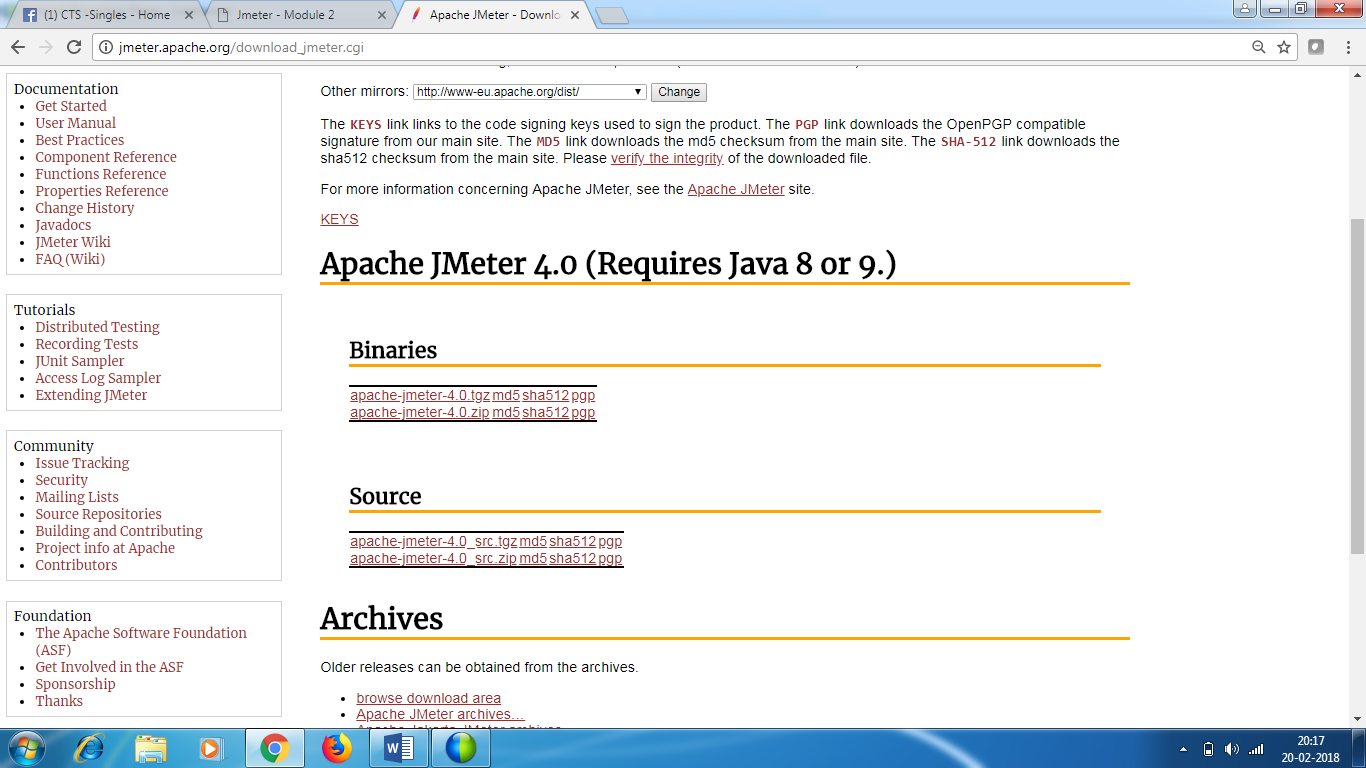
J meter:

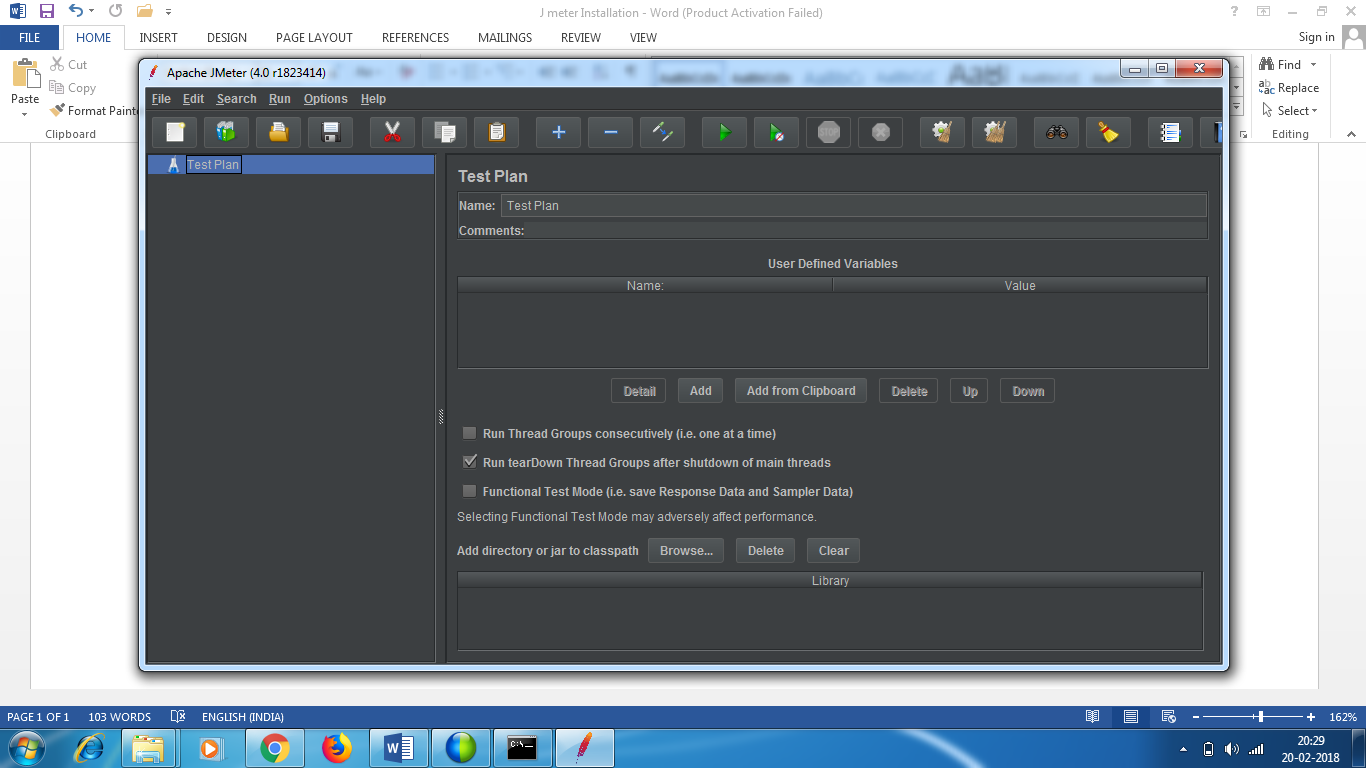
* Open sourced tool
* Java based application used to do performance testing
* It was meant to do website testing but now it has upgraded.
* Webservices , database – apache jmeter

**Install jmeter**

* Click on this link : <http://jmeter.apache.org/>
* Click download releases
* Binary – u can launch jmeter gui –
* Source – will give access to code /source code



* Click on jmeter 4.0 zip
* Tgz – for unix
* You must have java jdk installed on your machine.
* Unzip the zip file and open the bin folder
* Click on jmeter windows batch file
* Click run
* Jmeter is installed.



