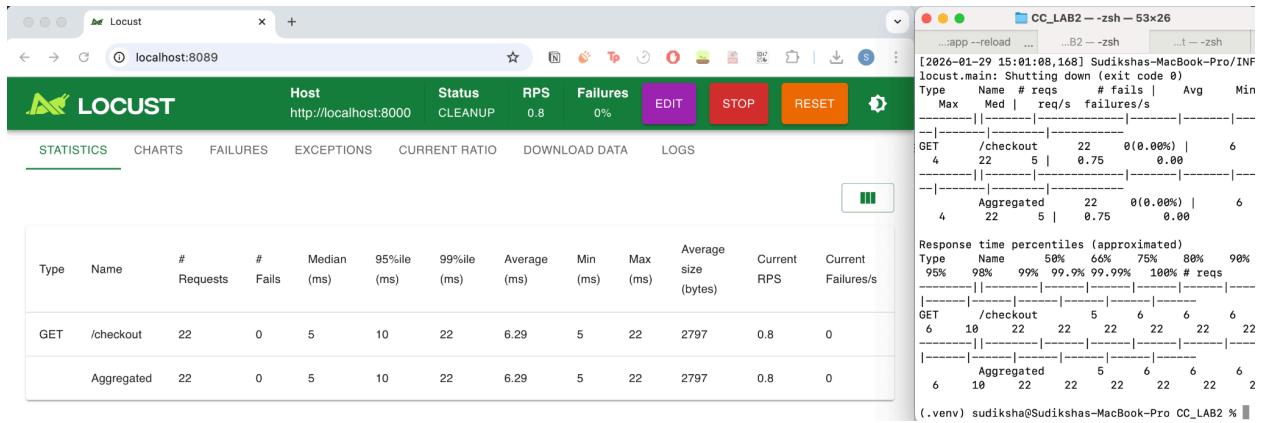
STATISTICS table (partial data):

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
GET	/checkout	19	0	6	9	9	5.82	3	9
	Aggregated	19	0	6	9	9	5.82	3	9

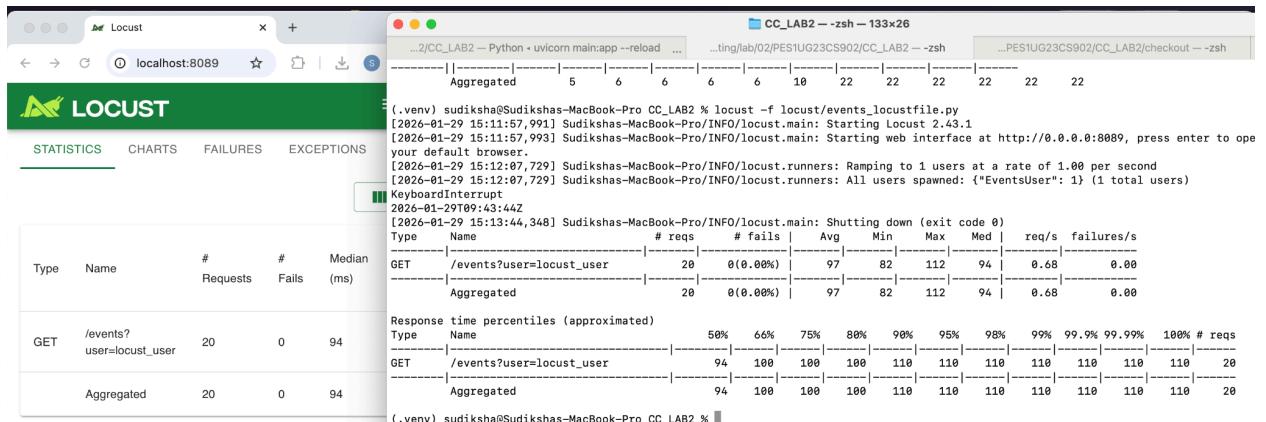
LOG section (partial log output):

```
(.venv) sudiksha@Sudikshas-MacBook-Pro CC_LAB2 % locust --zsh
...23CS902/CC_LAB2 --zsh ...C_LAB2/checkout --zsh
uvicorn main:app --reload ...
[2026-01-29 14:54:00,582] Sudikshas-MacBook-Pro/INFO/locust.main: Starting Locust 2.43.
[2026-01-29 14:54:00,583] Sudikshas-MacBook-Pro/INFO/locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.
[2026-01-29 14:54:28,897] Sudikshas-MacBook-Pro/INFO/locust.runners: Ramping to 1 users
+ 1 user of 1.00 per second
[2026-01-29 14:54:28,897] Sudikshas-MacBook-Pro/INFO/locust.runners: All users spawned: "CheckoutUser", 1 (1 total users)
KeyboardInterrupt
[2026-01-29 14:56:11,629] Sudikshas-MacBook-Pro/INFO/locust.main: Shutting down (exit code 0)
Check out User: 1 (1 total users)
[2026-01-29 14:56:11,629] Sudikshas-MacBook-Pro/INFO/locust.main: Shutting down (exit code 0)
(.venv) sudiksha@Sudikshas-MacBook-Pro CC_LAB2 % pwd
/Users/sudiksha/sudiksha/acad/sem-6/cloud-computing/lab/02/PES1UG23CS902/CC_LAB2
(.venv) sudiksha@Sudikshas-MacBook-Pro CC_LAB2 %
```

