Ex:No: 7	Introduction to JMeter and Setup JMeter Environment for Testing	

AIM:

To **Introduction to JMeter and Setup JMeter Environment forTesting**

ALGORITHM:

Step 1: Install Java JDK.

Step 2: To check whether it is installed or not type this command "java -version" in command prompt.

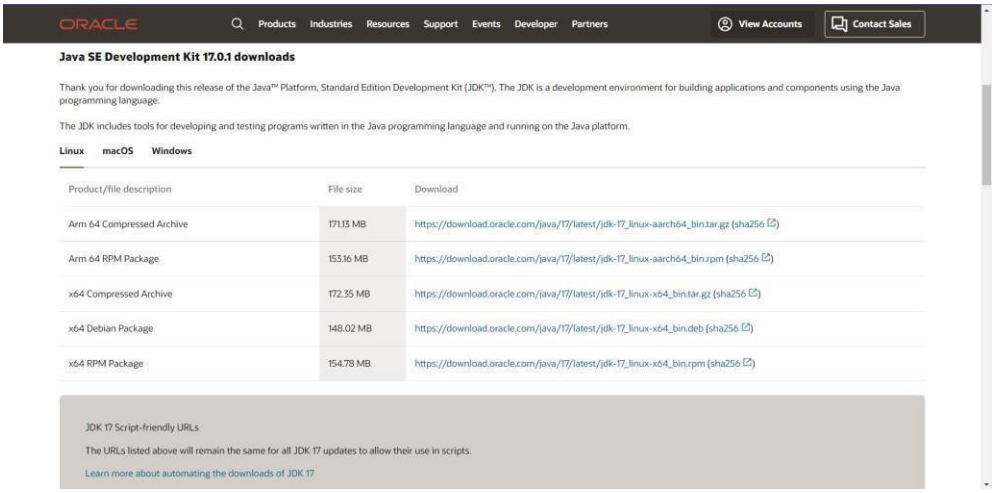
Step 3: Download JMeter.

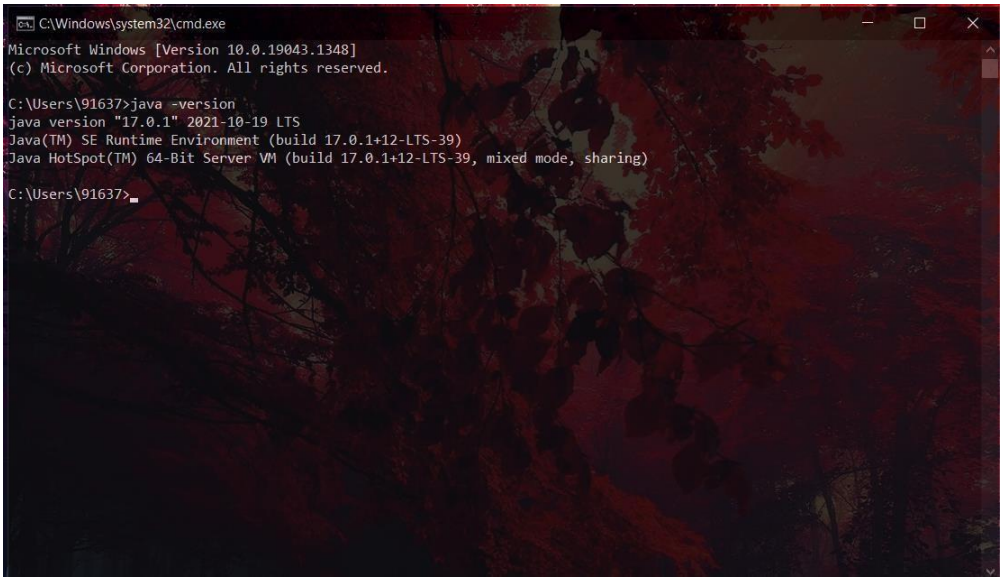
Step 4: Extract the JMeter.

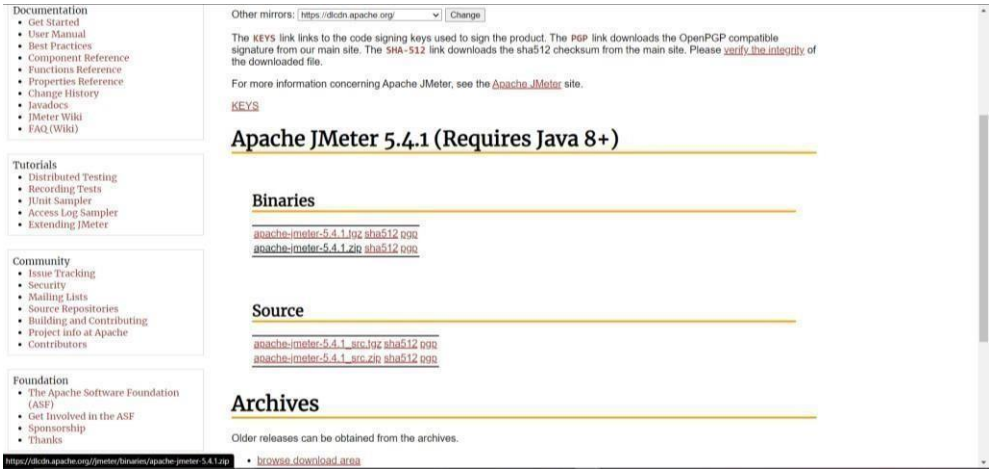
Step 5: Navigate to bin folder of the Apache JMeter folder.

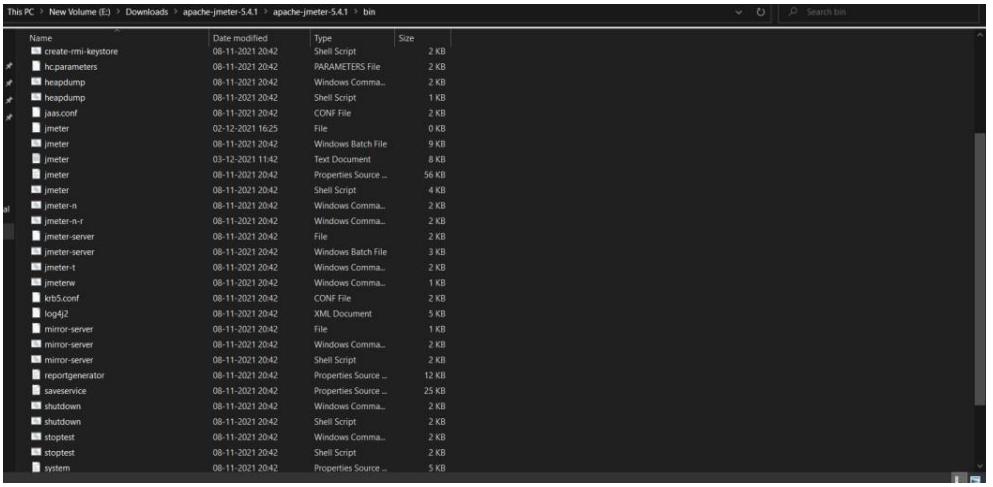
Step 6: Open jmeter.bat file in the bin folder to open JMeter in GUI mode

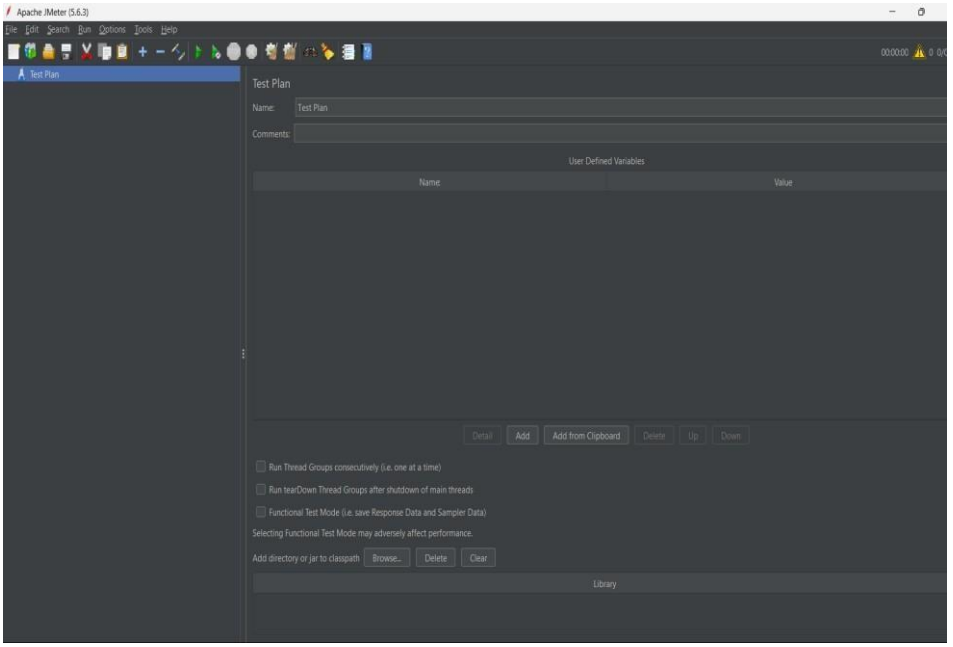
OUTPUT:


