

CC Lab

Lab 2 - Submission

Name: Gauthamdev R Holla

SRN: PES2UG23CS197

Branch: CSE

Sem: 6

Section: C

SS1:

The screenshot shows a web application for event registration. At the top, there's a navigation bar with a logo, the text "Fest Monolith FastAPI + SQLite + Locust", and links for "Events", "My Events", "Checkout", and "Logout". The main section is titled "Events" with a sub-section header "Events". It displays six event cards arranged in two rows of three. Each card includes an event ID, price, and a "Register" button.

Event ID	Price	Event Name	Description	Action
1	₹ 500	Hackathon	Includes certificate • instant registration • limited seats	Register
2	₹ 300	Dance	Includes certificate • instant registration • limited seats	Register
3	₹ 500	Hackathon	Includes certificate • instant registration • limited seats	Register
4	₹ 300	Dance Battle	Includes certificate • instant registration • limited seats	Register
5	₹ 400	AI Workshop	Includes certificate • instant registration • limited seats	Register
6	₹ 200	Photography Walk	Includes certificate • instant registration • limited seats	Register

SS2:

The screenshot shows a web application for handling a monolith failure. At the top, there's a navigation bar with a logo, the text "Fest Monolith FastAPI + SQLite + Locust", and links for "Login" and "Create Account". The main section features a large red banner with the text "Monolith Failure" and "HTTP 500". Below the banner, there are two boxes: one for the error message and one for troubleshooting steps.

Error Message
division by zero

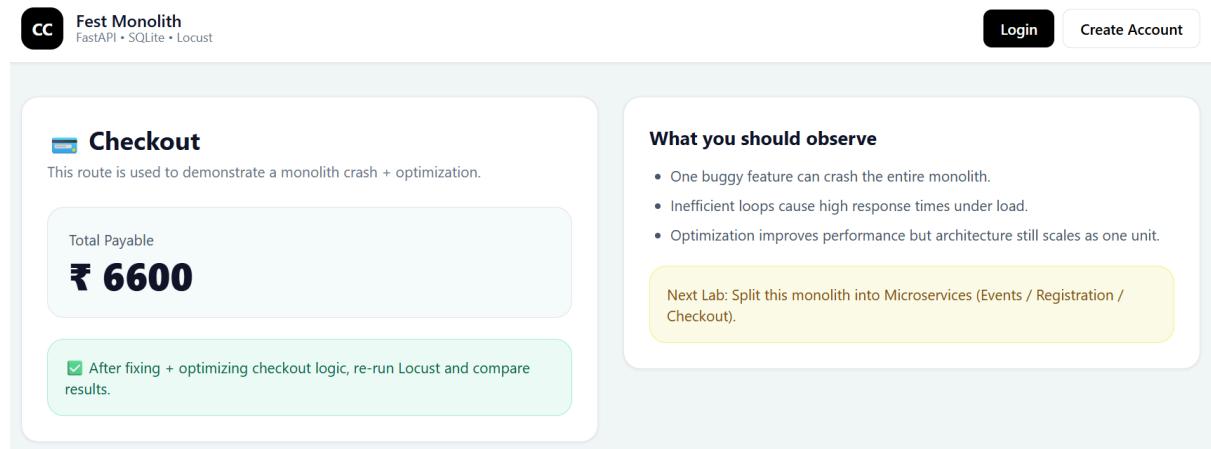
Why did this happen?
Because this is a **monolithic application**: all modules share the same runtime and deployment. When one feature crashes, it affects the whole system.