Configuring class path (Optional step)

* Mostly you will not require this step
* If the script doesn’t get generated , then you can do this step.
* Right click on my computer - > properties -> advanced ->Envt variables

**System Variables** - > In the path , append with entering ; jre bin path at last.

* You need to create one more user variable

JAVA\_HOME

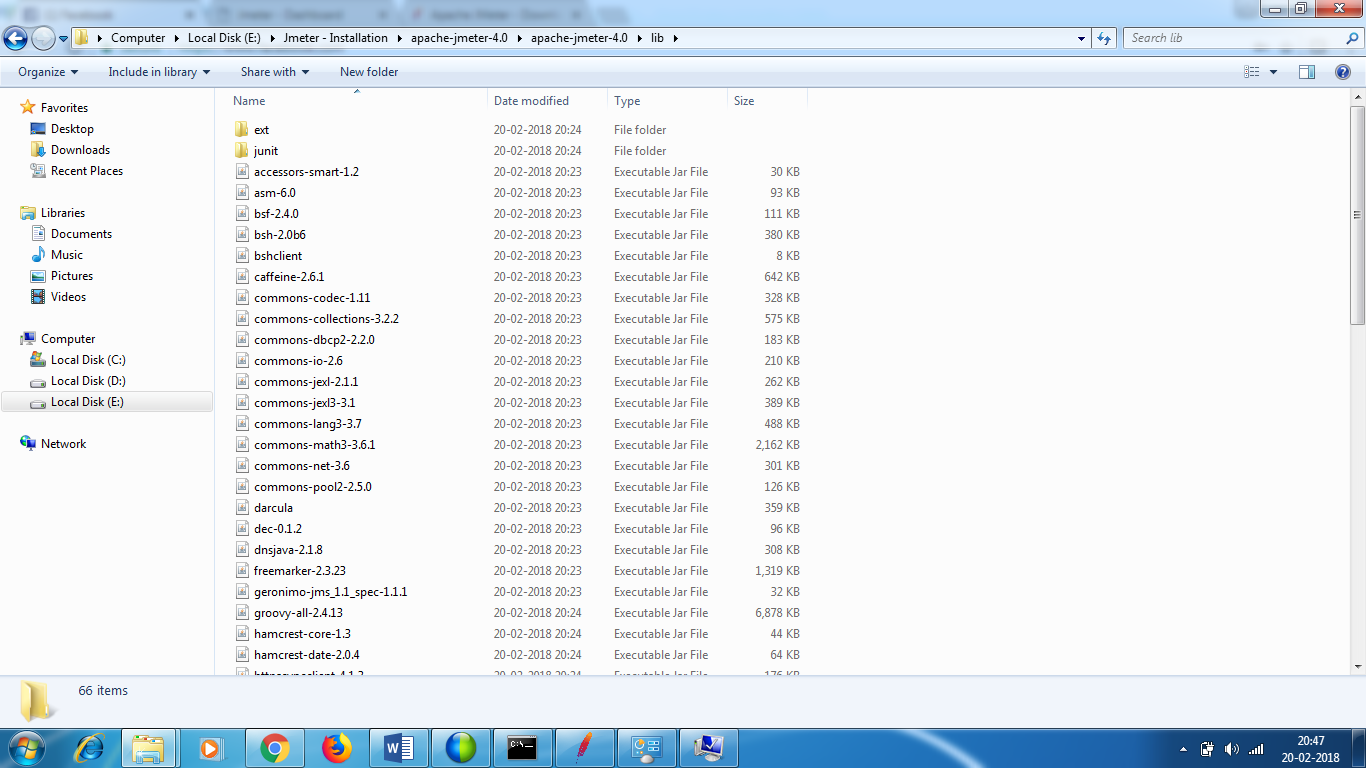
C:\Program Files\Java\jdk1.8.0\_151\jre

* In jemeter folder ,

E:\Jmeter - Installation\apache-jmeter-4.0\apache-jmeter-4.0\lib

You will see lot of executable jar files

* Jmeter will look for class files that requires to work in lib folder or extension folder



Test Plan – container to hold your script

Workbench – store house – temporary folder – whatever you do , you store it in workbench and whatever steps are requiredfor test plan , you can move to test plan folder.

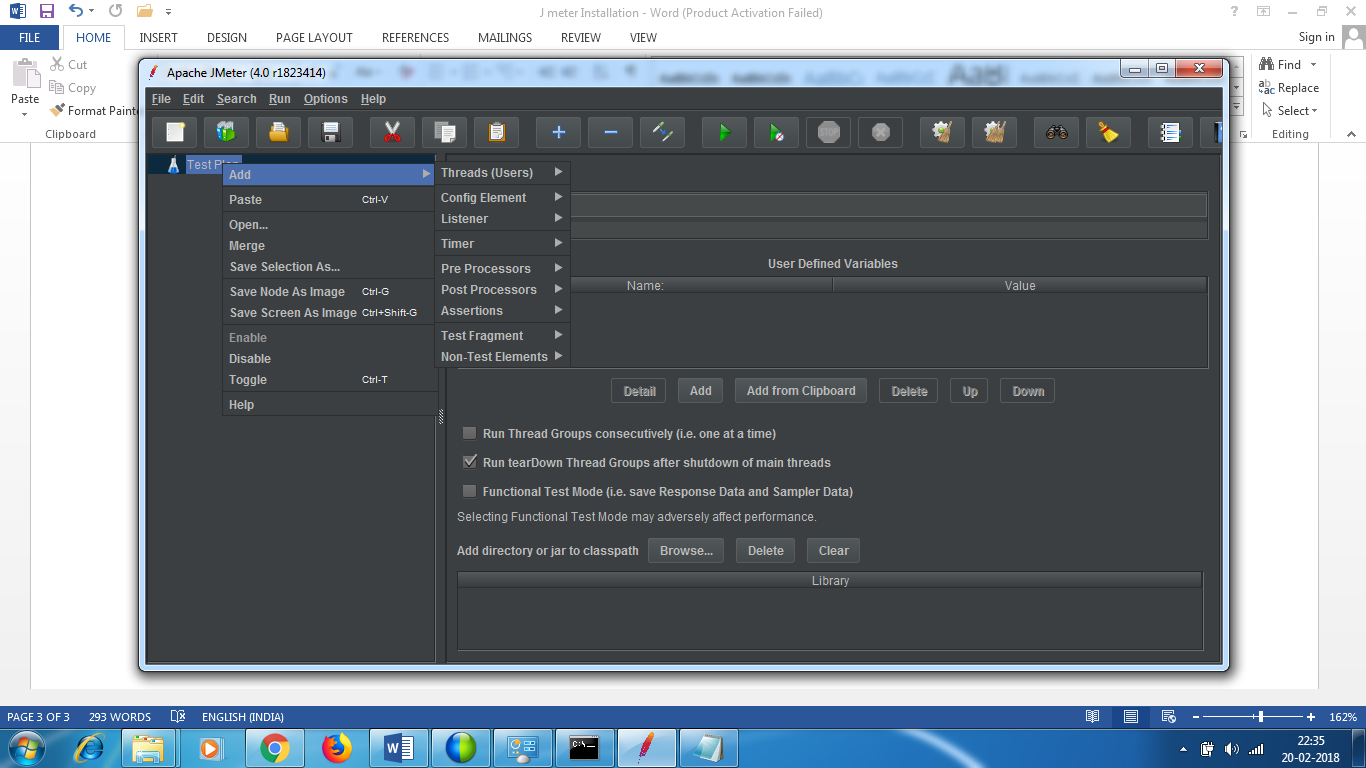
Workbench is not available for jemeter 4.

