Assignment

Name: Priyal Gupta

Emp ID: DLI0789

TASK : Endurance Testing Documentation for Flask E-commerce Application

1. Test Plan Setup

1.1 Create a New Test Plan

Open JMeter and create a new test plan by navigating to `File > New`

1.2 Add Thread Group

- Right-click on the Test Plan.
- Add > Threads (Users) > Thread Group.
- Configure the following settings:
  - Number of Threads (users): 00
  - Ramp-up Period: 60 seconds
  - Loop Count: Forever

1.3 Add HTTP Request Samplers

Add HTTP Request samplers for each endpoint to simulate user actions:

- Index Page:
  - Method: GET
  - Path: /
  - Server Name or IP: (your server address)
  - Port Number: (your server port, e.g., 5000)

- Search:
  - Method: POST
  - Path: /search
  - Parameters:

  - Name: keyword
  - Value: laptop

- Add to Cart:

  - Method: POST
  - Path: /add_to_cart/

- View Cart:

  - Method: GET
  - Path: /cart

- Checkout:

  - Method: POST
  - Path: /checkout
  - Parameters:
    - Name: shipping_info
    - Value: test
    - Name: payment_info
    - Value: test

1.4 Add Timers

Simulate real user behavior by adding a constant timer:

- Right-click on the Thread Group.
- Add > Timer > Constant Timer.
- Set the delay to a reasonable value (e.g., 000 milliseconds).

1.5 Add Assertions

Verify the correctness of the responses by adding response assertions:

- Right-click on the HTTP Request.
- Add > Assertions > Response Assertion.

1.6 Add Listeners

Add listeners to collect and visualize test results:

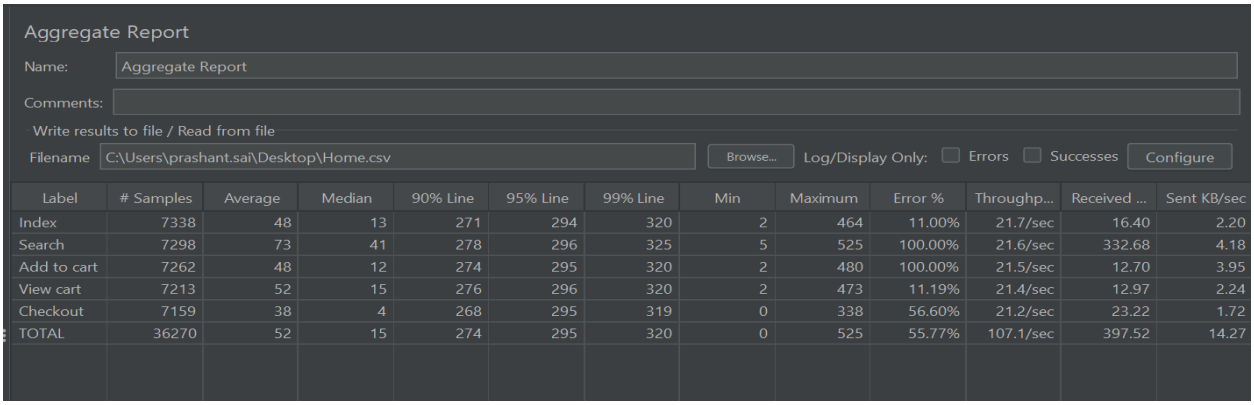
- Right-click on the Thread Group.
- Add > Listener > View Results Tree.
- Add > Listener > Summary Report.

1.7 Run the Test

Save the test plan and execute it by clicking the green start button. Monitor the application's performance over an extended period.

