PROJECT TITLE:TESTING TAMILNADU ELECTRICITY BOARD WEBSITE USING JMETER

TOOL USED:JMETER

WEBSITE:TAMILNADU ELECTRICITY BOARD

> **"TESTING SHOWS THE PRESENCE, NOT THE ABSENCE OF BUG."**

IDENTIFICATION OF THE TESTING TOOL

JMETER

WHY JMETER?

JMeter is a popular choice for performance testing and load testing for several reasons, despite the availability of numerous other tools in the market:

1. **Open Source**: JMeter is open-source software, which means it's free to use and can be easily customized to fit specific testing needs. This makes it accessible to organizations with limited budgets.
2. **Versatility**: JMeter is not just limited to web applications; it can also be used to test FTP, JDBC databases, LDAP, SOAP, JMS, POP3, IMAP, and other protocols. This versatility makes it a valuable tool for testing various types of applications and systems.
3. **User-Friendly Interface**: JMeter provides a user-friendly interface that allows testers to create and execute test plans without extensive programming knowledge. Test plans can be created using a GUI, making it accessible to both technical and nontechnical users.
4. **Extensive Documentation and Community Support**: JMeter has been around for many years, which means there is extensive documentation and a large community of users who can provide support and assistance. This makes it easier for new users to get started and troubleshoot issues.

5. **Plugins and Extensions**: JMeter has a rich ecosystem of plugins and extensions that extend its functionality. These plugins can be used to integrate with other tools, add additional features, or support specific protocols.
6. **Scalability**: JMeter is highly scalable and can simulate a large number of users and requests, making it suitable for testing hightraffic websites and applications.

7. **Integration with Continuous Integration (CI) Systems**: JMeter can be easily integrated with continuous integration systems like Jenkins, allowing for automated performance testing as part of the development workflow.
   IMPORTANCE OF JMETER:

This allows for automated performance testing as part of the development process, ensuring consistent application performance through .The importance of JMeter lies in its ability to effectively conduct performance testing, load testing, and stress testing for various types of applications.

1. **Ensuring Application Performance**: Performance testing is crucial for ensuring that an application can handle the expected load and deliver satisfactory performance to users. JMeter allows testers to simulate a large number of users accessing the application simultaneously, helping identify performance bottlenecks and areas for optimization.

2. **Detecting Scalability Issues**: JMeter helps in evaluating the scalability of an application by simulating increasing loads and monitoring its performance metrics. This enables organizations to identify scalability issues early in the development lifecycle and address them before they impact real users.

3. **Validating System Stability**: Load testing with JMeter helps validate the stability of a system under normal and peak loads. By subjecting the application to stress tests, testers can determine its capacity to handle sudden spikes in traffic and ensure that it remains stable and available during peak usage periods.

4. **Optimizing Resource Utilization**: Through performance testing, JMeter helps identify resource-intensive operations within an application, such as database queries or API calls. This information allows developers to optimize resource

utilization, improve response times, and enhance overall application efficiency.

5. **Supporting Continuous Integration/Continuous Deployment (CI/CD)**: Integrating JMeter into CI/CD pipelines enables automated performance testing as part of the development process. This ensures that performance considerations are addressed early in the development lifecycle, leading to faster delivery of high-quality software.

6. **Cost-Effectiveness**: JMeter is open-source software, making it a costeffective solution for organizations with limited testing budgets. Its free availability allows teams to conduct comprehensive performance testing without the need for expensive commercial tools.

7. **Flexibility and Customization**: JMeter offers flexibility and customization options, allowing testers to create complex test scenarios tailored to the specific requirements of their applications. Its extensibility through plugins and scripting capabilities further enhances its versatility.

8. **Community Support and Documentation**: JMeter benefits from a large and active community of users who contribute to its development, provide support, and share best practices. Extensive documentation and online resources make it easier for users to learn and master the tool.

SIMILARITIES WITH OTHER TOOLS:

There are many performance testing tools like 1loadrunner,loadview,neoload,apploader,silk performer,k6,Tauru

Open Source vs. Commercial: JMeter, k6, Taurus, and Grinder are open-source, while LoadRunner, NeoLoad, Silk Performer, and AppLoader are commercial tools.

Scripting Language: JMeter uses its own scripting language or integrates with JavaScript and Groovy. Scripting languages vary across other tools (e.g., k6 uses JavaScript, NeoLoad has its own scripting language).

Focus: JMeter is a versatile tool suitable for various testing needs. AppLoader is specifically designed for mobile applications.

Community Support: Open-source tools like JMeter and k6 benefit from strong online communities. Users can share best practices, troubleshoot issues, and find plugins or extensions to enhance functionality. While commercial tools offer vendor support, JMeter's active community can be a valuable resource.

Customization: JMeter, along with tools like Taurus, allows for extensive customization through plugins and scripting. This enables tailoring the testing process to specific application requirements and integrating with other testing tools.

Cross-Platform Compatibility: Similar to JMeter, most performance testing tools run on various operating systems like Windows, macOS, and Linux. This provides flexibility in choosing the testing environment.

Learning Resources: A wealth of tutorials, documentation, and online courses are available for JMeter, making it easier to learn and use. While commercial tools often have good documentation, JMeter's vast community resources can be a significant advantage.

Integration with CI/CD Pipelines: JMeter, along with tools like Taurus and k6, can be integrated with continuous integration anhout the development lifecycle.

1. **COMPARISON OF JMETER PERFORMANCE WITH OTHER TOOLS:**

   JMeter

   Features: JMeter offers a wide range of features for load testing, including support for various protocols like HTTP, HTTPS, JDBC, FTP, and more. It provides listeners for result analysis, distributed testing, and scripting capabilities.

   Ease of Use: JMeter has a user-friendly GUI, making it relatively easy to create and execute test plans. However, configuring complex scenarios may require scripting.

   Scalability: JMeter can handle large-scale load testing through distributed testing capabilities, allowing multiple JMeter instances to be controlled from a single controller.

   Community Support: JMeter has a large and active community, providing extensive documentation, tutorials, and plugins to extend its functionality.

   Cost: JMeter is open-source and free to use, making it a cost-effective choice for performance testing.

   Apache Benchmark :

Features: Apache Benchmark is a lightweight tool primarily designed for benchmarking HTTP servers. It's focused on sending a large number of requests to measure server performance under load.

Ease of Use: Apache Benchmark is a command-line tool, which makes it simple and straightforward to use. However, it lacks the advanced features and flexibility of GUI-based tools like JMeter.

Scalability: Apache Benchmark is suitable for basic load testing but may lack the scalability and advanced features required for complex performance testing scenarios.

Community Support: Apache Benchmark is maintained as part of the Apache HTTP Server project. While it has community support, it may not be as extensive as JMeter.

Cost: Apache Benchmark is open-source and free to use, like JMeter.

Gatling:

Features: Gatling is a modern performance testing tool designed for high concurrency and real-time analytics. It uses a scenario-based approach and provides support for HTTP, WebSocket, and JMS protocols.

Ease of Use: Gatling uses a DSL (Domain Specific Language) for defining test scenarios, which can be more complex than GUI-based tools like JMeter. However, it offers excellent flexibility and scalability.

Scalability: Gatling is highly scalable and can simulate thousands of virtual users on a single machine. It also supports distributed testing for even greater scalability.

Community Support: Gatling has a growing community and provides comprehensive documentation and tutorials. However, it may not be as extensive as JMeter's community.

Cost: Gatling is open-source and free to use, but it also offers a commercial version with additional features and support.

WHY JMETER BECAME POPULAR:

Community Support: JMeter benefits from a large and active community of users who contribute to its development, share knowledge, and provide support through forums, mailing lists, and other channels. This community support ecosystem ensures that users have access to assistance and guidance when encountering challenges or seeking advice.

Compatibility with Various Environments: JMeter's compatibility extends to various environments, including different web servers, application servers, databases, and operating systems. This broad compatibility allows testers to assess the performance of applications regardless of their underlying technology stack or deployment environment.

Integration with Monitoring Tools: JMeter can integrate seamlessly with various monitoring tools and platforms, enabling testers to gather additional performance

metrics and insights during testing. Integration with tools like Grafana, Prometheus, and New Relic enhances visibility into system behavior and performance trends.

Flexibility in Test Execution: JMeter offers flexibility in how tests are executed, allowing testers to customize parameters such as ramp-up time, thread count, and iteration count based on testing objectives and constraints. This flexibility enables testers to simulate realistic user scenarios and workload patterns accurately.

Support for Parameterization and Data-Driven Testing: JMeter supports parameterization and data-driven testing, allowing testers to dynamically change input values and test scenarios based on external data sources. This capability facilitates the creation of robust and reusable test scripts that cover a wide range of test scenarios.

Cross-Functional Testing Capabilities: In addition to performance testing, JMeter can be used for various other types of testing, including functional testing, regression testing, and API testing. This versatility allows organizations to leverage JMeter as a comprehensive testing solution for multiple testing needs.

Extensive Protocol Support: JMeter's support for a wide range of protocols enables testers to assess the performance of diverse systems, including web applications, APIs, databases, messaging systems more. This broad protocol support makes JMeter suitable for testing complex, multi-tiered architectures

Educational Resources and Training: JMeter offers a wealth of educational resources, including tutorials, documentation, and training courses, to help users learn how to effectively use the tool. These resources cater to users of varying skill levels, from beginners to advanced users, facilitating skill development and proficiency in JMeter.

Active Development and Updates: JMeter undergoes continuous development and improvement, with regular updates and releases that introduce new features, enhancements, and bug fixes. This commitment to ongoing development ensures that JMeter remains relevant and competitive in the evolving landscape of software testing tools.

Global Adoption and Recognition: JMeter's widespread adoption and recognition across industries and geographies attest to its effectiveness and reliability as a performance testing tool. Its reputation as a trusted and proven solution further contributes to its popularity among testers and organizations worldwide

WHY JMETER IS REQUIRED IN INDUSTRY:

Good online support: You can find a lot of online study material or video tutorials that can help you to learn this tool.

Quick updates: Every year, Apache software foundation release 2 or 3 versions of JMeter to keep this tool updated as per the market need.

Low learning curve: It is not necessary to learn any programming language for developing JMeter scripts, but if you are working on any complex scenario that cannot be developed using existing elements of JMeter then you might need to learn Java, Bean shell or Groovy languages.

Extensible: You can create your own plugins or add plugins created by other developers in the market. For example, you can add plugin 'MQTT Plugin' created by third-party to develop scripts for the application that supports MQTT protocol.

Global Recognition and Adoption: JMeter is globally recognized and widely adopted across industries, attesting to its effectiveness and reliability as a performance testing tool. Its reputation as a trusted and proven solution further contributes to its preference in the industry.

Comprehensive Testing Capabilities: JMeter goes beyond performance testing and supports various other types of testing, including functional testing, regression testing, and API testing. This comprehensive testing capability enables organizations to consolidate their testing efforts within a single ddddcccdctool.

FEATURES OF WEBSITE:

Tamil Nadu Electricity Board (TNEB) provides electricity services to the state of Tamil Nadu, India. While specific features may vary based on the services offered and the evolving needs of consumers, here are some common features associated with TNEB:

Electricity Distribution: TNEB oversees the distribution of electricity across Tamil Nadu, ensuring that power is supplied to residential, commercial, industrial, and agricultural consumers.

Tariff Structure: TNEB establishes tariff structures for electricity consumption, which may vary based on consumer categories, consumption slabs, and time-ofday usage. The tariff structure is periodically revised and approved by regulatory authorities.

Metering Services: TNEB installs, maintains, and reads electricity meters to measure consumption accurately for billing purposes. This includes traditional electromechanical meters as well as modern smart meters for advanced metering infrastructure (AMI).

Billing and Payment: TNEB generates electricity bills based on meter readings or estimated consumption and offers various payment options for consumers, including online payments, mobile wallets, bank payments, and collection centers.

Consumer Services: TNEB provides consumer services such as new connections, connection transfers, meter replacements, name changes, and disconnection/reconnection requests. Consumers can also report complaints or service issues to TNEB for resolution.

Energy Efficiency Initiatives: TNEB promotes energy conservation and efficiency through awareness campaigns, incentives, and programs aimed at reducing energy consumption, improving load management, and implementing energysaving technologies.

Renewable Energy Integration: TNEB facilitates the integration of renewable energy sources such as solar, wind, and biomass into the electricity grid. This includes procurement of renewable power through power purchase agreements (PPAs) and implementation of renewable energy projects.

Power Quality and Reliability: TNEB strives to maintain high standards of power quality and reliability by minimizing voltage fluctuations, reducing outage durations, and improving system resilience through investments in infrastructure and maintenance.

Online Services: TNEB offers online services such as bill payment, consumption tracking, service requests, and grievance redressal through its official website and

mobile applications. These digital platforms enhance convenience and accessibility for consumers.

Regulatory Compliance: TNEB adheres to regulatory guidelines and standards set by state electricity regulatory commissions (SERCs) and national regulatory authorities. This includes compliance with electricity laws, regulations, safety norms, and environmental policies.

Public Outreach and Education: TNEB engages in public outreach and education initiatives to promote awareness of electricity usage, safety practices, and conservation measures among consumers, stakeholders, and the general public.

Emergency Response: TNEB maintains emergency response teams and protocols to address power outages, equipment failures, natural disasters, and other contingencies promptly. This includes restoration efforts, disaster management, and coordination with relevant authorities.

Why jmeter became popular than any other tool in industry

- **Commercial Tools (e.g. LoadRunner):** Often more powerful and feature-rich, but come with licensing costs and may have steeper learning curves.
- **Other Open-Source Tools:** May offer specific functionalities or cater to niche needs, but JMeter's overall versatility and community support make it a strong contender.

MANUAL TEST CASES:

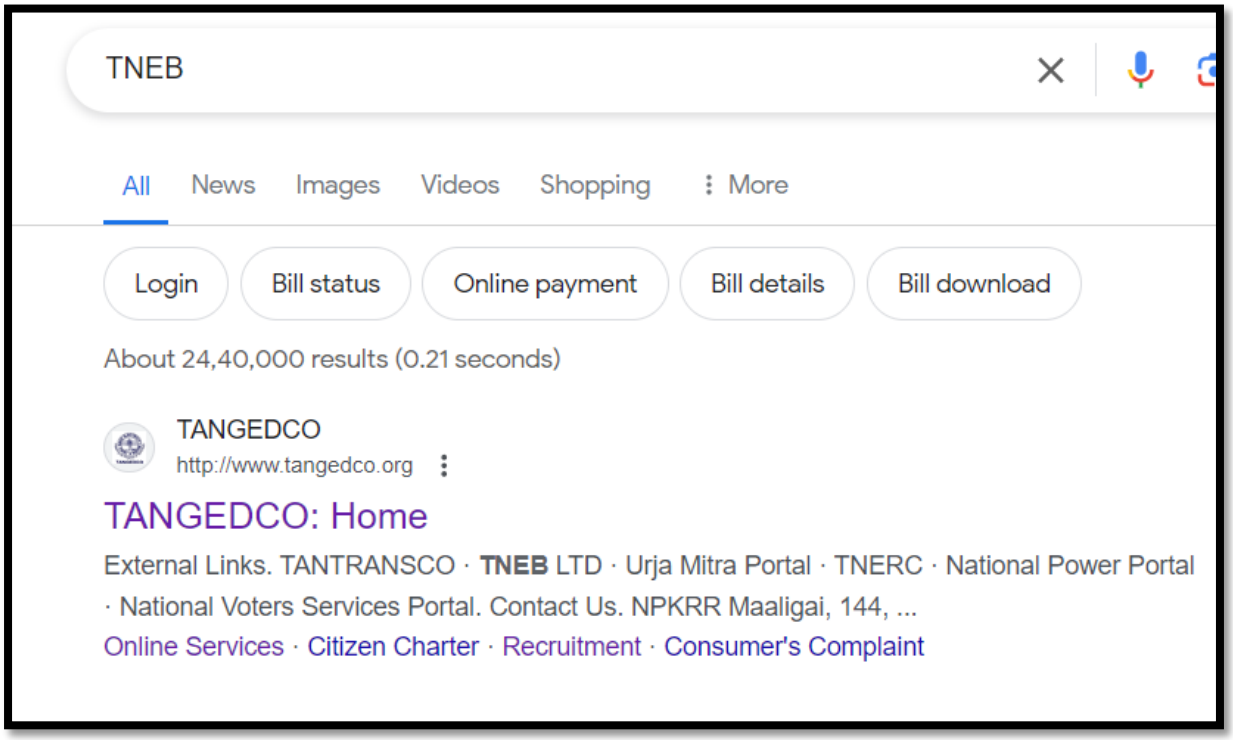
TEST CASE ID	TEST CASE DESCRIPTION	TEST STEPS	EXPECTED RESULT	ACTUAL RESULT	PASS / FAIL
TC001	User Authentication	Enter valid credentials and log in.	User is successfully logged in.	Logged in successful.	PASS
TC002	User authentication	Enter invalid credentials	System displays appropriate	Logged in successful.	FAIL

		and attempt to log in.	error message.		
TC003	Customer Information.	Update customer contact information.	Changes are reflected in the system.	Changes done.	PASS
TC004	Customer Information.	Attempt to update customer information with invalid data.	System displays an error message.	Changes done.	FAIL
TC005	Meter Reading	Enter a valid meter reading.	Reading is accurately recorded in the system.	Meter Recorded correctly.	PASS
TC006	Meter Reading	Enter a negative meter reading.	System displays an error message.	Meter Recorded correctly	FAIL
TC007	Meter Reading	Enter a reading with non-numeric characters.	System rejects the input and displays an error message.	Meter Recorded correctly	FAIL

TC008	Bill Generation	Verify bill generation for multiple tariff rates	Bills are generated accurately for different tariff rates.	Bills are generated.	PASS
TC009	Bill Generation	Verify bill generation for zero usage.	Bill reflects zero usage and charges.	Bills are Generated for zero usage.	PASS
TC010	Payment Processing	Attempt to make a	System displays an	System Displays	PASS
		payment with an invalid payment method.	error message.	Error Message	
TC011	Payment Processing	Verify the payment history is updated after a successful payment.	Payment transaction is recorded accurately	Transaction Received.	PASS
TC012	Notifications	Opt-out of receiving overdue payment notifications.	Notifications are not received after opting out.	Payments Received.	FAIL