OS jmeter supports – unix , linux and windows

JDBC supports HTTP req , FTP req , Soap /xml req , jdbc , ldap request

Test elements

Rt click on Test Plan - > Add - > You see various test elements

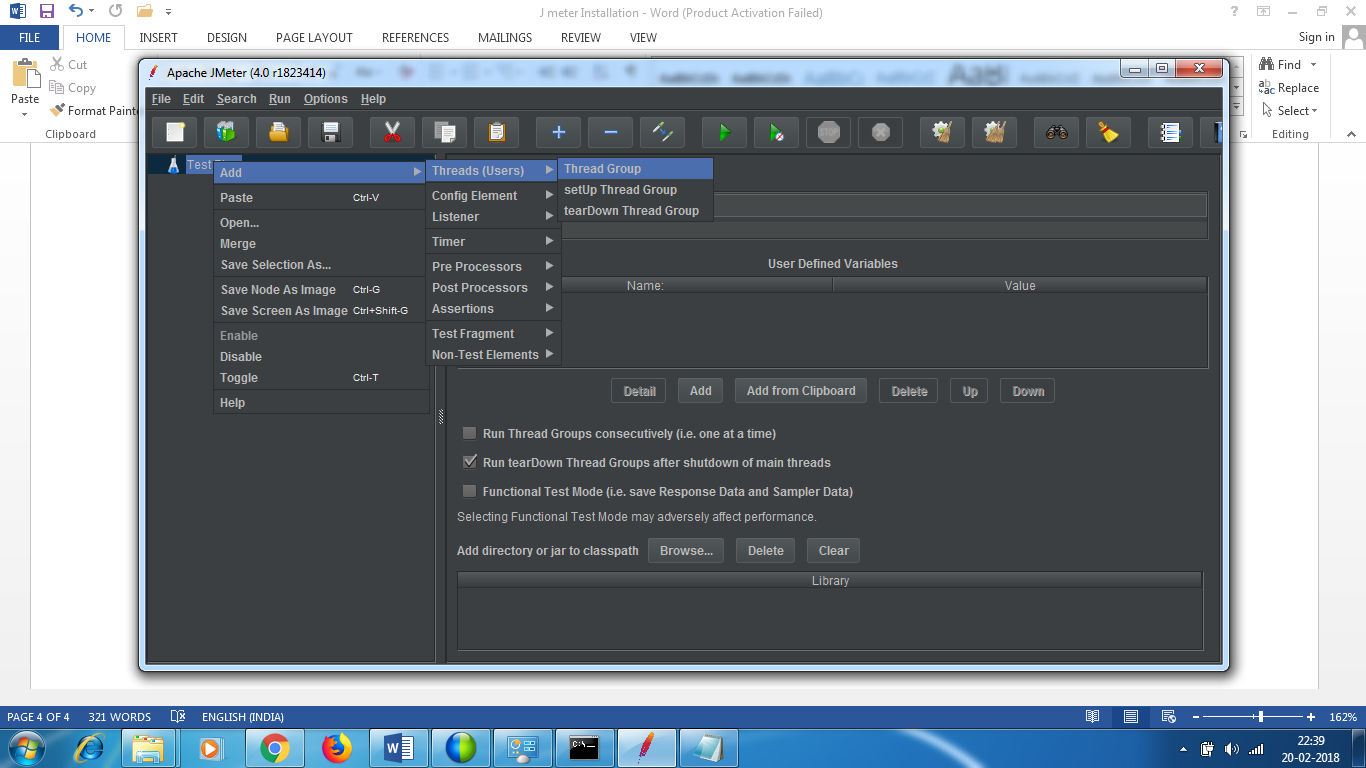


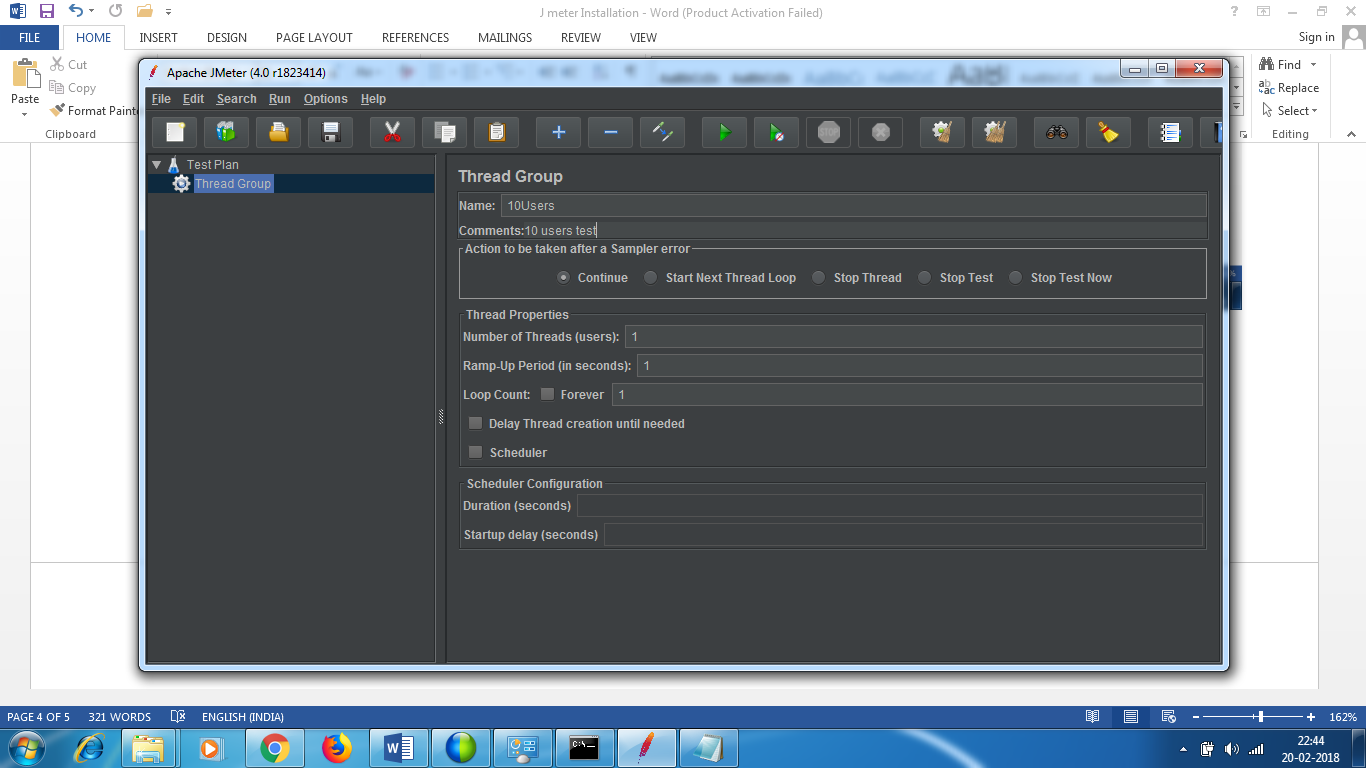
Threads – are users / vusersbecause they are created by tool.

