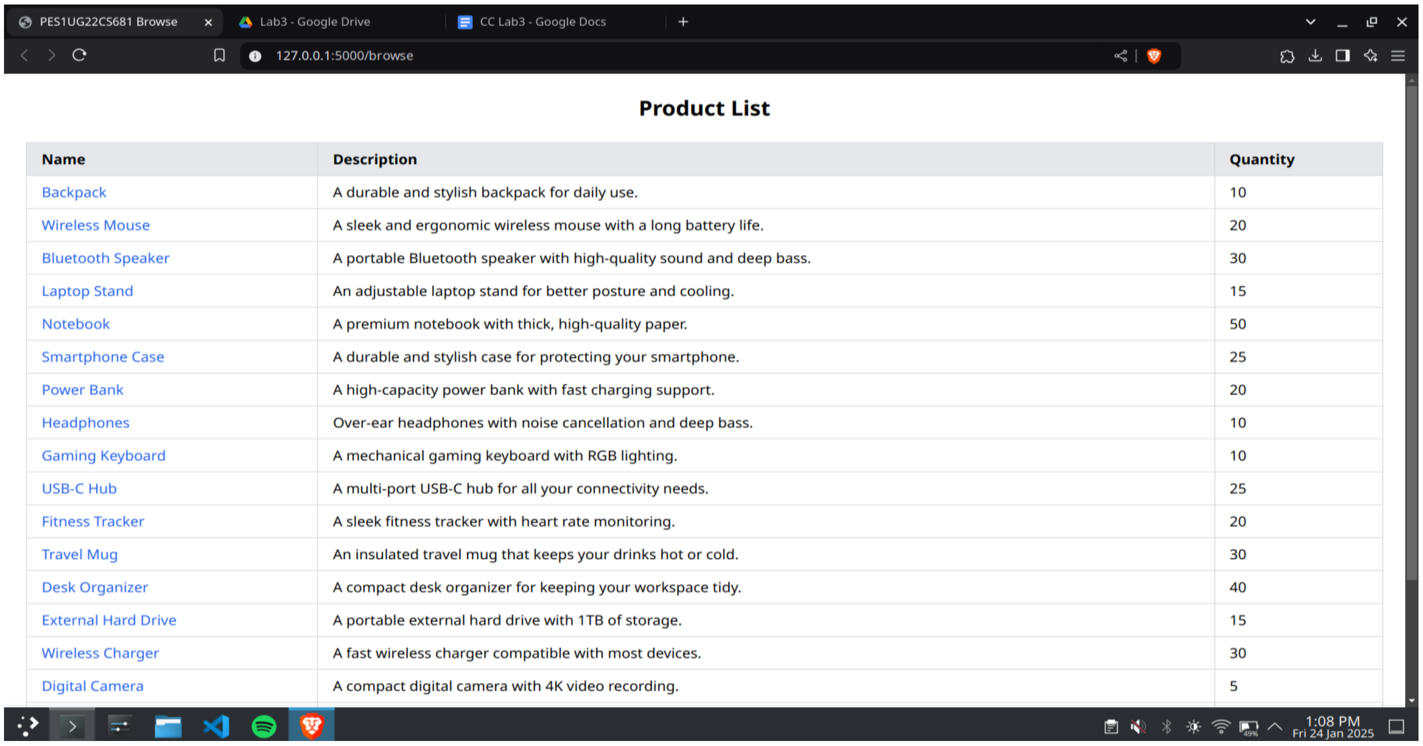
CC Lab 3

Name : Vedant Varma
SRN : PES1UG22CS681
Section : L section 6th Sem CSE
Github : https://github.com/Vedx0609/CC_Lab3

SS1 :

=> SRN has been modified and it is visible in the title of the website
=> We finished the registration with the username as Vedant Varma and the password has been set.
=> Upon logging in we get redirected to the browse page.

