


NAME : SAIHRITESH
SRN : PES2UG23CS511

```
[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: pip install --upgrade pip
(ccvenv) saihritesh@apples-MacBook-Air-2 CC_Lab-2 % python3 insert_events.py
✓ Events inserted successfully!
(ccvenv) saihritesh@apples-MacBook-Air-2 CC_Lab-2 % uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['/Users/saihritesh/Content/mac_bkup_6oct/College/SEM-5/Cloud Computing /PES2UG23CS511/CC_Lab-2']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [37116] using StatReload
INFO: Started server process [37118]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

SS1

**Fast Monolith**
FastAPI • SQLite • Locust


Logged in as PES2UG23CS511

Events

My Events

Checkout

Logout

**Events**

View My Events →

Welcome PES2UG23CS511. Register for events below.

Event ID: 1

₹ 500

Hackathon

Includes certificate • instant registration • limited seats

Register

Event ID: 2

₹ 300

Dance

Includes certificate • instant registration • limited seats

Register

Event ID: 3

₹ 500

Hackathon

Includes certificate • instant registration • limited seats

Register

Event ID: 4

₹ 300

Dance Battle

Includes certificate • instant registration • limited seats

Register

Event ID: 5

₹ 400

AI Workshop

Includes certificate • instant registration • limited seats

Register

Event ID: 6

₹ 200

Photography Walk

Includes certificate • instant registration • limited seats

Register

Event ID: 7

₹ 350

Gaming Tournament

Includes certificate • instant registration • limited seats

Register

Event ID: 8

₹ 250

Music Night

Includes certificate • instant registration • limited seats

Register

Event ID: 9

₹ 150

Treasure Hunt

Includes certificate • instant registration • limited seats

Register

SS2

Monolith Failure

HTTP 500

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

```
INFO: 127.0.0.1:52174 - "GET /events?user=PES2UG23CS511 HTTP/1.1" 200 OK
INFO: 127.0.0.1:52176 - "GET /register_event/404?user=PES2UG23CS511 HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
```

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).

CC Week X • Monolithic Applications Lab

SS4



Host
http://localhost:8000


Status
STOPPED

RPS
0.7

Failures
0%

NEW

RESET



STATISTICS

CHARTS


FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS



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	6	6	3.48	2	6	2797	0.7	0
	Aggregated	20	0	3	6	6	3.48	2	6	2797	0.7	0

The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom.

File Explorer: The file explorer on the left shows a project structure for 'PES2UG23CS511'. The files are:

- CC_Lab_2
 - __pycache__
 - checkout
 - __pycache__
 - __init__.py
 - event.py
 - locust
 - templates
 - database.py
 - test.db
 - insert_events.py
 - main.py
 - requirements.txt
- Monolith_CC_Lab-2.zip

Code Editor: The code editor shows the content of 'main.py':


```

1  from database import get_db
2
3  def checkout_logic():
4      db = get_db()
5      db.row_factory = None
6
7      events = db.execute("SELECT fee FROM events").fetchall()
8
9      # Uncomment this line initially for the crash screenshot task
10     # 1 / 0
11
12     total = 0
13     for e in events:
14         total += e[0]
15
16     return total
17
    
```

Terminal: The terminal shows the output of a test run. It includes a timestamp, a message about shutting down, and a table of metrics for GET requests.

2025-01-20T09:58:29Z
 [2025-01-20 15:20:29,430] apples-MacBook-Air-2/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%)	3	2	5	3	0.68	0.00
	Aggregated	20	0(0.00%)	3	2	5	3	0.68	0.00

Response time percentiles (approximated)

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# r
GET	/checkout	3	5	5	5	5	6	6	6	6	6	6	
	Aggregated	3	5	5	5	5	6	6	6	6	6	6	

SS5

The image shows a VS Code editor interface. On the left is the file explorer showing a project named 'Monolith_CC_Lab-2.zip' with files like 'CC_Lab_2', 'requirements.txt', and 'main.py'. The center pane shows the 'main.py' file with a 'checkout' endpoint implementation. The bottom pane shows a terminal with the command 'locust -f locust/checkout_locustfile.py' and its output, including a performance summary table.

