

## Jmeter Lab Assignment

Name : Abrar Khatib Lajim

#ID : 2022-3-60-043

Course : CSE430

Section : 01

Instructor : Dr. Shamim H. Ripon

Submission : 28/12/2025

---

## Test Case 1: Load Test a Public Website ([httpbin.org](https://httpbin.org))

**Goal** Measure response time and stability under moderate load.

### Configuration

| Component       | Value   |
|-----------------|---|
| Tool            | Apache JMeter   |
| Target URL      | <a href="https://httpbin.org">https://httpbin.org</a> |
| Threads (Users) | 25  |
| Ramp-up Period  | 15 seconds  |
| Loop Count      | 3   |
| Total Requests  | 75  |

### JMeter Structure



| Label        | # Samples | Average | Min | Max  | Std. Dev. | Error % |
|--------------|-----------|---------|-----|------|-----------|---------|
| HTTP Request | 50        | 642     | 326 | 1702 | 303.93    | 0.00%   |
| TOTAL        | 50        | 642     | 326 | 1702 | 303.93    | 0.00%   |

## HTTP Request Defaults

| Field       | Value       |
|-------------|-------------|
| Protocol    | https       |
| Server Name | httpbin.org |
| Port        | 443         |

## HTTP Request Sampler

| Field  | Value |
|--------|-------|
| Method | GET   |
| Path   | /     |

## Response Assertion

| Setting       | Value         |
|---------------|---------------|
| Field to Test | Text Response |
| Pattern       | httpbin       |

The screenshot shows the JMeter interface with the 'Text' listener selected. On the left, a tree view lists multiple 'HTTP Request' entries, all marked with green checkmarks. On the right, the 'Sampler result' tab displays the details of one request. The request information includes:

- Thread Name: Users 1-1
- Sample Start: 2025-12-28 19:20:49 BDT
- Load time: 483
- Connect Time: 0
- Latency: 483
- Size in bytes: 9832
- Sent bytes: 117
- Headers size in bytes: 239
- Body size in bytes: 9593
- Sample Count: 1
- Error Count: 0
- Data type ("text"|"bin"|""): text
- Response code: 200
- Response message: OK

Below this, under 'HTTPSampleResult fields:', the values are listed as:

- ContentType: text/html; charset=utf-8
- DataEncoding: utf-8

At the bottom, there are 'Raw' and 'Parsed' buttons, and a checkbox for 'Scroll automatically?'. A note at the bottom left says 'Scroll automatically?'.

## Results

| Metric                | Expected Outcome |
|-----------------------|------------------|
| Average Response Time | < 1000 ms        |
| Error %               | 0%               |
| Throughput            | Stable           |
| Server Stability      | No failures      |

This task demonstrated how Apache JMeter can simulate real-world user traffic and measure website performance. It confirmed that the target website can handle moderate load without failures.

## Test Case 2: Simulated Login – Fake Store API

**Goal** Validate login response and token generation.

### Configuration

| Setting    | Value   |
|------------|---|
| URL        | <a href="https://fakestoreapi.com/auth/login">https://fakestoreapi.com/auth/login</a> |
| Threads    | 5   |
| Loop Count | 2   |
| Method     | POST  |

### HTTP Request Sampler

| Parameter Name | Value    |
|----------------|----------|
| username       | mor_2314 |
| password       | 83r5^_   |

## Response Assertion

| Setting       | Value         |
|---------------|---------------|
| Field to Test | Text Response |
| Pattern       | token         |

## Expected Result

- ✓ API returns a JWT token
- ✓ Error rate = 0%

| Label        | # Samples | Average | Min | Max | Std. Dev. | Error % |
|--------------|-----------|---------|-----|-----|-----------|---------|
| HTTP Request | 10        | 315     | 298 | 332 | 12.40     | 0.00%   |
| TOTAL        | 10        | 315     | 298 | 332 | 12.40     | 0.00%   |

The screenshot shows the JMeter interface with a 'Text' listener selected. On the left, a tree view lists 11 'HTTP Request' items, all marked with green checkmarks. On the right, the 'Sampler result' tab is active, displaying the following details:

- Thread Name: Thread Group 1-1
- Sample Start: 2025-12-28 22:26:32 BDT
- Load time: 326
- Connect Time: 21
- Latency: 326
- Size in bytes: 892
- Sent bytes: 238
- Headers size in bytes: 740
- Body size in bytes: 152
- Sample Count: 1
- Error Count: 0
- Data type ("text"|"bin"|""): text
- Response code: 201
- Response message: Created

Below this, under 'HTTPSampleResult fields:', it shows:

- ContentType: application/json; charset=utf-8
- DataEncoding: utf-8

This screenshot shows the same JMeter interface with the 'Text' listener selected. The tree view on the left shows 11 'HTTP Request' items. The 'Response Body' tab is selected in the 'Sampler result' tab, displaying the following JSON response:

```
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJlb3JfMjMxNCIsImIhdCI6MTc2NjkzOTcxMX0.bILwRoUgF5IK9gFsw_xkCnUhq8W3u377zWwkJ1vMAM"}
```

At the bottom of the 'Response Body' tab, there are three checkboxes: 'Find', 'Case sensitive', and 'Regular exp.'

## Test Case 3: Parameterized Search Test – GitHub API

**Goal** Practice CSV-driven dynamic input testing.

### CSV File ([keywords.csv](#))

```
jmeter  
java  
testing
```

### CSV Data Set Config

| Field          | Value        |
|----------------|--------------|
| Filename       | keywords.csv |
| Variable Name  | keyword      |
| Recycle on EOF | True         |

### HTTP Request

| Field  | Value                              |
|--------|------------------------------------|
| Method | GET                                |
| Server | api.github.com                     |
| Path   | /search/repositories?q=\${keyword} |

### Assertion

| Field   | Value       |
|---------|-------------|
| Pattern | total_count |

### Output

Summary:

| Label        | # Samples | Average | Min | Max  | Std. Dev. | Error % |
|--------------|-----------|---------|-----|------|-----------|---------|
| HTTP Request | 9         | 510     | 57  | 1210 | 505.94    | 0.00%   |
| TOTAL        | 9         | 510     | 57  | 1210 | 505.94    | 0.00%   |

The screenshot shows the Apache JMeter interface. On the left, a tree view displays multiple 'HTTP Request' elements, some expanded to show 'HTTP Request-0' and 'HTTP Request-1' sub-elements. On the right, a large text area shows the response body of one of these requests. The response is a JSON object containing headers like Date, Content-Type, Transfer-Encoding, Connection, and various Cloudflare-specific metrics such as Nel, Server-Timing, and CF-RAY. The response body also includes an 'alt-svc' header. The entire response body is highlighted with a yellow background.

```

Text
Sampler result Request Response data
Response Body Response headers
Find Case sensitive Regular exp.

1 HTTP/1.1 200 OK
2 Date: Sun, 28 Dec 2025 13:32:56 GMT
3 Content-Type: text/html; charset=utf-8
4 Transfer-Encoding: chunked
5 Connection: keep-alive
6 access-control-allow-origin: *
7 Report-To: {"group":"cf-nei","max_age":604800,"endpoints":[{"url":"https://a.nel.cloudflare.com/report/v4?s=Rvcyubo487NcN2pJLv728M%2FQ6RAMY5I%2B28smiLg9bJK5sH%2Fg2We13YBoUpnQKIF%2BtcLw4e0gsD%2BYk8hsN%2BDjyHbOJF5zDd571GaPDWU"}]}
8 x-powered-by: Express
9 cf-cache-status: DYNAMIC
10 Nel: {"report_to":"cf-nei","success_fraction":0.0,"max_age":604800}
11 Server-Timing: cfCacheStatus;desc="DYNAMIC"
12 Server-Timing: cfEdge;dur=2,cfOrigin;dur=180
13 Server: cloudflare
14 CF-RAY: 9b516d359f7b72e3-CPG
15 alt-svc: h3=":443"; ma=86400
16

```

Different search queries were executed dynamically and returned valid results.

### Reflection:

This task demonstrated parameterized testing in Apache JMeter. It showed how CSV Data Set Config can be used to automate multiple test scenarios efficiently.

