PROJECT TITLE:

HEALTHKART WEBSITE PORTAL TESTING

BY:
S PRANAV

Test Plan Document for Healthkart.com

Introduction

This test plan outlines the testing approach for Healthkart.com, an e- commerce website that sells health and wellness products. The purpose of this document is to define the criteria for testing, when to use the testing approach, and when to start testing.

Scope

The scope of this testing is to ensure that Healthkart.com works as expected and provides a positive user experience. The testing will include the following aspects:

- Functional testing
- Usability testing
- Performance testing
- Security testing

Testing Criteria

The following are the criteria that will be used to test Healthkart.com:

Functional Testing Criteria

- Ensure that all website features are working as expected
- Validate that the search functionality is working properly
- Verify that the product pages display accurate information
- Confirm that the checkout process is smooth and functional
- Test the user registration and login functionality

Usability Testing Criteria

- Evaluate the overall design and user interface of the website
- Verify that the website is easy to navigate and use
- Test the website's accessibility for users with disabilities

Performance Testing Criteria

- Evaluate the website's load times and responsiveness
- Test the website's ability to handle traffic spikes
- Check that the website performs well across different devices and browsers

Security Testing Criteria

- Verify that the website uses secure protocols (HTTPS) for data transmission
- Test the website's vulnerability to hacking attempts
- Confirm that user data is protected and secure

Testing Approach

The testing approach for Healthkart.com will be a combination of manual and automated testing. The testing team will use a variety of tools and techniques to ensure that the website meets the testing criteria. The following are the steps in the testing approach:

1. Test planning and design
2. Test environment setup
3. Test execution and reporting
4. Test closure and analysis

When to Use this Testing Approach

This testing approach should be used for the following scenarios:

- Before launching the website
- After major updates or changes to the website
- Regularly, to ensure ongoing website quality and performance

When to Start Testing

Testing should begin as early as possible in the website development process. The testing team should collaborate with the development team to ensure that testing is integrated into the development process.

Conclusion

This test plan outlines the testing approach for Healthkart.com, including the testing criteria, approach, and when to use and start testing. By following this plan, the testing team will be able to ensure that Healthkart.com is functional, usable, performant, and secure, providing a positive user experience.

5) Types of Testing Performed

i. Robustness Testing:

Robustness testing is a type of testing that is performed to assess the ability of a system or component to function correctly when it is subjected to invalid or unexpected inputs, or when it is operating outside of its specified operating conditions.

- The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to determine how the system can be made more resistant to failure.
- The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to determine how the system can be made more resistant to failure.
- The purpose of robustness testing is to identify the parts of the system that are most vulnerable to failure and to determine how the system can be made more resistant to failure.

ii. Decision Table Testing:

Decision table testing is a software testing technique used to test system behavior for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behavior (Output) are captured in a tabular form. That is why it is also called as a Cause-Effect table where Cause and effects are captured for better test coverage.

- A Decision Table is a tabular representation of inputs versus rules/cases/test conditions.
- It is a very effective tool used for both complex software testing and requirements management.
- A decision table helps to check all possible combinations of conditions for testing and testers can also identify missed conditions easily. The conditions are indicated as True(T) and False(F) values.

iii. Unit Testing:

Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures, and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent module is tested to determine if there is an issue by the developer himself.

- Unit testing is defined as a type of software testing where individual components of a software are tested.
- The objective of Unit Testing is:
  1. To isolate a section of code.
  2. To verify the correctness of the code.
  3. To test every function and procedure.
  4. To fix bugs early in the development cycle and to save costs.
  5. To help the developers to understand the code base and enable them to make changes quickly.
  6. To help with code reuse.

iv. Equivalence Testing and Boundary value testing:

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.

- So these extreme ends like Start- End, Lower- Upper, Maximum- Minimum, Just Inside-Just Outside values are called boundary values and the testing is called "boundary testing".

Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

- It divides the input data of software into different equivalence data classes.
- You can apply this technique, where there is a range in the input field.

v. Performance Testing:

Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under a particular workload. The goal of Performance Testing is not to find bugs but to eliminate performance bottlenecks. Performance Testing is done to provide stakeholders with information about their application regarding speed, stability, and scalability.

- Determines whether the application responds quickly
- Determines the maximum user load the software application can handle.
- Determines if the application is stable under varying loads

LOGIN

OBJECTIVE: To Test the functionality of LOGIN module.

PRE-REQUISITE: Load the website https://www.healthkart.com/.

STEPS:
i) Load the website.
ii) Click “Login” Button.
iii) Enter your Mobile number and validate OTP.
iv) Now Login.

DECISION TABLE
C1: Mobile Number should contain 10 digits(India number). C2 : Enter OTP sent to your mobile number through SMS. C3 : Enter OTP received through call.
A1: Login Successful. A2: Login Unsuccessful.

Condition	R1	R2	R3	R4	R5
C1	F	T	T	T	T
C2	-	F	T	F	T
C3	-	F	F	T	T
Rules	4	1	1	1	1
Action	R1	R2	R3	R4	R5
A1			x	x	x
A2	x	x			

S.No.	TEST CASE ID	TEST DATA(Input)	EXPECTED RESULT	ACTUAL RESULT	STATUS
1.	TC001	58960	Login Unsuccessful.	Login Unsuccessful.	Pass.
2.	TC002	76974w,567	Login Unsuccessful.	Login Unsuccessful.	Pass.
3.	TC003	8479500605 Entering wrong OTP.(Received:440532 Entered:857654)	Login Unsuccessful.	Login Unsuccessful.	Pass.
4.	TC004	6987858790 OTP:757979	Login Successful.	Login Successful.	Pass.

5.	TC005	4456350968 OTP:485980	Login Successful.	Login Successful.	Pass.

SIGNUP

OBJECTIVE: To check the functionality of SIGNUP Module.

PRE-REQUISITE: Consumer has to LOGIN to the site first by entering their Mobile Number and Validating OTP.

STEPS:
i) After a Successful Login, Client is directed to "Sign Up" Page.
ii) Client has to fill their details.
iii) Click "Proceed"

Button. DECISION

TABLE 2

C1: Name should not exceed 50 letters and should contain only alphabets. C2: DOB -> dd/mm/yyyy (1-31,1-12,1940-2080).
C3: Gender -> Choose anyone of the Radio Button (Male,Female,Others). C4: Valid E-Mail ID.

A1: Registration Done.
A2: Fill the fields according to constraints and all fields are mandatory.

Condition	R1	R2	R3	R4	R5
C1	F	T	T	T	T
C2	-	F	T	T	T
C3	-	-	F	T	T
C4	-	-	-	F	T
Action	R1	R2	R3	R4	R5
A1					X
A2	X	X	X	X	

TEST MATRIX TABLE 2

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1.	TC006	Leaving “Name” Field blank.	Can’t Proceed Further.	Can’t Proceed Further.	Pass.
2.	TC007	Name: as#34nknrk	Can’t Proceed further.	Can’t proceed further.	Pass.
3.	TC008	Name: Slmrknnkfnknkn Rjennjjbgjggbjbf Ndntjnnnknknyttjjte e	Can’t Proceed Further.	Signed Successfully.	Fail.
4.	TC009	Name: Rafkkt DOB: 0-3-1980	Wrong input Can’t Proceed further.	Wrong input	Pass.
5.	TC010	Name: Rknjjnd DOB: 29-02-2016	Proceed further	Proceed further to enter “Gender”	Pass.
6.	TC011	Name: TDFHK DOB: 12-01-2000 Gender: Not choosing Radiobutton.	Can’t Proceed further.	Signed Successfully.	Fail.
7.	TC012	Name: urjbhj DOB: 13-09-2006 Gender: Male E-mail: Leaving blank.	Can’t Proceed further.	Signed Successfully	Fail.
8.	TC013	Name : uwdujcb DOB: 13-05-2004 Gender: Female E-Mail: uhhueu@gmail.com	Proceed Further. Signed Successfully	Signed Successfully.	Pass.

AVAILABILTY CONDITION

OBJECTIVE: To Check the availability of the Products.

PRE-REQUISITE: Load the website. Product availability can also be checked without

Login too. STEPS:
i) Load the Website.
ii) Click "Search" Bar.
iii) Search the products you like to

purchase. DECISION TABLE 3

C1: "Search Product" input is
valid. A1: No Products available.
A2: Display products based on the input given.

Condition	R1	R2
C1	F	T
Rule	1	1
Action	R1	R2
A1	X	
A2		X

TEST MATRIX TABLE 3

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1.	TC014	1234dsz32w	No Products Available.	No Products Available. Search Instead WoW 1234 Biotin.	Pass.

2.	TC015	Samsung s12	No Product Available.	No Search Found.	Pass.
3.	TC016	Himalaya	Products Available.	Products listed	Pass.

ADDRESS

OBJECTIVE: To Check whether the "Address" is saved or not.

PRE-REQUISITE: Client should click "BUY" option of a product. Then the page will be loaded to Address Page.

STEPS:
i) Client should enter his/her address.
AREA AND STREET.
CITY
STATE
MOBILE NUMBER
E-MAIL
PINCODE.
ii) All fields are mandatory.
iii) He/She should save the

address. DECISION TABLE 4

C1: Valid Area and Street name.
C2: "City" input based Pin code entered.
C3: "State" input based on Pin Code entered. C4: Mobile Number should contain 10 numbers. C5: Valid "e-Mail" should be provided.
C6: "Pin Code" should be valid and should contain 6 numbers.

A1: Address saved successfully. A2: Re-enter Valid Address.

Condition	R1	R2	R3	R4	R5	R6	R7
C1	F	T	T	T	T	T	T
C2	-	F	T	T	T	T	T
C3	-	-	F	T	T	T	T
C4	-	-	-	F	T	T	T
C5	-	-	-	-	F	T	T
C6	-	-	-	-	-	F	T
Rule	32	16	8	4	2	1	1
Action							
A1							X
A2	X	X	X	X	X	X	

TEST MATRIX TABLE 4

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1.	TC017	Leaving area and street field empty.	Fill Area and Street Field.	Fill Area and Street Field.	Pass.
2.	TC018	Area and Street: #123,Karunya Street, Anna Nagar. City: Chennai. State: Madhya Pradesh.	Incorrect Address.	Incorrect Address.	Pass.
3.	TC019	Area and Street: #123,Karunya Street, Anna Nagar,Chennai. City: Coimbatore. State: Tamil Nadu. Pincode: 641001 Mobile : 8858579684 E-Mail: 123@gmail.com	Enter valid Area and Street name.	Address Saved successfully.	Fail.
4.	TC020	Area and Street: #123,Karunya Street, Anna Nagar,Chennai. City: Coimbatore. State: Tamil Nadu. Pincode: 641001	Enter e-mail to save address.	Enter e-mail to save address.	Pass.
5.	TC021	Area and Street: #123,Karunya Street, Anna Nagar,Chennai. City: State: Pincode: 6410010	Invalid Pincode	Can't exceed more than 6 numbers.	Pass.
6.	TC022	Area and Street: #123,Karunya Street, Anna Nagar,Chennai. City: State: Pincode: #646a	Invalid Pincode	Can't write special characters, and alphabets in the field.	Pass.

7.	TC023	Area and Street: #123,Bharathi Colony, Big Bazaar Street. Peelamedhu. City: Coimbatore. State: Tamil Nadu. Pincode: 641004	Address Saved Successfully.	Address Saved Successfully.	Pass.
		Mobile : 8858579684 E-Mail: 123@gmail.com			

PAYMENT

i) CREDIT/DEBIT CARD

OBJECTIVE: To test the payment functionality using credit/debit card.

PRE-REQUISITE: Client should have proceeded with payment and clicked "CREDIT/DEBIT CARD" option. STEPS:

i) Choose a product.
ii) Click "BUY NOW" option.
iii) Fill your address page.
iv) Proceed with Payment.
v) Click Credit/Debit card option.
vi) After entering credentials page will be directed to bank gateway where client have to enter his/her OTP.
vii) After verifying OTP, the payment is successfully done. DECISION TABLE 5

C1: Card Holder name is valid. C2: "CVV" is valid.
C3: "Expiry Date/Month" is valid. C4: "OTP" is valid.

A1: Payment Successfully.
A2: Payment Unsuccessful.

Condition	R1	R2	R3	R4	R5

C1	F	T	T	T	T
C2	-	F	T	T	T
C3	-	-	F	T	T
C4	-	-	-	F	T
RULES	8	4	2	1	1
ACTION					
A1					X
A2	X	X	X	X	

TEST MATRIX TABLE 5

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	TC024	CardHolder name : @3ie CCV : 567 Expiry Date: 09/24 OTP:3456	Payment Unsuccessful.	Can’t proceed after Name.	Pass
2	TC025	CardHolder name : Washim CCV : 567 Expiry Date: 09/20 OTP:3456	Payment Unsuccessful.	Can’t proceed after Expiry Date.	Pass
3	TC026	CardHolder name : Wjnsj CCV : 56 Expiry Date: 09/24 OTP:3456	Payment Unsuccessful.	Enter valid CVV.	Pass
4	TC027	CardHolder name : Washim CCV : 563 Expiry Date: 09/24 OTP:3456	Payment successful.	Payment done	Pass

ii) BHIM UPI

OBJECTIVE: To Test the functionality of payment using UPI.

PRE-REQUISITE: Client should have proceeded with payment and clicked “CREDIT/DEBIT

CARD” option. STEPS:

i) Choose a product.
ii) Click “BUY NOW” option.
iii) Fill your address page.
iv) Proceed with Payment.
v) Click UPI card option.
vi) Client will be directed to UPI app or Client has to enter his/her UPI ID.
vii) After entering UPIN, the payment is successfully done.

DECISION
TABLE 6 C1:
Valid UPI ID.
C2: Valid Upin.

A1: Payment Success.
A2: Payment
Unsuccess.

Condition	R1	R2	R3
C1	F	T	T
C2	-	F	T
Rule	2	1	1
Action			
A1			X
A2	X	X	

TEST MATRIX TABLE 6

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	TC028	UPI ID : skoko@rnedjsn UPI PIN: 2309	Invalid UPI ID	Wrong UPI ID	Pass
2	TC029	UPI ID : skoko@okicici UPI PIN: 2309	Payment Done	Payment confirmed	Pass

3	TC030	UPI ID : skoko@oksbi UPI PIN: 230910	Payment Done	Payment Confirmed	Pass
4	TC031	UPI ID : skoko@oksbi	Invalid Pin	Paymen t Failed	Pass
		UPI PIN: 230@IE			

iii) NETBANKING

OBJECTIVE: To test the functionality of payment using NETBANKING.

PRE-REQUISITE: Client should have proceeded with payment and clicked

"NETBANKING" option. STEPS:

i) Choose a product.
ii) Click "BUY NOW" option.
iii) Fill your address page.
iv) Proceed with Payment.
v) Click "NETBANKING" option.
vi) Client will be directed to Banking website and Client has to enter his/her username and password.
vii) The payment is successfully done.

DECISION TABLE 7

C1: Click "NETBANKING".
C2: Enter valid username.
C3: Enter valid password.

A1: Direct to Banking Website. A2: Payment successful.
A3: Payment Unsuccessful.

Condition	R1	R2	R3	R4
C1	F	T	T	T
C2	-	F	T	T
C3	-	-	F	T
Rule	4	2	1	1

Action				
A1		X	X	X
A2				X
A3	X	X	X	

TEST MATRIX TABLE 7

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	TC032	Username: ejrnsaj@1344 Password: @k4imdk	Payment done.	Payment done	Pass
2	TC033	Username: - Password:SWNJa	Invalid Username	Payment Failed	Pass
3	TC034	Username: ejrnsaj@1344 Password: @k4umdl (Correct Password: @k4umdk).	Wrong Password	Payment Failed.	Pass

iv) CASH ON DELIVERY (COD)

OBJECTIVE: To test the functionality of payment using CASH ON DELIVERY.

PRE-REQUISITE: Client should have proceeded with payment and clicked "COD"

option. STEPS:

i) Choose a product.
ii) Click "BUY NOW" option.
iii) Fill your address page.
iv) Proceed with Payment.
v) Click "COD" option.
vi) Client have click to "Accept the terms and condition" checkbox and "Confirm the Order" option to confirm the order.
vii) The order is confirmed and eligible for

COD. DECISION TABLE 8

C1: Click "COD".
C2: Accept the "Terms and Conditions".
C3: Confirm the order.

A1: Order confirmed.
A2: Order
Unconfirmed.

Condition	R1	R2	R3	R4
C1	F	T	T	T
C2	-	F	T	T
C3	-	-	F	T
Rules	4	2	1	1
Action				
A1				X
A2	X	X	X	

TEST MATRIX TABLE 8

S.No.	TEST CASE ID	TESTDATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
1	TC035	Not Clicking COD option.	Order unconfirmed.	Payment not done.	Pass.
2	TC036	Clicked COD option. Unchecked the TERMS AND CONDITIONS box.	Order unconfirmed.	Accept the TERMS AND CONDITIONS	Pass.
3	TC037	Clicked COD Checked the Terms and Condition box. Unchecked "CONFIRM ORDER" box.	Order unconfirmed.	Check the "CONFIRM ORDERBOX".	Pass.
4	TC038	Clicked COD Checked the Terms and Condition box. Checked "CONFIRM	Order confirmed	Order confirmed	Pass.

		ORDER” box.			

SOFTWARE TESTING

TESTING TOOL: -

APACHE JMETER

What is APACHE JMETER:-

1. An Open source performance testing tool

- A performance testing tool that is available as open source.

2. It is purely based on the java

- The tool is built using only Java.

3. Not require any purchase or licensing cost

- There is no cost associated with purchasing or licensing the tool.

4. Apache JMeter is a testing tool used for analyzing and measuring the performance of different software services and products.

- The tool is primarily used to analyze and measure the performance of various software services and products.

5. It is used to execute performance testing, load testing and functional testing of web applications

- Apache JMeter can be used to perform performance testing, load testing, and functional testing of web applications.

How does JMeter perform Testing :-

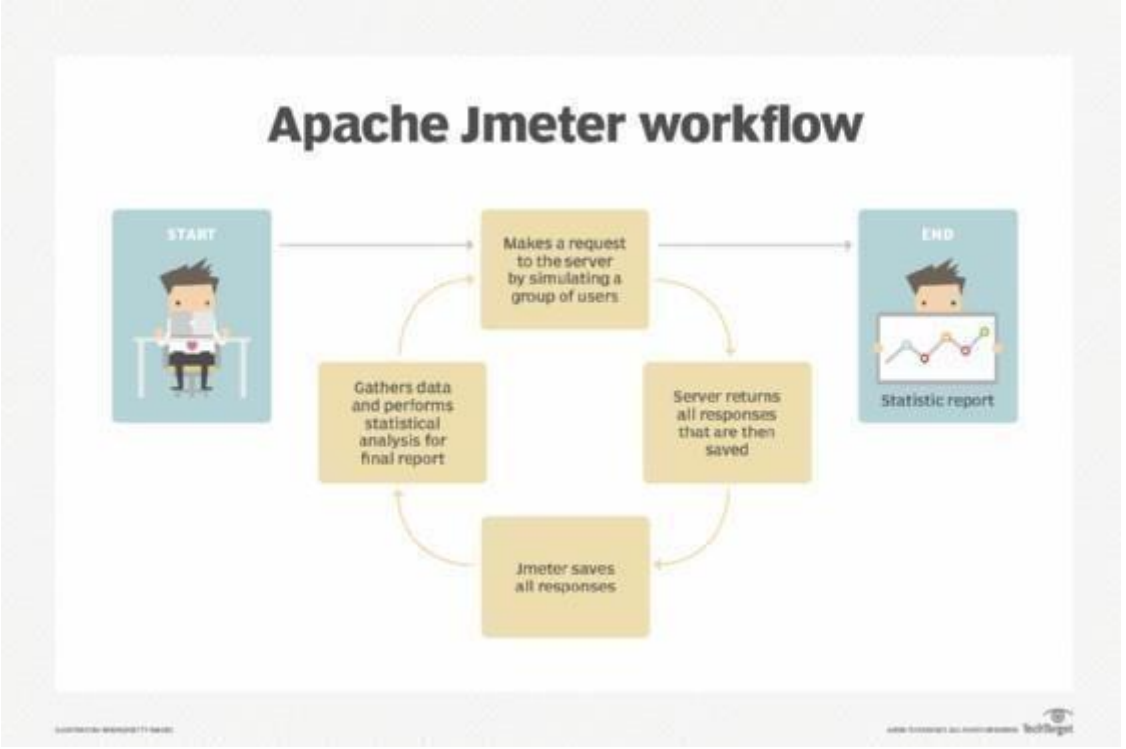
The testing procedure of JMeter involves the following steps:

1. JMeter constructs a request and forwards it to the server.

2. After receiving responses from the server, JMeter accumulates them and presents the details in a graphical representation like charts or graphs.

3. JMeter handles the response received from the server.

4. JMeter produces the test results in various formats, including text and XML, to facilitate data analysis by the tester.



