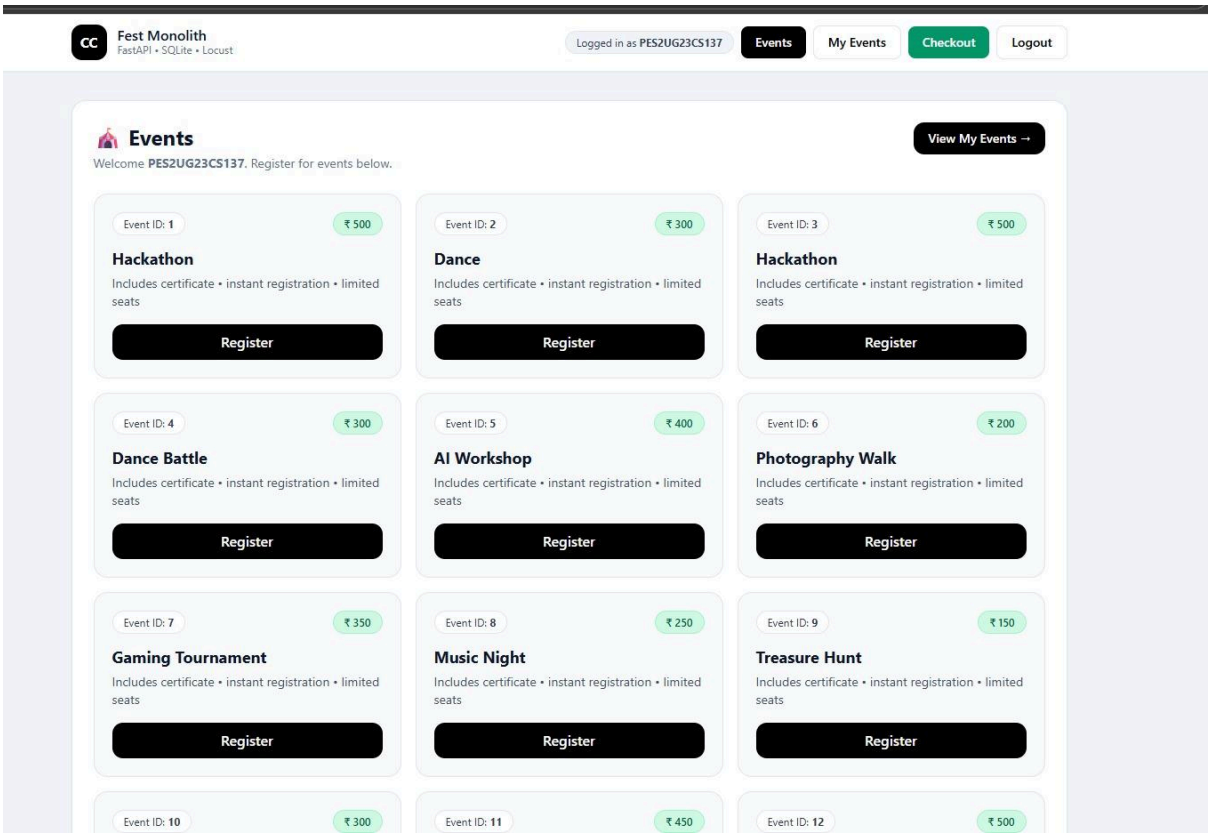
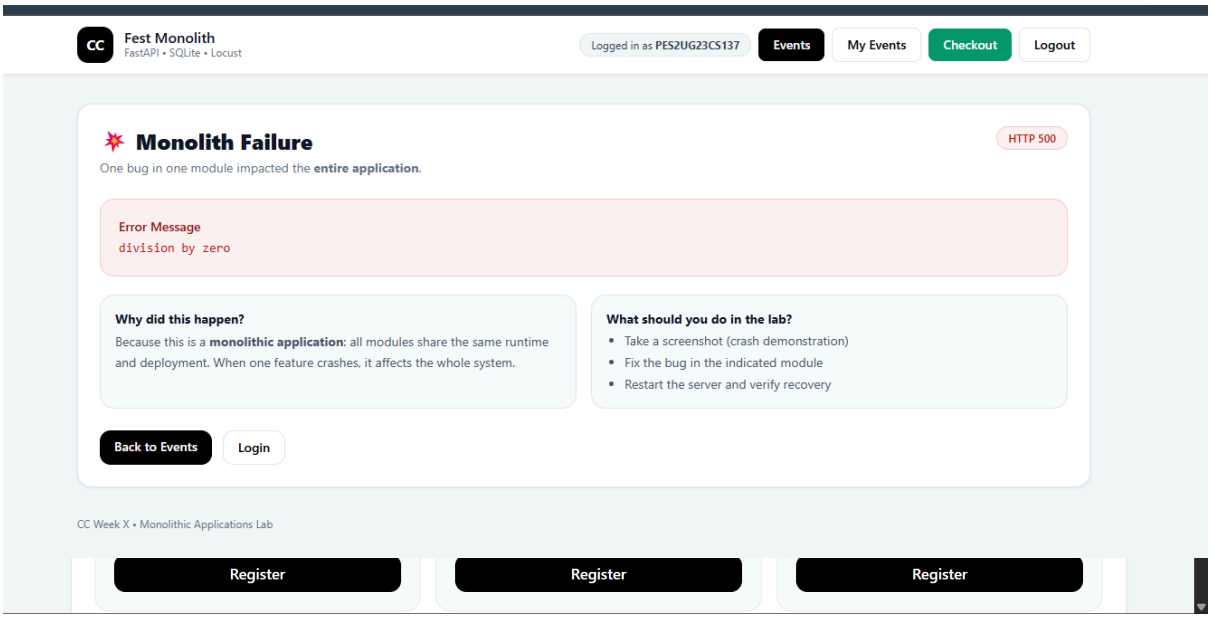
# CC LAB-2

