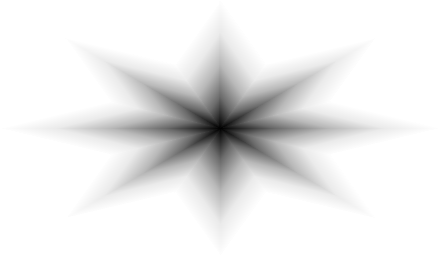
khung3khung1khung4

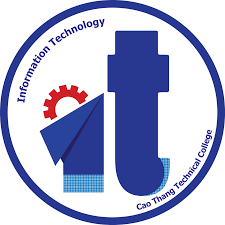
**BỘ CÔNG THƯƠNG**

**TRƯỜNG CĐ KỸ THUẬT CAO THẮNG**

**KHOA ĐIỆN TỬ - TIN HỌC**



**JMETER TOOL**



Giáo Viên BM: NGUYỄN THỊ NGỌC

Lớp : CĐ-TH 17 PMA

Sinh Viên Thực Hiện :

1. PHẠM VỊNH HY (0306171357)

2. NGUYỄN QUANG HUY (0306171351)

3. TRÀ TRẦN THANH DUY (0306171334)

...........**, tháng** 11 **năm** 2019

khung2

**I) Jmeter là gì? Tại sao nên sử dụng công cụ kiểm thử Jmeter?**

**1) JMeter là gì?**

Apache JMeter là phần mềm mã nguồn mở được viết bằng Java thuần túy, được phát triển lần đầu tiên bởi Stefano Mazzocchi. Jmeter được thiết kế nhằm kiểm thử chức năng (functional testing), kiểm thử tải (load testing) và kiểm thử hiệu năng (**performance testing**). Bạn có thể sử dụng JMeter để phân tích và đo lường hiệu năng của ứng dụng web. Kiểm thử hiệu năng là kiểm thử một ứng dụng web chống lại mức tải nặng (heavy load), lượng truy cập người dùng nhiều và đồng thời.

**2) Tại sao nên dùng công cụ kiểm thử Jmeter ?**

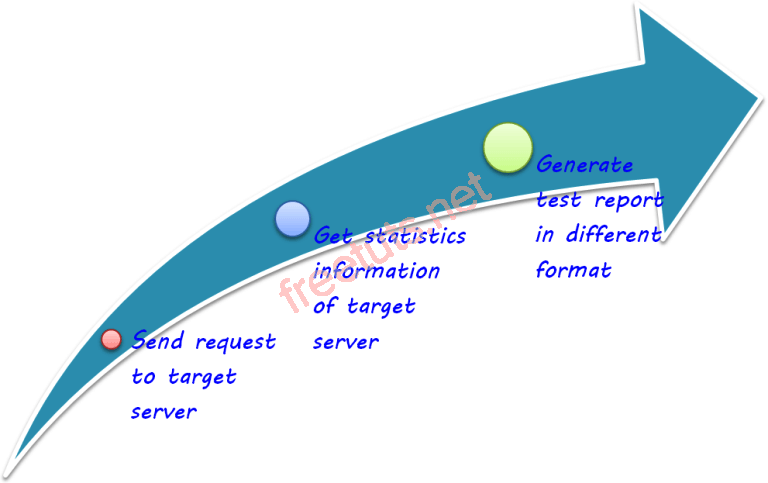
Bạn đã bao giờ kiểm thử một web server (máy chủ web) để biết hiệu quả hoạt động của website như thế nào chưa? Có bao nhiêu người dùng cùng một thời điểm có thể truy cập máy chủ web?  
Giả sử một ngày, sếp yêu cầu bạn thực hiện kiểm thử hiệu năng trang www.google.com khi cho 100 người dùng cùng truy cập. Bạn sẽ thực hiện như thế nào?

Không khả thi nếu sắp xếp 100 người với 100 PC đồng thời truy cập trang google.com. Hãy suy nghĩ về yêu cầu cơ sở hạ tầng khi bạn kiểm thử 10.000 người dùng (một con số nhỏ cho một trang web như google). Do vậy, bạn cần một công cụ phần mềm như JMeter để mô phỏng hành vi của người dùng thực tế để kiểm thử hiệu năng và kiểm thử tải trên website.