**Step 1 :**

* Create thread group

Rt click on test plan - >Add - > Threads->Thread group

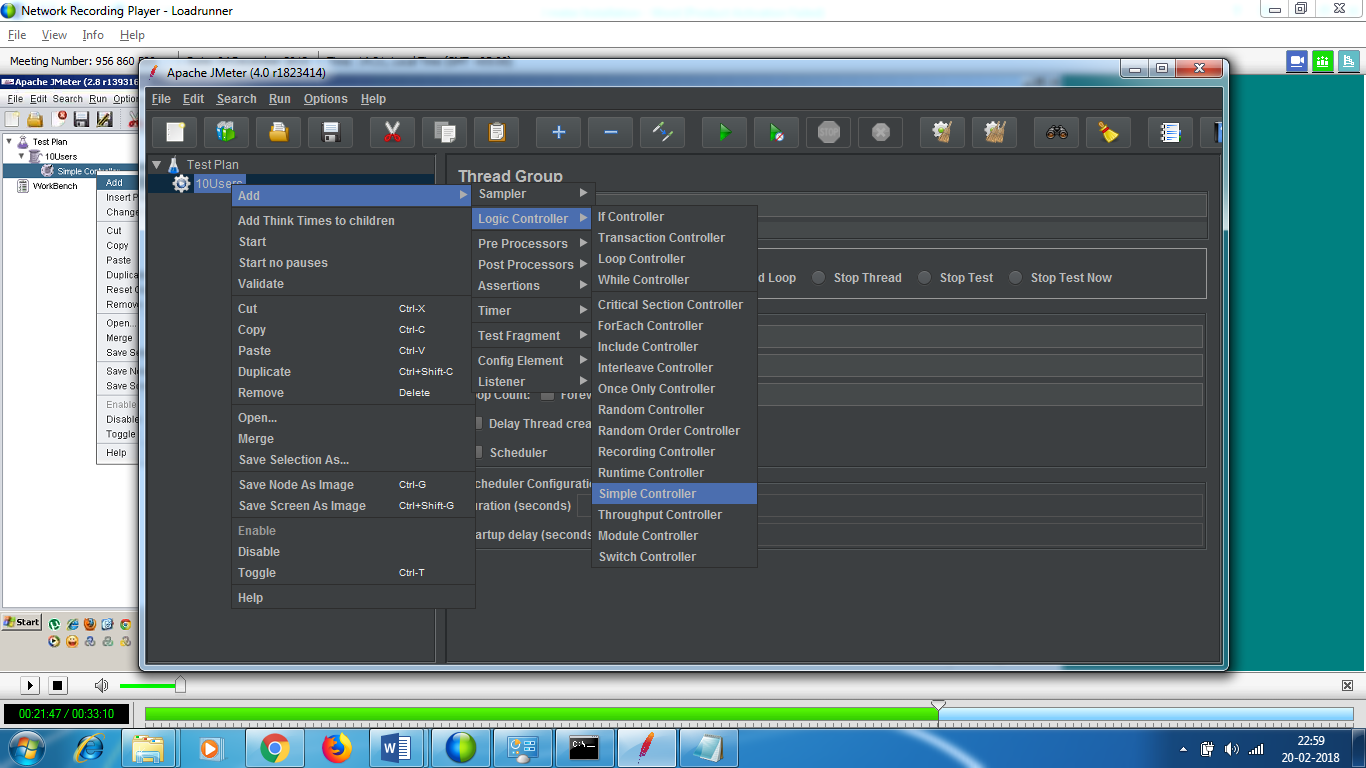




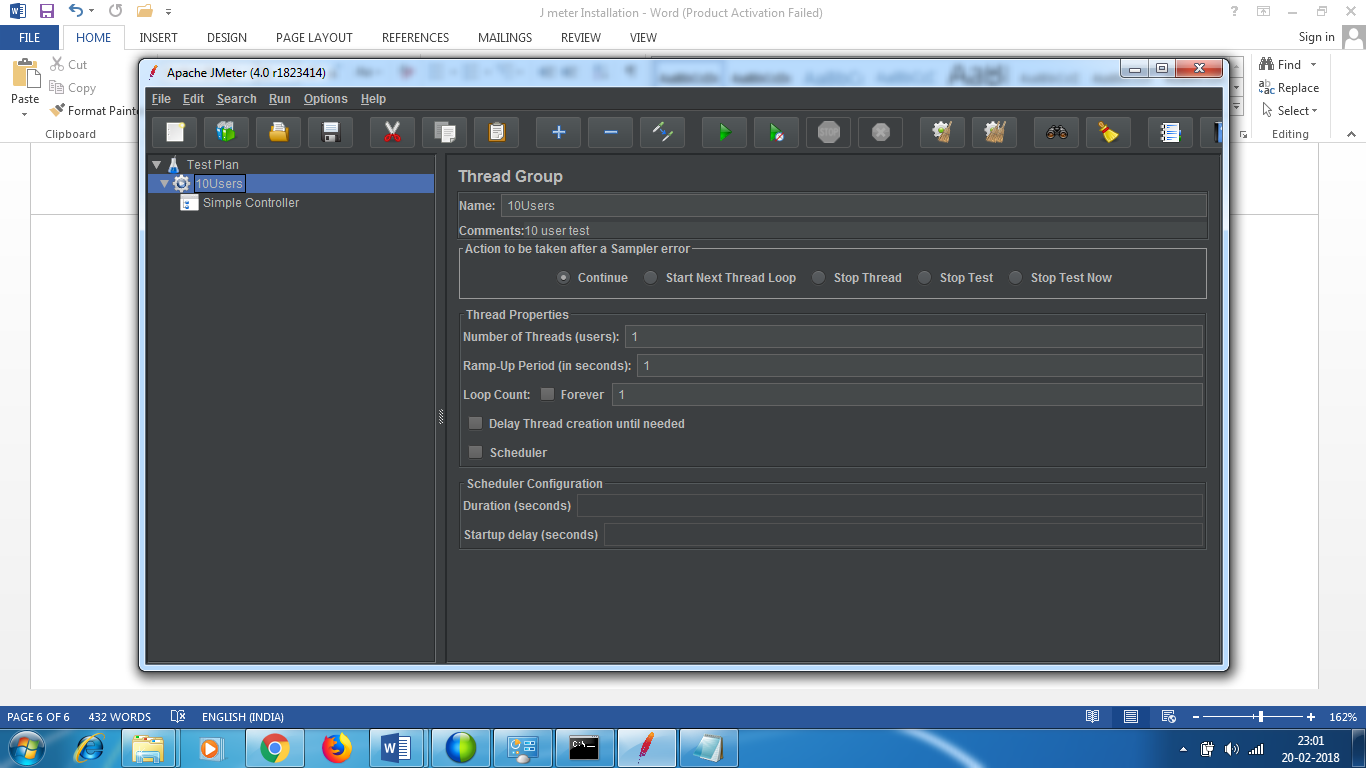
* Change the name to 10 users
* No of users – 1
* Ramp up period - What is the duration in which you want all the users to be in system
* Loop count – we will see later - set to 1 – we want the test to be run only once

**Step 2 :**

* Create controller for the thread. (simple controller)
* - container which will hold similar steps or any no of steps - > you can have entire steps involved in entire scenario . It is just to organize your code.
* You want to add logic to it.
* You can add not only organize your code but to add logic to it.

