CLOUD COMPUTING

LAB 2 - MONOLITHIC ARCHITECTURE

NAME: SOWMYA HARDAGERI

SRN: PES2UG23CS590

SECTION: I

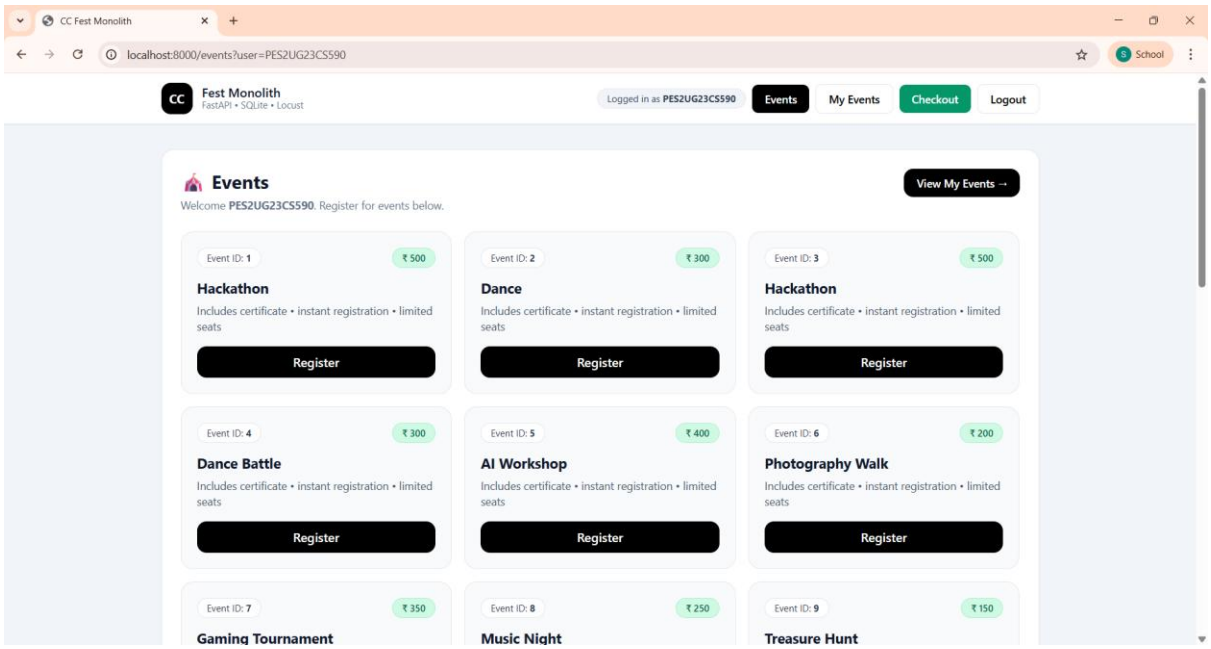
CAMPUS: ELECTRONIC CITY CAMPUS

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

Screenshots:

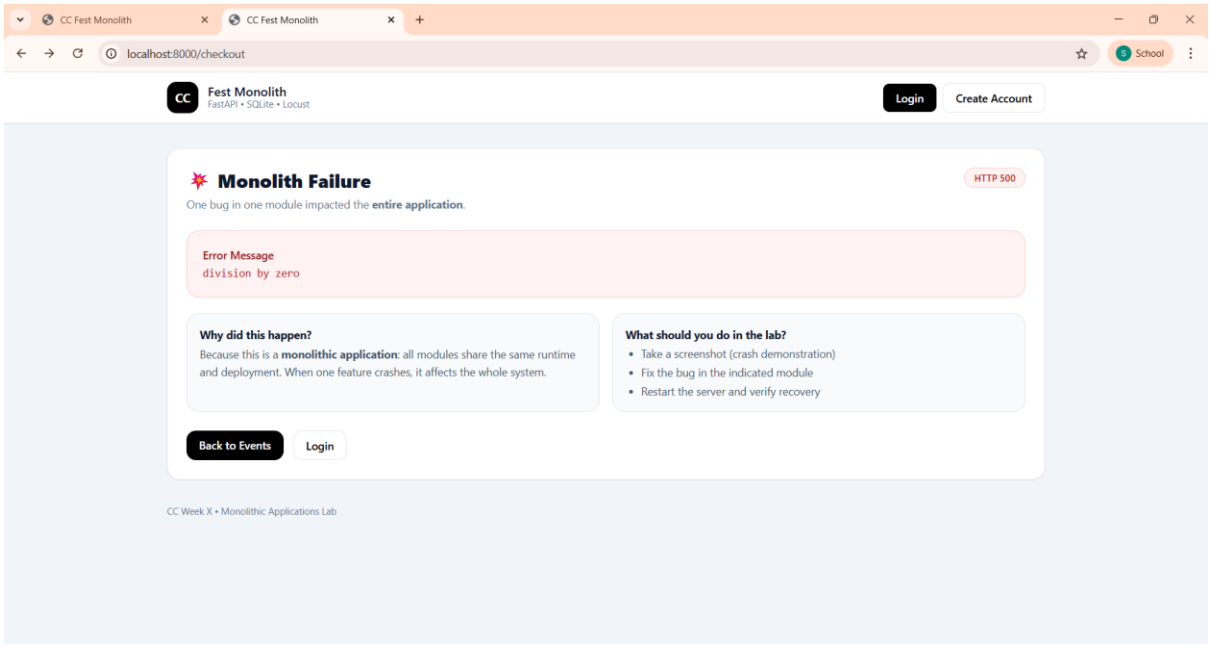
PART 2: Use the Application

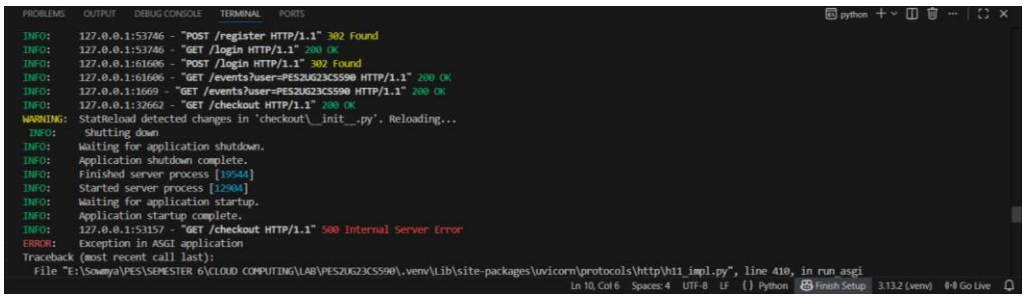
Account Created Successfully



PART 3: Observe Monolithic Failure (Crash)