Screenshot SS5

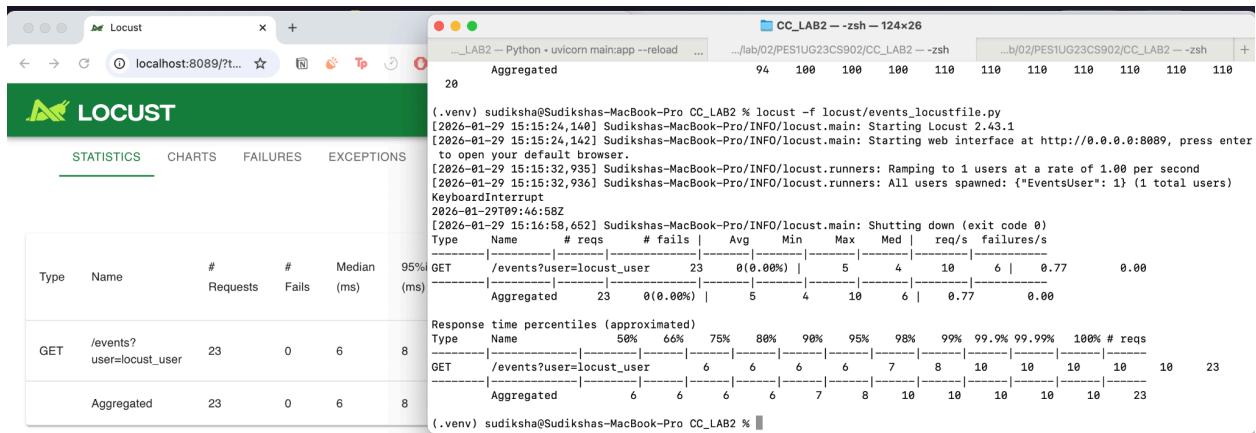


Screenshot SS6

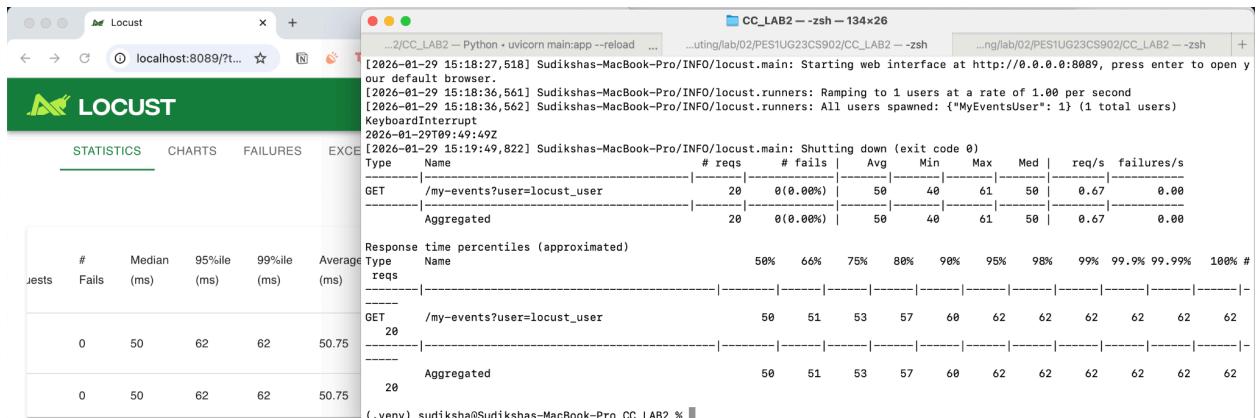


Optimisation: removed for loop that updated variable ‘waste’ over several iterations

Screenshot SS7

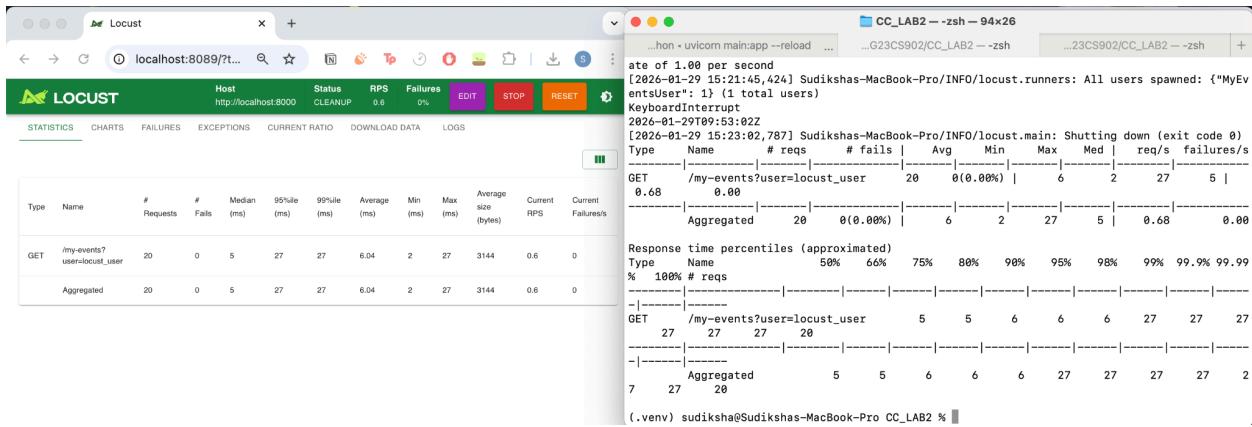


Screenshot SS8



Optimised code: removed for loop that updated variable 'dummy'

Screenshot SS9



Explanation of optimisations:

Route 1: /events

Bottleneck: The function contained a CPU-intensive for loop iterating 3,000,000 times to calculate a waste variable, which served no functional purpose.

Change: The unnecessary loop and the waste variable calculation were completely removed.

Improvement: Removing the loop eliminated the artificial CPU processing delay, allowing the server to return the template response immediately after fetching data from the database. Reduced from an Average of 97 to 23

Route 2: /my-events

Bottleneck: The function included a for loop iterating 1,500,000 times to increment a dummy variable, creating an artificial delay.

Change: The dummy variable loop was identified as dead code and removed.

Why it improved: By removing the blocking loop, the request handling time was reduced to just the database query and template rendering time, significantly lowering latency from an average of 50ms to 6ms