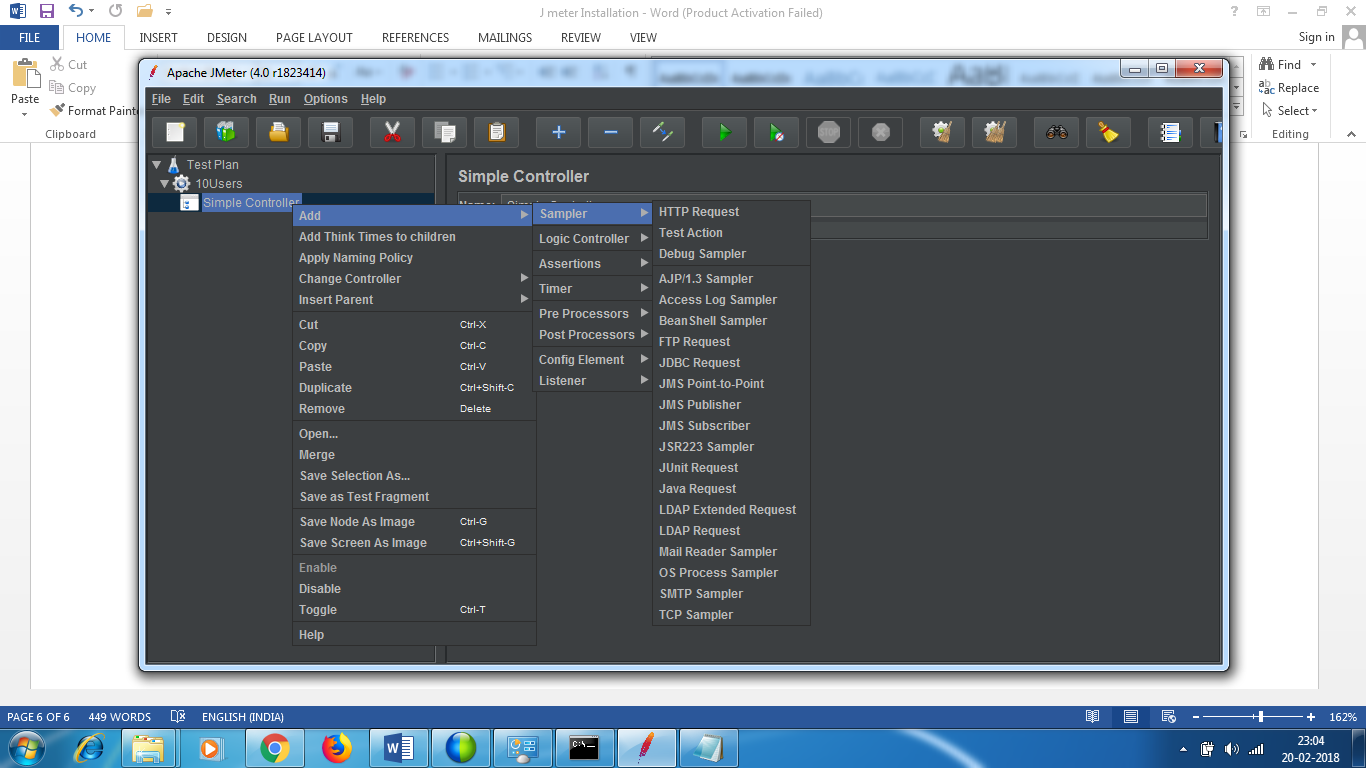


Here I am adding simple controller



**Step 3** : Add sampler – add request – what kind of request you want to send to server.

HTTP request.



Browser

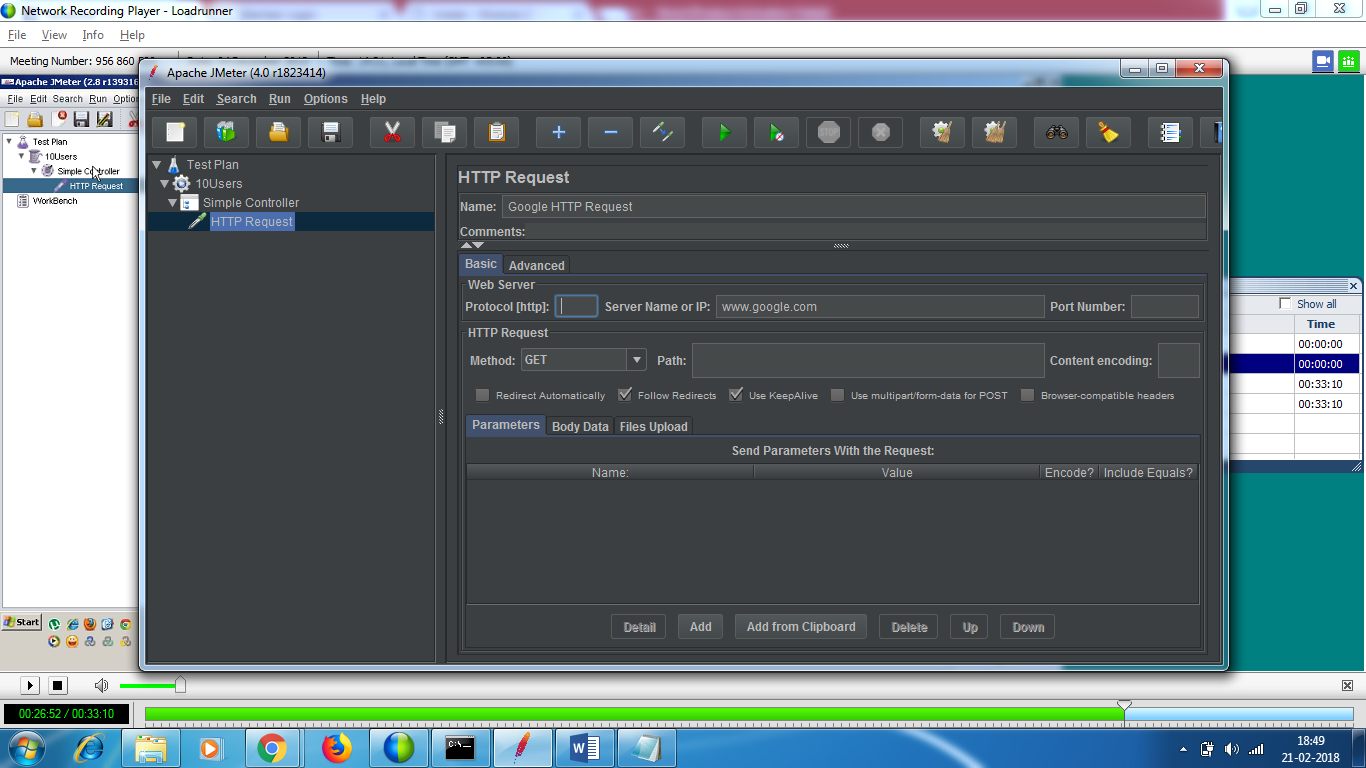
res

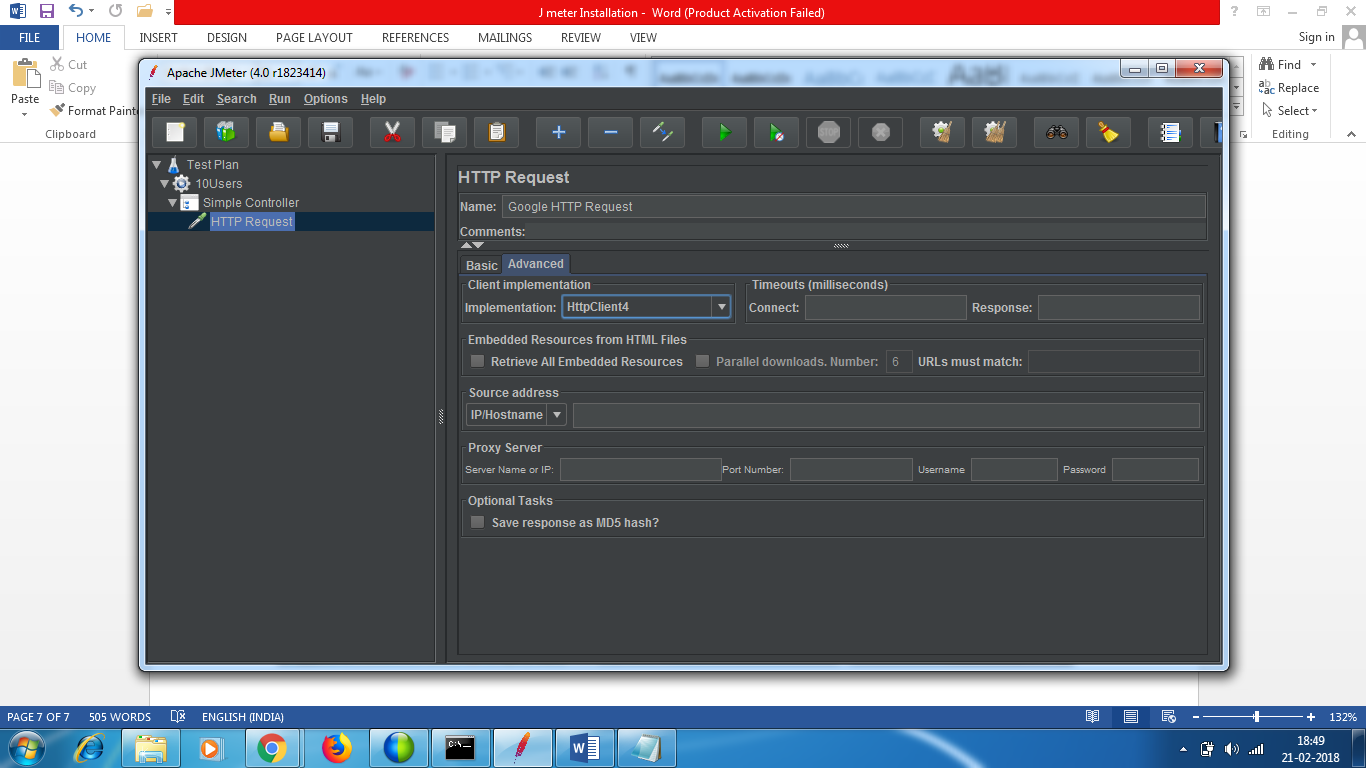
req

Webserver

In a browser when we click on a click ,it sends a requests in a form of html and the webserver decodes it and creates a response in html format . The protocol is http . So we call it as http request.