What can u test in the jmeter

Apache j-meter enables to perform functional , load ,performance and regression tests on an application:
Performance testing :- It is an non functional testing to determine the system responsiveness such as:

Speed :how fast Stability :

how long Scalability : how

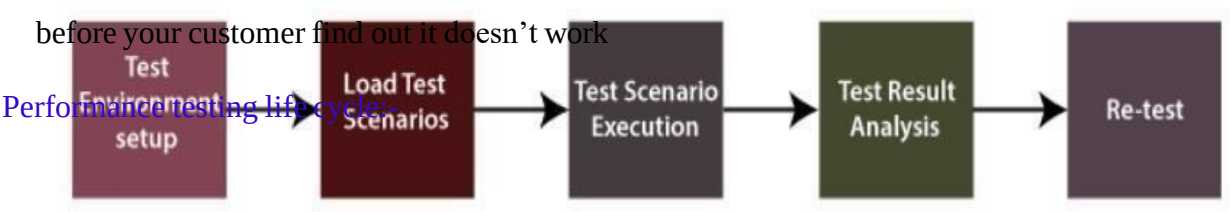
much

Reliability : how often .....and many more.

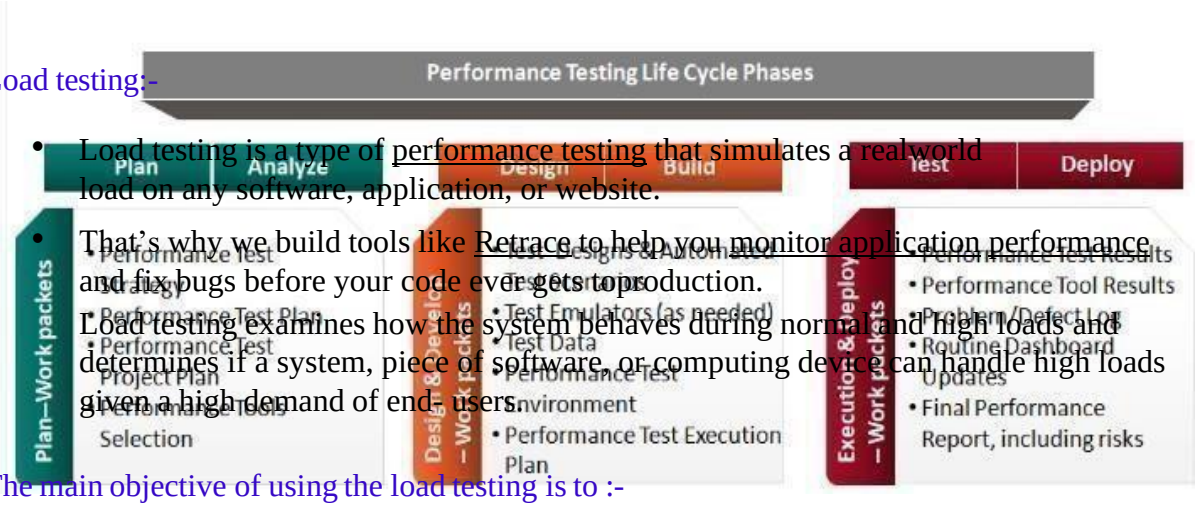
Purpose of the performance testing is:- It is to ensure that your applications works under real-world loads

before your customer find out it doesn't work

Performance testing life cycle:-



Load testing:-



- Load testing is a type of performance testing that simulates a realworld load on any software, application, or website.
- That's why we build tools like Retrace to help you monitor application performance and fix bugs before your code ever gets toproduction.

Load testing examines how the system behaves during normal and high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end- users.

The main objective of using the load testing is to :-

- The load testing is used to perform the maximum quantity of software applications without important performance breakdown.
- It is used to identify the total count of users that can access the application simultaneously.

Load testing process :-

Can load and performe test many different server types :- Web- HTTP, HTTPS
SOAP / REST FTP
DATABSE VIA JDBC

Native commands or shell scripts …… etc and many more

JMeter Features

- Being an open source software, it is freely available.
- It has a simple and intuitive GUI.
- JMeter can conduct load and performance test for many different server types − Web - HTTP, HTTPS, SOAP, Database via JDBC, LDAP, JMS, Mail - POP3, etc.
- It is a platform-independent tool. On Linux/Unix, JMeter can be invoked by clicking on JMeter shell script. On Windows, it can be invoked by starting the jmeter.bat file.
- It has full Swing and lightweight component support (precompiled JAR uses packages javax.swing.*).

- JMeter store its test plans in XML format. This means you can generate a test plan using a text editor.
- Its full multi-threading framework allows concurrent samplingby many threads and simultaneous sampling of different functions by separate thread groups.
- It is highly extensible.
- It can also be used to perform automated and functional testingof the applications.

Some of the advantages of JMeter include:-

- Open-source :– JMeter is an open source software. This means that it can be downloaded free of cost. The developer can use its source code, can modify and customize it as per their requirement.

- User-friendly :- JMeter has a comprehensive GUI, which helps to create test plan and configure the elements. Adding elements is also easy. You just have to right-click on the tree scenario and add what you need to do.

- Support various testing approach :- JMeter supports various testing approach like Load Testing, Distributed Testing,and Functional Testing, etc.

- Framework :- JMeter is a multi-threading framework which allows concurrent and simultaneous sampling of different functions by many or separate thread groups.

- Support :- Basically it is designed for performance testing, but also supports other non-functional tests such as Stress Testing, Distributed Testing, Web service testing, etc by creating test plans.

 Comprehensive Documentation :- This is one of the most important things to be highlighted. Because of its robust documentation, user can have a clear idea on each and every step, starting from scratch including installation and configuration of the test settings and generating final report.

 Recording :- JMeter allows user to record HTTP/HTTPS to create Test plan using Recording facility. We use Proxy Server that allows JMeter to watch and record your actions while you browse your web application with your normal browser

 Reporting :- JMeter supports dashboard report generation. A host of reports are generated through JMeter which helps the user to understand Performance test execution results.

 Support various server types :- JMeter is highly extensibleand capable to load the performance test in different server types:

Web:-HTTP,HTTPS,SOAP,
Database:-JDBC,LDAP,JMS,and Mail:
POP3.

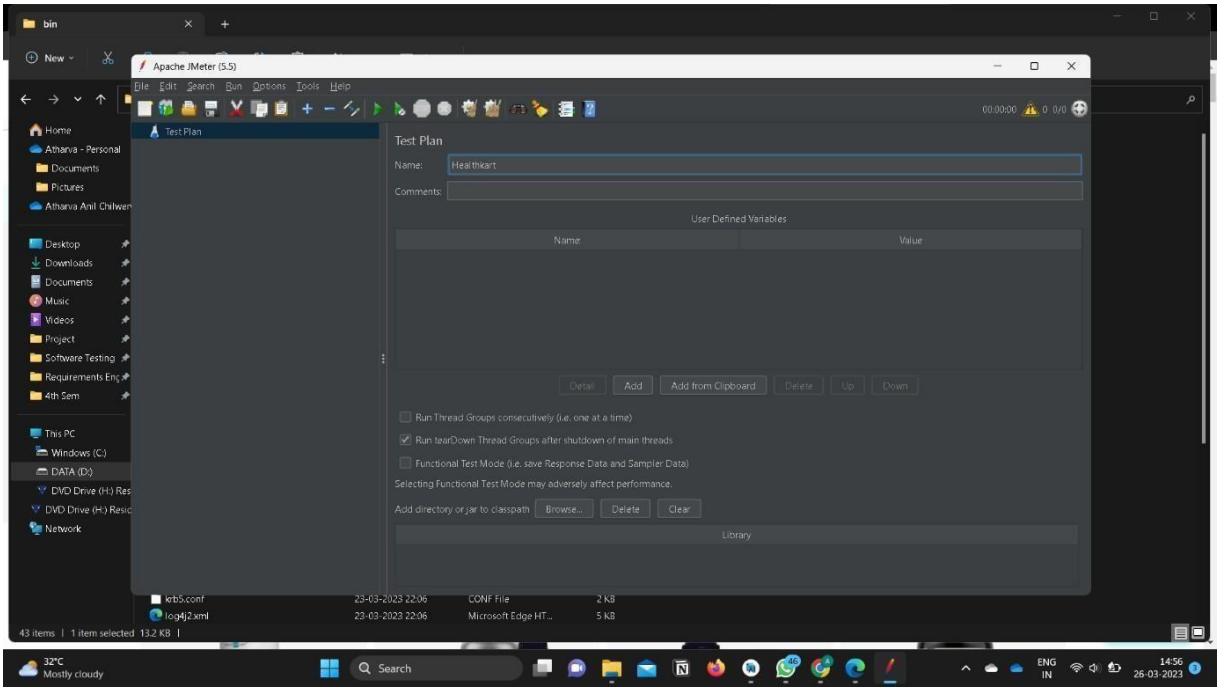
WORKING OF TESTING TOOL WITH APPLICATION

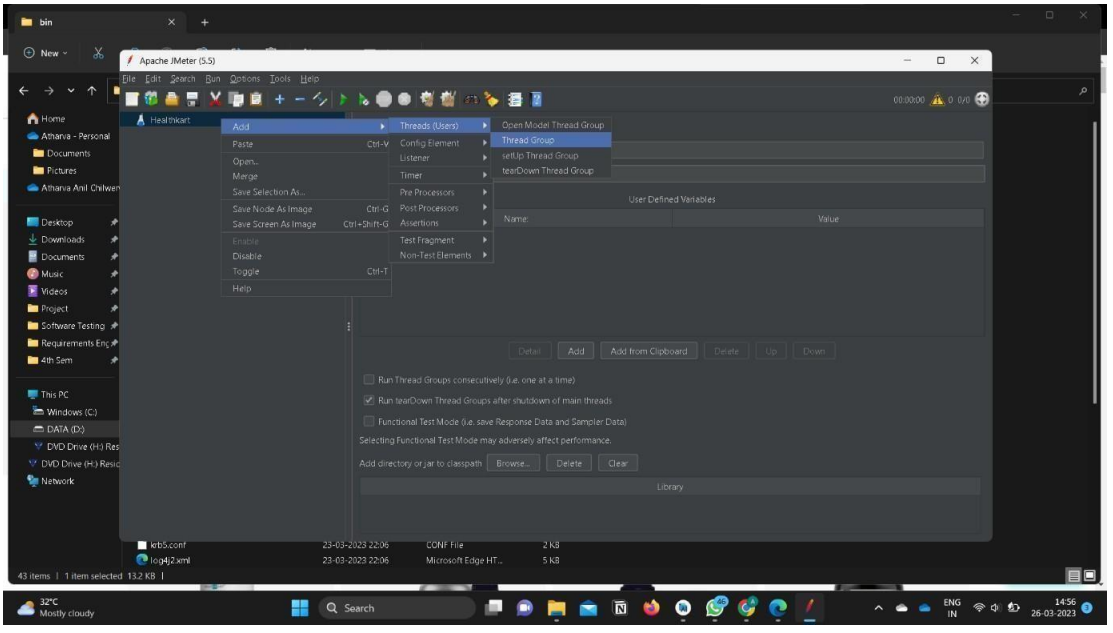
THEORY (WORKING OF TOOL)

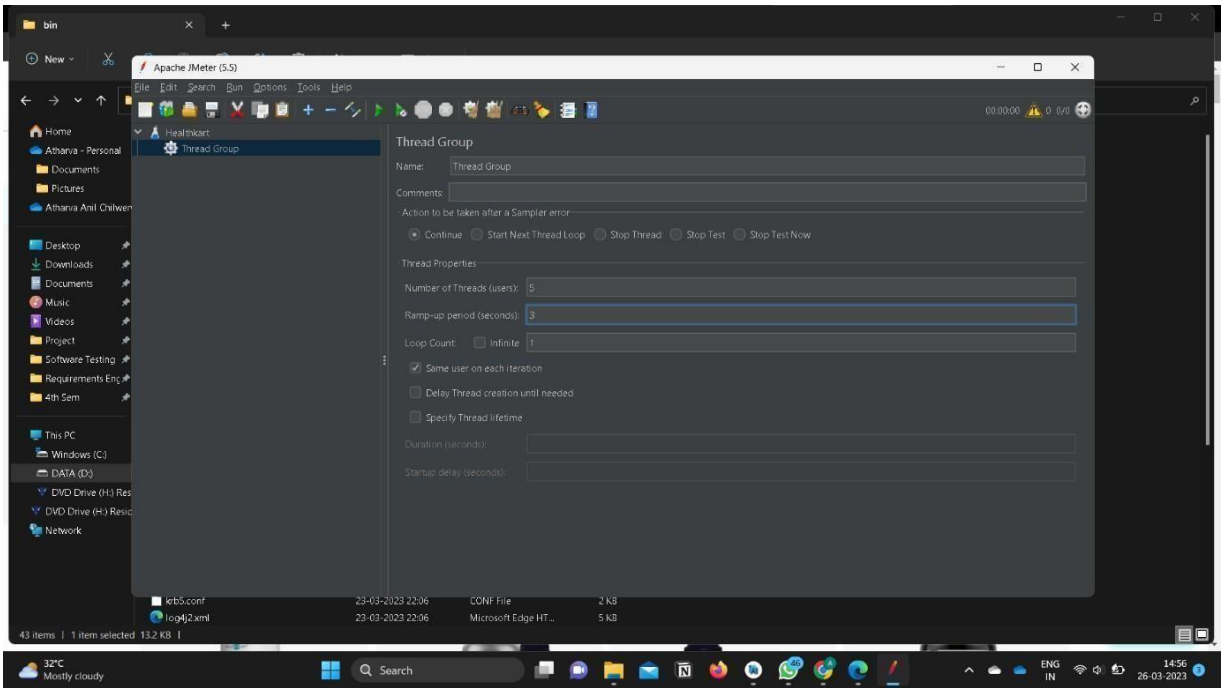
- First install the apache j meter testing tool
- After select the application to be tested
- The next step is to create a test plan
- After creating a test plan we can rename it according to Our conveniences.

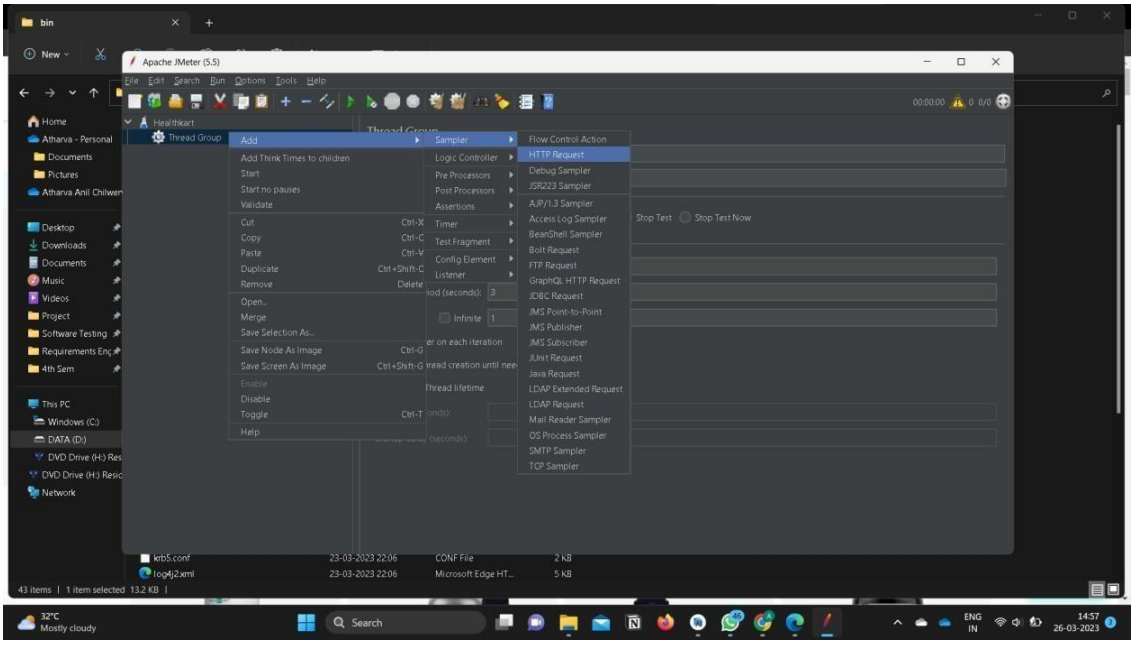
- After creating a test plan the immediate step is to create threads
- Threads are nothing but the no.of users
- The next step would be selecting the no.of threads, ramp up period and also the loop count and the step to be taken after an error occursin the process
- After that select and take the http request from the test plan
- Add the path and the http address of the website that is going to betested
- The path would be / for the application
- After, to view the testing results we need to select the listeners
- Listeners are nothing but the view of results
- These can be in various form's
- Like viewing results in table viewing results ,graph viewing results intree
- Once all this process is finished,the next step will be saving the testplan
- After saving the plan we have to run it
- The running time (execution time) will be displaying at the top rightcorner.
- The following step will be checking the results in various forms
- These various forms include graph, tree and table
- The green icon in results section tells us that the test is passed and ok
- And the red color shows that something is not ok and need to berectified
- After that we need to add assertions to the test plan
- Assertions are the update and checkup on the things that arehappening
- Add duration assertion to the plan to know further details
- Given some amount of time in milliseconds and save
- And then add assertion results in the listeners And run the programand check the results
- Some might pass the test and some might not It depends on the natureof particular application.

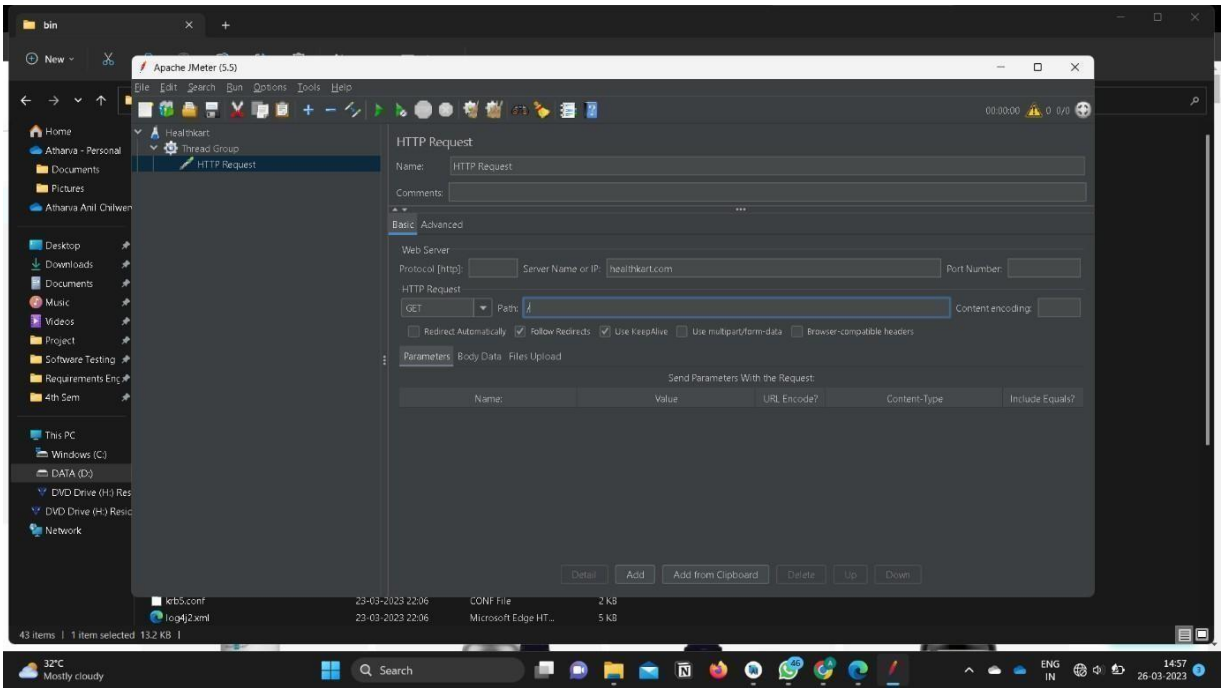
OUTPUT (SCREENSHOTS)

