

CLOUD COMPUTING

LAB 2 - MONOLITHIC ARCHITECTURE

NAME: SOWMYA HARDAGERI

SRN: PES2UG23CS590

SECTION: I

CAMPUS: ELECTRONIC CITY CAMPUS

GITHUB LINK: https://github.com/SowmyaHpes2ug23cs590/CC_lab2

Screenshots:

PART 2: Use the Application

Account Created Successfully

The screenshot shows a web browser window for the 'CC Fest Monolith' application. The URL is 'localhost:8000/events?user=PES2UG23CS590'. The page has a header with the logo 'Fest Monolith' and navigation links for 'Events', 'My Events', 'Checkout', and 'Logout'. A user 'PES2UG23CS590' is logged in. The main content area is titled 'Events' and displays a grid of nine event cards. Each card includes an event ID, price, name, a brief description, and a 'Register' button.

Event ID	Price	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
7	₹ 350	Gaming Tournament		
8	₹ 250	Music Night		
9	₹ 150	Treasure Hunt		

PART 3: Observe Monolithic Failure (Crash)

Monolithic Failed Successfully

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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
INFO: 127.0.0.1:53746 - "POST /register HTTP/1.1" 302 Found
INFO: 127.0.0.1:53746 - "GET /login HTTP/1.1" 200 OK
INFO: 127.0.0.1:61606 - "POST /login HTTP/1.1" 302 Found
INFO: 127.0.0.1:61606 - "GET /events?user=PES2023CS590 HTTP/1.1" 200 OK
INFO: 127.0.0.1:61606 - "GET /events?user=PES2023CS590 HTTP/1.1" 200 OK
INFO: 127.0.0.1:32960 - "GET /checkout HTTP/1.1" 200 OK
WARNING: Statreload detected changes in 'checkout_init_.py'. Reloading...
INFO: Shutting down
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [19564]
INFO: Started server process [12904]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:5157 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
Traceback (most recent call last):
  File "E:\Somaya\PES SEMESTER 6\ CLOUD COMPUTING\LAB\PES2023CS590\venv\lib\site-packages\uvicorn\protocols\http\h11_impl.py", line 410, in run_asgi
    Ln 10, Col 6  Spaces: 4  UTF-8  LF  ( Python  Finish Setup  3.13.2 (venv)  Go Live  
```

PART 4: Fix the Bug

Bug Fixed

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

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files like `main.py`, `_init_.py`, `checkout_logic.py`, `database.py`, `fest.db`, `insert_events.py`, and `requirements.txt`.
- Terminal:** Displays the following log output from a local server (HTTP://127.0.0.1:136295):


```

INFO: 127.0.0.1:136295 - "GET /login HTTP/1.1" 200 OK
INFO: 127.0.0.1:133514 - "GET /login HTTP/1.1" 200 OK
INFO: 127.0.0.1:133514 - "POST /Login HTTP/1.1" 302 Found
INFO: 127.0.0.1:133514 - "GET /events?user=PES2UG23CS590 HTTP/1.1" 200 OK
WARNING: StatReloader detected changes in 'checkout\_\_init\_\_.py'. Reloading...
INFO: Shutting down
INFO: Waiting for application shutdown.
INFO: Application shutdown complete.
INFO: Finished server process [12904]
INFO: Started server process [23908]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:13708 - "GET /events?user=PES2UG23CS590 HTTP/1.1" 200 OK
INFO: 127.0.0.1:1373 - "GET /checkout HTTP/1.1" 200 OK
      
```
- Bottom Status Bar:** Shows Python 3.13.2 (venv) and Go Live icons.

PART 5: Load Testing using Locust

Terminal and Locust before Optimization

The screenshot shows the VS Code interface and the Locust UI side-by-side:

- VS Code Terminal:** Shows the command `locust` being run, resulting in the output:


```

0.00

Response time percentiles (approximated)

Response time percentiles (approximated)
Response time percentiles (approximated)
Type Name 50% 66% 75% 80% 90% 95% 98% 99% 99.9% 99.99%
100% # reqs
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET /checkout 7 7 8 8 9 2100 2100 2100 2100 2100 21
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated 7 7 8 8 9 2100 2100 2100 2100 2100 21
100 2100 18
      