ORACLE
Products Industries Resources Support Events Developer Partners
View Accounts
Contact Sales
Java SE Development Kit 17.0.1 downloads
Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.
Linux macOS Windows
Product/file description
File size
Download
Arm 64 Compressed Archive
171.13 MB
https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.tar.gz (sha256)
Arm 64 RPM Package
153.16 MB
https://download.oracle.com/java/17/latest/jdk-17_linux-aarch64_bin.rpm (sha256)
x64 Compressed Archive
172.35 MB
https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz (sha256)
x64 Debian Package
148.02 MB
https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.deb (sha256)
x64 RPM Package
154.78 MB
https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.rpm (sha256)
JDK 17 Script-friendly URLs
The URLs listed above will remain the same for all JDK 17 updates to allow their use in scripts.
Learn more about automating the downloads of JDK 17


C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.
C:\Users\91637>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)
C:\Users\91637>


Documentation
Get Started
User Manual
Best Practices
Component Reference
Functions Reference
Properties Reference
Change History
Javadocs
JMeter Wiki
FAQ (Wiki)
Tutorials
Distributed Testing
Recording Tests
JUnit Sampler
Access Log Sampler
Extending JMeter
Community
Issue Tracking
Security
Mailing Lists
Source Repositories
Building and Contributing
Project info at Apache
Contributors
Foundation
The Apache Software Foundation (ASF)
Get Involved in the ASF
Sponsorship
Thanks
Other mirrors: https://dlcdn.apache.org/ Change
The KEYS link links to the code signing keys used to sign the product. The PGP link downloads the OpenPGP compatible signature from our main site. The SHA-512 link downloads the sha512 checksum from the main site. Please verify the integrity of the downloaded file.
For more information concerning Apache JMeter, see the Apache JMeter site.
KEYS
Apache JMeter 5.4.1 (Requires Java 8+)
Binaries
apache-jmeter-5.4.1.tgz sha512 pgp
apache-jmeter-5.4.1.zip sha512 pgp
Source
apache-jmeter-5.4.1_src.tgz sha512 pgp
apache-jmeter-5.4.1_src.zip sha512 pgp
Archives
Older releases can be obtained from the archives.
browse download area
https://dlcdn.apache.org//jmeter/binaries/apache-jmeter-5.4.1.zip


This PC > New Volume (E:) > Downloads > apache-jmeter-5.4.1 > apache-jmeter-5.4.1 > bin
Search bin
Name
Date modified
Type
Size
create-rmi-keystore 08-11-2021 20:42 Shell Script 2 KB
hc.parameters 08-11-2021 20:42 PARAMETERS File 2 KB
heapdump 08-11-2021 20:42 Windows Comma... 2 KB
heapdump 08-11-2021 20:42 Shell Script 1 KB
jaas.conf 08-11-2021 20:42 CONF File 2 KB
jmeter 02-12-2021 16:25 File 0 KB
jmeter 08-11-2021 20:42 Windows Batch File 9 KB
jmeter 03-12-2021 11:42 Text Document 8 KB
jmeter 08-11-2021 20:42 Properties Source ... 56 KB
jmeter 08-11-2021 20:42 Shell Script 4 KB
jmeter-n 08-11-2021 20:42 Windows Comma... 2 KB
jmeter-n-r 08-11-2021 20:42 Windows Comma... 2 KB
jmeter-server 08-11-2021 20:42 File 2 KB
jmeter-server 08-11-2021 20:42 Windows Batch File 3 KB
jmeter-t 08-11-2021 20:42 Windows Comma... 2 KB
jmeterw 08-11-2021 20:42 Windows Comma... 1 KB
krb5.conf 08-11-2021 20:42 CONF File 2 KB
log4j2 08-11-2021 20:42 XML Document 5 KB
mirror-server 08-11-2021 20:42 File 1 KB
mirror-server 08-11-2021 20:42 Windows Comma... 2 KB
mirror-server 08-11-2021 20:42 Shell Script 2 KB
reportgenerator 08-11-2021 20:42 Properties Source ... 12 KB
saveservice 08-11-2021 20:42 Properties Source ... 25 KB
shutdown 08-11-2021 20:42 Windows Comma... 2 KB
shutdown 08-11-2021 20:42 Shell Script 2 KB
stoptest 08-11-2021 20:42 Windows Comma... 2 KB
stoptest 08-11-2021 20:42 Shell Script 2 KB
system 08-11-2021 20:42 Properties Source ... 5 KB


Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
Test Plan
Name: Test Plan
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

Particulars	Max Marks	Mark Secured
Program and Execution	**15**	
Viva	**10**	
Total	**25**	

RESULT:

Thus, the above JMeter tool has been executed successfully and the output was verified.

Ex:No: 8b	Use JMeter to perform Load Testing Using Youtube	

AIM:

To **Use JMeter to perform Load TestingUsing Facebook website**

ALGORITHM:

Step 1: Open JMeter in GUI mode.

Step 2: First you must create a test plan in the user interface of JMeter.

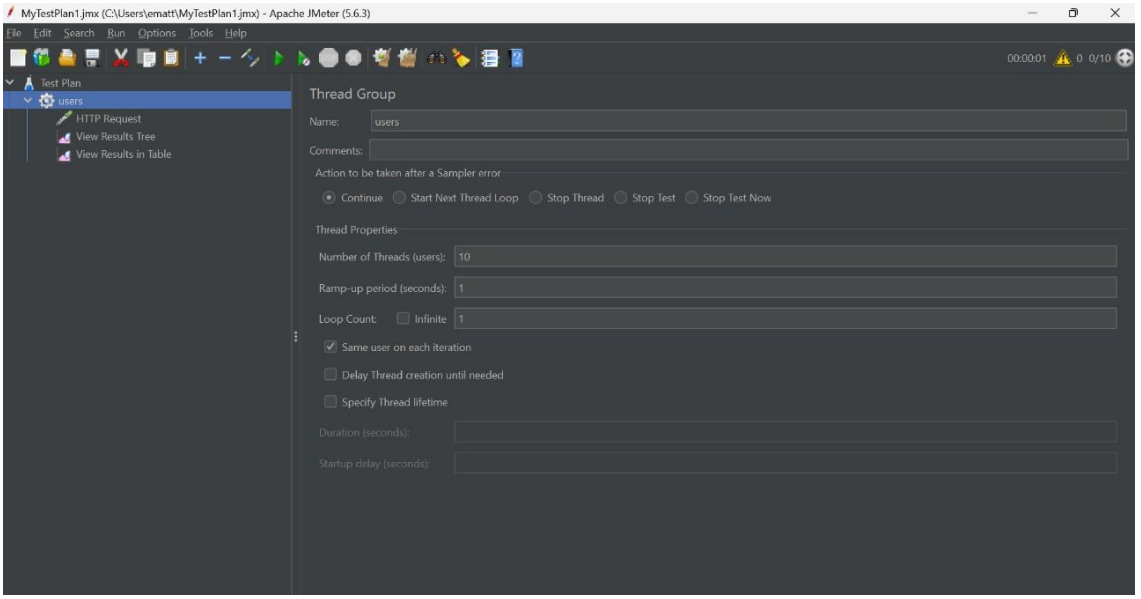
Step 3: Next step is to add thread groups and specify the number of threads or users and loop count.

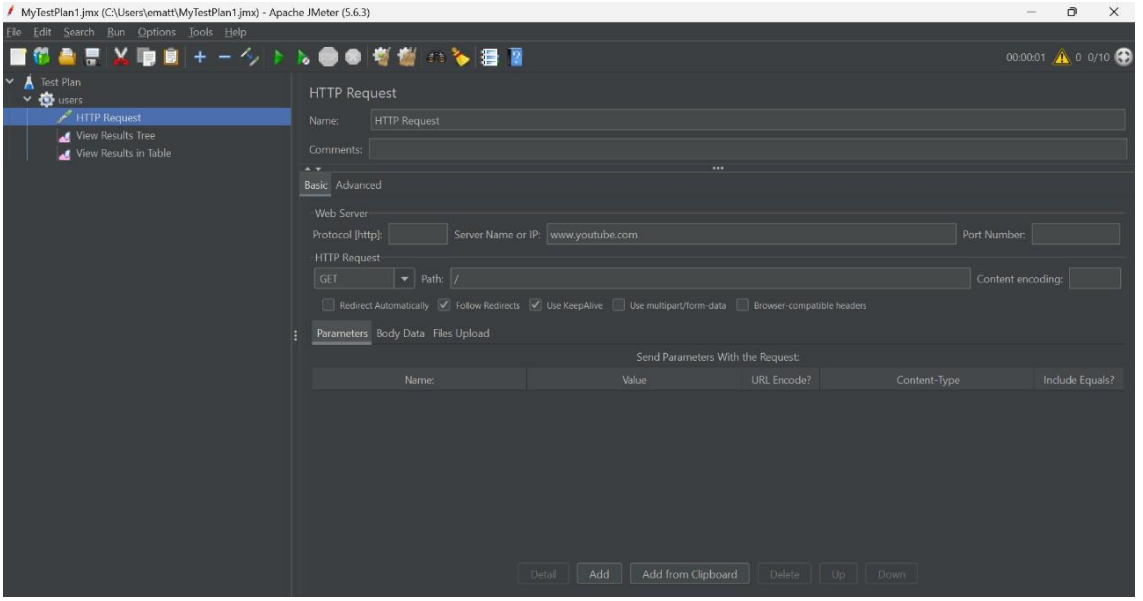
Step 4: Once the thread group is created, next step is to add the HTTP Request and specify the server's name and path. Now save your Test Plan in any folder.