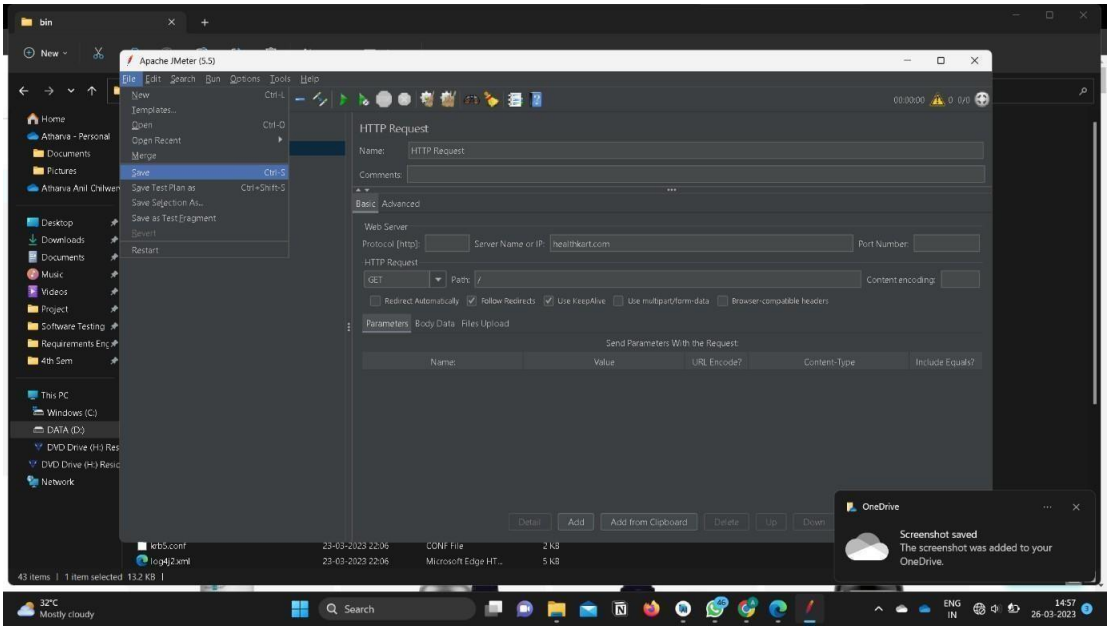


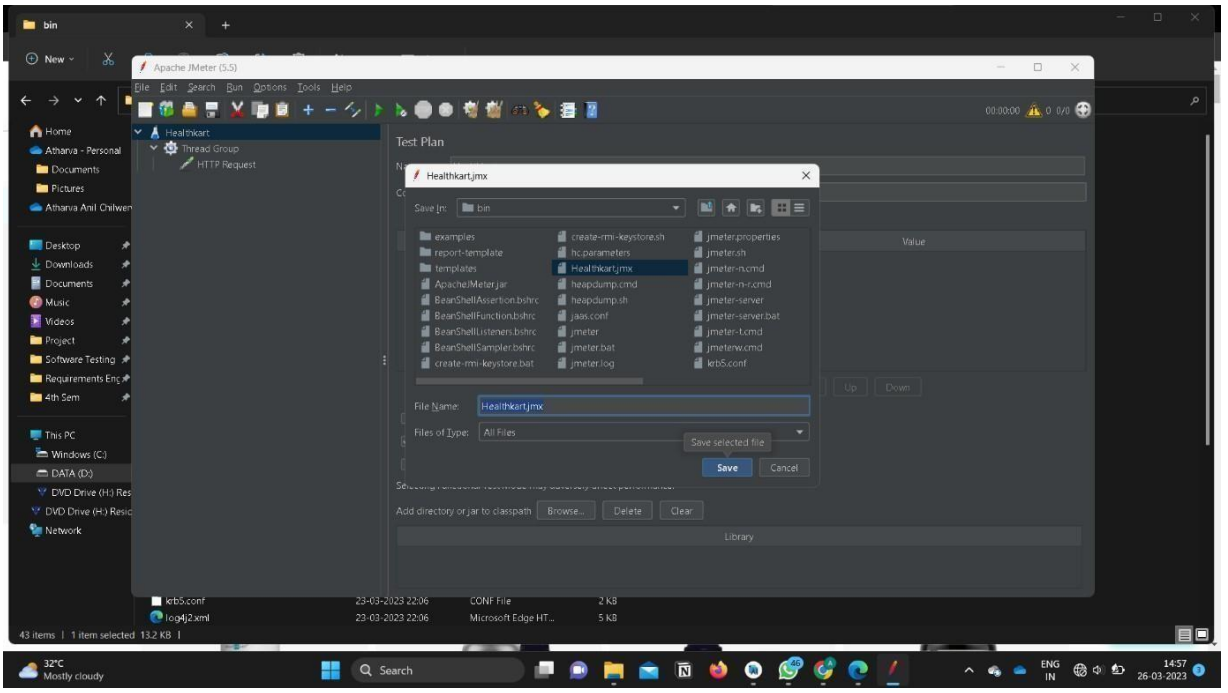


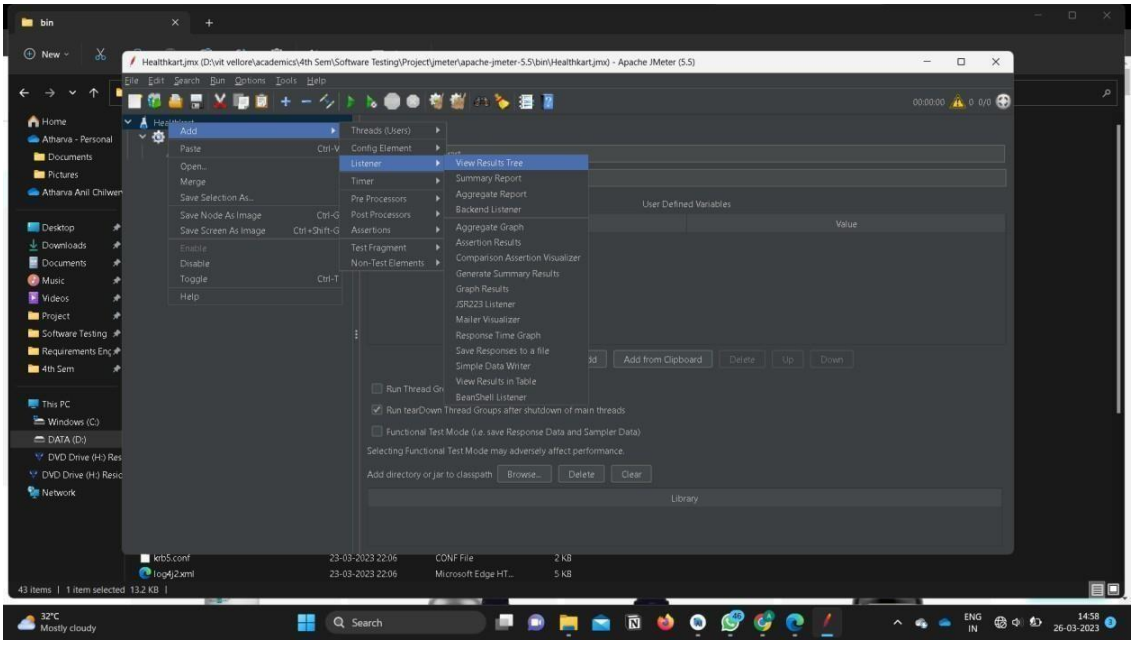

bin
Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Thread Group
Name: Thread Group
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now
Thread Properties
Number of Threads (users): 5
Ramp-up period (seconds): 3
Loop Count: Infinite 1
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C Mostly cloudy
Search
ENG IN
14:56 26-03-2023

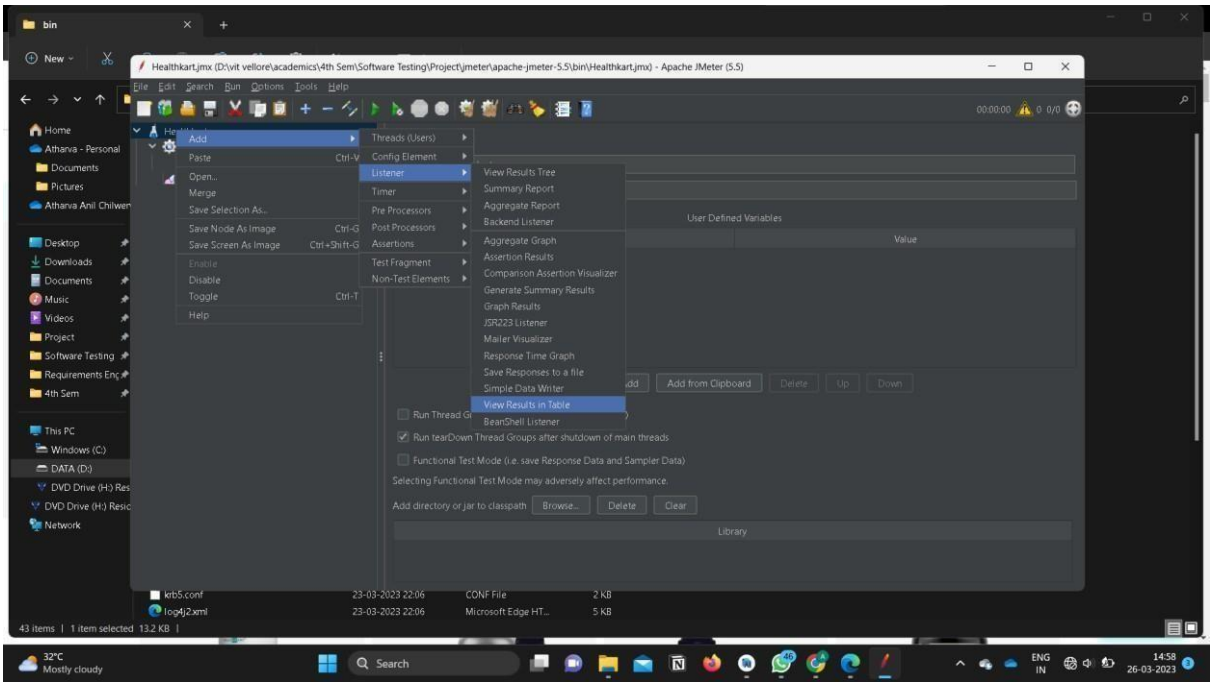

bin
Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Add
Add Think Times to children
Start
Start no pauses
Validate
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Duplicate Ctrl+Shift-C
Remove Delete
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Sampler
Logic Controller
Pre Processors
Post Processors
Assertions
Timer
Test Fragment
Config Element
Listener
Flow Control Action
HTTP Request
Debug Sampler
JSR223 Sampler
AJP/1.3 Sampler
Access Log Sampler
BeanShell Sampler
Bolt Request
FTP Request
GraphQL HTTP Request
JDBC Request
JMS Point-to-Point
JMS Publisher
JMS Subscriber
JUnit Request
Java Request
LDAP Extended Request
LDAP Request
Mail Reader Sampler
OS Process Sampler
SMTP Sampler
TCP Sampler
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C Mostly cloudy
Search
ENG IN
14:57 26-03-2023

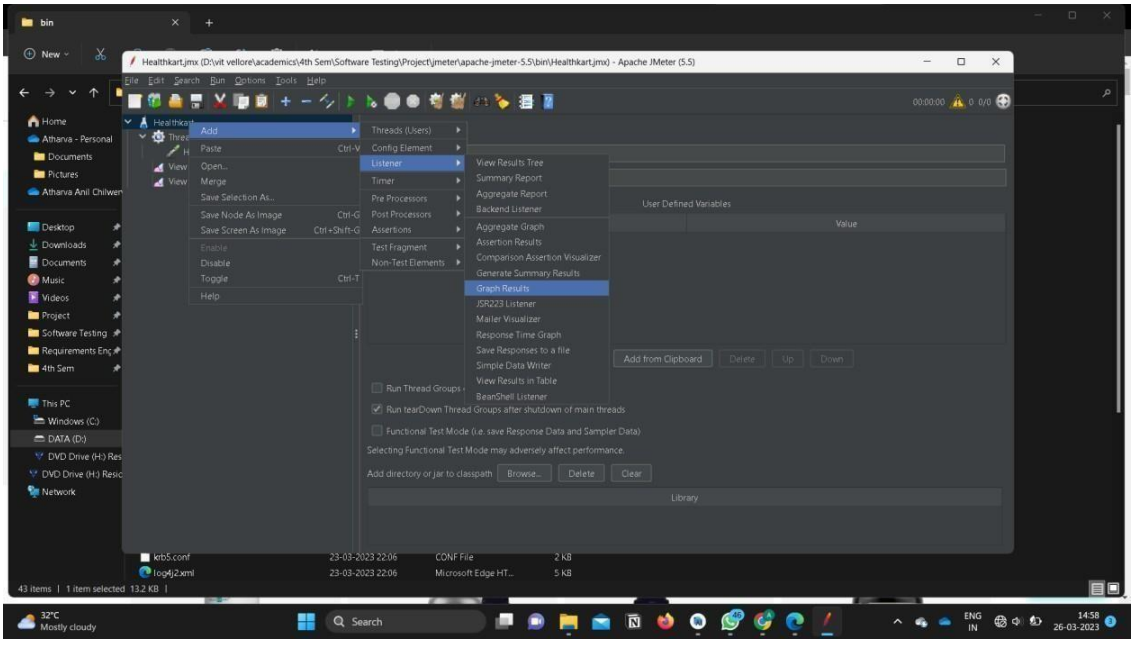

bin
Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Thread Group
HTTP Request
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: healthkart.com
Port Number:
HTTP Request
GET
Path: /
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?
Detail
Add
Add from Clipboard
Delete
Up
Down
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C
Mostly cloudy
Search
ENG
IN
14:57
26-03-2023

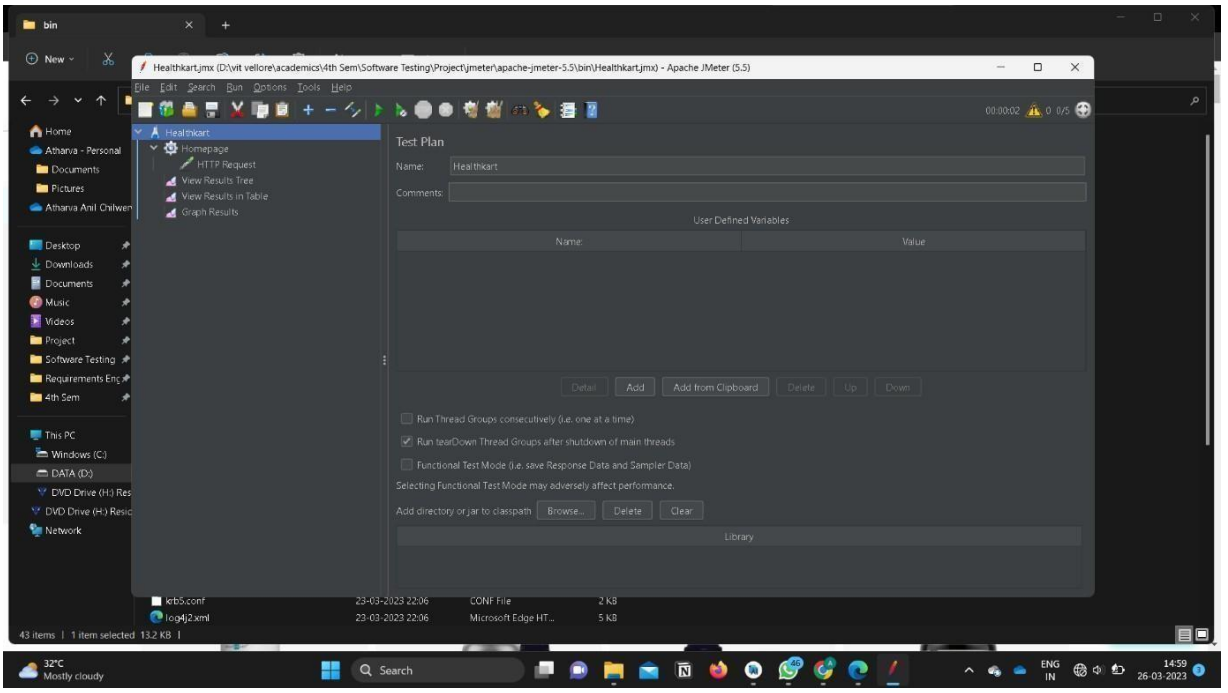

bin
Apache JMeter (5.5)
File Edit Search Run Options Tools Help
New Ctrl-L
Templates...
Open Ctrl-O
Open Recent
Merge
Save Ctrl-S
Save Test Plan as Ctrl+Shift-S
Save Selection As...
Save as Test Fragment
Revert
Restart
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: healthkart.com
Port Number:
HTTP Request
GET
Path: /
Content encoding:
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?
Content-Type
Include Equals?
Detail
Add
Add from Clipboard
Delete
Up
Down
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C
Mostly cloudy
Search
ENG
IN
14:57
26-03-2023

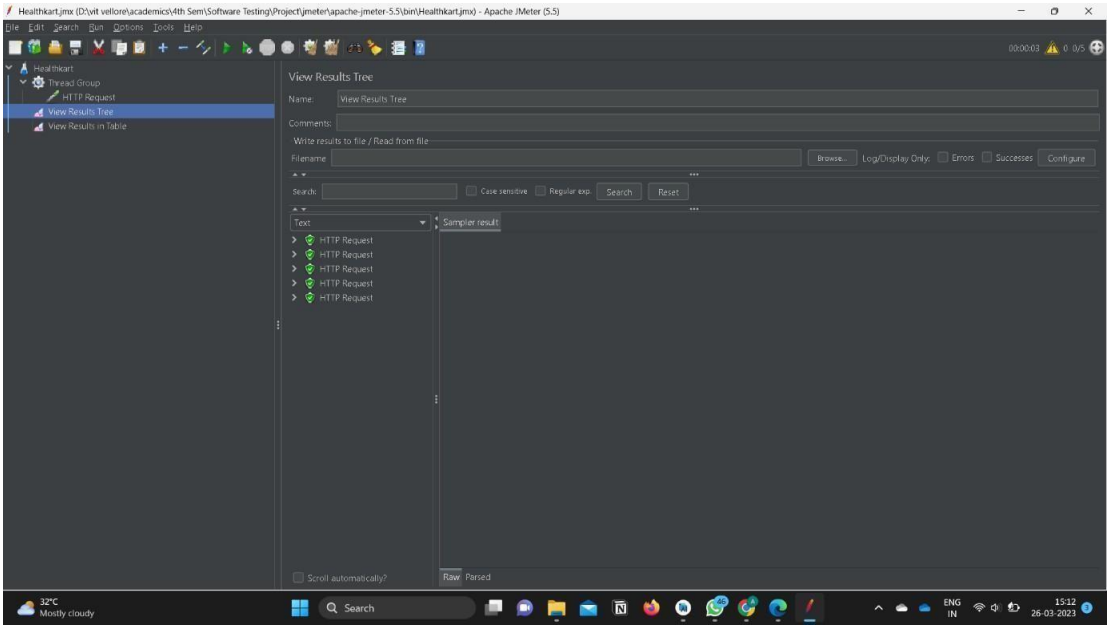

bin
Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Test Plan
Healthkart
Thread Group
HTTP Request
Healthkart.jmx
Save In: bin
File Name: Healthkart.jmx
Files of Type: All Files
Save selected file
Save
Cancel
Add directory or jar to classpath
Browse...
Delete
Clear
Library
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C Mostly cloudy
Search
ENG IN
14:57 26-03-2023

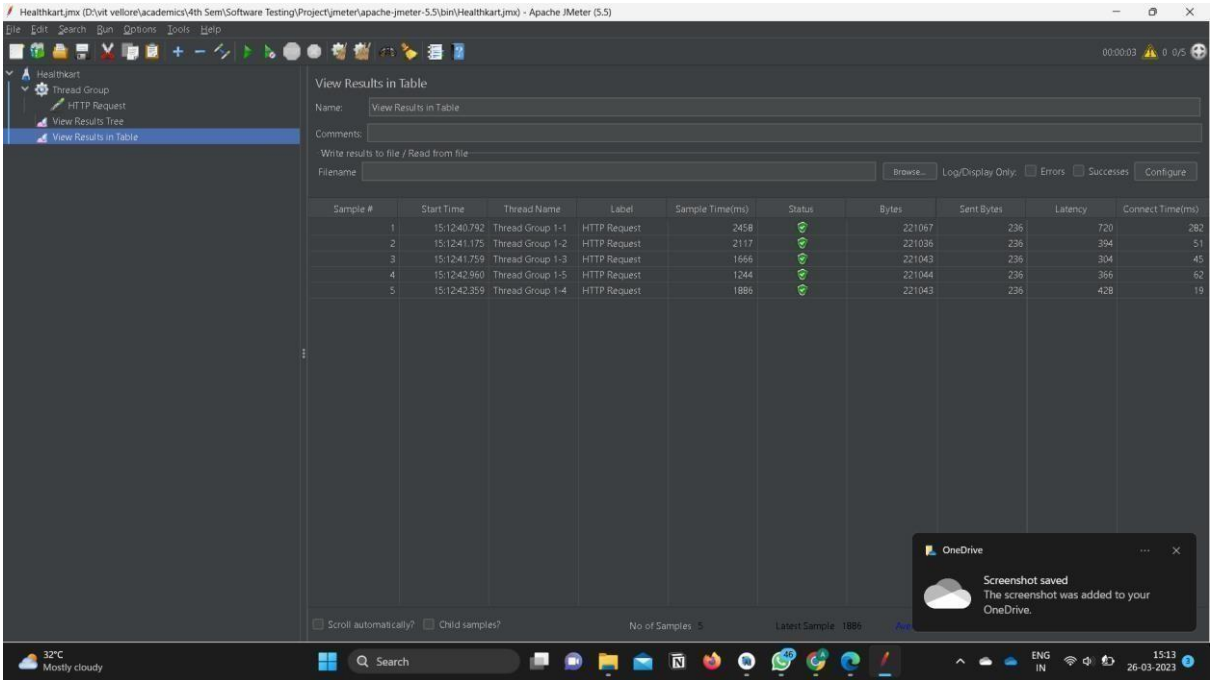

bin
Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Add
Paste Ctrl-V
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Threads (Users)
Config Element
Listener
Timer
Pre Processors
Post Processors
Assertions
Test Fragment
Non-Test Elements
View Results Tree
Summary Report
Aggregate Report
Backend Listener
Aggregate Graph
Assertion Results
Comparison Assertion Visualizer
Generate Summary Results
Graph Results
JSR223 Listener
Mailer Visualizer
Response Time Graph
Save Responses to a file
Simple Data Writer
View Results in Table
BeanShell Listener
User Defined Variables
Value
Add from Clipboard
Delete
Up
Down
Run tearDown Thread Groups after shutdown of main threads
Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C Mostly cloudy
Search
ENG IN
14:58 26-03-2023

