

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN HỌC PHẦN
KIỂM THỬ PHẦN MỀM
TÌM HIỂU VÀ SỬ DỤNG CÁC KỸ THUẬT KIỂM THỬ ĐƯỢC HỖ TRỢ BỞI
CÔNG CỤ KIỂM THỬ LOADFOCUS

GVHD: Ths. Nguyễn Thái Cường

Nhóm: 14 **Lớp:** 20232IT6048004

Sinh viên:

Hồ Nam Tú	- 2021604025
Phan Anh Thắng	- 2021601236
Nguyễn Viết Vượng	- 2021601739
Phùng Nguyễn Huy	- 2021604071
Nguyễn Quang Huy	- 2020603337

Hà Nội – Năm 2024

LỜI CẢM ƠN

Xã hội ngày càng hiện đại, các kỹ thuật công nghệ ngày càng phát triển, đi cùng với nó là các nghiên cứu phát triển không ngừng về lĩnh vực công nghệ thông tin, cho ra đời các hệ thống trang web, máy móc thông minh ứng dụng rộng rãi trong hầu hết các lĩnh vực đời sống như giáo dục, y tế, quân sự, ... đặc biệt là về phát triển các hệ thống trang web.

Việc kiểm tra trang web, API hoặc ứng dụng dành cho thiết bị di động là một việc hết sức cần thiết và quan trọng. Kiểm tra sớm có thể giúp chúng ta phát hiện lỗi và cải thiện chất lượng, hiệu suất tổng thể của trang web.

Công việc này có thể khó khăn, tốn thời gian và cung cấp được ít thông tin chi tiết cần thiết cho sản phẩm, nhà tuyển dụng và nhóm phát triển. Vì vậy, nhóm chúng em đã lựa chọn đề tài **“Tìm hiểu và sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi công cụ kiểm thử Loadfocus”** để có thể kiểm tra được toàn diện trang web hơn.

Công cụ này sử dụng nền tảng độc lập hiện đại và hoàn thiện chỉ sau vài phút, không cần kỹ năng về code, có thể giúp hoàn thành mọi tác vụ thử nghiệm cần thiết.

Để hoàn thành bài tập lớn này, nhóm 18 chúng em xin gửi lời cảm ơn đến Thầy Ths.Nguyễn Thái Cường đã tận tình hướng dẫn nhóm trong suốt thời gian thực hiện để chúng em hoàn thành tốt đề tài của mình. Với vốn kiến thức được tiếp thu trong thời gian học không chỉ là nền tảng cho quá trình làm đề tài mà còn là hành trang quý báu để chúng em tiếp tục quá trình học tập với tâm thế vững chắc và tự tin.

Chúng em xin chân thành cảm ơn!

Nhóm thực hiện

Nhóm 14

MỤC LỤC

MỤC LỤC	3
PHẦN 1. MỞ ĐẦU	5
1.1. Lý do chọn đề tài	5
1.2. Nội dung đề tài	5
1.3. Kiến thức cần có	6
PHẦN 2. KẾT QUẢ NGHIÊN CỨU	7
2.1. Tổng quan	7
2.1.1. Sơ lược về kiểm thử phần mềm.....	7
2.1.2. Quy trình kiểm thử phần mềm.....	9
2.1.3. Các phương pháp kiểm thử phần mềm.....	13
2.1.3.1. Kiểm thử hộp đen.....	13
2.1.3.2. Kiểm thử hộp trắng.....	14
2.2. Công cụ kiểm thử LoadFocus	15
2.2.1. Giới thiệu công cụ	15
2.2.2. Đặc điểm.....	16
2.2.3. Cài đặt và sử dụng công cụ LoadFocus.....	20
2.3. Giới thiệu trang web www.hau.edu.vn	20
2.3.1. Giới thiệu	20
2.3.2. Tài liệu đặc tả yêu cầu phần mềm	20
2.4. Kiểm thử trang web www.hau.edu.vn	21
2.4.1. Lập kế hoạch kiểm thử	21
2.4.2. Kiểm tra tải Jmeter (Hồ Nam Tú).....	22
2.4.2.1. Phân tích thiết kế kiểm thử.....	22
2.4.2.2. Thực hiện kiểm thử	23
2.4.3. Kiểm tra tải Load Testing (Nguyễn Viết Vượng)	27
2.4.3.1. Phân tích thiết kế kiểm thử.....	27
2.4.3.2. Thực hiện kiểm thử	30
2.4.4. Kiểm tra tốc độ trang Speed Monitor Testing (Phan Anh Thắng)	33
2.4.4.1. Phân tích thiết kế kiểm thử	33

2.4.4.2. <i>Thực hiện kiểm thử</i>	38
2.5. Báo cáo kiểm thử	39
PHẦN 3: KIẾN THỨC VÀ BÀI HỌC KINH NGHIỆM	43
TÀI LIỆU THAM KHẢO	45

PHẦN 1. MỞ ĐẦU

1.1. Lý do chọn đề tài

Trước hết, kiểm thử phần mềm là một phần cần thiết và không thể thiếu trong quá trình phát triển phần mềm. Đã có những sự cố vô cùng tai hại khi kiểm thử phần mềm chưa phát triển mạnh mẽ như ngày nay như sự cố Y2k. Việc đảm bảo chất lượng và độ tin cậy của phần mềm là mục tiêu chung, và việc thực hiện kiểm thử một cách cẩn thận là một yếu tố quan trọng để đạt được mục tiêu này. Hiện nay, ngành kiểm thử phần mềm đã vô cùng phát triển, đi kèm với đó là hàng loạt các công cụ kiểm thử phần mềm được sinh ra với mục đích đơn giản hoá, cũng như nâng cao chất lượng của phần mềm.

LoadFocus là một trong số đó. Vì vậy chúng em quyết định lựa chọn đề tài “Tìm hiểu và sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi công cụ kiểm thử loadfocus “. Công cụ kiểm thử Loadfocus được chọn để nghiên cứu và sử dụng trong đề tài này vì tính linh hoạt và hiệu quả của nó. Loadfocus cung cấp các tính năng mạnh mẽ để kiểm thử hiệu suất, kiểm thử tải và kiểm thử chức năng của các ứng dụng web và API. Việc tìm hiểu và sử dụng công cụ này giúp nhóm hiểu rõ cách thức hoạt động của nó và áp dụng nó vào các dự án kiểm thử phần mềm của mình.

Qua việc nghiên cứu và sử dụng các kỹ thuật kiểm thử được hỗ trợ bởi công cụ Loadfocus, nhóm em mong muốn nắm vững các phương pháp và tiến trình kiểm thử phần mềm, từ đó nâng cao khả năng đánh giá chất lượng phần mềm và tạo ra những ứng dụng tin cậy và tốt hơn.

1.2. Nội dung đề tài

Nội dung đề tài được chia làm 3 Phần:

PHẦN 1: MỞ ĐẦU

- Nêu lên lý do chọn đề tài
- Nội dung và kiến thức cần có để làm bài tập lớn

PHẦN 2: KẾT QUẢ NGHIÊN CỨU

- Tổng quan về kiểm thử phần mềm
 - + Sơ lược về kiểm thử phần mềm
 - + Nêu lên quy trình
 - + Các chiến lược kiểm thử phần mềm

- Tìm hiểu về công cụ kiểm thử LoadFocus
 - + Giới thiệu về công cụ
 - + Đặc điểm và cách sử dụng
- Giới thiệu về phần mềm cần được kiểm thử
 - + Giới thiệu về trang web của trường Đại học Công nghiệp Hà Nội
 - + Tài liệu đặc tả phần mềm
- Kiểm thử trang web trường Đại học Công nghiệp Hà Nội
 - + Sử dụng công cụ Load Testing
 - + Sử dụng công cụ Speed Monitor Testing
 - + Sử dụng công cụ Jmeter Testing

PHẦN 3: CÁC KIẾN THỨC NHẬN ĐƯỢC VÀ BÀI HỌC KINH NGHIỆM TÀI LIỆU THAM KHẢO

1.3. Kiến thức cần có

- *Kiến thức về kiểm thử phần mềm:* Hiểu về các khái niệm và phương pháp kiểm thử phần mềm, bao gồm kiểm thử chức năng, kiểm thử hiệu suất, kiểm thử phi chức năng và kiểm thử hộp đen/hộp trắng.
- *Kiến thức về quy trình phát triển phần mềm:* Hiểu về các bước và giai đoạn trong quy trình phát triển phần mềm, từ việc thu thập yêu cầu, thiết kế, lập trình cho đến kiểm thử và triển khai.
- *Kiến thức về công cụ kiểm thử Loadfocus:* Nắm vững các tính năng và chức năng của công cụ Loadfocus, bao gồm cách cài đặt, cấu hình và sử dụng các tính năng chính như kiểm thử hiệu suất, kiểm thử tải và kiểm thử chức năng.
- *Kiến thức về công cụ Jmeter:* Nắm vững các tính năng, cách sử dụng công cụ Jmeter để kiểm thử phần mềm.
- *Kỹ năng phân tích và ghi nhật ký:* Khả năng đọc hiểu tiếng anh tốt, phân tích các kết quả kiểm thử, lỗi và ghi nhật ký