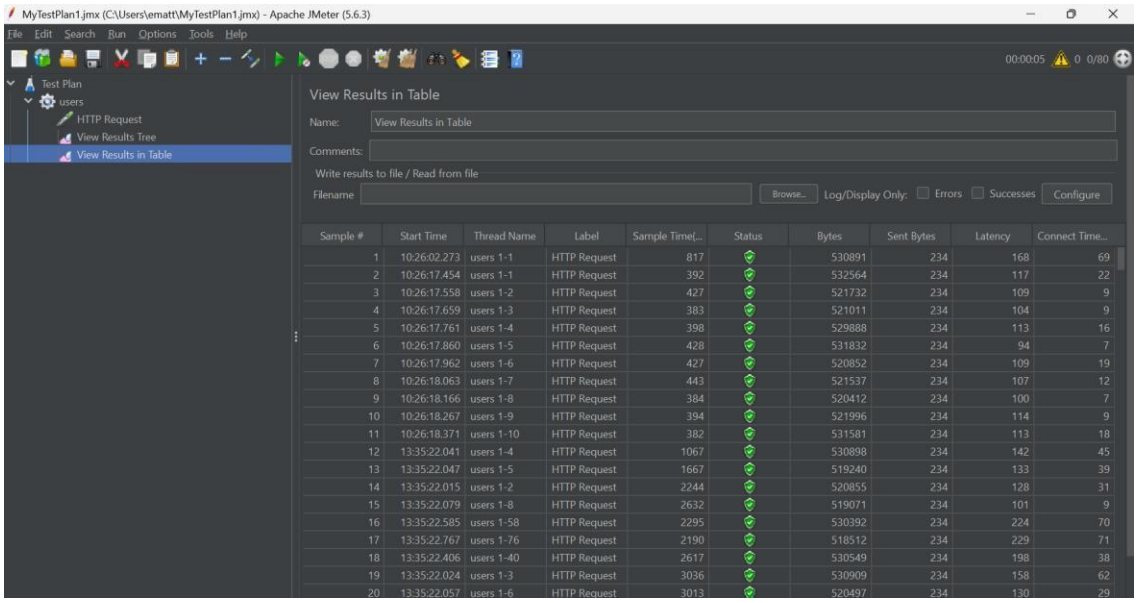
Step 5: Add Table result.

Step 6: Run and get the test result.

OUTPUT:


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:01 0 0/10
Test Plan
users
HTTP Request
View Results Tree
View Results in Table
Thread Group
Name: users
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): 1
Loop Count: Infinite 1
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:01 0 0/10
Test Plan
users
HTTP Request
View Results Tree
View Results in Table
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]: Server Name or IP: www.youtube.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


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:05 0 0/80
Test Plan
users
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(... Status Bytes Sent Bytes Latency Connect Time...
1 10:26:02.273 users 1-1 HTTP Request 817 530891 234 168 69
2 10:26:17.454 users 1-1 HTTP Request 392 532564 234 117 22
3 10:26:17.558 users 1-2 HTTP Request 427 521732 234 109 9
4 10:26:17.659 users 1-3 HTTP Request 383 521011 234 104 9
5 10:26:17.761 users 1-4 HTTP Request 398 529888 234 113 16
6 10:26:17.860 users 1-5 HTTP Request 428 531832 234 94 7
7 10:26:17.962 users 1-6 HTTP Request 427 520852 234 109 19
8 10:26:18.063 users 1-7 HTTP Request 443 521537 234 107 12
9 10:26:18.166 users 1-8 HTTP Request 384 520412 234 100 7
10 10:26:18.267 users 1-9 HTTP Request 394 521996 234 114 9
11 10:26:18.371 users 1-10 HTTP Request 382 531581 234 113 18
12 13:35:22.041 users 1-4 HTTP Request 1067 530898 234 142 45
13 13:35:22.047 users 1-5 HTTP Request 1667 519240 234 133 39
14 13:35:22.015 users 1-2 HTTP Request 2244 520855 234 128 31
15 13:35:22.079 users 1-8 HTTP Request 2632 519071 234 101 9
16 13:35:22.585 users 1-58 HTTP Request 2295 530392 234 224 70
17 13:35:22.767 users 1-76 HTTP Request 2190 518512 234 229 71
18 13:35:22.406 users 1-40 HTTP Request 2617 530549 234 198 38
19 13:35:22.024 users 1-3 HTTP Request 3036 530909 234 158 62
20 13:35:22.057 users 1-6 HTTP Request 3013 520497 234 130 29

Particulars	Max Marks	Mark Secured
Program and Execution	**15**	
Viva	**10**	
Total	**25**	

RESULT:

Thus, the above JMeter tool has been executed successfully and the output was verified.

Ex: No:9b	Use JMeter to perform Stress Testing using Instagram website	

AIM:

To **Use JMeter to perform Stress Testing using Facebook website**

ALGORITHM:

Step 1: Open JMeter in GUI mode.

Step 2: First you must create a test plan in the user interface of JMeter.

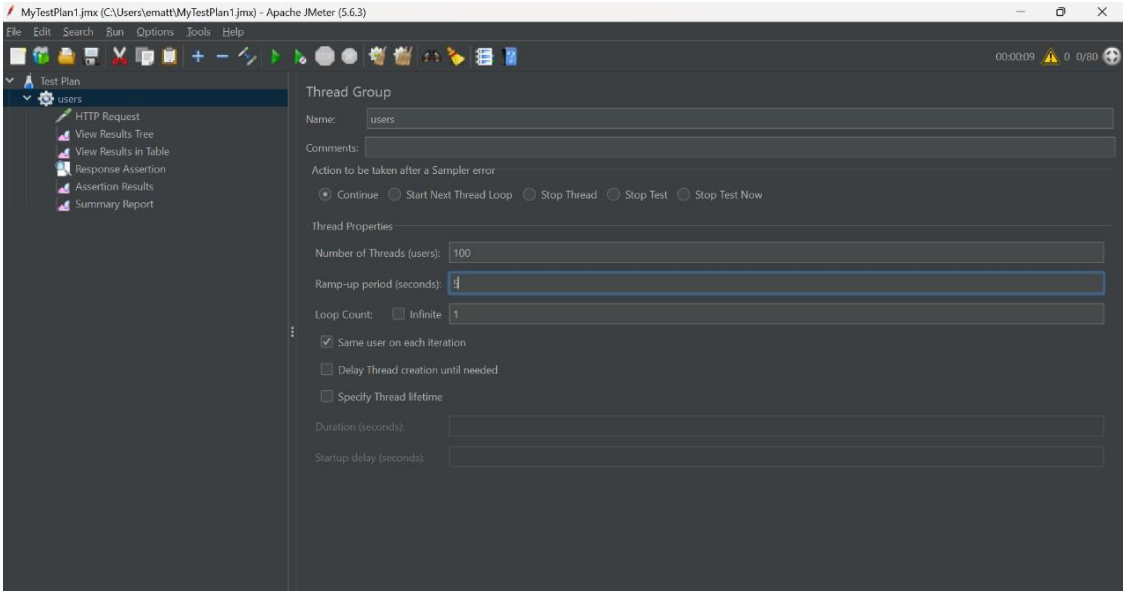
Step 3: Insert a Thread Group and add an HTTP request with the server's name of your website on which you want to perform stress testing.