Similar is ftp protocol. Not Web based.

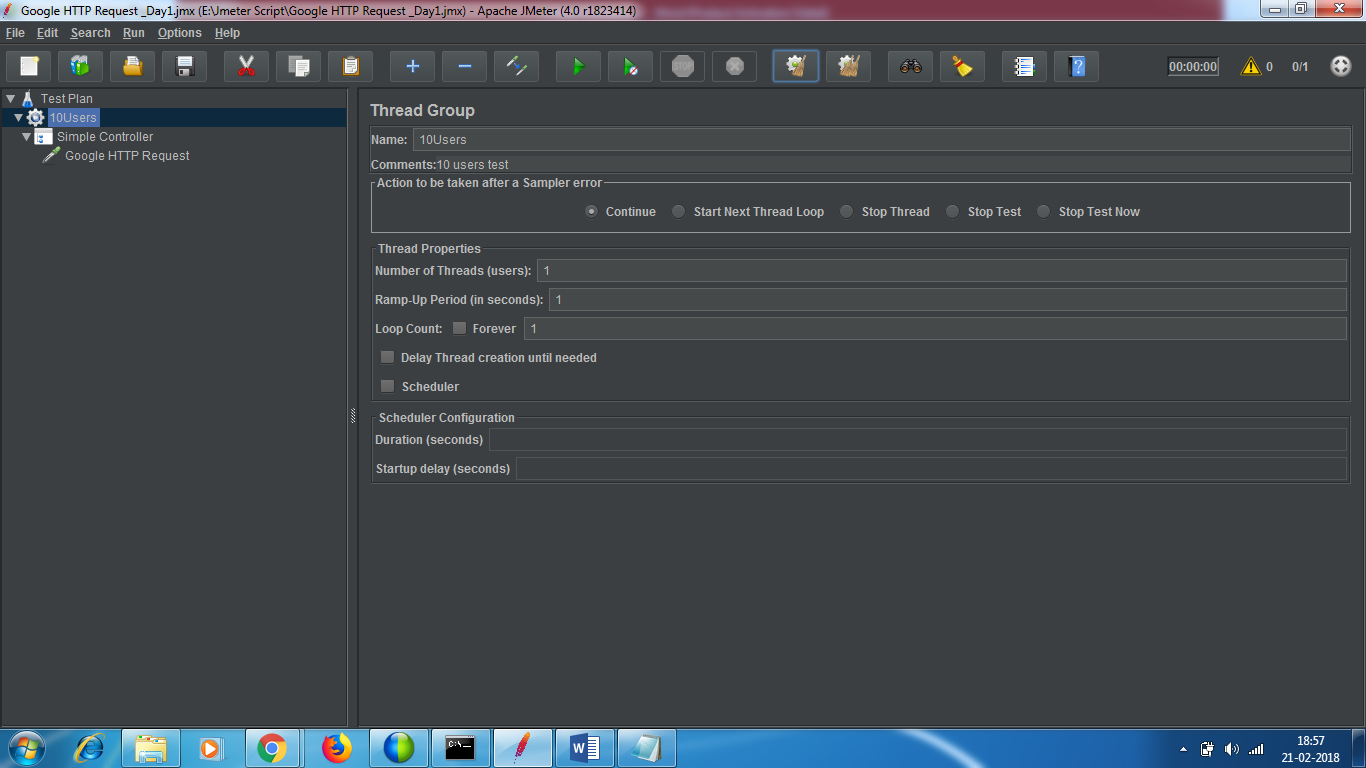




**Step 4 :** Save the script – it will save in .jmx format

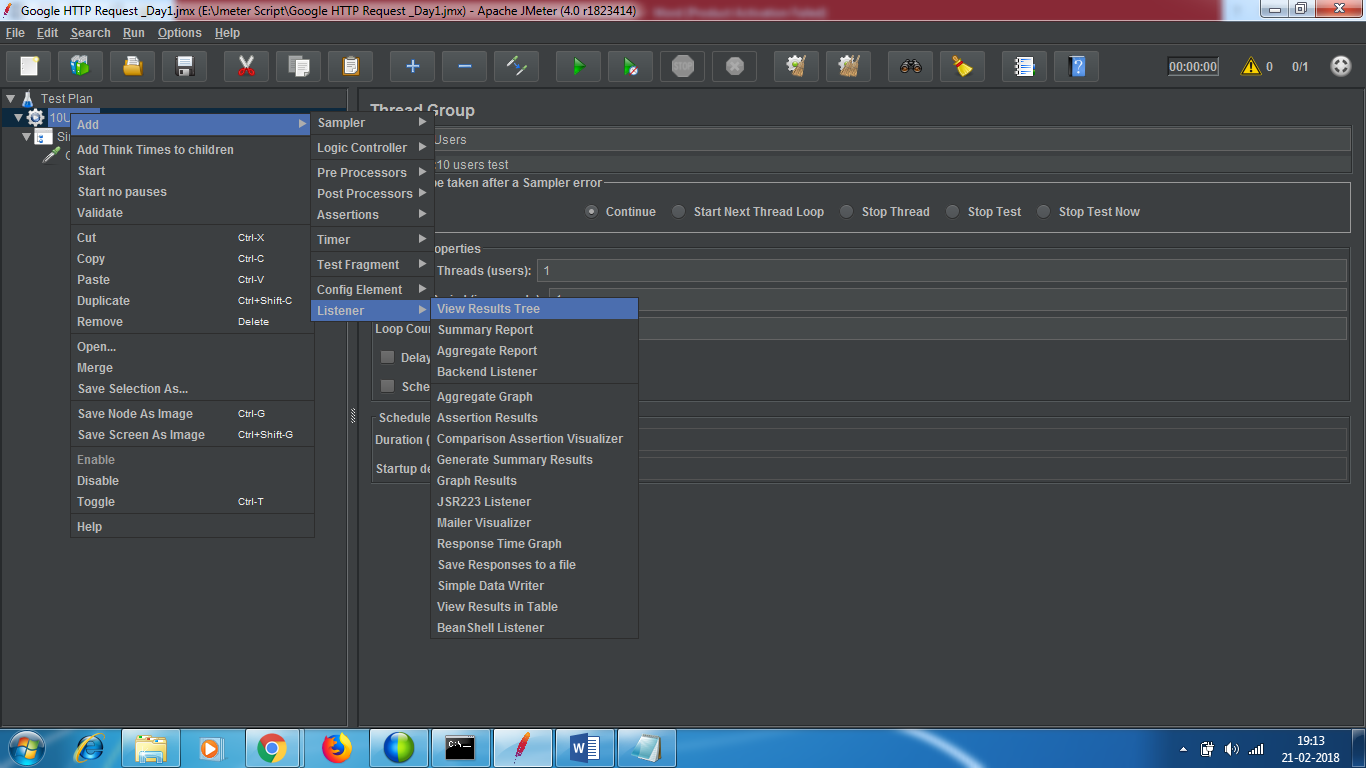
**Step 5 :** Run the script – control + r