PHẦN 2. KẾT QUẢ NGHIÊN CỨU

2.1. Tổng quan

2.1.1. Sơ lược về kiểm thử phần mềm

- Theo Glenford Myers, kiểm thử là quá trình vận hành chương trình để tìm ra lỗi.
- Kiểm thử phần mềm rất quan trọng vì nếu có bất kỳ lỗi hoặc lỗi nào trong phần mềm, nó có thể được xác định sớm và có thể được giải quyết trước khi chuyển giao sản phẩm phần mềm. Sản phẩm phần mềm được kiểm thử đúng cách đảm bảo độ tin cậy, bảo mật và hiệu suất cao, giúp tiết kiệm thời gian, hiệu quả chi phí và sự hài lòng của khách hàng.
- Mục tiêu của kiểm thử:
 - + Tìm ra được càng nhiều lỗi càng tốt trong điều kiện về thời gian đã định và nguồn lực sẵn có.
 - + Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó.
 - + Xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu.
 - + Thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, tiết kiệm được thời gian công sức.
- Trong kiểm thử có 7 nguyên tắc cơ bản:
 1. Kiểm thử chỉ ra sự hiện diện của lỗi trong phần mềm
 2. Kiểm thử tất cả các trường hợp là điều không thể
 3. Nên thực hiện kiểm thử càng sớm càng tốt
 4. Phân cụm của các lỗi
 5. Nghịch lý thuốc trừ sâu
 6. Kiểm thử theo các ngữ cảnh độc lập
 7. Sự sai lầm về việc không có lỗi
- Dữ liệu kiểm thử (test data): Dữ liệu cần cung cấp để phần mềm có thể thực thi để kiểm thử
- Kịch bản kiểm thử (test scenario): Các bước thực hiện khi kiểm thử
- Kỹ sư kiểm thử (tester): Người thực hiện kiểm thử

- Xác minh (Verification):

- + Là quy trình xác định xem sản phẩm của một công đoạn trong quy trình phát triển phần mềm có thỏa mãn các yêu cầu đặt ra trong công đoạn trước hay không.
- + Xác minh quan tâm tới việc ngăn chặn lỗi giữa các công đoạn
- + Xác minh thường là hoạt động kỹ thuật vì nó có sử dụng các kiến thức về các yêu cầu, các đặc tả rời rạc của phần mềm
- + Các hoạt động của xác minh bao gồm: Kiểm thử (Testing) và Rà soát loại (Review).

- Thẩm định (Validation):

- + Là tiến trình nhằm chỉ ra toàn bộ hệ thống đã phát triển xong phù hợp với tài liệu mô tả yêu cầu.
- + Thẩm định là quá trình kiểm chứng chúng ta xây dựng phần mềm có đúng theo yêu cầu khách hàng không
- + Thẩm định chỉ quan tâm đến sản phẩm cuối cùng không còn lỗi
- + Ca kiểm thử (test case):
 - + Chứa các thông tin cần thiết để kiểm thử thành phần phần mềm theo 1 mục tiêu xác định.
 - + Test case gồm bộ 3 thông tin: tập dữ liệu đầu vào, thứ tự thực hiện, tập kết quả kỳ vọng, các thông tin đó là:
 - Tập dữ liệu đầu vào (input): gồm các giá trị dữ liệu cần thiết để thành phần phần mềm dùng và xử lý.
 - Thứ tự thực hiện: các bước để hoàn thành ca kiểm thử từ lúc nhập dữ liệu đầu vào tới lúc nhận được kết quả đã qua xử lý của phần mềm.
 - Tập kết quả kỳ vọng (output): kết quả mong muốn sau khi thành phần phần mềm xử lý dữ liệu nhập.
 - Thiết kế các ca kiểm thử dựa trên thứ tự thực hiện các ca kiểm thử: Kiểm thử nối tầng và kiểm thử độc lập.

- Kiểm thử nối tầng:

- + Một ca kiểm thử này có thể được xây dựng dựa trên một ca kiểm thử khác.

- + Ưu điểm của phong cách này là mỗi ca kiểm thử sẽ trở nên nhỏ hơn và đơn giản hơn.
- + Nhược điểm là nếu một ca kiểm thử sai, sẽ dẫn tới ca kiểm thử xây dựng dựa trên ca kiểm thử đó sẽ sai theo
- Kiểm thử độc lập:
 - + Mỗi ca kiểm thử được xây dựng độc lập, không dựa vào các ca kiểm thử khác, và không đòi hỏi các ca kiểm thử khác phải thực hiện thành công.
 - + Ưu điểm của phong cách này là một ca kiểm thử có thể thực hiện bất cứ lúc nào, không phụ thuộc vào thứ tự thực hiện các ca kiểm thử.
 - + Nhược điểm chính là mỗi ca kiểm thử sẽ trở nên cồng kềnh và phức tạp hơn, và cũng làm cho quá trình thiết kế, thực hiện và bảo trì trở nên khó khăn hơn.

2.1.2. Quy trình kiểm thử phần mềm

- Quy trình kiểm thử phần mềm xác định các giai đoạn/ pha trong kiểm thử phần mềm. Về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:

Requirement analysis - Phân tích yêu cầu:

- + *Đầu vào*: đầu vào của giai đoạn phân tích yêu cầu bao gồm các tài liệu như: tài liệu đặc tả yêu cầu, tài liệu thiết kế hệ thống, tài liệu khách hàng yêu cầu về các tiêu chí chấp nhận của sản phẩm, bản prototype của khách hàng yêu cầu.
- + *Hoạt động*:
 - Phân tích yêu cầu là giai đoạn đầu tiên trong quy trình kiểm thử phần mềm.
 - QA team sẽ thực hiện đọc hiểu, nghiên cứu và phân tích cụ thể các yêu cầu trong tài liệu đặc tả của dự án hoặc tài liệu khách hàng.
 - Ngoài ra, trong quá trình phân tích, nghiên cứu tài liệu, nếu có câu hỏi phát sinh hay đề xuất giải quyết, QA team sẽ đưa ra câu hỏi với các bên liên quan như BA (Business Analysis), PM (Project Manager), team leader, khách hàng để hiểu chính xác hơn về yêu cầu của sản phẩm. Các câu hỏi nên được đưa ra dưới dạng Yes/No question hoặc các lựa chọn để tiết kiệm thời gian trả lời cũng như hỗ trợ đưa ra những gợi ý hay để xây dựng sản phẩm ngay từ đầu.

+ *Đầu ra*: Đầu ra của giai đoạn phân tích yêu cầu bao gồm tài liệu chứa các câu hỏi và câu trả lời liên quan đến nghiệp vụ của hệ thống, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.

Test planning - Lập kế hoạch kiểm thử

+ *Đầu vào*: Đầu vào của giai đoạn lập kế hoạch kiểm thử là các tài liệu đặc tả đã được cập nhật thông qua các câu hỏi và trả lời được đưa ra trong giai đoạn phân tích yêu cầu, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.

+ *Hoạt động*: Dựa vào các tài liệu được cung cấp và cập nhật mới nhất, thông thường, test manager hoặc test leader sẽ là người lập kế hoạch kiểm thử cho cả QA team. Lập kế hoạch kiểm thử nhằm xác định một số yếu tố quan trọng sau:

- Xác định phạm vi (Scope) dự án: Dự án thực hiện trong thời gian bao lâu? Bao gồm những công việc gì cho từng khoảng thời gian xác định?
- Xác định phương pháp tiếp cận: Nói về cách tiếp cận để kiểm thử cho một đối tượng nào đó, thì phải dựa vào nhiều thứ, Từ đó, test manager có thể đưa ra những phương pháp và kế hoạch phù hợp nhất cho cả quá trình thực hiện dự án sao cho đúng với các tiêu chí chấp nhận của sản phẩm và kịp tiến độ với các mốc thời gian bàn giao, phát hành.
- Xác định nguồn lực: Bao nhiêu người tham gia dự án, ai sẽ test phần nào, bao nhiêu tester tham gia? Tester và nhóm phát triển có kinh nghiệm về lĩnh vực này không? Số lượng server, version, máy tính, mobile để thực hiện test là bao nhiêu.
- Lên kế hoạch thiết kế công việc test: Các chức năng cần kiểm thử.
- Xác định điều kiện bắt đầu: xác định những điều kiện tối thiểu để bắt đầu hoạt động kiểm thử cho từng chức năng.
- Xác định điều kiện kết thúc: khi có những điều kiện nào thì sẽ kết thúc việc kiểm thử.

+ *Đầu ra*: Đầu ra của giai đoạn lập kế hoạch bao gồm các tài liệu như test plan, test estimation, test schedule.

Test case development - Thiết kế kịch bản kiểm thử

+ *Đầu vào*: Đầu vào của giai đoạn thiết kế kịch bản kiểm thử là test plan, test estimation, test schedule, các tài liệu đặc tả đã được cập nhật.

+ *Hoạt động*:

- Review tài liệu: Đầu tiên, các kiểm thử viên cần review lại tất cả các tài liệu để xác định công việc cần làm, các công việc có khác gì so với dự án trước khách hàng đưa cho, chức năng nào cần test, chức năng nào không cần test lại nữa. Từ đó, vừa có thể tiết kiệm thời gian mà vẫn đưa ra được một kịch bản kiểm thử đầy đủ và hiệu quả.

- Viết test case/ check list, chuẩn bị dữ liệu kiểm thử: Đội kiểm thử cần vận dụng các kỹ năng của mình để tạo ra test case cần thiết, và chuẩn bị dữ liệu cho việc kiểm thử.