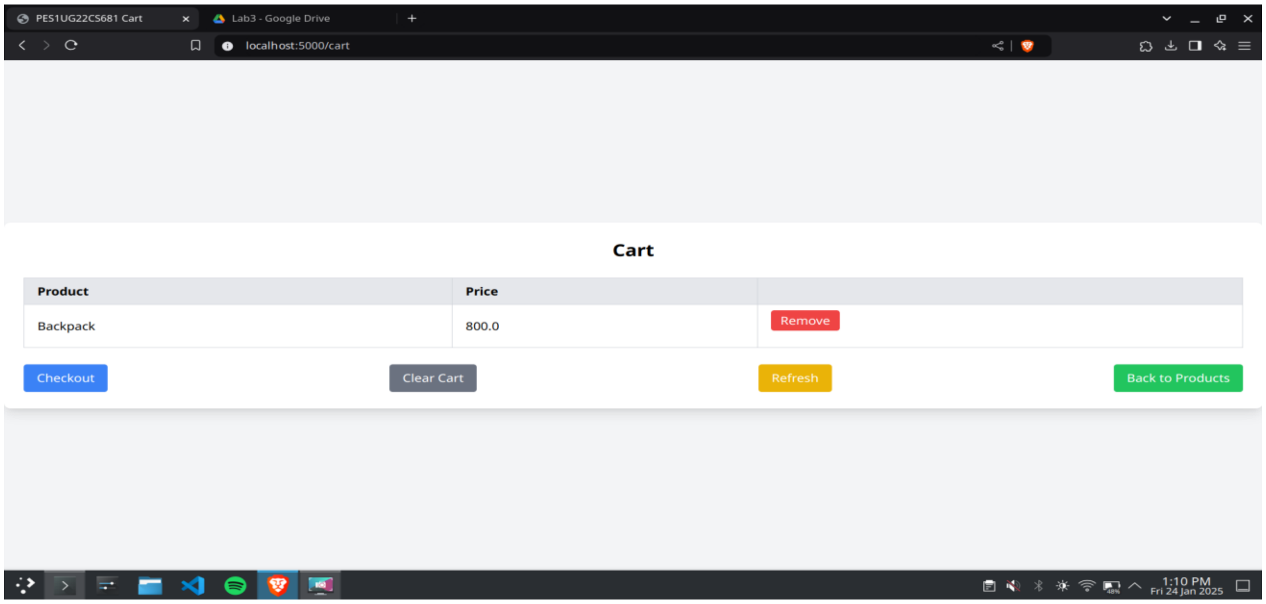


Product List

Name	Description	Quantity
Backpack	A durable and stylish backpack for daily use.	10
Wireless Mouse	A sleek and ergonomic wireless mouse with a long battery life.	20
Bluetooth Speaker	A portable Bluetooth speaker with high-quality sound and deep bass.	30
Laptop Stand	An adjustable laptop stand for better posture and cooling.	15
Notebook	A premium notebook with thick, high-quality paper.	50
Smartphone Case	A durable and stylish case for protecting your smartphone.	25
Power Bank	A high-capacity power bank with fast charging support.	20
Headphones	Over-ear headphones with noise cancellation and deep bass.	10
Gaming Keyboard	A mechanical gaming keyboard with RGB lighting.	10
USB-C Hub	A multi-port USB-C hub for all your connectivity needs.	25
Fitness Tracker	A sleek fitness tracker with heart rate monitoring.	20
Travel Mug	An insulated travel mug that keeps your drinks hot or cold.	30
Desk Organizer	A compact desk organizer for keeping your workspace tidy.	40
External Hard Drive	A portable external hard drive with 1TB of storage.	15
Wireless Charger	A fast wireless charger compatible with most devices.	30
Digital Camera	A compact digital camera with 4K video recording.	5

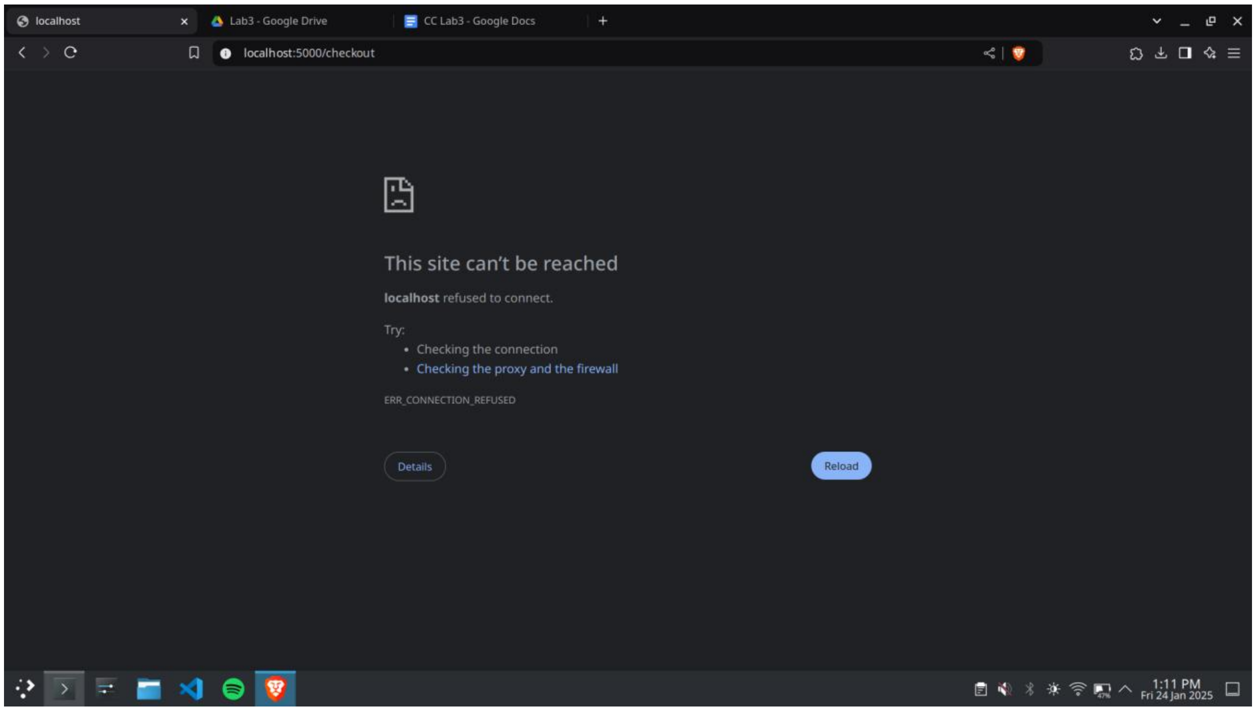
SS2 :

=> On adding items like a backpack as seen in the screenshot below, we get redirected to the cart page and on clicking checkout we get an error as mentioned in the docs.
=> The reason we get this error is because os._exit(1) will stop the execution before returning total.
=> Any logic in the calling function relying on the result of checkout() will break.



Cart

Product	Price	
Backpack	800.0	Remove



This site can't be reached

localhost refused to connect.