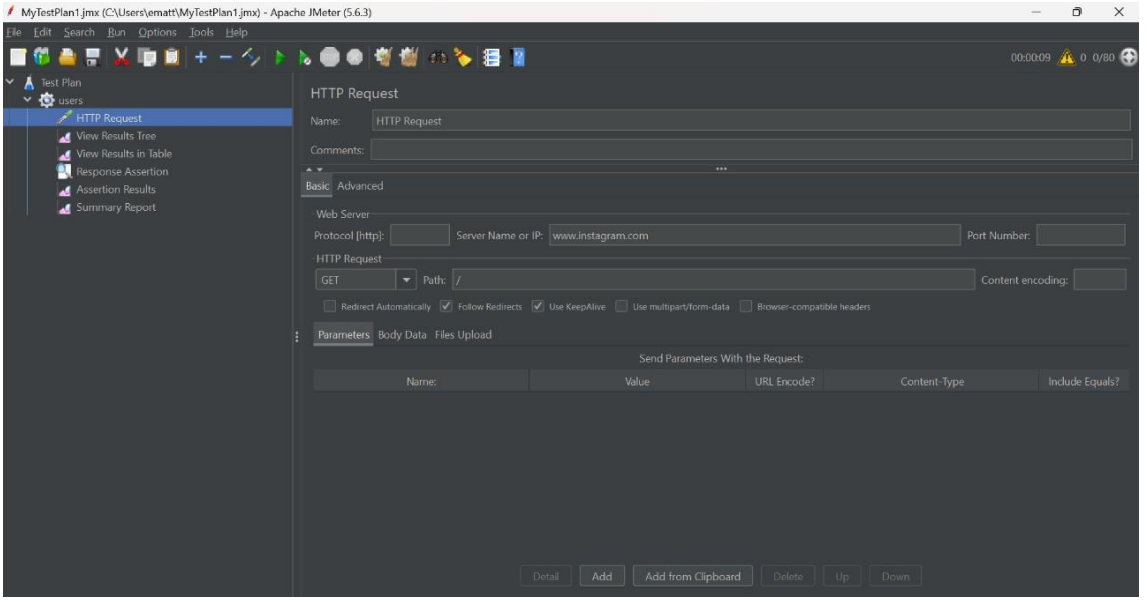
Step 4: Next, you need to add a listener inside the thread group and view the test results. It will show the status of the test that has taken place.

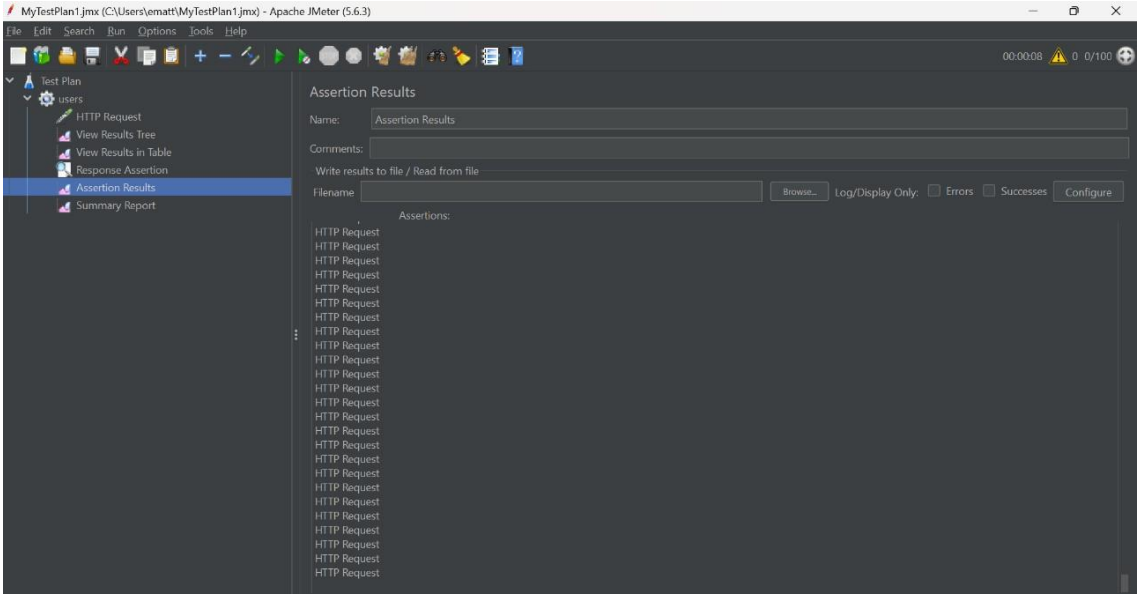
Step 5: Now you need to add a response assertion inside your thread group.

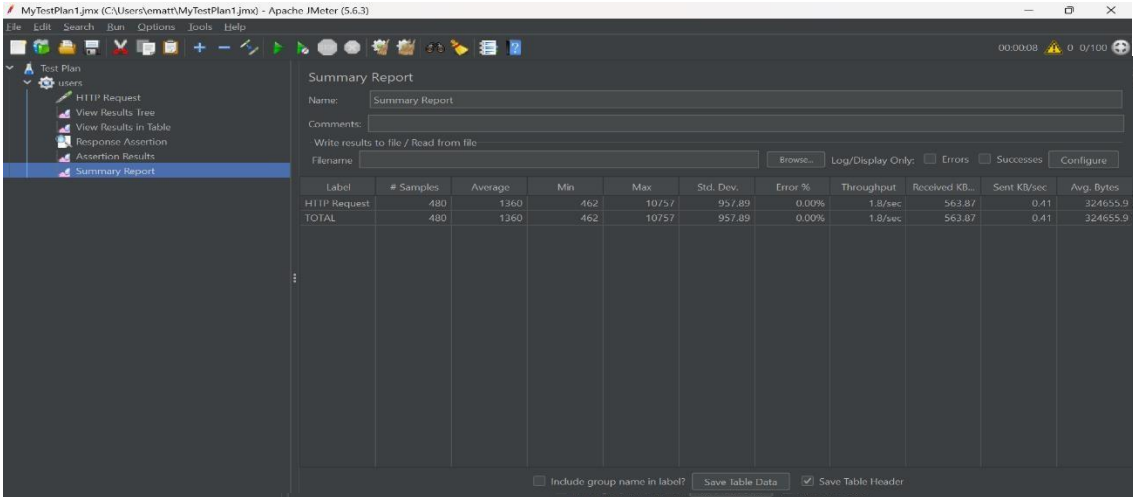
Step 6: Run and get the test result.

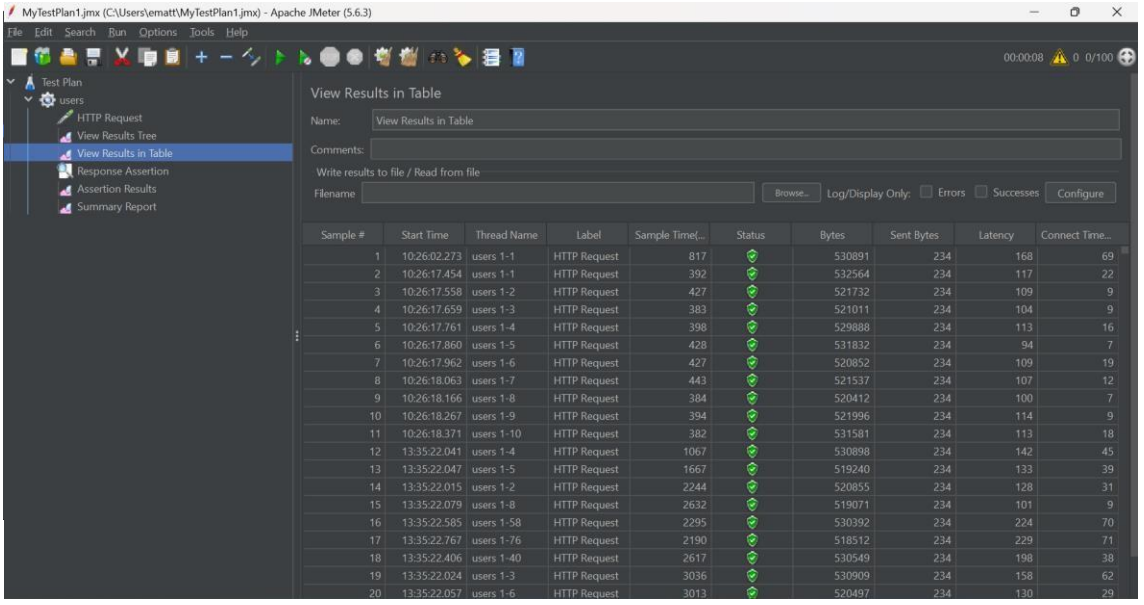
OUTPUT:


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
View Results Tree
View Results in Table
Response Assertion
Assertion Results
Summary Report
Thread Group
Name: users
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): 100
Ramp-up period (seconds): 5
Loop Count: Infinite 1
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
View Results Tree
View Results in Table
Response Assertion
Assertion Results
Summary Report
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: www.instagram.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


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
View Results Tree
View Results in Table
Response Assertion
Assertion Results
Summary Report
Assertion Results
Name: Assertion Results
Comments:
Write results to file / Read from file
Filename
Browse...
Log/Display Only: Errors Successes
Configure
Assertions:
HTTP Request





Particulars	Max Marks	Mark Secured
Program and Execution	15	
Viva	10	
Total	25	

RESULT:

Thus, the above Jmeter tool has been executed successfully and the output was verified

Ex: No:10b	Introduction to Timers in JMeter and Generate a Load using Timers with Instagram website	

AIM:

To **Introduction to Timers in JMeter and Generate a Load using Timers with Facebook website**

ALGORITHM:

Step 1: Open jmeter in GUI mode.