- Review test case/ check list: Sau khi hoàn thành, các thành viên trong đội kiểm thử hoặc test leader cũng cần review lại test case đã tạo để có thể bổ sung

+ *Đầu ra*: Sau khi hoàn thành thiết kế kịch bản kiểm thử, đội kiểm thử sẽ có các tài liệu kiểm thử bao gồm: test design, test case, check list, test data, test automation script.

Test environment set up - Thiết lập môi trường kiểm thử

+ *Đầu vào*: Đầu vào của giai đoạn cài đặt môi trường kiểm thử là test plan, smoke test case, test data.

+ *Hoạt động*:

- Việc cài đặt môi trường kiểm thử là giai đoạn cũng rất quan trọng trong vòng đời phát triển phần mềm. Môi trường kiểm thử sẽ được quyết định dựa trên những yêu cầu của khách hàng, hay đặc thù của sản phẩm ví dụ như server/ client/ network, ...

- Tester cần chuẩn bị một vài test case để kiểm tra xem môi trường cài đặt đã sẵn sàng cho việc kiểm thử hay chưa. Đây chính là việc thực thi các smoke test case.

+ *Đầu ra*: Đầu ra của giai đoạn này là môi trường đã được cài đặt đúng theo yêu cầu, sẵn sàng cho việc kiểm thử và kết quả của smoke test case.

Test execution - Thực hiện kiểm thử

+ *Đầu vào*: Tài liệu đầu vào của giai đoạn này là test plan, test design, test case, check list, test data, test automation script.

+ *Hoạt động*:

- Thực hiện các test case như thiết kế và mức độ ưu tiên đã đưa ra trên môi trường đã được cài đặt.
- So sánh với kết quả mong đợi sau báo cáo các bug xảy ra lên tool quản lý lỗi và theo dõi trạng thái của lỗi đến khi được sửa thành công.
- Đo và phân tích tiến độ: Trong quá trình làm cũng cần theo dõi sát sao tiến độ, để tránh việc làm chậm tiến độ gây ảnh hưởng đến dự án.
- Report thường xuyên cho PM và khách hàng về tình hình thực hiện dự án: Cung cấp thông tin trong quá trình kiểm thử đã làm được những chức năng nào, còn chức năng nào, hoàn thành được bao nhiêu phần trăm công việc, báo cáo các trường hợp phát sinh sớm, tránh ảnh hưởng tiến độ công việc của cả ngày.

+ *Đầu ra*: Đầu ra của giai đoạn này là test results (kết quả kiểm thử), defect reports (danh sách các lỗi tìm được).

Test cycle closure - Đóng chu trình kiểm thử

+ *Đầu vào*: Đầu vào của giai đoạn đóng chu trình kiểm thử là bao gồm tất cả những tài liệu liên quan đã được tổng hợp, ghi chép và hoàn thiện đầy đủ trong suốt quy trình kiểm thử của dự án: tài liệu phân tích đặc tả yêu cầu, test plan, test results, defect reports, tài liệu Q&A, ...

+ *Hoạt động*:

- Đây là giai đoạn cuối cùng trong quy trình kiểm thử phần mềm.
- Ở giai đoạn này, QA team thực hiện tổng kết, báo cáo kết quả về việc thực thi test case, bao nhiêu case pass/ fail, bao nhiêu case đã được fix, mức độ nghiêm trọng của lỗi, bao nhiêu lỗi cao/ thấp, lỗi còn nhiều ở chức năng nào, dev nào nhiều lỗi. Chức năng nào đã hoàn thành test/ chưa hoàn thành test/ trễ tiến độ bàn giao.

- Ngoài ra, giai đoạn này cũng thảo luận tất cả những điểm tốt, điểm chưa tốt và rút ra bài học kinh nghiệm cho những dự án sau, giúp cải thiện quy trình kiểm thử.

+ *Đầu ra*: Đầu ra của giai đoạn này bao gồm các tài liệu: Test report, Test results (final)

Các giai đoạn kiểm thử được thực hiện một cách tuần tự. Mỗi giai đoạn sẽ có những mục tiêu khác nhau, đầu vào và kết quả đầu ra khác nhau nhưng mục đích cuối cùng vẫn là đảm bảo chất lượng sản phẩm phần mềm tốt nhất.

2.1.3. Các phương pháp kiểm thử phần mềm

2.1.3.1. Kiểm thử hộp đen

- Kiểm thử hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.

- *Đối tượng được kiểm thử*: Là thành phần phần mềm (TPPM) có thể là 1 hàm chức năng, 1 module chức năng, 1 phân hệ chức năng...

- *Phương pháp thử nghiệm*: Dựa vào chức năng Kiểm thử hộp đen (Black box test) có thể được áp dụng hầu như đến mọi cấp độ của kiểm thử phần mềm:

- + Kiểm thử đơn vị (Unit test)
- + Kiểm thử tích hợp (Integration test)
- + Kiểm thử hệ thống (System test)
- + Kiểm thử chấp nhận (Acceptance test).

- *Đặc điểm*:

- + Là chiến lược kiểm thử TPPM dựa vào thông tin duy nhất là các đặc tả về yêu cầu chức năng của TPPM tương ứng.

- + Người kiểm thử không cần thiết phải có kiến thức về việc mã hoá, cấu trúc bên trong của TPPM, cũng như không yêu cầu phải biết lập trình phần mềm.

+ Việc kiểm thử được tiến hành dựa vào việc kiểm thử TPPM làm được gì, có phù hợp với yêu cầu của người dùng hay không. Các tester nhập số liệu vào phần mềm và chỉ cần xem kết quả của phần mềm và các mục tiêu kiểm tra.

+ Tester phải viết test case đầy đủ trước khi test; khi test, đơn giản chỉ cần thực hiện theo các bước mô tả trong test case thao tác và nhập data vào, sau đó xem kết quả trả về hoặc hành vi của phần mềm, rồi so sánh với kết quả mong đợi được viết trong testcase.

+ Khi viết test case: Dựa vào yêu cầu và giao diện bên ngoài của chương trình (Không can thiệp vào bên trong code của chương trình)

+ Khi thực hiện test: Thực hiện trên giao diện của chương trình (yêu cầu chương trình phải chạy được mới test được, không can thiệp vào code)

2.1.3.2. Kiểm thử hộp trắng

- Kiểm thử hộp trắng (While box test) là phương pháp thử nghiệm phần mềm, trong đó các thiết kế, cấu trúc giải thuật bên trong, và việc thực hiện các công việc đều được biết đến

- *Đối tượng kiểm thử*: Là 1 thành phần của phần mềm (1 chức năng, 1 module chức năng, 1 phân hệ chức năng....)

- *Phương pháp thử nghiệm*: Dựa vào thuật giải Kiểm thử hộp trắng dựa vào thuật giải cụ thể, vào cấu trúc dữ liệu bên trong của đơn vị phần mềm cần kiểm thử để xác Định đơn vị phần mềm đó có thực hiện đúng không. Kỹ thuật white box test thích hợp dùng để kiểm thử đơn vị (Unit test)

- *Đặc điểm*

+ Là chiến lược kiểm thử TPPM dựa vào giải thuật, cấu trúc bên trong chức năng của TPPM tương ứng.

+ Người kiểm thử phải có kiến thức nhất định về việc mã hoá, cấu trúc bên trong của chức năng, biết lập trình phần mềm.

+ Việc kiểm thử được tiến hành dựa vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không.

+ Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code; khi test, sẽ set điều kiện và data để chạy vào đủ tất cả các nhánh trong giải thuật, đảm bảo thực hiện đầy đủ.

+ Khi viết test case: Dựa vào yêu cầu và nội dung Source Code (can thiệp vào bên trong Code của chương trình)

Khi thực hiện test: Thực thi test trong code (không cần thực thi chương trình, vì thực hiện test white box sẽ sử dụng framework nào đó hỗ trợ (Ví dụ như test kiểu debug)

Trong kiểm tra này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật giải trong thành phần phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử

2.2. Công cụ kiểm thử LoadFocus

2.2.1. Giới thiệu công cụ

Kiểm thử tự động: Là xử lý một cách tự động các bước thực hiện các testcase, kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

Kiểm thử tự động: là một kỹ thuật tự động trong đó người kiểm thử tự viết các tập lệnh và sử dụng phần mềm phù hợp để kiểm thử phần mềm. Nó về cơ bản là một quá trình tự động hóa của một quy trình kiểm thử thủ công. Giống như kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để kiểm thử ứng dụng theo quan điểm tải, hiệu năng và ứng suất. Kiểm thử tự động giúp giảm chi phí kiểm thử bằng cách hỗ trợ quá trình kiểm thử thông qua các công cụ phần mềm.

Kiểm thử tự động hay sử dụng phần mềm để kiểm thử với các ưu điểm:

- Có thể thực hiện các kiểm thử một cách liên tục, lặp lại và giảm chi phí cho nhân lực kiểm thử.
- Luôn đảm bảo hoạt động theo một kịch bản duy nhất – không bị ảnh hưởng như với kiểm thử viên.

Công cụ kiểm thử LoadFocus là nền tảng thử nghiệm đám mây tất cả trong một dành cho các trang web và API với tính năng kiểm tra tải, chức năng kiểm tra tải Apache JMeter và theo dõi tốc độ trang.