**3) Ưu Điểm JMeter ?**

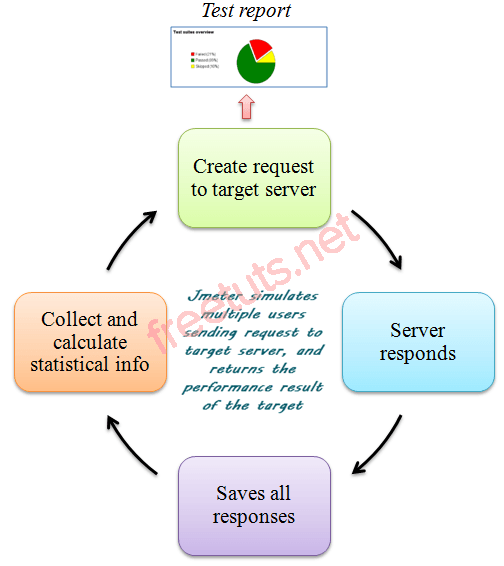
* *Là công cụ mã nguồn mở (Open Source*): JMeter hoàn toàn miễn phí, cho phép developer sử dụng mã nguồn.
* *Giao diện thân thiện (Friendly GUI)*: JMeter cực kỳ dễ sử dụng và không mất thời gian để làm quen.
* *Nền tảng độc lập (Platform independent):* JMeter là ứng dụng Desktop Java thuần túy 100%. Vì vậy, JMeter có thể chạy trên nhiều nền tảng.
* *Đa luồng (Full multithreading framework):* JMeter cho mô phỏng đồng thời và cùng thời điểm nhiều sampler với các chức năng khác nhau trên các thread group khác nhau.
* *Hiển thị kết quả kiểm thử (Visualize Test Result)*: Kết quả kiểm thử có thể được hiển thị ở định dạng khác như biểu đồ, bảng, tree và file log.
* *Cài đặt dễ dàng (Easy installation)*: Chỉ cần copy và chạy tập tin \* .bat để khởi động JMeter, không cần cài đặt.
* *Khả năng mở rộng cao (Highly Extensible)*: Tester có thể viết các kịch bản kiểm thử của riêng mình. JMeter cũng hỗ trợ các plugin trực quan.
* *Nhiều chiến lược kiểm thử (Multiple testing strategy)*: JMeter hỗ trợ nhiều chiến lược kiểm thử như Load Testing, Distributed Testing, và Functional Testing.
* *Mô phỏng (Simulation)*: JMeter có thể mô phỏng nhiều người dùng với các threads đồng thời, tạo tải nặng đối với ứng dụng web đang được kiểm thử.
* *Hỗ trợ đa giao thức (Support multi-protocol):* JMeter không chỉ hỗ trợ kiểm thử ứng dụng web mà còn đánh giá hiệu năng máy chủ cơ sở dữ liệu. Tất cả các giao thức cơ bản như HTTP, JDBC, LDAP, SOAP, JMS và FTP đều được JMeter hỗ trợ.
* *Record & Playback*: Ghi lại hoạt động của người dùng trên trình duyệt và mô phỏng chúng trong một ứng dụng web sử dụng JMeter.
* *Kịch bản test (Script Test):* Jmeter có thể được tích hợp với Bean Shell & Selenium để kiểm thử tự động.

**4) Jmeter hoạt động như thế nào ?**

****Luồng hoạt động cơ bản của JMeter như hình bên dưới:

Luồng hoạt động cơ bản của JMeter

JMeter mô phỏng một nhóm người dùng gửi yêu cầu đến một máy chủ và trả về thông tin thống kê của máy chủ thông qua biểu đồ.  
Quy trình khi gửi một request trong JMeter thể hiện như trong hình bên dưới:



Quy trình khi gửi một request trong JMeter

**5) Các phiên bản Jmeter?**

|  |  |
| --- | --- |
| **Version** | **Mô tả** |
| 1 | Phiên bản phát hành chính thức đầu tiên |
| 2.13 | Java 6+ |
| 3 | Java 7+ |
| 3.1 | Java 7+ |
| 3.2 | Java 8+ |
| 3.3 | Java 8 |
| 4 | Java 8 / 9 |