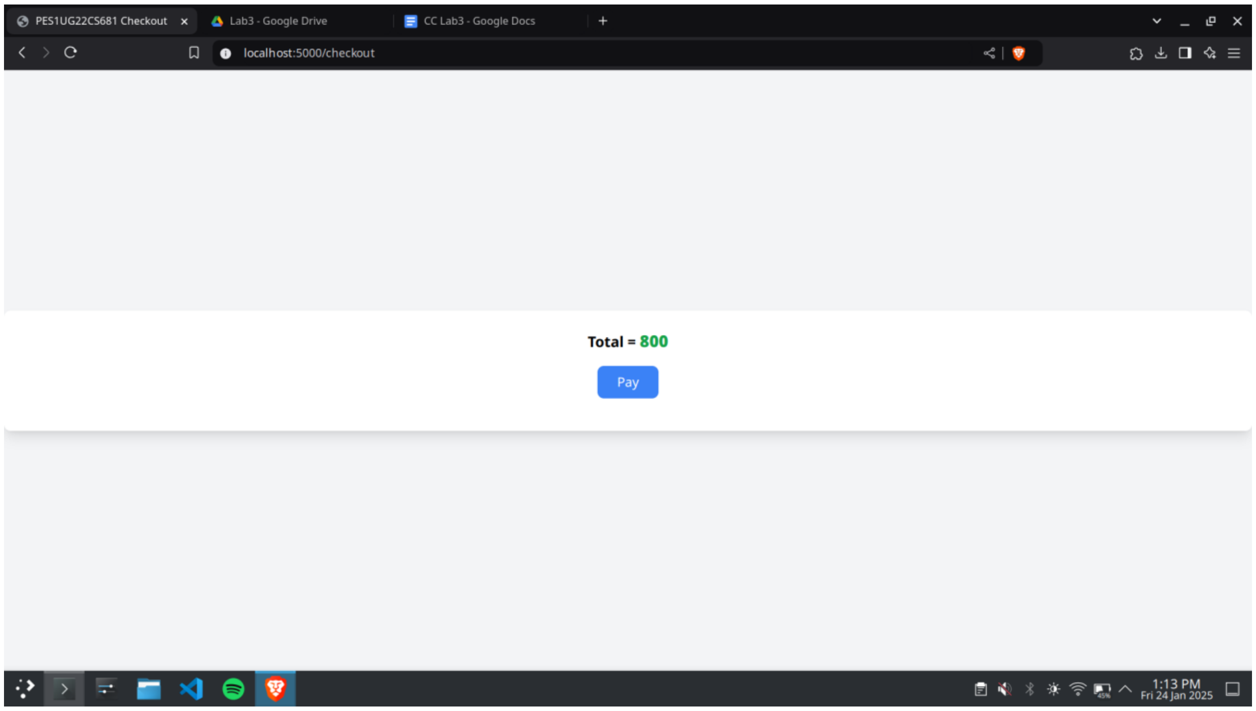
Try:

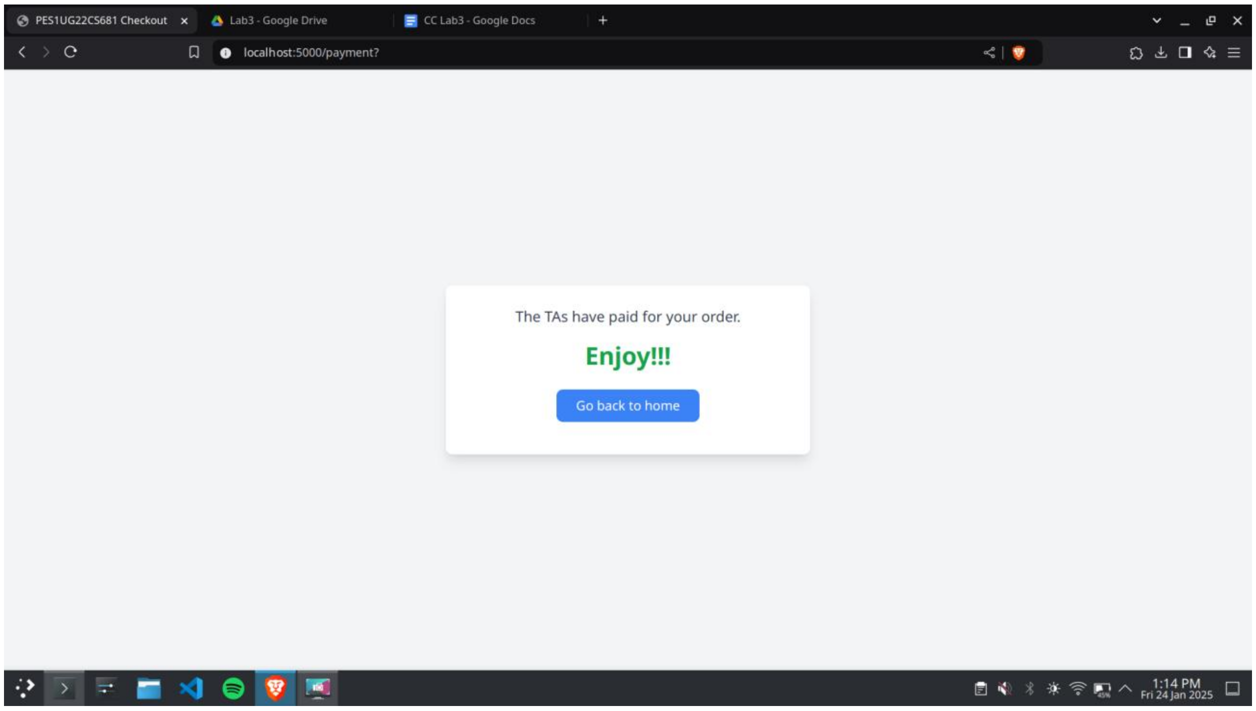
- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

SS3 :

=> We fixed the __init__.py in the checkout route by commenting the os._exit(1) line