2.2.2. Đặc điểm

Công cụ kiểm thử LoadFocus là một công cụ mã nguồn mở. LoadFocus hỗ trợ 3 loại kiểm thử chủ yếu :

1. LoadTesting

- Load Testing là một loại kiểm thử hiệu năng được sử dụng để đánh giá hiệu suất của một trang web, ứng dụng hoặc giao diện lập trình ứng dụng (API) dưới điều kiện tải bình thường và tải cao. Mục đích của Load Testing là xác định hành vi và thời gian phản hồi của hệ thống khi phải xử lý số lượng người dùng hoặc yêu cầu tải lớn, đảm bảo tính ổn định, đáng tin cậy và khả năng mở rộng của nó.
- Kiểm thử tải bao gồm mô phỏng một lượng lớn người dùng đồng thời, yêu cầu hoặc giao dịch để đánh giá hiệu suất hệ thống và xác định các vấn đề về hiệu suất hoặc hạn chế tiềm ẩn. Bằng cách tạo ra các loại tải khác nhau, kiểm thử tải có thể đo lường các thông số như thời gian phản hồi, khả năng xử lý, tài nguyên sử dụng và tính ổn định của hệ thống dưới các mức tải khác nhau.
- Kiểm thử tải có thể được áp dụng cho các trang web, ứng dụng web, ứng dụng di động, API hoặc bất kỳ hệ thống nào tương tác với người dùng hoặc xử lý số lượng lớn dữ liệu. Nó giúp phát hiện các vấn đề liên quan đến hiệu suất như thời gian phản hồi chậm, rò rỉ bộ nhớ, vấn đề cơ sở dữ liệu, yếu tố cấu hình yếu kém hoặc phân bổ tài nguyên không đủ.
- Quá trình kiểm thử Load Testing thông thường bao gồm các bước sau:
 1. Lập kế hoạch kiểm thử
 2. Thiết kế kiểm thử
 3. Thực hiện kiểm thử
 4. Phân tích hiệu suất
 5. Tối ưu hóa và kiểm thử lại
- Kiểm thử Load Testing mang lại nhiều lợi ích:
 - + Xác nhận hiệu suất: Kiểm thử tải giúp xác nhận các đặc điểm hiệu suất của một hệ thống, đảm bảo thời gian phản hồi và khả năng xử lý theo yêu cầu trong các tình huống tải khác nhau.

+ Đánh giá khả năng mở rộng: Bằng cách mô phỏng các tải tăng dần, kiểm thử tải giúp đánh giá khả năng mở rộng và khả năng xử lý yêu cầu người dùng gia tăng. Nó giúp xác định khả năng tối đa của hệ thống và quyết định khi nào cần thêm tài nguyên hoặc nâng cấp cơ sở hạ tầng.

+ Xác định điểm hạn chế: Kiểm thử tải xác định các vấn đề về hiệu suất, chẳng hạn như các truy vấn cơ sở dữ liệu chậm, thuật toán không hiệu quả hoặc vấn đề về độ trễ mạng. Thông tin này giúp nhà phát triển tối ưu hóa hệ thống để cải thiện hiệu suất.

+ Tính đáng tin cậy và ổn định: Kiểm thử tải phát hiện các vấn đề về ổn định có thể xảy ra khi tải nặng, chẳng hạn như rò rỉ bộ nhớ, sự cố hoặc việc sử dụng tài nguyên quá mức. Bằng cách xác định và khắc phục các vấn đề này, hệ thống trở nên đáng tin cậy và ổn định hơn.

2. Jmeter Load Testing:

- JMeter Load Testing (Kiểm thử tải bằng JMeter) là một phương pháp kiểm thử hiệu năng sử dụng công cụ mã nguồn mở Apache JMeter để đánh giá hiệu suất và khả năng mở rộng của một ứng dụng web hoặc API. JMeter được sử dụng rộng rãi cho kiểm thử tải vì nó cho phép người dùng tạo và thực thi các kịch bản kiểm thử để mô phỏng các hành động của người dùng và tạo ra một lượng lớn yêu cầu.

- Một tính năng của JMeter Load Testing là khả năng chạy các kịch bản kiểm thử từ nhiều địa điểm khác nhau. Tính năng này có lợi khi đánh giá hiệu suất và thời gian phản hồi của ứng dụng từ các địa điểm địa lý khác nhau, cung cấp thông tin về cách ứng dụng hoạt động dưới các điều kiện mạng và địa lý đa dạng.

- Dưới đây là một tổng quan về cách JMeter Load Testing hoạt động:

+ Tạo kịch bản kiểm thử: Trong JMeter, bạn có thể tạo các kịch bản kiểm thử mô phỏng các hành động của người dùng như điều hướng qua các trang web, gửi biểu mẫu hoặc tương tác với các API. Các kịch bản này xác định hành vi người dùng và tải mà bạn muốn tạo ra trong quá trình kiểm thử.

+ Cấu hình kiểm thử: Trong JMeter, bạn có thể cấu hình các tham số khác nhau như số lượng người dùng đồng thời, thời gian tăng dần và thời gian thực thi kiểm thử. Các thiết lập này xác định mức tải và thời lượng kiểm thử.

+ Kiểm thử phân tán: JMeter cho phép phân tán tải trên nhiều máy chủ, cho phép bạn chạy các kịch bản kiểm thử từ các địa điểm khác nhau cùng một lúc. Mỗi máy chủ đại diện cho một địa điểm từ đó tải được tạo ra.

+ Cấu hình mạng: Để thực thi kiểm thử từ nhiều địa điểm, bạn cần cấu hình mạng một cách phù hợp. Điều này bao gồm thiết lập cơ sở hạ tầng mạng để đảm bảo giao tiếp giữa các phiên bản JMeter đang chạy trên các máy chủ khác nhau.

+ Thực thi kiểm thử: Sau khi kịch bản kiểm thử và cấu hình mạng đã sẵn sàng, bạn có thể bắt đầu kiểm thử tải. JMeter sẽ thực thi kịch bản từ mỗi địa điểm, mô phỏng các hành động của người dùng và tạo ra mức tải mong muốn trên ứng dụng hoặc API mục tiêu.

+ Phân tích kết quả: Sau khi kiểm thử tải hoàn thành, JMeter cung cấp các chỉ số hiệu suất và kết quả chi tiết. Các chỉ số này bao gồm thời gian phản hồi, công suất xử lý, tỷ lệ lỗi và sử dụng tài nguyên. Bạn có thể phân tích các kết quả này để xác định các vấn đề về hiệu suất, hạn chế hệ thống và vùng cần tối ưu.

3. *Page Speed Monitoring:*

- Page speed monitoring là quá trình liên tục theo dõi và đo lường hiệu suất và tốc độ tải trang của một trang web. Nó liên quan đến việc theo dõi các chỉ số hiệu suất khác nhau và thu thập thông tin để tối ưu tốc độ trang web và cải thiện trải nghiệm người dùng. Một công cụ thông dụng được sử dụng để Giám sát Tốc độ Trang là công cụ kiểm thử Loadfocus

- Dưới đây là tổng quan về cách Page speed Monitoring hoạt động:

+ Performance Metrics: Loadfocus đo đạc một số chỉ số hiệu suất quan trọng của trang web, bao gồm thời gian tải trang, first contentful paint (FCP), largest contentful paint

(LCP), thời gian tương tác (TTI) và nhiều hơn nữa. Những chỉ số này cung cấp thông tin về tốc độ tải trang và tương tác của trang đối với người dùng.

+ Test Execution: Để giám sát tốc độ trang bằng Loadfocus, bạn có thể chạy kiểm tra trên các trang web cụ thể hoặc thiết lập giám sát tự động để đánh giá hiệu suất theo thời gian. Loadfocus báo cáo chi tiết với điểm hiệu suất và thông tin hành động cho mỗi trang đã được kiểm tra.

+ Insights and Recommendations: cung cấp nhiều thông tin và đề xuất dựa trên các kiểm tra đã thực hiện. Những thông tin này giúp xác định các vấn đề về hiệu suất, vấn đề truy cập và các lĩnh vực mà các tiêu chuẩn tốt nhất có thể được thực hiện.

+ Historical Performance Tracking: Bằng cách thường xuyên giám sát trang bằng LoadFocus, bạn có thể theo dõi hiệu suất lịch sử của trang web của mình theo thời gian.

- Lợi ích của giám sát tốc độ trang bằng Loadfocus bao gồm:

+ Cải thiện Trải nghiệm Người dùng: Bằng cách giám sát tốc độ trang và thực hiện các tối ưu hóa được đề xuất bởi Lighthouse, bạn có thể cung cấp một trải nghiệm nhanh hơn và mượt mà cho người truy cập trang web của bạn. Thời gian tải trang nhanh giúp giảm tỷ lệ thoát, tăng cường tương tác và cải thiện sự hài lòng của người dùng.

+ Tối ưu hóa Công cụ tìm kiếm (SEO): Tốc độ trang là một yếu tố quan trọng trong thứ hạng trên công cụ tìm kiếm. Giám sát và cải thiện tốc độ trang bằng Lighthouse giúp đảm bảo trang web của bạn đáp ứng các yêu cầu hiệu suất của các công cụ tìm kiếm, từ đó cải thiện khả năng tìm thấy và thứ hạng trên kết quả tìm kiếm.