**5) Các hệ điều hành hỗ trợ JMeter?**

* Linux
* Windows
* Mac OS
* Ubuntu

**II) Cài đặt Jmeter**

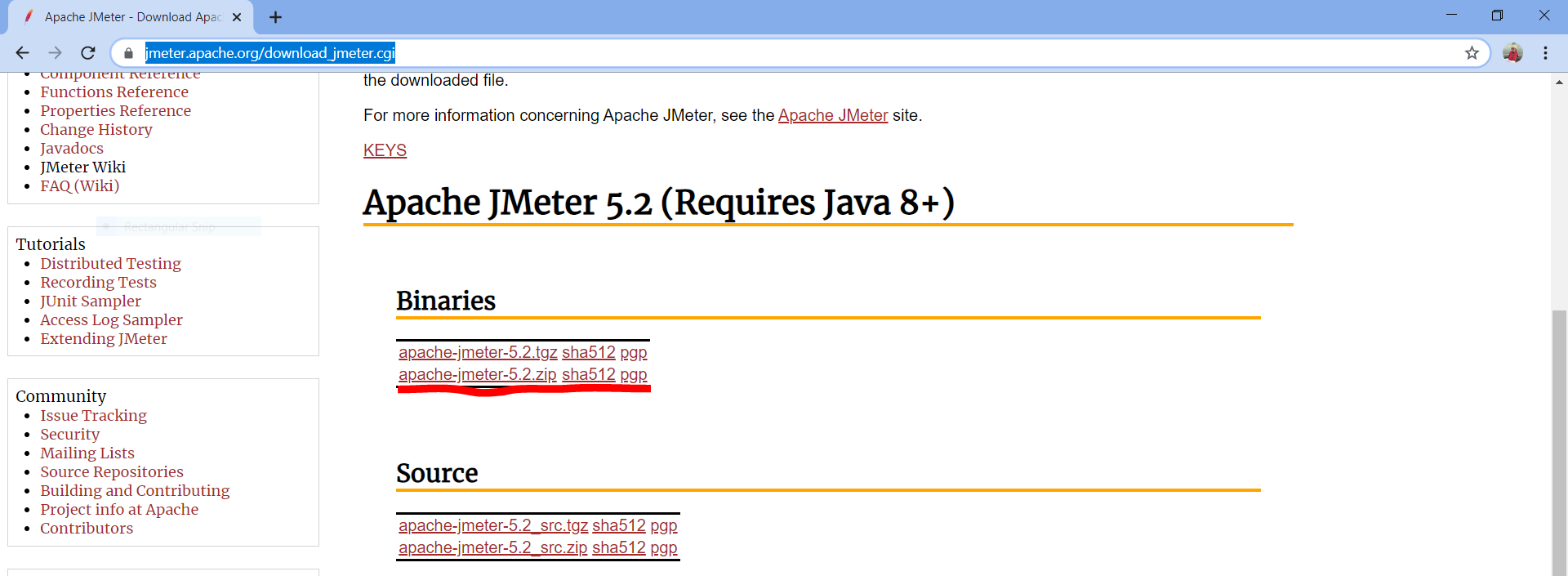
**1) cài đặt trên window**

**Điều kiện:** Jmeter viết bằng java nên muốn chạy Jmeter trước hết ta cài đặt **JR**E và **JDK**

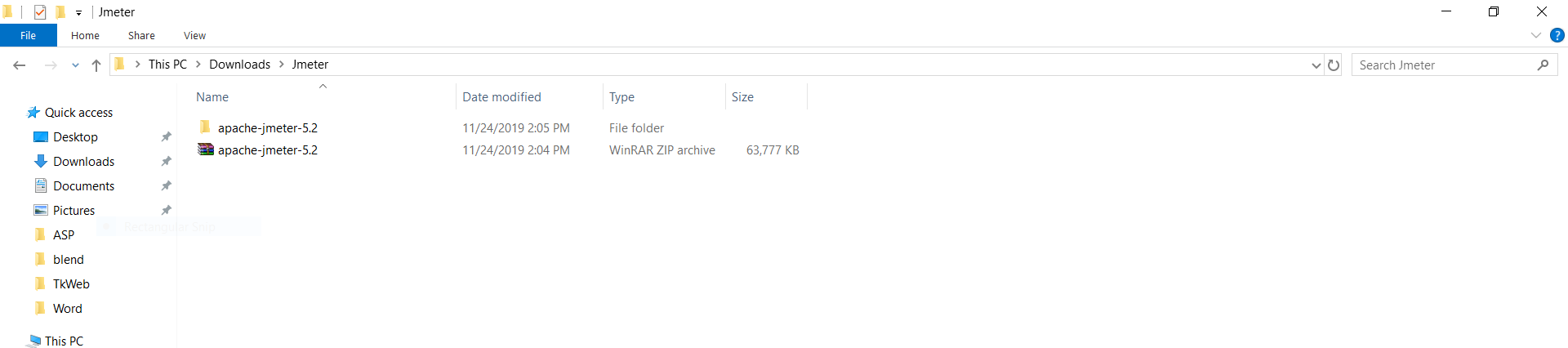
**Bước 1: vào đường link dưới đây để tải JMeter**

