**Assignment – 5**

Name : Om Kulkarni Date : 09/02/25

Class : TY CSE

PRN : 22510034 Batch : T-3

**Step 1: Define Test Objectives & Metrics**

Before running tests, clarify **what you want to measure**. Common performance metrics include:

| **Metric** | **Definition** |
| --- | --- |
| Response Time | Time taken to get a response from the server |
| Requests Per Second (RPS) | Number of requests handled per second |
| CPU & Memory Usage | How much system resources are used |
| Throughput | Amount of data transferred per second |
| Error Rate | Percentage of failed requests |
| Database Query Time | Time taken for a query to execute |

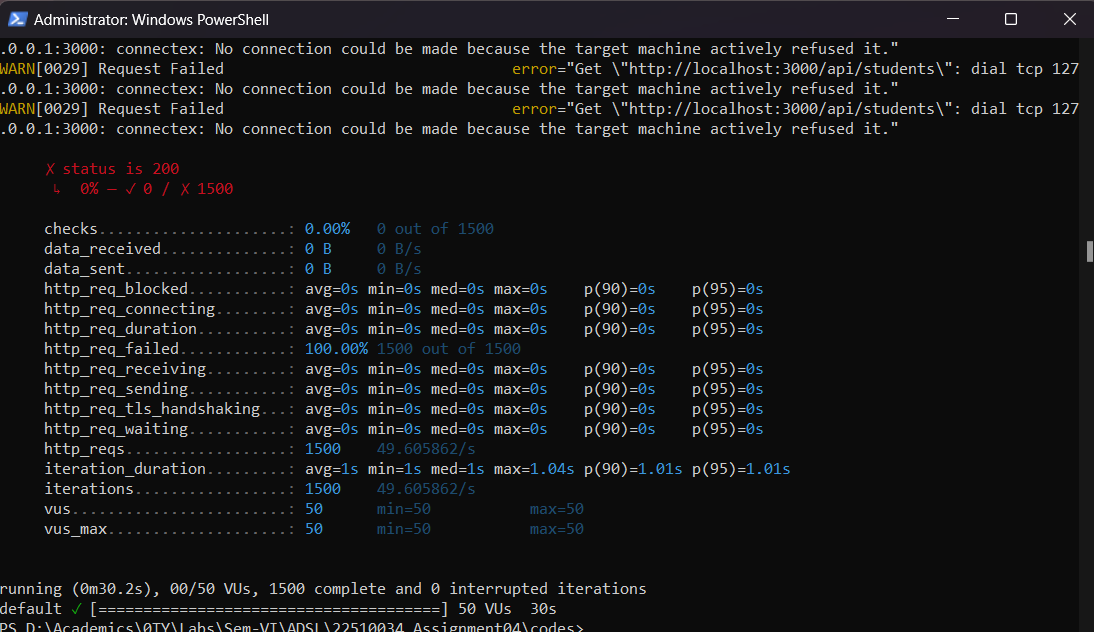
**Test Scenarios for Your Projects**

| **Scenario** | **Assignment 3 (MIS)** | **Assignment 4 (MCQ Exam System)** |
| --- | --- | --- |
| Load Test (100-1000 users) | Student data retrieval | Simultaneous quiz attempts |
| Stress Test (Pushing system to limit) | Bulk user creation | Large exam submissions |
| Database Query Performance | Fetching student records | Retrieving questions & results |
| API Response Time | CRUD operations | Fetching questions & saving answers |

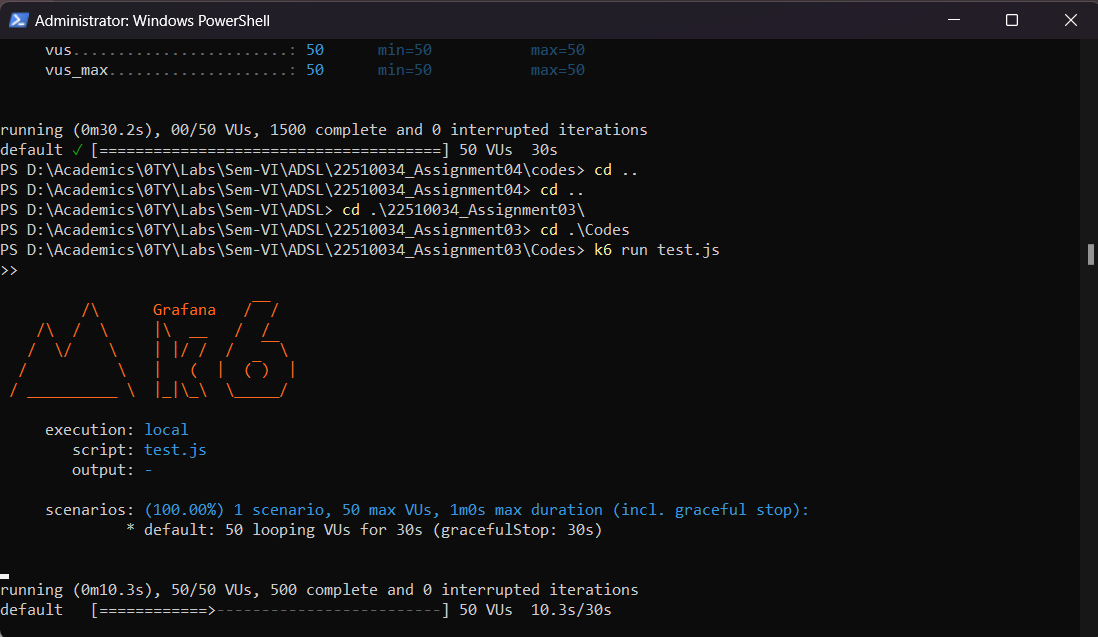
**Step 2: Select Open-Source Performance Testing Tools**