Step 2: First you must create a test plan in the user interface of JMeter.

Step 3: Next step is to add thread groups and specify the number of threads or users and loop count.

Step 4: Once the thread group is created, next step is to add the HTTP Request and specify the server's name and path. Now save your Test Plan in any folder.

Step 5: Add Constant Timer.

Step 6: Specify the constant time.

Step 7: Add Table result.

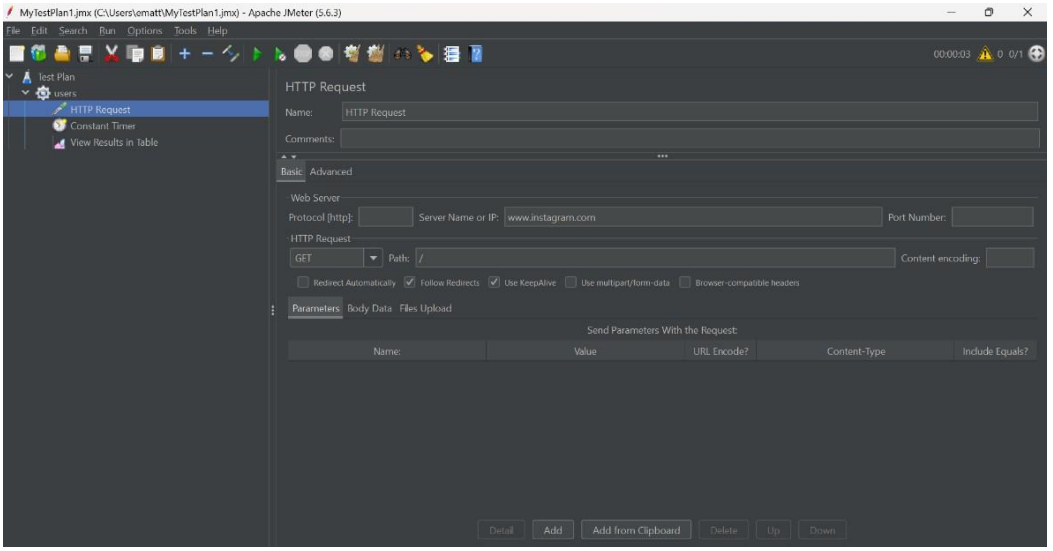
Step 8: Run and get the test result.

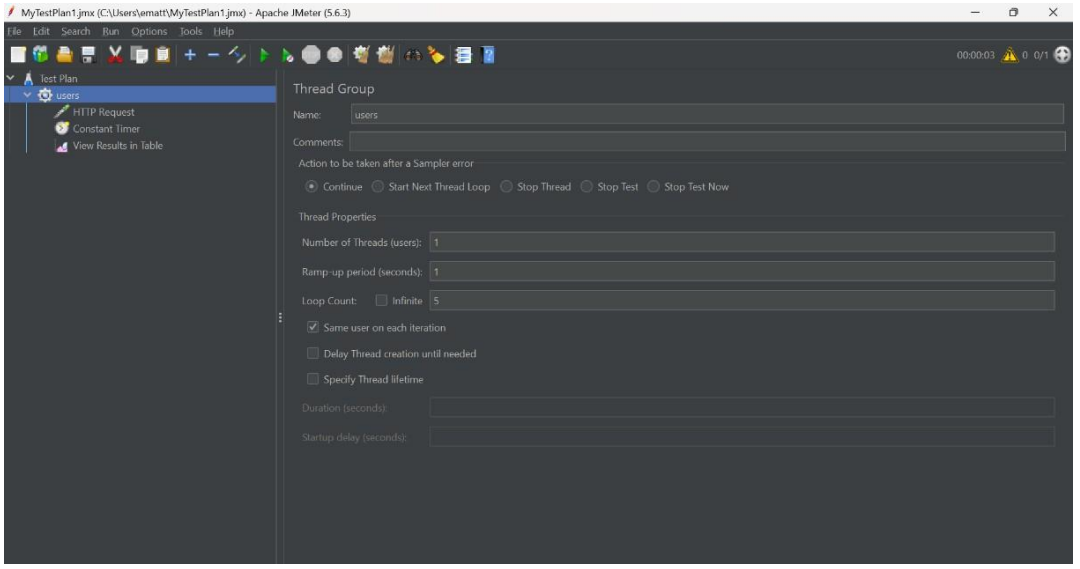
Particulars	Max Marks	Mark Secured
Program and Execution	**15**	
Viva	**10**	
Total	**25**	

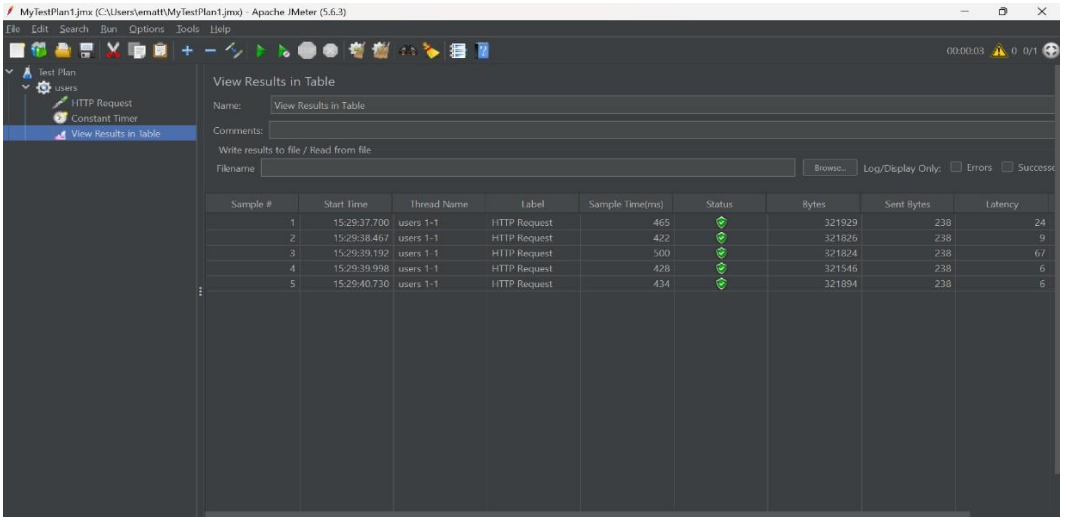
RESULT:

Thus, the above Jmeter tool has been executed successfully and the output was verified.

OUTPUT:


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
Constant Timer
View Results in Table
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: www.instagram.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


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
Constant Timer
View Results in Table
Thread Group
Name: users
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): 1
Ramp-up period (seconds): 1
Loop Count:
Infinite
5
Same user on each iteration
Delay Thread creation until needed
Specify Thread lifetime
Duration (seconds):
Startup delay (seconds):


MyTestPlan1.jmx (C:\Users\ematt\MyTestPlan1.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
Test Plan
users
HTTP Request
Constant Timer
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
Success
Sample #
Start Time
Thread Name
Label
Sample Time(ms)
Status
Bytes
Sent Bytes
Latency
1 15:29:37.700 users 1-1 HTTP Request 465 321929 238 24
2 15:29:38.467 users 1-1 HTTP Request 422 321826 238 9
3 15:29:39.192 users 1-1 HTTP Request 500 321824 238 67
4 15:29:39.998 users 1-1 HTTP Request 428 321546 238 6
5 15:29:40.730 users 1-1 HTTP Request 434 321894 238 6

Ex: No:11 B	Introduction to JMeter Response Assertion and Assert Response from Web Page	

AIM:

To **Introduction to JMeter Response Assertion and Assert Response from Web Page**

ALGORITHM:

Step 1: Open jmeter in GUI mode.

Step 2: First you must create a test plan in the user interface of JMeter.

Step 3: Next step is to add thread groups and specify the number of threads or users and loop count.

Step 4: Once the thread group is created, next step is to add the HTTP Request and specify the server'sname and path. Now save your Test Plan in any folder.

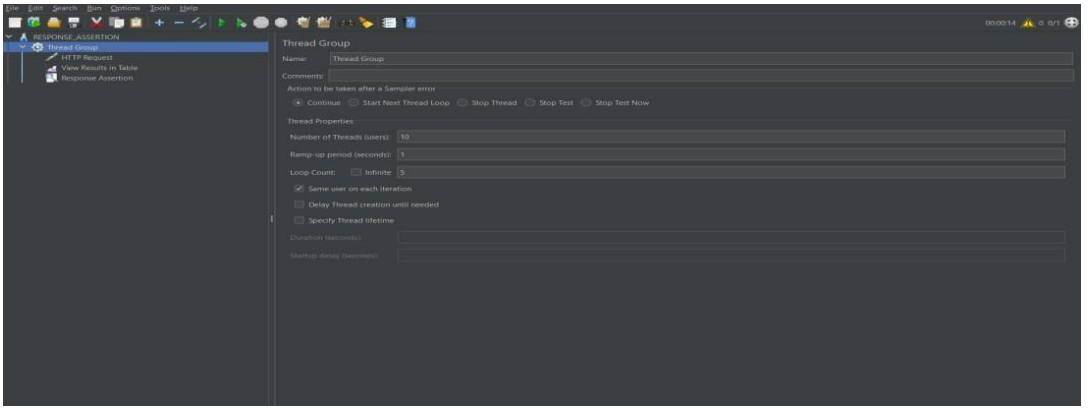
Step 5: Next, you need to add a listener inside the thread group and view the test results. It will show the status of the test that has taken place.