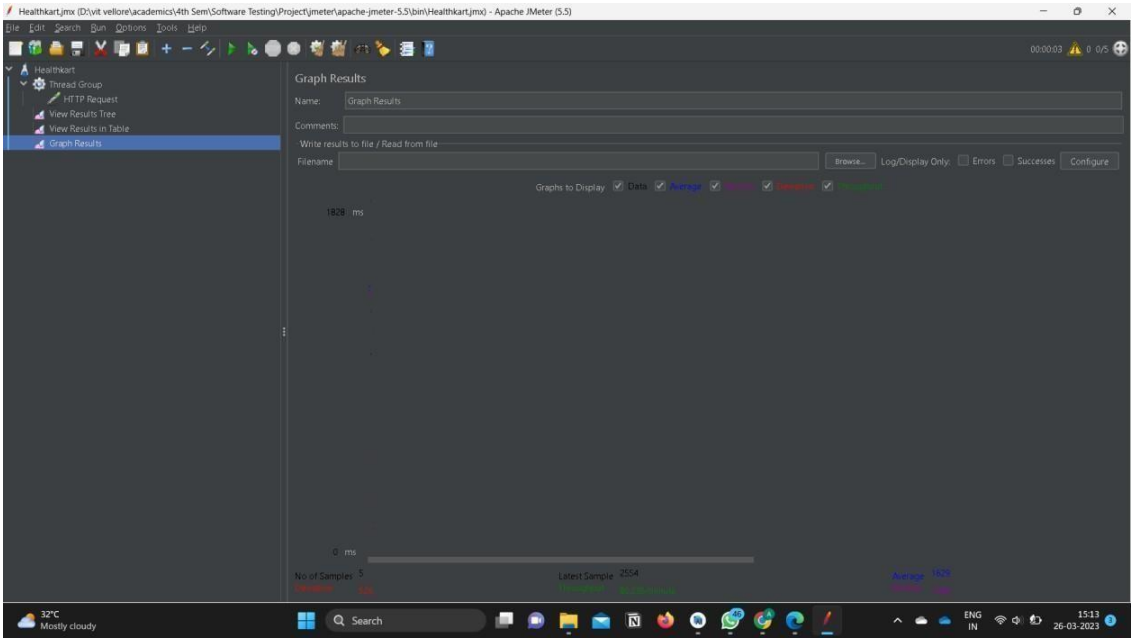

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
Add
Threads (Users)
Config Element
Listener
Timer
Pre Processors
Post Processors
Assertions
Test Fragment
Non-Test Elements
View Results Tree
Summary Report
Aggregate Report
Backend Listener
Aggregate Graph
Assertion Results
Comparison Assertion Visualizer
Generate Summary Results
Graph Results
JSR223 Listener
Mailer Visualizer
Response Time Graph
Save Responses to a file
Simple Data Writer
View Results in Table
BeanShell Listener
User Defined Variables
Add from Clipboard
Run tearDown Thread Groups after shutdown of main threads
Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library

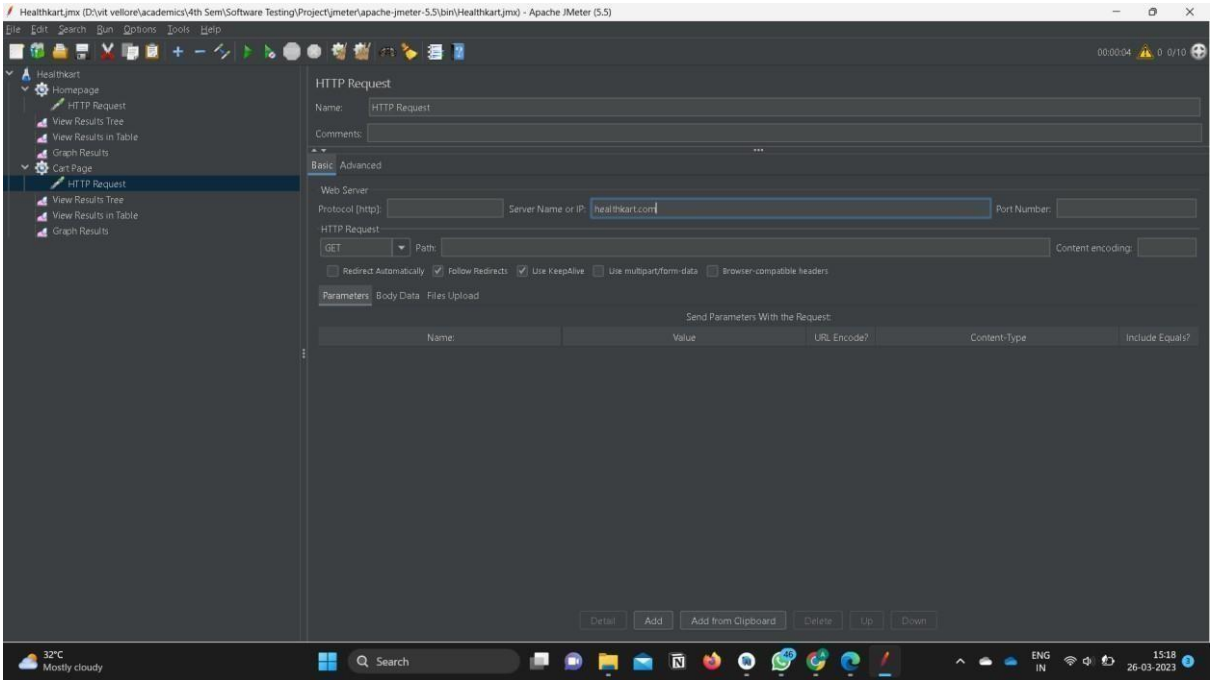

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
Add
Threads (Users)
Config Element
Listener
Timer
Pre Processors
Post Processors
Assertions
Test Fragment
Non-Test Elements
View Results Tree
Summary Report
Aggregate Report
Backend Listener
Aggregate Graph
Assertion Results
Comparison Assertion Visualizer
Generate Summary Results
Graph Results
JSR223 Listener
Mailer Visualizer
Response Time Graph
Save Responses to a file
Simple Data Writer
View Results in Table
BeanShell Listener
User Defined Variables
Add from Clipboard
Run tearDown Thread Groups after shutdown of main threads
Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library

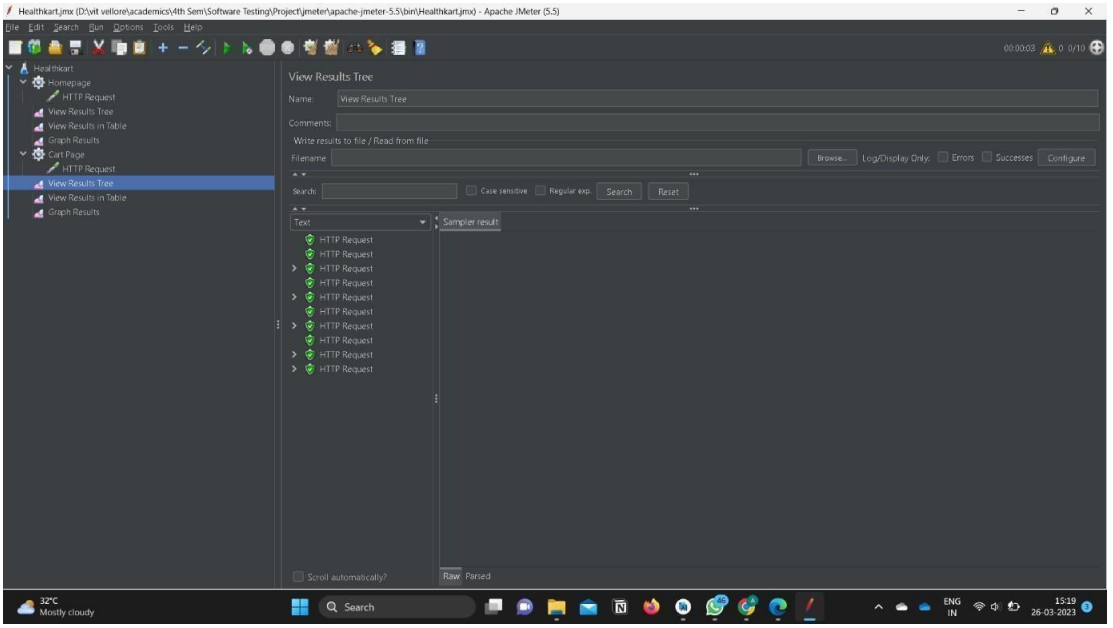

bin
Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Test Plan
Name: Healthkart
Comments:
User Defined Variables
Name:
Value
Detail
Add
Add from Clipboard
Delete
Up
Down
Run Thread Groups consecutively (i.e. one at a time)
Run tearDown Thread Groups after shutdown of main threads
Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library
krb5.conf 23-03-2023 22:06 CONF File 2 KB
log4j2.xml 23-03-2023 22:06 Microsoft Edge HT... 5 KB
43 items | 1 item selected 13.2 KB
32°C
Mostly cloudy
Search
ENG
IN
14:59
26-03-2023

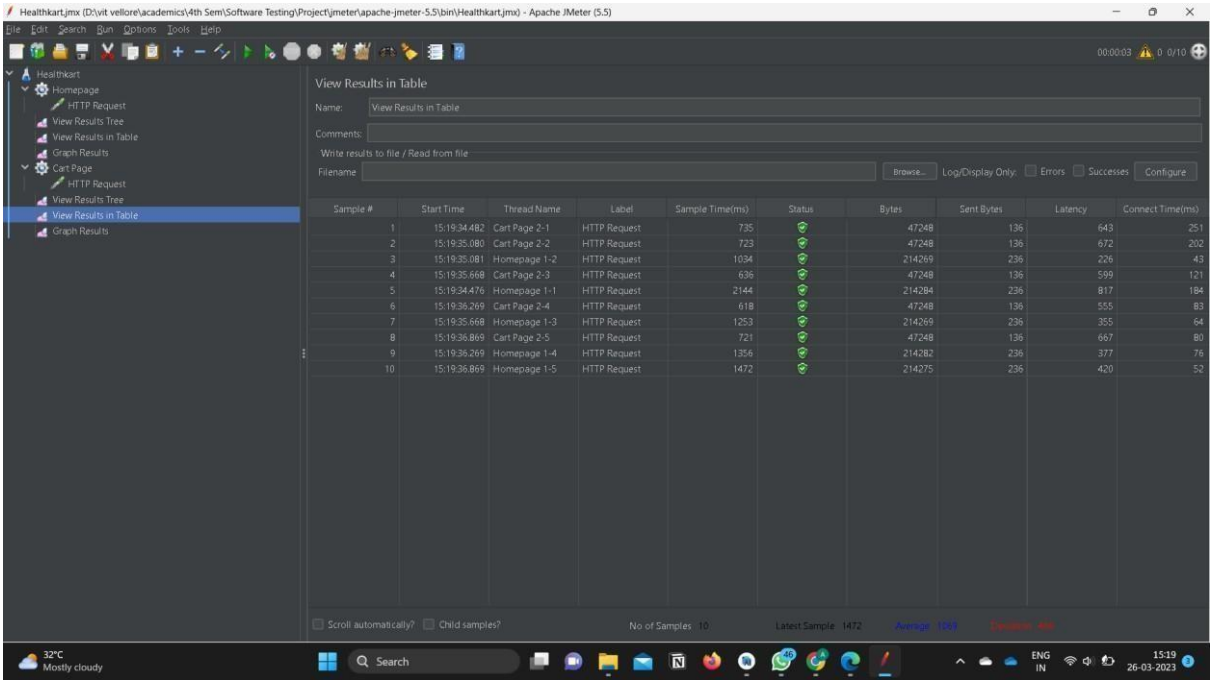

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Thread Group
HTTP Request
View Results Tree
View Results in Table
View Results Tree
Name: View Results Tree
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Search:
Case sensitive
Regular exp.
Search
Reset
Text
Sampler result
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
Scroll automatically?
Raw
Parsed
32°C
Mostly cloudy
Search
ENG
IN
15:12
26-03-2023

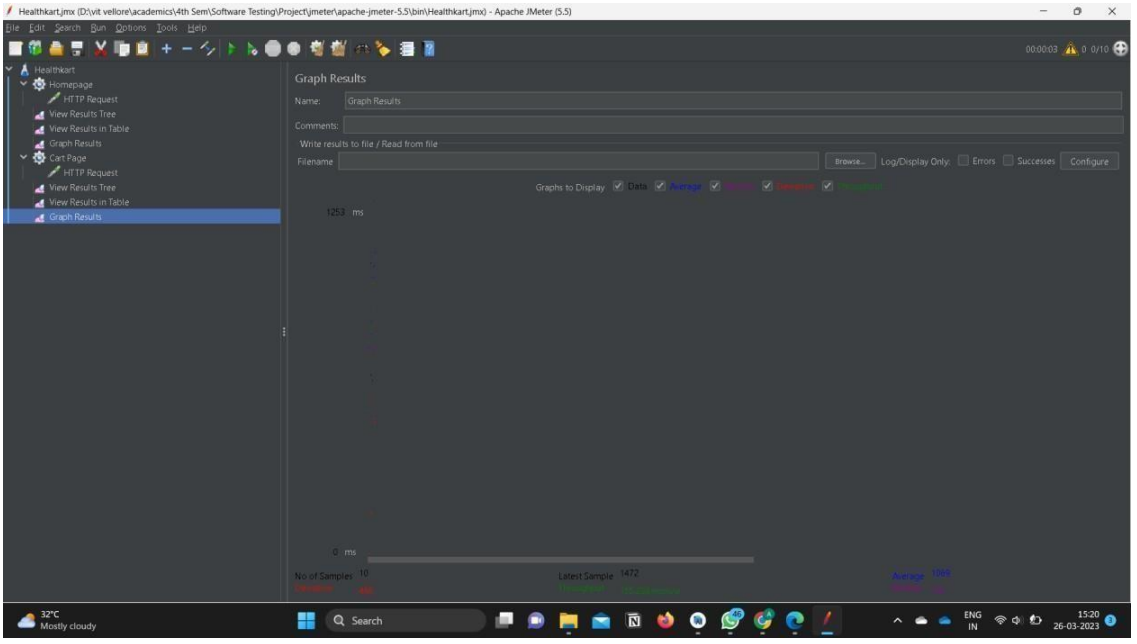

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Thread Group
HTTP Request
View Results Tree
View Results in Table
View Results in Table
Name: View Results in 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(ms) Status Bytes Sent Bytes Latency Connect Time(ms)
1 15:12:40.792 Thread Group 1-1 HTTP Request 2458 221067 236 720 282
2 15:12:41.175 Thread Group 1-2 HTTP Request 2117 221036 236 394 51
3 15:12:41.759 Thread Group 1-3 HTTP Request 1666 221043 236 304 45
4 15:12:42.960 Thread Group 1-5 HTTP Request 1244 221044 236 366 62
5 15:12:42.359 Thread Group 1-4 HTTP Request 1886 221043 236 428 19
Scroll automatically? Child samples?
No of Samples 5
Latest Sample 1886
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
32°C
Mostly cloudy
Search
ENG
IN
15:13
26-03-2023

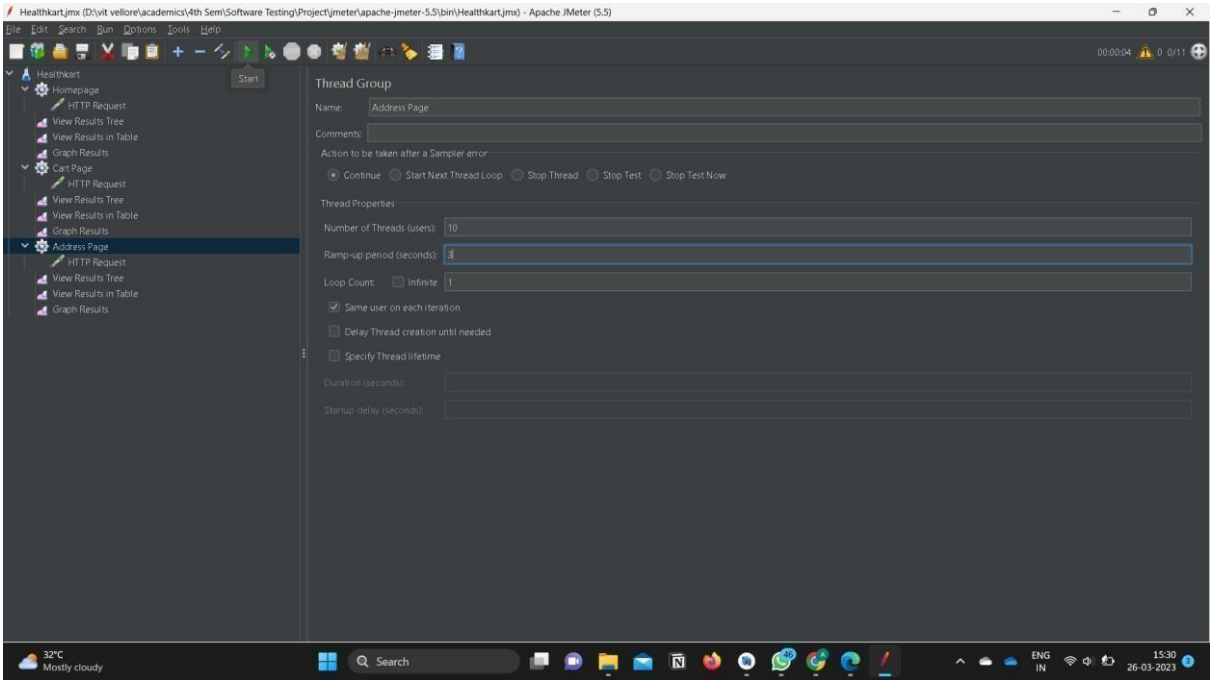

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Thread Group
HTTP Request
View Results Tree
View Results in Table
Graph Results
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse... Log/Display Only: Errors Successes Configure
Graphs to Display Data Average
1829 ms
0 ms
No of Samples 5
Latest Sample 2554
Average 1829
32°C
Mostly cloudy
Search
ENG
IN
15:13
26-03-2023

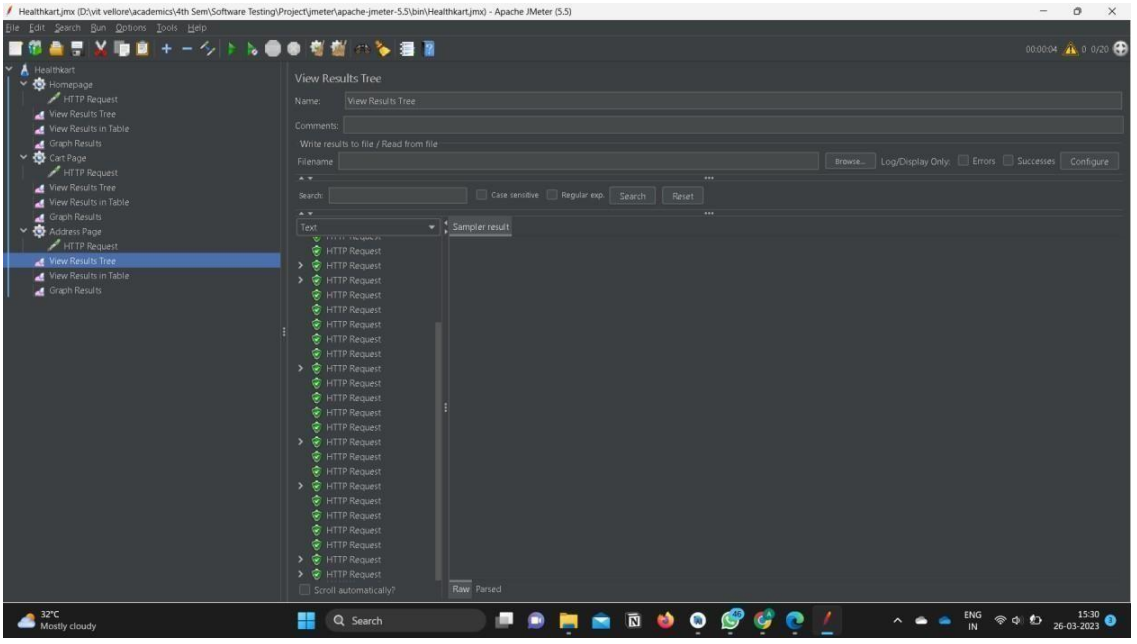

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol (http):
Server Name or IP: healthkart.com
Port Number:
HTTP Request
GET
Path:
Content encoding:
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?
Content-Type
Include Equals?
Detail
Add
Add from Clipboard
Delete
Up
Down
32°C
Mostly cloudy
Search
ENG
IN
15:18
26-03-2023