**0/1 -> 1 user is intended to run the test.**



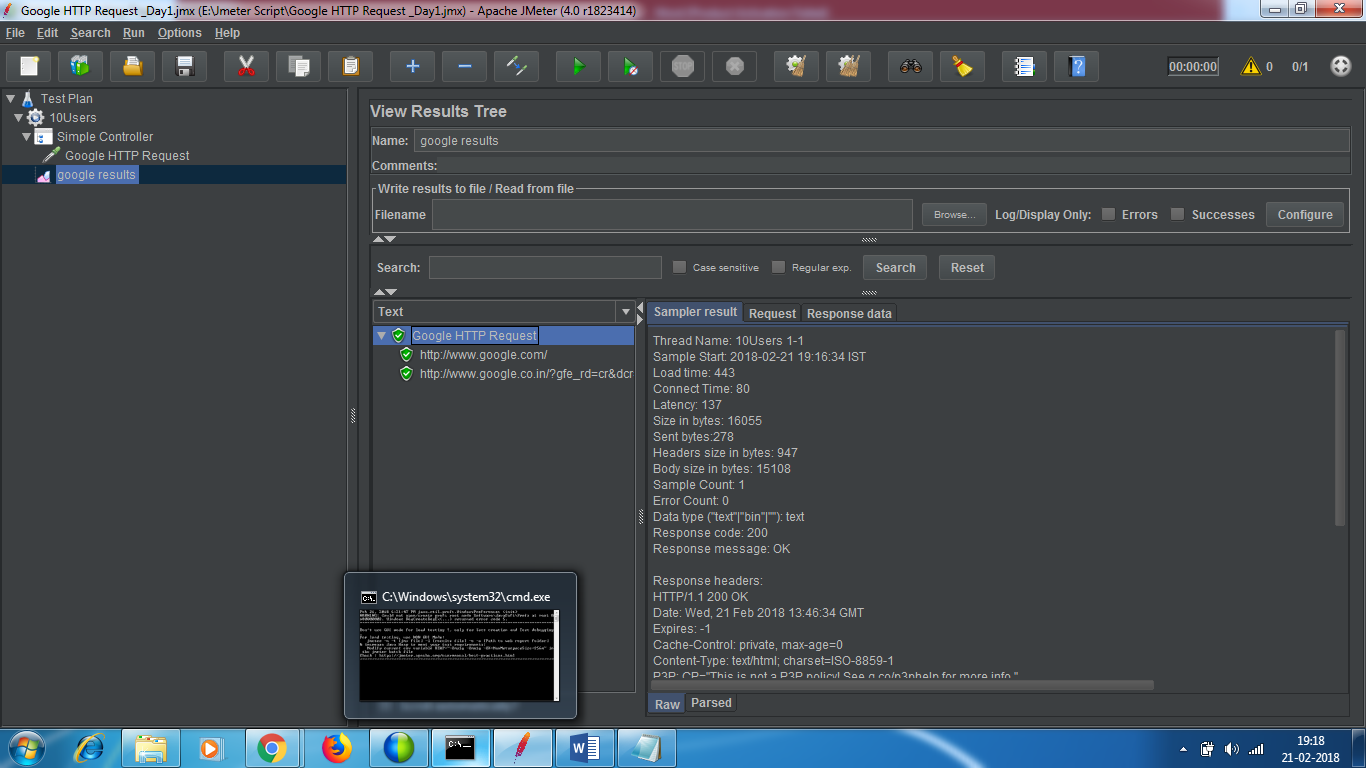
Since you don’t have listener’s added , you don’t get the results .

**Step 6** : Add listener – RT Click on test plan -> listener and click view result tree

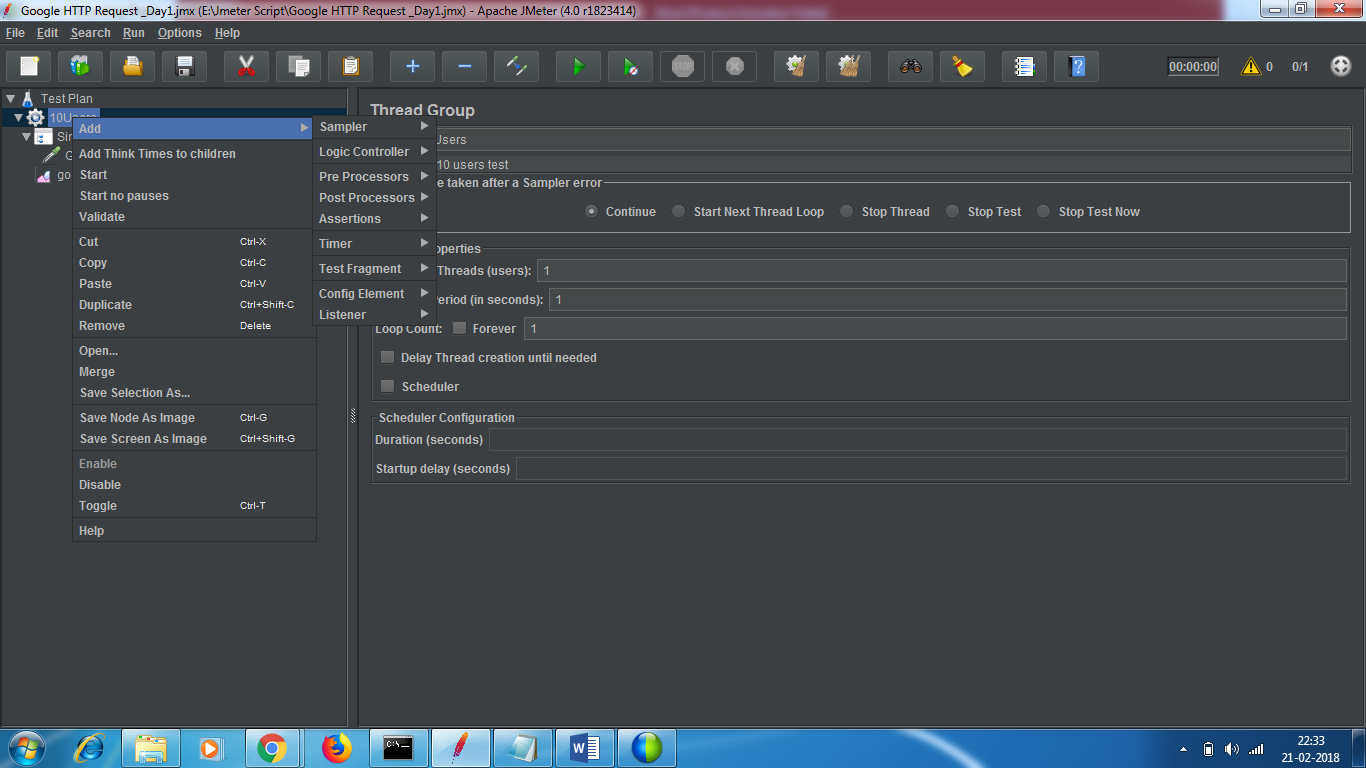


Run the test

You will get the response code as 200. You see a green tick mark , it means the test has passed.



* Whatever you see under test plan are test elements



**Logic controller** -> building logic in your test.

**Config element** – configuring your test - > before you create any test / before you create any request to hit the server or you want to do configuration with the request or you want to do cookie management or session management.

**Timer** – help you add delay’s in your tests. There are some user delays between 2 activities .

**Pre**-**processor –** before the request is fired , if you want to update the request or process the request in a certain manner , if you want to handle the session or handle dynamic data from the server or add user parameter , all these are happening before you are sending the request to server.

**Sampler**-request ,this you can compare with protocols. JMS,LDAP,JAVA,FTP,HTTP,SMTP and webservices . These are different protocols jmeter can support.

**POST Processor** -from the response you want to fetch the id or you want to assert some x path value ,you want to do something with response ,that is done in post processor.

**Assertions** – required to validate text or strings .you want to validate if response time is less than 30 ms . They are required for functional testing . Many a times , j meter is used for functional testing as well.