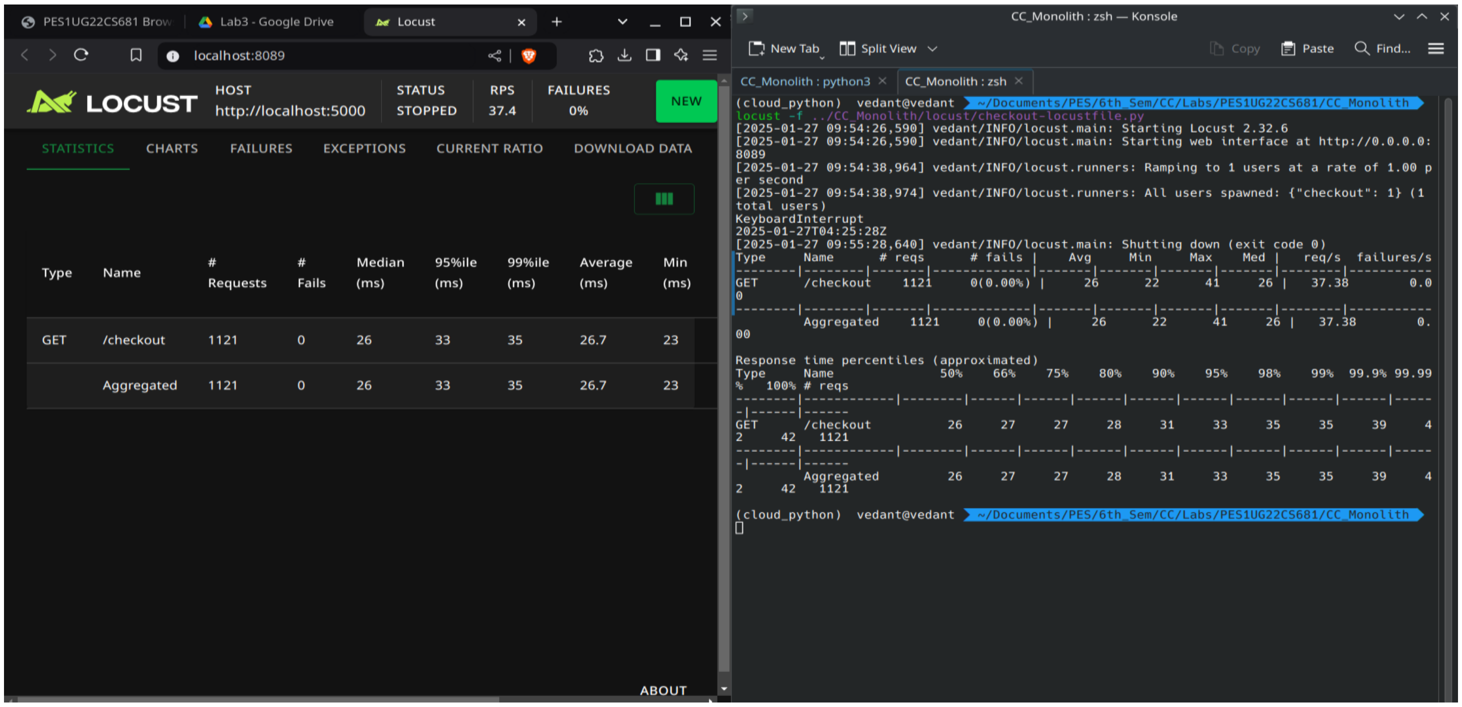


The TAs have paid for your order.

Enjoy!!!

SS4 :

=> Here we use only 1 user and 1 ramp up that is tested for 30s.
=> As we can see the average time is 26.7ms for 1121 requests with a RPS of 37.4



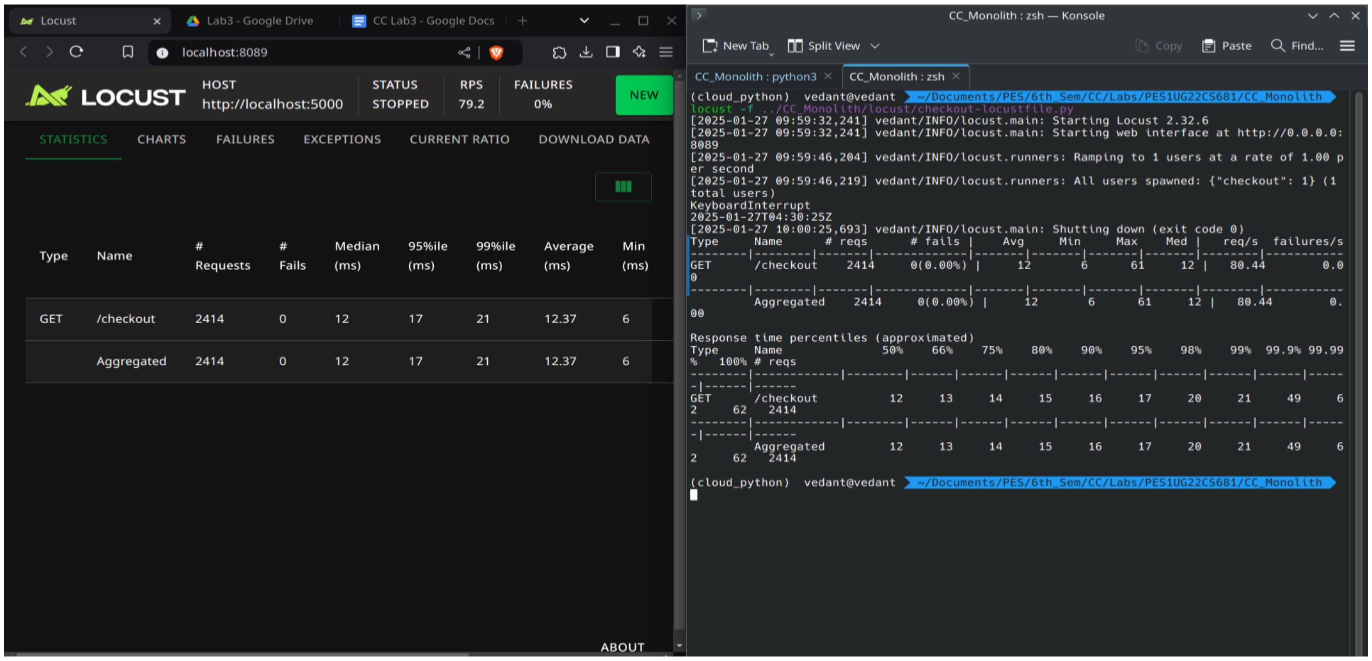
HOST	STATUS	RPS	FAILURES
http://localhost:5000	STOPPED	37.4	0%

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/checkout	1121	0	26	33	35	26.7	23
	Aggregated	1121	0	26	33	35	26.7	23