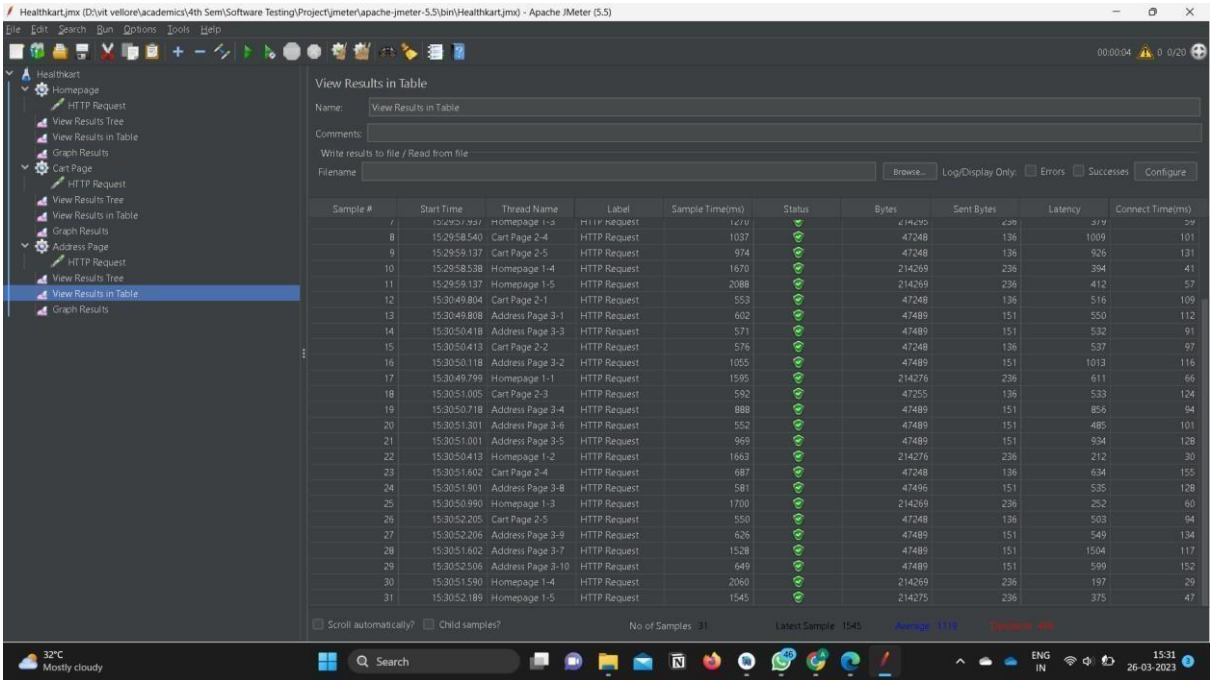

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
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
Sampler result
HTTP Request
Scroll automatically?
Raw
Parsed
32°C
Mostly cloudy
Search
ENG
IN
15:19
26-03-2023

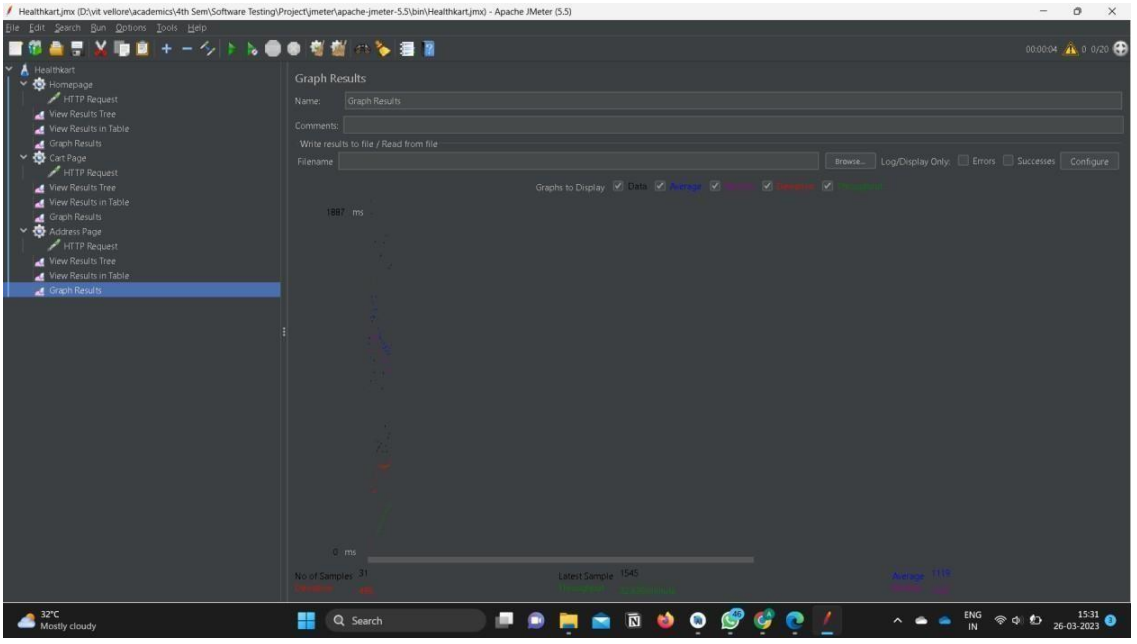

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
View Results in 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(ms) Status Bytes Sent Bytes Latency Connect Time(ms)
1 15:19:34.482 Cart Page 2-1 HTTP Request 735 47248 136 643 251
2 15:19:35.080 Cart Page 2-2 HTTP Request 723 47248 136 672 202
3 15:19:35.081 Homepage 1-2 HTTP Request 1034 214269 236 226 43
4 15:19:35.668 Cart Page 2-3 HTTP Request 636 47248 136 599 121
5 15:19:34.476 Homepage 1-1 HTTP Request 2144 214284 236 817 184
6 15:19:36.269 Cart Page 2-4 HTTP Request 618 47248 136 555 83
7 15:19:35.668 Homepage 1-3 HTTP Request 1253 214269 236 355 64
8 15:19:36.869 Cart Page 2-5 HTTP Request 721 47248 136 667 80
9 15:19:36.269 Homepage 1-4 HTTP Request 1356 214282 236 377 76
10 15:19:36.869 Homepage 1-5 HTTP Request 1472 214275 236 420 52
Scroll automatically? Child samples? No of Samples 10 Latest Sample 1472 Average 1069 Deviation 400
32°C Mostly cloudy
Search
ENG IN
15:19 26-03-2023

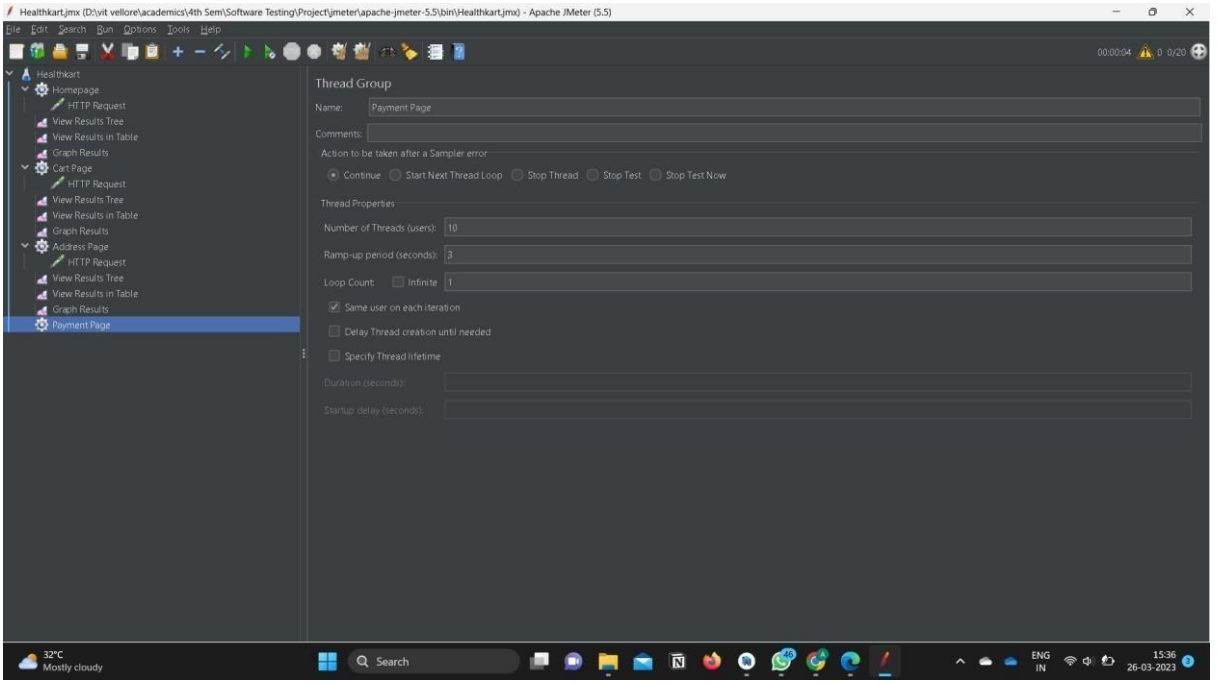

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse... Log/Display Only: Errors Successes Configure
Graphs to Display Data Average
1253 ms
0 ms
No of Samples 10
Latest Sample 1472
Average 1069
32°C Mostly cloudy
Search
ENG IN
15:20 26-03-2023

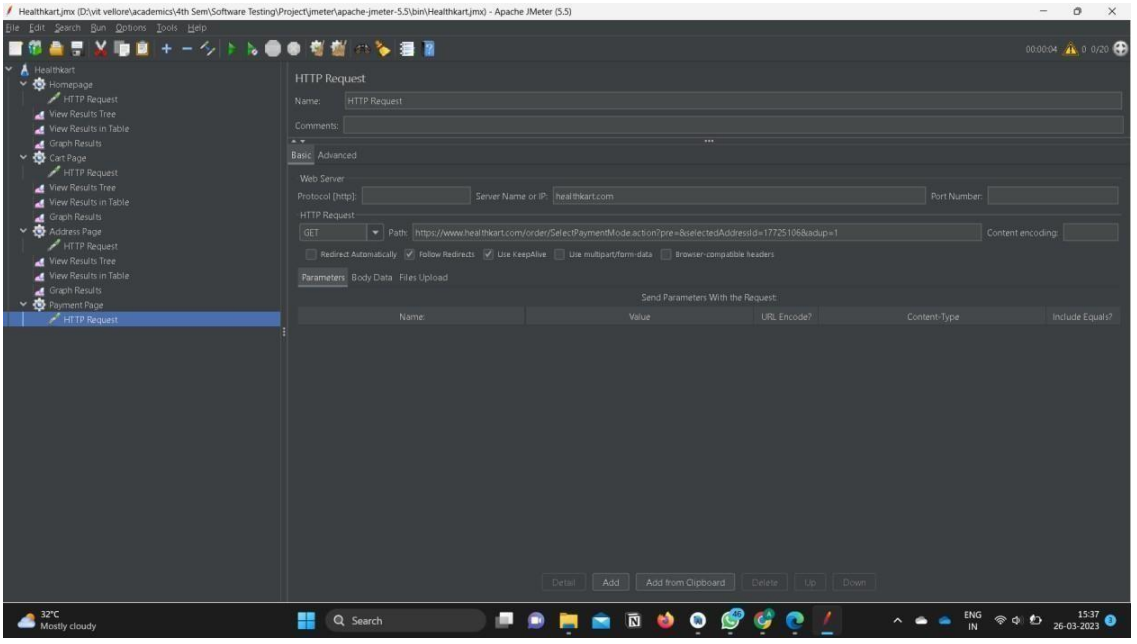

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Start
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Thread Group
Name: Address Page
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now
Thread Properties
Number of Threads (users): 10
Ramp-up period (seconds): 3
Loop Count: Infinite 1
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):
32°C
Mostly cloudy
Search
ENG
IN
15:30
26-03-2023

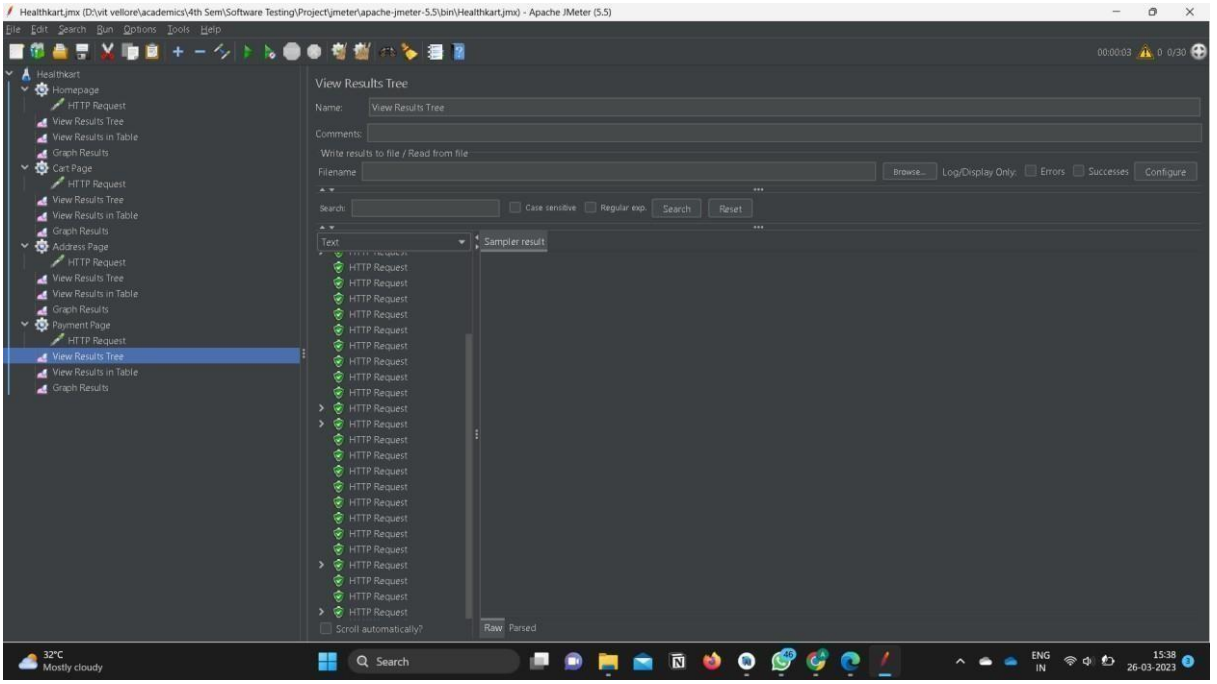

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
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
Sampler result
HTTP Request
Scroll automatically?
Raw Parsed
32°C
Mostly cloudy
Search
ENG
IN
15:30
26-03-2023

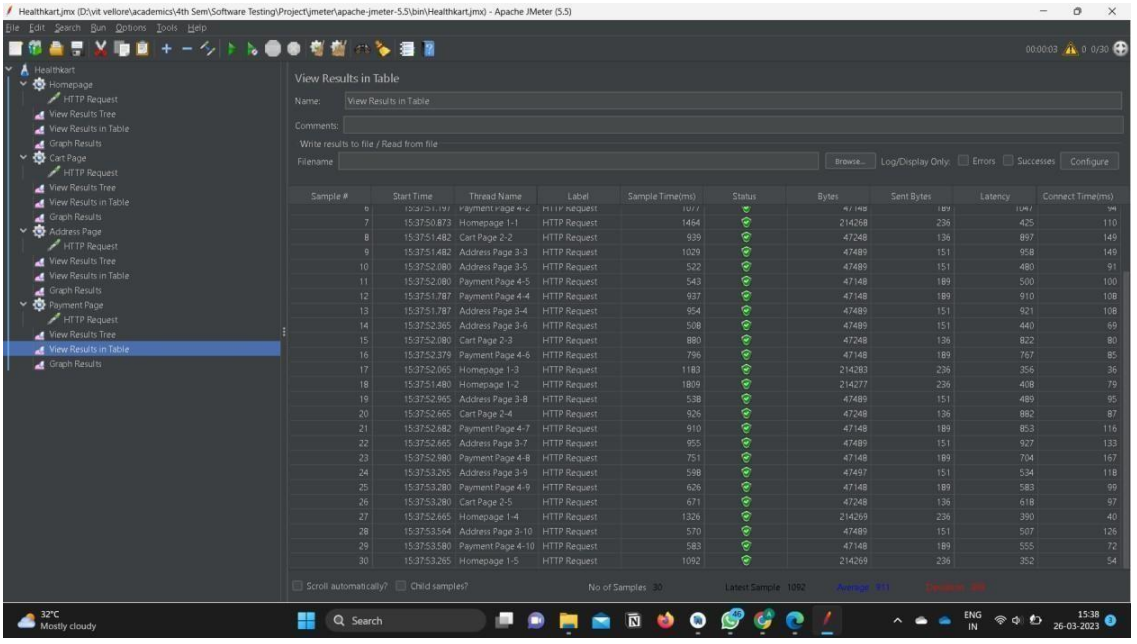

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
View Results in 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(ms) Status Bytes Sent Bytes Latency Connect Time(ms)
8 15:29:58.540 Cart Page 2-4 HTTP Request 1037 47248 136 1009 101
9 15:29:59.137 Cart Page 2-5 HTTP Request 974 47248 136 926 131
10 15:29:58.538 Homepage 1-4 HTTP Request 1670 214269 236 394 41
11 15:29:59.137 Homepage 1-5 HTTP Request 2088 214269 236 412 57
12 15:30:49.804 Cart Page 2-1 HTTP Request 553 47248 136 516 109
13 15:30:49.808 Address Page 3-1 HTTP Request 602 47489 151 550 112
14 15:30:50.418 Address Page 3-3 HTTP Request 571 47489 151 532 91
15 15:30:50.413 Cart Page 2-2 HTTP Request 576 47248 136 537 97
16 15:30:50.118 Address Page 3-2 HTTP Request 1055 47489 151 1013 116
17 15:30:49.799 Homepage 1-1 HTTP Request 1595 214276 236 611 66
18 15:30:51.005 Cart Page 2-3 HTTP Request 592 47255 136 533 124
19 15:30:50.718 Address Page 3-4 HTTP Request 888 47489 151 856 94
20 15:30:51.301 Address Page 3-6 HTTP Request 552 47489 151 485 101
21 15:30:51.001 Address Page 3-5 HTTP Request 969 47489 151 934 128
22 15:30:50.413 Homepage 1-2 HTTP Request 1663 214276 236 212 30
23 15:30:51.602 Cart Page 2-4 HTTP Request 687 47248 136 634 155
24 15:30:51.901 Address Page 3-8 HTTP Request 581 47496 151 535 128
25 15:30:50.990 Homepage 1-3 HTTP Request 1700 214269 236 252 60
26 15:30:52.205 Cart Page 2-5 HTTP Request 550 47248 136 503 94
27 15:30:52.206 Address Page 3-9 HTTP Request 626 47489 151 549 134
28 15:30:51.602 Address Page 3-7 HTTP Request 1528 47489 151 1504 117
29 15:30:52.506 Address Page 3-10 HTTP Request 649 47489 151 599 152
30 15:30:51.590 Homepage 1-4 HTTP Request 2060 214269 236 197 29
31 15:30:52.189 Homepage 1-5 HTTP Request 1545 214275 236 375 47
Scroll automatically? Child samples? No of Samples 31 Latest Sample 1545 Average 1119 Deviation 466
32°C Mostly cloudy
Search
ENG IN
15:31 26-03-2023

