

Introduction

Apache JMeter is a popular open-source tool utilized for assessing the performance, load capacity, and behavior of web applications. The primary objective of this lab report was to develop a wide JMeter Test Plan. This involved configuring Thread Groups, HTTP Samplers, Assertions, Listeners, Loop Controllers, and Timers.

For this lab report, I selected Facebook and LinkedIn as the example web applications. I aimed to simulate numerous users interacting with Facebook and LinkedIn to observe their response under pressure and to gain hands-on experience with a variety of JMeter components functioning together in a structured testing environment.

Test Plan Design

The test plan includes four separate Thread Groups. Each one shows specific JMeter functions. In this lab report, I use some key elements of JMeter and configure some parameters as per the instructions to build this test plan. These components and configures include:

Components :

JMeter Components	Purpose
<ul style="list-style-type: none">• Thread Groups	<ul style="list-style-type: none">• Indicate total number of virtual users and executed setting
<ul style="list-style-type: none">• HTTP Samplers	<ul style="list-style-type: none">• HTTPS/HTTP request will be making requests to our server
<ul style="list-style-type: none">• Listeners	<ul style="list-style-type: none">• Results generated will be retrieved and displayed
<ul style="list-style-type: none">• Assertions	<ul style="list-style-type: none">• Check all received responses against expected responses.
<ul style="list-style-type: none">• Loop Controllers	<ul style="list-style-type: none">• Number of times sampler will be sent based on defined repetitions.
<ul style="list-style-type: none">• Timer	<ul style="list-style-type: none">• Simulated real user actions by including realistic wait times between requests.

Configurations (Based on my ID: 232-35-268):

1. Number of Users (Threads): 8 (Last digit = 8)
2. Ramp-up Period: 6 seconds (2nd last digit = 6)
3. Loop Count: 2 (3rd last digit = 2)
4. Timer Delay: 6000 ms (2nd last digit = 6)
5. Loop Controller Count: 8 (Last digit = 8)

Thread Group 1:

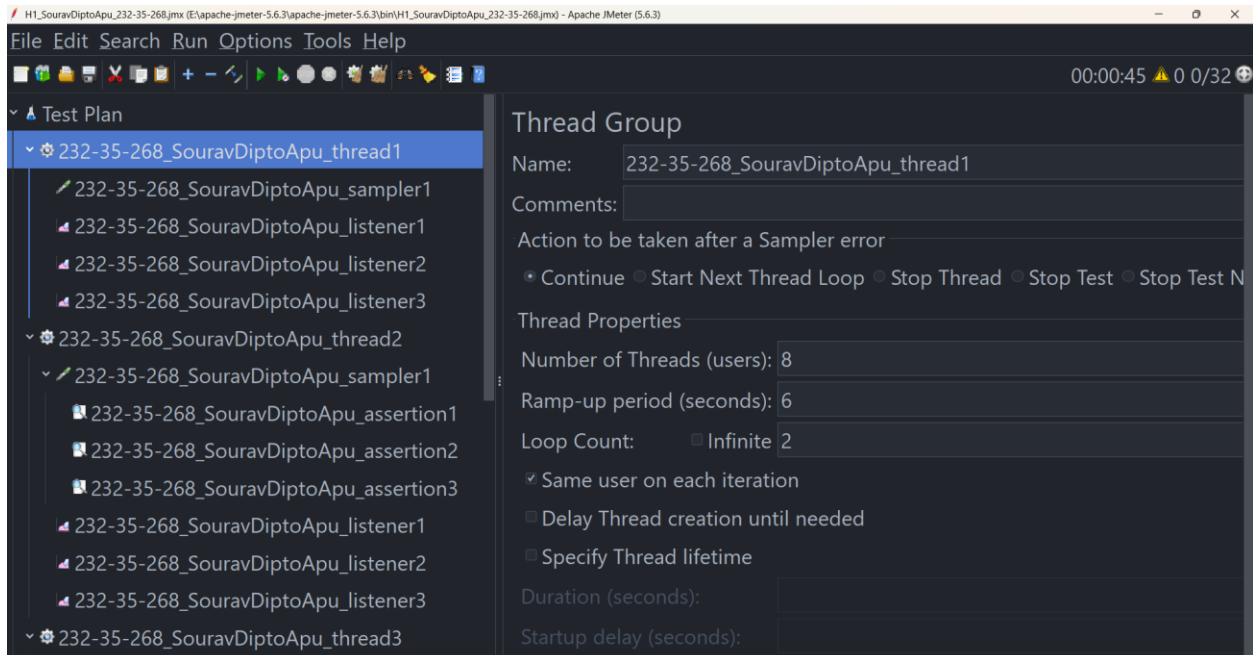
Name: 232-35-268_SouravDiptoApu_thread1

Purpose: To execute basic HTTP requests with configured user load and visualize results using multiple listeners.

Configuration:

Number of Users: 8
Ramp-up Period: 6 seconds
Loop Count: 2

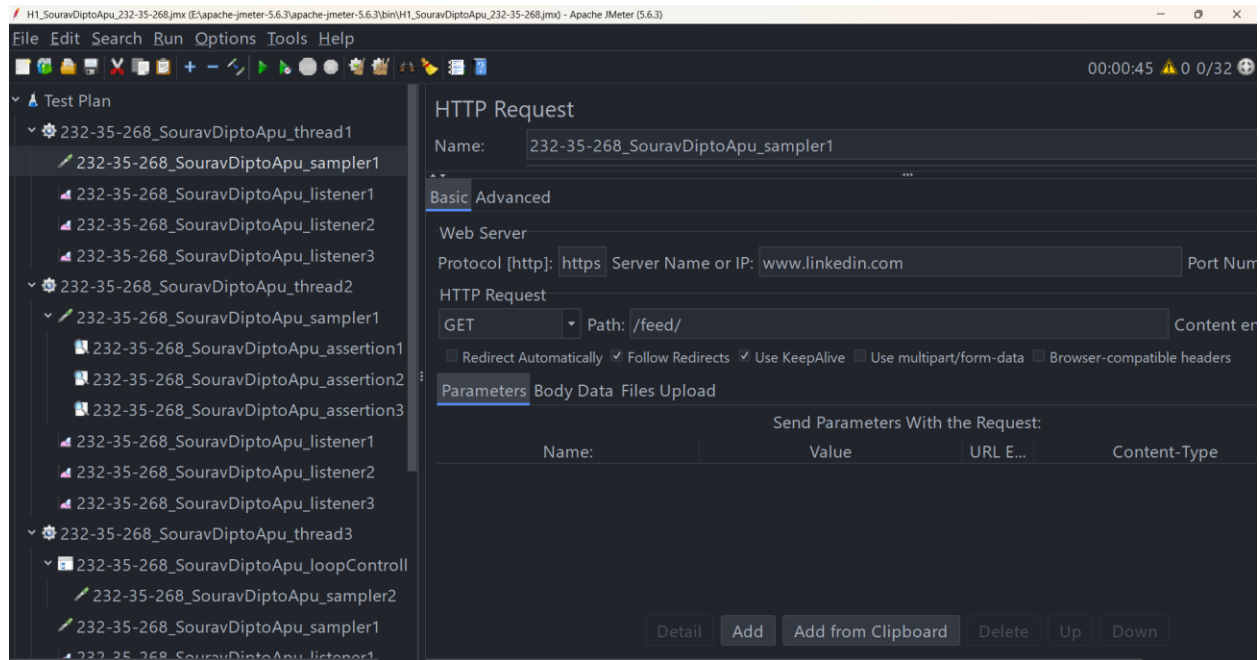
Thread Group 01 Screenshot:



Elements Included:

HTTP Sampler:
Name: 232-35-268_SouravDiptoApu_sampler1
Purpose: Send GET request to linkedin feed.
URL: https://www.linkedin.com
Path: /feed/

HTTP Sampler Screenshot:



Listeners:

232-35-268_SouravDiptoApu_listener1
232-35-268_SouravDiptoApu_listener2
232-35-268_SouravDiptoApu_listener3

Listeners screenshot (including all Listeners) :

H1_SouravDiptoApu_232-35-268.jmx (E:\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\H1_SouravDiptoApu_232-35-268.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:45 0 0/32

Test Plan

- 232-35-268_SouravDiptoApu_thread1
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_listener1**
 - 232-35-268_SouravDiptoApu_listener2
 - 232-35-268_SouravDiptoApu_listener3
- 232-35-268_SouravDiptoApu_thread2
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_assertion1
 - 232-35-268_SouravDiptoApu_assertion3
 - 232-35-268_SouravDiptoApu_assertion2
 - 232-35-268_SouravDiptoApu_listener1
 - 232-35-268_SouravDiptoApu_listener2
 - 232-35-268_SouravDiptoApu_listener3
- 232-35-268_SouravDiptoApu_thread3
 - 232-35-268_SouravDiptoApu_loopController
 - 232-35-268_SouravDiptoApu_sampler2
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_listener1

View Results Tree

Name: 232-35-268_SouravDiptoApu_listener1

Comments:

Write results to file / Read from file

Search: ☐ Case sensitive ☐ Regular exp.

Text Request Response data

H1_SouravDiptoApu_232-35-268.jmx (E:\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\H1_SouravDiptoApu_232-35-268.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:45 0 0/32

Test Plan

- 232-35-268_SouravDiptoApu_thread1
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_listener1
 - 232-35-268_SouravDiptoApu_listener2**
 - 232-35-268_SouravDiptoApu_listener3
- 232-35-268_SouravDiptoApu_thread2
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_assertion1
 - 232-35-268_SouravDiptoApu_assertion3
 - 232-35-268_SouravDiptoApu_assertion2
 - 232-35-268_SouravDiptoApu_listener1
 - 232-35-268_SouravDiptoApu_listener2
 - 232-35-268_SouravDiptoApu_listener3
- 232-35-268_SouravDiptoApu_thread3
 - 232-35-268_SouravDiptoApu_loopController
 - 232-35-268_SouravDiptoApu_sampler2
 - 232-35-268_SouravDiptoApu_sampler1
 - 232-35-268_SouravDiptoApu_listener1

View Results in Table

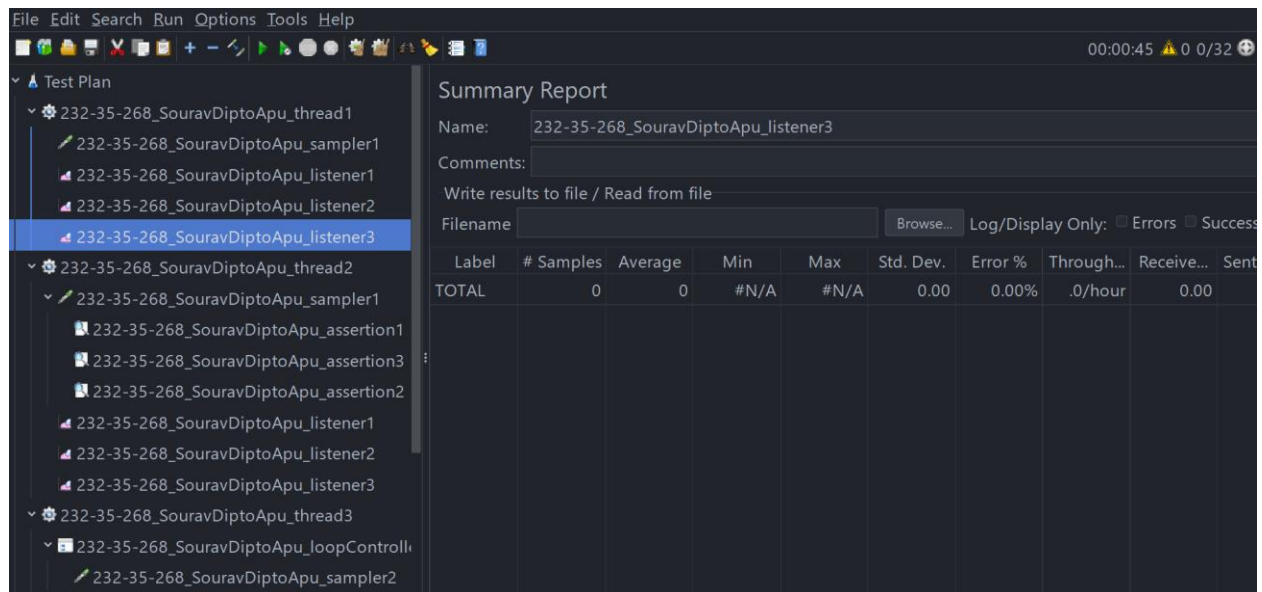
Name: 232-35-268_SouravDiptoApu_listener2