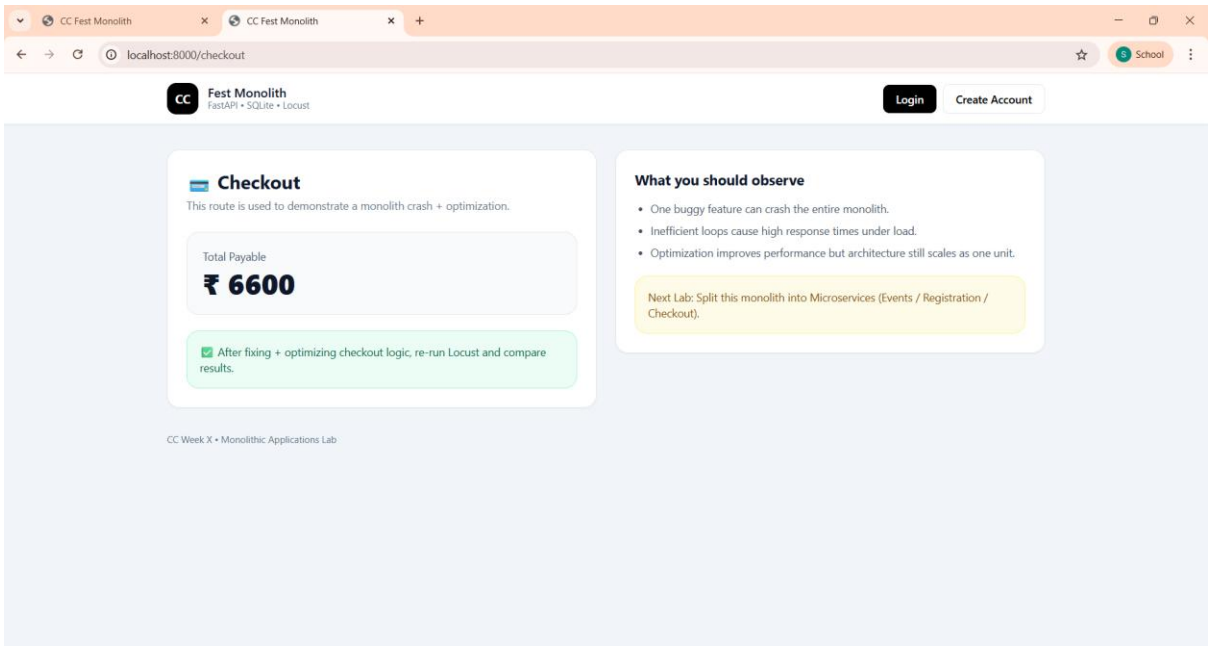
Monolithic Failed Successfully

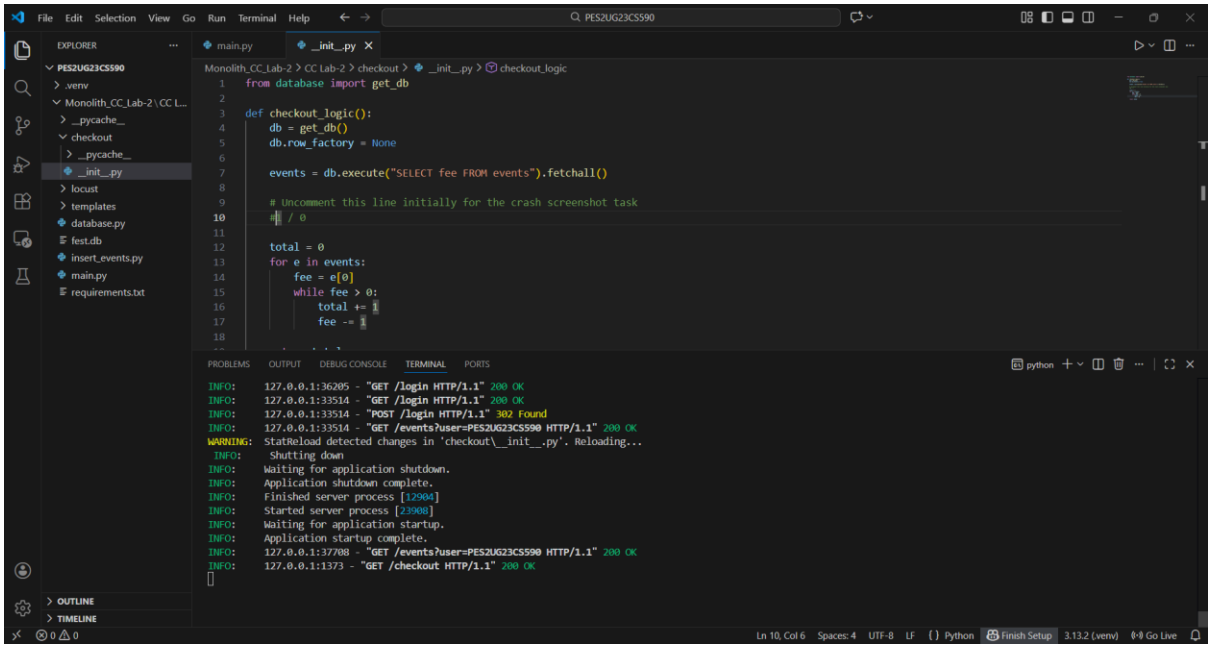




PART 4: Fix the Bug

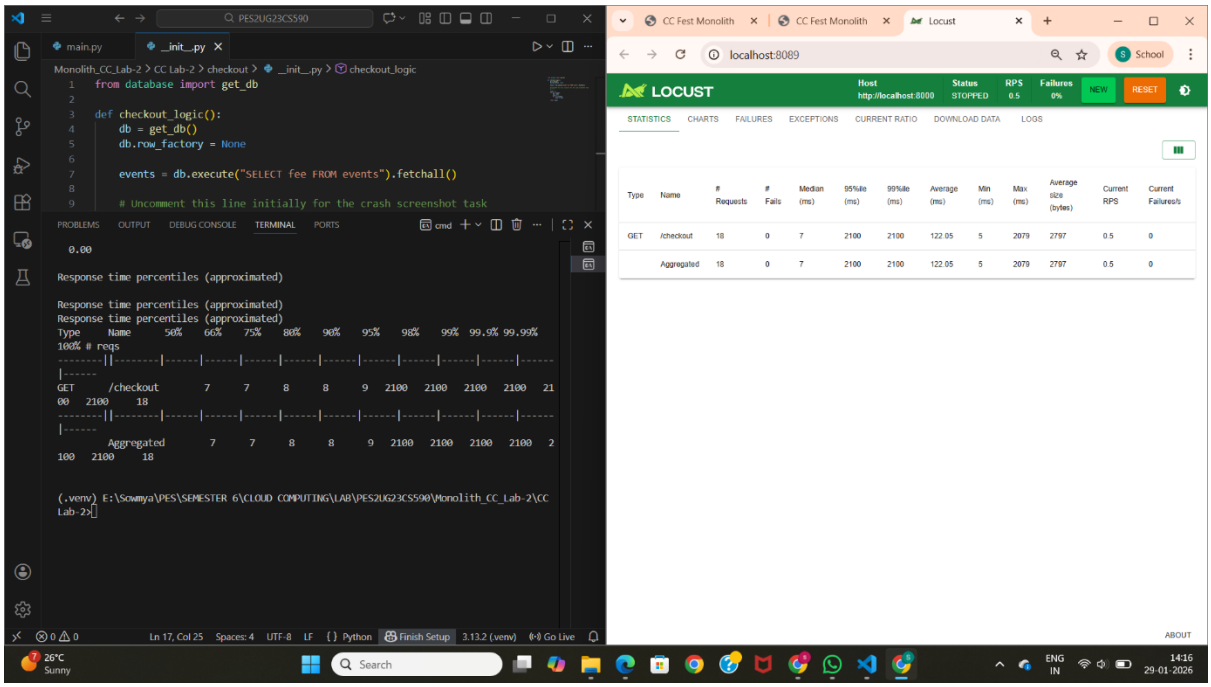
Bug Fixed