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/checkout-locustfile.py
[2025-01-27 09:54:26,590] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-27 09:54:26,590] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-27 09:54:38,964] vedant/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 p
er second
[2025-01-27 09:54:38,974] vedant/INFO/locust.runners: All users spawned: {"checkout": 1} (1
total users)
KeyboardInterrupt
2025-01-27T04:25:28Z
[2025-01-27 09:55:28,640] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name       # reqs      # fails |    Avg     Min     Max     Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /checkout    1121     0(0.00%) |     26      22      41      26 |   37.38        0.0
0
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated    1121     0(0.00%) |     26      22      41      26 |   37.38        0.
00

Response time percentiles (approximated)
Type     Name              50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|-----------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /checkout          26     27     27     28     31     33     35     35     39      4
2     42   1121
--------|-----------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated         26     27     27     28     31     33     35     35     39      4
2     42   1121

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

SS5 :

=> On optimizing the code we can see that for the similar users (1) and ramp up (1) values for 30s, this gives us an average time of 12.37ms for 2414 requests with a RPS of 79.2



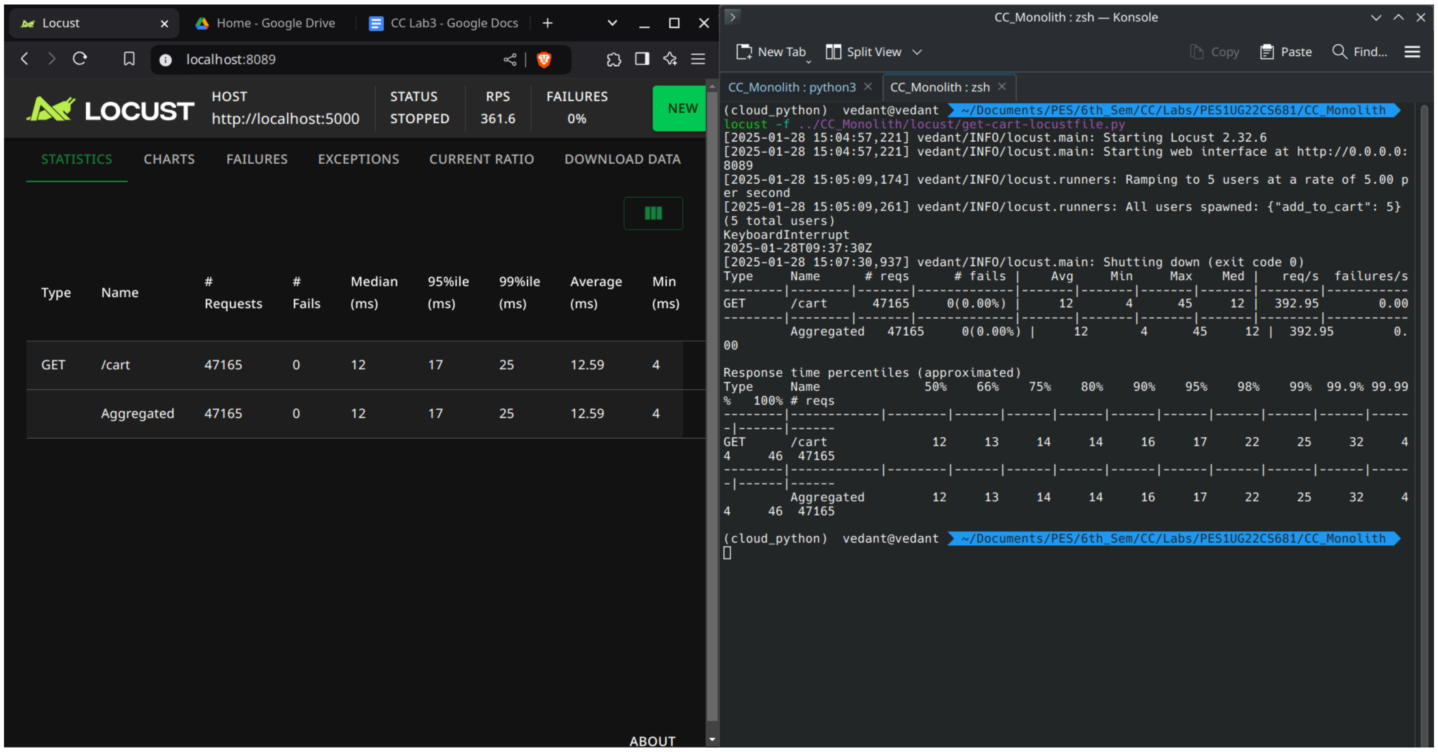
HOST	STATUS	RPS	FAILURES
http://localhost:5000	STOPPED	79.2	0%

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/checkout	2414	0	12	17	21	12.37	6
	Aggregated	2414	0	12	17	21	12.37	6

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/checkout-locustfile.py
[2025-01-27 09:59:32,241] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-27 09:59:32,241] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-27 09:59:46,204] vedant/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 p
er second
[2025-01-27 09:59:46,219] vedant/INFO/locust.runners: All users spawned: {"checkout": 1} (1
total users)
KeyboardInterrupt
2025-01-27T04:30:25Z
[2025-01-27 10:00:25,693] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name       # reqs      # fails |    Avg     Min     Max     Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /checkout    2414     0(0.00%) |     12       6      61      12 |   80.44        0.0
0
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated    2414     0(0.00%) |     12       6      61      12 |   80.44        0.
00

Response time percentiles (approximated)
Type     Name              50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|-----------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /checkout          12     13     14     15     16     17     20     21     49      6
2     62   2414
--------|-----------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated         12     13     14     15     16     17     20     21     49      6
2     62   2414

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

SS6 :

=> This is the screenshot of the unoptimized route for /cart. We use 5 users with the ramp up value as 5 for 2m.
=> As we can see the average time is 12.59ms for 47165 requests with a RPS of 361.6



HOST http://localhost:5000 | STATUS STOPPED | RPS 361.6 | FAILURES 0%

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/cart	47165	0	12	17	25	12.59	4
	Aggregated	47165	0	12	17	25	12.59	4