Comments:

Write results to file / Read from file

Filename

Sample #	Start Time	Thread Na...	Label	Sample Tim...	Status	Bytes	Sen
----------	------------	--------------	-------	---------------	--------	-------	-----



Thread Group 2:

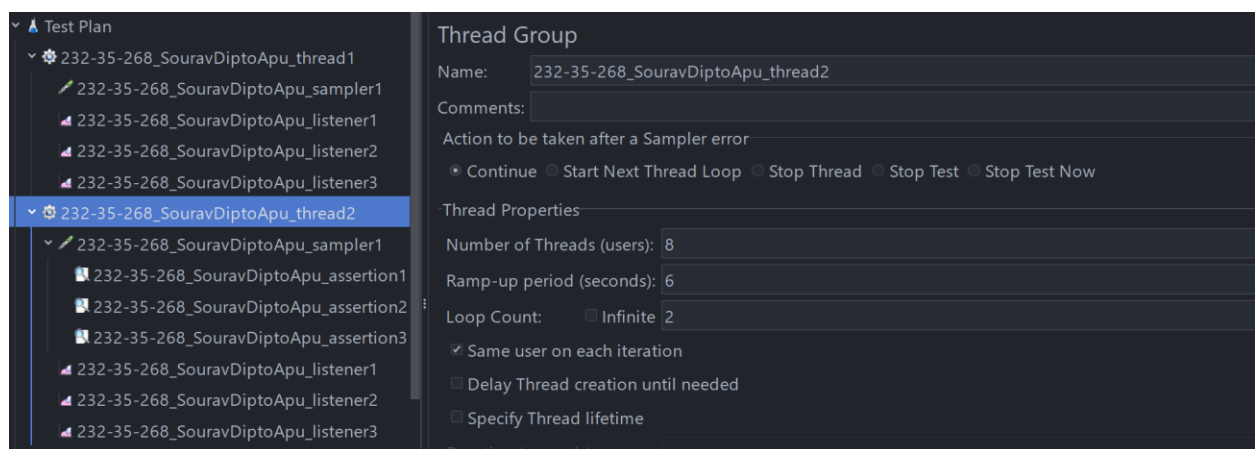
Name: `232-35-268_SouravDiptoApu_thread2`

Purpose: To verify the correctness of the server response by using different types of assertions during the performance test.

Configuration Details:

Number of Users (Threads): 8
Ramp-up Period: 6 seconds
Loop Count: 2

Screenshot:



HTTP Sampler:

Name: 232-35-268_SouravDiptoApu_sampler1

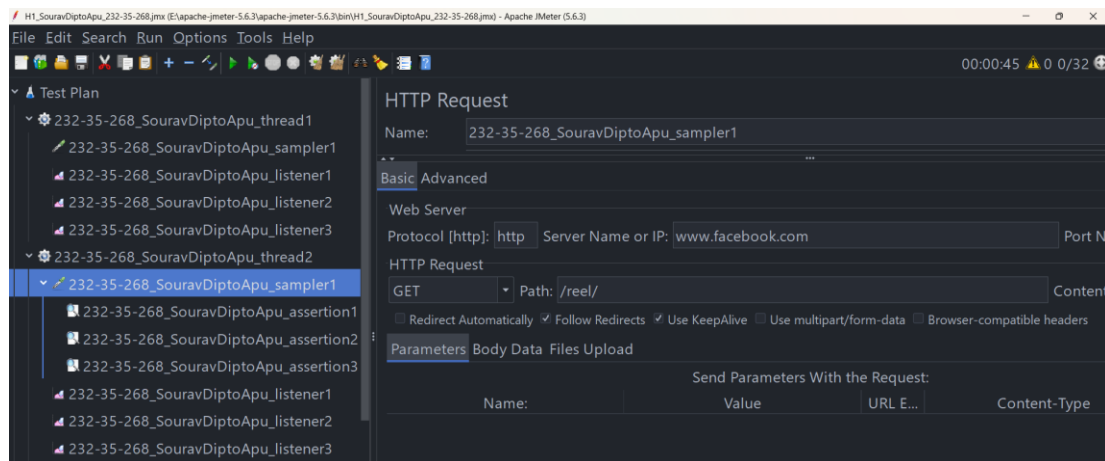
Purpose in Test Plan:

To send an HTTP GET request to the Facebook homepage and collect the server response for validation.

Configuration Details:

Protocol: https
Server Name: www.facebook.com
HTTP Method: GET
Path: /reel/

Screenshot:



Assertions:

1. Response Assertion

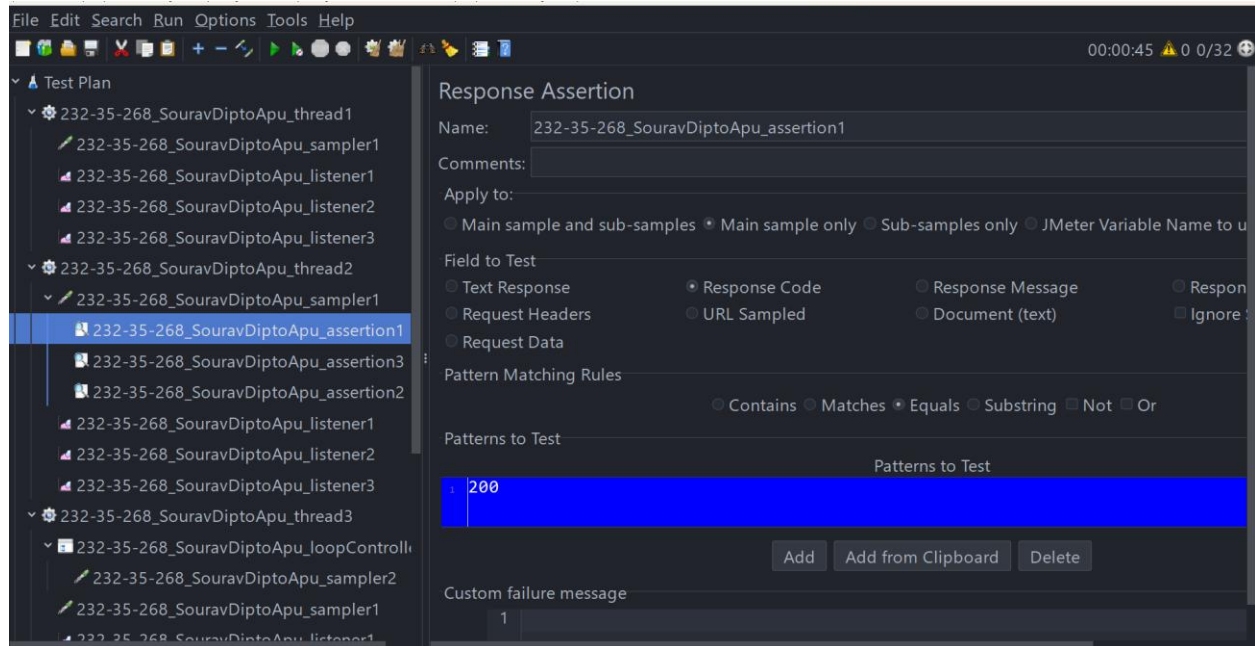
Name: 232-35-268_SouravDiptoApu_assertion1

Purpose in Test Plan: To check whether the server response contains the expected text.

Configuration Details:

Test Field: Response Body
Pattern Matching Rule: Contains
Expected Text: Facebook

Screenshot:



2. Duration Assertion (Intentional Failure)

Name: 232-35-268_SouravDiptoApu_assertion3

Purpose in Test Plan: To ensure that the server response time does not exceed the specified limit.

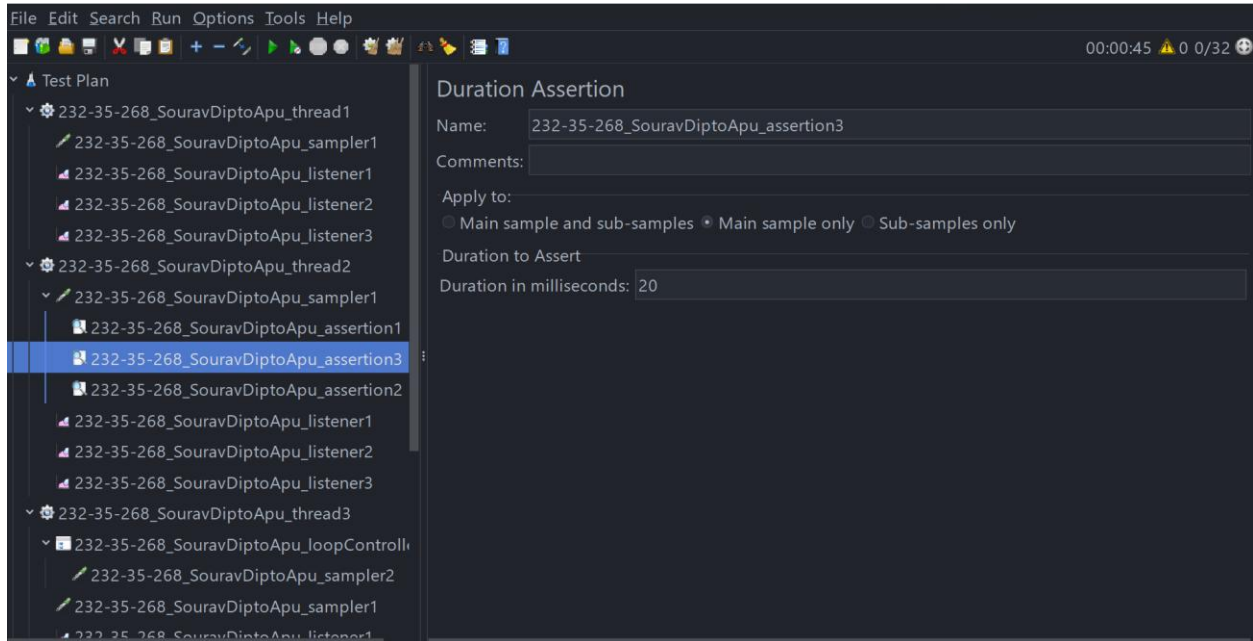
Configuration Details:

Maximum Response Time: 20 ms

Explanation of Duration Assertion Error:

Because the maximum permitted response time was set at just 20 milliseconds, which is unreasonably low for big web apps like Facebook, linkedin, the Duration Assertion failed. As anticipated, the assertion failed because the actual response time exceeded the configured limit.