+ Tối ưu hóa Hiệu suất: Lighthouse cung cấp thông tin chi tiết và đề xuất để tối ưu hiệu suất của trang web của bạn. Bằng cách giải quyết các vấn đề được xác định, chẳng hạn như giảm kích thước tệp, tối ưu hóa việc tải tài nguyên và cải thiện chiến lược bộ nhớ cache, bạn có thể cải thiện đáng kể tốc độ và hiệu suất tổng thể của trang web.

2.2.3. Cài đặt và sử dụng công cụ LoadFocus

- Công cụ kiểm thử LoadFocus là một nền tảng thử nghiệm đám mây tất cả trong một dành cho các trang web và API. Công cụ được xây dựng trang web có tên là LoadFocus.com
- Công cụ LoadFocus có thể được truy cập với mọi thiết bị với điều kiện thiết bị đó có kết nối mạng.
- Cách sử dụng cũng rất đơn giản: Người dùng chỉ cần truy cập vào trang loadFocus.com sau đó đăng ký 1 tài khoản. Sau đó chọn 1 bài test cho trang web của mình, cập nhật các thông số. Nhập địa chỉ các trang web và bắt đầu. Sau đó LoadFocus sẽ tự động chạy các bài test.

2.3. Giới thiệu trang web www.hau.edu.vn

2.3.1. Giới thiệu

- Là trang web chính thức của trường Đại học công nghiệp Hà Nội.
- Là trang web tổng hợp các thông tin bài báo, thông báo của trường, thư viện số, các tiện ích, dịch vụ của trường Đại học Công nghiệp Hà Nội, mục đích giúp cho giảng viên, sinh viên được tiếp cận với các thông tin này 1 cách nhanh chóng cũng như thuận tiện nhất.

2.3.2. Tài liệu đặc tả yêu cầu phần mềm

- *Mục đích*: Trang web được xây dựng với mục đích cập nhật tin tức, lịch học, thông báo và quản lý các tài nguyên của nhà trường
- *Phạm vi*: Website sẽ được truy cập qua trình duyệt web trên máy tính và các thiết bị di động.
- *Đối tượng sử dụng*: Giảng viên, sinh viên trường Đại học Công nghiệp Hà Nội
- *Yêu cầu về chức năng*: Trang web có đầy đủ tất cả các chức năng thiết yếu mà giảng viên và sinh viên cần.
- *Yêu cầu hiệu năng* :

+ Thời gian phản hồi: Website cần đáp ứng nhanh chóng khi người dùng thực hiện tìm kiếm, xem thông tin sản phẩm và đặt hàng.

+ Tải cao: Trang web có thể chịu tải được số lượng người dùng truy cập vào cùng 1 lúc lớn.

+ Độ tin cậy: Website cần hoạt động ổn định và không có lỗi khi có đồng thời hàng ngàn người dùng truy cập

- *Yêu cầu giao diện người dùng*

+ Giao diện đơn giản, dễ sử dụng và thân thiện với người dùng

+ Hiện thị danh sách, danh mục rõ ràng, rành mạch, dễ tìm kiếm và dễ sử dụng cho người mới

+ Cung cấp các bộ lọc và tùy chọn sắp xếp để giúp người dùng tìm kiếm dễ dàng hơn.

+ Giao diện đáp ứng để hiển thị tốt trên các thiết bị di động.

2.4. Kiểm thử trang web www.hau.edu.vn

2.4.1. Lập kế hoạch kiểm thử

Người thực hiện	Test case	Chức năng	Hành động	Kết quả mong muốn	Kết quả thực tế
Hồ Nam Tú	TC01	Kiểm tra tải	+ Tạo lệnh Jmeter + Tải tệp lệnh lên LoadFocus + Kiểm tra tải	Đáp ứng được tải cao	
Nguyễn Viết Vượng	TC02	Kiểm tra tải	+ Kiểm tra tải	Đáp ứng được tải cao	
Phan Anh Thắng	TC03	Kiểm tra tốc độ trang	+ Kiểm tra tốc độ tải trang	Thời gian tải nhanh	

Phương pháp kiểm thử :

- Kiểm thử hiệu năng :
 - + Kiểm tra tải Jmeter (Load Testing) của Load focus đối với test case TC01 (Hồ Nam Tú).
 - + Kiểm tra tải (Nguyễn Viết Vượng) của Load focus đối với test case TC02 (Nguyễn Viết Vượng).
 - + Kiểm thử tốc độ tải trang (Speed monitor tetsing) của Load focus đối với test case TC03 (Phan Anh Thắng).

2.4.2. Kiểm tra tải Jmeter (Hồ Nam Tú)

2.4.2.1. Phân tích thiết kế kiểm thử

- **Giao diện chức năng**



- **Thanh điều hướng:** Thanh điều hướng của trang web nằm ở bên trên cùng của trang web. Gồm các chỉ mục.
- **Nội dung chính:** Mô tả nội dung chính của trang web, bao gồm các phần tử như văn bản, hình ảnh, video, bảng, biểu đồ và các thành phần tương tác khác. Nội dung chính thường được hiển thị ở giữa hoặc ở phần trung tâm của trang web.
- **Chân trang:** Chân trang có thể chứa các liên kết bổ sung, thông tin bản quyền, liên hệ, chính sách, số điện thoại, địa chỉ liên hệ và các thông tin khác.
- Là 1 trang web của trường đại học. Có số lượng truy cập lớn. Nên bài kiểm tra sẽ sử dụng số lượng người dùng ảo truy cập vào trang web là lớn để kiểm tra trang web

- Các điều kiện ràng buộc :

+ **Ràng buộc quyền truy cập:** Tất cả người dùng đều có thể truy cập trang web

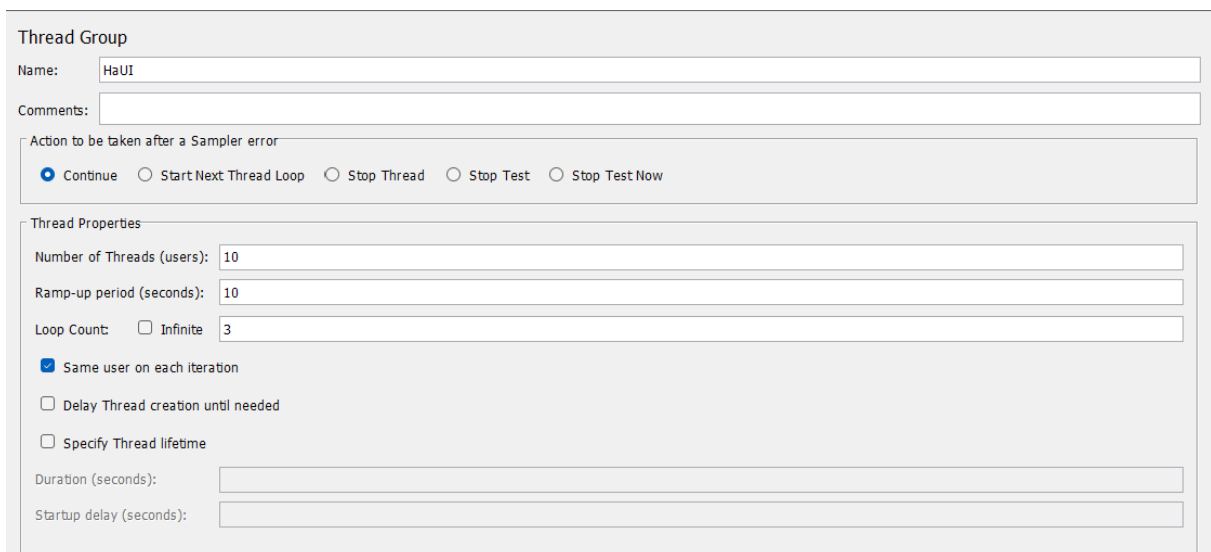
+ **Ràng buộc tương tác:** Mô tả các tương tác giữa các phần tử trong giao diện. Ví dụ, khi nhấn vào một nút "Gửi", dữ liệu từ các trường nhập liệu phải được gửi đi.

+ **Ràng buộc định dạng:** Đảm bảo giao diện tuân theo các quy tắc thiết kế và định dạng.