<https://jmeter.apache.org/download_jmeter.cgi>

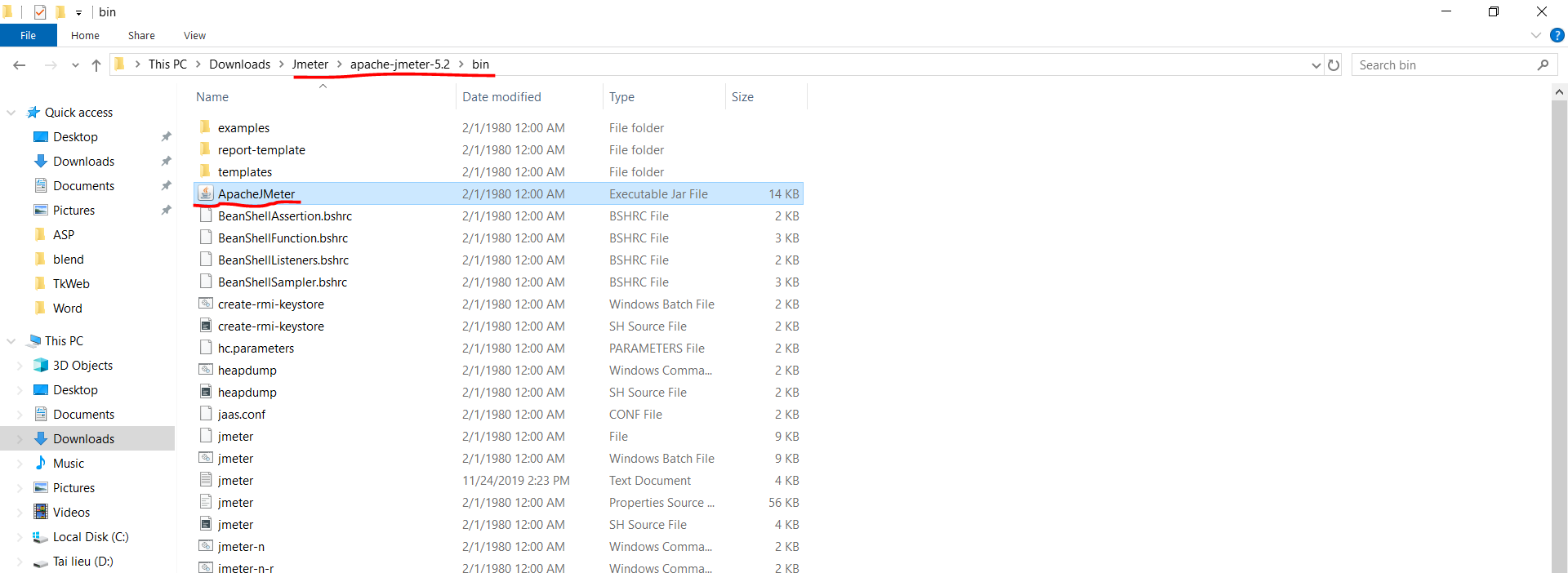
**Bước 2: download về máy**



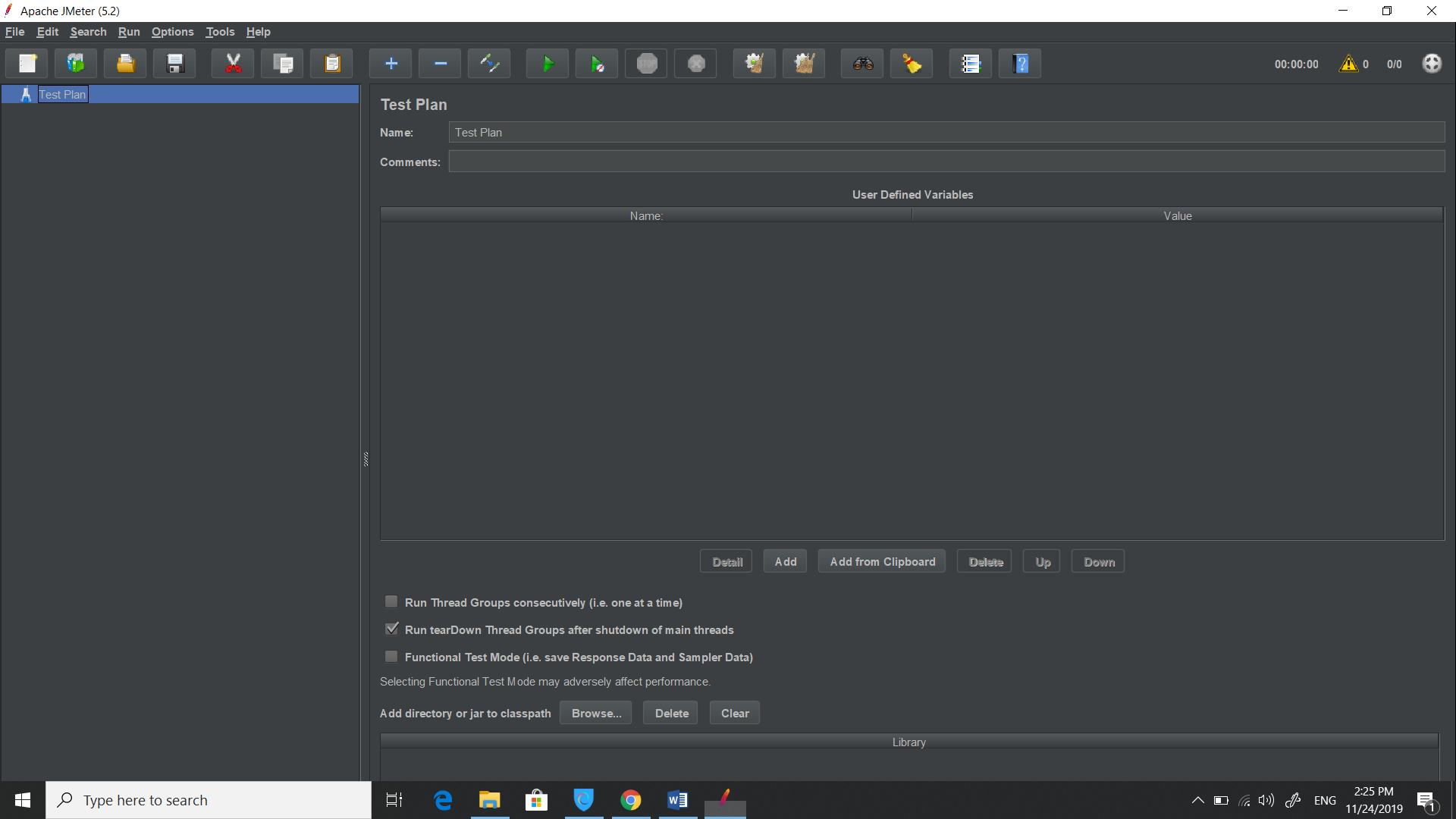