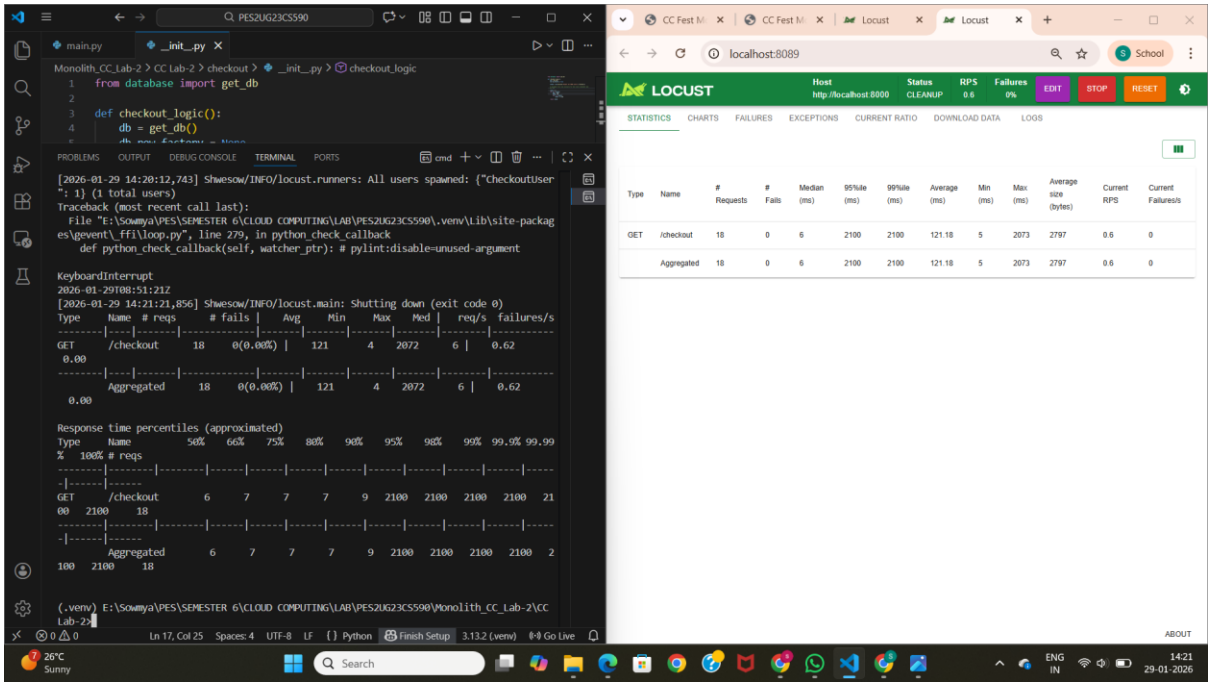
PART 5: Load Testing using Locust

Terminal and Locust before Optimization



PART 6: Optimize the Checkout Route

Terminal and Locust after OptimizatioN