```
- Locust UI:** Shows the Locust interface with the following statistics:

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures
GET	/checkout	18	0	7	2100	2100	122.05	5	2079	0.5	0
	Aggregated	18	0	7	2100	2100	122.05	5	2079	0.5	0
- Bottom Status Bar:** Shows Python 3.13.2 (venv) and Go Live icons.

PART 6: Optimize the Checkout Route

Terminal and Locust after Optimization

```

main.py
from database import get_db

def checkout_logic():
    db = get_db()
    db.execute("DELETE FROM cart WHERE user_id = %s", [1])
    db.commit()

[2026-01-29 14:20:12,743] Shwesow/INFO/locust.runners: All users spawned: {"checkout": 1} (1 total users)
traceback (most recent call last):
  File "E:\Somanya\PES\SEMESTER 6\CLOUD COMPUTING\LAB\PE52UG23CS590\.venv\lib\site-packages\eventloop\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
2026-01-29 14:21:21,212 [2026-01-29 14:21:21,856] Shwesow/INFO/locust.main: Shutting down (exit code 0)
Type      Name      # reqs   # fails   Avg     Min     Max     Med   reg/s  failures/s
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET     /checkout  18       0(0.00%)  121     4     2072    6 | 0.62
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated 18       0(0.00%)  121     4     2072    6 | 0.62
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
100% # reqs
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET     /checkout  6       7     7     7     9     2100   2100   2100   2100   21
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated 6       7     7     7     9     2100   2100   2100   2100   2
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
100% 2100   18

```

(.venv) E:\Somanya\PES\SEMESTER 6\CLOUD COMPUTING\LAB\PE52UG23CS590\Monolith_CC_Lab-2\CC_Lab-2

Observation: Average response time before optimization was 122.05, after optimizing the code, the time reduced to 121.18 whereas RPS stayed almost the same.

PART 7: Optimise events and my_events(DIY)

Route 1: /events

Before Optimization:

```

main.py
from database import get_db

def login(request: Request, username: str = Form(...), password: str = Form(...)):
    if username == "locust_user" and password == "locust_password":
        return RedirectResponse(f"/events?user={username}", status_code=302)