Screenshot:



3. Size Assertion:

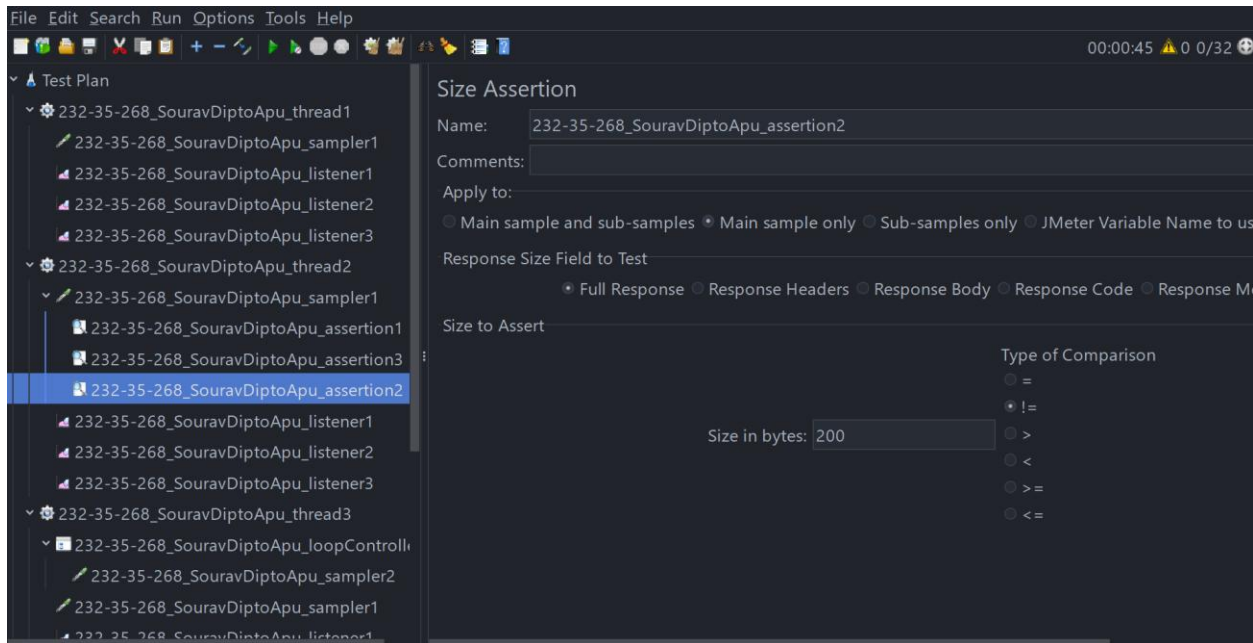
Name: 232-35-268_SouravDiptoApu_assertion2

Purpose in Test Plan: To validate the size of the server response.

Configuration Details:

Expected Size (Bytes): 200
Condition: Not Equals

Screenshot:



Listeners

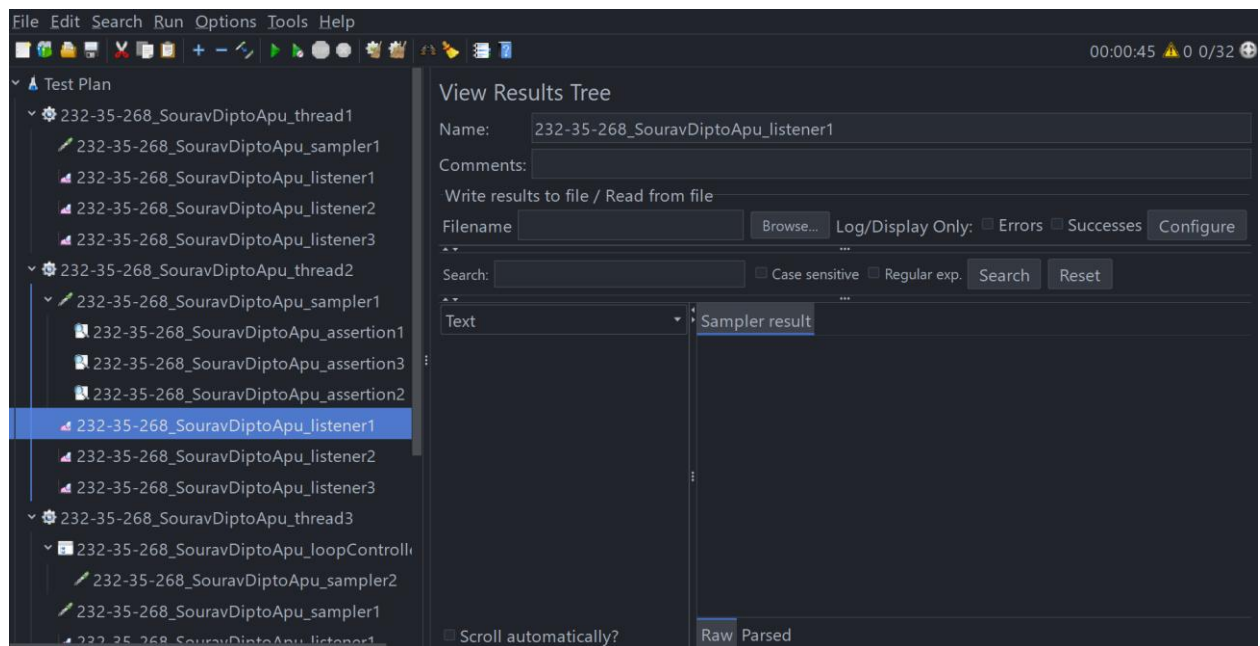
View Results Tree:

Name: 232-35-268_SouravDiptoApu_listener1

Purpose in Test Plan: To display detailed request and response information for each sample.

Configuration Details: Default

Screenshot:



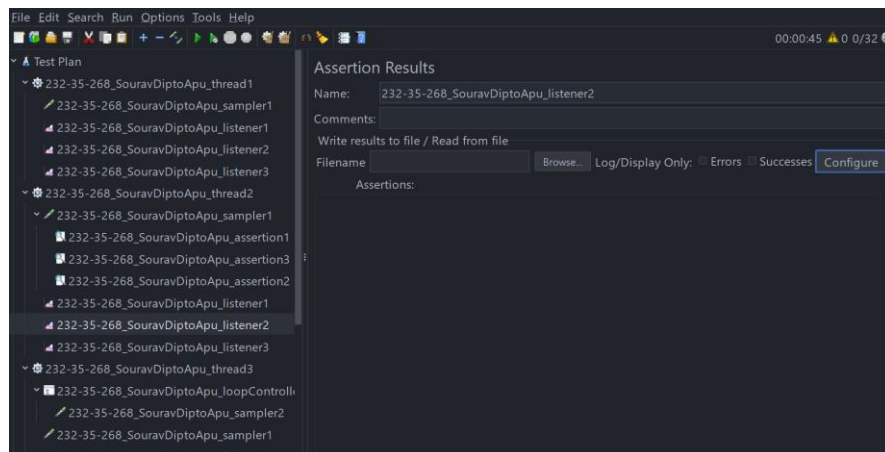
Assertion Results:

Name: 232-35-268_SouravDiptoApu_listener2

Purpose in Test Plan: To show which assertions pass or fail during execution.

Configuration Details: Default

Screenshot:



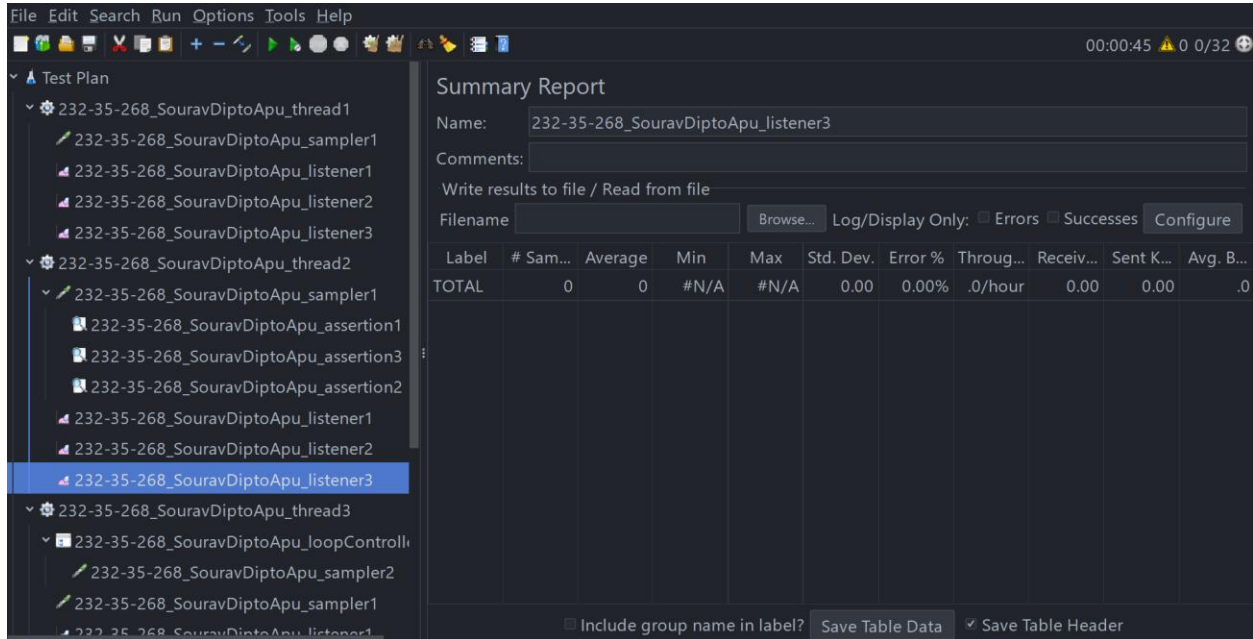
Summary Report:

Name: 232-35-268_SouravDiptoApu_listener

Purpose in Test Plan: To summarize performance metrics like average time, throughput, and error rate.

Configuration Details: Default

Screenshot:

The screenshot shows a software interface with a dark theme. On the left is a 'Test Plan' tree view containing several items, including '232-35-268_SouravDiptoApu_thread1', '232-35-268_SouravDiptoApu_thread2', and '232-35-268_SouravDiptoApu_thread3'. The right pane is titled 'Summary Report' and displays a table of performance metrics. The table has columns: Label, # Sam..., Average, Min, Max, Std. Dev., Error %, Throug..., Receiv..., Sent K..., and Avg. B... The only row of data is 'TOTAL' with values: 0, 0, #N/A, #N/A, 0.00, 0.00%, .0/hour, 0.00, 0.00, and .0. Above the table are fields for Name (232-35-268_SouravDiptoApu_listener3), Comments, and a section for writing results to file. At the bottom of the table are checkboxes for 'Include group name in label?', 'Save Table Data', and 'Save Table Header'.