Name:C Bhargav
SRN: PES2UG23CS137
Section: C

SS1

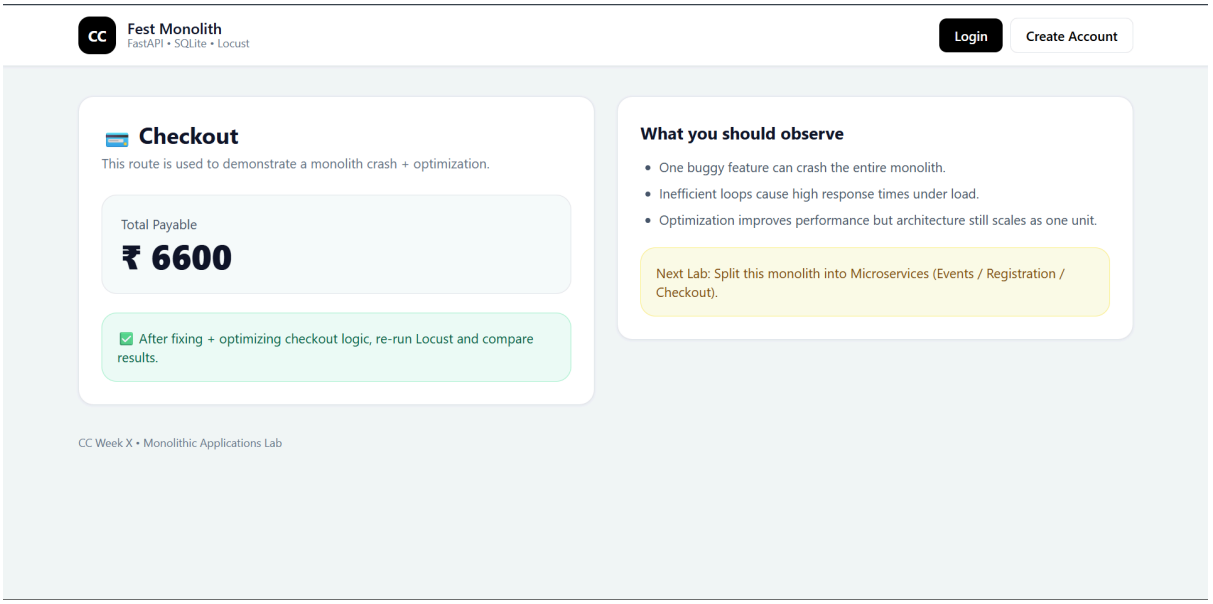


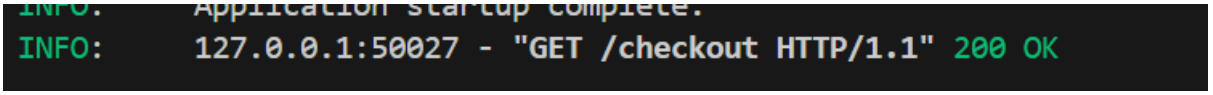
SS2



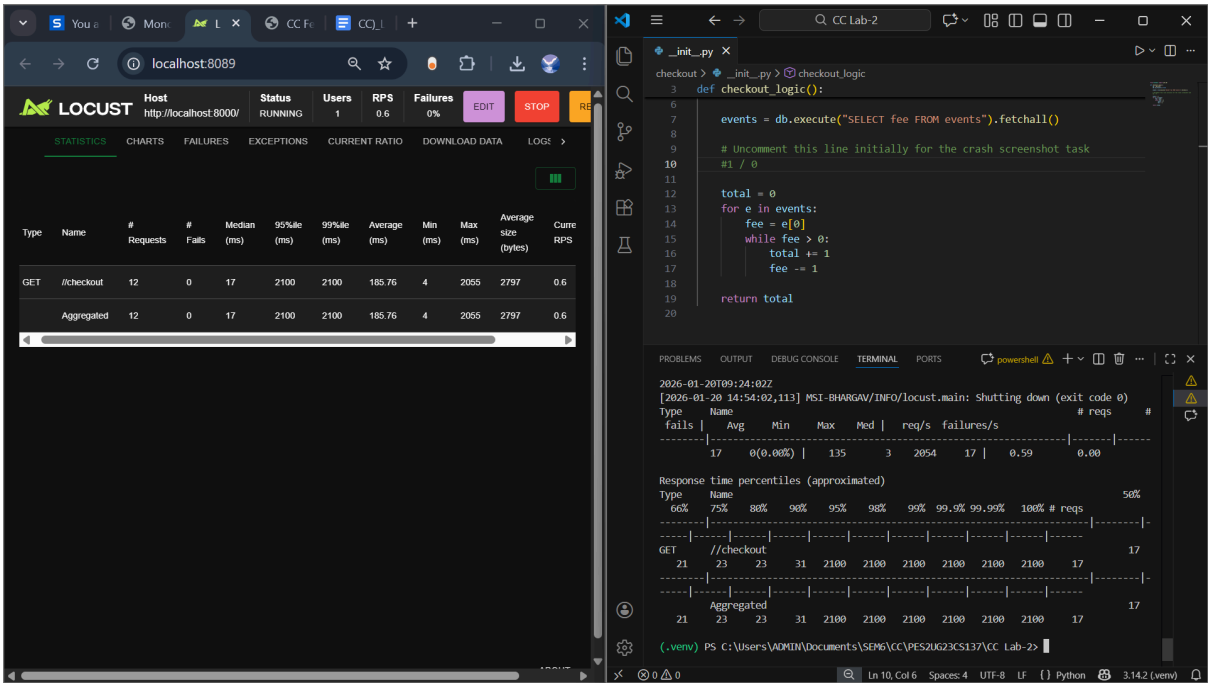

```
INFO:     127.0.0.1:53390 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR:    Exception in ASGI application
Traceback (most recent call last):
  File "C:\Users\ADMIN\Documents\SEM6\CC\PES2UG23CS137\CC Lab-2\.venv\Lib\site-packages\uvicorn\protocols\http\h11_impl.py", line 410, in run_asgi
    result = await app(  # type: ignore[func-returns-value]
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
        self.scope, self.receive, self.send
```