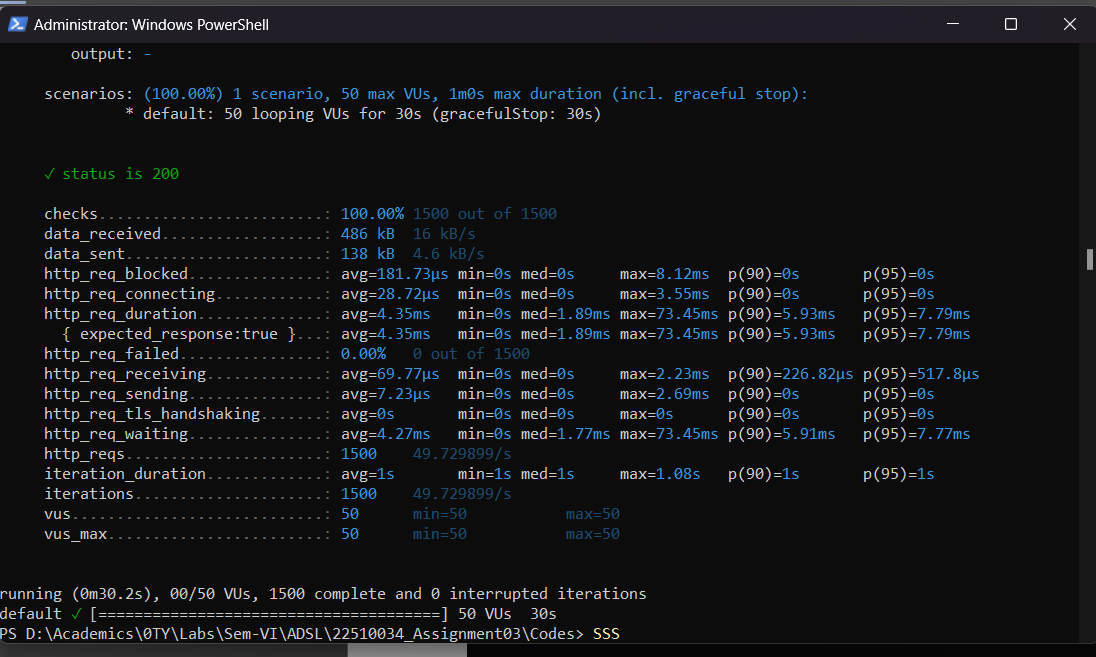
Choose tools based on what you're testing:

| **Tool** | **Best for** | **Usage** |
| --- | --- | --- |
| **JMeter** | API & load testing | Simulates multiple users sending requests |
| **k6** | Load & stress testing | Lightweight CLI tool for testing APIs |
| **Artillery** | API performance & stress testing | Easy-to-configure Node.js-based testing |
| **Lighthouse** | Frontend performance | Measures page speed, responsiveness, etc. |
| **MySQL Slow Query Log** | Database performance | Identifies slow queries in MySQL |
|  |  |  |

Assignment 4 Test Result using k6 tool   


Assignment 3 Test Result using k6 tool





|  |  |
| --- | --- |
| **This is test.js** |  |

import http from 'k6/http';

import { check, sleep } from 'k6';

export let options = {

vus: 50, // Simulate 50 users

duration: '30s', // Run test for 30 seconds

};

export default function () {

let res = http.get('http://localhost:3000/api/students'); // Change to your API

check(res, { 'status is 200': (r) => r.status === 200 });

sleep(1);

}