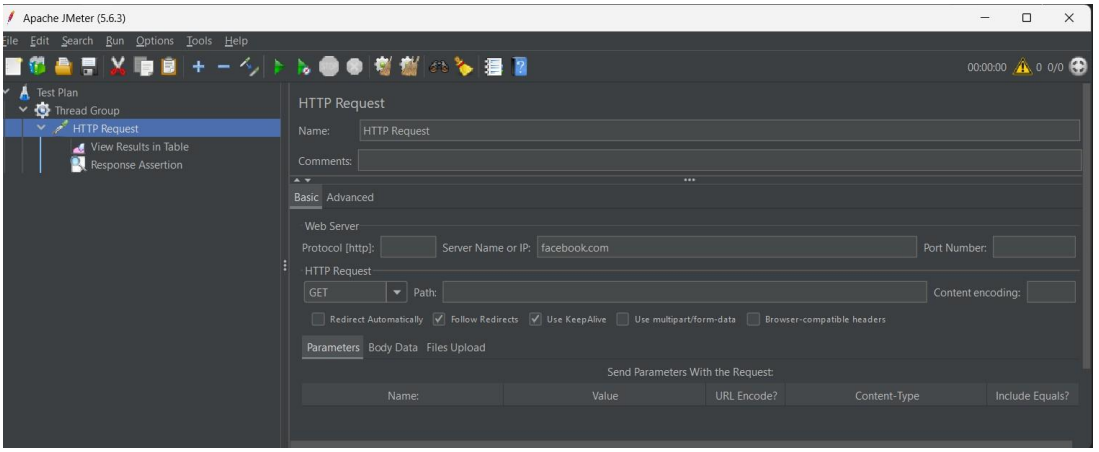
Step 6: Now you need to add a response assertion inside your thread group.

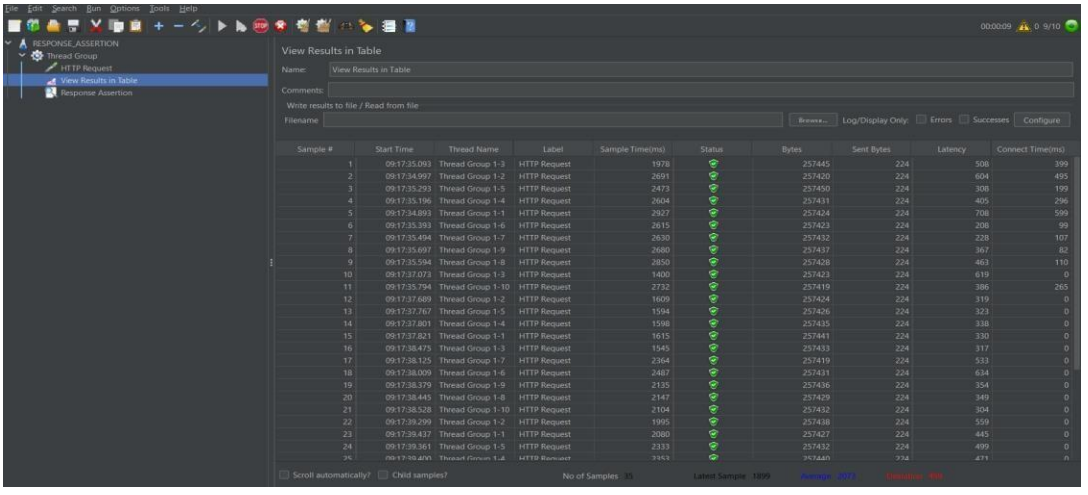
Step 7: Add Tree result.

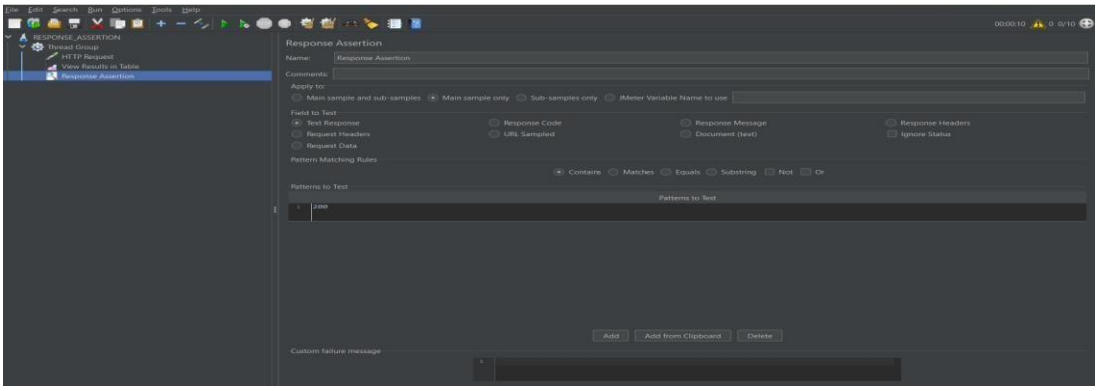
Step 8: Run and get the test result.

OUTPUT:




Apache JMeter (5.6.3)
Test Plan
Thread Group
HTTP Request
View Results in Table
Response Assertion
HTTP Request
Name: HTTP Request
Comments:
Basic Advanced
Web Server
Protocol [http]:
Server Name or IP: facebook.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?


View Results in Table
Name: View Results in Table
Comments:
Write results to file / Read from file
Filename
Sample #
Start Time
Thread Name
Label
Sample Time(ms)
Status
Bytes
Sent Bytes
Latency
Connect Time(ms)


Response Assertion
Name: Response Assertion
Comments:
Apply to:
Field to Test
Pattern Matching Rules
Patterns to Test
Add
Add from Clipboard
Delete
Custom failure message

Particulars	Max Marks	Mark Secured
Program and Execution	**15**	
Viva	**10**	
Total	**25**	

RESULT:

Thus, the above Jmeter tool has been executed successfully and the output was verified.

EX.NO		TEST AND PROVIDE TEST REPORTS FOR THE GIVEN API USING POSTMAN
DATE		

AIM:

To test and provide test reports for the given API using POSTMAN

PROCEDURE:

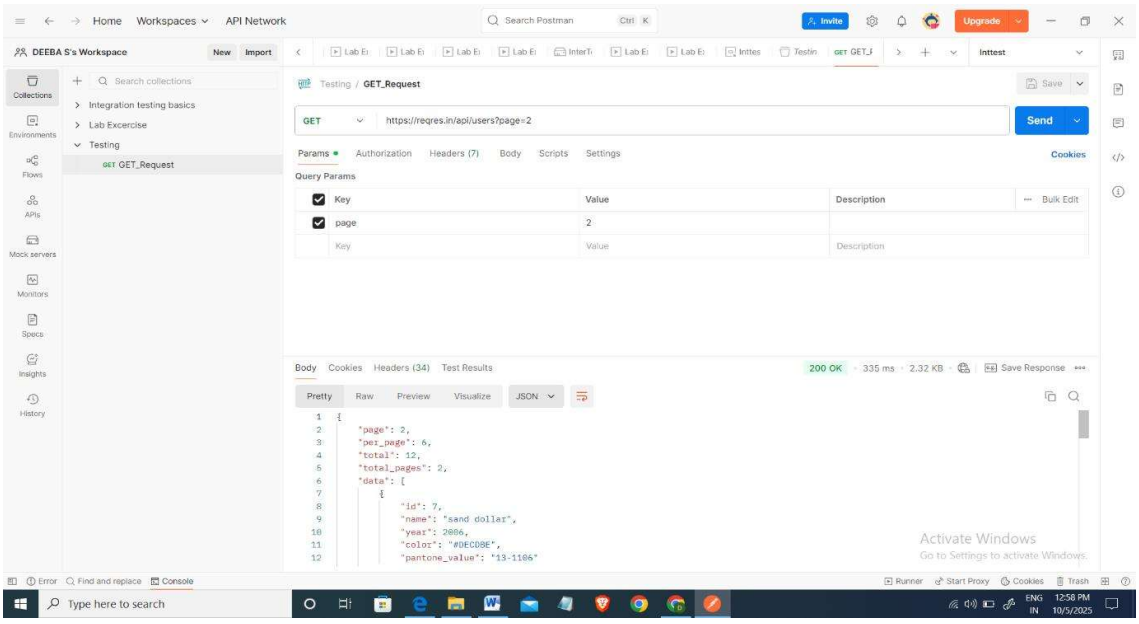
Testing GET, POST, PUT, DELETE Request

Get Request:

Create a New Request

1. Click + **New** → **HTTP Request**
2. Choose the request type as GET.
3. Enter your API URL. https://reqres.in/api/users?page=2
4. Add this header to your API requests: x-api-key: reqres-free-v1
5. Click Send button.
6. Status code is 200. Message is "**OK".**

OUTPUT:



Post Request:

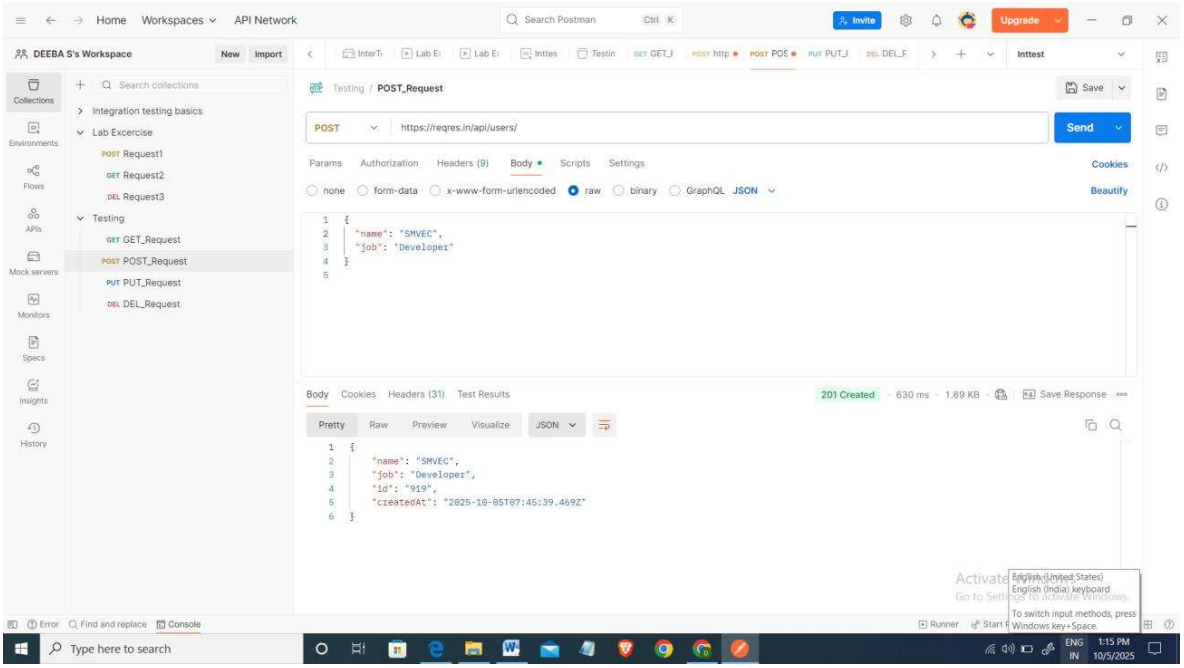
1. Click + **New** → **HTTP Request**
2. Choose the request type as POST.
3. Enter your API URL. https://reqres.in/api/users/
4. Add this header to your API requests: x-api-key: reqres-free-v1
5. In the body part select raw option and JSON and add the following code.

```
{

    "name": "SMVEC",

     "job": "Developer"

}
```

6. Click Send button.

7. Status code is 201. Message is “Created”

OUTPUT:



Put Request:

1. Click + **New** → **HTTP Request**

2. Choose the request type as PUT.

3. Enter your API URL. https://reqres.in/api/users/2

4. Add this header to your API requests: x-api-key: reqres-free-v1

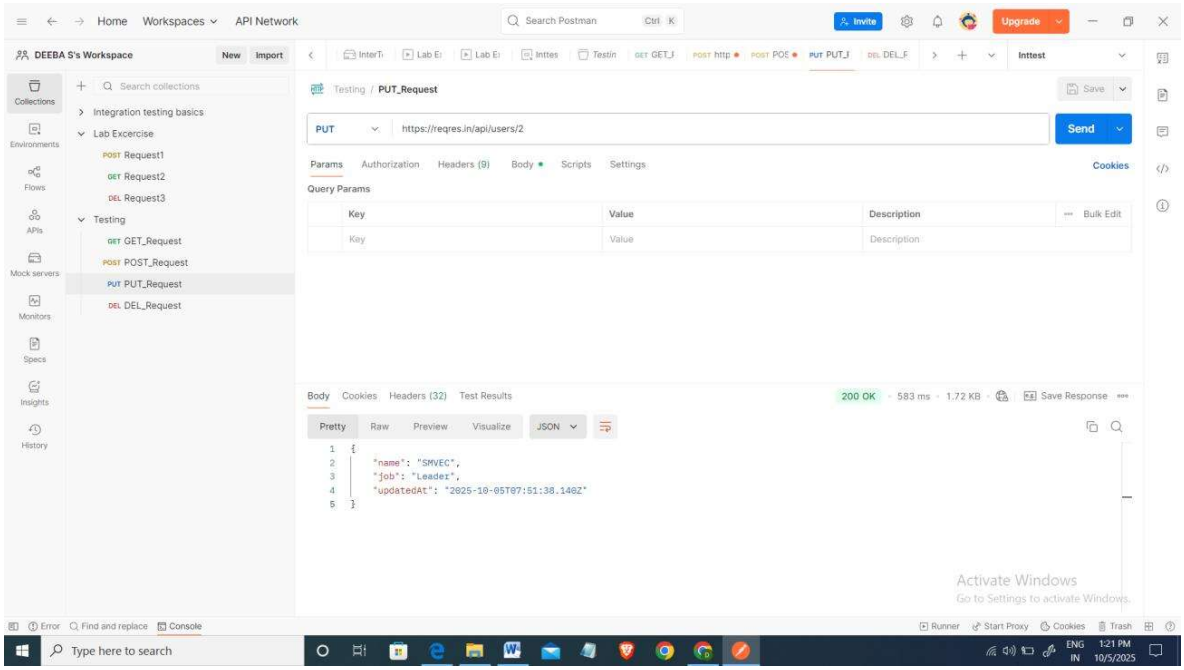
5. In the body part select raw option and JSON and add the following code.

```
{

    "name": "SMVEC",

    "job": "Leader"}
```

6. Click Send button.

7. Status code is 200. Message is “OK”

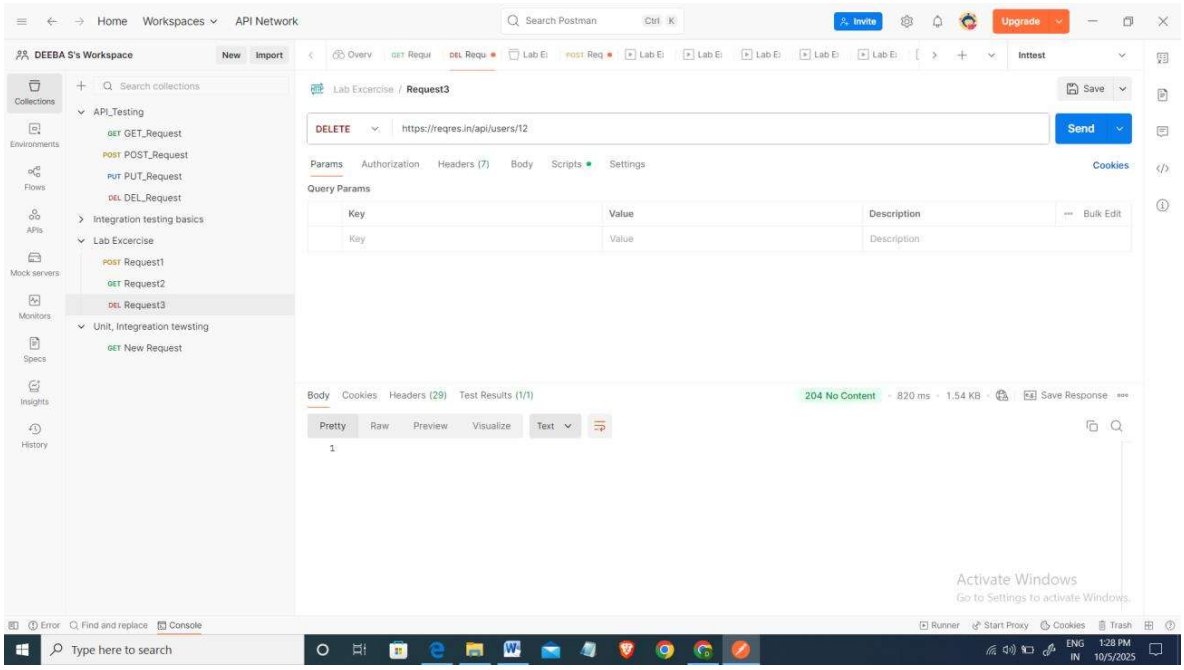
OUTPUT:



Delete Request:

1. Click + **New** → **HTTP Request**
2. Choose the request type as Delete.
3. Enter your API URL. https://reqres.in/api/users/12
4. Add this header to your API requests: x-api-key: reqres-free-v1
5. Click Send button.
6. Status code is 200. Message is “OK”.