**Bước 3: giải nén file vừa tải về**



**Bước 4: vào thư mục apache-jmeter-5.2 => vào bin chọn ApaheJMeter**



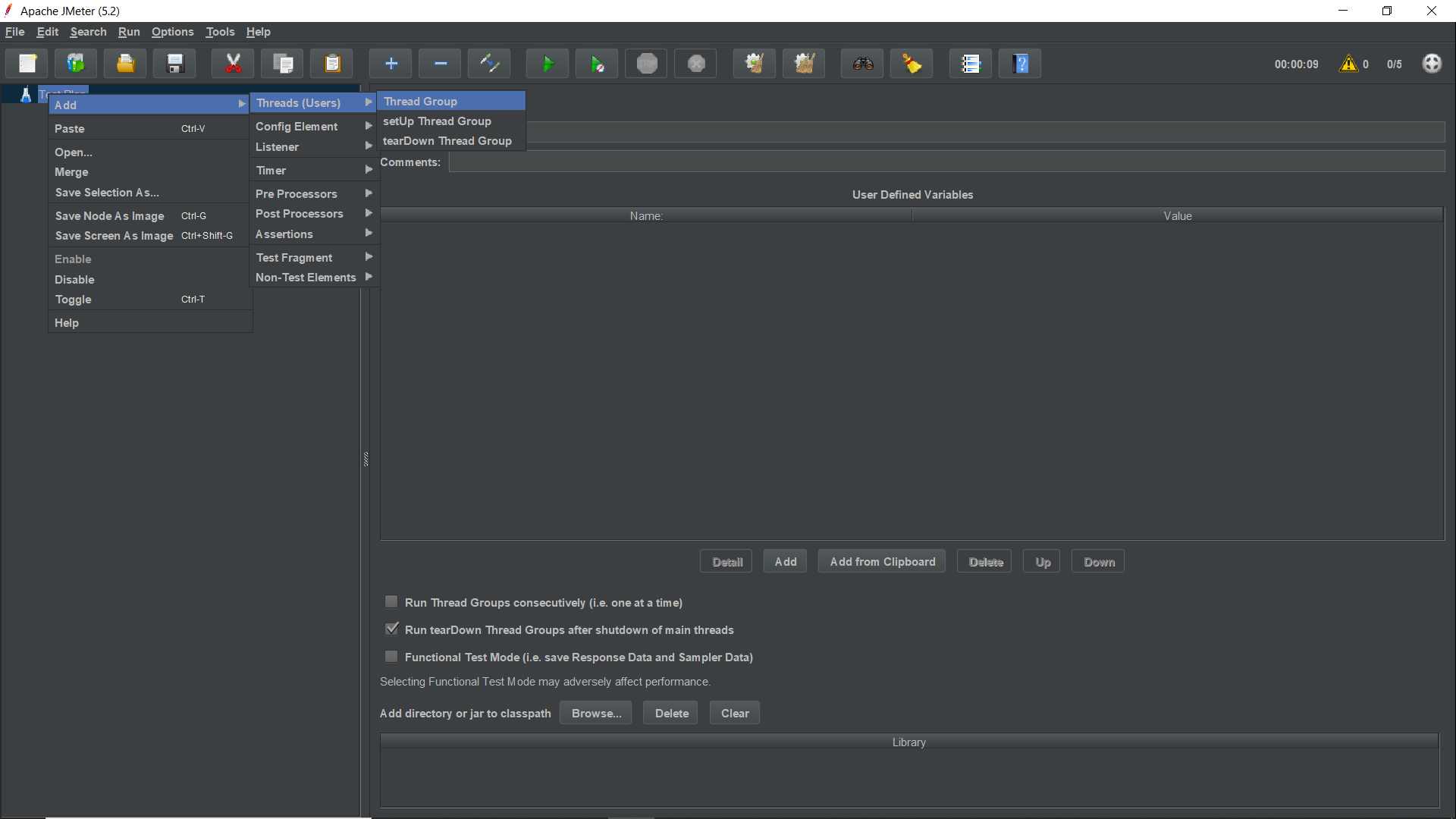
**Bước 5: Giao diện khi chạy thành công**