Thread Group 3

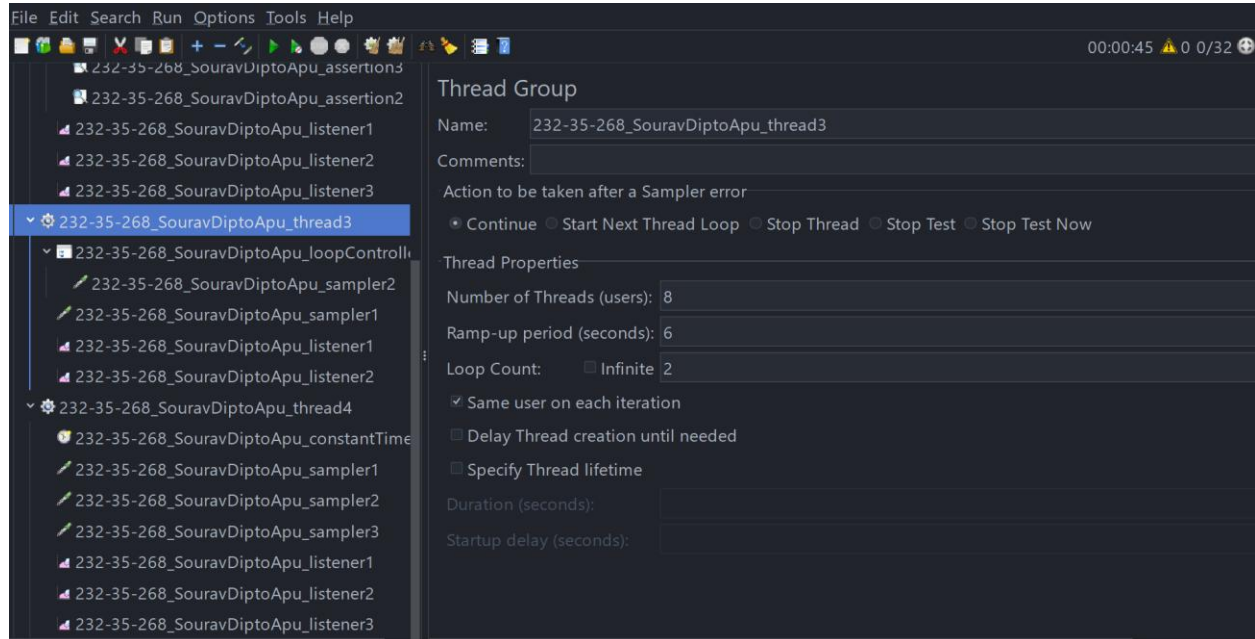
Name: 232-35-268_SouravDiptoApu_thread3

Purpose in Test Plan: To repeat a specific HTTP request multiple times inside a Loop Controller.

Configuration Details:

Number of Users (Threads): 8
Ramp-up Period: 6 seconds
Loop Count: 2

Screenshot:



HTTP Sampler

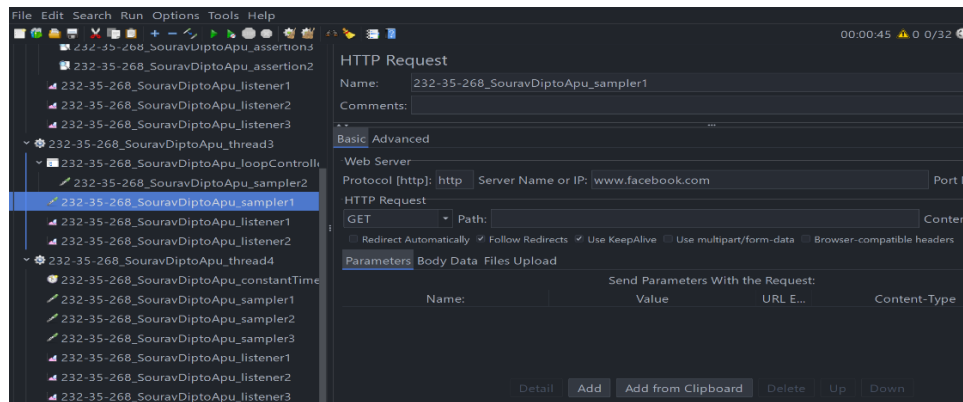
Name: 232-35-268_SouravDiptoApu_sampler1

Purpose in Test Plan: To send a GET request to Facebook homepage before the Loop Controller executes.

Configuration Details:

Protocol: https
Server Name: www.facebook.com
HTTP Method: GET
Path: N/A

Screenshot:



Loop Controller

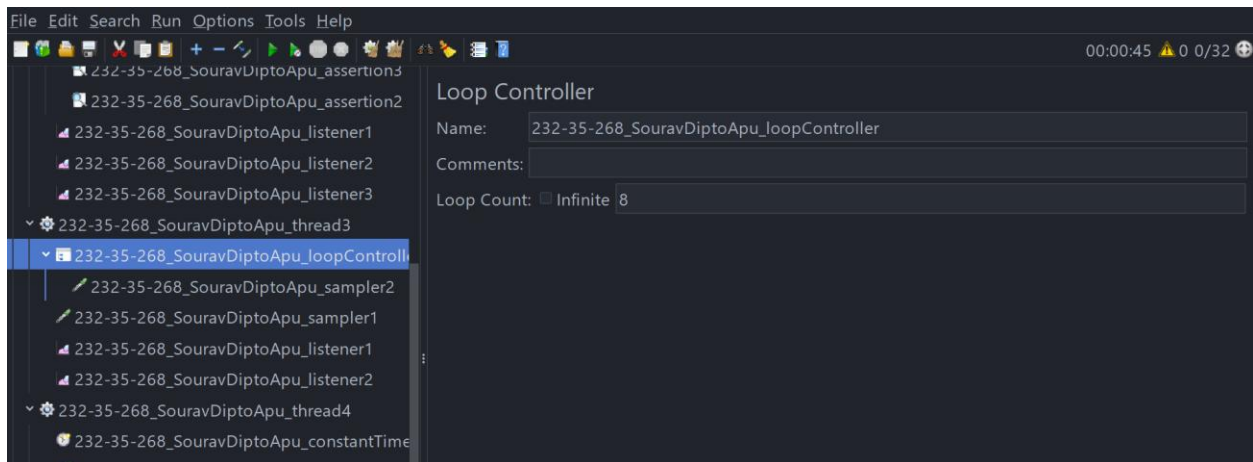
Name: 232-35-268_SouravDiptoApu_loopController

Purpose in Test Plan: To repeat a sampler multiple times according to Loop Count.

Configuration Details:

Loop Count: 8 (last digit of student ID)
--

Screenshot:



HTTP Sampler inside Loop Controller

Name: 232-35-268_SouravDiptoApu_sampler2

Purpose in Test Plan: To repeatedly send a GET request to Facebook homepage inside the loop.

Configuration Details:

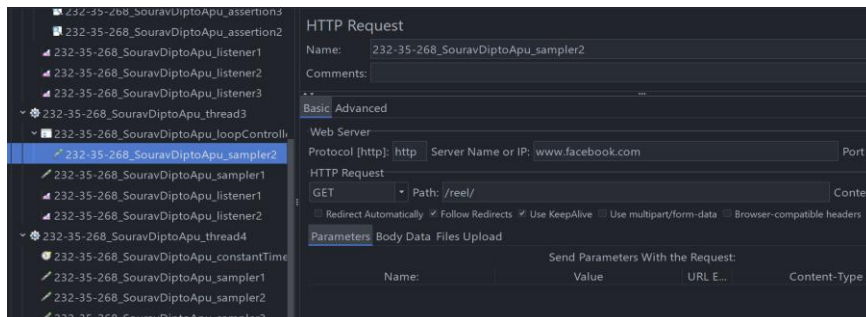
Protocol: https

Server Name: www.facebook.com

HTTP Method: GET

Path: /reel/

Screenshot:



Listeners:

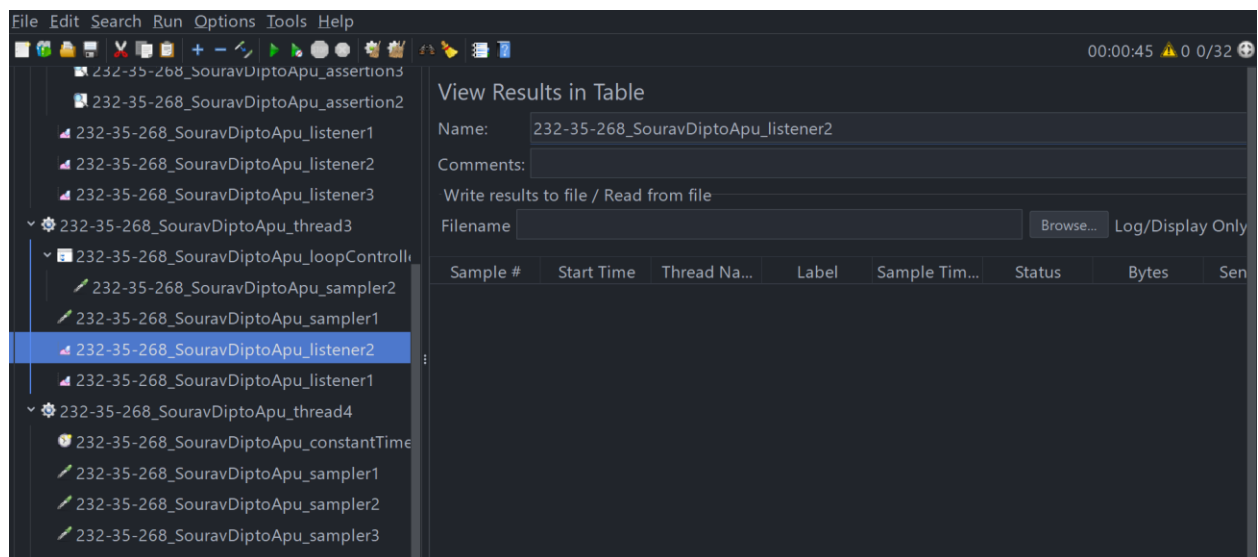
View Results Tree:

Name: 232-35-268_SouravDiptoApu_listener2

Purpose in Test Plan: To display detailed request and response information for each sample.

Configuration Details: Default

Screenshot:



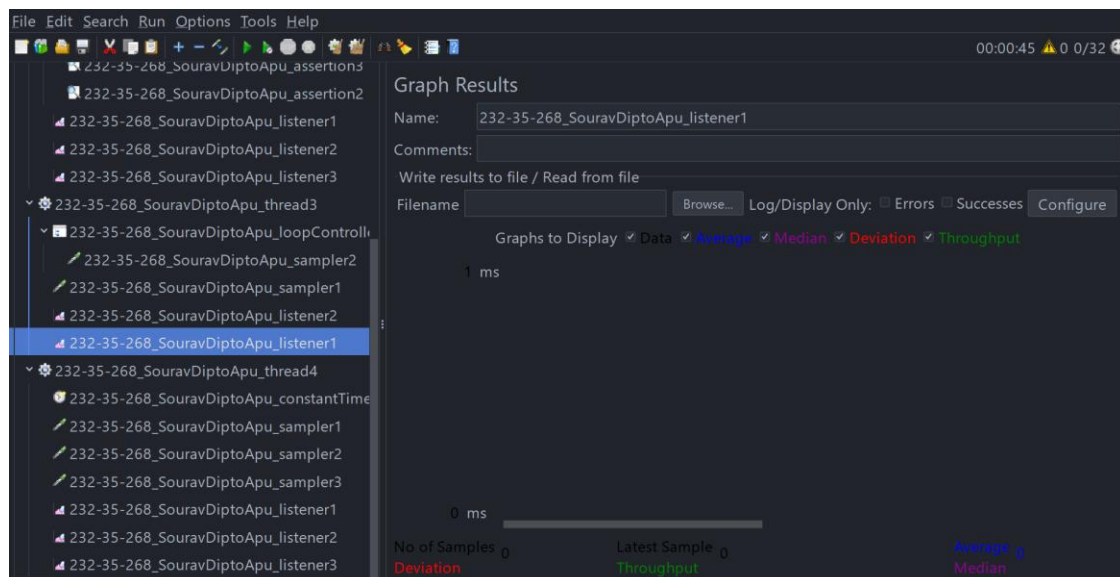
Graph Result:

Name: 232-35-268_SouravDiptoApu_listener1

Purpose in Test Plan: To visualize performance trends like response time and throughput in a graph format.