+ **Ràng buộc hiệu suất:** Có thể chịu được tải cao trong trường hợp có nhiều người truy cập

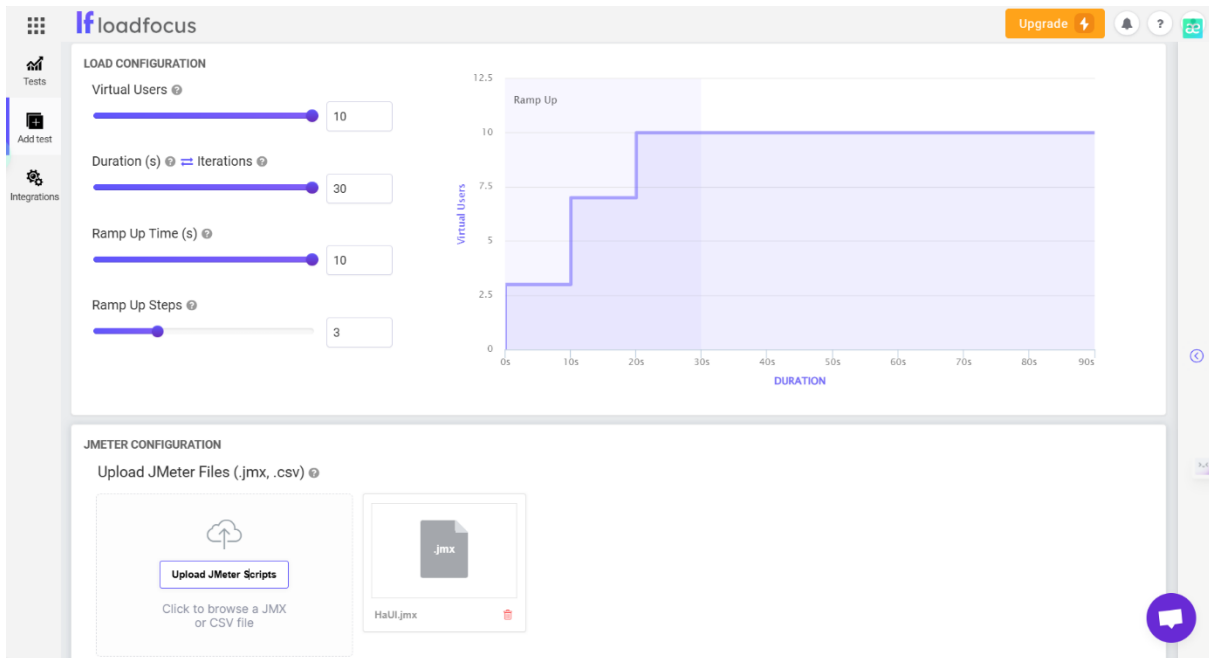
2.4.2.2. Thực hiện kiểm thử

Bước 1: Thực hiện setup cấu hình cho tệp Jmeter, số lượng người dùng ảo, Vòng lặp, Ramp-up

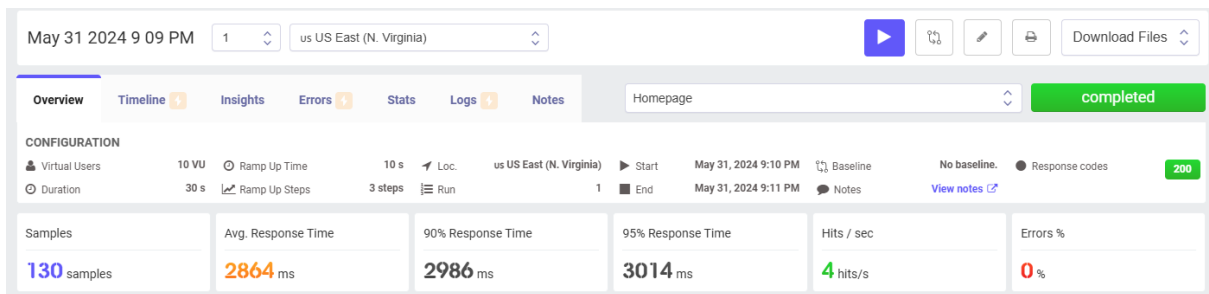


The screenshot shows the configuration for a Thread Group in JMeter. The 'Name' field is set to 'HaUI'. The 'Comments' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users)' set to 10, 'Ramp-up period (seconds)' set to 10, 'Loop Count' set to 3 (with 'Infinite' unchecked), 'Same user on each iteration' checked, 'Delay Thread creation until needed' unchecked, and 'Specify Thread lifetime' unchecked. The 'Duration (seconds)' and 'Startup delay (seconds)' fields are empty.

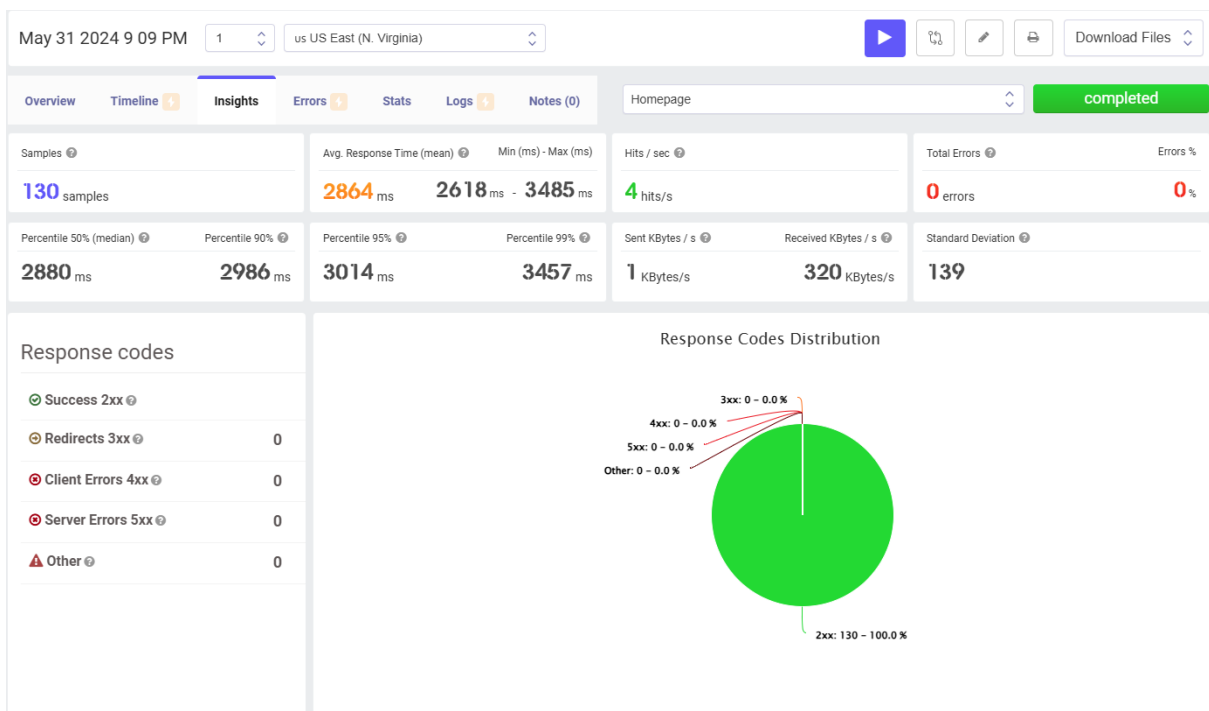
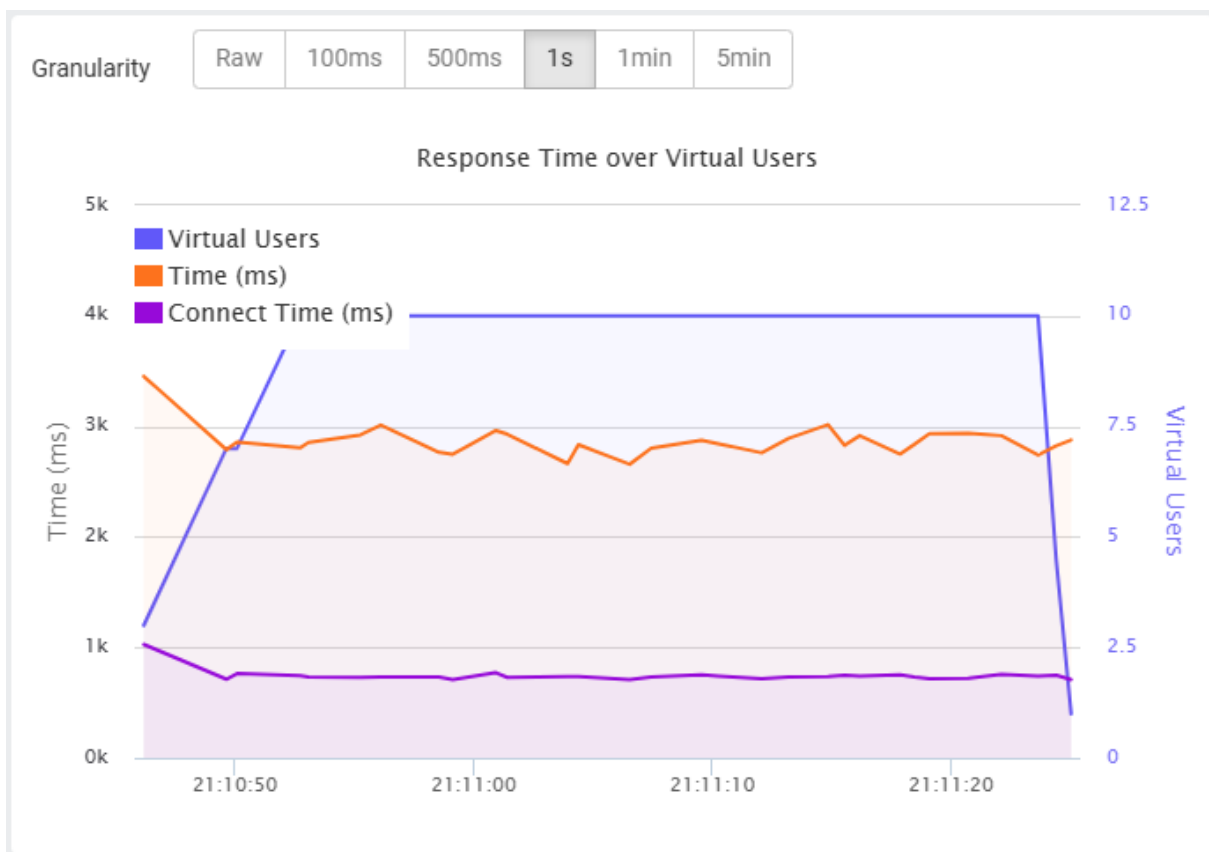
Bước 2: Tạo 1 HTTP Request Default với những thông số đơn giản như sau