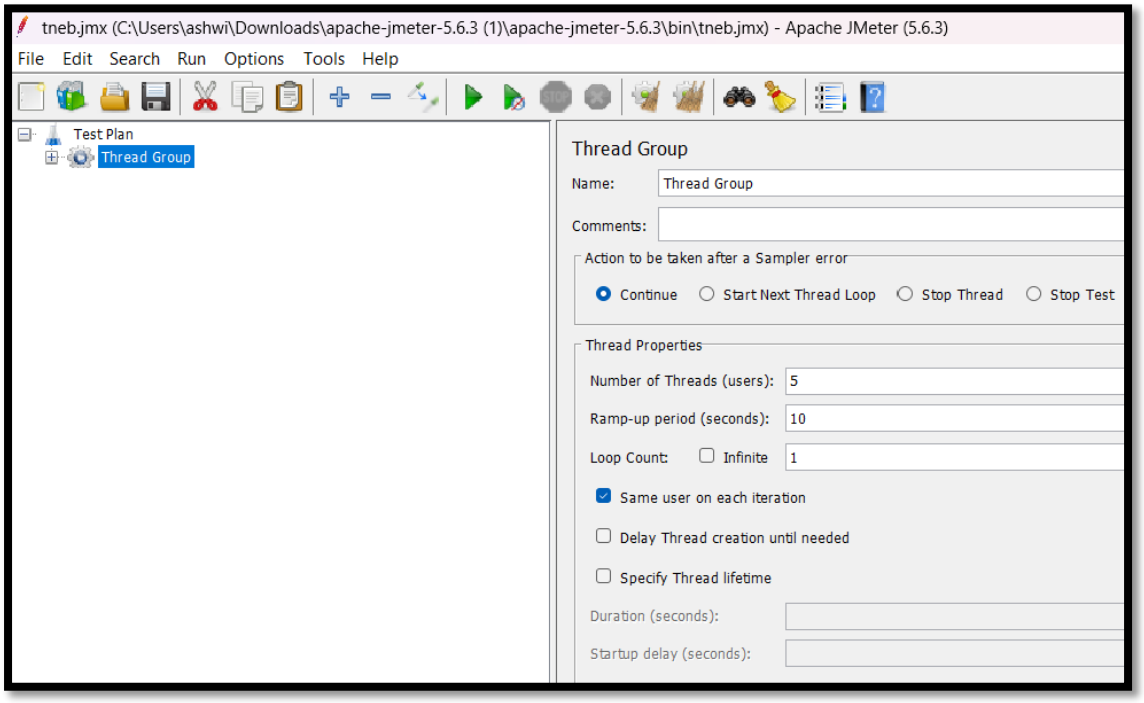
TC013	Notifications	Verify notifications are sent for system maintenance affecting billing.	Customers receive appropriate notifications	Notifications Sent to Customer.	PASS
TC014	System Security	Attempt to access another customer's billing information.	System denies access and displays an error message.	Accepts to access another customer billing information	FAIL
TC015	System Performance	Simulate high traffic during bill generation.	System generates bills without performance issues.	Performs with some issues.	FAIL
TC016	System Performance	Simulate high traffic during bill generation	System generates bills without performance issues.	Performs without some issues.	PASS

WEBSITE:



1.CREATING A NEW TEST PLAN AND ADDING THREAD GROUP TO IT

THREAD GROUP:POOL OF USERS



2.CREATING A HTTP REQUEST

SERVER NAME:WWW.TANGEDCO.ORG

PATH:THE PAGE THAT HAS TO BE TESTED

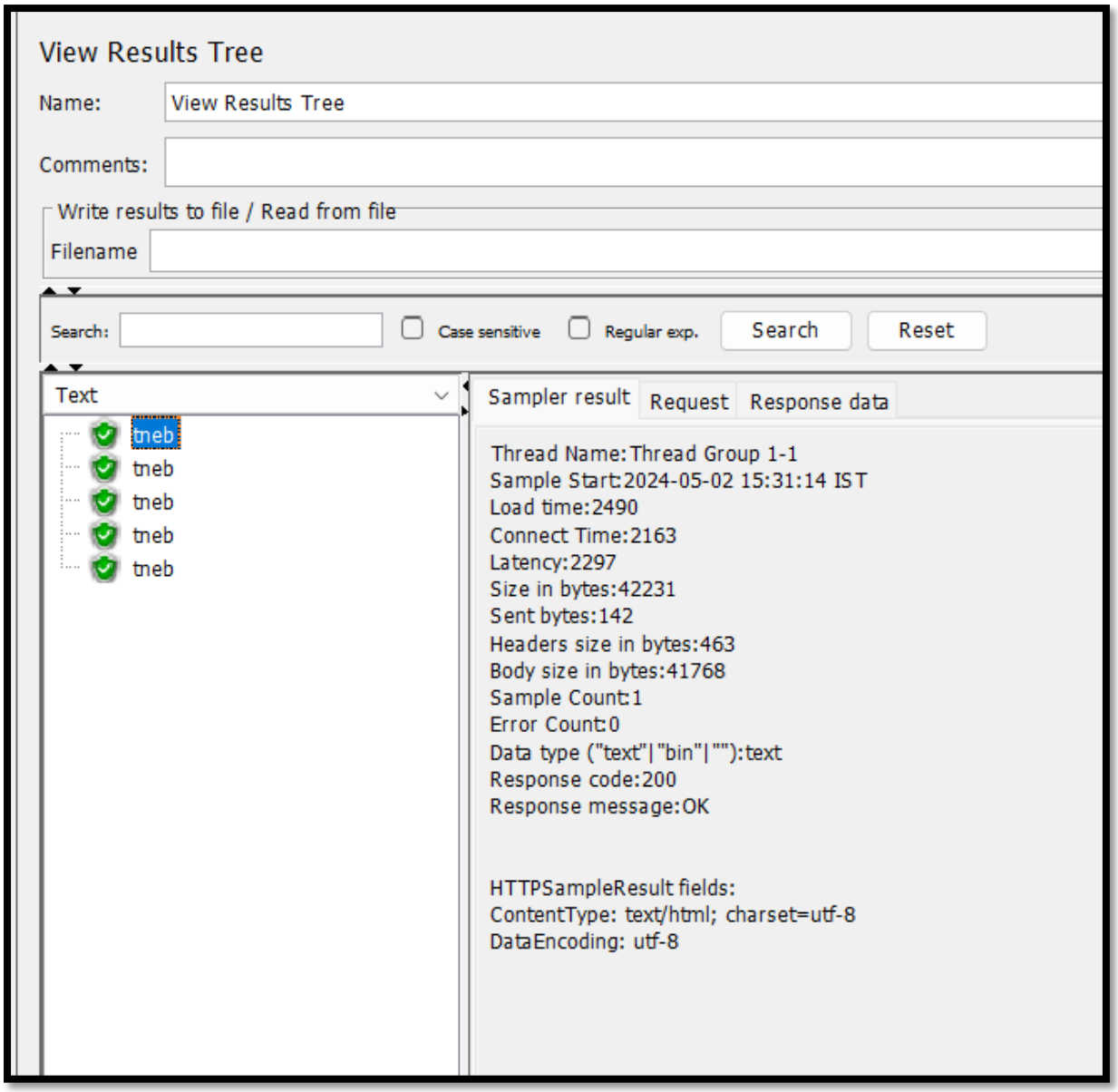
3.ADDING LISTENERS

LISTENERS:VISUAL RESULT OF THE PERFORMANCE OF THE WEBSITE

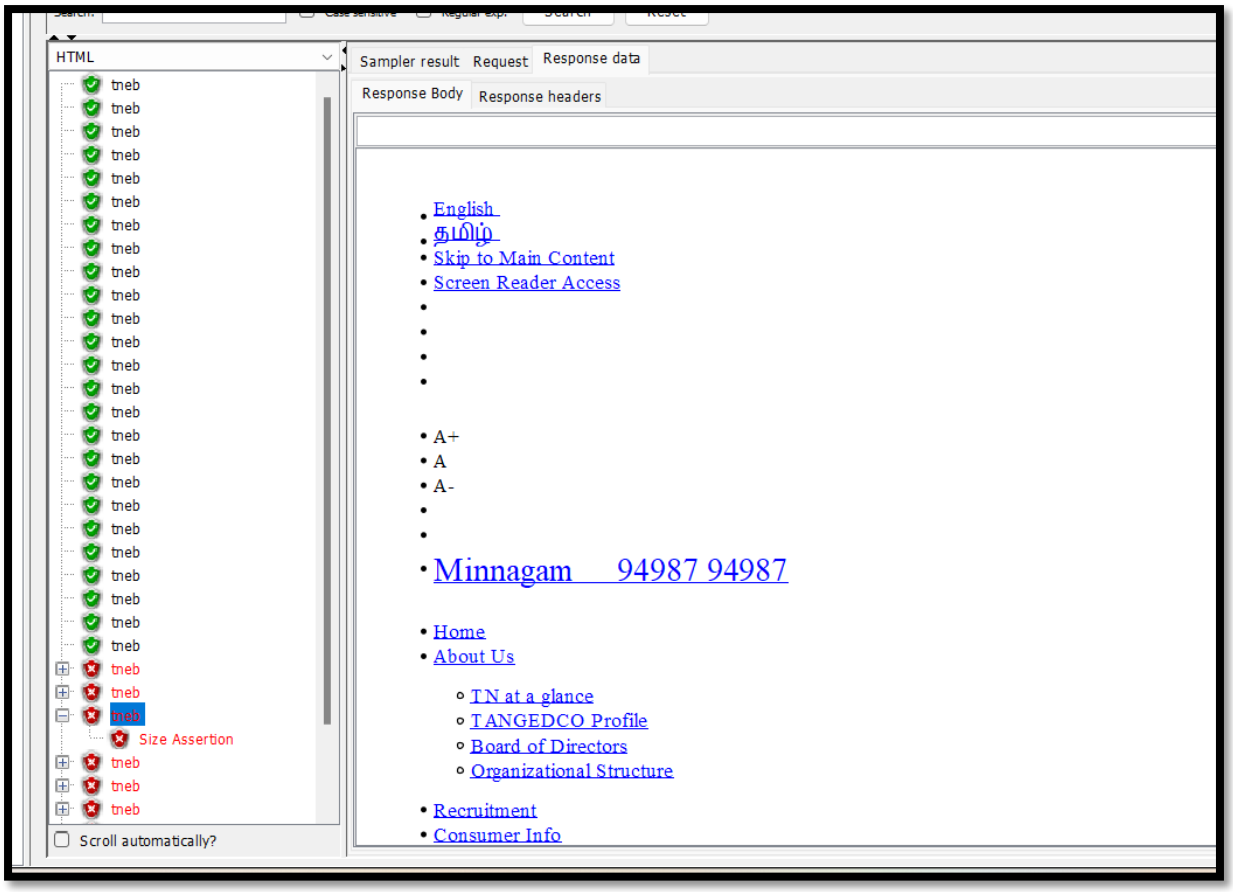
LISTENER:VIEW RESULTS TREE

CONTENTS:

- SAMPLER RESULT
- REQUEST
- RESPONSE DATA
- RESPONSE BODY
- RESPONSE HEADER

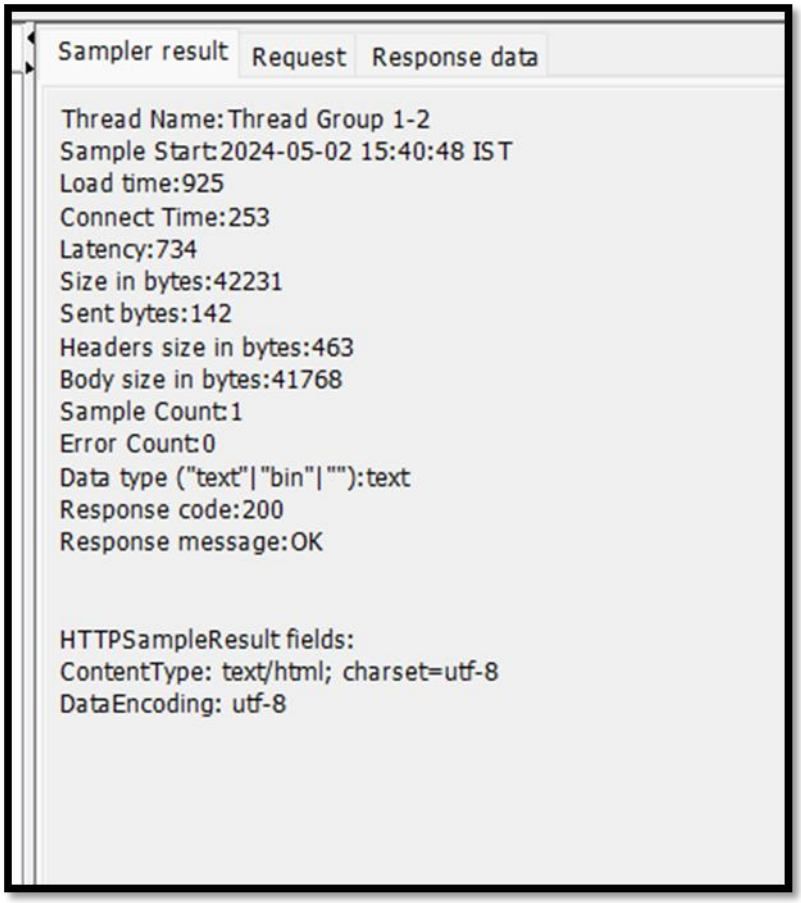


THE HTML FORMAT SHOWS THE CORRESPONDING PAGE THAT IS MENTIONED IN THE PATH



SAMPLER RESULT:

THE SAMPLER RESULTS SHOWS THE NO OF ITERATIONS,THE ORDER OF THREADS,RESPONSE CODE(200-SUCCESS),ERROR COUNT,LATENCY ETC.,



REQUEST BODY:

DISPLAYS THE URL OF THE PAGE OF THE WEBSITE THAT IS SELECTED.BELOW IS THE PAGE OF RECRUITMENT OF TNEB WEBSITE

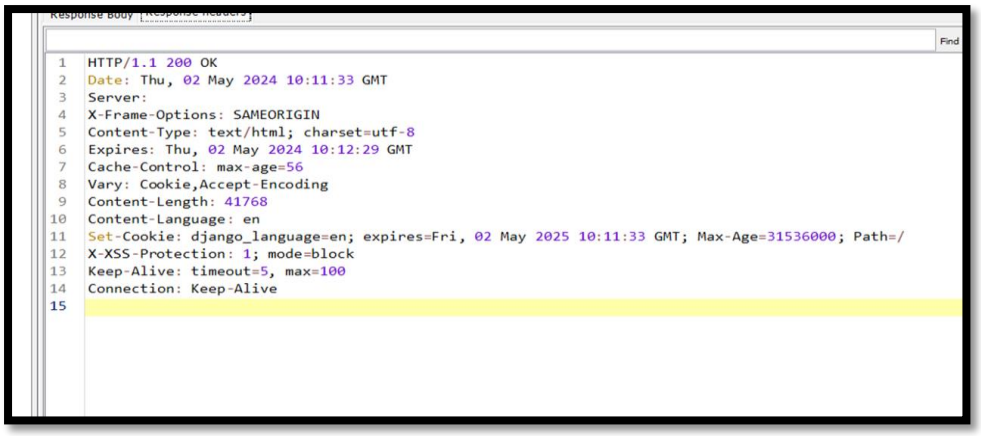
RESPONSE BODY:

SHOWS THE SCRIPTIN THE FORM OF HTML FORMAT

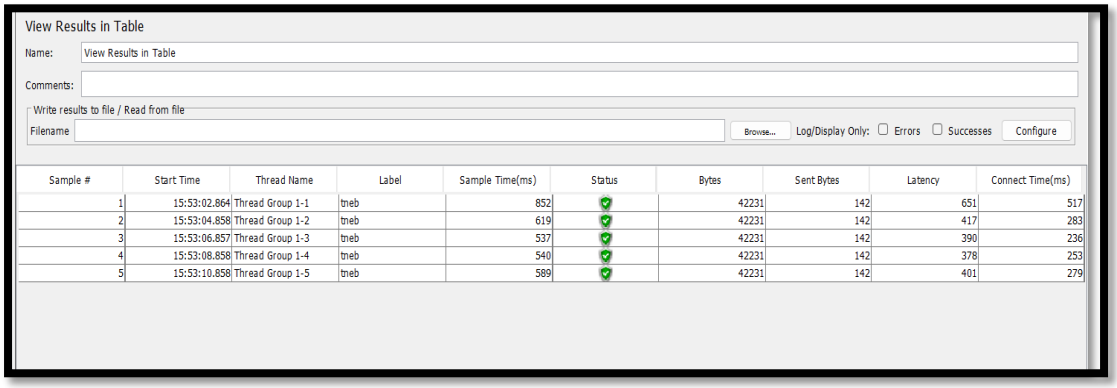
```
<!DOCTYPE html>
<html lang="en">

<head>

 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
 <meta name="description" content="">
 <meta name="author" content="">
 <meta http-equiv="X-XSS-Protection" content="0">

 <!-- Favicons -->
 <link href="/static/tangedco/assets/img/tangedco_icon.png" rel="icon">


 <!-- Vendor CSS Files -->
 <link href="/static/tangedco/assets/vendor/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/icofont/icofont.min.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet">

 <link href="/static/tangedco/assets/vendor/vendor/ionicons/css/ionicons.min.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/animate.css/animate.min.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/venobox/venobox.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/owl.carousel/assets/owl.carousel.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/vendor/aos/aos.css" rel="stylesheet">

 <!-- Template Main CSS File -->
 <link href="/static/tangedco/assets/css/jquery.bsPhotoGallery.css" rel="stylesheet">
 <link href="/static/tangedco/assets/vendor/css/style2.css" rel="stylesheet">
```

RESPONSE HEADER:

RESPONSE HEADER SHOWS ADDITIONAL INFORMATION ABOUT THE RESPONSE BODY



LISTENER:VIEW RESULTS IN TABLE



CONTENTS:

- START TIME
- THREAD NAME
- LABEL
- SAMPLE TIME
- BYTES
- SENT BYTES
- LATENCY
- CONNECT TIME

START TIME

THE TIME TAKEN BY EACH THREAD TO START

Start Time
15:53:02.864
15:53:04.858
15:53:06.857
15:53:08.858
15:53:10.858

THREAD NAME:

SPECIFIES THE ITERATION AND NO OF THREADS

Thread Name
Thread Group 1-1
Thread Group 1-2
Thread Group 1-3
Thread Group 1-4
Thread Group 1-5

LABEL:

REFERS TO THE TITLE WE HAD SPECIFIED IN THE HTTP REQUEST

Label
tneb
tneb
tneb
tneb
tneb

SAMPLE TIME:

DIFFERENT AMOUNT OF TIME TAKEN BY DIFFERENT SAMPLES

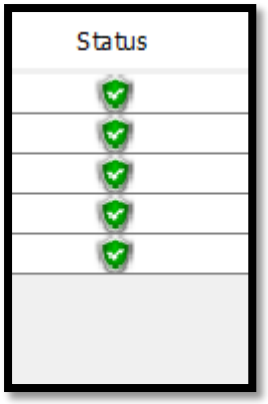
Sample Time(ms)
852
619
537
540
589

STATUS:

REFERS TO THE SUCCESS OR FAIURE OF THE CONNECTION

GREEN-REPRESENTS SUCCESS

RED-WARNING



BYTES:

IT REFERS TO THE NO OF BYTES IN THE SAMPLE

Bytes
42231
42231
42231
42231
42231

SENT BYTES:

REFERS TO NO OF BYTES SENT FOR THE SAMPLE

Sent Bytes
142
142
142
142
142

LATENCY:

REFERS TO TIME TO FIRST BYTE

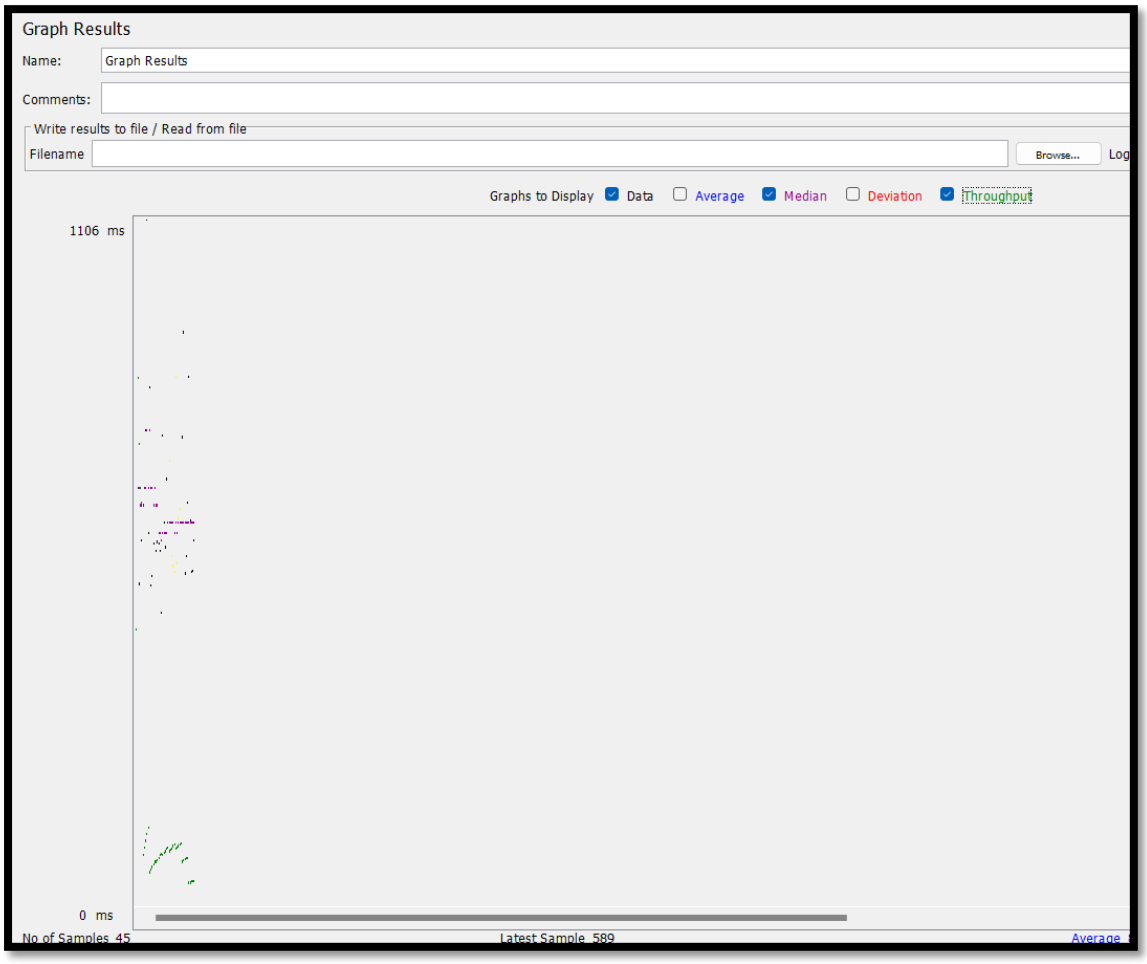
Latency
651
417
390
378
401

CONNECT TIME:

DIFFERENT TIME TAKEN TO CONNECT DIIFFERENT SAMPLES

Connect Time(ms)
517
283
236
253
279

LISTENER:GRAPH RESULTS



CONTENTS:

- AVERAGE-AVERAGE TIME TAKEN BY ALL THE SAMPLES
- MEDIAN-HALF OF THE TIME TAKEN BY THE SAMPLES
- DEVIATION-THE DEVIATION TIME OF ALL THE SAMPLES

LISTENER:AGGREGATE GRAPH

CONTENTS

- **AVERAGE:**
  AVERAGE AMOUNT OF TIME TAKEN BY THE SAMPLES

Label	# Samples	Average
tneb	45	833
TOTAL	45	833

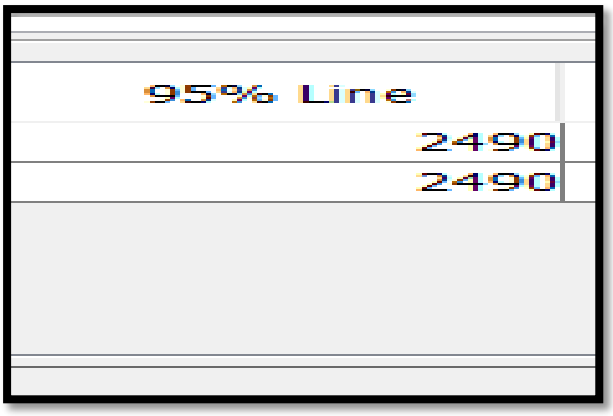
- **MEDIAN**
  TIME TAKEN BY HALF OF THE SAMPLES

Median
618
618

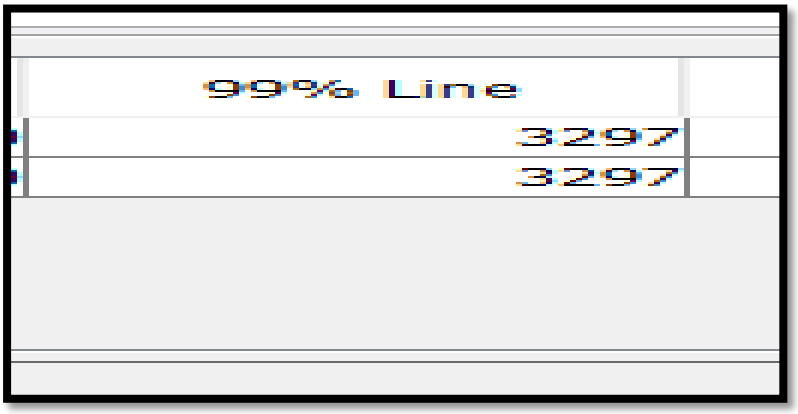
- **90%LINE**
  TIME TAKEN BY 90 %OF THE SAMPES

90% Line
1106
1106

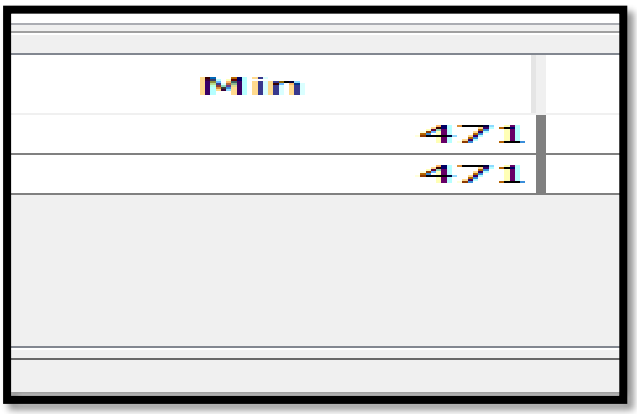
- **95%LINE**
  TIME TAKEN BY 95% OF THE SAMPLES



- **99%LINE**
  TIME TAKEN BY 99% OF THE SAMPLES.THE MAXIMUM TIME IS ALSO SAME AS THIS

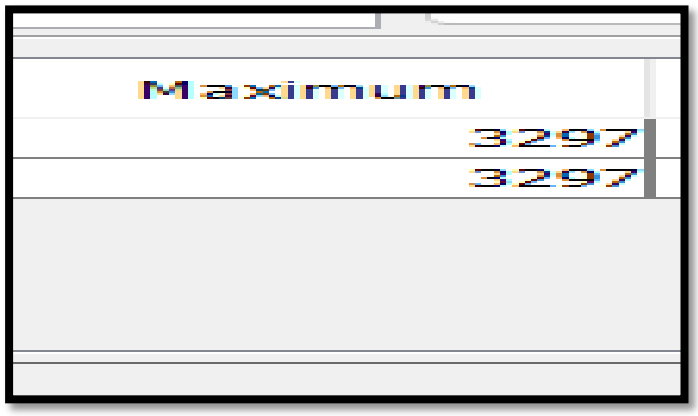


- **MINIMUM TIME**
  MINIMUM TIME TAKEN BY ALL THE SAMPLES



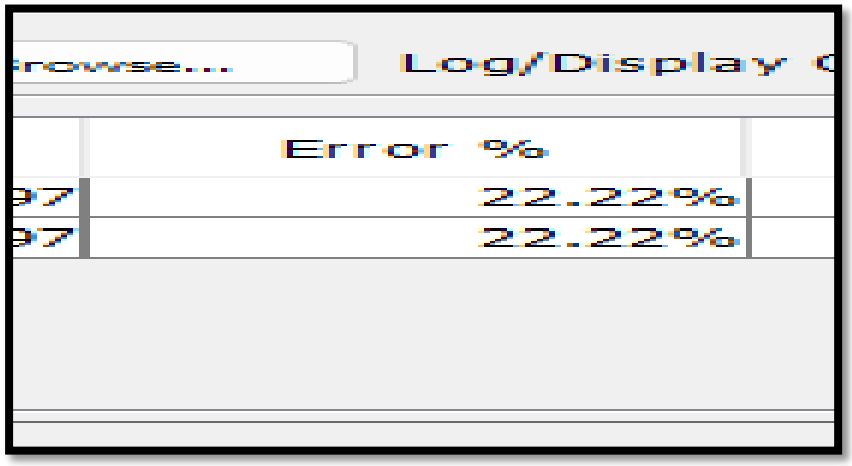
- **MAXIMUM TIME**

MAXIMUM TIME TAKEN BY ALL THE SAMPLES



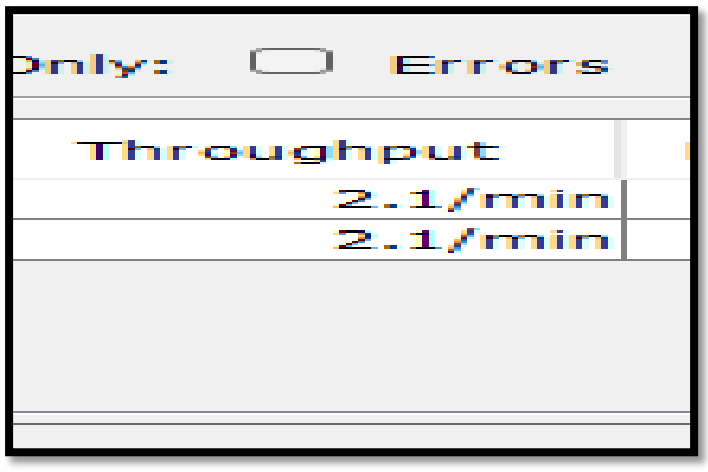
- **ERROR PERCENTAGE**

  PERCENTAGE OF ERROR FOUND



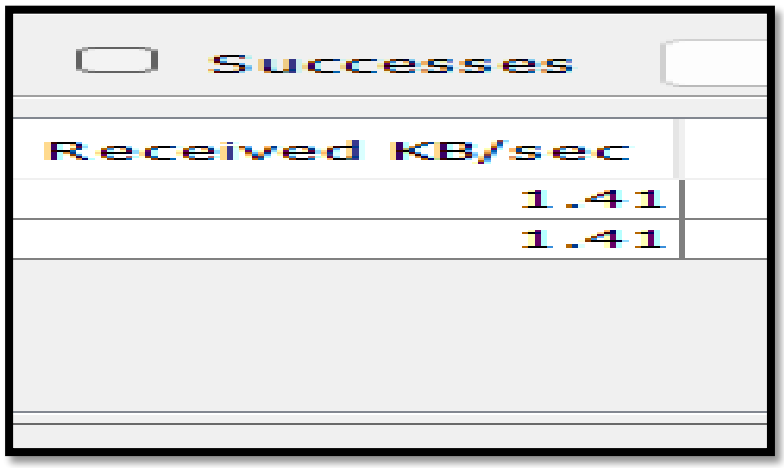
- **THROUGHPUT**

  NO OF REQUESTS PER UNIT TIME



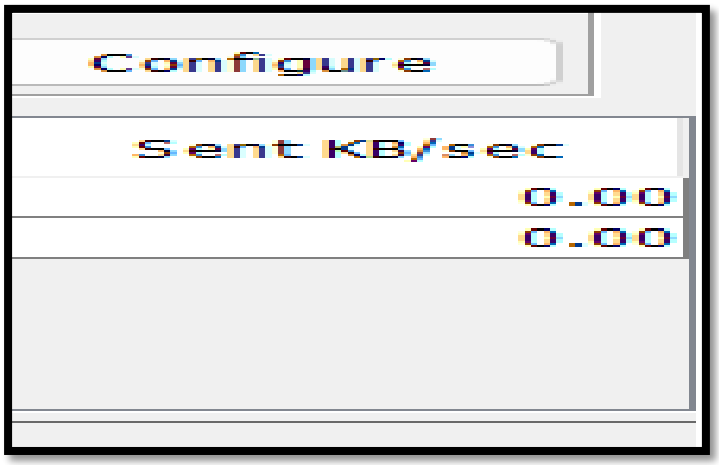
- **RECEIVED BYTES**

  BYTES RECEIVED AFTER REQUEST IS SENT



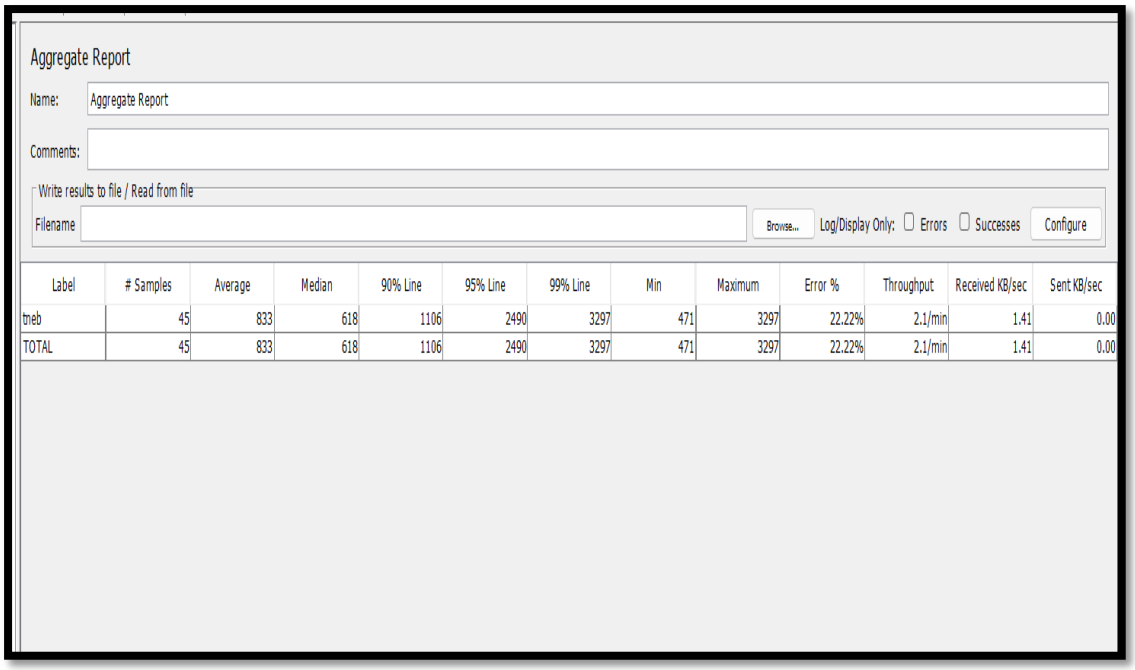
- **SENT BYTES**

  BYTES ENT FROM JMETER SIDE TO CONNECT TO THE WEBSITE



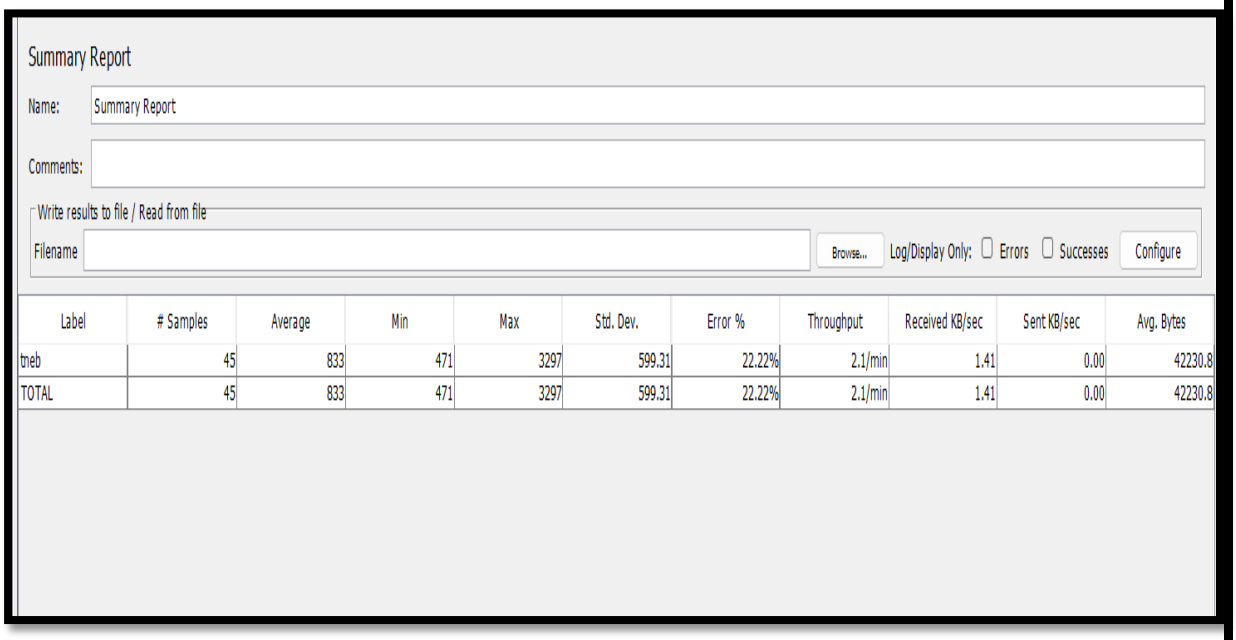
LISTENER:AGGREGATE REPORT

THIS LISTENER GIVES THE SAME RESULT AS AGGREGATE GRAPH BUT WITHOUT A GRAPHICAL REPRESENATION



LISTENER:SUMMARY REPORT

THIS LISTENER GIVE STHE REPORT OF ALL THE ACTIONS PERFORMED,THAT IS IT GIVES A CUMULATIVE REPORT OF ALL THE LISTENER'S CONTENT