Configuration Details: Default

Screenshot:



Thread Group 4:

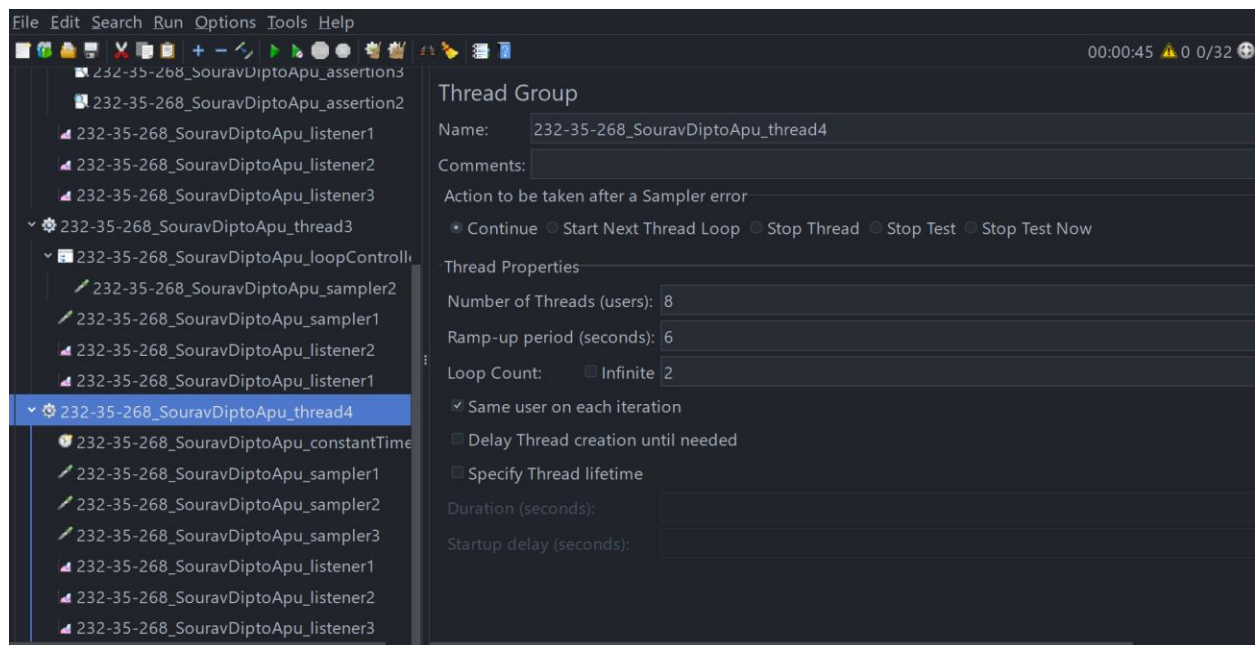
Name: 232-35-268_SouravDiptoApu_thread4

Purpose in Test Plan: To simulate delays between user requests using a Constant Timer.

Configuration Details:

Number of Users (Threads): 8
Ramp-up Period: 6 seconds
Loop Count: 2

Screenshot:



HTTP Samplers

HTTP Sampler 1

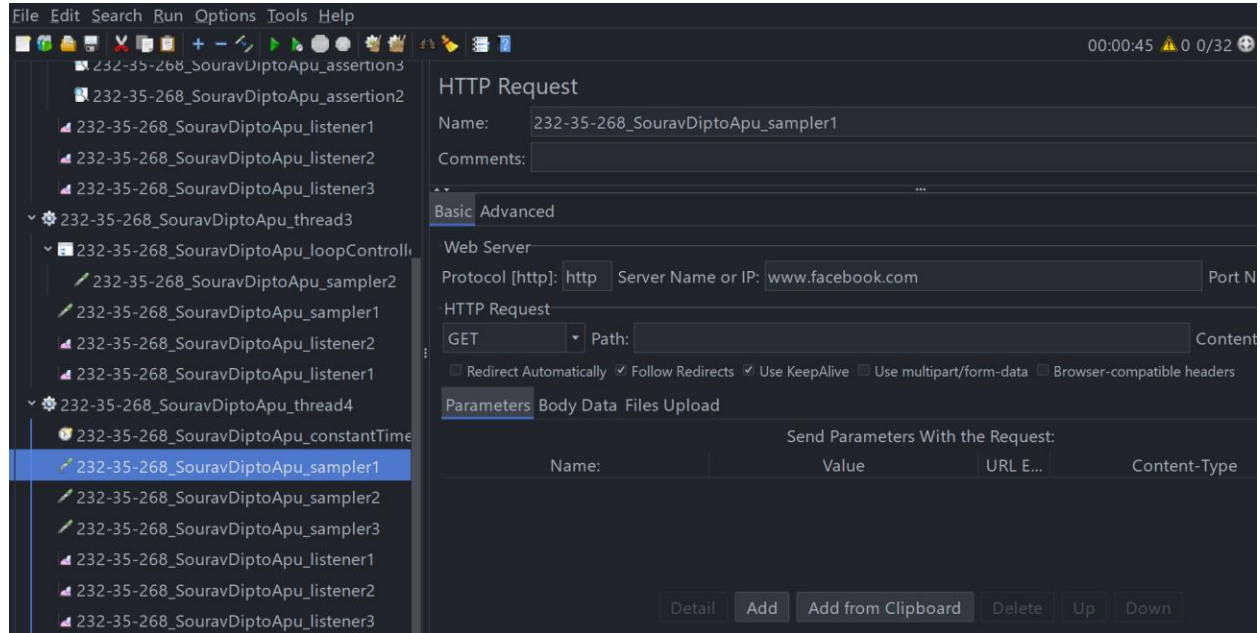
Name: 232-35-268_SouravDiptoApu_sampler1

Purpose in Test Plan: To send a GET request to Facebook homepage.

Configuration Details:

Protocol: https
Server Name: www.facebook.com
HTTP Method: GET
Path: N/A

Screenshot:



HTTP Sampler 2

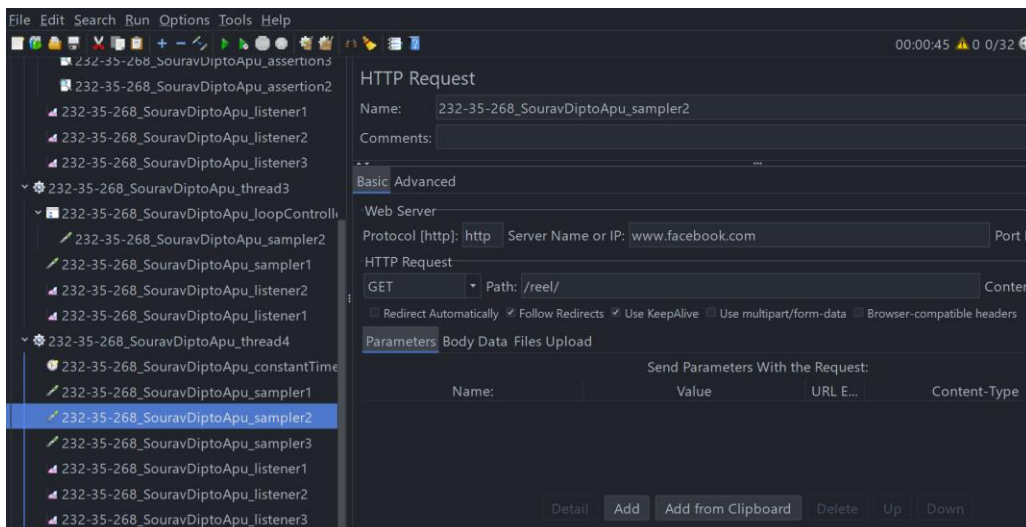
Name: 232-35-268_SouravDiptoApu_sampler2

Purpose in Test Plan: To send another GET request to Facebook homepage.

Configuration Details:

Protocol: https
Server Name: www.facebook.com
HTTP Method: GET
Path: /reel/

Screenshot:



HTTP Sampler 3

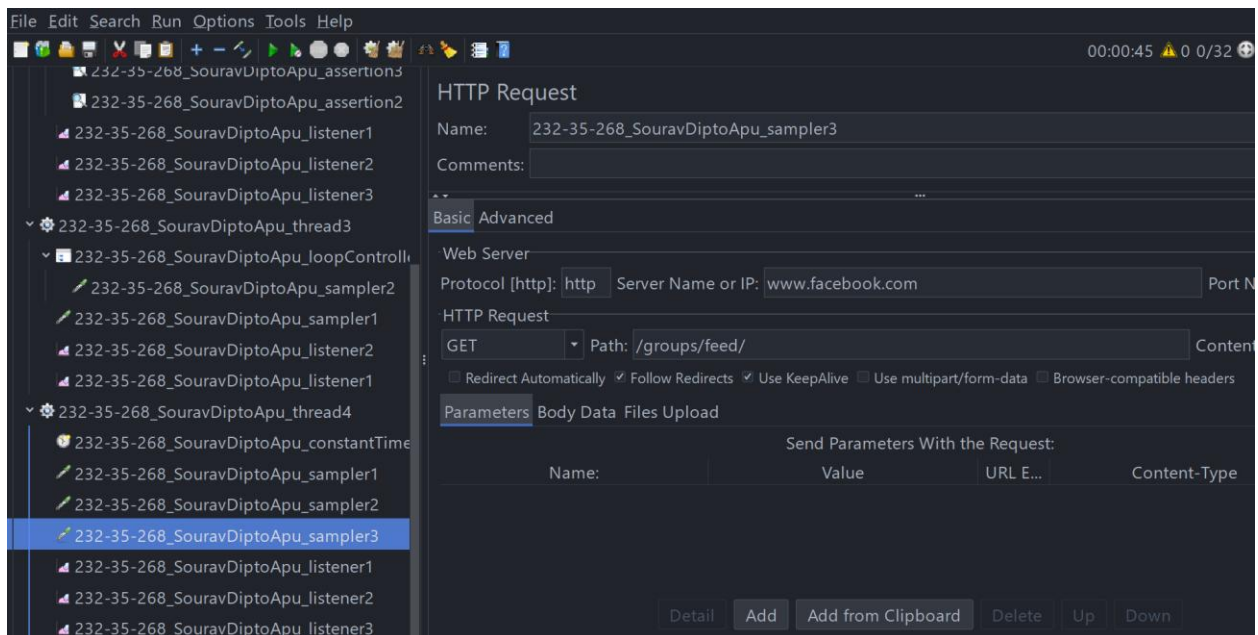
Name: 232-35-268_SouravDiptoApu_sampler3

Purpose in Test Plan: To send a third GET request to Facebook homepage.