OUTPUT:



PARTICULARS	MAX MARKS	MARKS SECURED
PROGRAM AND EXECUTION	15	
VIVA	10	
TOTAL MARKS	25	

RESULT:

Thus, to test and provide test reports for the given API using POSTMAN was executed successfully.

EX.NO		TEST AND PROVIDE TEST REPORTS FOR THE UNIT TEST USING POSTMAN
DATE		

AIM:

Test and provide test reports for the unit test using POSTMAN.

PROCEDURE:

Unit Testing:

1. Click + **New** → **HTTP Request**
2. Choose the request type as POST.
3. Enter your API URL. https://reqres.in/api/register
4. Add this header to your API requests: x-api-key: reqres-free-v1
5. In the body part select raw option and JSON and add the following code.

```
{
    "email": "eve.holt@reqres.in",
    "password": "pistol"
}
```

6. Click Scripts->Post response add the following code

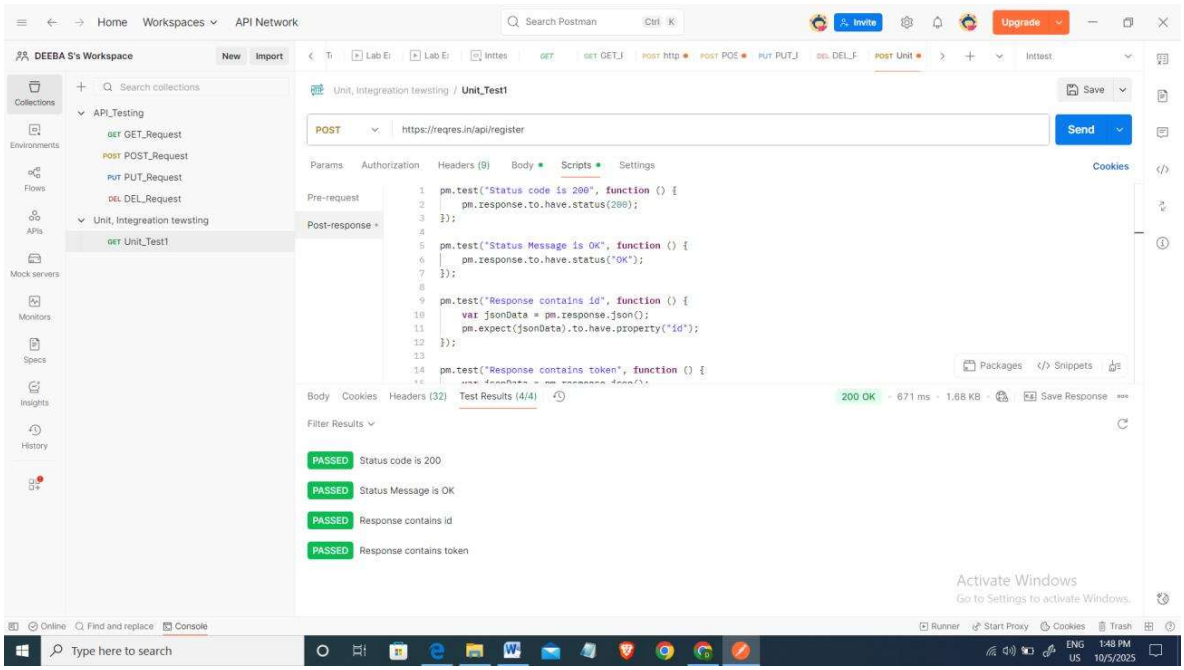
```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
    });


    pm.test("Status Message is OK", function () {
    pm.response.to.have.status("OK");
    });
    pm.test("Response contains id", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData).to.have.property("id");
    });
    pm.test("Response contains token", function () {
     var jsonData = pm.response.json();
     pm.expect(jsonData).to.have.property("token");
});
```

7. Click Send button.

8. Status code is 200. Message is “OK”

9. You can view the test results.

OUTPUT:



PARTICULARS	MAX MARKS	MARKS SECURED
PROGRAM AND EXECUTION	15	
VIVA	10	
TOTAL MARKS	25	

RESULT:

Thus, to test and provide test reports for the unit test using POSTMAN was executed successfully.

EX.NO		PERFORM UNIT TESTING WITH DIFFERENT ORDER OF EXECUTION
DATE		

AIM:

To perform Unit testing with different order of execution.

PROCEDURE:

Unit Test:

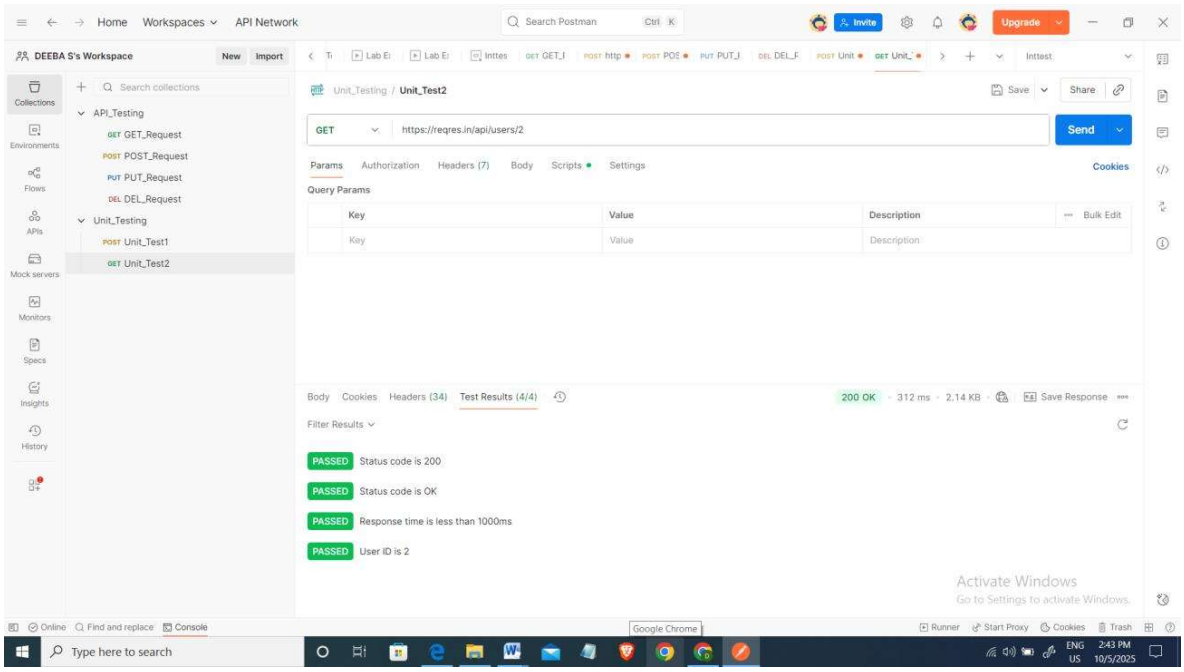
1. Click + **New → HTTP Request**
2. Choose the request type as GET.
3. Enter your API URL. https://reqres.in/api/users/2
4. Add this header to your API requests: x-api-key: reqres-free-v1
5. Click Scripts->Post response add the following code

```
pm.test("Status code is 200", function () {

    pm.response.to.have.status(200);

    });



    pm.test("Status code is OK", function () {
```

```
    pm.response.to.have.status("OK");

    });



    pm.test("Response time is less than 1000ms", function () {

    pm.expect(pm.response.responseTime).to.be.below(1000);

    });



    pm.test("User ID is 2", function () {

    var jsonData = pm.response.json();

    pm.expect(jsonData.data.id).to.eql(2);

});
```

6. Click Send button.
7. Status code is 200. Message is “**OK”.**

OUTPUT:


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
DEEBA S's Workspace
New
Import
Collections
Environments
Flows
APIs
Mock servers
Monitors
Specs
Insights
History
Search collections
API_Testing
GET GET_Request
POST POST_Request
PUT PUT_Request
DEL DEL_Request
Unit_Testing
POST Unit_Test1
GET Unit_Test2
Unit_Testing / Unit_Test2
Save
Share
GET
https://reqres.in/api/users/2
Send
Params
Authorization
Headers (7)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (34)
Test Results (4/4)
200 OK
312 ms
2.14 KB
Save Response
Filter Results
PASSED Status code is 200
PASSED Status code is OK
PASSED Response time is less than 1000ms
PASSED User ID is 2
Activate Windows
Go to Settings to activate Windows.
Online
Find and replace
Console
Runner
Start Proxy
Cookies
Trash
Google Chrome
Type here to search
ENG
US
2:43 PM
10/5/2025

PARTICULARS	MAX MARKS	MARKS SECURED
PROGRAM AND EXECUTION	15	
VIVA	10	
TOTAL MARKS	25	

RESULT:

Thus, to perform Unit testing with different order of execution executed successfully.