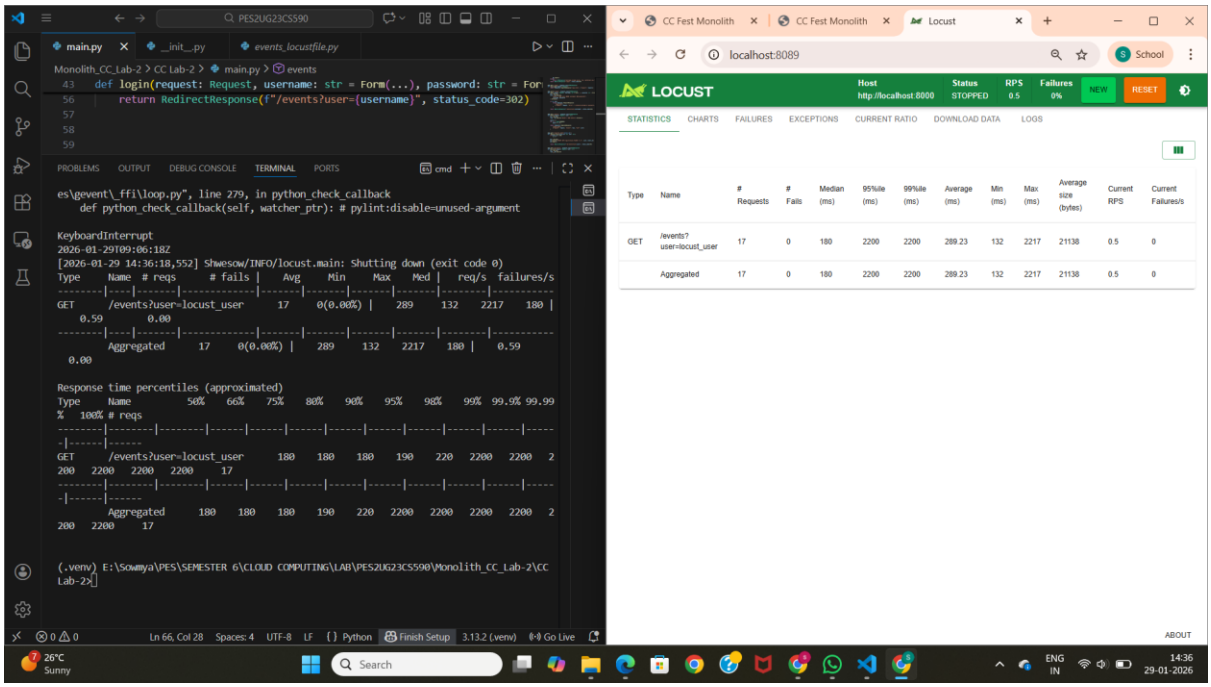


Observation: Average response time before optimization was 122.05, after optimizing the code, the time reduced to 121.18 whereas RPS stayed almost the same.