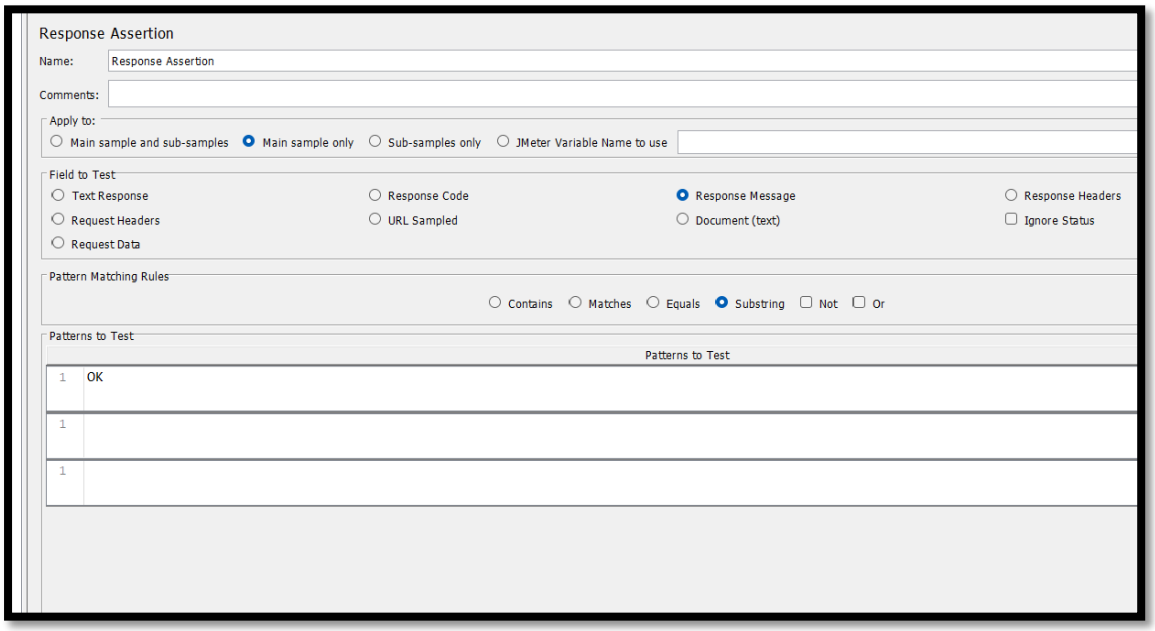


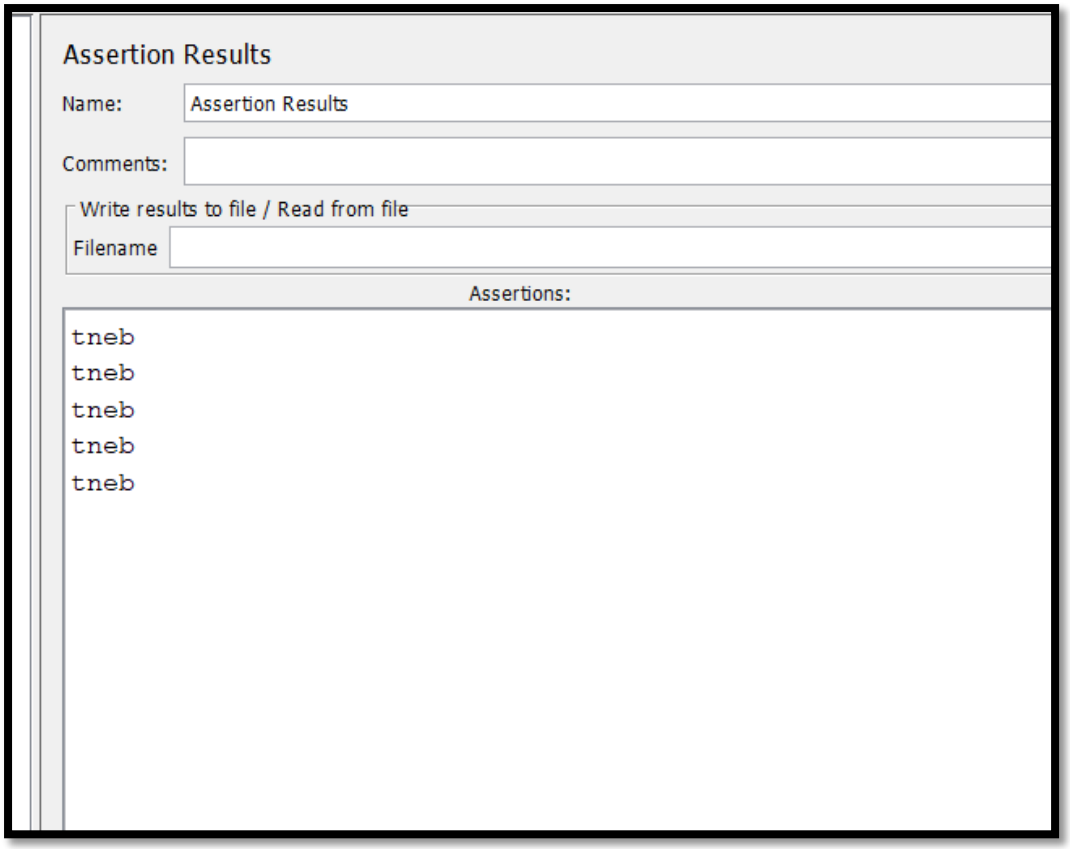
ASSERTIONS:

USED TO VALIDATE THE RESPONSE TO THE REQUEST THAT WE SENT TO THE SERVER

RESPONSE ASSERTION:

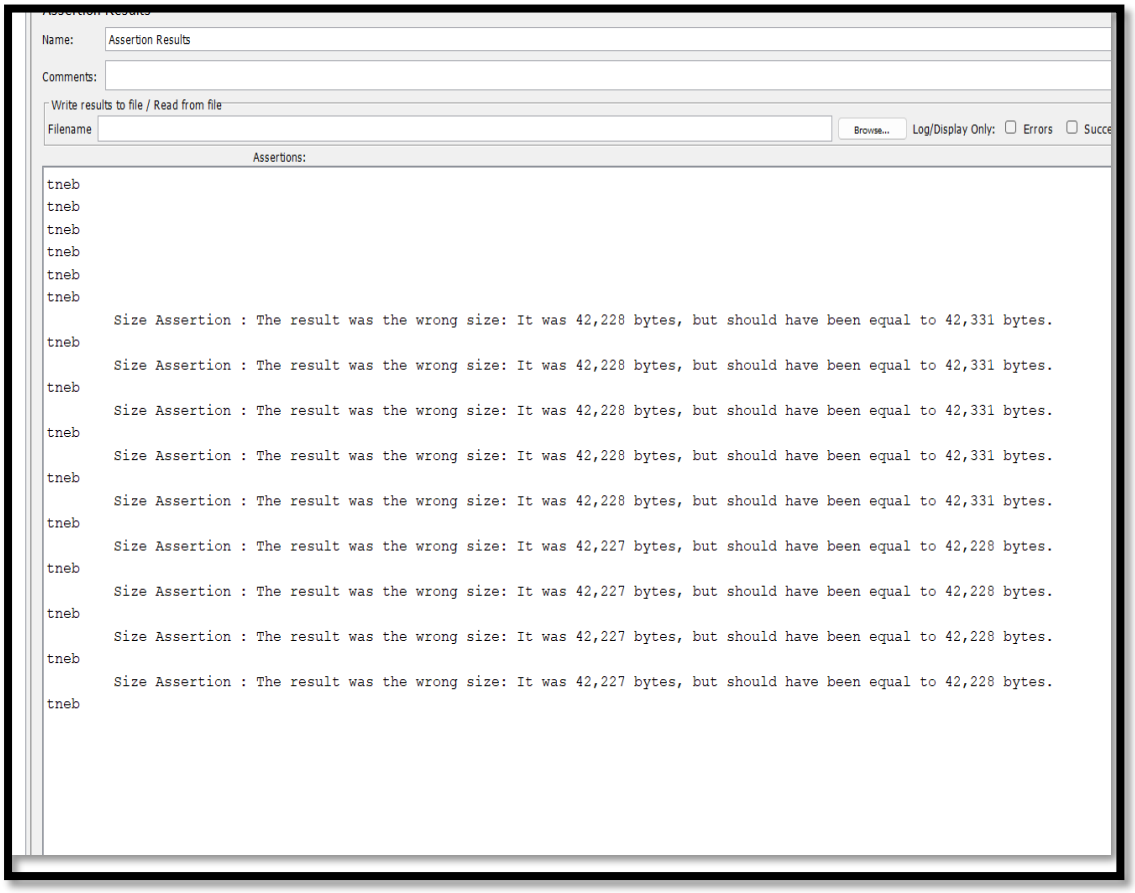
ALLOWS TO VALIDATE THAT THE RESPONSE IS EQUAL TO,MATCHES,GREATER THAN ETC TO THE SPECIFIED RESPONSE

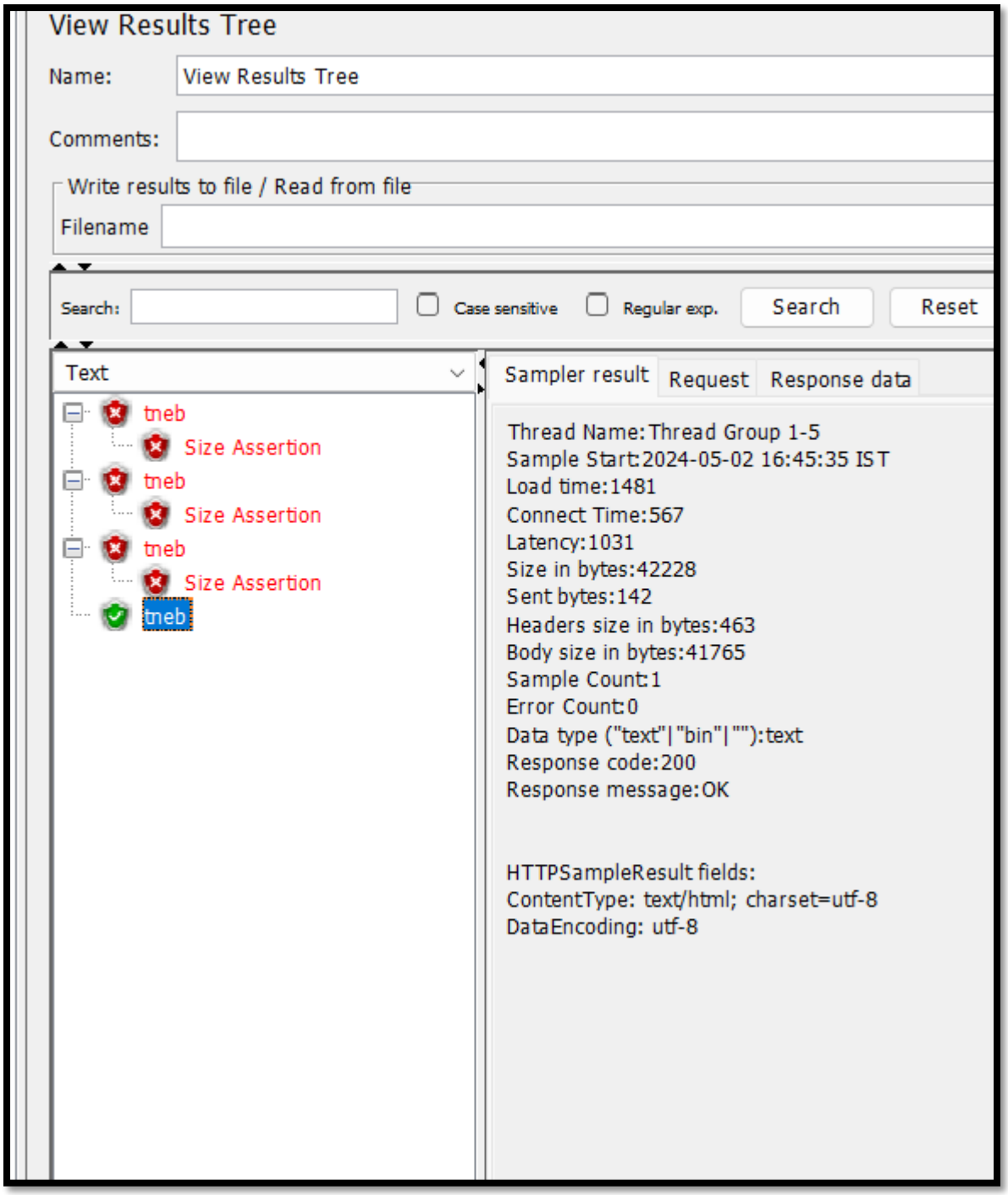




SIZE ASSERTION:

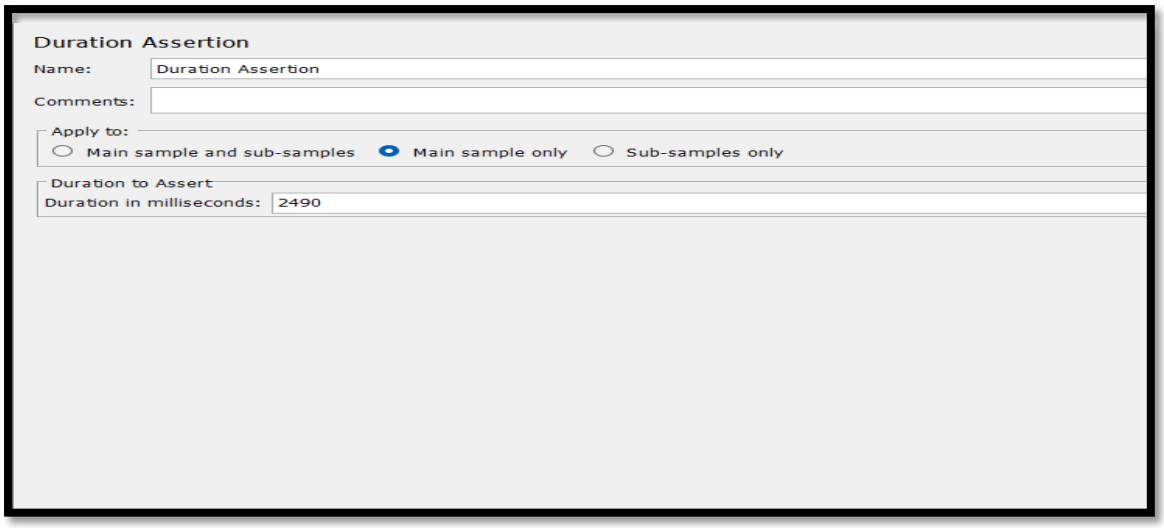
CHECKS THAT THE RESPONSE CONTAINS THE SPECIFIED SIZE

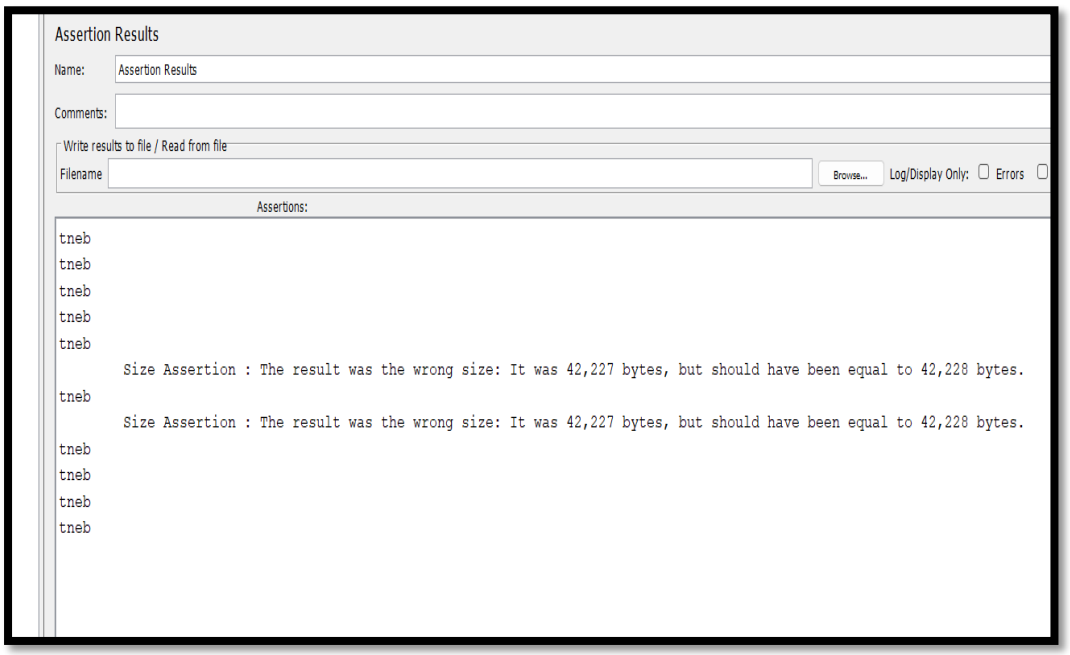




DURATION ASSERTION:

CHECKS WHETHER THE RESPONSE CONTAIN THE SPECIFIED AMOUNT OF TIME





SIMPLE DATA WRITER:

WRITES THE REPORT TO A CSV FILE

Fields

Column type:

	Standard	Standard	Standard	Standard	Standard
45	1714648375929	2379	tneb	200	OK
46	1714648377928	736	tneb	200	OK
47	1714648379928	829	tneb	200	OK
48	1714648393062	756	tneb	200	OK
49	1714648395062	761	tneb	200	OK
50	1714648397062	804	tneb	200	OK
51	1714648399063	1404	tneb	200	OK
52	1714648401064	3356	tneb	200	OK
53	1714648502638	1271	tneb	200	OK
54	1714648504633	989	tneb	200	OK
55	1714648506632	3679	tneb	200	OK
56	1714648508629	3227	tneb	200	OK
57	1714648510626	2869	tneb	200	OK
58	1714648527032	969	tneb	200	OK
59	1714648529026	989	tneb	200	OK
60	1714648531026	1002	tneb	200	OK
61	1714648533028	1314	tneb	200	OK
62	1714648535028	1481	tneb	200	OK
63	1714648664048	700	tneb	200	OK
64	1714648666033	579	tneb	200	OK
65	1714648668043	518	tneb	200	OK
66	1714648670038	859	tneb	200	OK
67	1714648672034	482	tneb	200	OK
68	1714648674416	493	tneb	200	OK
69	1714648676406	1745	tneb	200	OK
70	1714648678421	456	tneb	200	OK
71	1714648680412	546	tneb	200	OK
72	1714648682410	880	tneb	200	OK

GENERATE HTML REPORT:

GENERATES HTML REPORT

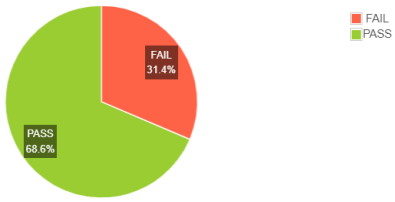
Test and Report information

Source file	"report1.csv"
Start Time	"4/20/24, 12:06 PM"
End Time	"5/2/24, 4:51 PM"
Filter for display	""

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.337	**500 ms**	**1 sec 500 ms**	**Total**
0.337	500 ms	1 sec 500 ms	tneb

Requests Summary



Statistics

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**86**	**27**	**31.40%**	**1488.77**	**456**	**42451**	**642.50**	**2484.30**	**3565.95**	**42451.00**	**0.00**	**0.00**	**0.00**
tneb	86	27	31.40%	1488.77	456	42451	642.50	2484.30	3565.95	42451.00	0.00	0.00	0.00

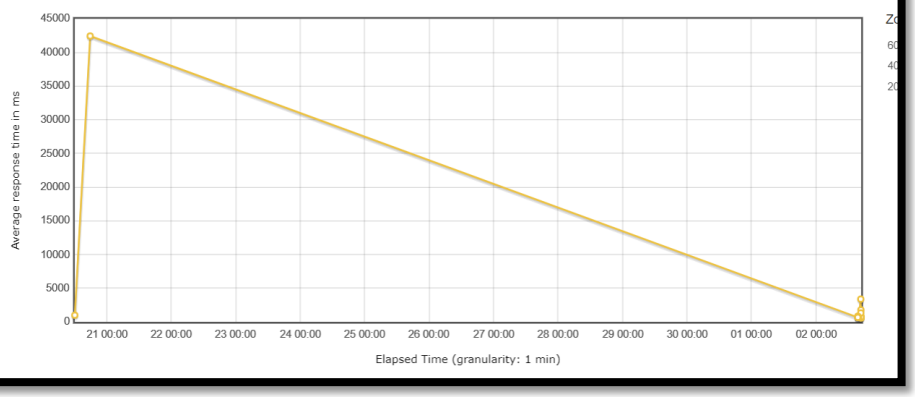
Top 5 Errors by sampler

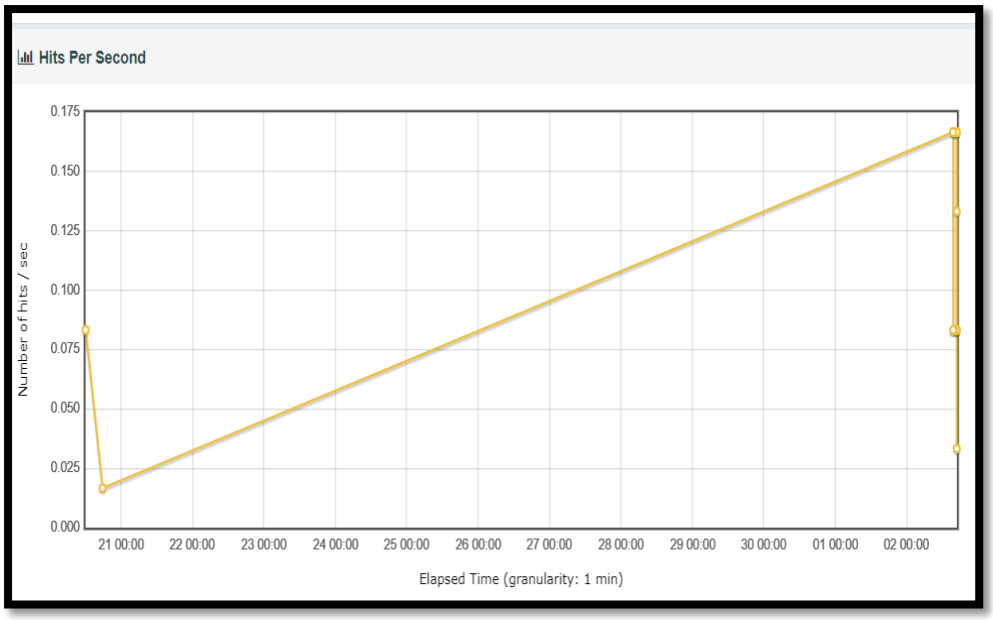
Sample	#Samples	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors	Error	#Errors
Total	**86**	**27**	**The result was the wrong size: It was 42,227 bytes, but should have been equal to 42,228 bytes.**	**6**	**503/Service Unavailable**	**5**	**The result was the wrong size: It was 42,228 bytes, but should have been equal to 42,331 bytes.**	**5**	**The result was the wrong size: It was 42,231 bytes, but should have been equal to 42,331 bytes.**	**5**	**Error parsing variable name: value: null**	**4**
tneb	86	27	The result was the wrong size: It was 42,227 bytes, but should have been equal to 42,228 bytes.	6	503/Service Unavailable	5	The result was the wrong size: It was 42,228 bytes, but should have been equal to 42,331 bytes.	5	The result was the wrong size: It was 42,231 bytes, but should have been equal to 42,331 bytes.	5	Error parsing variable name: value: null	4