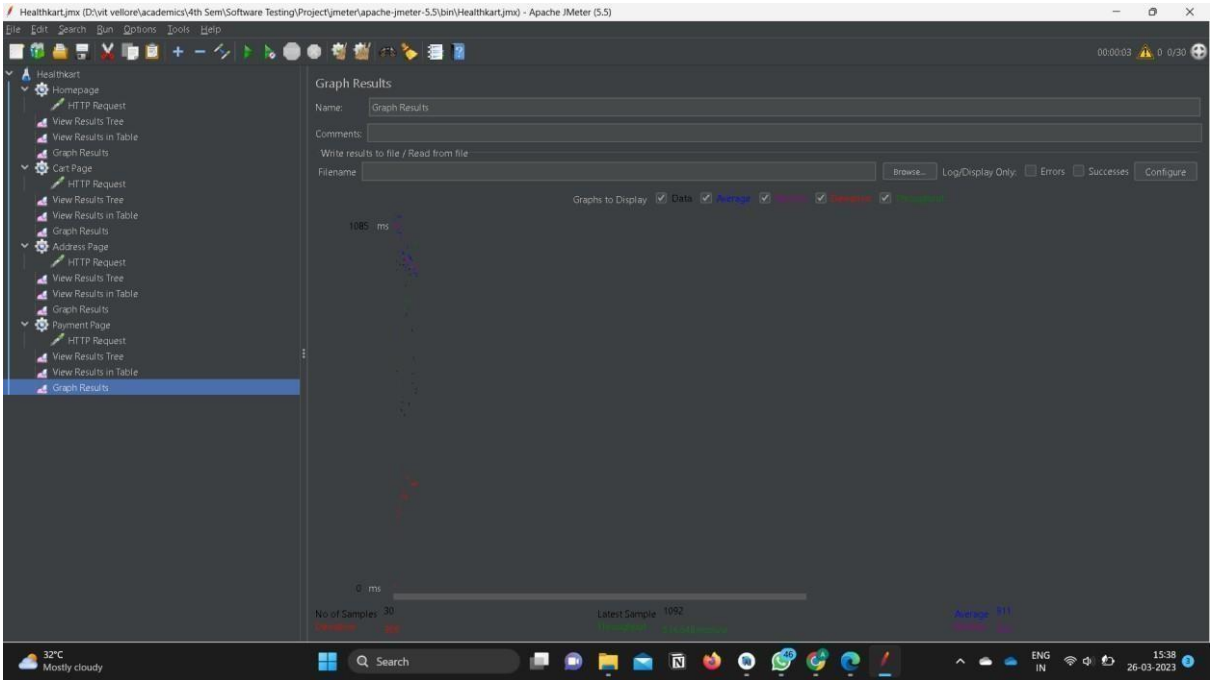

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse... Log/Display Only: Errors Successes Configure
Graphs to Display Data Average
1887 ms
0 ms
No of Samples 31
Latest Sample 1545
Average 1119
32°C Mostly cloudy
Search
ENG IN
15:31 26-03-2023


Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
Thread Group
Name: Payment Page
Comments:
Action to be taken after a Sampler error
Continue Start Next Thread Loop Stop Thread Stop Test Stop Test Now
Thread Properties
Number of Threads (users): 10
Ramp-up period (seconds): 3
Loop Count: Infinite 1
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):
00:00:04 0 0/20
32°C Mostly cloudy
Search
ENG IN
15:36 26-03-2023

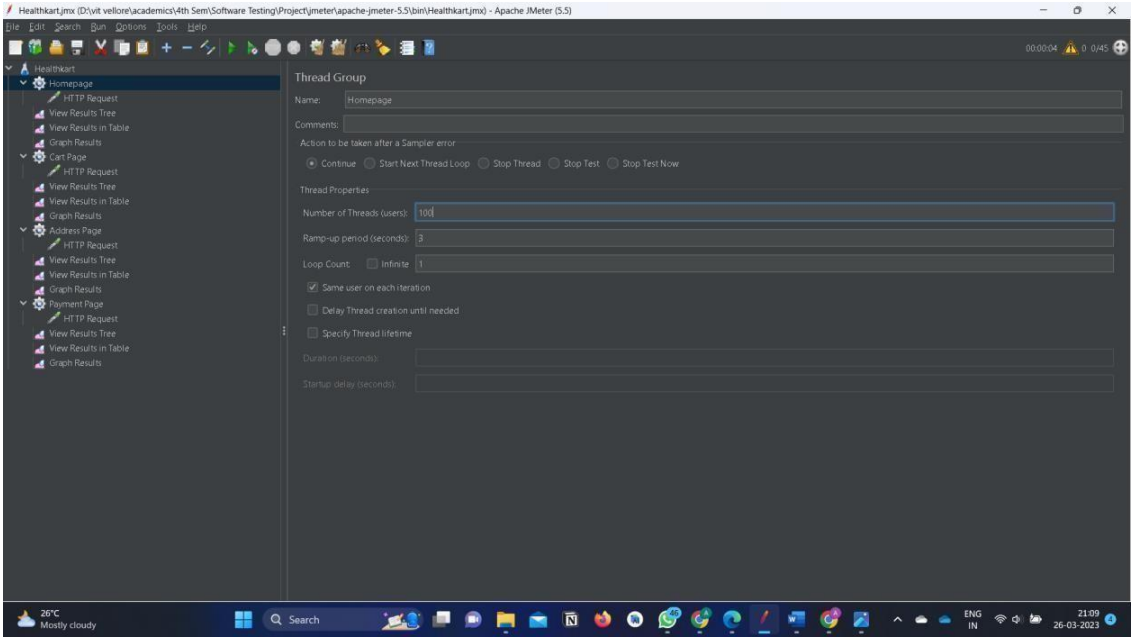

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]: Server Name or IP: healthkart.com Port Number:
HTTP Request
GET Path: https://www.healthkart.com/order/SelectPaymentMode.action?pre=&selectedAddressId=17725106&adcup=1 Content encoding:
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? Content-Type Include Equals?
Detail Add Add from Clipboard Delete Up Down
00:00:04 0 0/20
32°C Mostly cloudy
Search
ENG IN
15:37 26-03-2023

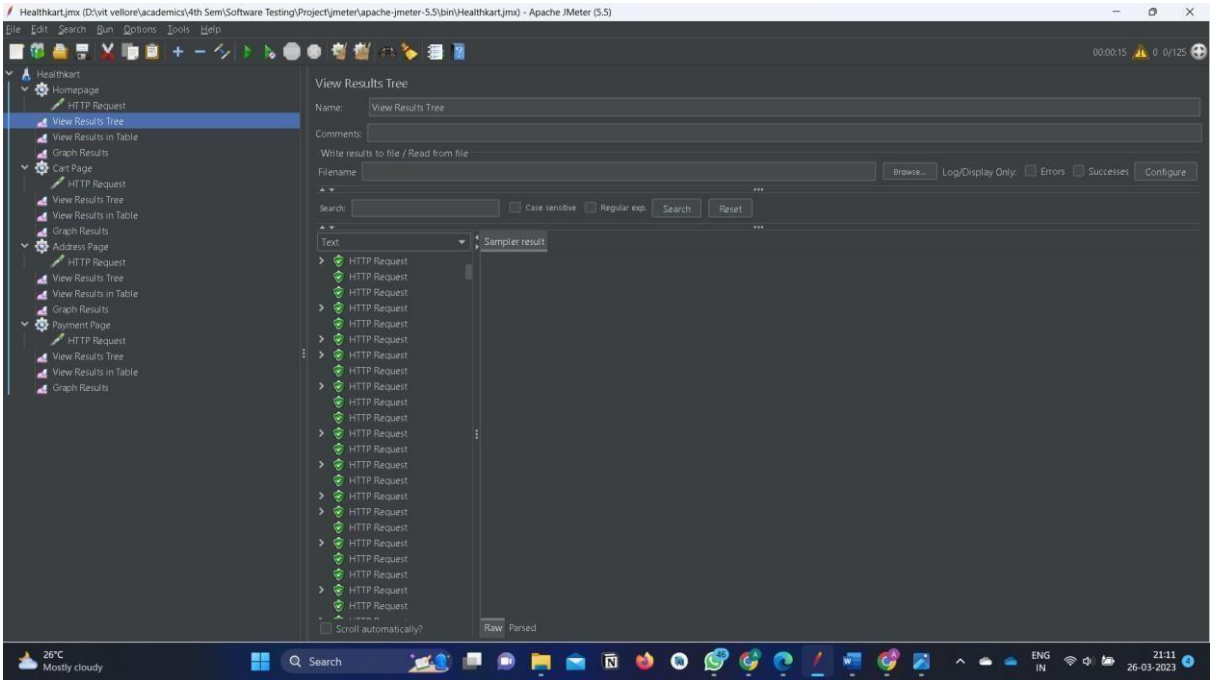

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
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
Sampler result
Scroll automatically?
Raw Parsed
32°C Mostly cloudy
Search
ENG IN
15:38 26-03-2023

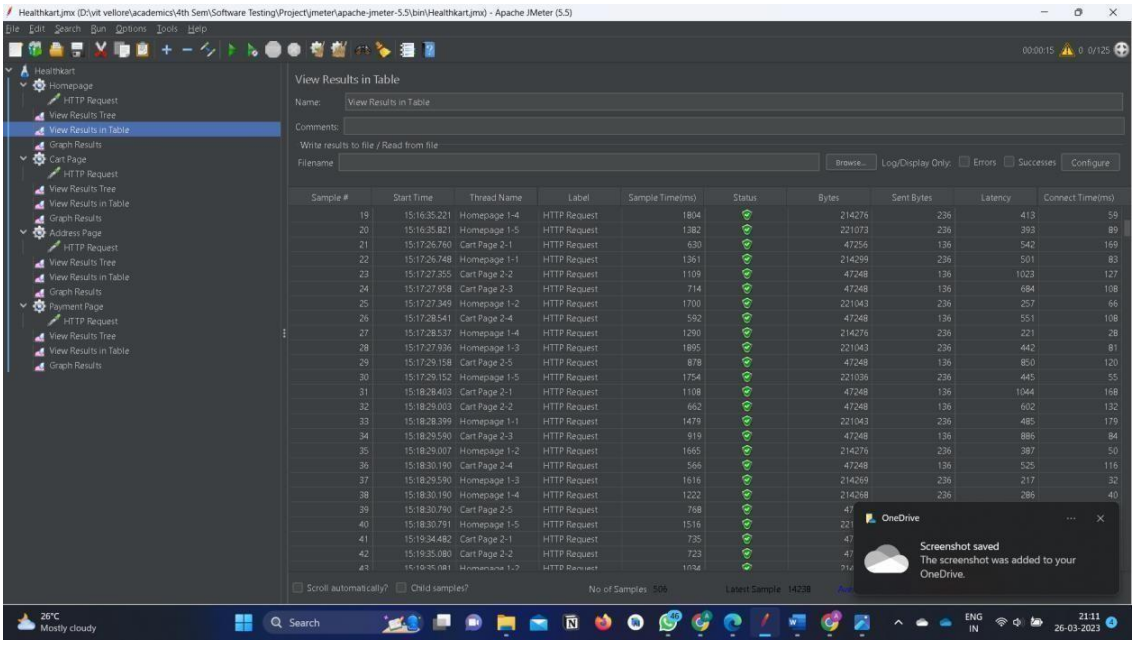

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
View Results in 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(ms) Status Bytes Sent Bytes Latency Connect Time(ms)
7 15:37:50.873 Homepage 1-1 HTTP Request 1464 214268 236 425 110
8 15:37:51.482 Cart Page 2-2 HTTP Request 939 47248 136 897 149
9 15:37:51.482 Address Page 3-3 HTTP Request 1029 47489 151 958 149
10 15:37:52.080 Address Page 3-5 HTTP Request 522 47489 151 480 91
11 15:37:52.080 Payment Page 4-5 HTTP Request 543 47148 189 500 100
12 15:37:51.787 Payment Page 4-4 HTTP Request 937 47148 189 910 108
13 15:37:51.787 Address Page 3-4 HTTP Request 954 47489 151 921 108
14 15:37:52.365 Address Page 3-6 HTTP Request 508 47489 151 440 69
15 15:37:52.080 Cart Page 2-3 HTTP Request 880 47248 136 822 80
16 15:37:52.379 Payment Page 4-6 HTTP Request 796 47148 189 767 85
17 15:37:52.065 Homepage 1-3 HTTP Request 1183 214283 236 356 36
18 15:37:51.480 Homepage 1-2 HTTP Request 1809 214277 236 408 79
19 15:37:52.965 Address Page 3-8 HTTP Request 538 47489 151 489 95
20 15:37:52.665 Cart Page 2-4 HTTP Request 926 47248 136 882 87
21 15:37:52.682 Payment Page 4-7 HTTP Request 910 47148 189 853 116
22 15:37:52.665 Address Page 3-7 HTTP Request 955 47489 151 927 133
23 15:37:52.980 Payment Page 4-8 HTTP Request 751 47148 189 704 167
24 15:37:53.265 Address Page 3-9 HTTP Request 598 47497 151 534 118
25 15:37:53.280 Payment Page 4-9 HTTP Request 626 47148 189 583 99
26 15:37:53.280 Cart Page 2-5 HTTP Request 671 47248 136 618 97
27 15:37:52.665 Homepage 1-4 HTTP Request 1326 214269 236 390 40
28 15:37:53.564 Address Page 3-10 HTTP Request 570 47489 151 507 126
29 15:37:53.580 Payment Page 4-10 HTTP Request 583 47148 189 555 72
30 15:37:53.265 Homepage 1-5 HTTP Request 1092 214269 236 352 54
Scroll automatically? Child samples? No of Samples 30 Latest Sample 1092
32°C Mostly cloudy
Search
ENG IN
15:38 26-03-2023

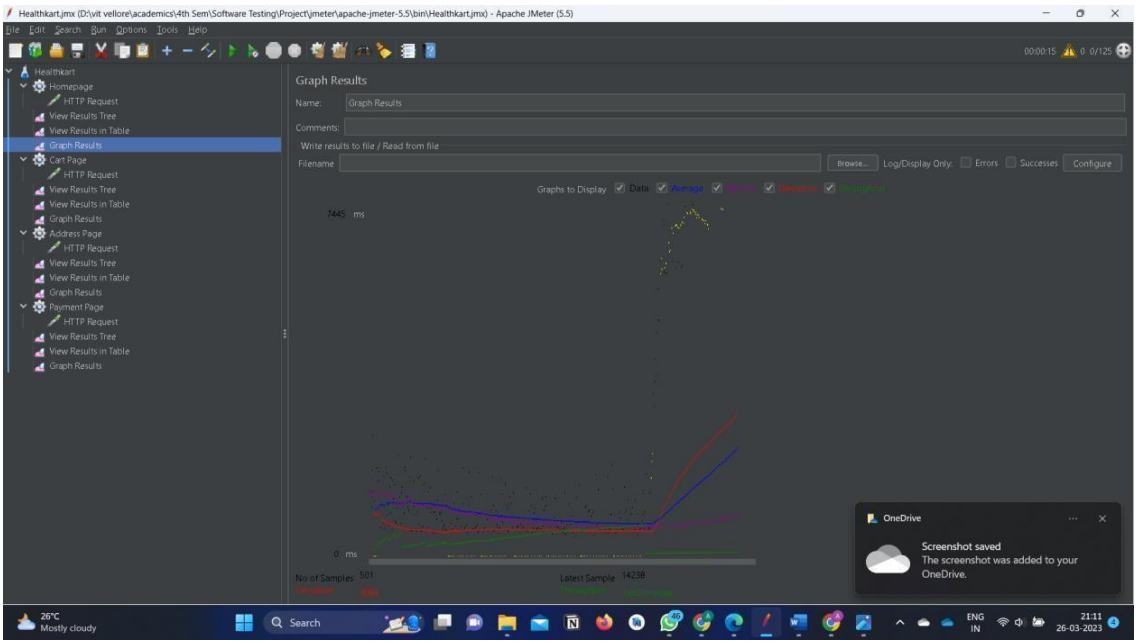


After finishing the process we can also change the no. of threads (no. of users)




Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
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
Sampler result
HTTP Request
Scroll automatically?
Raw Parsed
26°C
Mostly cloudy
Search
ENG
IN
21:11
26-03-2023

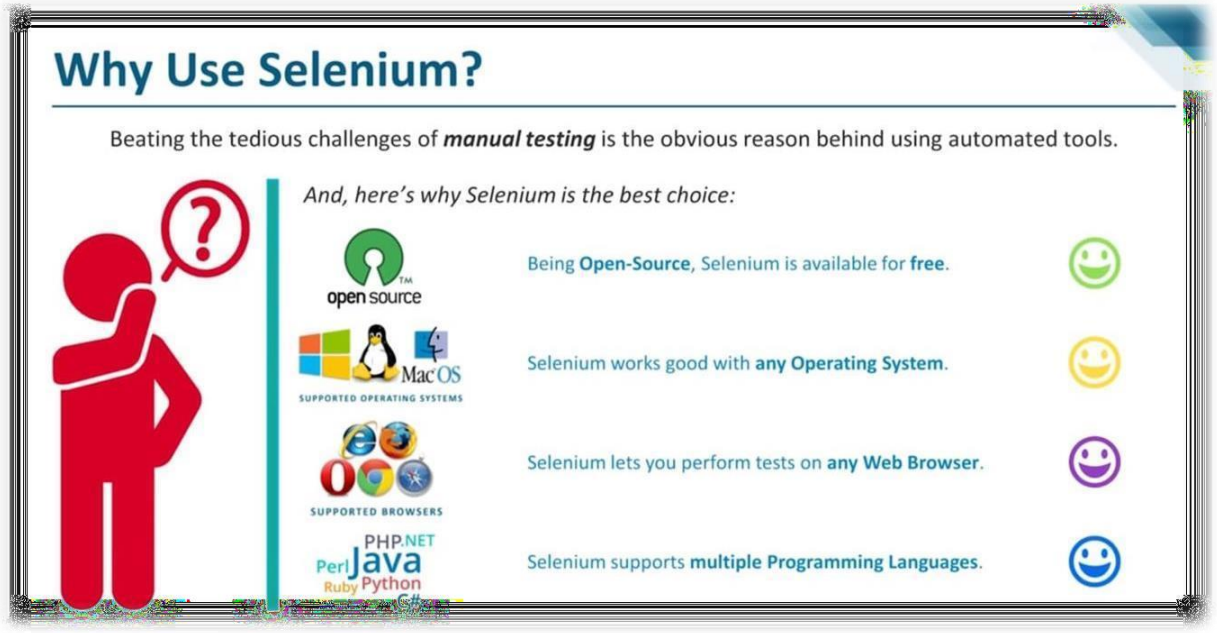

Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
View Results in 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(ms) Status Bytes Sent Bytes Latency Connect Time(ms)
19 15:16:35.221 Homepage 1-4 HTTP Request 1804 214276 236 413 59
20 15:16:35.821 Homepage 1-5 HTTP Request 1382 221073 236 393 89
21 15:17:26.760 Cart Page 2-1 HTTP Request 630 47256 136 542 169
22 15:17:26.748 Homepage 1-1 HTTP Request 1361 214299 236 501 83
23 15:17:27.355 Cart Page 2-2 HTTP Request 1109 47248 136 1023 127
24 15:17:27.958 Cart Page 2-3 HTTP Request 714 47248 136 684 108
25 15:17:27.349 Homepage 1-2 HTTP Request 1700 221043 236 257 66
26 15:17:28.541 Cart Page 2-4 HTTP Request 592 47248 136 551 108
27 15:17:28.537 Homepage 1-4 HTTP Request 1290 214276 236 221 28
28 15:17:27.936 Homepage 1-3 HTTP Request 1895 221043 236 442 81
29 15:17:29.158 Cart Page 2-5 HTTP Request 878 47248 136 850 120
30 15:17:29.152 Homepage 1-5 HTTP Request 1754 221036 236 445 55
31 15:18:28.403 Cart Page 2-1 HTTP Request 1108 47248 136 1044 168
32 15:18:29.003 Cart Page 2-2 HTTP Request 662 47248 136 602 132
33 15:18:28.399 Homepage 1-1 HTTP Request 1479 221043 236 485 179
34 15:18:29.590 Cart Page 2-3 HTTP Request 919 47248 136 886 84
35 15:18:29.007 Homepage 1-2 HTTP Request 1665 214276 236 387 50
36 15:18:30.190 Cart Page 2-4 HTTP Request 566 47248 136 525 116
37 15:18:29.590 Homepage 1-3 HTTP Request 1616 214269 236 217 32
38 15:18:30.190 Homepage 1-4 HTTP Request 1222 214268 236 286 40
39 15:18:30.790 Cart Page 2-5 HTTP Request 768 47
40 15:18:30.791 Homepage 1-5 HTTP Request 1516 221
41 15:19:34.482 Cart Page 2-1 HTTP Request 735 47
42 15:19:35.080 Cart Page 2-2 HTTP Request 723 47
Scroll automatically? Child samples? No of Samples 506 Latest Sample 14238
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
26°C
Mostly cloudy
Search
ENG
IN
21:11
26-03-2023


Healthkart.jmx (D:\vit vellore\academics\4th Sem\Software Testing\Project\jmeter\apache-jmeter-5.5\bin\Healthkart.jmx) - Apache JMeter (5.5)
File Edit Search Run Options Tools Help
Healthkart
Homepage
HTTP Request
View Results Tree
View Results in Table
Graph Results
Cart Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Address Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Payment Page
HTTP Request
View Results Tree
View Results in Table
Graph Results
Graph Results
Name: Graph Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only:
Errors
Successes
Configure
Graphs to Display
Data
Average
7445 ms
0 ms
No of Samples 501
Latest Sample 1423
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
26°C
Mostly cloudy
Search
ENG
IN
21:11
26-03-2023

SELENIUM TESTING TOOL

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to
as **Selenium Testing.**

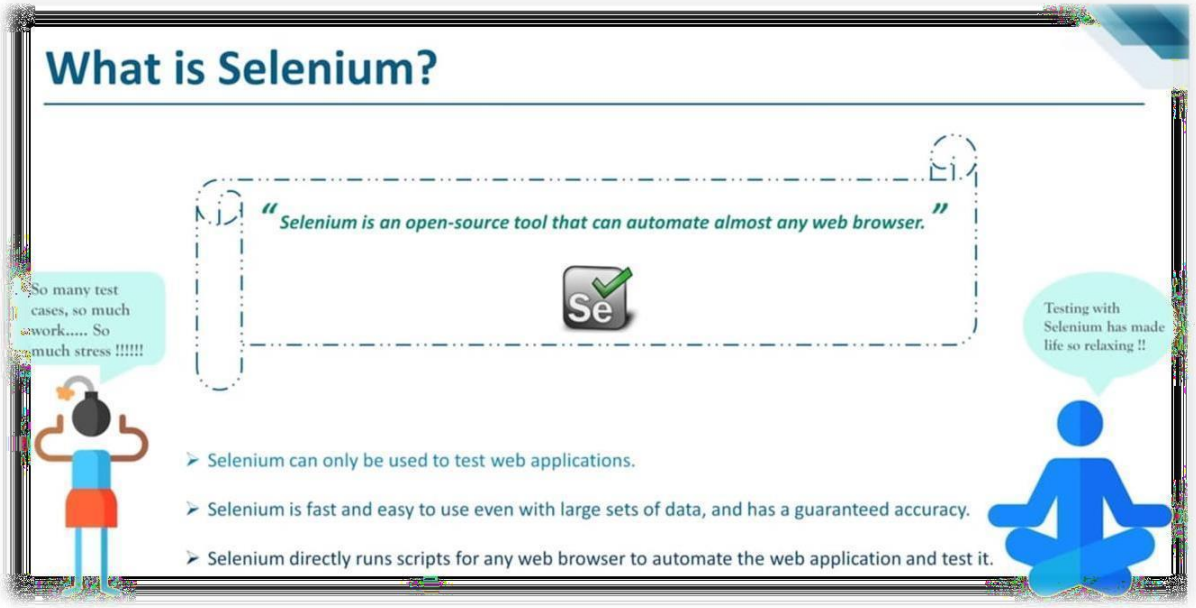
Why use selenium ?