```

1  from database import get_db
2
3  def checkout_logic():
4      db = get_db()
5      db.row_factory = None
6
7      events = db.execute("SELECT fee FROM events").fetchall()
8
9      # Uncomment this line initially for the crash screenshot task
10     # 1 / 0
11
12     total = 0
13     for e in events:
14         total += e[0]
15
16     return total
17

```

Terminal Output:

```

sahitshes@apples-MacBook-Air-2 CC_Lab_2 % locust -f locust/checkout_locustfile.py
[2026-01-20 15:24:50,390] apples-MacBook-Air-2/INFO/locust.main: Starting locust 2.43.1
[2026-01-20 15:24:52,342] apples-MacBook-Air-2/INFO/locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.
[2026-01-20 15:25:06,072] apples-MacBook-Air-2/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:25:06,072] apples-MacBook-Air-2/INFO/locust.runners: All users spawned: {"checkoutuser": 1} (1 total users)
KeyboardInterrupt
2026-01-20T05:57:52Z
[2026-01-20 15:27:52,372] apples-MacBook-Air-2/INFO/locust.main: Shutting down (exit code 0)

Type      Name      # reqs      # fails      Avg      Min      Max      Med      req/s      failures/s
GET        /checkout  22          0(0.00%)     2        2        5        3        0.76       0.00
Aggregated                22          0(0.00%)     2        2        5        3        0.76       0.00

Response time percentiles (approximated)
Type      Name      50%      66%      75%      80%      90%      95%      98%      99%      99.9%      99.99%      100% # r
GET        /checkout  3        3        3        3        5        5        5        5        5        5        5
22
Aggregated  3        3        3        3        5        5        5        5        5        5        5

```

SS6

EXPLORER

PES2U023C5611

CC_Lab_2

> __pycache__

> checkout

> cvenv

> locust

locust

checkout_locustfile.py

events_locustfile.py

myevents_locustfile.py

templates

database.py

test.db

insert_events.py

main.py

requirements.txt

Monolith_CC_Lab-2.zip

main.py 3 X

__init__.py

CC_Lab_2 > main.py > register_event

84

return RedirectResponse(f"/my-events?user={user}", status_code=302)

85

86

87 @app.get("/my-events", response_class=HTMLResponse)

88 def my_events(request: Request, user: str):

89 db = get_db()

90 rows = db.execute(

91 """

92 SELECT events.name, events.fee

93 FROM events

94 JOIN registrations ON events.id = registrations.event_id

95 WHERE registrations.username=?

96 """

97 (user,)

98).fetchall()

99

100

101

102 dummy = 0

103 for _ in range(1500000):

104 dummy += 1

105

106 return templates.TemplateResponse(

107 "my_events.html",

108 {"request": request, "events": rows, "user": user}

109)

110

111 @app.get("/checkout", response_class=HTMLResponse)

112 def checkout(request: Request):

113 total = checkout_logic()

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

saihrithesh@Apples-MacBook-Air-2 CC_Lab_2 % locust -f locust/events_locustfile.py

[2026-01-20 15:38:31,943] apples-MacBook-Air-2/INFO/locust.main: Starting Locust 2.43.1

[2026-01-20 15:38:31,946] apples-MacBook-Air-2/INFO/locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.

[2026-01-20 15:39:30,191] apples-MacBook-Air-2/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second

[2026-01-20 15:39:30,192] apples-MacBook-Air-2/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)

KeyboardInterrupt

[2026-01-20 15:40:10,422]

[2026-01-20 15:40:42,836] apples-MacBook-Air-2/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	18	0(0.00%)	125	118	184	120	0.63	0.00
	Aggregated	18	0(0.00%)	125	118	184	120	0.63	0.00

Response time percentiles (approximated)

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# r
GET	/events?user=locust_user	120	120	120	130	130	180	180	180	180	180	180	
	Aggregated	120	120	120	130	130	180	180	180	180	180	180	

SS7

LOCUST

Host

http://localhost:8000

Status

STOPPED

RPS

0.7

Failures

0%

NEW

RESET

STATISTICS

CHARTS

FAILURES

EXCEPTIONS

CURRENT RATIO

DOWNLOAD DATA

LOGS

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	21	0	3	5	9	3.69	3	9	21138	0.7	0
	Aggregated	21	0	3	5	9	3.69	3	9	21138	0.7	0

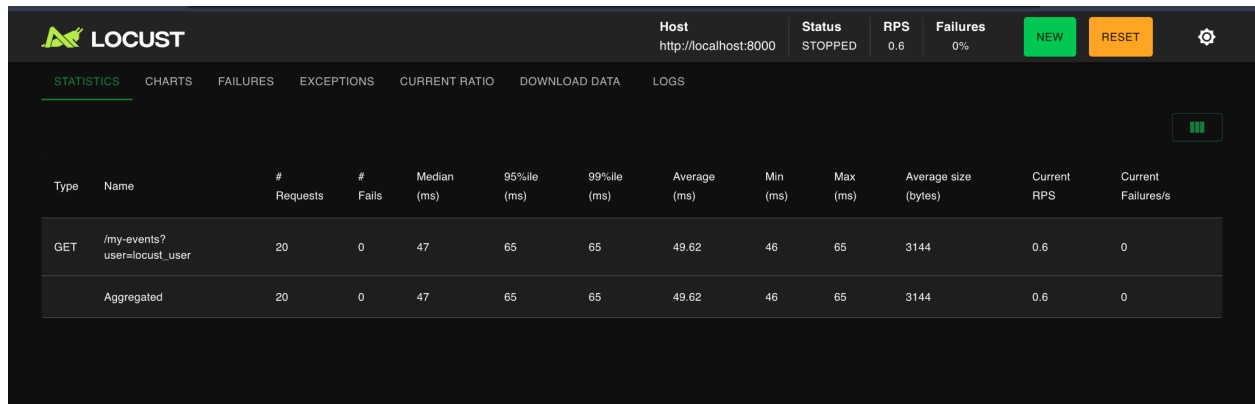
[2026-01-20 15:42:54,942] apples-MacBook-Air-2/INFO/locust.main: Shutting down (exit code 0)

Type	Name	# reqs	# fails	Avg	Min	Max	Med	req/s	failures/s
GET	/events?user=locust_user	21	0(0.00%)	3	2	9	3	0.71	0.00
	Aggregated	21	0(0.00%)	3	2	9	3	0.71	0.00

Response time percentiles (approximated)

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# r
GET	/events?user=locust_user	3	3	4	5	5	5	9	9	9	9	9	
	Aggregated	3	3	4	5	5	5	9	9	9	9	9	

SS8



The screenshot shows a code editor with the following content:

```
CC_Lab_2 > main.py > my_events
73 def register_event(event_id: int, user: str):
81     return RedirectResponse(f"/my-events?user={user}", status_code=302)
82
83
84 @app.get("/my-events", response_class=HTMLResponse)
85 def my_events(request: Request, user: str):
86     db = get_db()
87     rows = db.execute(
88         """
89         SELECT events.name, events.fee
90         FROM events
91         JOIN registrations ON events.id = registrations.event_id
92         WHERE registrations.username=?
93         """,
94         (user,)
95     ).fetchall()
96
97     dummy = 0
98     for _ in range(1500000):
99         dummy += 1
100
101     return templates.TemplateResponse(
102         "my_events.html",
103         {"request": request, "events": rows, "user": user}
104     )
105
106
107
108 @app.get("/checkout", response_class=HTMLResponse)
109 def checkout(request: Request):
```

The terminal output shows the following commands and results:

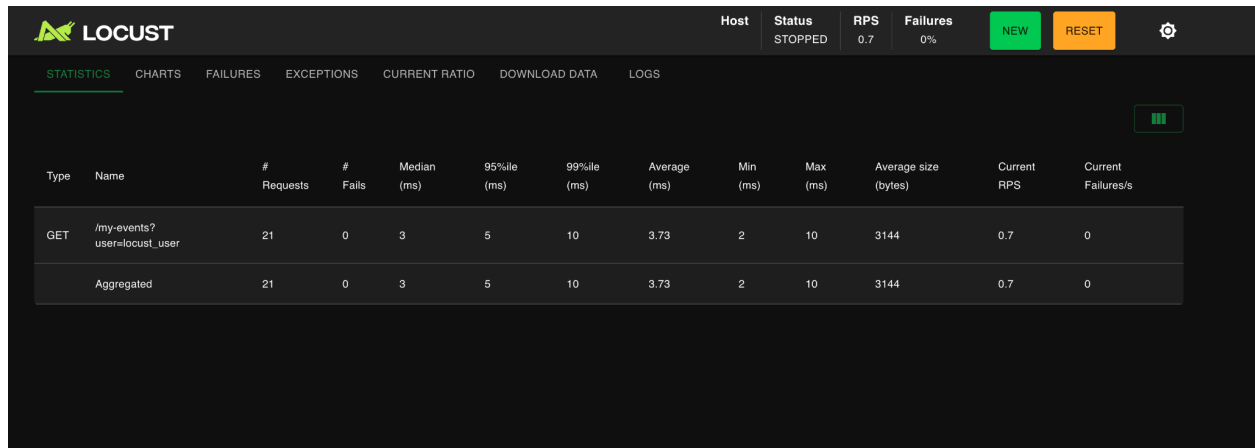
```
saihrithes@apples-MacBook-Air-2 CC_Lab_2 % locust -f locust/myevents_locustfile.py
[2026-01-20 15:45:41,256] apples-MacBook-Air-2/INFO/locust.main: Starting Locust 2.43.1
[2026-01-20 15:45:41,258] apples-MacBook-Air-2/INFO/locust.main: Starting web interface at http://0.0.0.0:8089, press enter to open your default browser.
[2026-01-20 15:45:52,256] apples-MacBook-Air-2/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-20 15:45:52,257] apples-MacBook-Air-2/INFO/locust.runners: All users spawned: ("MyEventsUser": 1) (1 total users)
KeyboardInterrupt
[2026-01-20 15:46:44,236] apples-MacBook-Air-2/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	20	0(0.00%)	49	45	65	47	0.67	0.00
Aggregated		20	0(0.00%)	49	45	65	47	0.67	0.00

