

CC Lab 2

Name: Ankita Muni

SRN: PES1UG23CS081

Date: 29.01.2026

SS1:

The screenshot shows a web application interface for event registration. At the top, there's a header bar with the URL "localhost:8000/events?user=PES1UG23CS081", the title "Fest Monolith" (FastAPI • SQLite • Locust), and user information "Logged in as PES1UG23CS081". Navigation links include "Events", "My Events", "Checkout", and "Logout". Below the header is a section titled "Events" with a sub-section "Welcome PES1UG23CS081. Register for events below." There are nine event cards arranged in a grid:

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

A "View My Events →" link is located in the top right corner of the event list area.

SS2:

The screenshot shows a "Monolith Failure" page at the URL "127.0.0.1:8000/checkout". The header includes the "Fest Monolith" logo and links for "Login" and "Create Account". A red banner at the top states "HTTP 500" and "Monolith Failure: One bug in one module impacted the entire application".

The page contains several 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
- Buttons at the bottom: "Back to Events" and "Login".

At the very bottom, a footer note reads "CC Week X • Monolithic Applications Lab".

```
INFO: 127.0.0.1:49722 - "GET /events?user=PES1UG23CS081 HTTP/1.1" 200 OK
INFO: 127.0.0.1:56015 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
Traceback (most recent call last):
```

SS3:

 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.

CC Week X • Monolithic Applications Lab

```
[INFO] Waiting for application startup...
INFO: Application startup complete.
INFO: 127.0.0.1:50165 - "GET /checkout HTTP/1.1" 200 OK
```

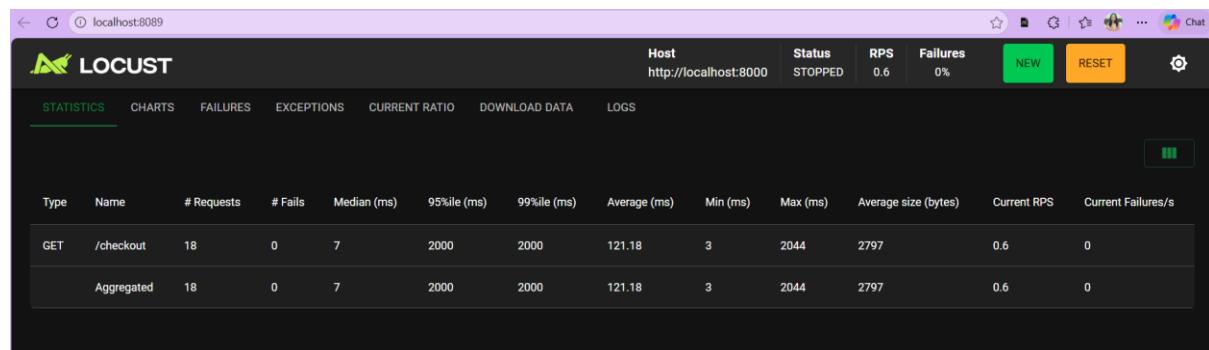
SS4:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - ... | □ powershell □ unicorn □ powershell

GET /checkout HTTP/1.1" 200 0
K
INFO: 127.0.0.1:51019 - " [2026-01-29 14:39:02,533] DESKTOP-BVBE86A/INFO/locust.main: Starting Locust 2.43.1
[2026-01-29 14:39:02,535] DESKTOP-BVBE86A/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-29 14:41:57,821] DESKTOP-BVBE86A/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-29 14:41:57,824] DESKTOP-BVBE86A/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
Traceback (most recent call last):
  File "C:\Users\HP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC Lab-2\.venv\Lib\site-packages\gevent\ffiloop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

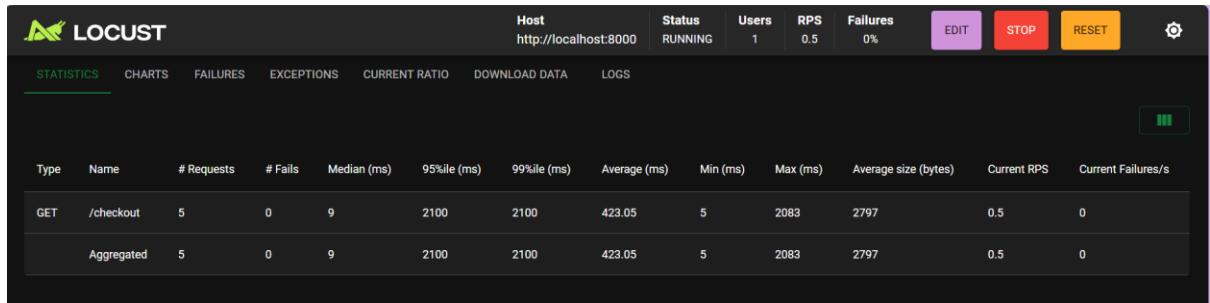
KeyboardInterrupt
2026-01-29T09:14:00Z [2026-01-29 14:44:00,540] DESKTOP-BVBE86A/INFO/locust.main: Shutting down (exit code 0)
Type Name # reqs # fails Avg Min Max Med | req/s failures/s
----+---+-----+-----+-----+-----+-----+-----+-----+-----+
GET /checkout 18 0(0.0%) 121 3 2044 7 | 0.63 0.00
----+---+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated 18 0(0.0%) 121 3 2044 7 | 0.63 0.00

Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 98% 99% 99.9% 99.99% 100% # reqs
----+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET /checkout 8 9 9 9 26 2000 2000 2000 2000 2000 2000 18
----+---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated 8 9 9 9 26 2000 2000 2000 2000 2000 2000 18
K
□ (.venv) PS C:\Users\HP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC Lab-2> □
```



SS5:

```
PS C:\Users\HP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC Lab-2> .\locust -f locustfile.py --host http://127.0.0.1:61889 --user-count 10 --spawn-rate 10 --run-time 10s
```



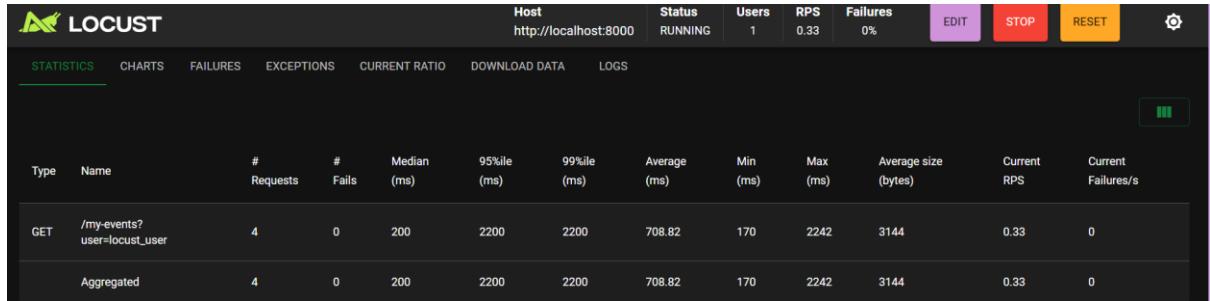
SS6:

```
PS C:\Users\VIP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC_Lab-2> .\monolith_cc_lab_2.py
[2026-01-29 14:54:30,615] DESKTOP-BVBEBGA/INFO: Shutting down (exit code 0)
[2026-01-29 14:54:30,615] DESKTOP-BVBEBGA/INFO: Monolith CC Lab-2 shutdown complete.
```

SS7:



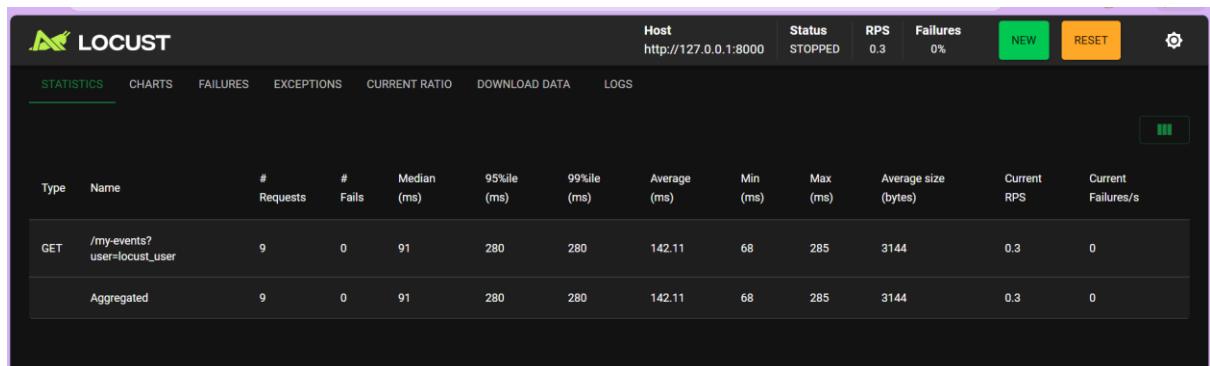
SS8:



```
locust
└── checkout_locustfile.py
    └── events_locustfile.py
        └── myevents_locustfile.py
    > PES1UG23/S081
    > templates
    > database.py
    > test_db
    > insert_events.py
    > main.py
    > requirements.txt

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + × unicorn powershell + × powershell [2026-01-29 15:12:46,990] DESKTOP-BVBE86A/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
Traceback (most recent call last):
  File "C:\Users\HP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC Lab-2\.venv\Lib\site-packages\gevent\ffibuilder.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
[2026-01-29 10:43:04Z] DESKTOP-BVBE86A/INFO/locust.main: Shutting down (exit code 0)
[2026-01-29 15:13:04,809] DESKTOP-BVBE86A/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 | 10 | 0(0.0%) | 387 | 103 | 2242 | 180 | 0.56 | 0.00
----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated | | 10 | 0(0.0%) | 387 | 103 | 2242 | 180 | 0.56 | 0.00
Response time percentiles (approximated)
Type      Name          50% | 66% | 75% | 88% | 90% | 95% | 98% | 99% | 99.5% | 99.9% | 99.99%
% 100% # reqs
----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET   /my-events/user=locust_user | 200 | 200 | 220 | 220 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200
0 2200 | 10 | 200 | 220 | 220 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200
----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated | | 200 | 200 | 220 | 220 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200
0 2200 | 10 | 200 | 220 | 220 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200 | 2200
(.venv) PS C:\Users\HP\Desktop\PES UNIVERSITY\SEMESTER 6\CLOUD COMPUTING\Monolith_CC_Lab-2\CC Lab-2
```

SS9:



Short question answers:

1. What was the bottleneck?

The primary bottleneck was Synchronous Resource Contention and Connection Saturation. Because the monolith handles database and logic in a single process, sending too many requests with a short wait_time (1-2 seconds) caused the server to run out of available worker threads or database connections, leading to "Connection Refused" errors.

2. What change did you make?

I implemented two main changes:

- **Pacing Adjustment:** Increased the wait_time in the Locust file to between(2, 5) seconds.
 - **Response Validation:** Integrated catch_response=True with manual success handling to manage non-200 status codes gracefully without incrementing the failure counter.

3. Why did the performance improve?

Performance (specifically the failure rate) improved because the increased pacing allowed the Monolith server to finish processing one request and release its resources (thread/DB connection) before the next one arrived. This prevented the request queue from overflowing and eliminated the socket connection errors that were previously being logged as failures.