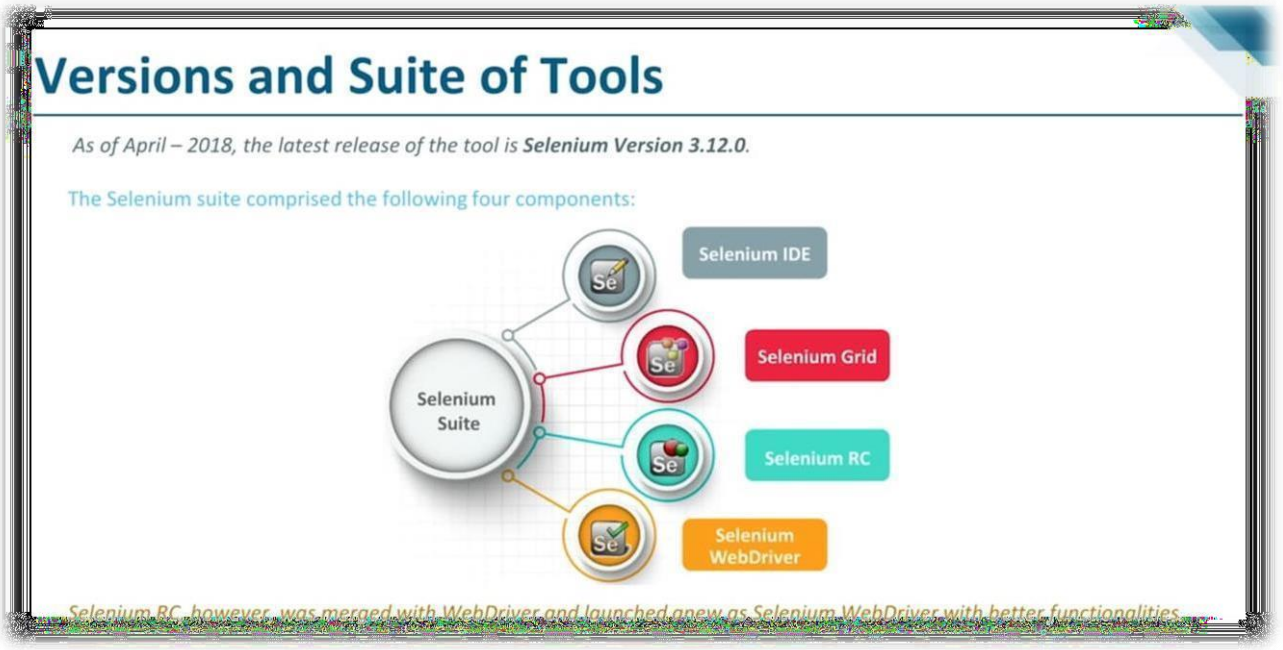


Selenium vs. its Counterparts

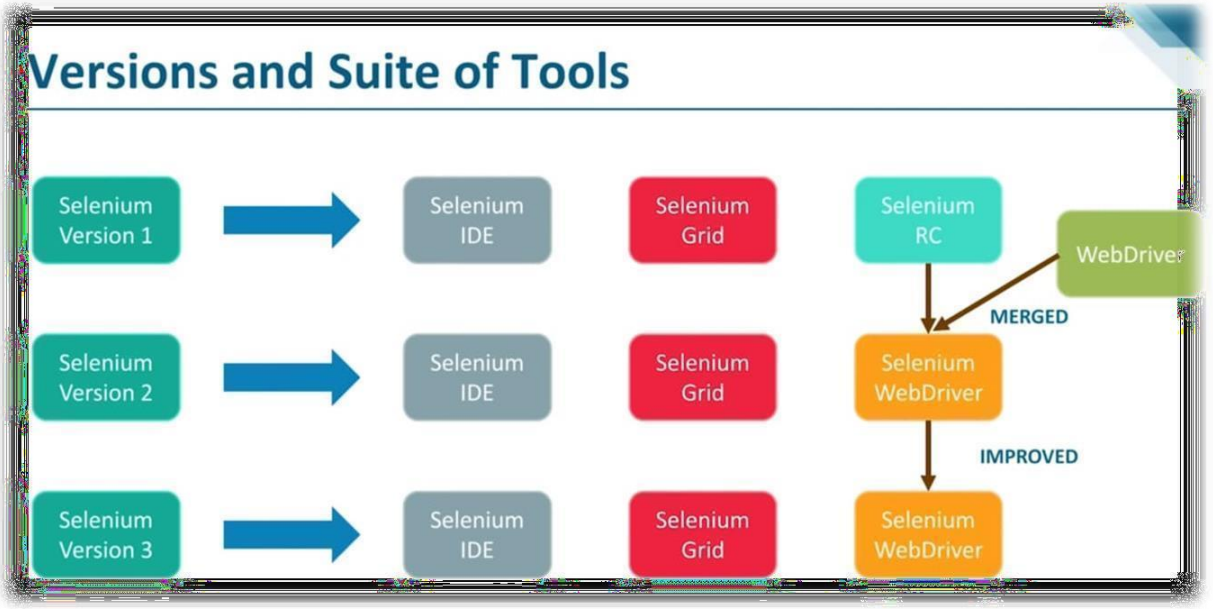
Features	HP QTP	IBM RFT	TestComplete	Selenium
License	Required	Required	Required	Open Source
Cost	High	High	High	Free
Customer support	Yes	Yes	Yes	Yes; Open source community
Coding skills	Low	Low	High	Very High
Environment support	Only Windows	Only Windows	Windows only (7, Vista, Server 2008 or later OS)	Windows, Linux, Mac
Language support	VB Script	Java and C#	VB Script, JS Script, Delphi Script, C++ & C#	Java, C#, Ruby, Python, Perl & PHP

What is selenium ?