Response time percentiles (approximated)

Type	Name	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# r
GET	/my-events?user=locust_user	48	50	51	51	58	65	65	65	65	65	65	20
Aggregated		48	50	51	51	58	65	65	65	65	65	65	20

SS9



```

EXPLORER
PES2UG23CS511
  CC_Lab_2
    __pycache__
    checkout
    cvenv
    locust
      checkout_locustfile.py
      events_locustfile.py
      myevents_locustfile.py
      templates
      database.py
      fest.db
      insert_events.py
      main.py
    requirements.txt
    CC_Lab_2.pdf
    Monolith_CC_Lab-2.zip

main.py 3
73 def register_event(event_id: int, user: str):
74     return RedirectResponse(f"/my-events?user={user}", status_code=302)
75
76 @app.get("/my-events", response_class=HTMLResponse)
77 def my_events(request: Request, user: str):
78     db = get_db()
79     rows = db.execute(
80         """
81         SELECT events.name, events.fee
82         FROM events
83         JOIN registrations ON events.id = registrations.event_id
84         WHERE registrations.username=?
85         """,
86         (user,)
87     ).fetchall()
88
89     return templates.TemplateResponse(
90         "my_events.html",
91         {"request": request, "events": rows, "user": user}
92     )
93
94 @app.get("/checkout", response_class=HTMLResponse)
95 def checkout(request: Request):
96     total = checkout_logic()
97     return templates.TemplateResponse(
98         "checkout.html",
99         {"request": request, "total": total}
100     )
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

◆ Route: /events

What was the bottleneck?

The `/events` route contained an unnecessary CPU-intensive loop that iterated millions of times. This loop did not contribute to any business logic and significantly increased response time.

What change did you make?

The artificial delay loop was removed, allowing the route to only perform the required database query and template rendering.

Why did performance improve?

Removing the unnecessary computation reduced CPU usage, resulting in faster response times and higher throughput under load.

◆ Route: /my-events**What was the bottleneck?**

The `/my-events` route contained a dummy loop that performed a large number of iterations without doing meaningful work, causing increased latency.

What change did you make?

The dummy loop was removed so that the route executes only the database query and response rendering.

Why did performance improve?

Eliminating the artificial delay reduced execution time per request, allowing the server to handle more concurrent users efficiently.