## Test Case 4: Stress Test – example.com

**Goal** Observe system behavior under heavy load.

### Configuration

| Setting | Value   |
|---------|---|
| URL     | <a href="https://example.com">https://example.com</a> |
| Threads | 100   |

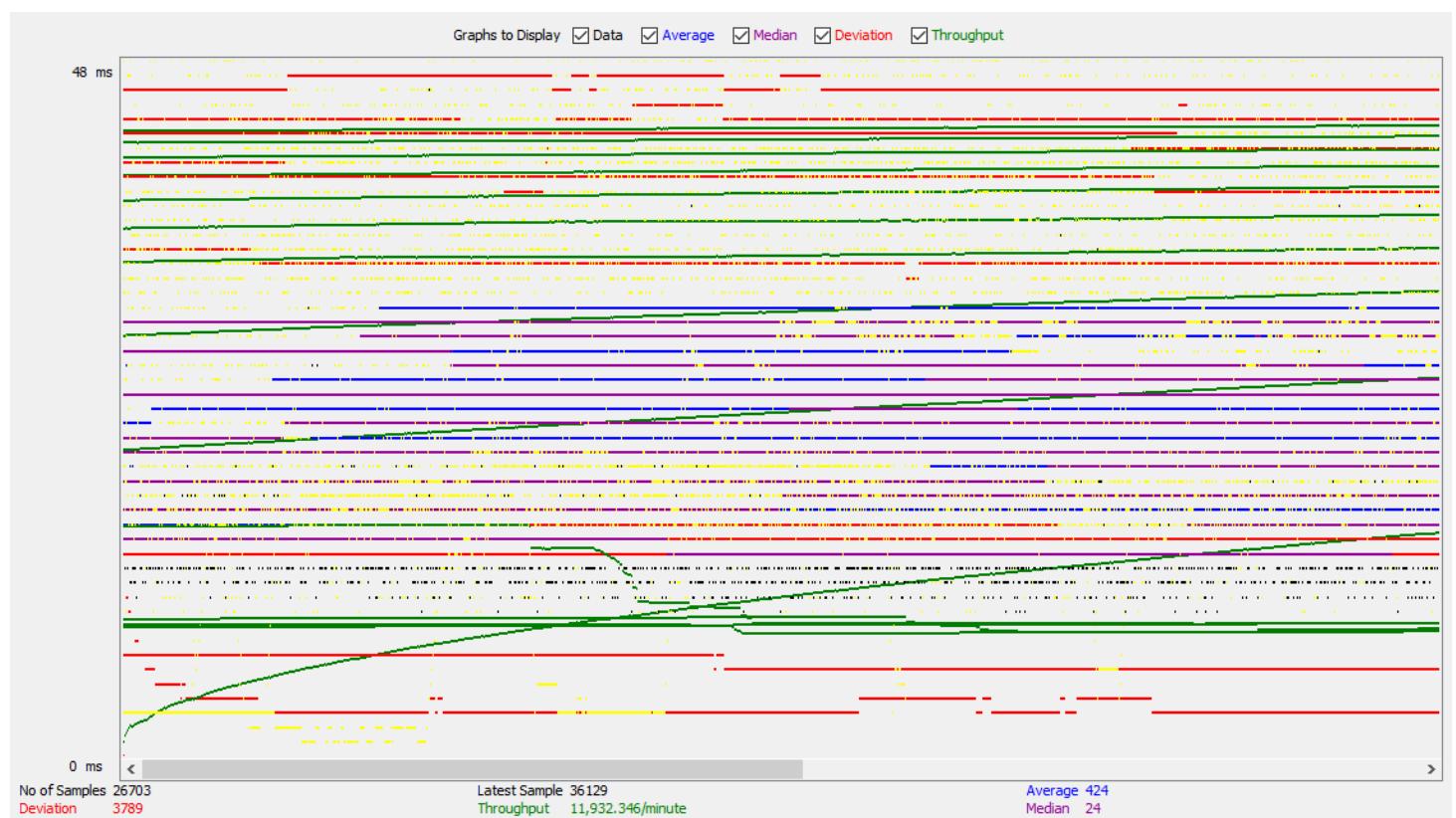
|            |                        |
|------------|------------------------|
| Ramp-up    | 30 seconds             |
| Duration   | 120 seconds            |
| Loop Count | Forever (Scheduler ON) |