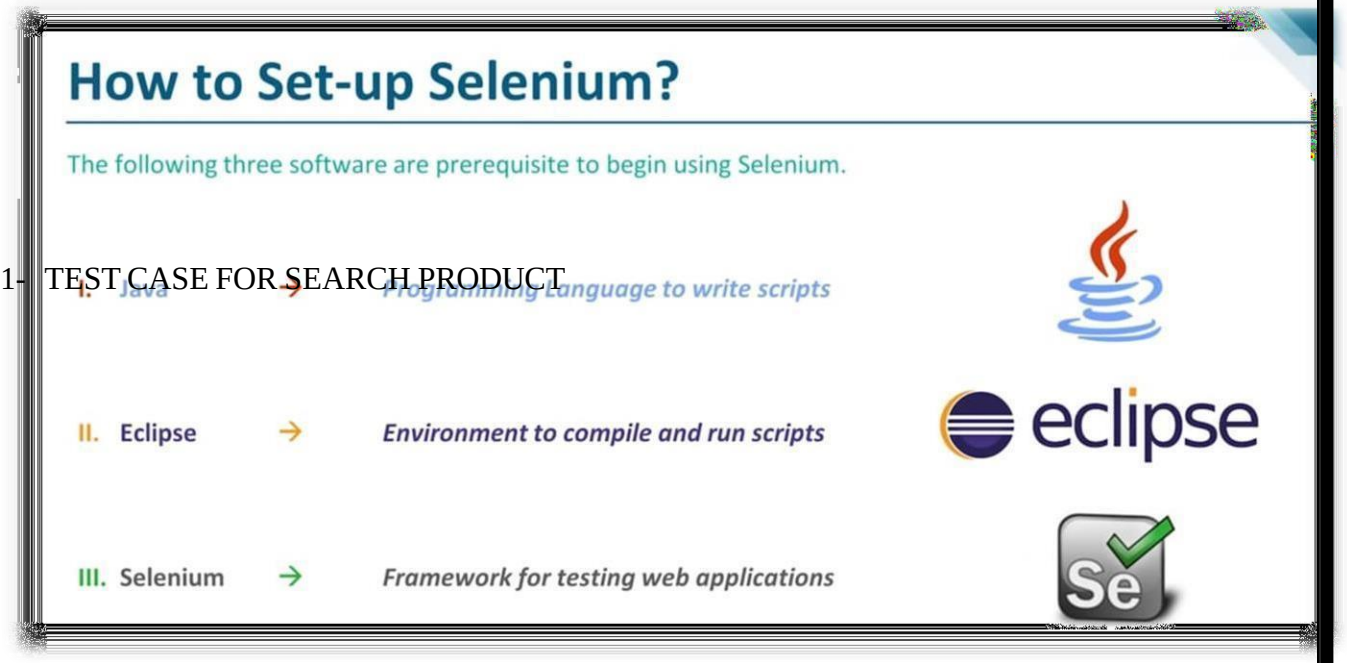




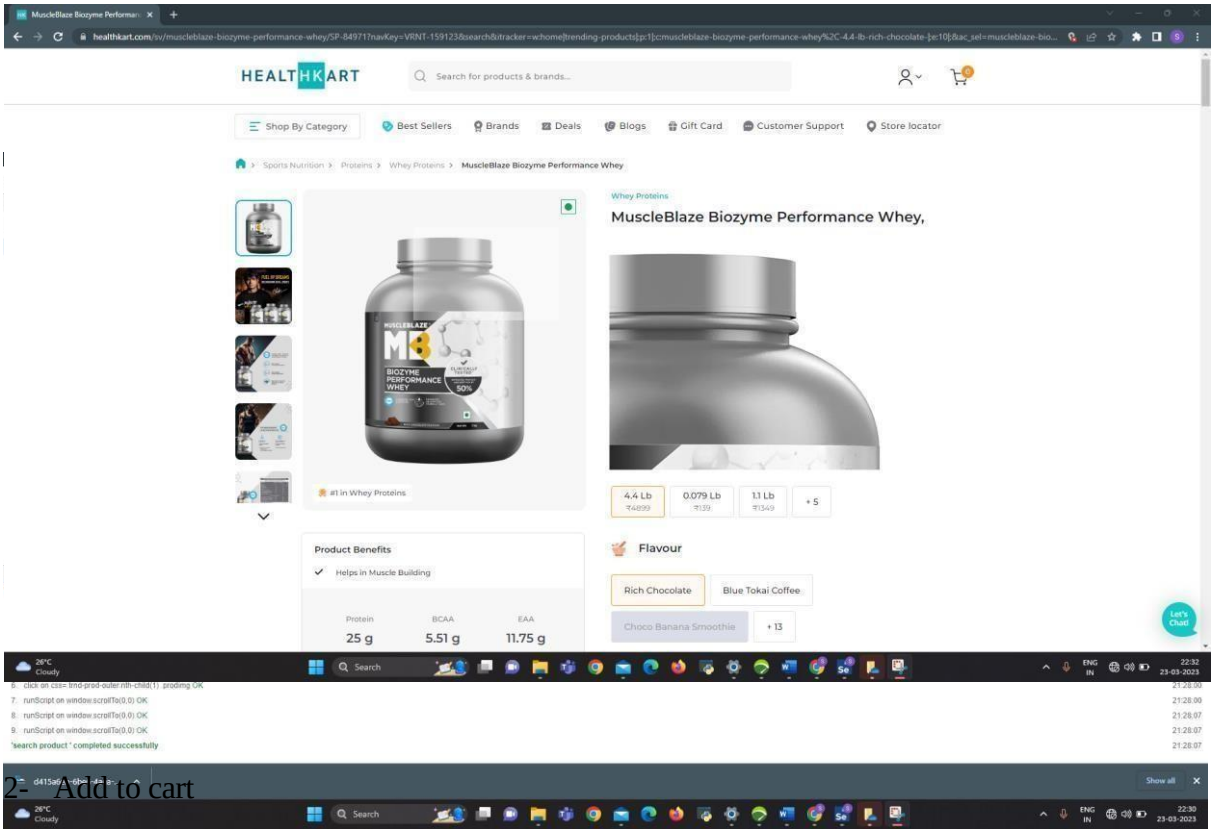
Version and suite of tools for selenium



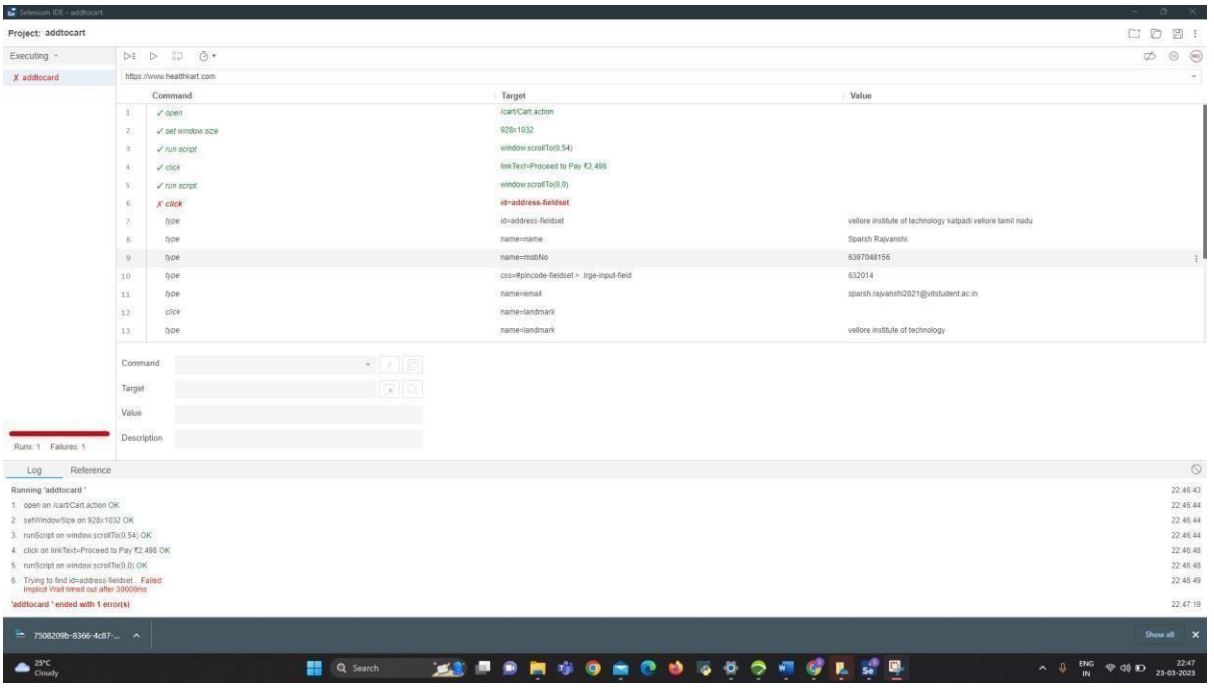
How to setup selenium

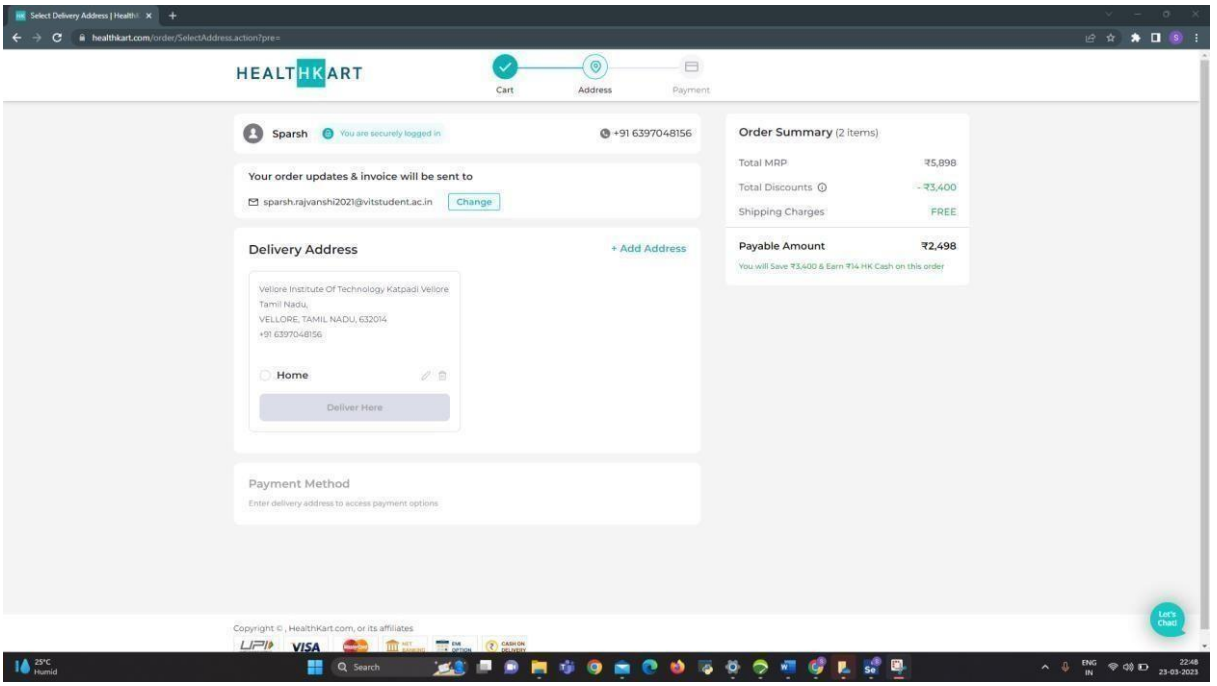


1- TEST CASE FOR SEARCH PRODUCT

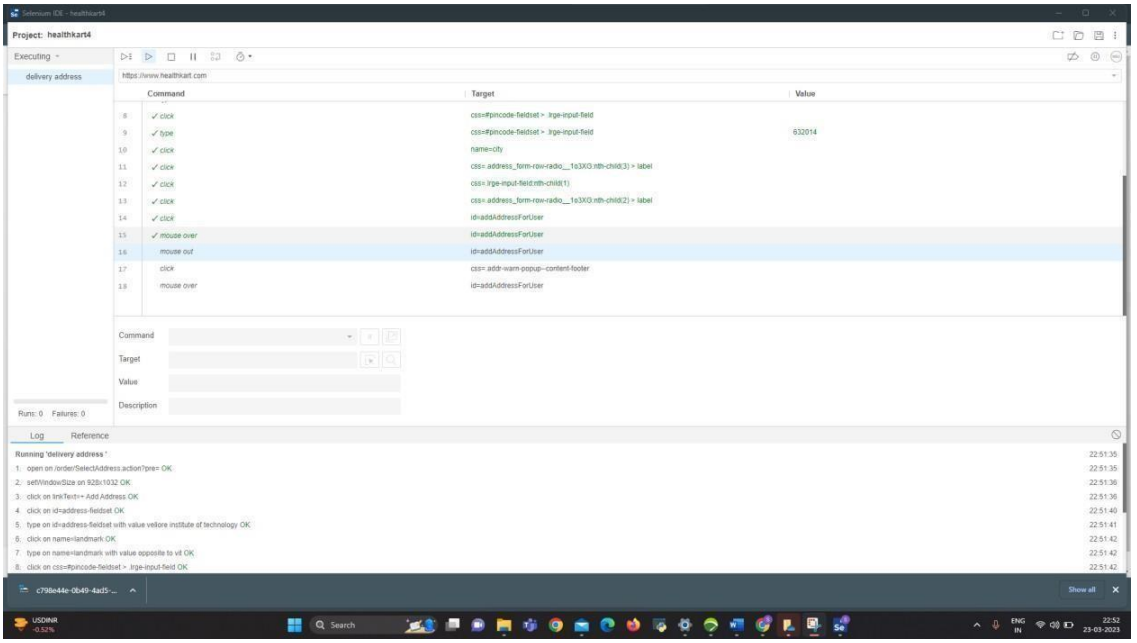


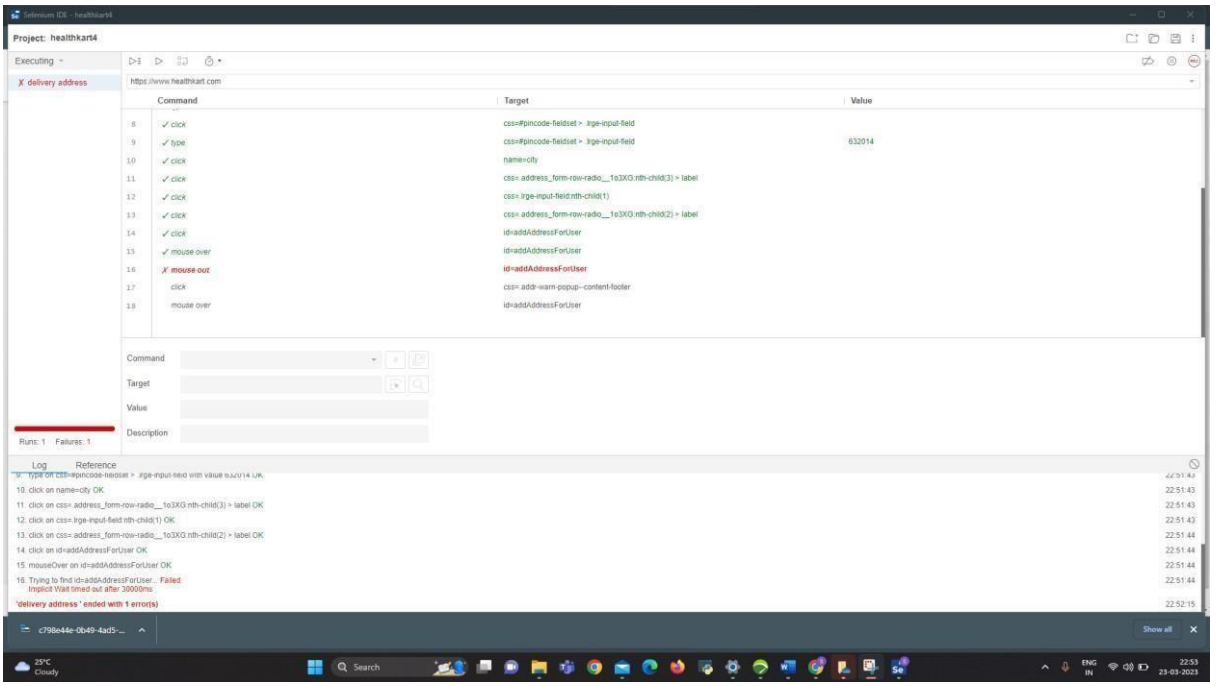
2- Add to cart


Selenium IDE - addtocart
Project: addtocart
Executing
addtocard
https://www.healthkart.com
Command
Target
Value
1 open /cart/Cart.action
2 set window size 928x1032
3 run script window.scrollTo(0,54)
4 click linkText=Proceed to Pay ₹2,498
5 run script window.scrollTo(0,0)
6 click id=address-fieldset
7 type id=address-fieldset vellore institute of technology katpadi vellore tamil nadu
8 type name=name Sparsh Rajvanshi
9 type name=mobNo 6397048156
10 type css=#pincode-fieldset > .lrge-input-field 632014
11 type name=email sparsh.rajvanshi2021@vitstudent.ac.in
12 click name=landmark
13 type name=landmark vellore institute of technology
Command
Target
Value
Description
Runs: 1 Failures: 1
Log
Reference
Running 'addtocard'
1. open on /cart/Cart.action OK 22:46:43
2. setWindowSize on 928x1032 OK 22:46:44
3. runScript on window.scrollTo(0,54) OK 22:46:44
4. click on linkText=Proceed to Pay ₹2,498 OK 22:46:48
5. runScript on window.scrollTo(0,0) OK 22:46:48
6. Trying to find id=address-fieldset... Failed: Implicit Wait timed out after 30000ms 22:46:49
'addtocard' ended with 1 error(s) 22:47:19
Show all
25°C Cloudy
Search
ENG IN
22:47 23-03-2023

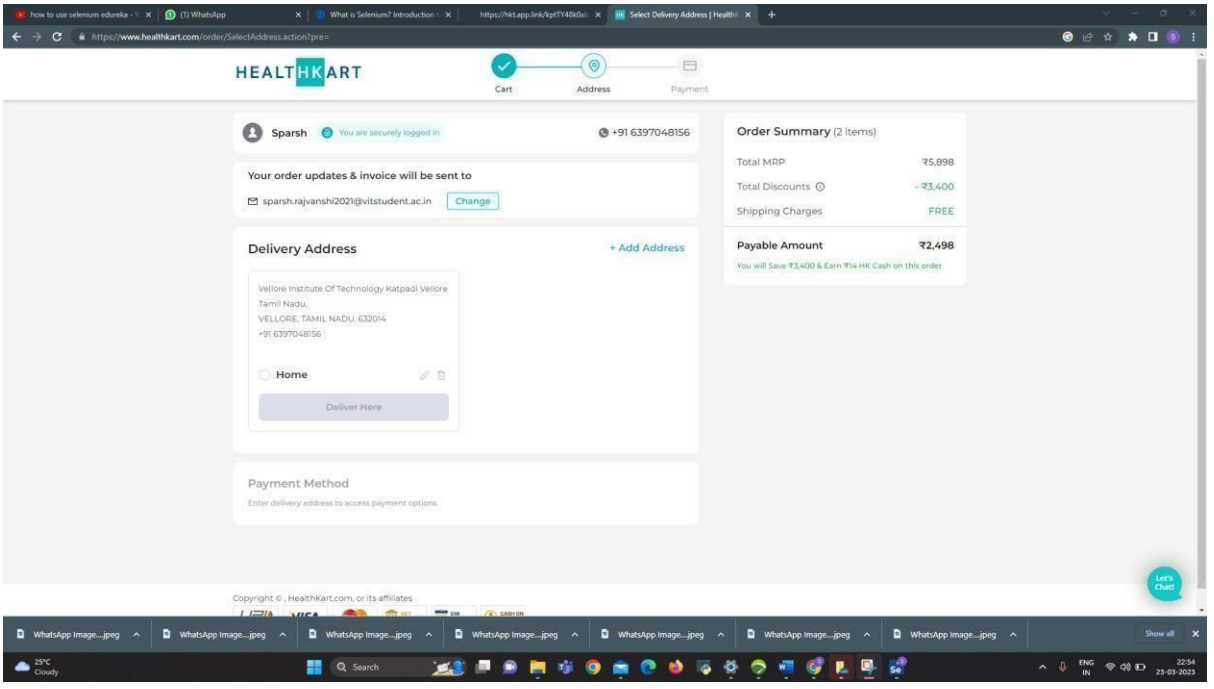


3 delivery address



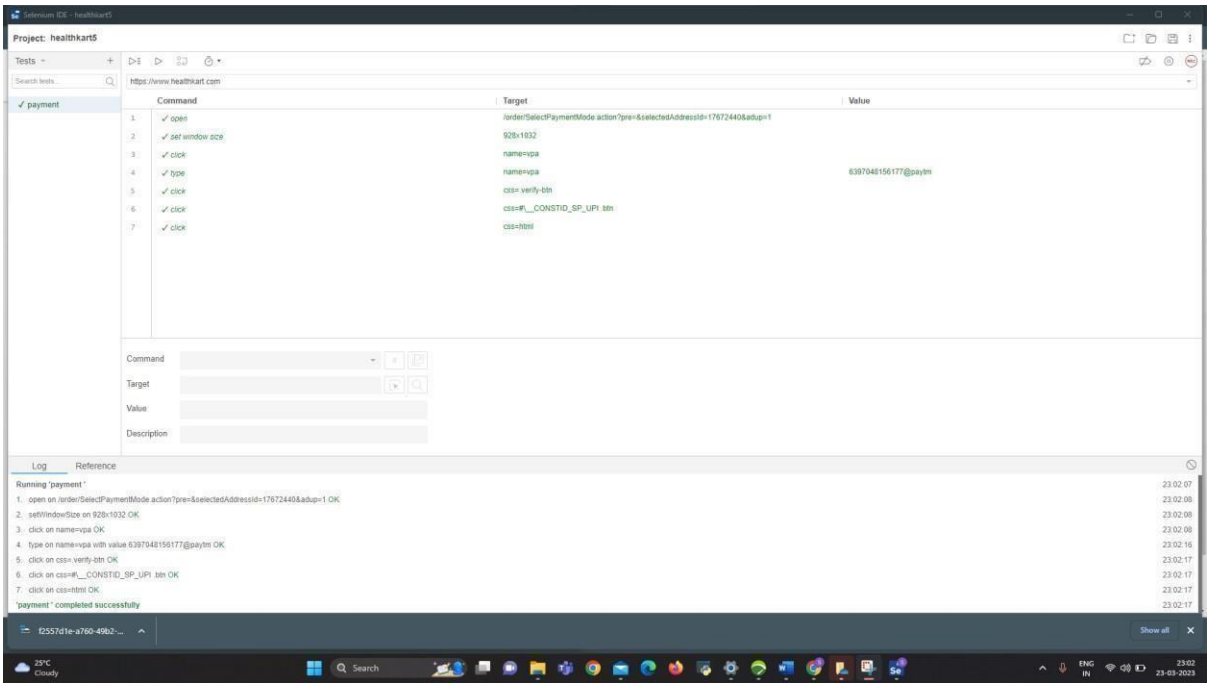


Output

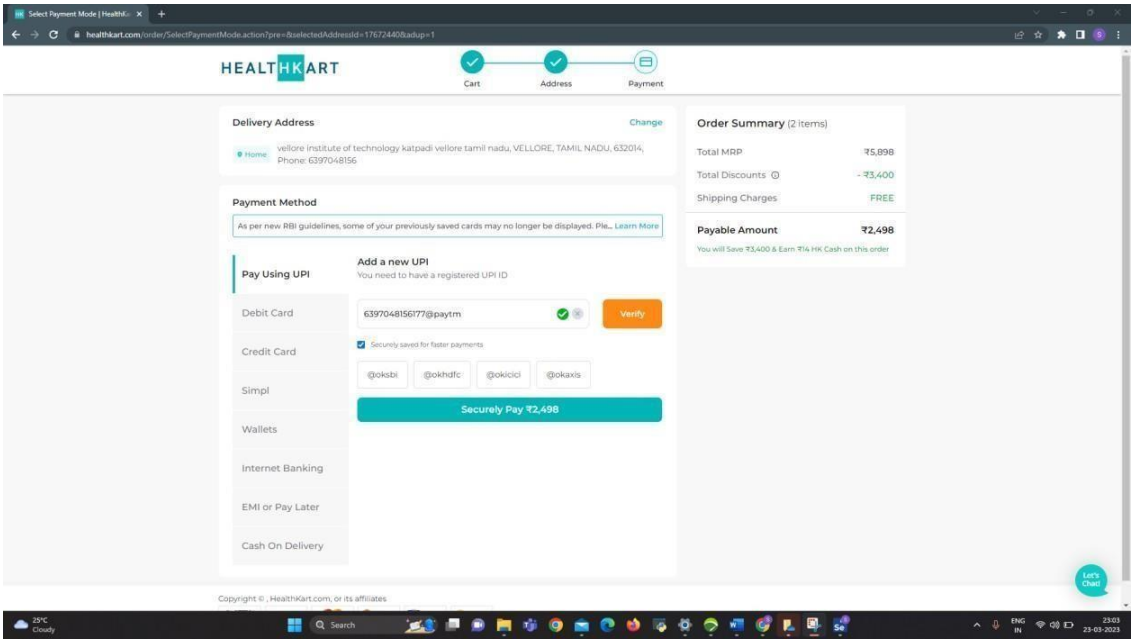


4 – payment

4 – Payment



Output –



JUNIT

JUnit is an open source Unit Testing Framework for JAVA. It is useful for Java Developers to write and run repeatable tests. Erich Gamma and Kent Beck initially develop it. It is an instance of xUnit architecture. As the name implies, it is used for Unit Testing of a small chunk of code.
Developers who are following test-driven methodology must write and execute unit test first before any code. Once you are done with code, you should execute all tests, and it should pass. Every time any code is added, you need to re-execute all test cases and makes sure nothing is broken.

Why you need JUnit testing

- It finds bugs early in the code, which makes our code more reliable.
- JUnit is useful for developers, who work in a test-driven environment.
- Unit testing forces a developer to read code more than writing.
- You develop more readable, reliable and bug-free code which builds confidence during development.

ANNOTATIONS OF JUNIT TESTING

@Test annotation specifies that method is the test method.

@Test(timeout=1000) annotation specifies that method will be failed if it takes longer than 1000 milliseconds (1 second).

@BeforeClass annotation specifies that method will be invoked only once, before starting all the tests.

@Before annotation specifies that method will be invoked before each test.

@After annotation specifies that method will be invoked after each test.

@AfterClass annotation specifies that method will be invoked only once, after finishing all the tests.

Methods of Assert class

1. **void assertEquals(boolean expected,boolean actual)**: checks that two primitives/objects are equal. It is overloaded.
2. **void assertTrue(boolean condition)**: checks that a condition is true.
3. **void assertFalse(boolean condition)**: checks that a condition is false.
4. **void assertNull(Object obj)**: checks that object is null.
5. **void assertNotNull(Object obj)**: checks that object is not null.

Testing Login Module
Java Code (NETBEANS)
TEST 1

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click
nbfs://nbhost/SystemFileSystem/Templates/Project/Maven2/JavaApp/src/main/java/${packagePath}/${main
Clas sName}.java to edit this template
 */

package sprjava.mavenproject2;
```

```
import java.util.Scanner;

/**
 *
 * @author sprpr
 */
public class Mavenproject2 {

      public static void main(String[] args) {
   }

   public static boolean login(String uname,String paswd)
   {
   String
  username="Hello";
  String
  password="123@";

   System.out.println("Log in:");
   System.out.println("username:
   ");

   System.out.println("password:  ");

   //users check = new users(username, password);

   if(username.equals(uname) &&

   password.equals(paswd))
   {
      System.out.println("You are logged in");
      return true;
   }
   else
      System.out.println("entered credentials are wrong
            "); return false ;


  //  if(username.equals(uname) && password.equals(paswd)){

         }
   //}}
```

JUnit Test Case

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to edit this template
```

```
 */
package sprjava.mavenproject2;

import
org.junit.jupiter.api.AfterEach;
import
org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
```

```
import static org.junit.jupiter.api.Assertions.*;

/**
 *
 * @author sprpr
 */
public class Mavenproject2Test {

    public Mavenproject2Test() {
    }

    @BeforeEach
    public void setUp() throws Exception {
    }

    @AfterEach
    public void tearDown() throws Exception {
    }

    /**
     * Test of main method, of class Mavenproject2.
     */
    @Tes
    t
    public void testMain() {

        System.out.println("checkCredentials");
        String un = "Hello";
        String p = "123@";
        Mavenproject2 instance = new Mavenproject2();
        boolean expResult = true;
        boolean result = instance.login(un, p);
        assertEquals(expResult, result);

    }

    /**
     * Test of login method, of class Mavenproject2.
     */

}
```

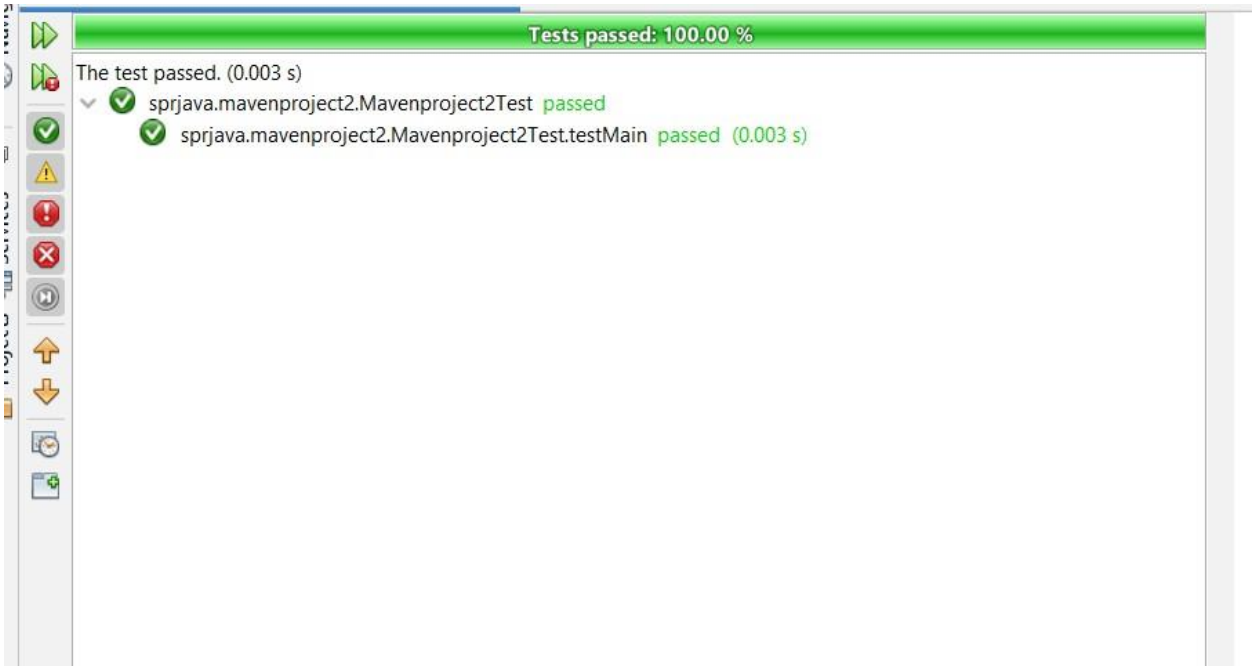
OUTPUT

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running sprjava.mavenproject2.Mavenproject2Test
checkCredentials
Log in:
username:
password:
You are logged in
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.980 s
Finished at: 2023-04-04T14:32:00+05:30
------------------------------------------------------------------------
```


Tests passed: 100.00 %
The test passed. (0.003 s)
sprjava.mavenproject2.Mavenproject2Test passed
sprjava.mavenproject2.Mavenproject2Test.testMain passed (0.003 s)

TEST 2

JUnit Test Case – 2

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to edit this template
 */
package sprjava.mavenproject2;

import
org.junit.jupiter.api.AfterEach;
import
org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

/**
 *
 * @author sprpr
 */
public class Mavenproject2Test {

    public Mavenproject2Test() {
    }

    @BeforeEach
    public void setUp() throws Exception {
    }

    @AfterEach
    public void tearDown() throws Exception {
    }

    /**
     * Test of main method, of class Mavenproject2.
     */
    @Tes
    t
    public void testMain() {

        System.out.println("checkCredentials");
        String un = "Welcome";
        String p = "123@";
        Mavenproject2 instance = new Mavenproject2();
        boolean expResult = true;
        boolean result = instance.login(un, p);
```

```
    assertEquals(expResult, result);
}
```

```
    /**
     * Test of login method, of class Mavenproject2.
     */

}
```

OUTPUT

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running sprjava.mavenproject2.Mavenproject2Test
checkCredentials
Log in:
username:
password:
entered credentials are wrong
Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.013 sec <<< FAILURE!
sprjava.mavenproject2.Mavenproject2Test.testMain()  Time elapsed: 0.012 sec  <<< FAILURE!
```