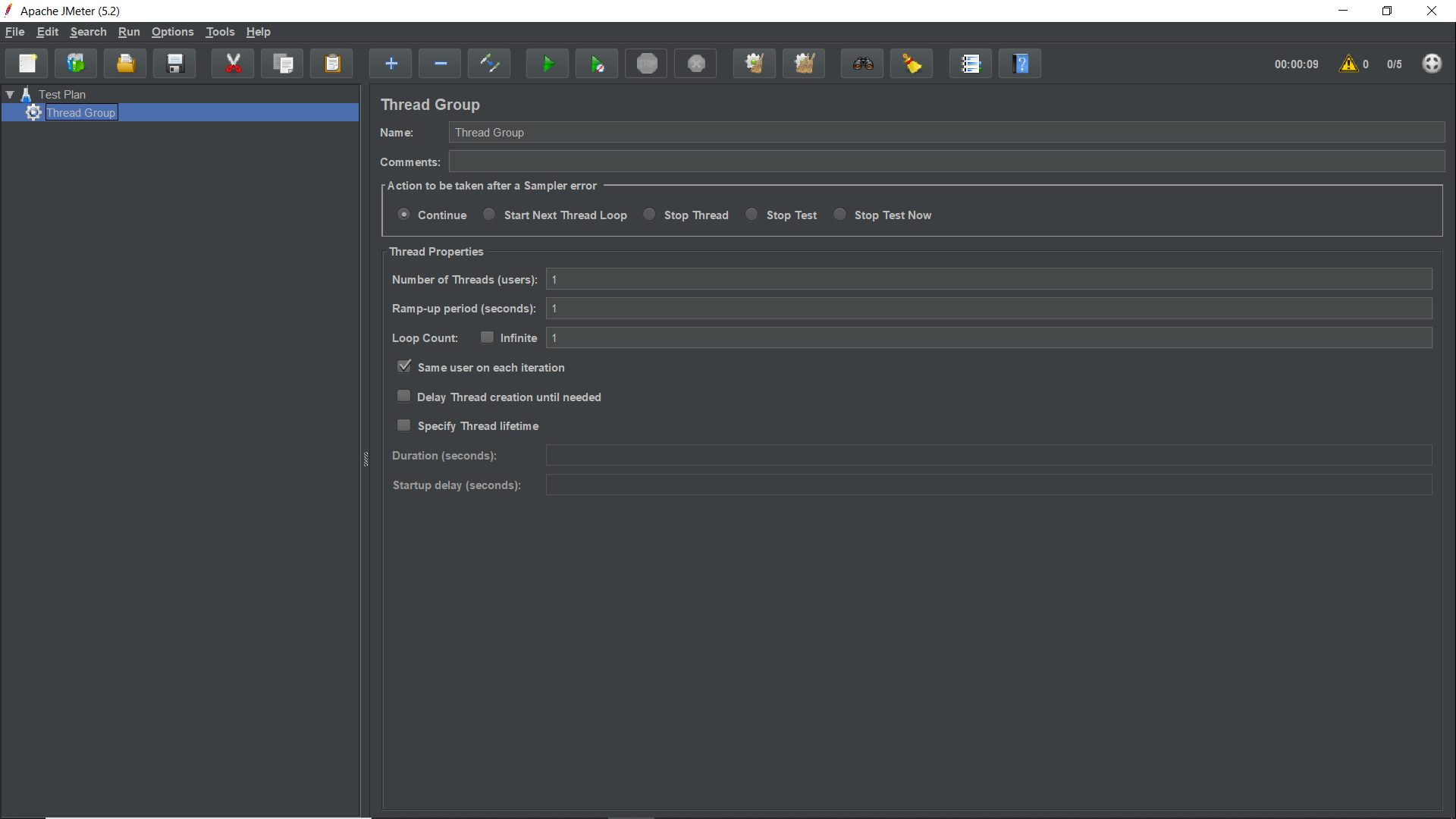
**III) Tạo project, đọc report**

**Bước 1: Tạo ThreadGroup**

Chuôt phải vào Test Plan, chọn Add > Threads (Users) > Thread Group



Tạo xong Thread Group



Các tham số trong Thread Group

Name: đặt tên cho ThreadGroup ( ở đây mình đặt JmeterUser ).

Number of Threads(users): số lượng người dùng mà ta muốn mô phỏng.

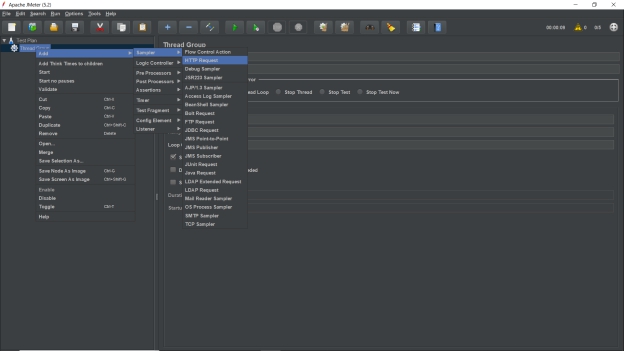
Ramp-up Period (in seconds): Cho biết thời gian để JMeter tạo ra tất cả những thread cần thiết. Ví dụ nếu tham số này là 10 thì trong 10 giây tất cả các Number of Threads đã khai báo ở trên