## SS3

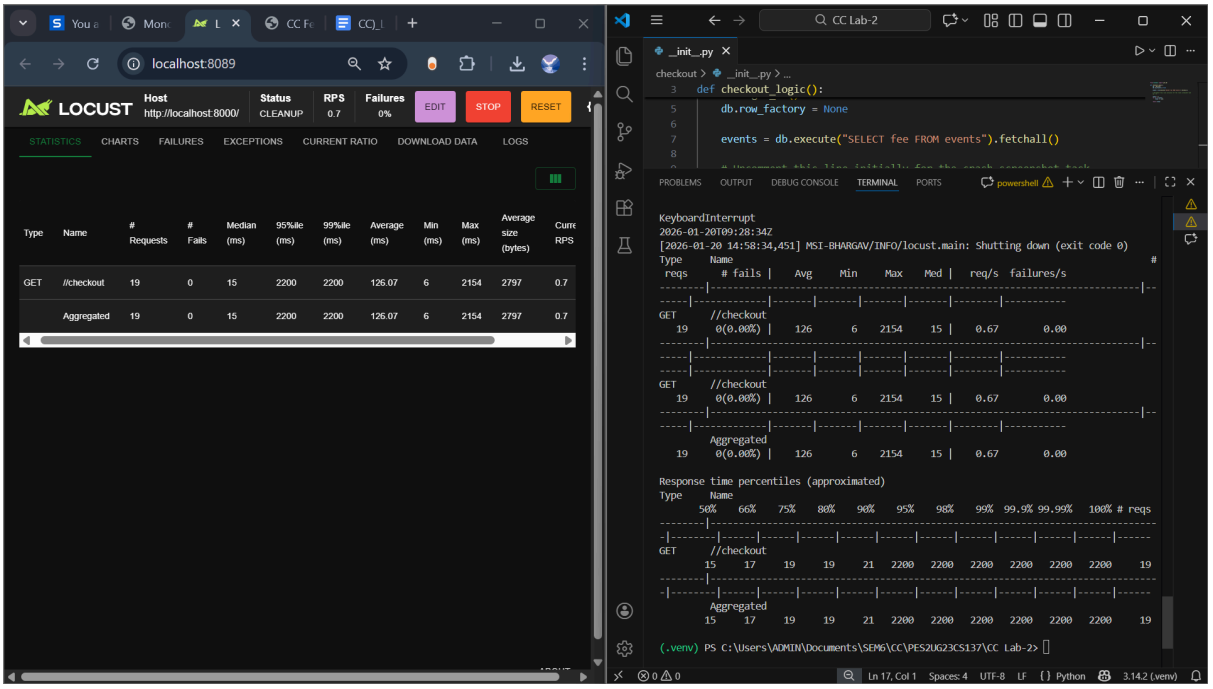


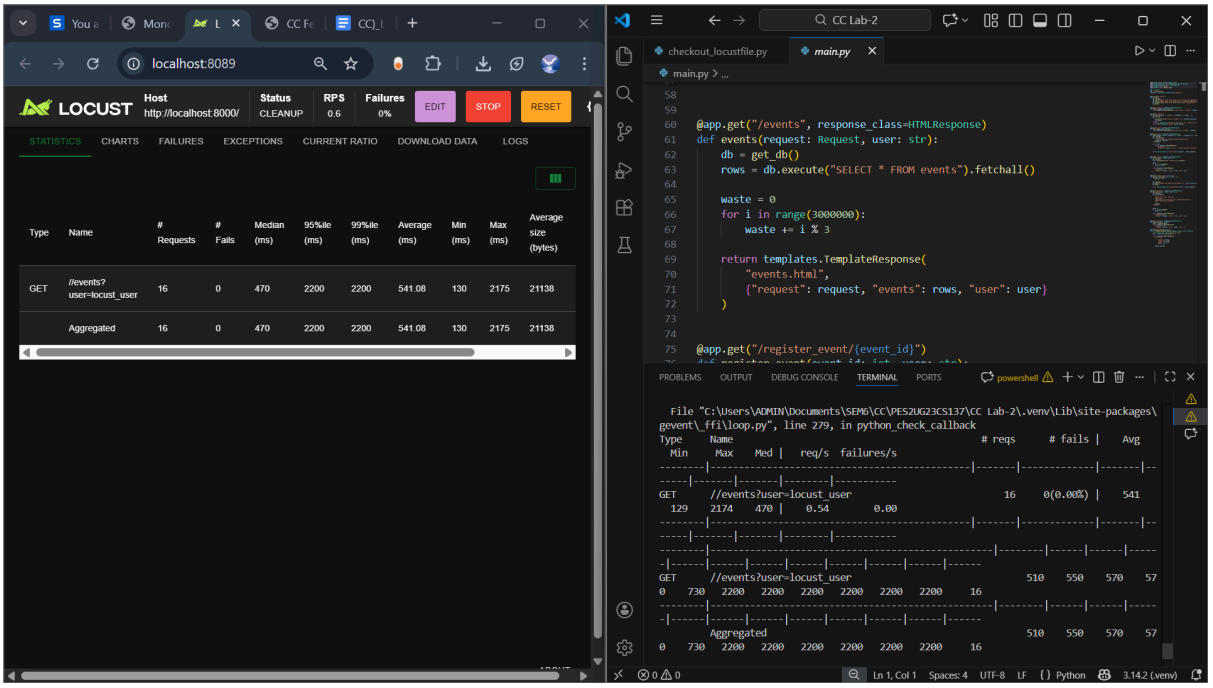