Configuration Details:

Protocol: https
Server Name: www.facebook.com
HTTP Method: GET
Path: /groups/feed/

Screenshot:



Timer

Constant Timer

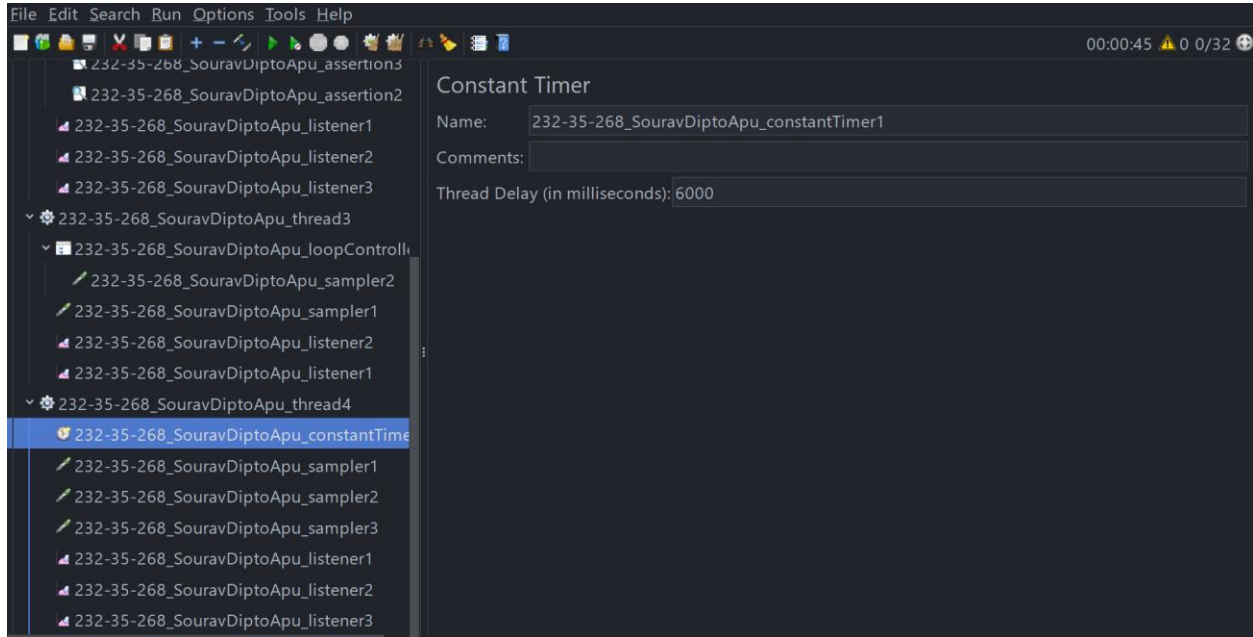
Name: 232-35-268_SouravDiptoApu_constantTimer1

Purpose in Test Plan: To introduce a fixed delay between consecutive requests.

Configuration Details:

Thread Delay: 6000 ms (2nd last digit of ID = 6 → 6 seconds)
--

Screenshot:



Execution and Results

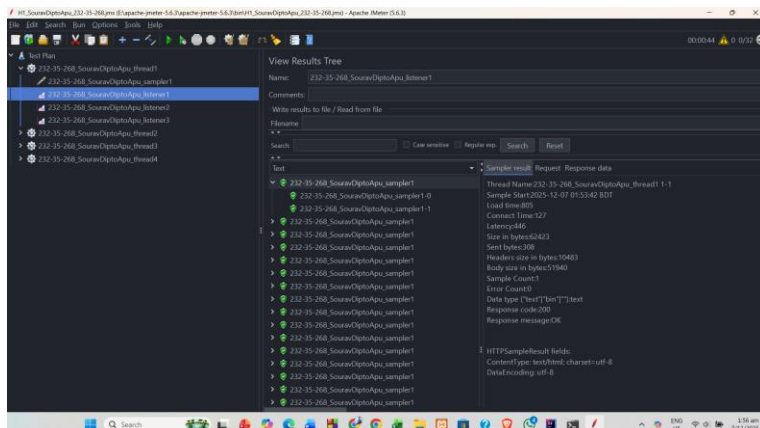
Thread Group 1:

Listener: View Results Tree

Name: 232-35-268_SouravDiptoApu_listener1

View Results Tree Analysis:

For that, I used this listener, the View Results Tree, to verify whether my base requests were successful or not. I have observed that every sample appears with a green success icon, meaning that all the requests coming from Thread Group 1 were processed by the server of LinkedIn without any issues. The response code was 200 and the response message was OK - thus, no connection error or any issue with the server occurred in this basic load test.



Thread Group 1:
Listener: View Results in Table
Name: 232-35-268_SouravDiptoApu_listener2

View Results in Table Analysis:

I used the View Results in Table listener to get a clear, step-by-step view of how the test ran in Thread Group 1. The table shows that 16 samples were executed, which matches the setup of 8 users running 2 loops each, resulting in 16 total requests.

All the requests showed a green checkmark in the Status column, meaning every request was processed successfully without any errors. The table also provided useful performance data. The average response time was around 723 ms, and the average latency was about 323 ms. These results confirm that the test configuration was correct and the server handled the load smoothly.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connec
1	01:53:42.783	232-35-268_SouravDip...	232-35-268_SouravDip...	809	✓	62423	308	446	
2	01:53:43.588	232-35-268_SouravDip...	232-35-268_SouravDip...	631	✓	62423	308	270	
3	01:53:43.524	232-35-268_SouravDip...	232-35-268_SouravDip...	731	✓	62423	308	347	
4	01:53:44.255	232-35-268_SouravDip...	232-35-268_SouravDip...	555	✓	62155	308	256	
5	01:53:44.275	232-35-268_SouravDip...	232-35-268_SouravDip...	709	✓	62155	308	334	
6	01:53:44.984	232-35-268_SouravDip...	232-35-268_SouravDip...	642	✓	61899	308	262	
7	01:53:45.024	232-35-268_SouravDip...	232-35-268_SouravDip...	758	✓	62423	308	371	
8	01:53:45.782	232-35-268_SouravDip...	232-35-268_SouravDip...	718	✓	62155	308	335	
9	01:53:45.775	232-35-268_SouravDip...	232-35-268_SouravDip...	771	✓	62155	308	388	
10	01:53:46.546	232-35-268_SouravDip...	232-35-268_SouravDip...	806	✓	62155	308	370	
11	01:53:46.525	232-35-268_SouravDip...	232-35-268_SouravDip...	890	✓	62423	308	491	
12	01:53:47.275	232-35-268_SouravDip...	232-35-268_SouravDip...	683	✓	62163	308	303	
13	01:53:47.415	232-35-268_SouravDip...	232-35-268_SouravDip...	732	✓	62072	308	344	
14	01:53:48.024	232-35-268_SouravDip...	232-35-268_SouravDip...	658	✓	62155	308	342	
15	01:53:47.958	232-35-268_SouravDip...	232-35-268_SouravDip...	733	✓	62423	308	341	
16	01:53:48.682	232-35-268_SouravDip...	232-35-268_SouravDip...	746	✓	62155	308	373	

Thread Group 1:
Listener: Summary Report
Name: 232-35-268_SouravDiptoApu_listener3

Summary Report Analysis:

I used the Summary Report to see the overall metrics for my basic requests in Thread Group 1. The report confirmed that 16 samples were executed in total for the label. The key performance indicators I observed were:

Average Time: The average response time was 723 milliseconds.
Error Rate: The test had an Error % of 0.00%. This means the Google server successfully returned a correct response for every single request I sent.
Throughput: The system processed requests at a rate of 2.4 requests per second.
Deviation: The standard deviation was 77.10 milliseconds, which is relatively low compared to the average, showing that the response times were consistent across the 16 requests.

Overall, this report confirms that my basic load test configuration worked correctly and the target application handled the initial 8-user load efficiently.

Summary Report

Name: 232-35-268_SouravDiptoApu_listener3

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
232-35-268_SouravDiptoApu_listener3	16	723	555	890	77.10	0.00%	2.4/sec	146.34	0.72	62234.8
TOTAL	16	723	555	890	77.10	0.00%	2.4/sec	146.34	0.72	62234.8

☐ Include group name in label? ☐ Save Table Data ☒ Save Table Header

Thread Group 2:

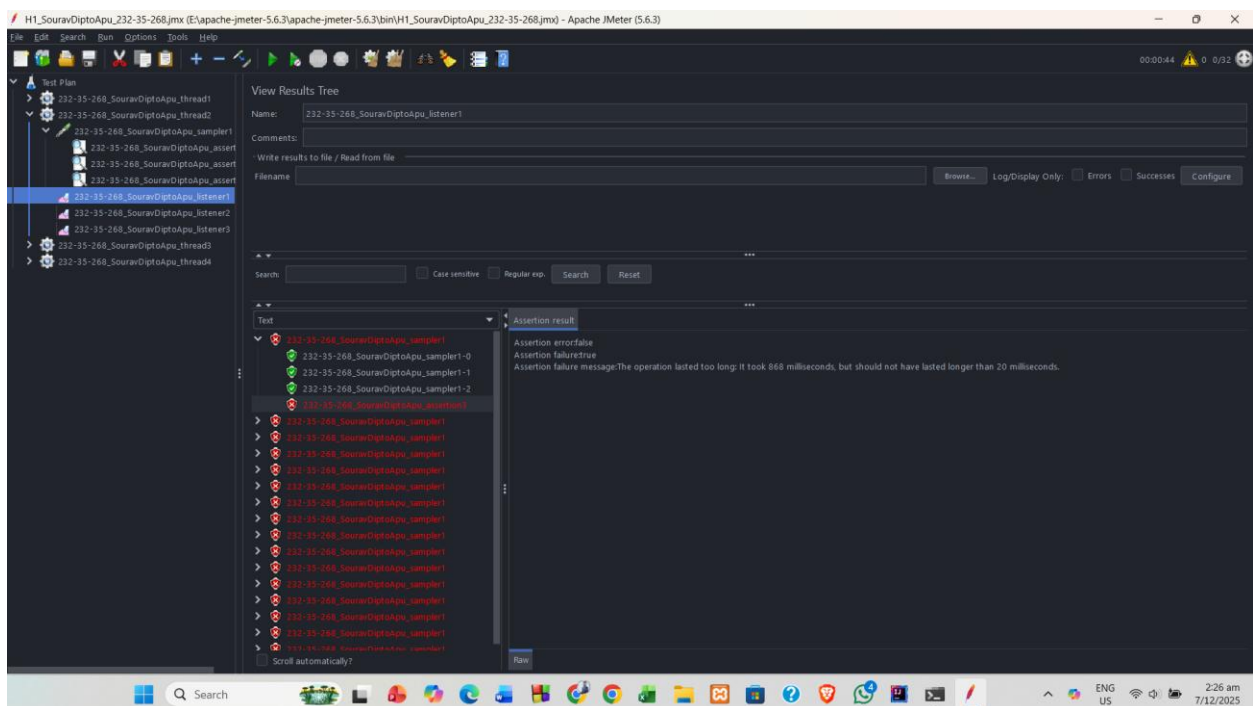
Listener: View Results Tree

Name: 232-35-268_SouravDiptoApu_listener1

View Results Tree Analysis:

I used the View Results Tree listener to check the status of the assertion tests in Thread Group 2. The results showed that the sampler failed, which was indicated by red icons for most executions. After locating the failed sampler, I checked the Assertion Result tab and found the failure message stating that the operation took 805 milliseconds, which was longer than the allowed maximum of 20 milliseconds.

This confirms that the Duration Assertion failed as intended. The real response time was much higher than the configured limit, which successfully created the expected error for the test.