sẽ được gửi đi trong 10 giây, nếu đặt tham số này là 0 thì tất cả các yêu cầu sẽ được gửi đi cùng một lúc.

Loop Count: số lần thực hiện thread này, nếu chọn check box Infinite thì nó sẽ lặp mãi.

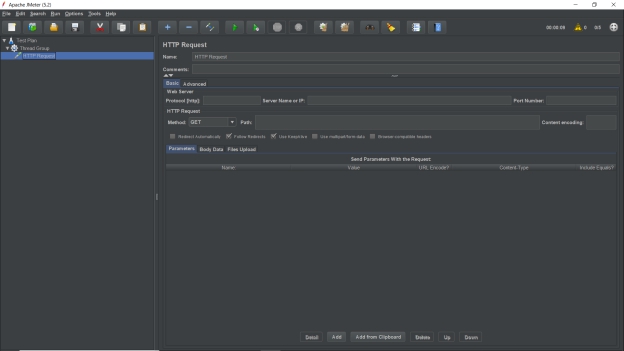
Ở đây mình đặt là Test Load Facebook cho dễ hiểu, sẽ có 10 user thực hiện gửi request trong cùng 1s và thực hiện 1 lần duy nhất.

**Bước 2: Tạo Request**

Chuột phải vào Thread Group –> Add -> Sampler –> HTTP Request



Tạo xong Request



Các tham số trong HTTP Request:

Name: đặt tên request

Server name or Ip: điền địa chỉ trang web cần truy cập.