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](#)

SS3:



Checkout

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

Total Payable
₹ 6600

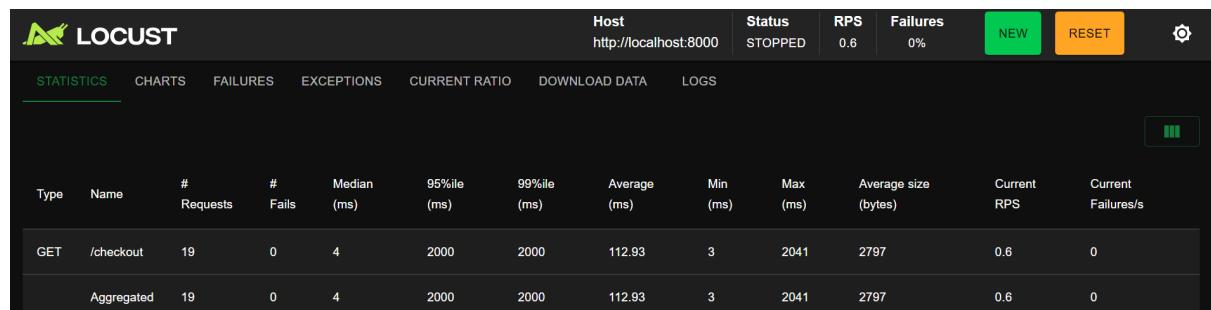
After fixing + optimizing checkout logic, re-run Locust and compare results.

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).

SS4:



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	4	2000	2000	112.93	3	2041	2797	0.6	0
	Aggregated	19	0	4	2000	2000	112.93	3	2041	2797	0.6	0

```

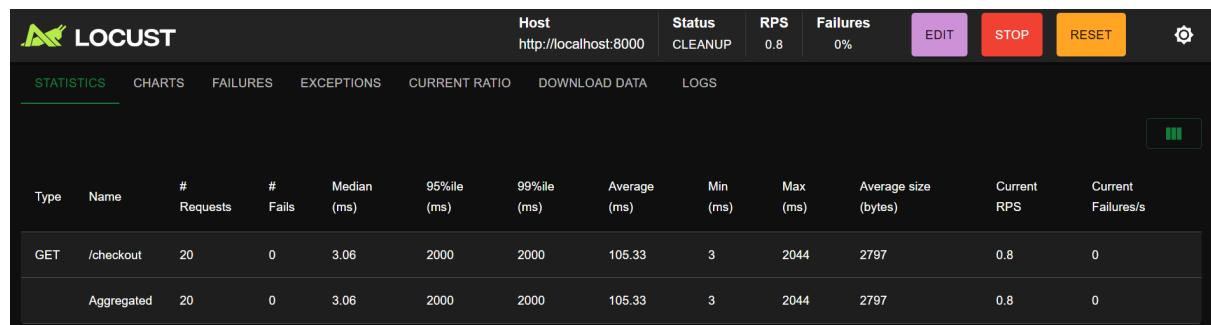
KeyboardInterrupt
2026-01-20T09:13:29Z
[2026-01-20 14:43:29,154] DESKTOP-Q06NKIK/INFO/locust.main: Shutting down (exit code 0)
Type      Name          # reqs   # fails | Avg     Min     Max     Med | req/s  failures/s
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET       /checkout      19      0(0.00%) | 112     3      2041    4 | 0.65   0.00
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
                    Aggregated  19      0(0.00%) | 112     3      2041    4 | 0.65   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	4	4	42	2000	2000	2000	2000	2000	2000	19
	Aggregated	4	4	4	4	42	2000	2000	2000	2000	2000	2000	19

SS5:



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	3.06	2000	2000	105.33	3	2044	2797	0.8	0
	Aggregated	20	0	3.06	2000	2000	105.33	3	2044	2797	0.8	0

```

KeyboardInterrupt
2026-01-20T09:25:45Z
[2026-01-20 14:55:45,483] DESKTOP-Q06NKIK/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%) | 105     3       2043    3 | 0.69     0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          20       0(0.00%) | 105     3       2043    3 | 0.69     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       3       4       4       2000   2000   2000   2000   2000   2000   20
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          3       3       3       4       4       2000   2000   2000   2000   2000   2000   20

```

SS6:

The screenshot shows the Locust web interface with the following details:

- Host:** http://localhost:8000
- Status:** CLEANUP
- RPS:** 0.6
- Failures:** 0%
- Buttons:** EDIT (purple), STOP (red), RESET (orange), and a gear icon.

Statistics Table:

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	17	0	170	2200	2200	289.05	165	2223	21138	0.6	0
Aggregated		17	0	170	2200	2200	289.05	165	2223	21138	0.6	0