## SS4(before)



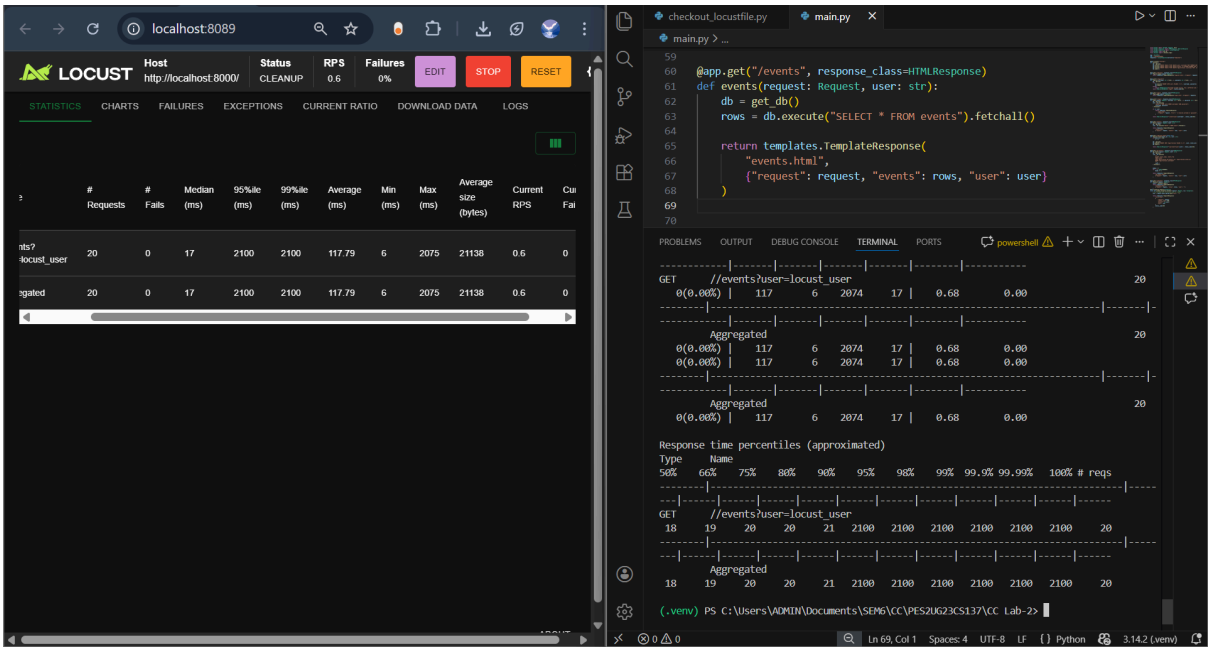
## SS5(After)