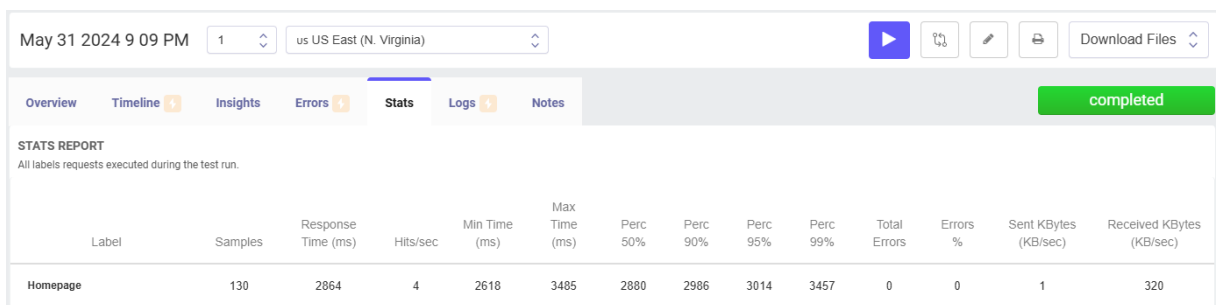
Bước 5: Bấm << Run Test >> và chờ kết quả



Bước 6: Tại mục OverView. Có thể xem biểu đồ kết quả



- Tại mục Insight có thể xem thông tin về các mã lỗi phản hồi. và tại mục Error có thể xem thông tin về các lỗi (Nếu có). Ngoài ra tại mục Log. Người dùng có thể xem thông tin về tệp .jmx (Jmeter) được tạo ra bởi LoadFocus.



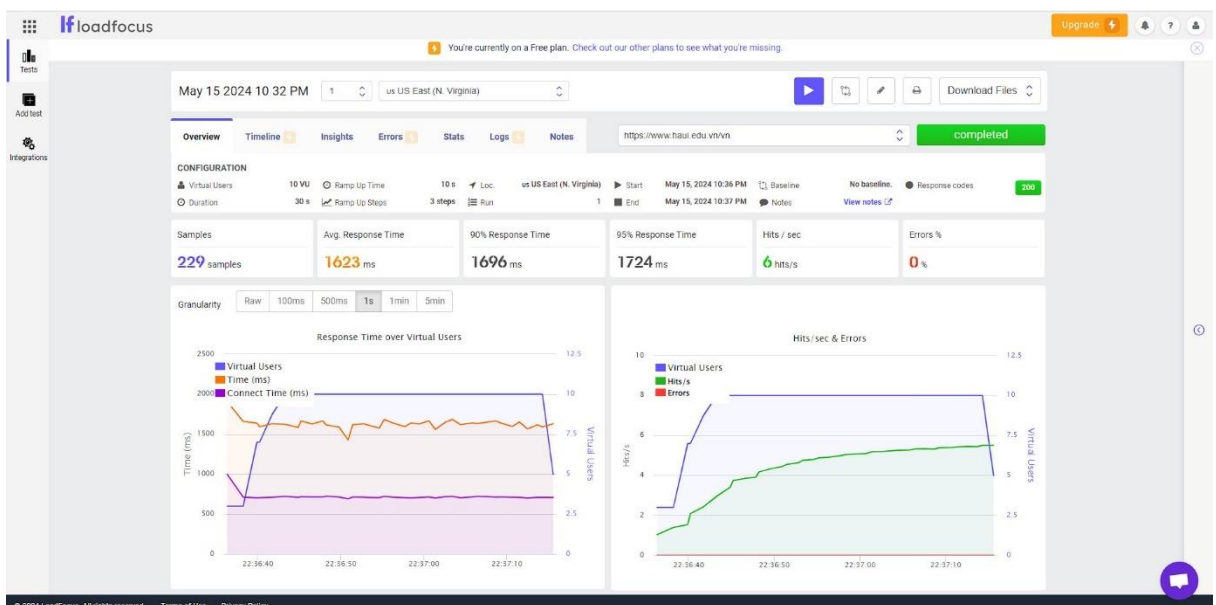
- Ngoài ra tại mục số liệu thống kê, Người dùng có thể xem thông tin tổng quan nhất về hoạt động kiểm thử

2.4.3. Kiểm tra tải Load Testing (Nguyễn Viết Vượng)

2.4.3.1. Phân tích thiết kế kiểm thử

Nội dung ca kiểm thử: Khởi tạo một số lượng user ảo (ở đây đang thực hiện là 10 user ảo) đồng thời cùng truy cập vào trang web, xem tốc độ phản hồi và hoạt động của trang web.

- Dưới đây là màn hình tổng quan sau khi chạy xong ca kiểm thử.



- Phân tích các thông số có trong bản báo cáo:

1. Virtual users: Số lượng người dùng ảo được khởi tạo

2. Duration: Tổng thời gian chạy, thời gian mà người dùng sẽ truy cập vào trang web.

3. Ramp Up Time: Độ trễ giữa lúc bắt đầu kiểm tra cho đến khi tất cả người dùng ảo đang chạy.

4. Ramp up steps: Số bước cho thời gian tăng tốc

5. Location: Nơi server chạy

6. Run: Lần bắt đầu

7. Start Time: Thời gian bắt đầu ca kiểm thử

8. End Time: Thời gian kết thúc ca kiểm thử

9. Baseline: Đường cơ sở

10. Notes: Chú thích

11. Response codes: Mã phản hồi

12. Samples: Tổng số lượng requests đã gửi đến server trong thời gian gian chạy ca kiểm thử

13. Avg. Response: là thời gian trung bình, khoảng thời gian tính từ thời điểm người dùng gửi yêu cầu cho đến thời điểm ứng dụng báo rằng yêu cầu đã hoàn thành. Giá trị trung bình số học cho tất cả các phản hồi (tổng của tất cả thời gian/số lượng phản hồi)

14. 90% Response Time: 90% thời gian hồi hồi hoặc phân vị thứ 90 là giá trị mà 90% phản hồi (điểm dữ liệu) nhỏ hơn và 10% lớn hơn

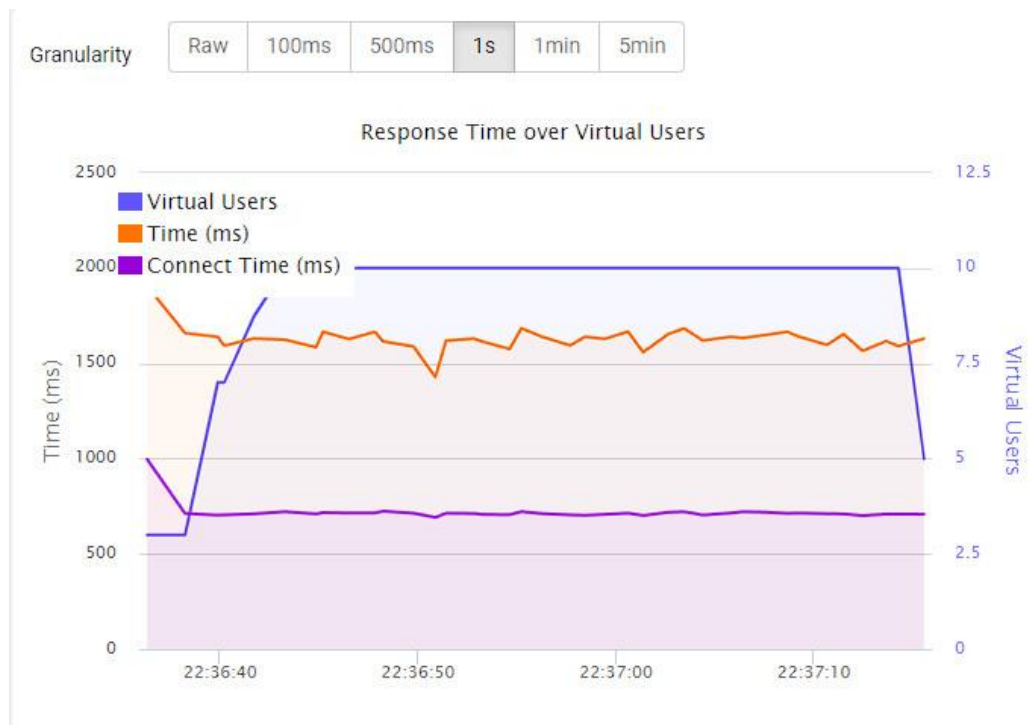
15. 95% Response Time: 95% thời gian hồi hồi hoặc phân vị thứ 95 là giá trị mà 95% phản hồi (điểm dữ liệu) nhỏ hơn và 5% lớn hơn