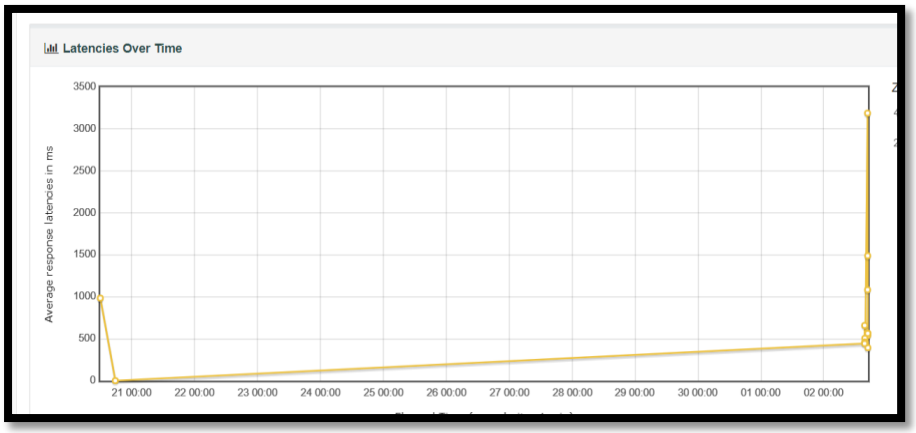
Errors

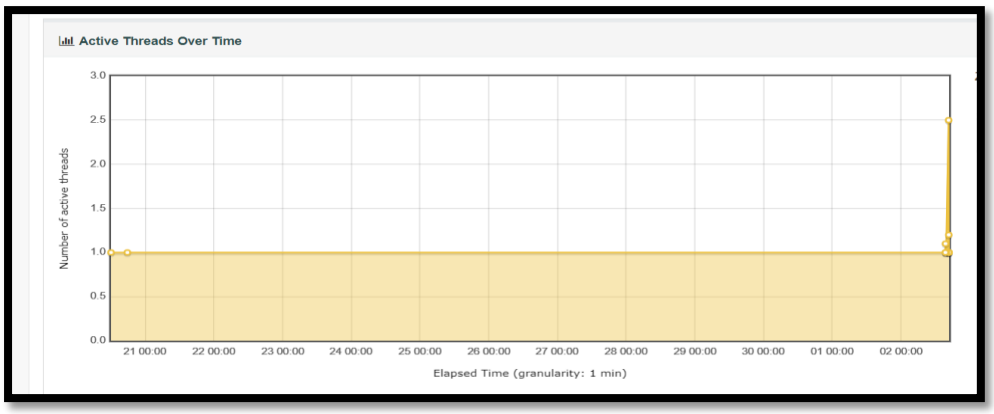
Type of error	Number of errors	% in errors	% in all samples
The result was the wrong size: It was 42,227 bytes, but should have been equal to 42,228 bytes.	6	22.22%	6.98%
503/Service Unavailable	5	18.52%	5.81%
The result was the wrong size: It was 42,228 bytes, but should have been equal to 42,331 bytes.	5	18.52%	5.81%
The result was the wrong size: It was 42,231 bytes, but should have been equal to 42,331 bytes.	5	18.52%	5.81%
Error parsing variable name: value: null	4	14.81%	4.65%
The operation lasted too long: It took 2,730 milliseconds, but should not have lasted longer than 2,490 milliseconds.	1	3.70%	1.16%
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to www.tangedco.org:443 [www.tangedco.org/103.234.149.151, www.tangedco.org/64.ff9b:0:0:0:0:67ea:9597] failed: Connection timed out: connect	1	3.70%	1.16%

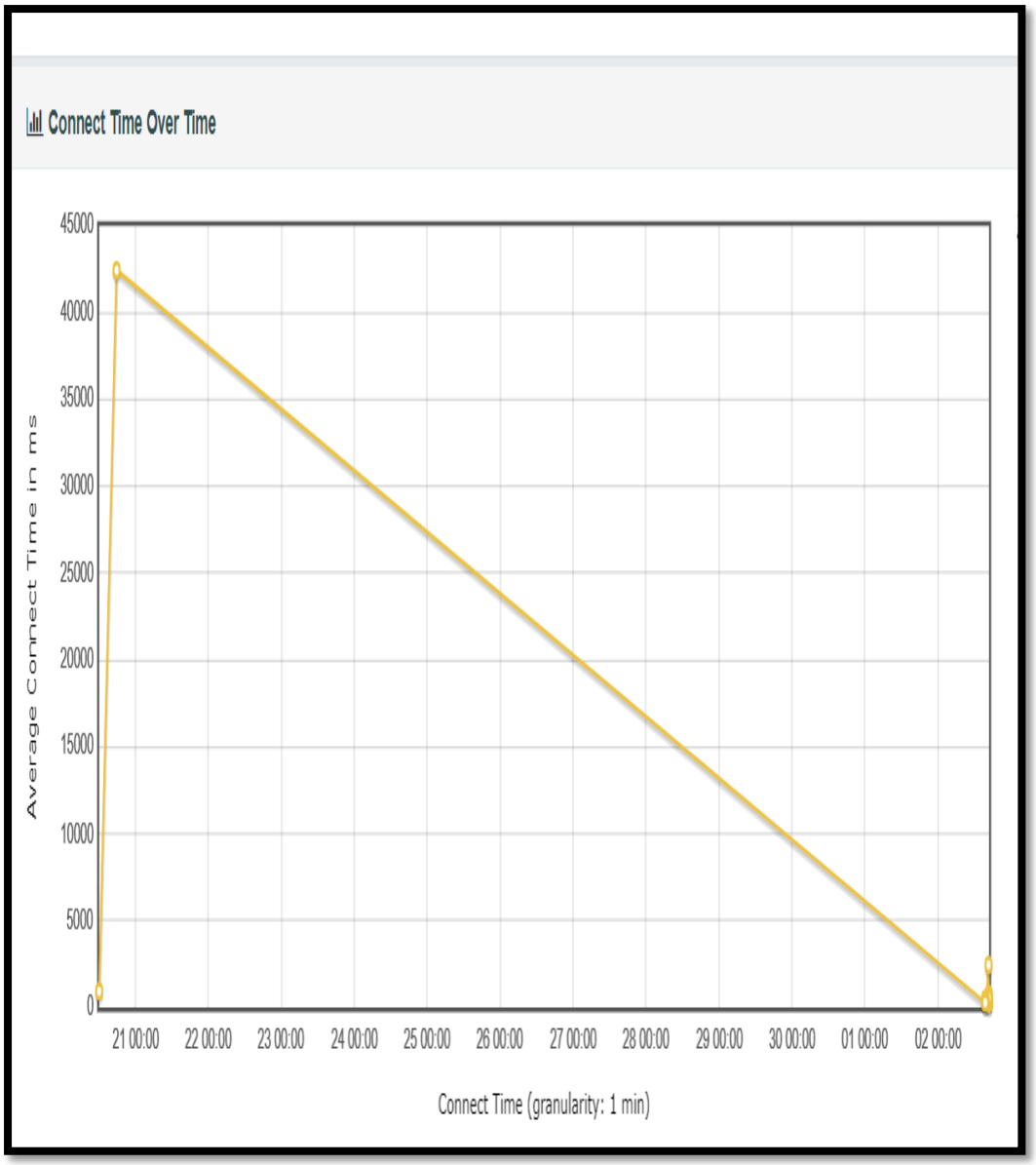
Response Times Over Time

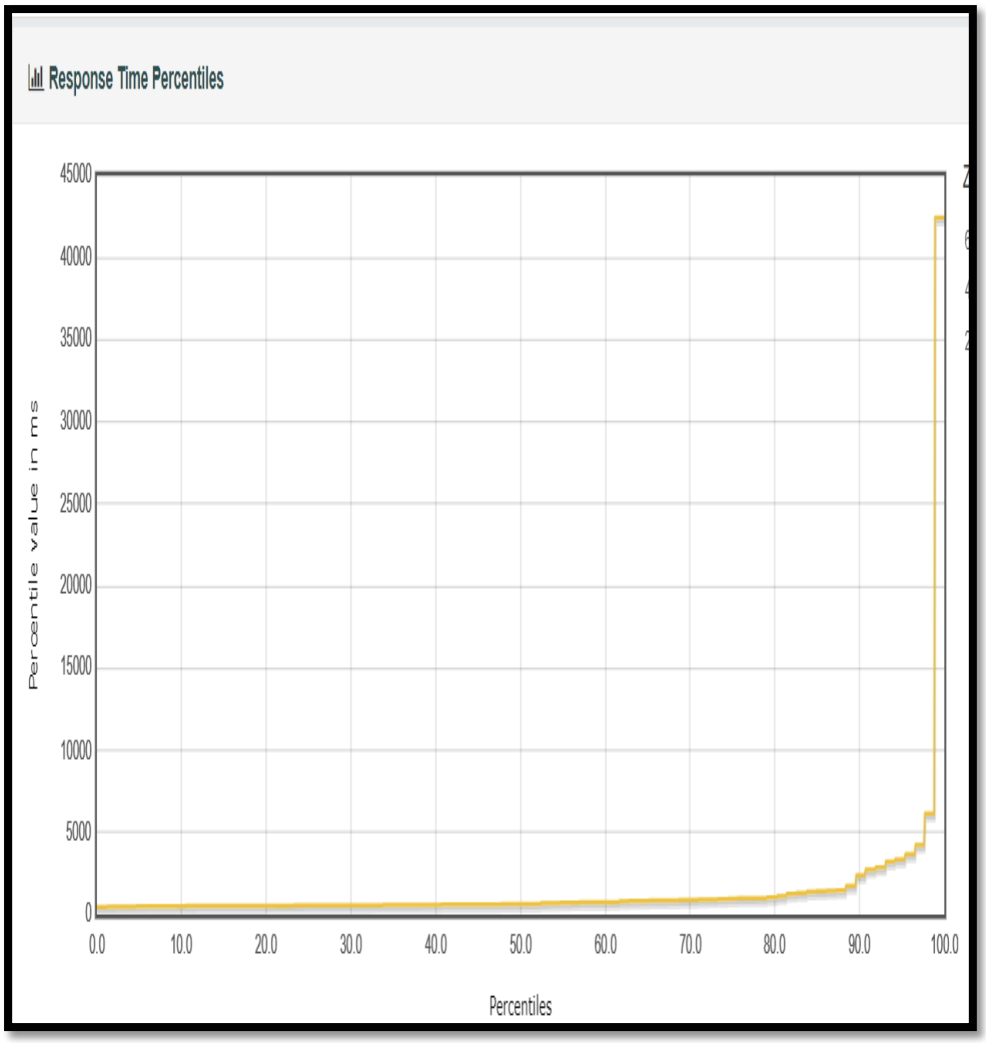



Hits Per Second
Number of hits / sec
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
21 00:00
22 00:00
23 00:00
24 00:00
25 00:00
26 00:00
27 00:00
28 00:00
29 00:00
30 00:00
01 00:00
02 00:00
Elapsed Time (granularity: 1 min)


Latencies Over Time
Average response latencies in ms
3500
3000
2500
2000
1500
1000
500
0
21 00:00
22 00:00
23 00:00
24 00:00
25 00:00
26 00:00
27 00:00
28 00:00
29 00:00
30 00:00
01 00:00
02 00:00


Active Threads Over Time
Number of active threads
3.0
2.5
2.0
1.5
1.0
0.5
0.0
21 00:00
22 00:00
23 00:00
24 00:00
25 00:00
26 00:00
27 00:00
28 00:00
29 00:00
30 00:00
01 00:00
02 00:00
Elapsed Time (granularity: 1 min)


Connect Time Over Time
Average Connect Time in ms
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
21 00:00
22 00:00
23 00:00
24 00:00
25 00:00
26 00:00
27 00:00
28 00:00
29 00:00
30 00:00
01 00:00
02 00:00
Connect Time (granularity: 1 min)

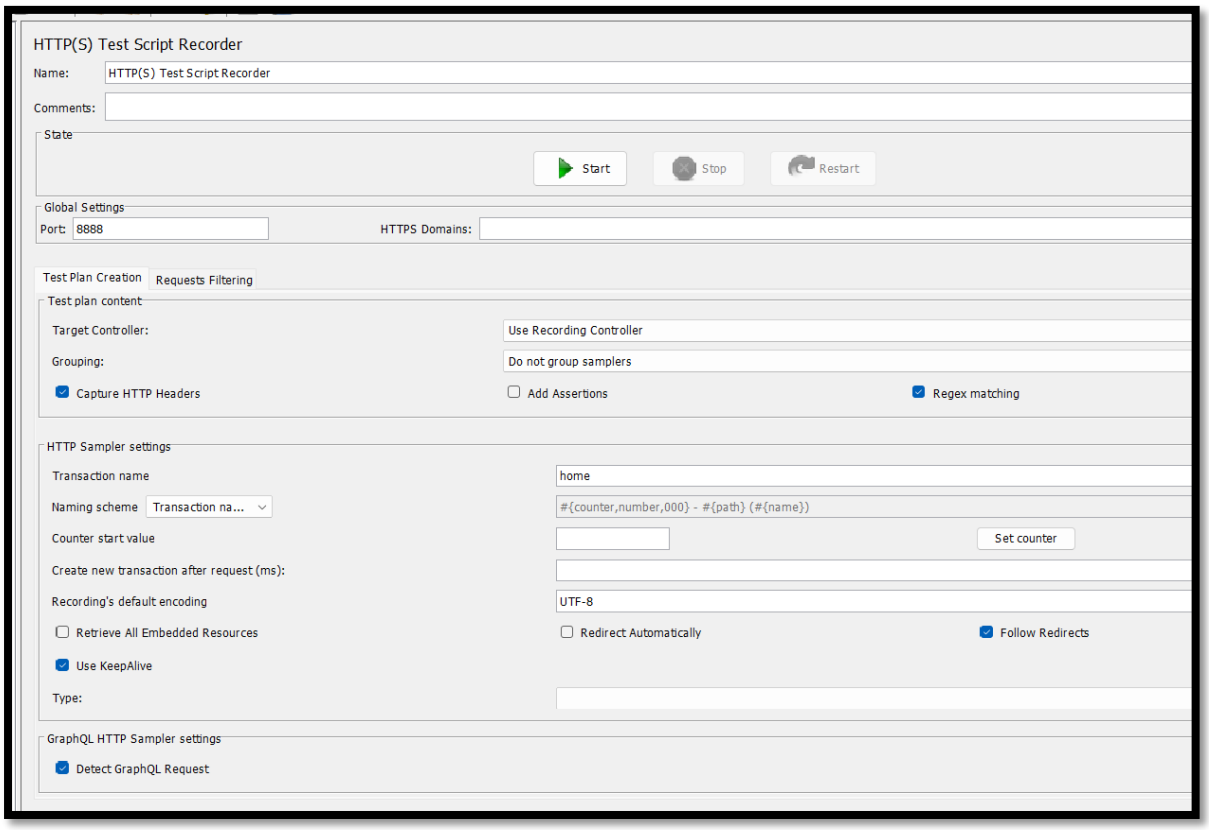

Response Time Percentiles
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
Percentile value in ms
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
Percentiles

RECORDING THE HTTPS REQUEST:

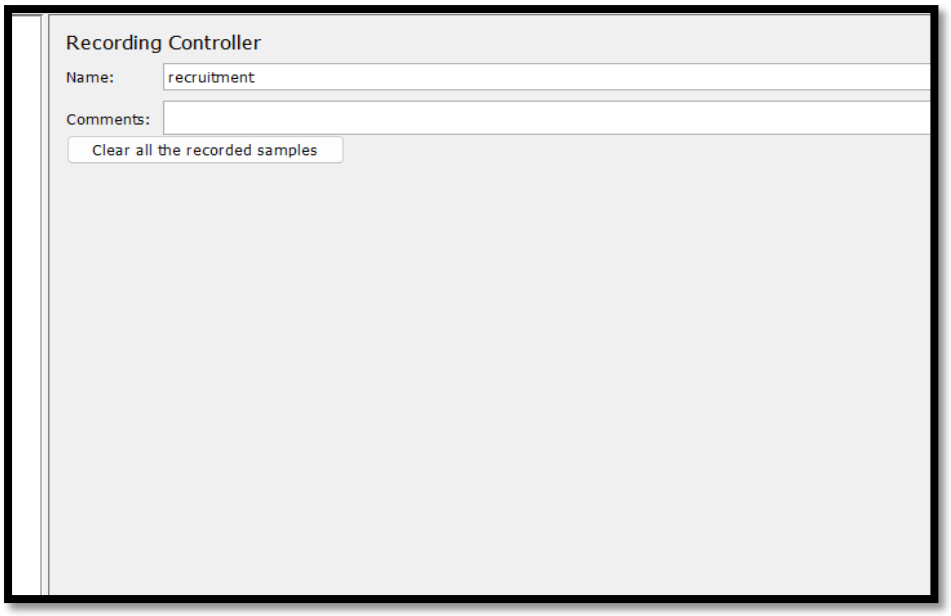
1.CHANGE THE PROXY SETTINGS IN THE RESPECTIVE BROWSER

2.INSTALL JMETER ROOT CA CERTIFICATE

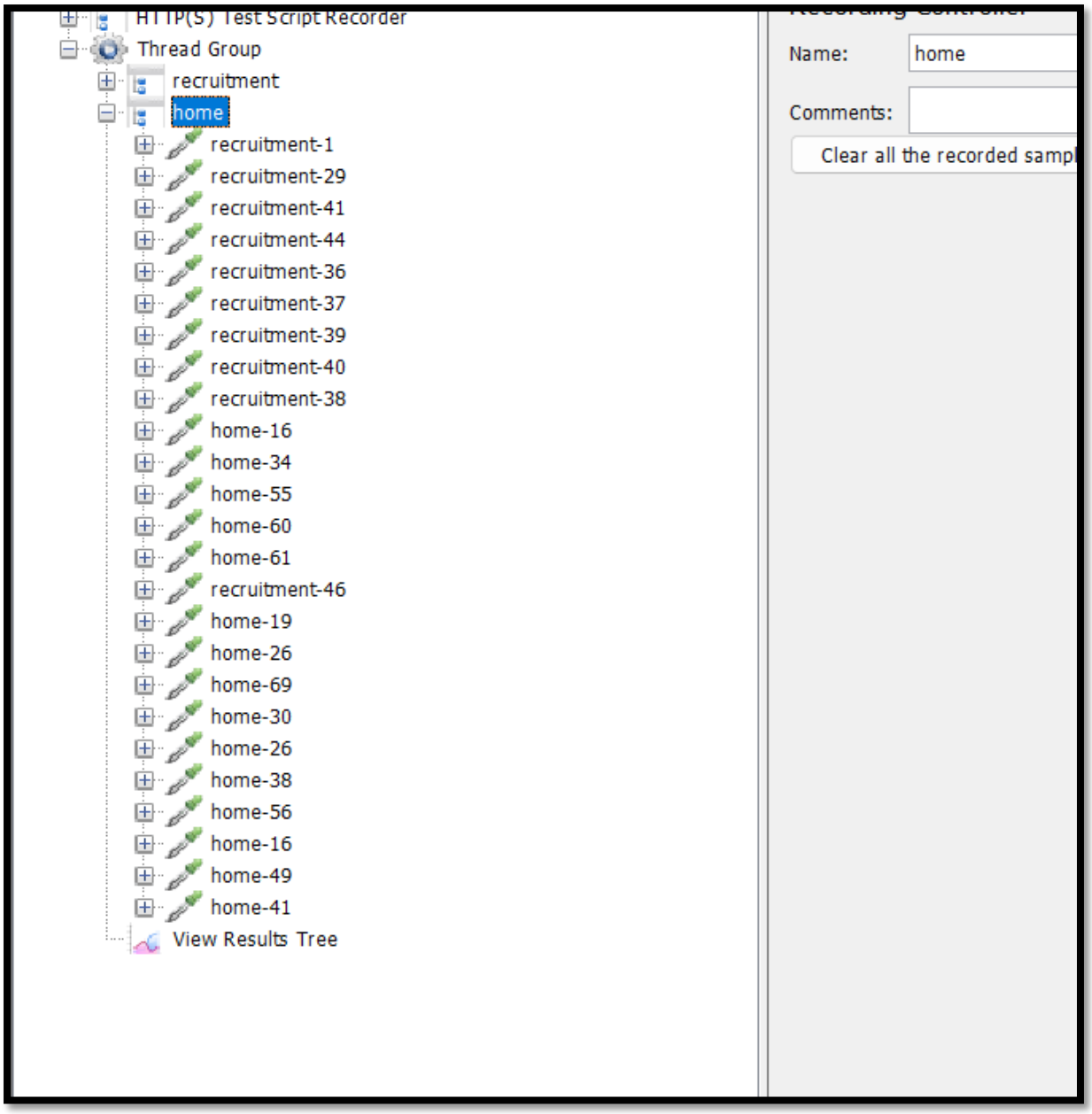
ADD HTTPS TEST SCRIPT RECORDER:



ADD RECORDING CONTROLLER:

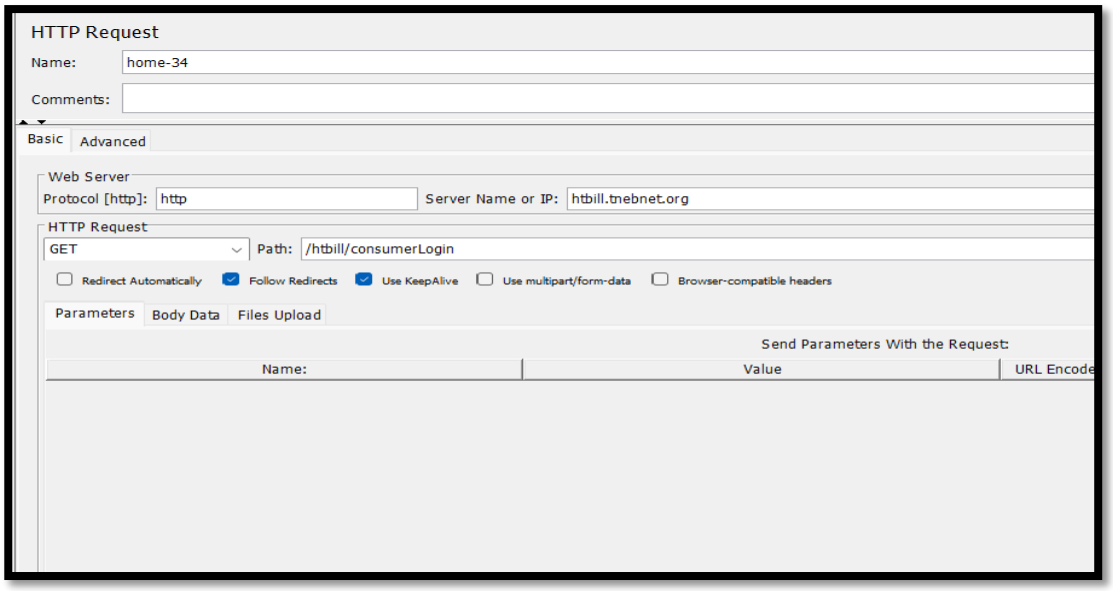


RECORDED CONTENTS:

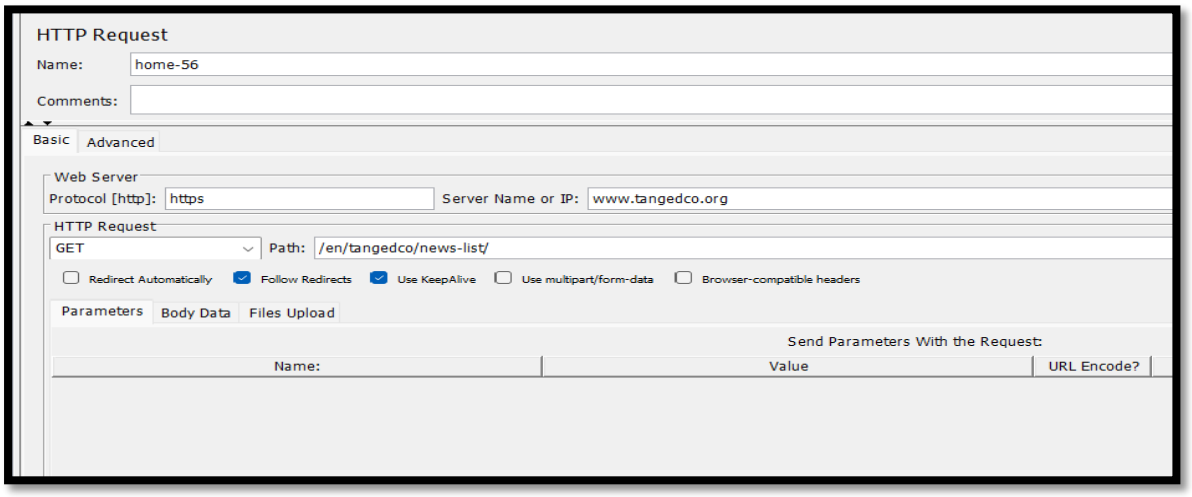


EACH PATH SHOWS THE RECORDING OF EACH PAGE:

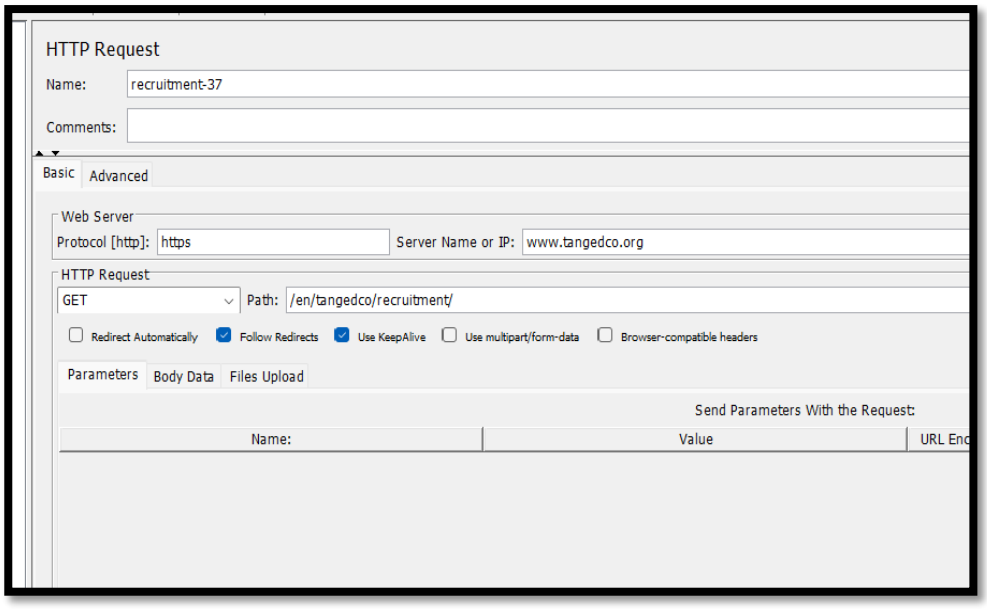
- **CONSUMER LOGIN PAGE**

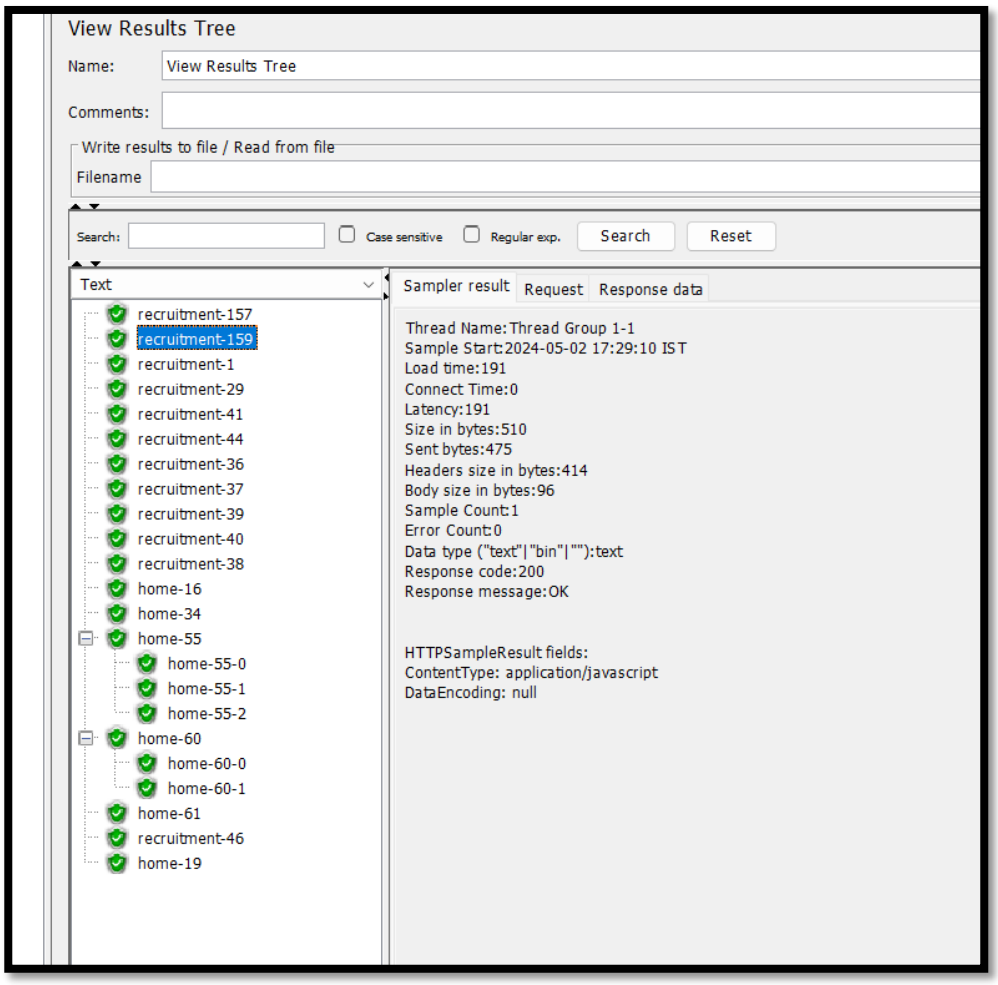


- **NEWS LIST PAGE:**



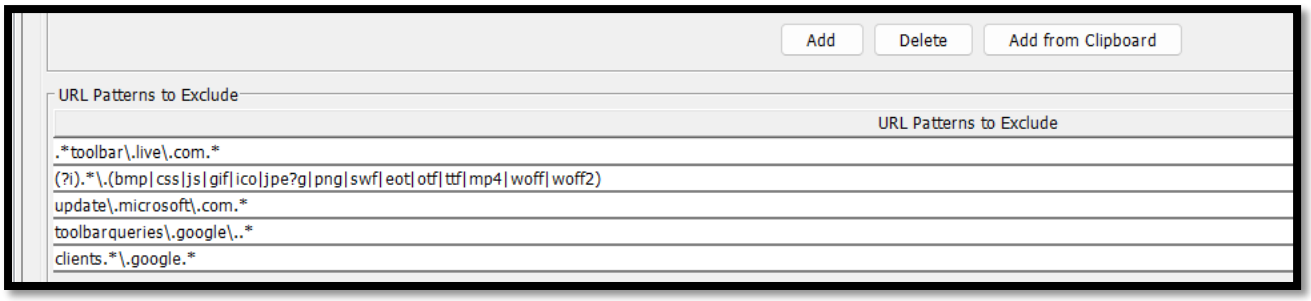
- **RECRUITMENT PAGE:**



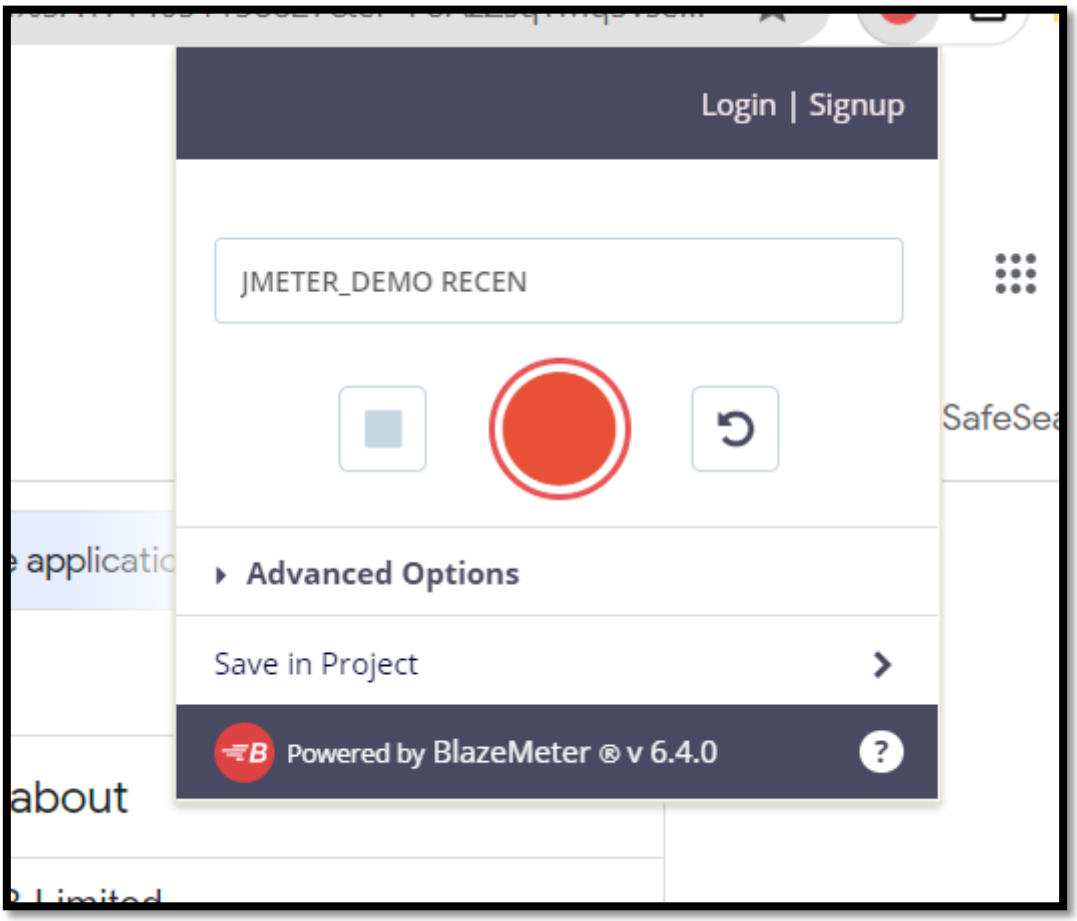


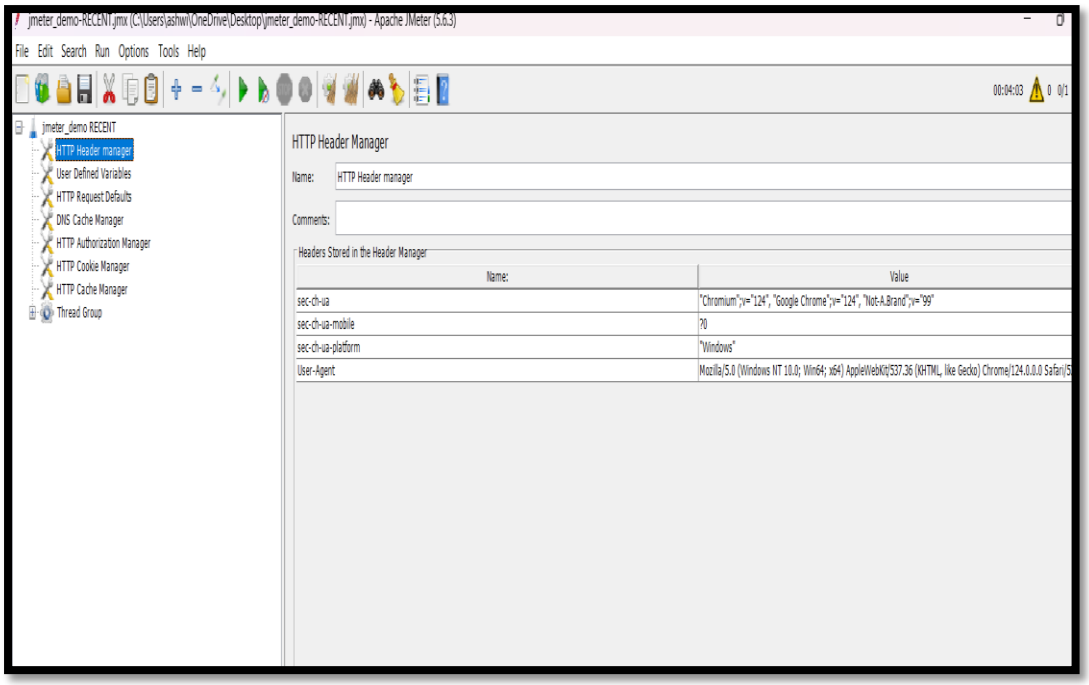
REQUESTS FILTERING:

EXCLUDES THE URL PATTERNS THAT ARE NOT NEEDED

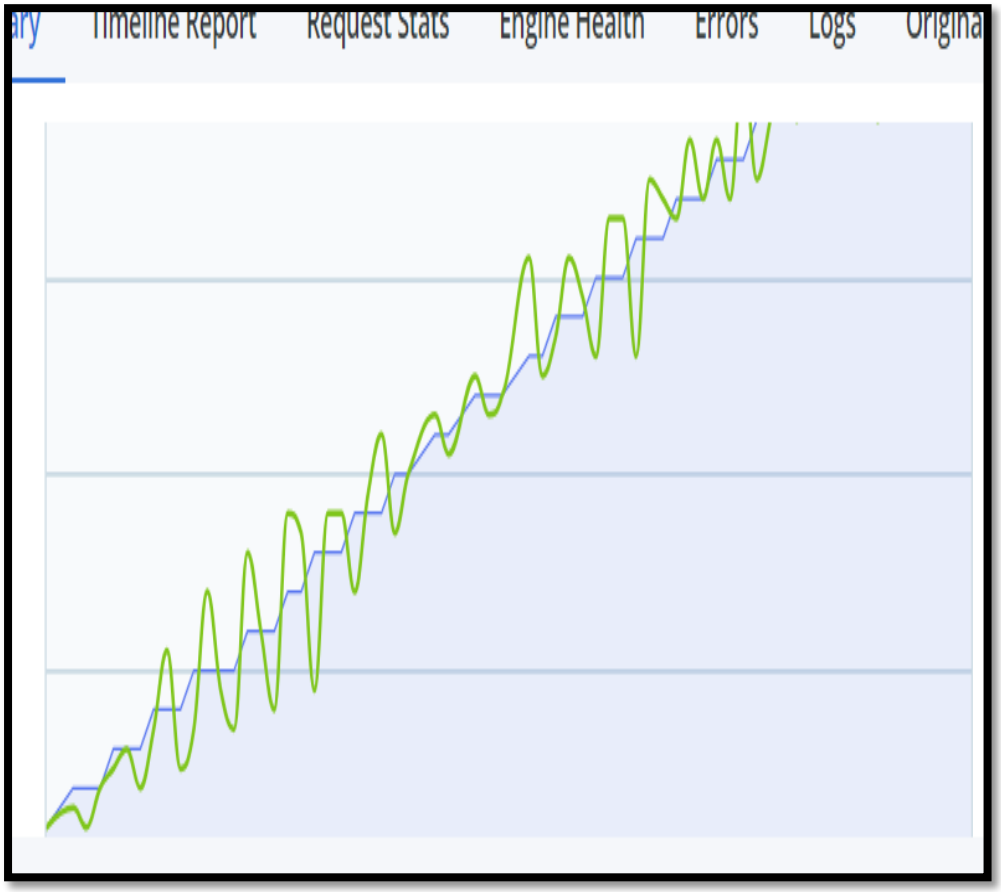


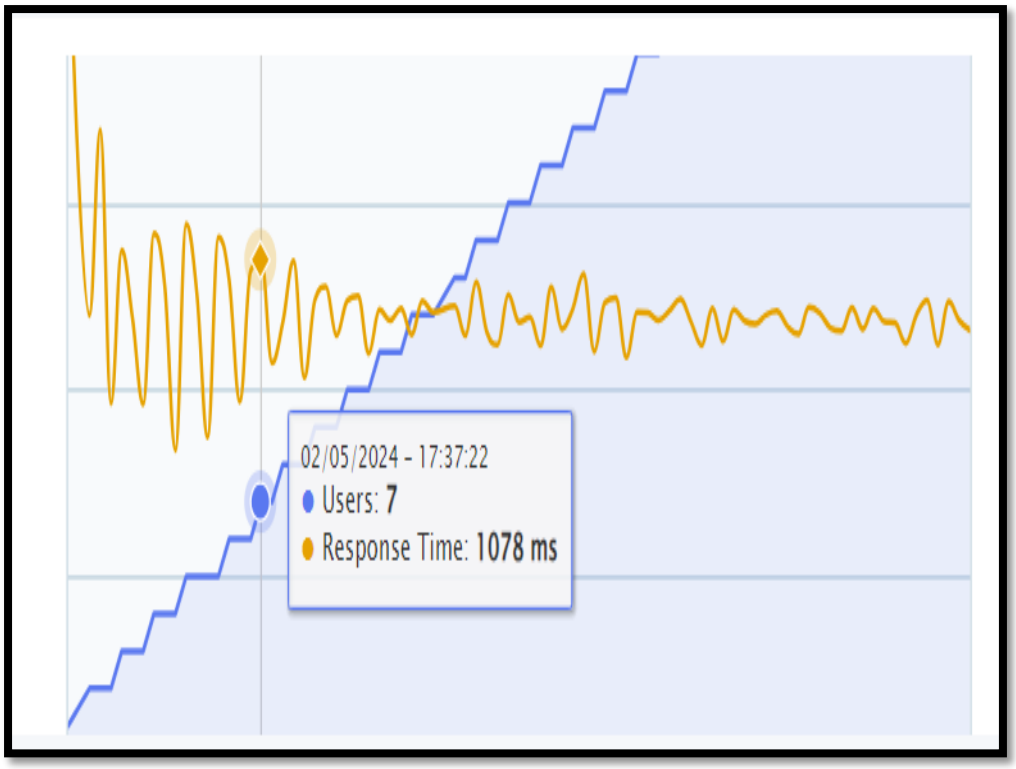
RECORDING USING BLAZEMETER EXTENSION:

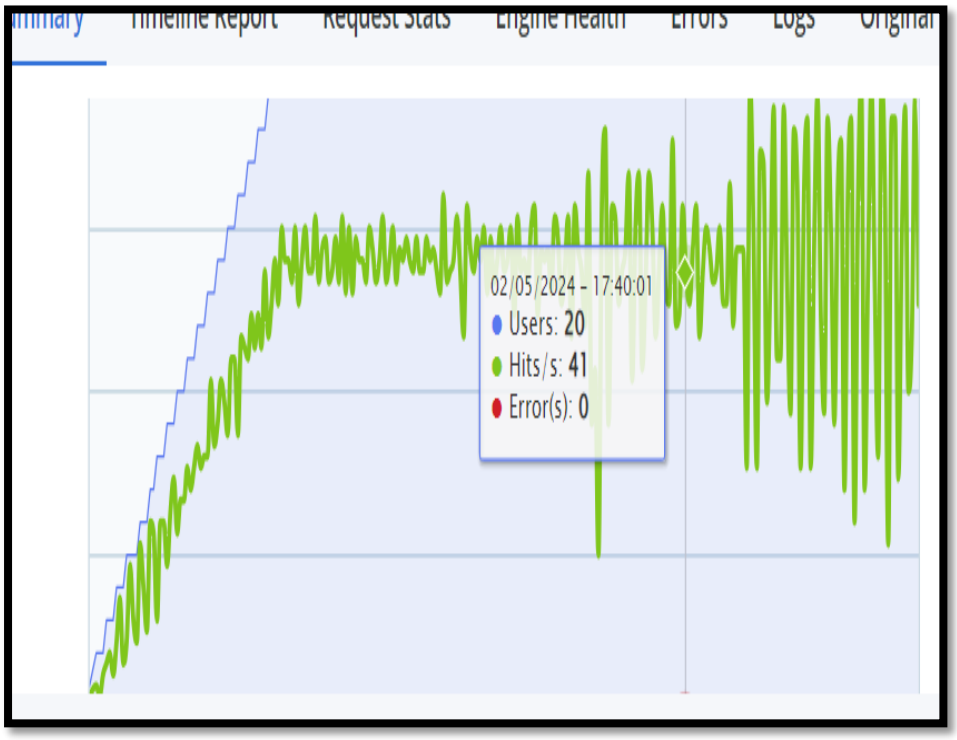


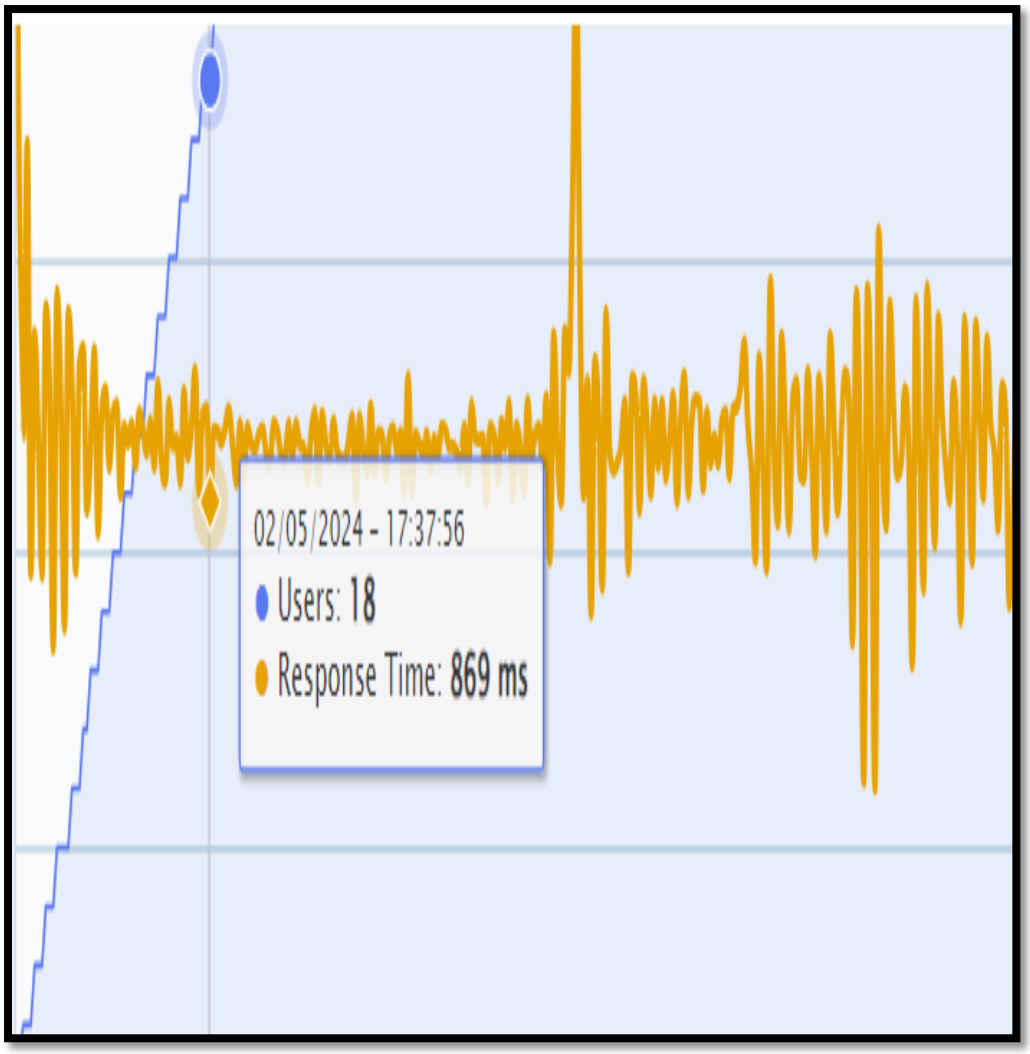


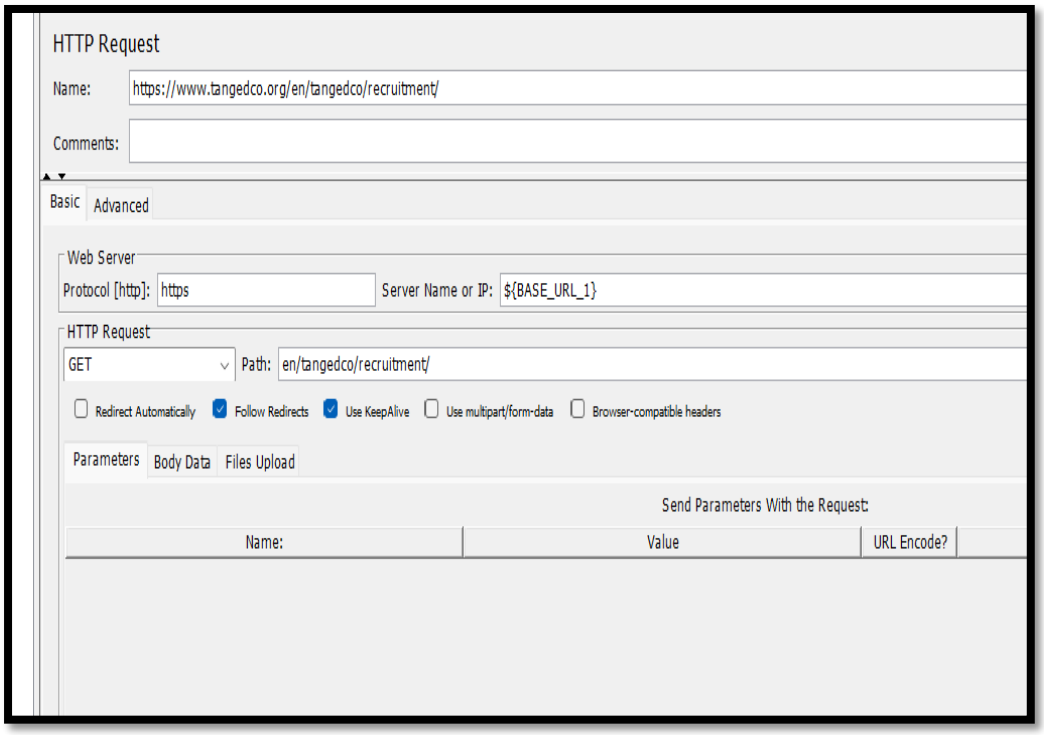
THE ADVANTAGE OF BLAZEMETER IS THAT IT ADDS THE MAJOR CONFIG ELEMENTS DEFAULTLY




02/05/2024 - 17:37:22
Users: 7
Response Time: 1078 ms


02/05/2024 - 17:40:01
Users: 20
Hits/s: 41
Error(s): 0


02/05/2024 - 17:37:56
Users: 18
Response Time: 869 ms

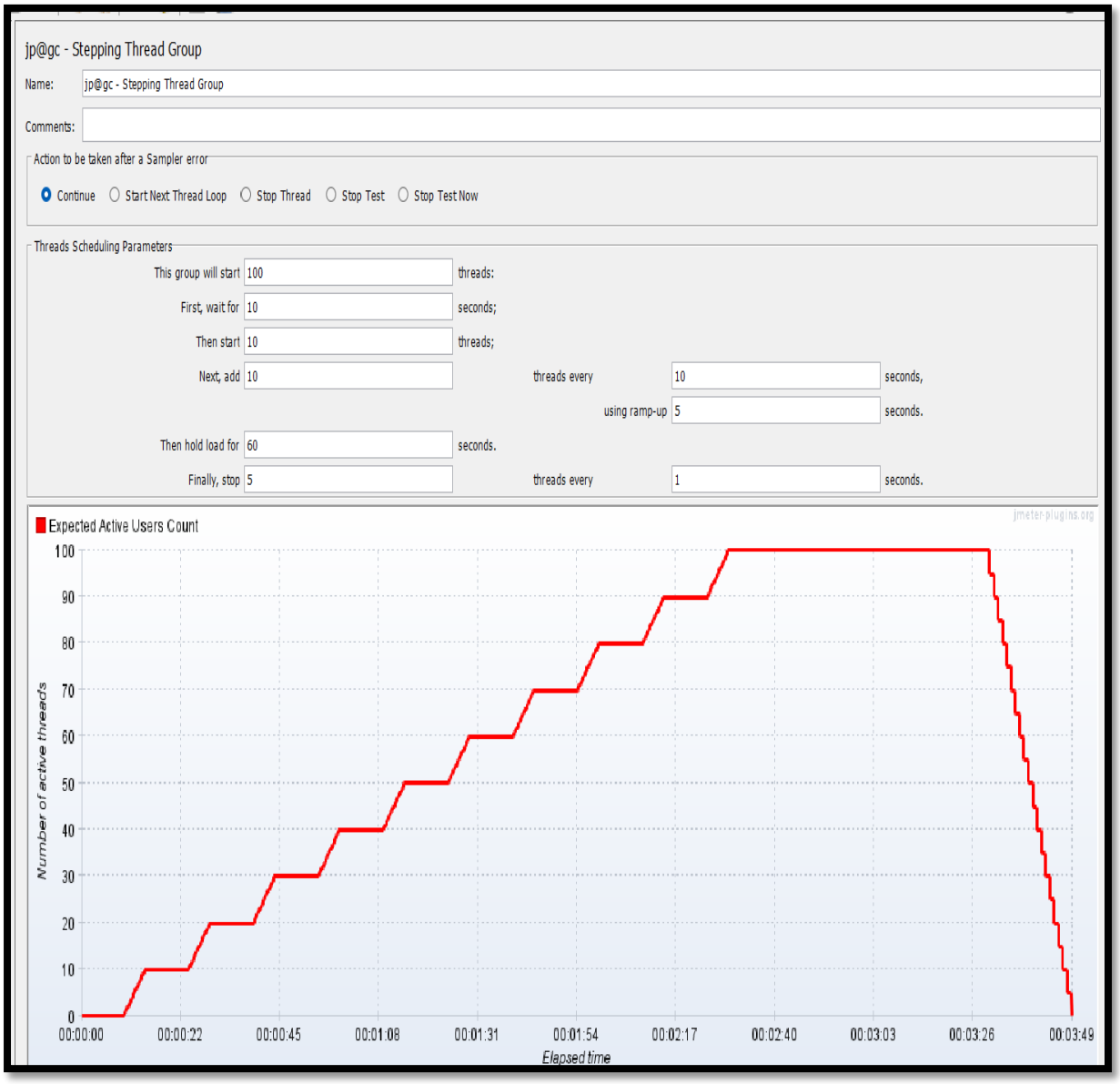

HTTP Request
Name: https://www.tangedco.org/en/tangedco/recruitment/
Comments:
Basic
Advanced
Web Server
Protocol [http]: https
Server Name or IP: ${BASE_URL_1}
HTTP Request
GET
Path: en/tangedco/recruitment/
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters
Body Data
Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?