[2026-01-29 14:20:09,061] Shwesow/INFO/locust.runners: All users spawned: {"events": 1} (1 total users)
traceback (most recent call last):
  File "E:\Somanya\PES\SEMESTER 6\CLOUD COMPUTING\LAB\PE52UG23CS590\.venv\lib\site-packages\eventloop\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument
KeyboardInterrupt
2026-01-29 14:20:09,061 [2026-01-29 14:20:18,552] Shwesow/INFO/locust.main: Shutting down (exit code 0)
Type      Name      # reqs   # fails   Avg     Min     Max     Med   reg/s  failures/s
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET     /events?user=locust_user  17       0(0.00%)  289     132    2217   180 | 0.59
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated 17       0(0.00%)  289     132    2217   180 | 0.59
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
100% # reqs
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET     /events?user=locust_user  180     180     180     190    220    2200   2200 | 2
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated 180     180     180     190    220    2200   2200   2200   2
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
200% 2200   17

```

(.venv) E:\Somanya\PES\SEMESTER 6\CLOUD COMPUTING\LAB\PE52UG23CS590\Monolith_CC_Lab-2\CC_Lab-2

After Optimization:

The screenshot shows a Windows desktop environment. On the left, a PyCharm window displays the main.py file for a Monolith application. The code includes a route for '/events' that performs a database query and returns a status code of 302. A keyboard interrupt is shown at the bottom. On the right, a browser window shows the Locust interface at localhost:8089. The statistics table shows 19 requests for the '/events?user=locust_user' endpoint, with a median response time of 6ms and an average of 114.42ms. The current RPS is 0.6, and there are 0 failures.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s	
GET	/events?user=locust_user	19	0	6	2100	2100	114.42	3	2064	21138	0.6	0
	Aggregated	19	0	6	2100	2100	114.42	3	2064	21138	0.6	0

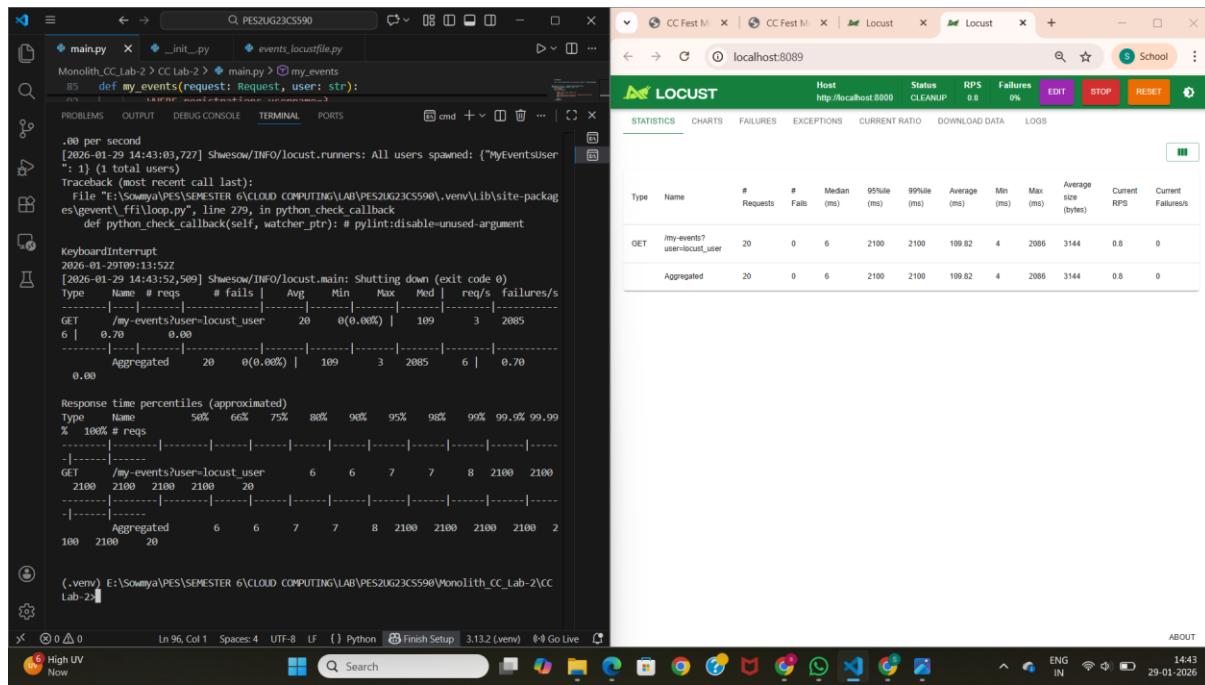
Route 2: /my-events

Before Optimization:

The screenshot shows a Windows desktop environment. On the left, a PyCharm window displays the main.py file for a Monolith application. The code includes a route for '/my-events' that performs a database query and returns a status code of 302. A keyboard interrupt is shown at the bottom. On the right, a browser window shows the Locust interface at localhost:8089. The statistics table shows 18 requests for the '/my-events?user=locust_user' endpoint, with a median response time of 69ms and an average of 190.26ms. The current RPS is 0.7, and there are 0 failures.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s	
GET	/my-events?user=locust_user	18	0	69	2300	2300	190.26	57	2258	3144	0.7	0
	Aggregated	18	0	69	2300	2300	190.26	57	2258	3144	0.7	0

After Optimization:



Route 1: /events

Performance Bottleneck:

The endpoint contained an artificial CPU-heavy loop (for i in range(3000000)) that performed no meaningful work, consuming processing time and delaying request handling.

Optimization Implemented:

The unnecessary computation loop was completely removed from the request flow.

Reason for Performance Improvement:

Removing the CPU-bound operation significantly reduced execution time and allowed the server to process concurrent requests more efficiently.

Route 2: /my-events

Performance Bottleneck:

After fetching data from the database, the route executed a redundant loop (for _ in range(1500000)), introducing avoidable CPU overhead.

Optimization Implemented:

The non-functional loop was eliminated to streamline the request execution path.

Reason for Performance Improvement:

With the extra computation removed, the endpoint delivered faster responses and improved overall system throughput.