```

KeyboardInterrupt
2026-01-20T09:29:26Z
[2026-01-20 14:59:26,801] DESKTOP-Q06NKIK/INFO/locust.main: Shutting down (exit code 0)
Type      Name          # reqs    # fails | Avg     Min     Max     Med | req/s   failures/s
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET      /events?user=locust_user 17       0(0.00%) | 289     165     2222    170 | 0.60     0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          17       0(0.00%) | 289     165     2222    170 | 0.60     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 170     170     170     170     170     2200   2200   2200   2200   2200   2200   17
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          170     170     170     170     170     2200   2200   2200   2200   2200   2200   17

```

SS7:

The screenshot shows the Locust web interface with the following details:

- Host:** http://localhost:8000
- Status:** RUNNING
- Users:** 1
- RPS:** 0.7
- Failures:** 0%
- Buttons:** EDIT (purple), STOP (red), RESET (orange), and a gear icon.

Statistics Table:

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	10	0	3.1	2000	2000	206.04	3	2029	21138	0.7	0
Aggregated		10	0	3.1	2000	2000	206.04	3	2029	21138	0.7	0

```

KeyboardInterrupt
2026-01-20T09:41:33Z
[2026-01-20 15:11:33,960] DESKTOP-Q06NKIK/INFO/locust.main: shutting down (exit code 0)
Type      Name          # reqs  # fails | Avg   Min   Max   Med | req/s failures/s
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET     /events?user=locust_user    11  0(0.00%) | 187   3   2028   3 | 0.64   0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          11  0(0.00%) | 187   3   2028   3 | 0.64   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    3    4    4    4    4    2000  2000  2000  2000  2000  2000  11
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          3    4    4    4    4    2000  2000  2000  2000  2000  2000  11

```

SS8:

The Locust interface shows the following configuration and results for SS8:

Host: http://localhost:8000

Status: RUNNING

Users: 1

RPS: 0.5

Failures: 0%

Test Details:

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	6	0	62	2100	2100	402.08	61	2103	3144	0.5	0
	Aggregated	6	0	62	2100	2100	402.08	61	2103	3144	0.5	0

```

KeyboardInterrupt
2026-01-20T09:44:56Z
[2026-01-20 15:14:56,892] DESKTOP-Q06NKIK/INFO/locust.main: Shutting down (exit code 0)
Type      Name          # reqs  # fails | Avg   Min   Max   Med | req/s failures/s
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET     /my-events?user=locust_user    7  0(0.00%) | 353   61   2103   62 | 0.65   0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          7  0(0.00%) | 353   61   2103   62 | 0.65   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    62   63   63   63   2100  2100  2100  2100  2100  2100  7
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          62   63   63   63   2100  2100  2100  2100  2100  2100  7

```

SS9:

The Locust interface shows the following configuration and results for SS9:

Host: http://localhost:8000

Status: RUNNING

Users: 1

RPS: 0.6

Failures: 0%

Test Details:

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	8	0	3	2000	2000	257.83	3	2040	3144	0.6	0
	Aggregated	8	0	3	2000	2000	257.83	3	2040	3144	0.6	0

```

KeyboardInterrupt
2026-01-20T09:46:45Z
[2026-01-20 15:16:45,254] DESKTOP-Q06NKIK/INFO/locust.main: Shutting down (exit code 0)
• Type      Name          # reqs  # fails | Avg   Min   Max   Med | req/s failures/s
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET     /my-events?user=locust_user    9  0(0.00%) | 229   2   2039   3 | 0.58   0.00
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          9  0(0.00%) | 229   2   2039   3 | 0.58   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    3    3    3    4    2000  2000  2000  2000  2000  2000  9
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated          3    3    3    4    2000  2000  2000  2000  2000  2000  9

```

Questions:

1) What was the bottleneck?

Unnecessary loops that were wasting memory.

2) What change did you make?

Removed the unnecessary loops.

3) Why did the performance improve?

Loops that were wasting memory were removed.