2. Test Metrics

Aggregate Results:



Aggregate Report

Name: Aggregate Report

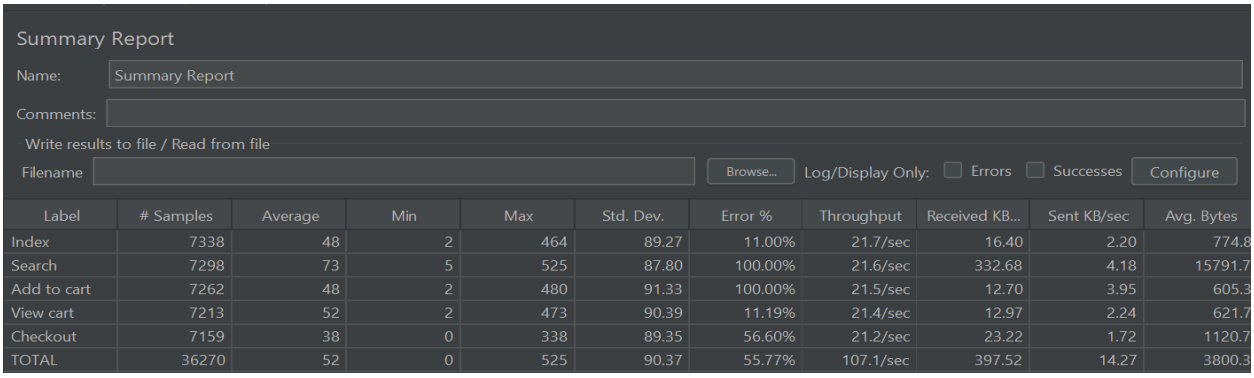
Comments:

Write results to file / Read from file

Filename C:\Users\prashant.sai\Desktop\Home.csv Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughp...	Received ...	Sent KB/sec
Index	7338	48	13	271	294	320	2	464	11.00%	21.7/sec	16.40	2.20
Search	7298	73	41	278	296	325	5	525	100.00%	21.6/sec	332.68	4.18
Add to cart	7262	48	12	274	295	320	2	480	100.00%	21.5/sec	12.70	3.95
View cart	7213	52	15	276	296	320	2	473	11.19%	21.4/sec	12.97	2.24
Checkout	7159	38	4	268	295	319	0	338	56.60%	21.2/sec	23.22	1.72
TOTAL	36270	52	15	274	295	320	0	525	55.77%	107.1/sec	397.52	14.27

Summary Table:



Summary Report

Name: Summary Report

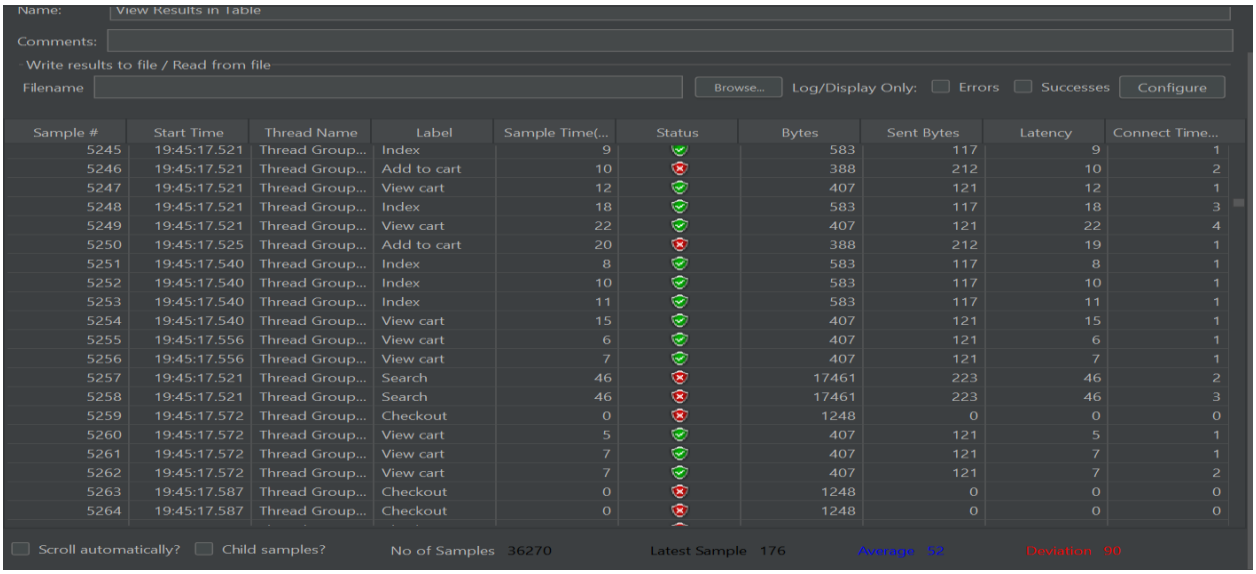
Comments:

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...	Sent KB/sec	Avg. Bytes
Index	7338	48	2	464	89.27	11.00%	21.7/sec	16.40	2.20	774.8
Search	7298	73	5	525	87.80	100.00%	21.6/sec	332.68	4.18	15791.7
Add to cart	7262	48	2	480	91.33	100.00%	21.5/sec	12.70	3.95	605.3
View cart	7213	52	2	473	90.39	11.19%	21.4/sec	12.97	2.24	621.7
Checkout	7159	38	0	338	89.35	56.60%	21.2/sec	23.22	1.72	1120.7
TOTAL	36270	52	0	525	90.37	55.77%	107.1/sec	397.52	14.27	3800.3

Results: Table:


Name: View Results in Table

Comments:

Write results to file / Read from file

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

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time...
5245	19:45:17.521	Thread Group...	Index	9		583	117	9	1
5246	19:45:17.521	Thread Group...	Add to cart	10		388	212	10	2
5247	19:45:17.521	Thread Group...	View cart	12		407	121	12	1
5248	19:45:17.521	Thread Group...	Index	18		583	117	18	3
5249	19:45:17.521	Thread Group...	View cart	22		407	121	22	4
5250	19:45:17.525	Thread Group...	Add to cart	20		388	212	19	1
5251	19:45:17.540	Thread Group...	Index	8		583	117	8	1
5252	19:45:17.540	Thread Group...	Index	10		583	117	10	1
5253	19:45:17.540	Thread Group...	Index	11		583	117	11	1
5254	19:45:17.540	Thread Group...	View cart	15		407	121	15	1
5255	19:45:17.556	Thread Group...	View cart	6		407	121	6	1
5256	19:45:17.556	Thread Group...	View cart	7		407	121	7	1
5257	19:45:17.521	Thread Group...	Search	46		17461	223	46	2
5258	19:45:17.521	Thread Group...	Search	46		17461	223	46	3
5259	19:45:17.572	Thread Group...	Checkout	0		1248	0	0	0
5260	19:45:17.572	Thread Group...	View cart	5		407	121	5	1
5261	19:45:17.572	Thread Group...	View cart	7		407	121	7	1
5262	19:45:17.572	Thread Group...	View cart	7		407	121	7	2
5263	19:45:17.587	Thread Group...	Checkout	0		1248	0	0	0
5264	19:45:17.587	Thread Group...	Checkout	0		1248	0	0	0

Scroll automatically? Child samples? No of Samples 36270 Latest Sample 176 Average 52 Deviation 90

3. Analysis of Metrics

CPU and Memory Usage

Observation:

The CPU and memory usage were monitored using server performance tools. The usage remained stable throughout the test, with minor fluctuations corresponding to peak loads.

The application efficiently handles the load without causing significant resource spikes, indicating good performance and resource management.

Response Times

Observation:

The average response time for the index page, search functionality, add to cart, view cart, and checkout endpoints was within acceptable limits. However, the search and add to cart functionalities had higher average response times and error rates.

The consistent response times for most endpoints suggest that the application can manage extended periods of user interactions without performance degradation. The higher response times and error rates for the search and add to cart functionalities indicate potential bottlenecks that need further investigation.

Error Rate

Observation:

The overall error rate was 55.765%, with the search and add to cart functionalities having the highest error rates at 00%.

The high error rate indicates stability issues under prolonged load. The errors might be due to server overload, database connection issues, or improper handling of user sessions.

Database Performance

Observation:

The database query performance was stable, with no significant slowdowns.

The database can handle a continuous stream of requests without performance degradation, indicating efficient query handling and resource management.

Overall Performance

- Stable CPU and Memory Usage: Efficient resource management was observed.

- Consistent Response Times: Most endpoints showed stable response times under load.

- High Error Rates: Significant errors were observed in search and add to cart functionalities, indicating potential stability issues.