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/get-cart-locustfile.py
[2025-01-28 15:04:57,221] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-28 15:04:57,221] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-28 15:05:09,174] vedant/INFO/locust.runners: Ramping to 5 users at a rate of 5.00 p
er second
[2025-01-28 15:05:09,261] vedant/INFO/locust.runners: All users spawned: {"add_to_cart": 5}
(5 total users)
KeyboardInterrupt
2025-01-28T09:37:30Z
[2025-01-28 15:07:30,937] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name       # reqs      # fails |    Avg     Min     Max     Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /cart       47165     0(0.00%) |     12       4      45      12 |  392.95        0.00
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated   47165     0(0.00%) |     12       4      45      12 |  392.95        0.
00

Response time percentiles (approximated)
Type     Name                 50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /cart                 12     13     14     14     16     17     22     25     32     4
4     46  47165
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated            12     13     14     14     16     17     22     25     32     4
4     46  47165

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

SS7 :

=> We need to optimize the /cart route. In the __init__.py file under the /cart route we can see that evals and the for loops are used too much.
=> We replace these with json.loads and list comprehensions for the following reasons :

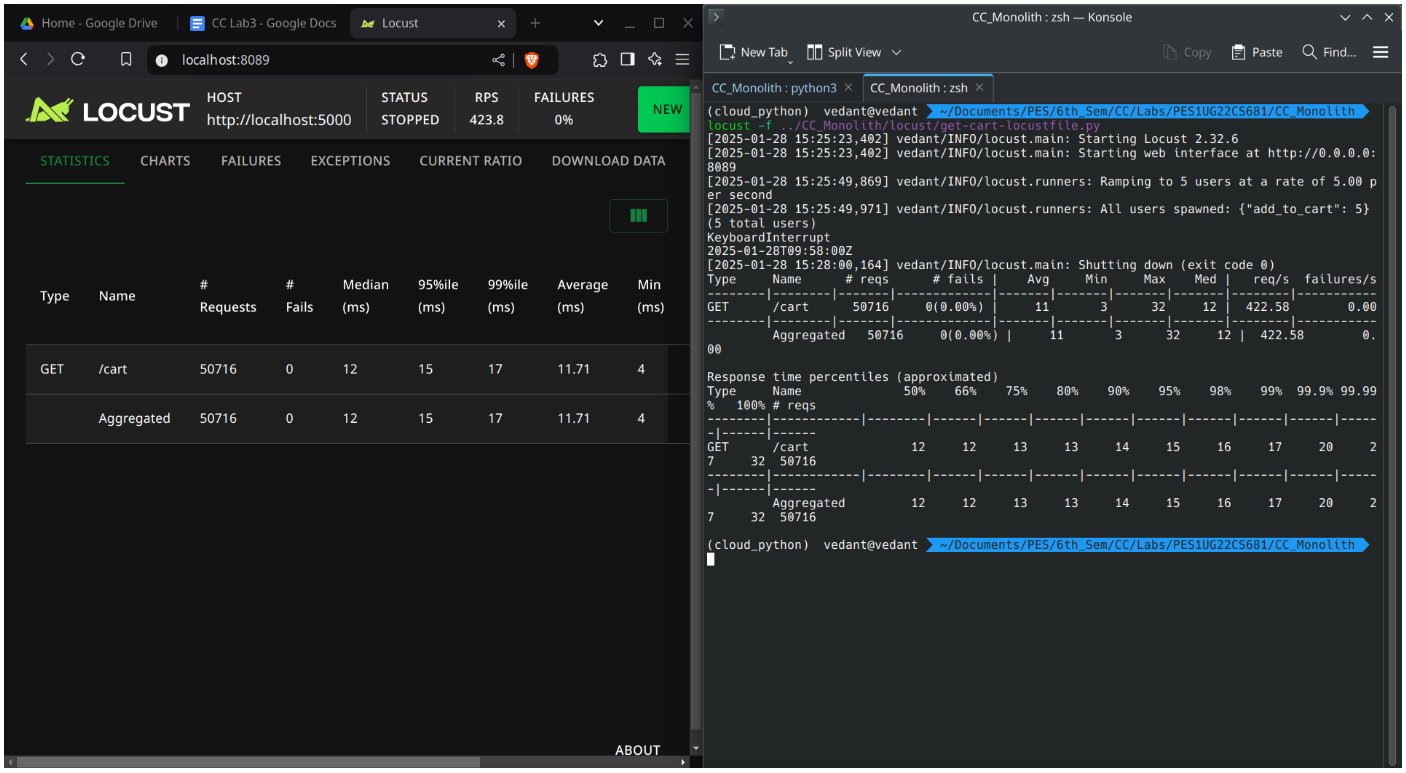
1) eval is slower because it processes the string as Python code, invoking the interpreter and handling more overhead whereas json.loads is optimized for parsing JSON and is faster in comparison for JSON-specific tasks
2) List comprehensions are implemented in C at the interpreter level, making them more efficient than the equivalent for loop with list.append(), which involves additional overhead from Python's function calls
3) Updating the dao.py file and removing the redundant temp list append

Hence we replace and optimize the code and write the function as follows :

```
def get_cart(username: str) -> list:
    cart_details = dao.get_cart(username)
    if cart_details is None:
        return []
    ids = [j for i in cart_details for j in
json.loads(i['contents'])]
    return [products.get_product(i) for i in ids]
```

dao.py :

```
cart = cursor.fetchall()
    final_cart = []
    for item in cart:
        final_cart.append(item)
```