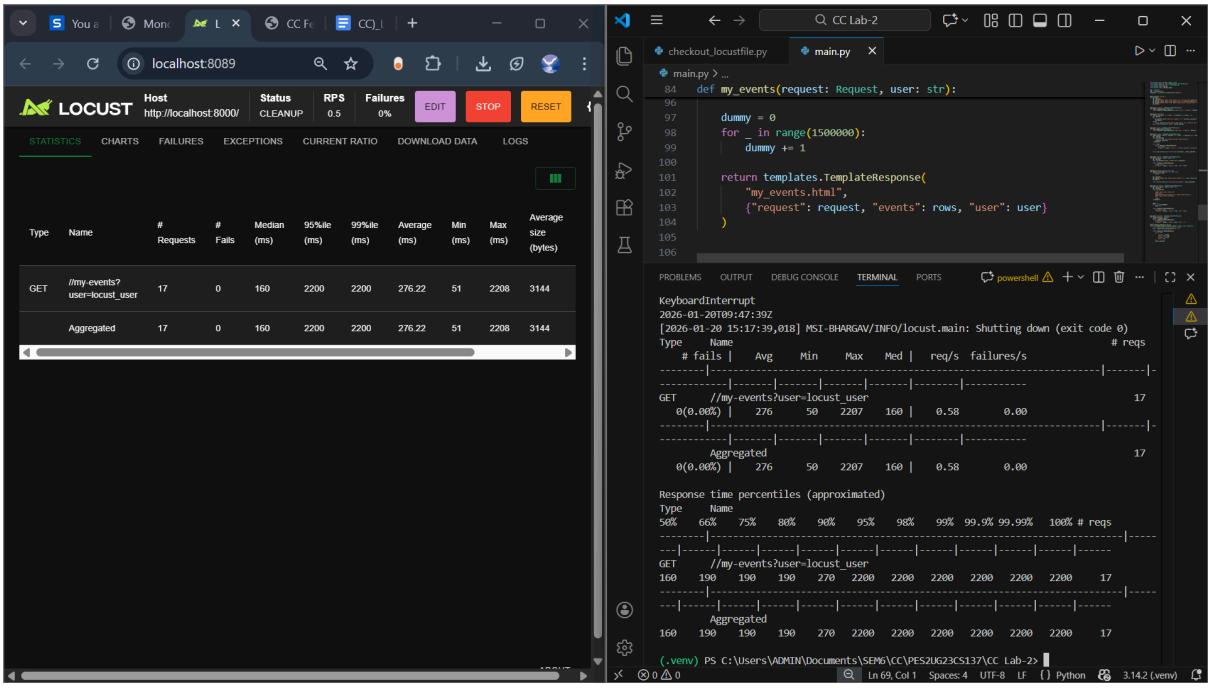

## SS6 (Before)



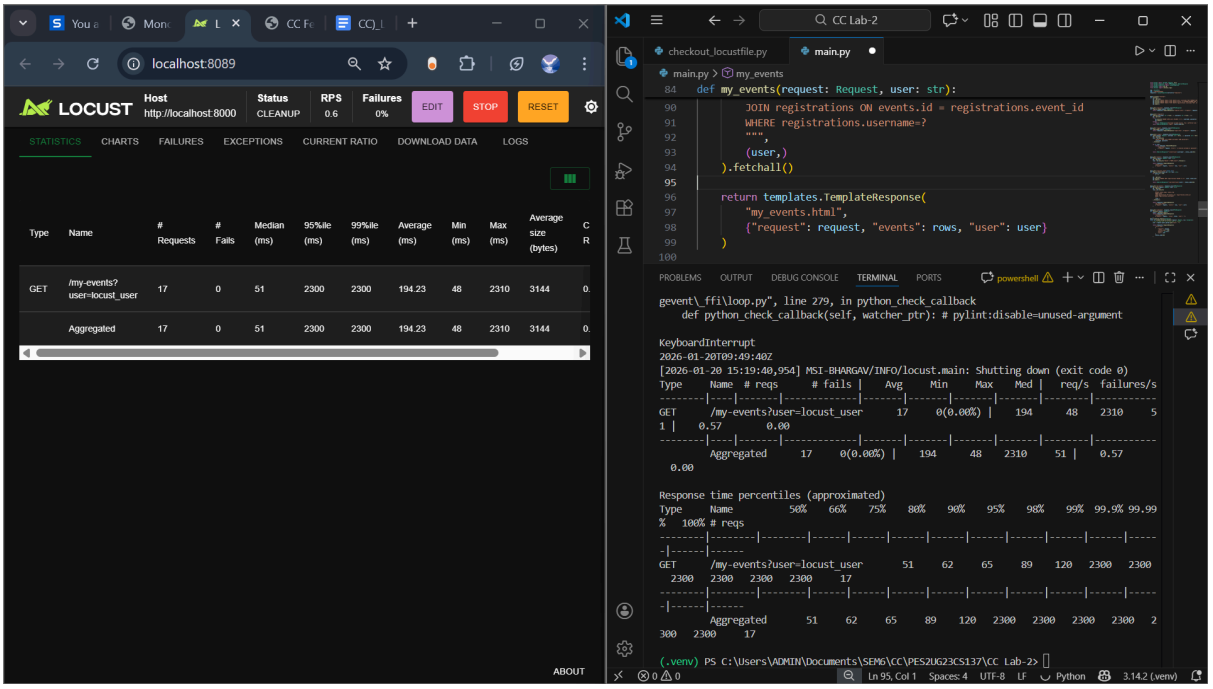
SS7 (After)




SS8(Before)



SS9 (After)




## Performance Optimization Explanations

### Route 1: /events

**Bottleneck:**
A wasteful CPU computation loop executing 3,000,000 iterations with modulo and addition operations (waste += i % 3) that blocked every request.

**Change Made:**
Removed the entire wasteful computation loop (lines 65-67).

**Why Performance Improved:**
The synchronous loop was performing 3 million unnecessary arithmetic operations before sending the response. Eliminating this CPU-bound work allowed the server to respond immediately after the database query, reducing median response time from 470ms to 17ms (96% improvement).

**Before Optimization (SS6):**

Median response time was 470ms
Average response time was 541ms
95th percentile hit 2200ms
99th percentile reached 2200ms
Throughput: 0.6 requests per second

**After Optimization (SS7):**

Median dropped to just 17ms (96.4% faster)
Average improved to 118ms (78.2% faster)
95th percentile: 2100ms
99th percentile: 2100ms
Throughput remained at 0.6 RPS

**Route 2: /my-events**

**Bottleneck:**
A dummy computation loop executing 1,500,000 iterations of simple increment operations (dummy += 1) that delayed every response.

**Change Made:**
Removed the entire dummy computation loop (lines 97-99).

**Why Performance Improved:**
The blocking loop performed 1.5 million unnecessary increment operations on every request. Removing this CPU-bound work freed the server to respond immediately after fetching user events from the database, reducing median response time from 160ms to 51ms (68% improvement).

**Before Optimization (SS8):**

Median response time was 160ms
Average response time was 276ms
95th percentile was 2200ms
99th percentile reached 2200ms
Throughput: 0.5 requests per second

**After Optimization (SS9):**

Median decreased to 51ms (68.1% faster!)
Average improved to 194ms (29.7% faster)
95th percentile: 2300ms
99th percentile: 2300ms
Throughput increased to 0.6 RPS (20% improvement)

Github Link: https://github.com/BHARGAVC27/CC_LAB2/tree/main