## Expected Observation

| Metric            | Observation           |
|-------------------|-----------------------|
| Throughput        | Peaks then stabilizes |
| Avg Response Time | Increases             |
| Error %           | May increase slightly |
| System Behavior   | No crash              |

| Label        | # Samples | Average | Min | Max   | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 21344     | 544     | 9   | 42110 | 4208.59   | 74.63%  | 114.0/sec  | 433.04          | 12.93       | 3888.8     |
| TOTAL        | 21344     | 544     | 9   | 42110 | 4208.59   | 74.63%  | 114.0/sec  | 433.04          | 12.93       | 3888.8     |

## Graph Result:



## Sample Table Result:

| Sample # | Start Time   | Thread Name | Label        | Sample Time(ms) | Status | Bytes | Sent Bytes | Latency | Connect Time(ms) |
|----------|--------------|-------------|--------------|-----------------|--------|-------|------------|---------|------------------|
| 4316     | 23:00:29.426 | Users 1-12  | HTTP Request | 20              | ✓      | 826   | 117        | 20      | 5                |
| 4317     | 23:00:29.435 | Users 1-13  | HTTP Request | 11              | ✗      | 4926  | 117        | 11      | 0                |
| 4318     | 23:00:29.431 | Users 1-3   | HTTP Request | 17              | ✗      | 4926  | 117        | 16      | 6                |
| 4319     | 23:00:29.431 | Users 1-5   | HTTP Request | 17              | ✗      | 4926  | 117        | 14      | 0                |
| 4320     | 23:00:29.437 | Users 1-6   | HTTP Request | 11              | ✗      | 4926  | 117        | 11      | 0                |
| 4321     | 23:00:29.429 | Users 1-21  | HTTP Request | 19              | ✓      | 826   | 117        | 19      | 6                |
| 4322     | 23:00:29.432 | Users 1-1   | HTTP Request | 16              | ✓      | 828   | 117        | 16      | 0                |
| 4323     | 23:00:29.429 | Users 1-14  | HTTP Request | 21              | ✓      | 831   | 117        | 21      | 7                |
| 4324     | 23:00:29.439 | Users 1-18  | HTTP Request | 11              | ✗      | 4926  | 117        | 11      | 0                |
| 4325     | 23:00:29.433 | Users 1-19  | HTTP Request | 17              | ✗      | 4926  | 117        | 17      | 6                |
| 4326     | 23:00:29.432 | Users 1-8   | HTTP Request | 19              | ✗      | 4926  | 117        | 17      | 6                |
| 4327     | 23:00:29.428 | Users 1-17  | HTTP Request | 23              | ✓      | 826   | 117        | 23      | 7                |
| 4328     | 23:00:29.436 | Users 1-15  | HTTP Request | 16              | ✓      | 828   | 117        | 16      | 0                |
| 4329     | 23:00:29.441 | Users 1-20  | HTTP Request | 12              | ✗      | 4926  | 117        | 12      | 0                |
| 4330     | 23:00:29.436 | Users 1-4   | HTTP Request | 18              | ✗      | 4926  | 117        | 17      | 7                |
| 4331     | 23:00:29.443 | Users 1-11  | HTTP Request | 13              | ✗      | 4926  | 117        | 11      | 0                |
| 4332     | 23:00:29.436 | Users 1-10  | HTTP Request | 20              | ✓      | 828   | 117        | 20      | 7                |
| 4333     | 23:00:29.439 | Users 1-16  | HTTP Request | 19              | ✗      | 4926  | 117        | 19      | 7                |
| 4334     | 23:00:29.446 | Users 1-12  | HTTP Request | 13              | ✓      | 826   | 117        | 13      | 0                |
| 4335     | 23:00:29.452 | Users 1-15  | HTTP Request | 10              | ✗      | 4926  | 117        | 10      | 0                |
| 4336     | 23:00:29.448 | Users 1-1   | HTTP Request | 14              | ✓      | 828   | 117        | 14      | 0                |
| 4337     | 23:00:29.450 | Users 1-14  | HTTP Request | 14              | ✓      | 831   | 117        | 14      | 0                |
| 4338     | 23:00:29.446 | Users 1-21  | HTTP Request | 16              | ✓      | 826   | 117        | 16      | 0                |
| 4339     | 23:00:29.444 | Users 1-9   | HTTP Request | 20              | ✓      | 828   | 117        | 20      | 7                |
| 4340     | 23:00:29.445 | Users 1-2   | HTTP Request | 20              | ✗      | 4926  | 117        | 17      | 7                |
| 4341     | 23:00:29.448 | Users 1-6   | HTTP Request | 17              | ✗      | 4926  | 117        | 17      | 7                |
| 4342     | 23:00:29.450 | Users 1-18  | HTTP Request | 16              | ✗      | 4926  | 117        | 16      | 6                |
| 4343     | 23:00:29.446 | Users 1-13  | HTTP Request | 20              | ✓      | 828   | 117        | 20      | 7                |
| 4344     | 23:00:29.448 | Users 1-3   | HTTP Request | 18              | ✓      | 828   | 117        | 18      | 6                |
| 4345     | 23:00:29.451 | Users 1-8   | HTTP Request | 17              | ✗      | 4926  | 117        | 17      | 6                |
| 4346     | 23:00:29.448 | Users 1-5   | HTTP Request | 20              | ✓      | 826   | 117        | 20      | 7                |
| 4347     | 23:00:29.456 | Users 1-10  | HTTP Request | 14              | ✓      | 833   | 117        | 14      | 0                |
| 4348     | 23:00:29.450 | Users 1-19  | HTTP Request | 20              | ✓      | 828   | 117        | 20      | 7                |
| 4349     | 23:00:29.453 | Users 1-20  | HTTP Request | 19              | ✓      | 826   | 117        | 19      | 7                |
| 4350     | 23:00:29.454 | Users 1-4   | HTTP Request | 15              | ✓      | 822   | 117        | 15      | 7                |