BLAZEMETER ADDS THE WEBSITE' S URL DEFAULTY USING USER DEFINED VARIABLES

JMETER'S PLUGIN MANAGER:

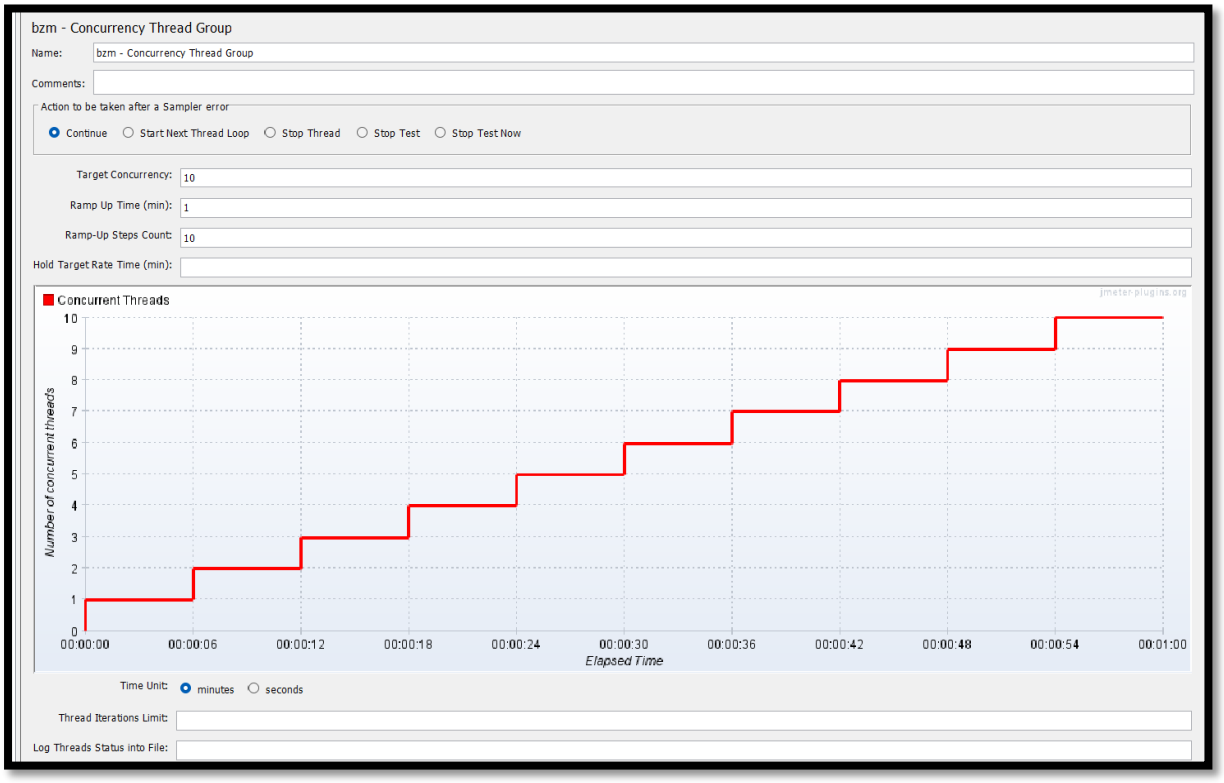
STEPPING THREAD GROUP:

HELPS TO GRADUALLY INCREASE THE LOAD OF APPLICATION



CONCURRENCY THREAD GROUP:

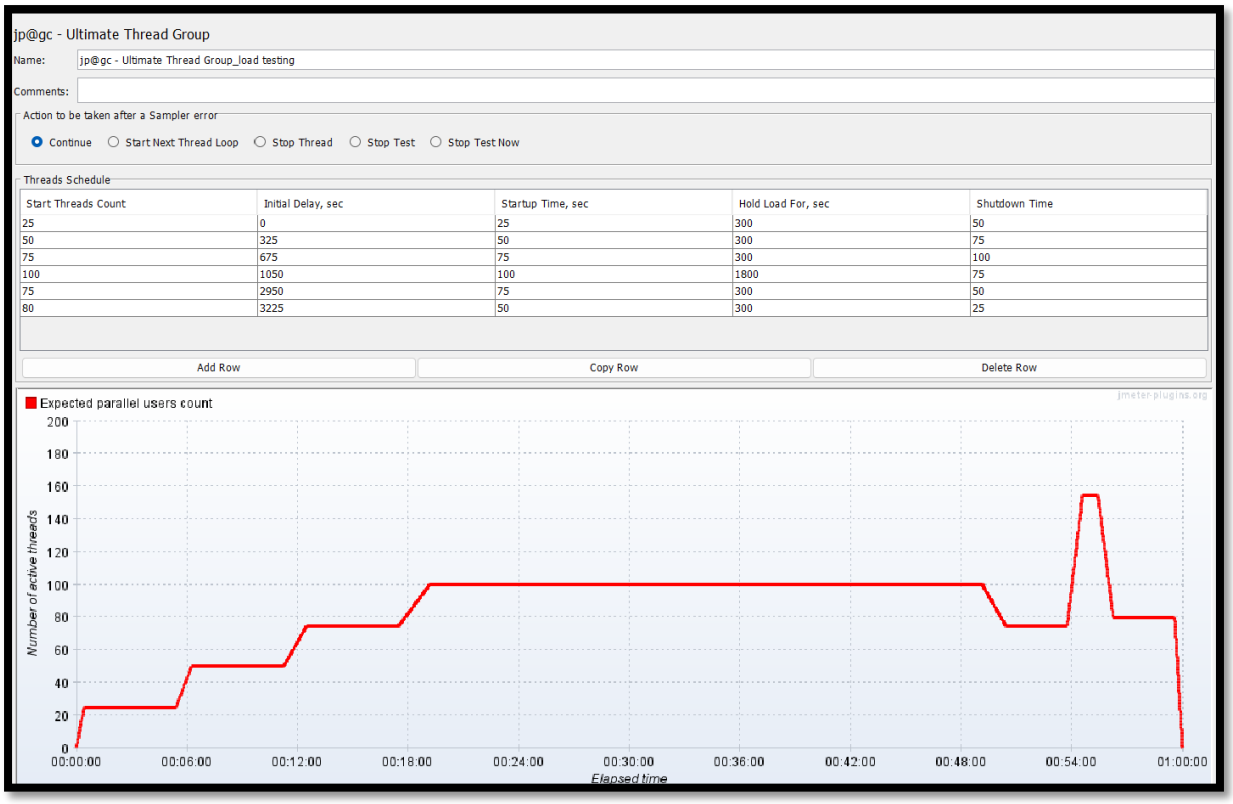
HELPS TO KNOW THE PERFORMANCE OF THE WEBSITE WHEN THERE ARE CONCURRENT NO OF USERS



ULTIMATE THREAD GROUP:

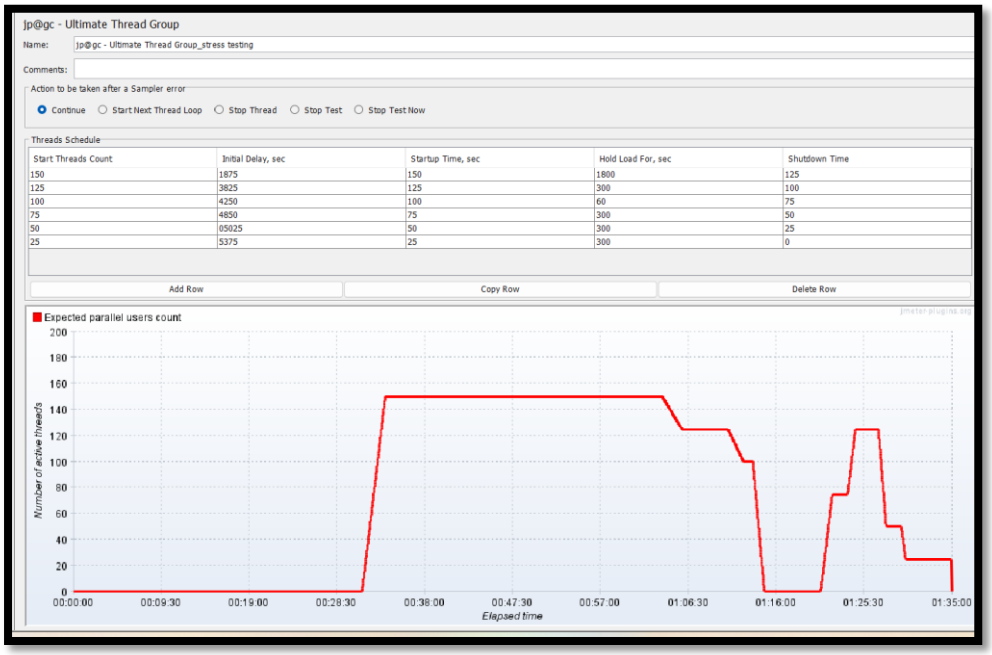
LOAD TESTING:

INCREASING THE LOAD THEREBY TO KNOW THE CAPACITY OF THE WEBSITE



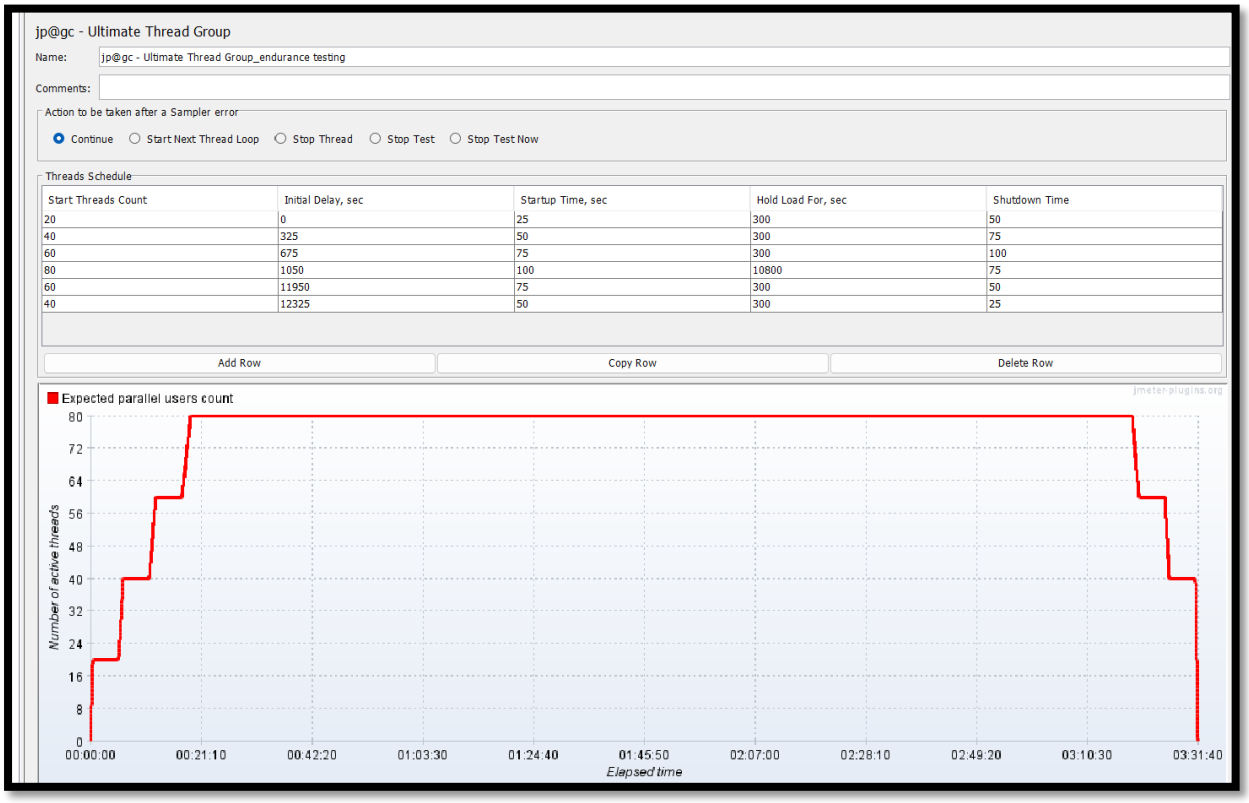
STRESS TESTING:

INCREASING THE NO OF THREADS MORE THAN LOAD THEREBY TO KNOW THE BREAKING POINT



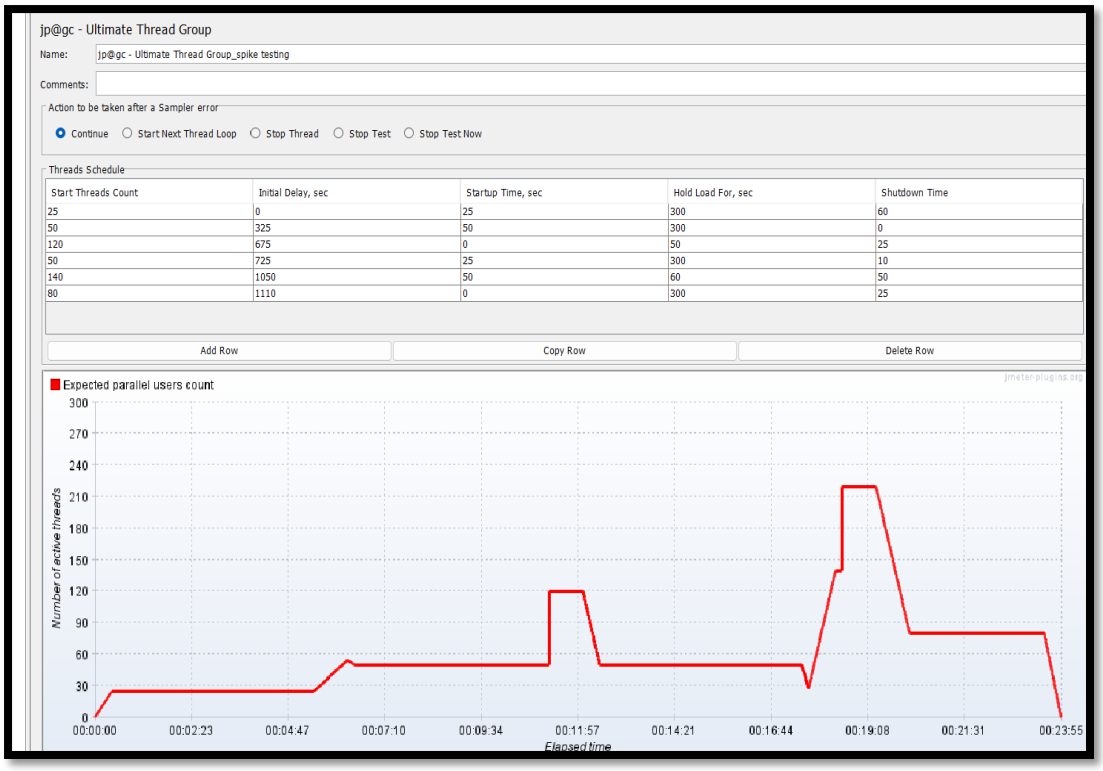
ENDURANCE TESTING:

INCREASING THE NO OF HOURS

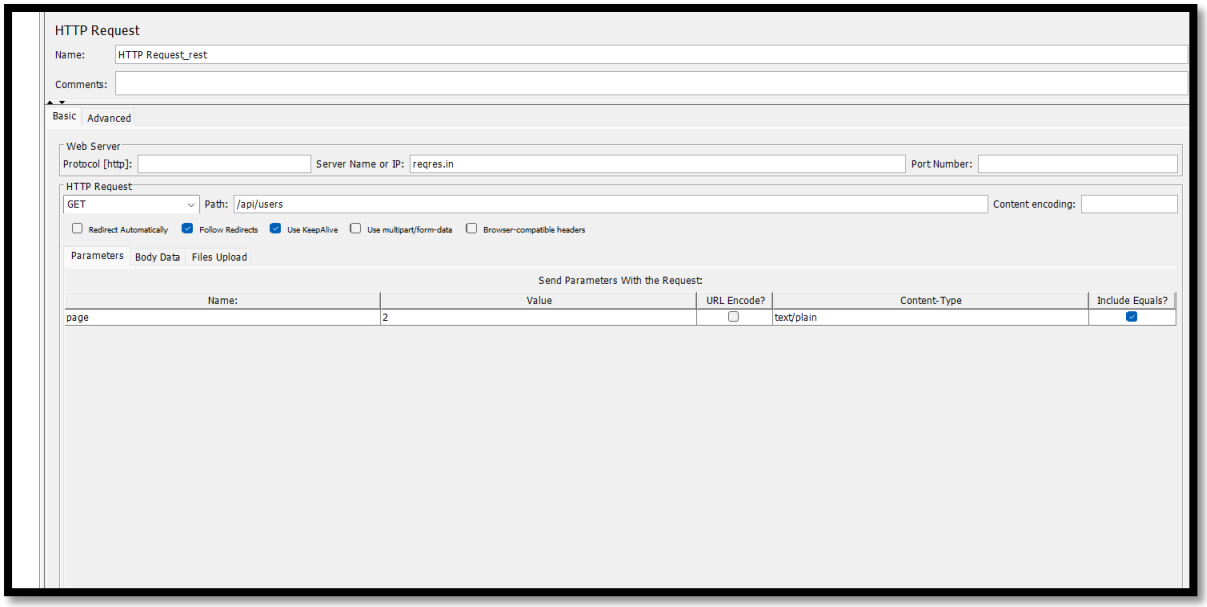


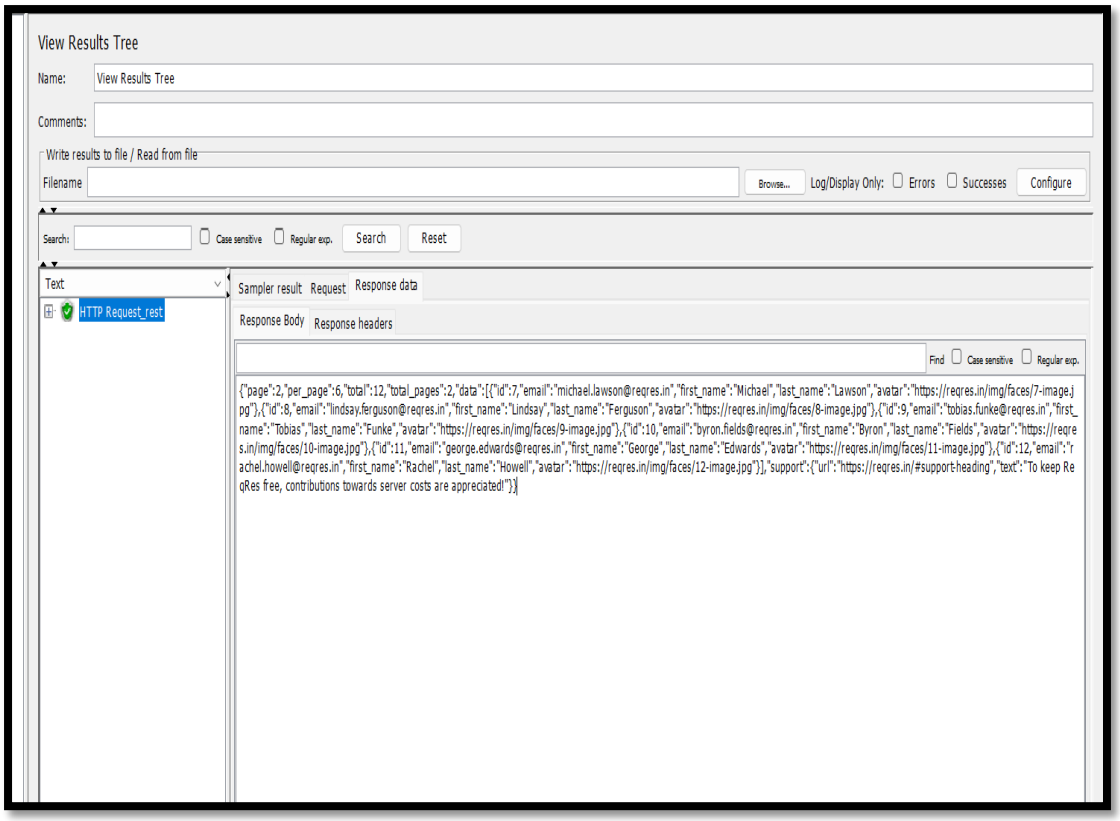
SPIKE TESTING:

INCREASING THE NO OF THREADS OR USERS ABRUPTLY



REST API TESTING:




View Results Tree
Name:
View Results Tree
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Search:
Case sensitive
Regular exp.
Search
Reset
Text
HTTP Request_rest
Sampler result
Request
Response data
Response Body
Response headers
Find
Case sensitive
Regular exp.
{"page":2,"per_page":6,"total":12,"total_pages":2,"data":[{"id":7,"email":"michael.lawson@reqres.in","first_name":"Michael","last_name":"Lawson","avatar":"https://reqres.in/img/faces/7-image.jpg"},{"id":8,"email":"lindsay.ferguson@reqres.in","first_name":"Lindsay","last_name":"Ferguson","avatar":"https://reqres.in/img/faces/8-image.jpg"},{"id":9,"email":"tobias.funke@reqres.in","first_name":"Tobias","last_name":"Funke","avatar":"https://reqres.in/img/faces/9-image.jpg"},{"id":10,"email":"byron.fields@reqres.in","first_name":"Byron","last_name":"Fields","avatar":"https://reqres.in/img/faces/10-image.jpg"},{"id":11,"email":"george.edwards@reqres.in","first_name":"George","last_name":"Edwards","avatar":"https://reqres.in/img/faces/11-image.jpg"},{"id":12,"email":"rachel.howell@reqres.in","first_name":"Rachel","last_name":"Howell","avatar":"https://reqres.in/img/faces/12-image.jpg"}],"support":{"url":"https://reqres.in/#support-heading","text":"To keep ReqRes free, contributions towards server costs are appreciated!"}}