**Listener** – it will show you the results. It will listen and will display the results of test.

Executing test with different settings

I want to start with 1 user , with 3 , then 7 and then 10

Settings is controlled by changing the thread roup

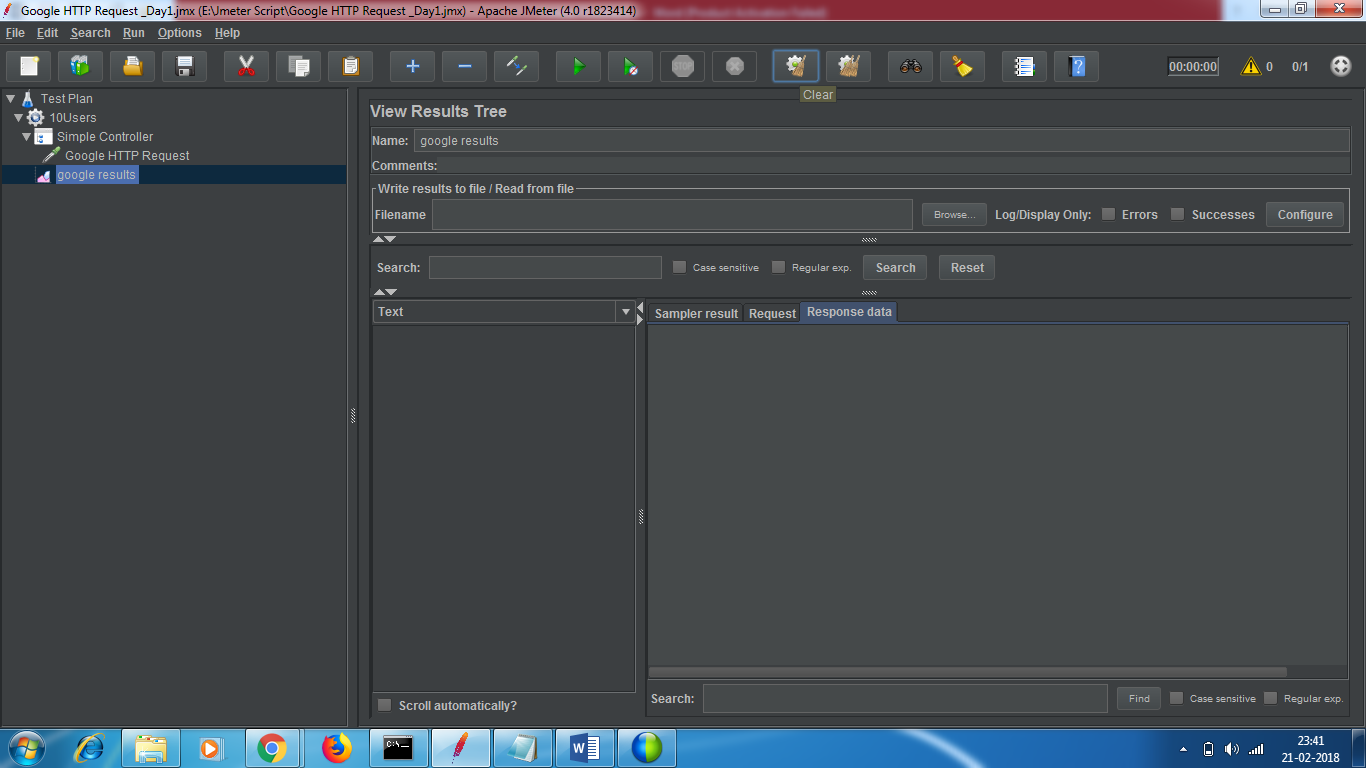
No of users : 3

Ramp up period – 3 sec – in 3 sec I want all the users to be there.

Loop count – 2

6 times the code will be run.

Every user will perform 2 times (as loop count is 2 )

Before I want to run the test , I want to clear the results .  Click on clear button to clear the previous results.

**Action to be taken after sample error :**

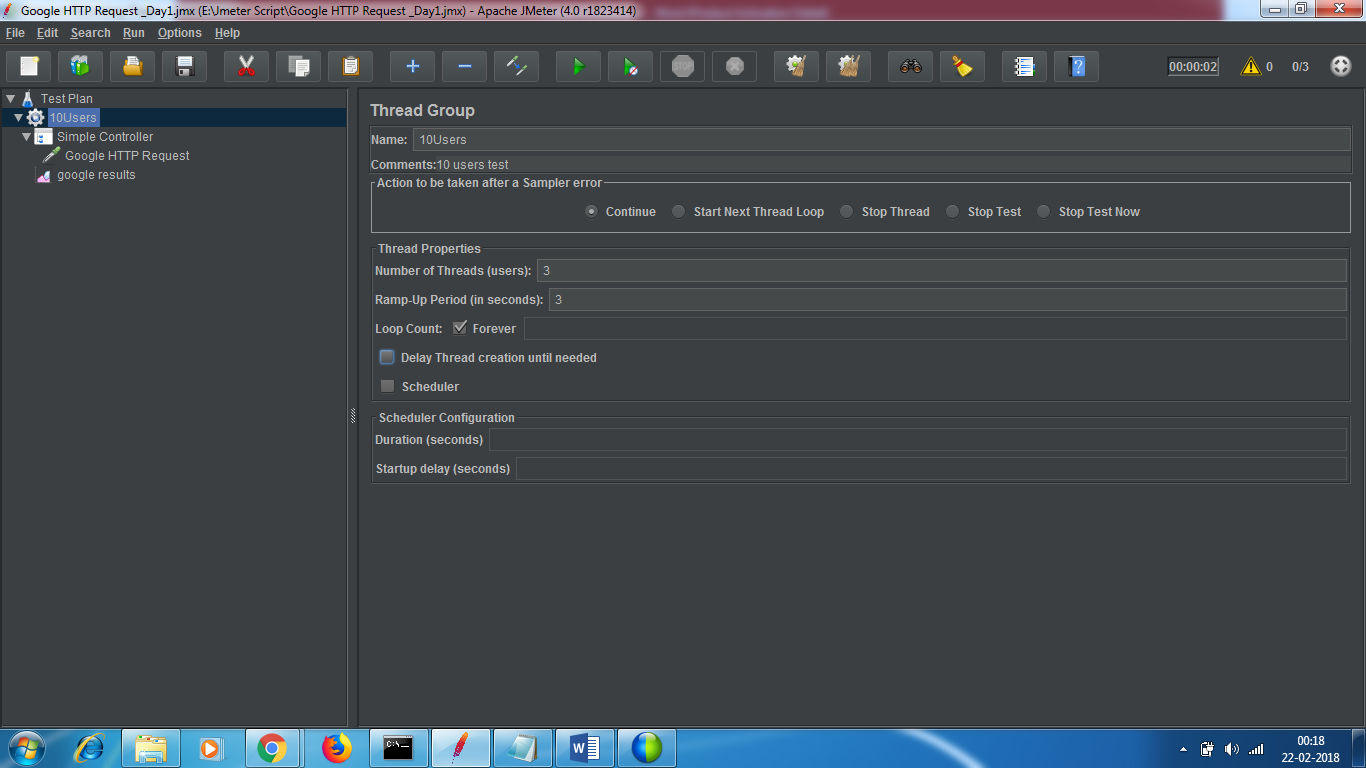
1-> whether you want to continue to next step

2 - > Start next thread loop – stop the current loop and start the next loop.

3 - > Stop the thread – stop that particular user

4 .-> Stop test - Stop the whole test >You will do some closure activities.

5 - > Stop test now – stop test right now whatever it is. Do not do any closure activities.



**Delay Thread creation until needed -**

When you don’t check this , the moment you hit start start button all the users will be created at once

You need not check this check box if the ramp up period is 10 and no of threads is 3. So you need to check this in order to delay the thread creation.

Schedule – when you want to schedule the test in future time or youwant to add start up delay.