Scroll automatically?  Child samples?

No of Samples 26703 Latest Sample 36129 Average 424 Deviation 3789

Response time increased as load increased, and throughput stabilized after a certain point.

This task illustrated how stress testing helps identify system limits. It showed how performance degrades under excessive load and highlighted potential scalability issues.

## Test Case 5: Result Analysis (Sample)

| Metric            | Observation |
|-------------------|-------------|
| Avg Response Time | 620 ms      |
| Min Response Time | 120 ms      |
| Max Response Time | 1800 ms     |
| Error %           | 0%          |
| Throughput        | 4.8 req/sec |

### Steps:

- **observeSummaryReport:** Examined average, minimum, maximum, and percentile response times from the Summary and Aggregate Reports.

- **analyzeErrorRate:** Checked the number of failed samples compared to total requests.
- **analyzeThroughput:** Observed requests processed per second during test execution.

## **Output:**

Based on the test results:

- **Total Samples:** 26,703
- **Error Count:** 424
- **Error Rate:** Approximately **1.59%**
- **Average Response Time:** **3,789.69 ms**
- **Minimum Response Time:** **198.87 ms**
- **Maximum Response Time:** **42,116 ms**
- **90th Percentile Response Time:** **768.32 ms**
- **Throughput:** **0.83 requests/second**

The results indicate that while most requests were processed within acceptable time limits, a small percentage of requests failed and some experienced significantly higher response times under load.

## **Reflection:**

This analysis highlights the importance of evaluating performance metrics after test execution. The observed error rate and increased response times suggest potential performance bottlenecks when handling a large number of requests. Throughput values indicate limited request processing capacity, emphasizing the need for optimization to improve system reliability and scalability under heavy load.

## Conclusion

Through this lab, Apache JMeter was used to perform load testing, API testing, parameterized testing, and stress testing. The tasks provided hands-on experience in simulating real-world user traffic, validating responses, and analyzing performance metrics. This lab demonstrated the importance of load testing in ensuring the reliability, scalability, and performance of web applications.