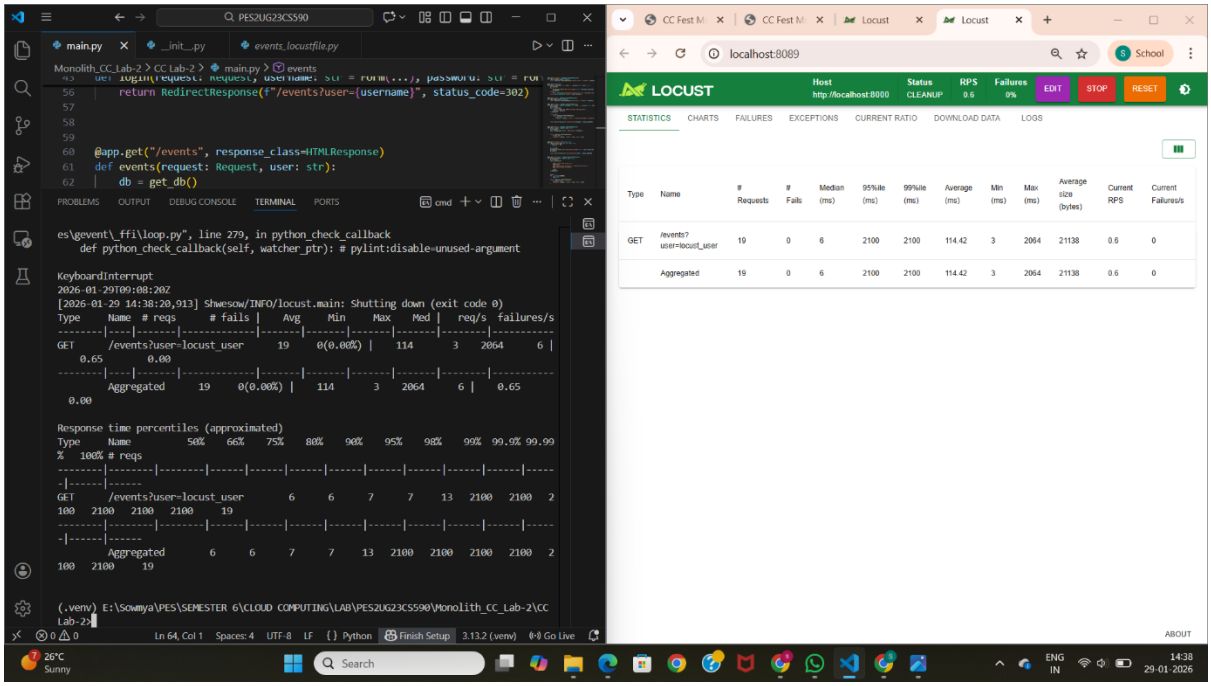
PART 7: Optimise events and my_events(DIY)

Route 1: /events

Before Optimization:

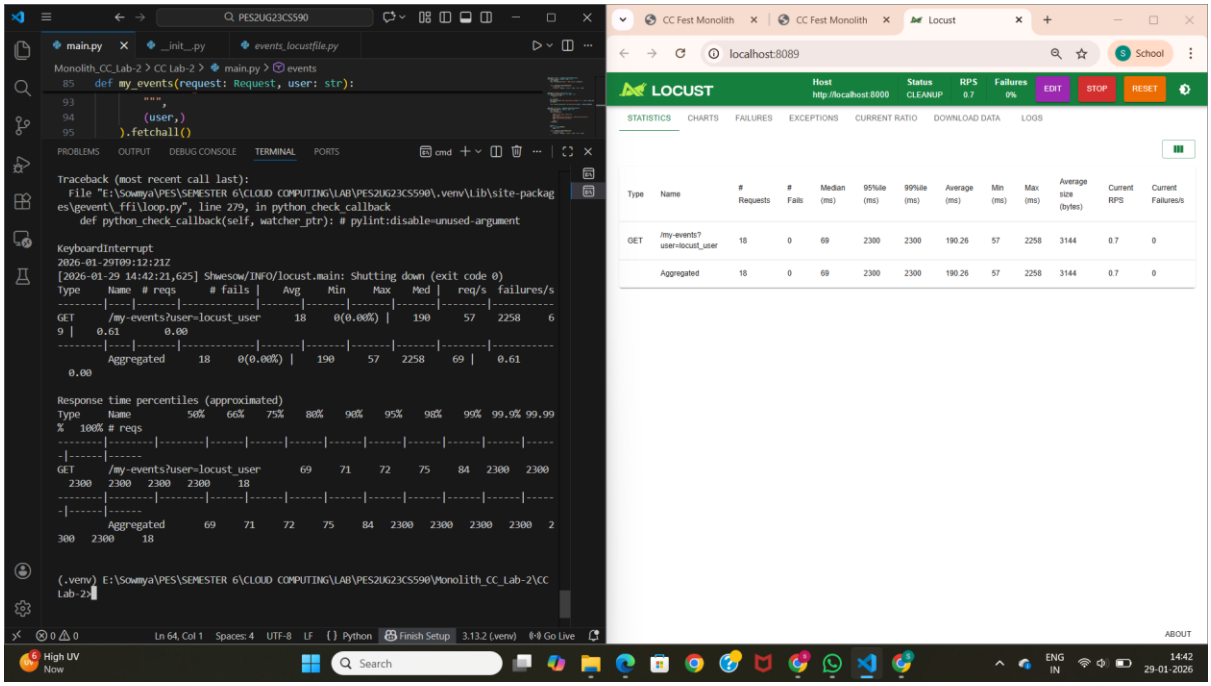


After Optimization:

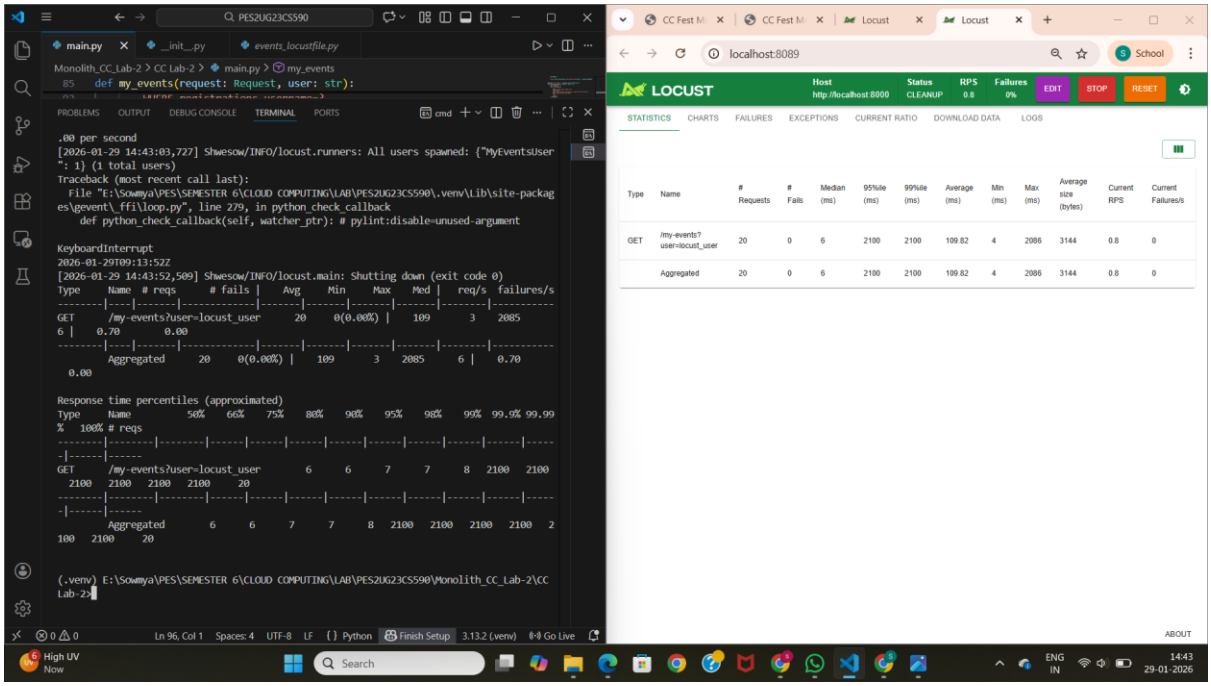


Route 2: /my-events

Before Optimization:



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.