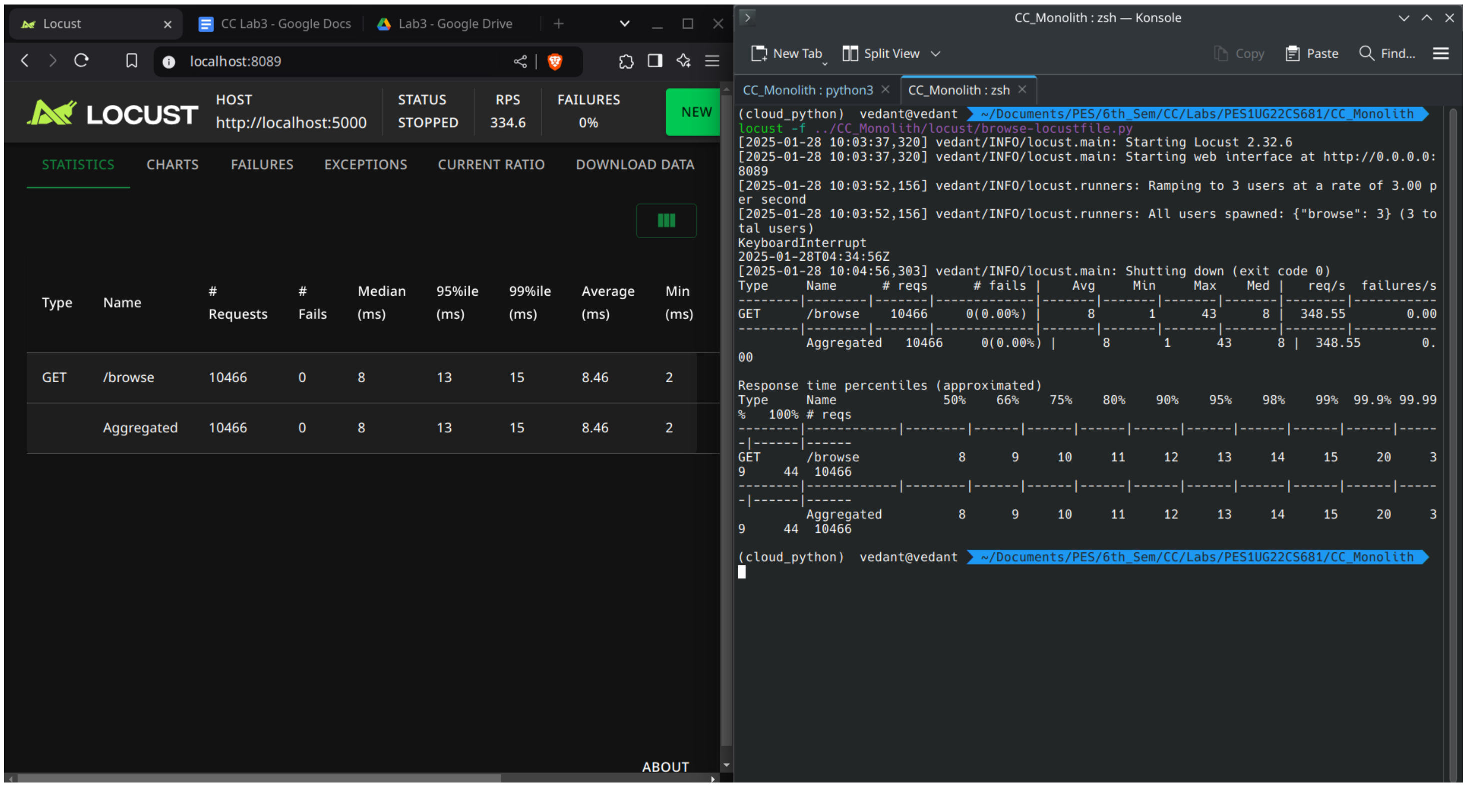
Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/cart	50716	0	12	15	17	11.71	4
	Aggregated	50716	0	12	15	17	11.71	4

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/get-cart-locustfile.py
[2025-01-28 15:25:23,402] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-28 15:25:23,402] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-28 15:25:49,869] vedant/INFO/locust.runners: Ramping to 5 users at a rate of 5.00 p
er second
[2025-01-28 15:25:49,971] vedant/INFO/locust.runners: All users spawned: {"add_to_cart": 5}
(5 total users)
KeyboardInterrupt
2025-01-28T09:58:00Z
[2025-01-28 15:28:00,164] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name      # reqs      # fails |    Avg     Min     Max    Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /cart      50716     0(0.00%) |     11       3      32     12 |  422.58        0.00
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated   50716     0(0.00%) |     11       3      32     12 |  422.58        0.
00

Response time percentiles (approximated)
Type     Name                  50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /cart                  12     12     13     13     14     15     16     17     20     2
7     32  50716
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated             12     12     13     13     14     15     16     17     20     2
7     32  50716

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

=> This is the screenshot of the optimized route for /cart. We use 5 users with the ramp up value as 5 for 2m.
=> As we can see the average time is 11.71ms for 50716 requests with a RPS of 423.8

SS8 :

As done for /cart, we perform a similar comparison for /browse. We use 3 users and the ramp up value is 3 which is tested for 30s. The average time is 8.46ms for 10466 requests with a RPS of 334.6



HOST	STATUS	RPS	FAILURES
http://localhost:5000	STOPPED	334.6	0%

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/browse	10466	0	8	13	15	8.46	2
	Aggregated	10466	0	8	13	15	8.46	2

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/browse-locustfile.py
[2025-01-28 10:03:37,320] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-28 10:03:37,320] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-28 10:03:52,156] vedant/INFO/locust.runners: Ramping to 3 users at a rate of 3.00 p
er second
[2025-01-28 10:03:52,156] vedant/INFO/locust.runners: All users spawned: {"browse": 3} (3 to
tal users)
KeyboardInterrupt
2025-01-28T04:34:56Z
[2025-01-28 10:04:56,303] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name       # reqs      # fails |    Avg     Min     Max    Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /browse    10466     0(0.00%) |      8       1      43      8 |  348.55        0.00
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated   10466     0(0.00%) |      8       1      43      8 |  348.55        0.
00

Response time percentiles (approximated)
Type     Name                  50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /browse                 8      9     10     11     12     13     14     15     20      3
9     44  10466
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated              8      9     10     11     12     13     14     15     20      3
9     44  10466

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

SS9 :

=> We need to optimize the /product route. In the __init__.py file under the /product route we can see that the for loops can be replaced with list comprehensions like done in the /cart route