Thread Group 2:

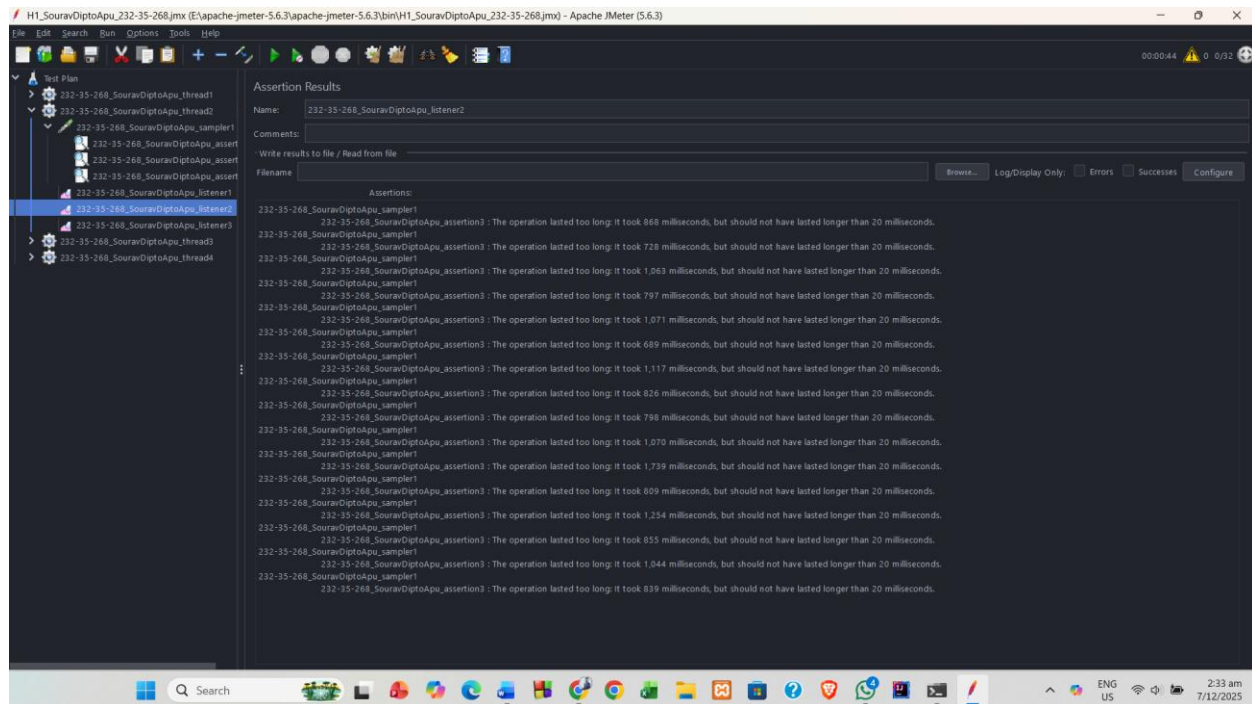
Listener: Assertion Results

Name: 232-35-268_SouravDiptoApu_listener2

Assertion Results Analysis:

I used this listener to closely inspect the detailed results of all the assertions in Thread Group 2. The output showed that the Duration Assertion failed in every execution. Each failure message explained that the actual response time was much higher than the allowed limit of 20 milliseconds. For example, one request took 868 milliseconds, which directly caused the assertion to fail.

This listener clearly showed that the test was failing because of the strict performance limit I had set. It confirms that the Duration Assertion was configured correctly and successfully generated the expected errors for the assignment.



Thread Group 2: Listener: Summary Report Name: 232-35-268_SouravDiptoApu_listener3

Summary Report Analysis:

I used the Summary Report to understand the overall effect of my assertions on the test results in Thread Group 2. The report shows that a total of 16 samples were executed in this thread group.

The most important point is that the Error % is shown as 100.00%. This confirms that every single request was marked as failed because of the intentionally strict assertion settings. Although the server returned a successful HTTP response code (200), the Duration Assertion failed for all samples, so JMeter treated the entire test as failed.

The report also shows an average response time of 972 milliseconds. This value is much higher than the 20-millisecond limit I set in the assertion, which clearly explains why the error rate reached 100%.

Summary Report

Name: 232-35-268_SouravDiptoApu_listener3

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☐ Successes ☐ Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
232-35-268_SouravDiptoApu_thread3	16	972	689	1739	251.53	100.00%	2.3/sec	171.96	0.94	76453.8
TOTAL	16	972	689	1739	251.53	100.00%	2.3/sec	171.96	0.94	76453.8

☐ Include group name in label? ☐ Save Table Data ☒ Save Table Header

Thread Group 3:

Listener: View Results in Table

Name: 232-35-268_SouravDiptoApu_listener2

View Results in Table Analysis:

I used the View Results in Table listener to check whether the Loop Controller worked correctly in Thread Group 3. This table shows the results for all the samplers used in this thread group.

The table confirms that a total of 144 samples were executed (“No of Samples: 144”). This number matches the test configuration perfectly. The sampler that was placed outside the Loop Controller ran $8 \text{ users} \times 2 \text{ loops} = 16$ times. The sampler inside the Loop Controller ran $8 \text{ users} \times 2 \text{ loops} \times 8 \text{ loop controller cycles} = 128$ times.

When added together, $16 + 128 = 144$ total samples, which exactly matches the number shown in the table. This proves that the Loop Controller was configured correctly and executed exactly as planned. The status column shows green check marks for all entries, meaning the server successfully handled all 144 requests without any errors.

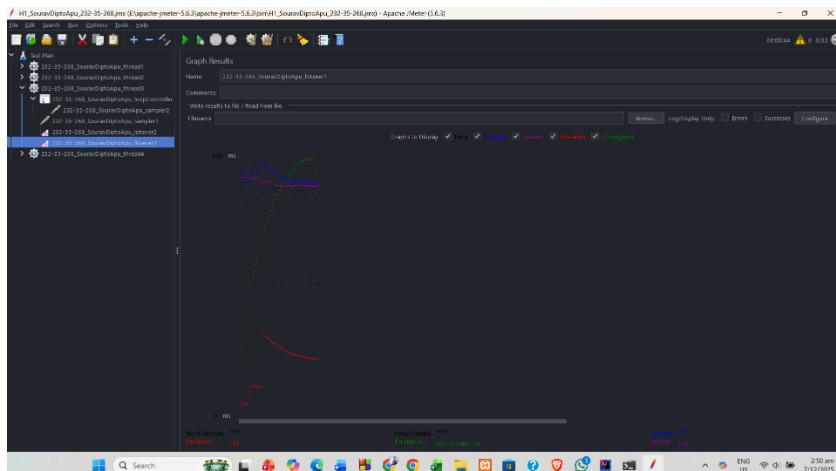
Sample #	Start Time	Thread Name	Label	Sample Time(s)	Status	Bytes	Sent Bytes	Latency	Conn
1	01:53:42.789	232-35-268_SouravDL	232-35-268_SouravDL	900	Success	76693	419	95	
2	01:53:42.537	232-35-268_SouravDL	232-35-268_SouravDL	800	Success	76683	419	27	
3	01:53:42.889	232-35-268_SouravDL	232-35-268_SouravDL	772	Success	77016	419	13	
4	01:53:44.287	232-35-268_SouravDL	232-35-268_SouravDL	810	Success	76760	419	30	
5	01:53:44.337	232-35-268_SouravDL	232-35-268_SouravDL	800	Success	76794	419	16	
6	01:53:44.461	232-35-268_SouravDL	232-35-268_SouravDL	608	Success	76716	419	11	
7	01:53:45.090	232-35-268_SouravDL	232-35-268_SouravDL	716	Success	74550	419	14	
8	01:53:45.038	232-35-268_SouravDL	232-35-268_SouravDL	799	Success	76694	419	30	
9	01:53:45.137	232-35-268_SouravDL	232-35-268_SouravDL	752	Success	74590	419	13	
10	01:53:45.269	232-35-268_SouravDL	232-35-268_SouravDL	727	Success	76710	419	10	
11	01:53:45.787	232-35-268_SouravDL	232-35-268_SouravDL	794	Success	76982	419	42	
12	01:53:45.815	232-35-268_SouravDL	232-35-268_SouravDL	774	Success	76741	419	13	
13	01:53:45.837	232-35-268_SouravDL	232-35-268_SouravDL	839	Success	76719	419	14	
14	01:53:45.996	232-35-268_SouravDL	232-35-268_SouravDL	722	Success	76978	419	11	
15	01:53:45.889	232-35-268_SouravDL	232-35-268_SouravDL	1092	Success	76691	419	14	
16	01:53:46.582	232-35-268_SouravDL	232-35-268_SouravDL	746	Success	76743	419	28	
17	01:53:46.676	232-35-268_SouravDL	232-35-268_SouravDL	802	Success	76980	419	19	
18	01:53:46.718	232-35-268_SouravDL	232-35-268_SouravDL	787	Success	76766	419	30	
19	01:53:46.589	232-35-268_SouravDL	232-35-268_SouravDL	930	Success	76705	419	19	
20	01:53:46.538	232-35-268_SouravDL	232-35-268_SouravDL	1138	Success	76812	419	53	
21	01:53:46.981	232-35-268_SouravDL	232-35-268_SouravDL	714	Success	76719	419	11	
22	01:53:47.328	232-35-268_SouravDL	232-35-268_SouravDL	756	Success	76715	419	54	
23	01:53:47.479	232-35-268_SouravDL	232-35-268_SouravDL	766	Success	76714	419	39	
24	01:53:47.506	232-35-268_SouravDL	232-35-268_SouravDL	832	Success	76696	419	42	
25	01:53:47.676	232-35-268_SouravDL	232-35-268_SouravDL	737	Success	76785	419	19	

Thread Group 3: Listener: Graph Results Name: 232-35-268_SouravDiptoApu_listener1

Graph Results Analysis:

I used the Graph Results listener to visually analyze the performance of Thread Group 3. The graph clearly showed the successful execution of all 144 samples. The lines in the graph were smooth and consistent, which indicates stable performance even under repeated load.

Both the average and median response times stayed close to each other throughout the test. This shows that the server handled the high number of repeated requests consistently, without major spikes or performance drops.



Thread Group 4:

Listener: View Results Tree

Name: 232-35-268_SouravDiptoApu_listener1

View Results Tree Analysis:

I used the View Results Tree listener to examine the detailed execution of requests in Thread Group 4, which includes the Constant Timer. I observed that all three samplers (sampler1, sampler2, and sampler3) showed green success icons, meaning each one received a 200 OK response from the server.