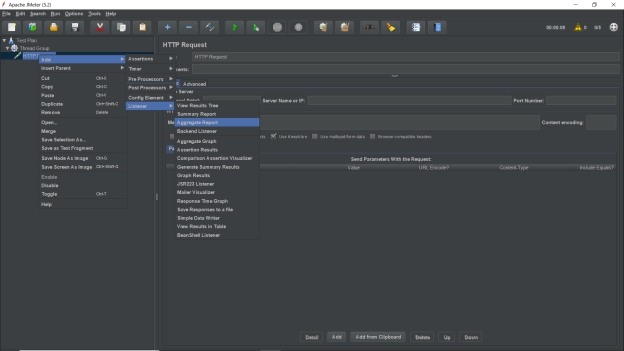
Port number: cổng chạy ứng dụng (thường cho HTTP là 80, nếu cổng khác thì nó sẽ thể hiện số cổng bên trong URL)

Method: method của request (ở đây mình tải trang lên là GET, những trương hợp login sẽ là POST… tùy theo ứng dụng web qui định)

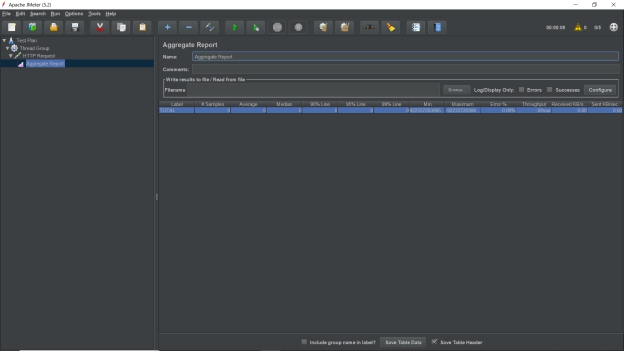
Parameter, Body Data, File Upload: Dùng thể thêm dữ liệu vào request khi gửi đi.

**Bước 3: Tạo listener**

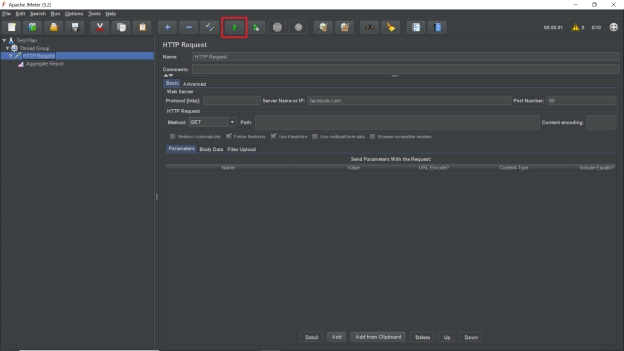
Chuột phải vào Thread Group –> Add -> Listener –> Aggregate Report



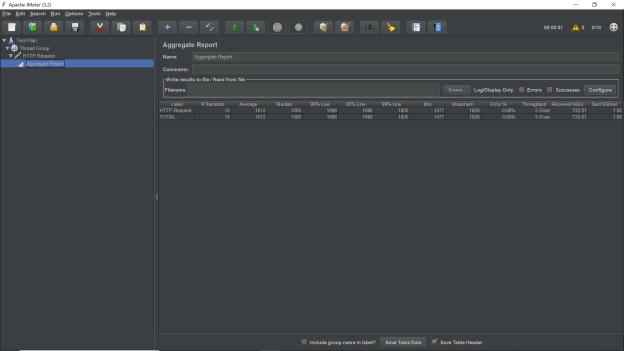
Tạo xong listener



**Bước 4: chạy project**



**Bước 5: kết quả**



Label: Hiển thị tên của từng requests có trong test plan của bạn.

Samples: Tổng số lần run của request.

Công thức: Samples = Number of Threads (users) \* Loop Count

Average (millisecond): Thời gian phản hồi trung bình của request, tính cho đến lần run cuối cùng.

Min (millisecond): Respone Time thấp nhất của request tính cho toàn bộ tất cả các lần run.

Max (millisecond): Respone Time cao nhất của request tính cho toàn bộ tất cả các lần run.

Median (millisecond): Thời gian xử lí request ở giữa.

90% Line (90th Percentile) (millisecond):nghĩa là 90% số requests sẽ có response time nhỏ hơn giá trị hiển thị trong table, 10% số requests còn lại sẽ có response time lớn hơn giá trị hiển thị trong table.

95% Line (90th Percentile) (millisecond):nghĩa là 95% số requests sẽ có response time nhỏ hơn giá trị hiển thị trong table, 5% số requests còn lại sẽ có response time lớn hơn giá trị hiển thị trong table

99% Line (90th Percentile) (millisecond):nghĩa là 99% số requests sẽ có response time nhỏ hơn giá trị hiển thị trong table, 1% số requests còn lại sẽ có response time lớn hơn giá trị hiển thị trong table

Error %: % số lượng request bị fail, bị lỗi. Ví dụ bạn run request A 100 lần và thấy có 15% errors, nghĩa là request A đã fail/error 15 lần (100\*15%)

Throughput: Thông lượng. Con số này cho bạn biết được số lượng requests được hệ thống (server) xử lý trong 1 đơn vị thời gian, có thể là giây, phút, hoặc giờ.