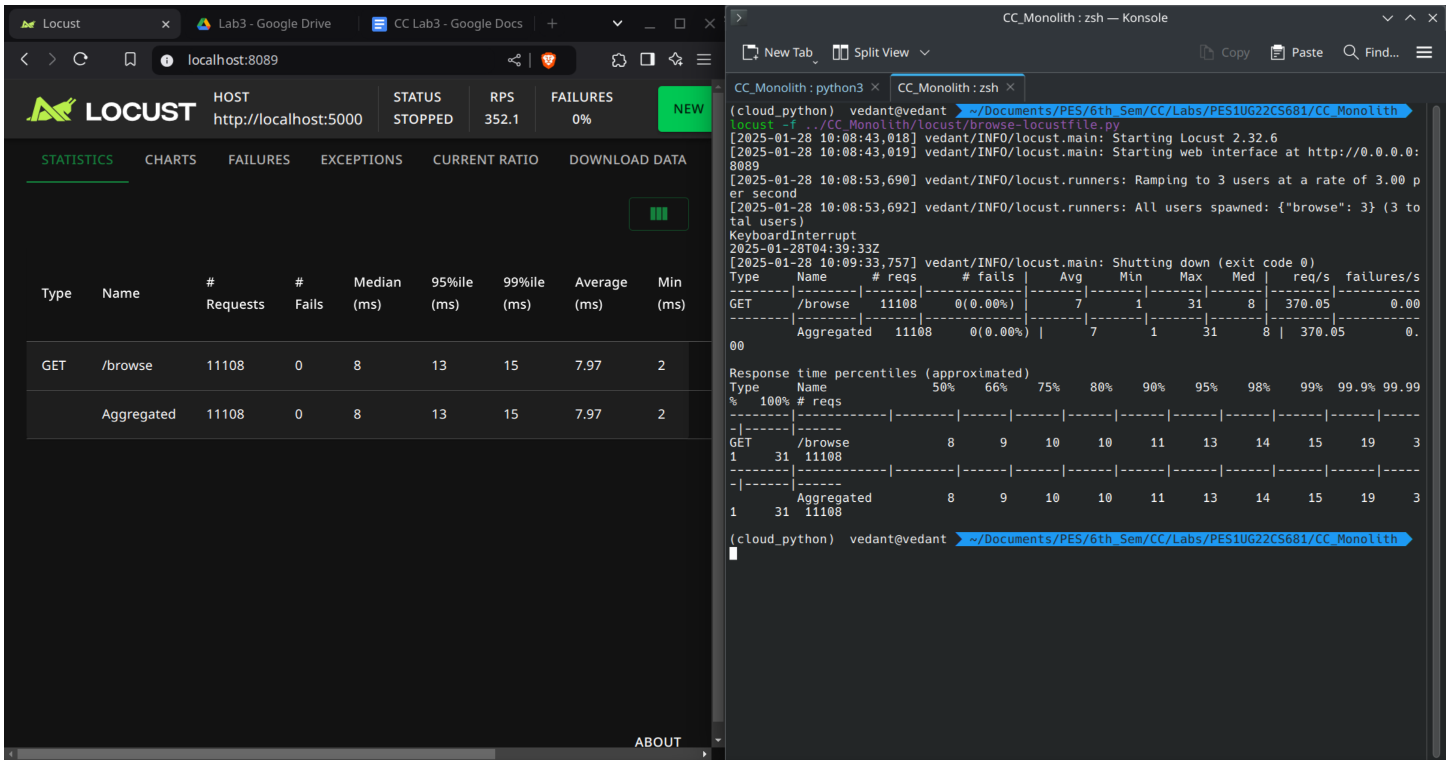
List comprehensions are implemented in C at the interpreter level, making them more efficient than the equivalent for loop with list.append(), which involves additional overhead from Python's function calls

Hence in the method list_products defined in the __init__.py and dao.py present in the /product route we replace the code by using a list comprehension :

```
def list_products() -> list[Product]:
    result = [Product.load(data) for data in dao.list_products()]
    return result
```

dao.py :

```
rows = cursor.fetchall()
    products = [i for i in rows]
```



LOCUST | HOST http://localhost:5000 | STATUS STOPPED | RPS 352.1 | FAILURES 0%

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)
GET	/browse	11108	0	8	13	15	7.97	2
	Aggregated	11108	0	8	13	15	7.97	2

```
(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
locust -f ../CC_Monolith/locust/browse-locustfile.py
[2025-01-28 10:08:43,018] vedant/INFO/locust.main: Starting Locust 2.32.6
[2025-01-28 10:08:43,019] vedant/INFO/locust.main: Starting web interface at http://0.0.0.0:
8089
[2025-01-28 10:08:53,690] vedant/INFO/locust.runners: Ramping to 3 users at a rate of 3.00 p
er second
[2025-01-28 10:08:53,692] vedant/INFO/locust.runners: All users spawned: {"browse": 3} (3 to
tal users)
KeyboardInterrupt
2025-01-28T04:39:33Z
[2025-01-28 10:09:33,757] vedant/INFO/locust.main: Shutting down (exit code 0)
Type     Name       # reqs      # fails |    Avg     Min     Max    Med |   req/s  failures/s
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
GET      /browse    11108     0(0.00%) |      7       1      31      8 |  370.05        0.00
--------|--------|-------|-------------|-------|-------|-------|-------|--------|-----------
         Aggregated   11108     0(0.00%) |      7       1      31      8 |  370.05        0.
00

Response time percentiles (approximated)
Type     Name              50%    66%    75%    80%    90%    95%    98%    99%  99.9% 99.99
%   100% # reqs
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
GET      /browse             8      9     10     10     11     13     14     15     19      3
1     31  11108
--------|------------|--------|------|------|------|------|------|------|------|------|-----
-|------|------
         Aggregated          8      9     10     10     11     13     14     15     19      3
1     31  11108

(cloud_python)  vedant@vedant  ~/Documents/PES/6th_Sem/CC/Labs/PES1UG22CS681/CC_Monolith
```

As mentioned earlier we use 3 users and ramp up as 3 and it’s evident that the average time has been reduced to 7.97ms for 11108 requests with a RPS of 352.1