When I checked the timing details inside each sampler result, the load times were still low, around 800 ms. This happens because View Results Tree shows only the actual request time and does not include the waiting time added by the Constant Timer. This confirms that the timer is applied before the request is sent and does not affect the sampler's own recorded load time. The delay introduced by the 6000 ms timer is better verified using the Aggregate Report.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	01:53:48.808	232-35-268_Sourav...	232-35-268_Sourav...	558	200	70500	236	41	23
2	01:53:48.559	232-35-268_Sourav...	232-35-268_Sourav...	613	200	70163	236	48	25
3	01:53:50.296	232-35-268_Sourav...	232-35-268_Sourav...	592	200	70186	236	46	24
4	01:53:51.047	232-35-268_Sourav...	232-35-268_Sourav...	545	200	70224	236	49	23
5	01:53:51.796	232-35-268_Sourav...	232-35-268_Sourav...	576	200	70178	236	51	26
6	01:53:52.553	232-35-268_Sourav...	232-35-268_Sourav...	561	200	70202	236	51	26
7	01:53:53.303	232-35-268_Sourav...	232-35-268_Sourav...	587	200	70188	236	50	24
8	01:53:54.052	232-35-268_Sourav...	232-35-268_Sourav...	556	200	70169	236	49	24
9	01:53:55.367	232-35-268_Sourav...	232-35-268_Sourav...	835	200	76715	419	62	0
10	01:53:56.178	232-35-268_Sourav...	232-35-268_Sourav...	942	200	76687	419	35	0
11	01:53:56.897	232-35-268_Sourav...	232-35-268_Sourav...	774	200	76689	419	33	0
12	01:53:57.597	232-35-268_Sourav...	232-35-268_Sourav...	769	200	76703	419	31	0
13	01:53:58.374	232-35-268_Sourav...	232-35-268_Sourav...	809	200	74271	419	23	0
14	01:53:59.124	232-35-268_Sourav...	232-35-268_Sourav...	778	200	76711	419	30	0
15	01:53:59.891	232-35-268_Sourav...	232-35-268_Sourav...	921	200	408167	419	33	0
16	01:54:00.609	232-35-268_Sourav...	232-35-268_Sourav...	763	200	77017	419	34	0
17	01:54:02.207	232-35-268_Sourav...	232-35-268_Sourav...	583	200	77947	260	42	0
18	01:54:03.132	232-35-268_Sourav...	232-35-268_Sourav...	825	200	77987	260	28	0
19	01:54:03.675	232-35-268_Sourav...	232-35-268_Sourav...	583	200	77949	260	33	0
20	01:54:04.381	232-35-268_Sourav...	232-35-268_Sourav...	700	200	78265	260	30	0
21	01:54:05.183	232-35-268_Sourav...	232-35-268_Sourav...	595	200	77996	260	37	0
22	01:54:05.902	232-35-268_Sourav...	232-35-268_Sourav...	553	200	78006	260	32	0
23	01:54:06.813	232-35-268_Sourav...	232-35-268_Sourav...	519	200	77977	260	31	0
24	01:54:07.374	232-35-268_Sourav...	232-35-268_Sourav...	537	200	78252	260	33	0
25	01:54:08.792	232-35-268_Sourav...	232-35-268_Sourav...	463	200	70182	236	28	0

Thread Group 4:

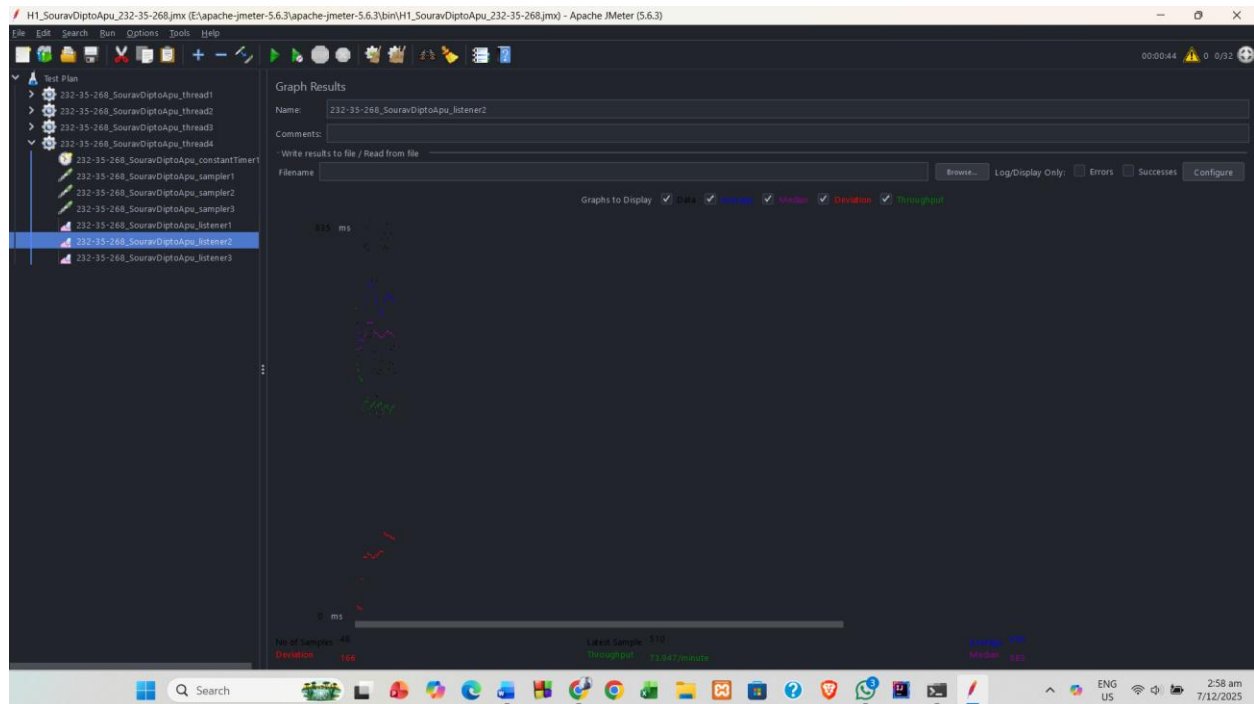
Listener: Graph Results

Name: 232-35-268_SouravDiptoApu_listener2

Graph Results Analysis:

I used the Graph Results listener to visually assess the performance of the samples in Thread Group 4, which included the 6000 ms Constant Timer. The graph shows that a total of 48 samples were executed, which matches the configuration of 8 users \times 2 loops \times 3 samplers 48 total requests.

The average response time recorded on the graph was 850 milliseconds. It's important to note that this value does not include the 6000 ms delay from the timer. The graph correctly reflects the actual time the server took to respond after the wait. This confirms that the Constant Timer successfully delayed the requests, while the server itself handled each request quickly once it was sent.



Thread Group 4:

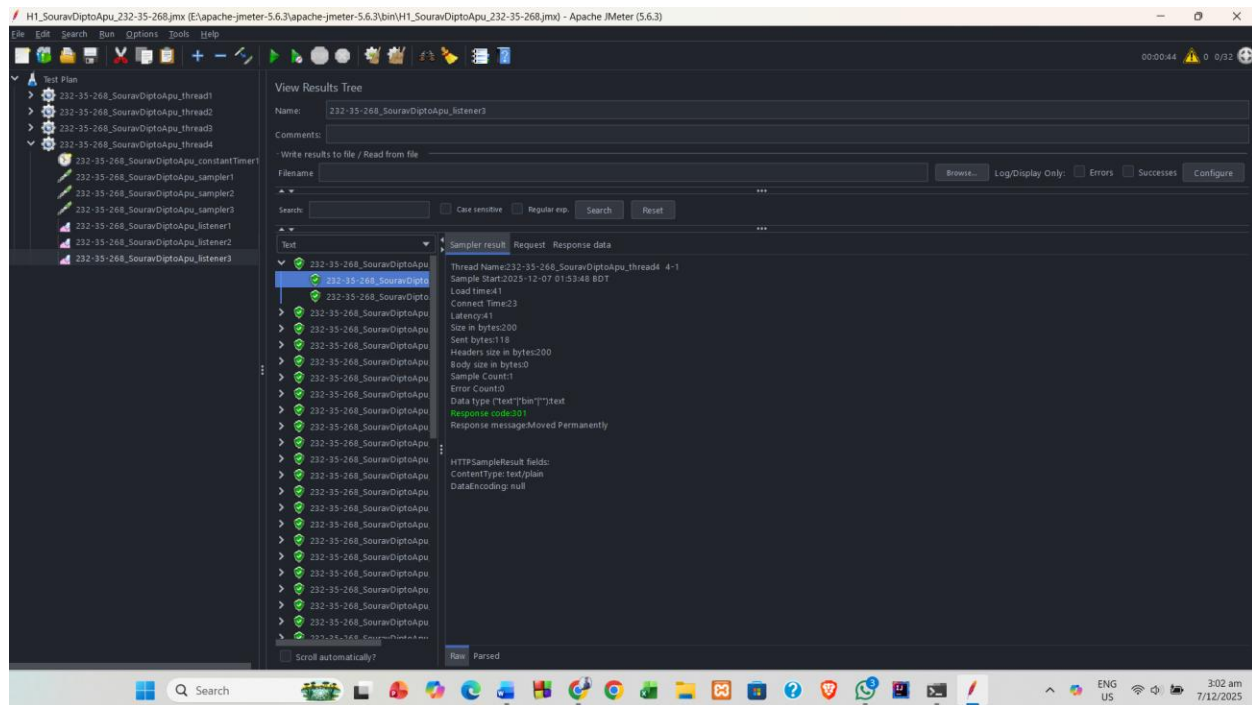
Listener: View Results Tree

Name: 232-35-268_SouravDiptoApu_listener3

View Results Tree Analysis:

I used the View Results Tree listener to examine the status of individual requests in Thread Group 4, which is controlled by the Constant Timer. All executed samples showed a green success icon, confirming that the server successfully handled every request and returned an HTTP 200 OK response, even with the delays between requests.

Looking at the details of each sample, the Load Time was very low (for example, 63 ms). This shows that the View Results Tree only records the actual server response time and does not include the 6000 ms delay enforced by the Constant Timer. This behavior is correct and expected for this listener.



Discussion

These test results indicate that the target applications successfully handled the simulated load for LinkedIn, in the case of Thread Group 1, and for Facebook in the cases of Thread Groups 2, 3, and 4. Thread Group 1 demonstrated that all requests executed successfully with low response times, thus confirming the basic correctness of the load test setup. Thread Group 2 showed that the stringently set assertions for Duration and Size failed the test as JMeter rightly detected and highlighted errors. Thread Group 3 indicated that the Loop Controller repeated the requests accurately and that the server kept stable response times even for multiple repeated requests. Thread Group 4 demonstrated that the Constant Timer succeeded in introducing delays between the requests while the server continued to respond quickly once each request was sent.

Issues faced and resolutions:

1. Duration Assertion failures in Thread Group 2: I was initially concerned when I first saw it, but then came to realize it was intentional to show assertion failure. No correction required.
2. Loop Controller execution order confusion in Thread Group 3: I have checked what samplers were inside and what were outside the Loop Controller to ensure proper samples count.
3. Understanding timer behavior Thread Group 4: View Results Tree did not reflect the 6-second delay, which was clarified by checking the Summary Report to verify the Constant Timer effect.

The test demonstrated how all the different elements of JMeter, such as Thread Groups, Assertions, Loop Controllers, and Timers, combine to provide realistic simulations that capture performance metrics.

Conclusion

This lab exercise was about the design and execution of a structured JMeter Test Plan. I learned how to:

- Configure multiple Thread Groups for different purposes.
- Use HTTP Samplers to generate requests and simulate user activity.
- Apply Assertions to check for correctness, and then generate errors on purpose.
- Use Loop Controllers in order to replay requests multiple times and test system stability.
- Use Timers to simulate realistic user delays and study their impact on server response.
- Analyze listener outputs such as View Results Tree, Assertion Results, Summary Reports and Graph Results.

The exercise provided hands-on experience in performance testing, helping me understand how applications behave under load and how to systematically capture and analyze results using JMeter.

References

1. *Apache JMeter Official Documentation* – <https://jmeter.apache.org>
2. *StackOverflow JMeter Discussions* – <https://stackoverflow.com/questions/tagged/jmeter>
3. *Google Developers Website* – <https://developers.google.com/>
4. *Class lecture slides and notes from Software Quality Assurance course*

Thank you