16. Hits/ sec: Lượt truy cập /s cho biết số lượt truy cập mà ứng dụng phải chịu mỗi. giá trị trung bình sẽ được hiển thị

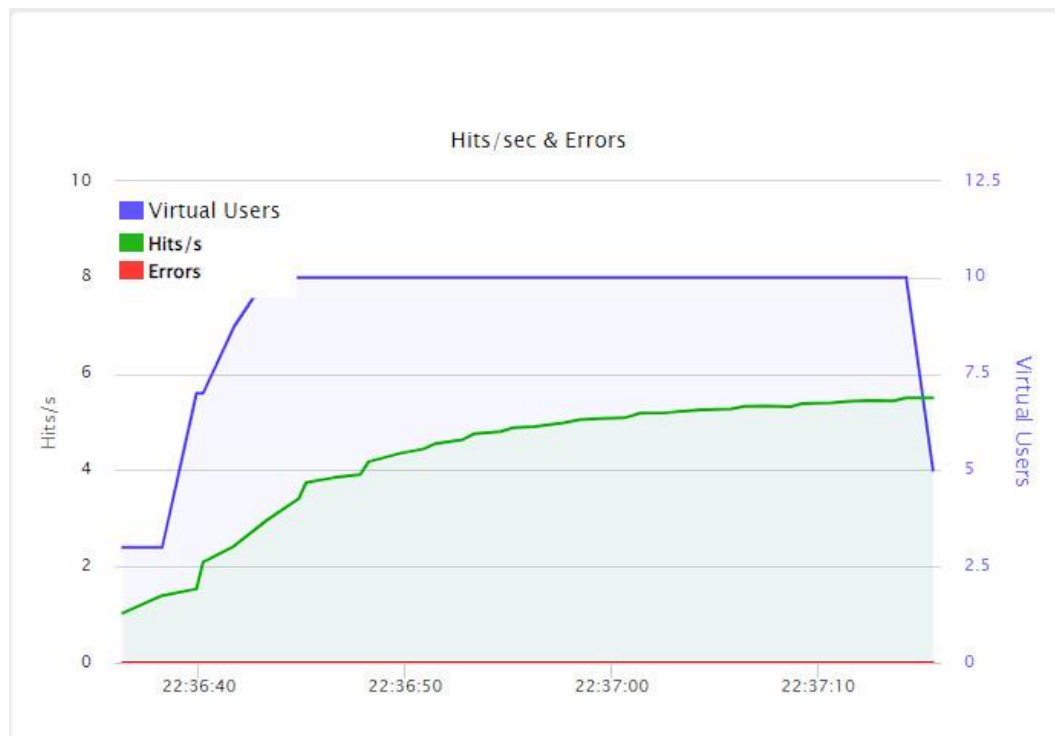
17. Errors: Tổng số lỗi gặp phải khi chạy kiểm tra đối với URL đã nhập (mã phản hồi 4xx và 5xx hoặc các lỗi kết nối khác). Tổng giá trị hiển thị

18. Granularity: là độ chi tiết, để hiển thị trên biểu đồ tất cả các thử nghiệm. Nếu không, các giá trị trung bình sẽ được hiển thị trên biểu đồ, được tính trung bình cứ sau 100 milis, 500 milis, 1s, 1 phút hoặc 5 phút.

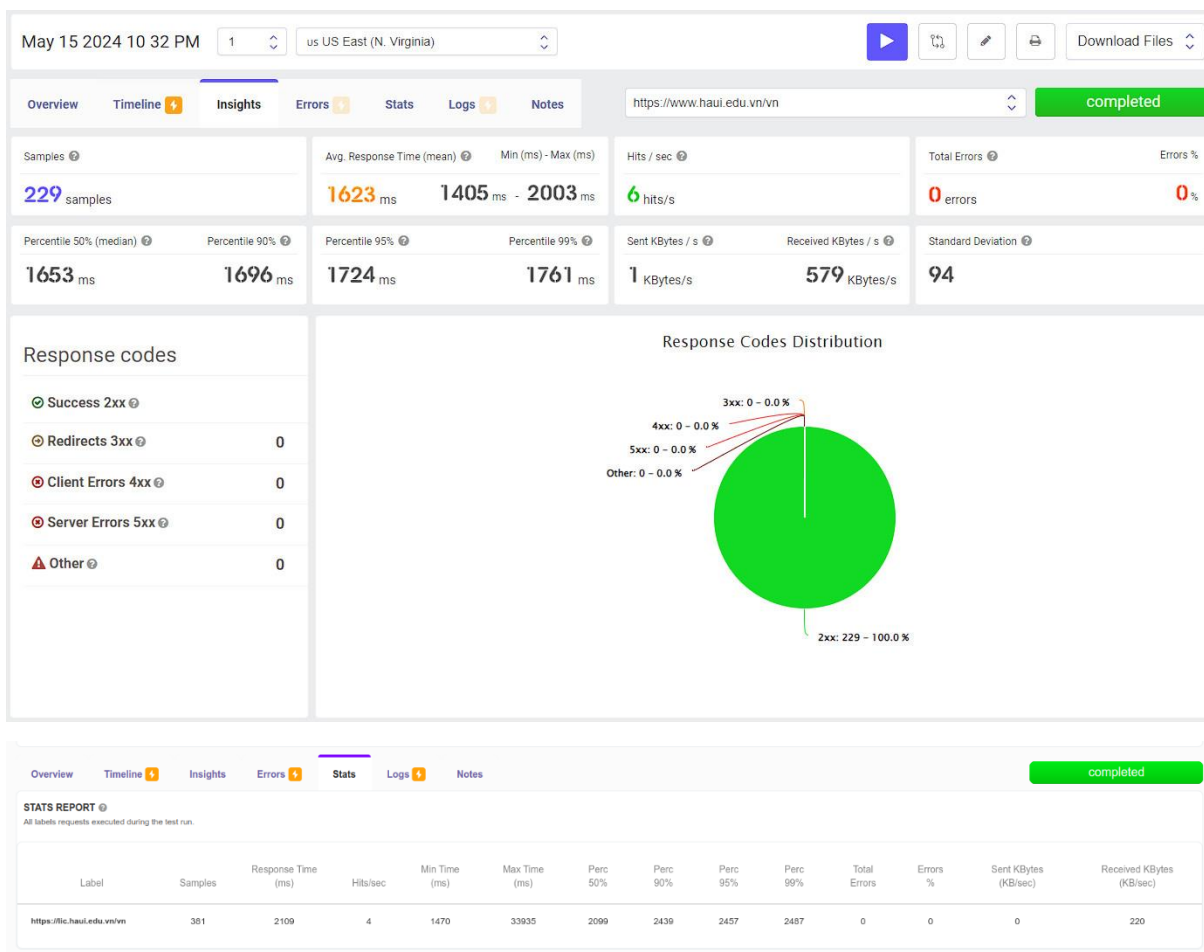
- Đây là đồ thị thể hiện mức độ thời gian phản hồi với số lượng user truy cập.



- Đây là đồ thị thể hiện mức độ lượt truy cập của user và mức độ lỗi theo thời gian.



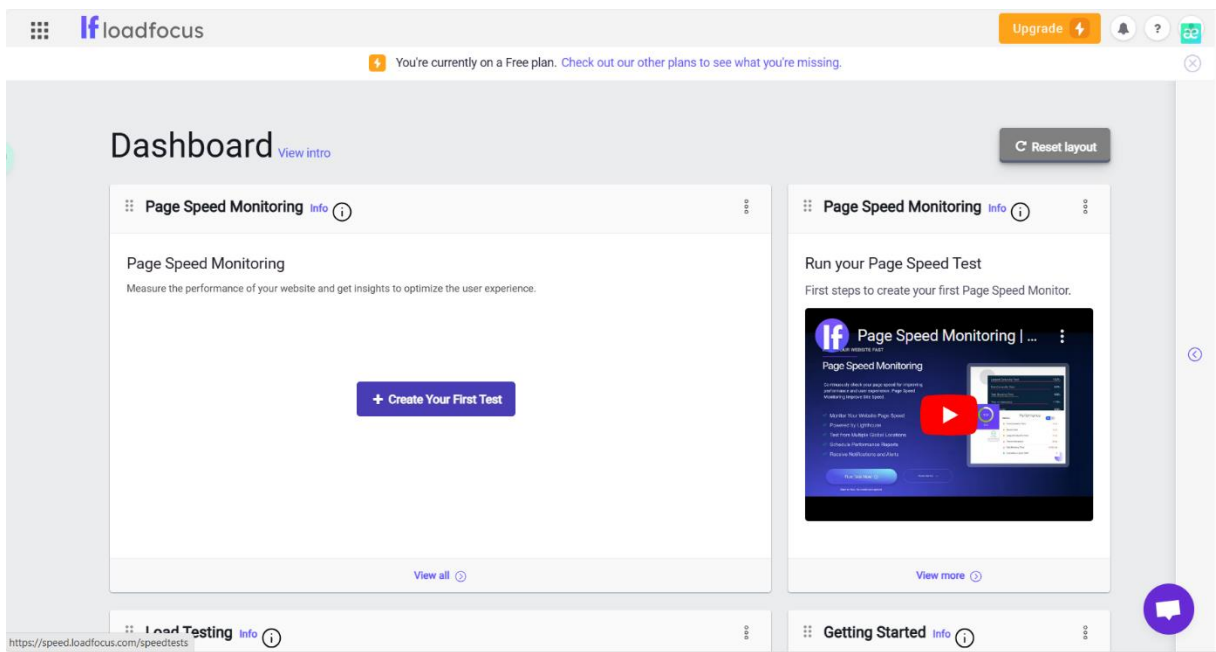
- Ngoài ra công cụ kiểm thử LoadFocus cho kiểm tra tải Load Testing còn cho phép người dùng xem chi tiết các thông số cụ thể ngoài mục overview.



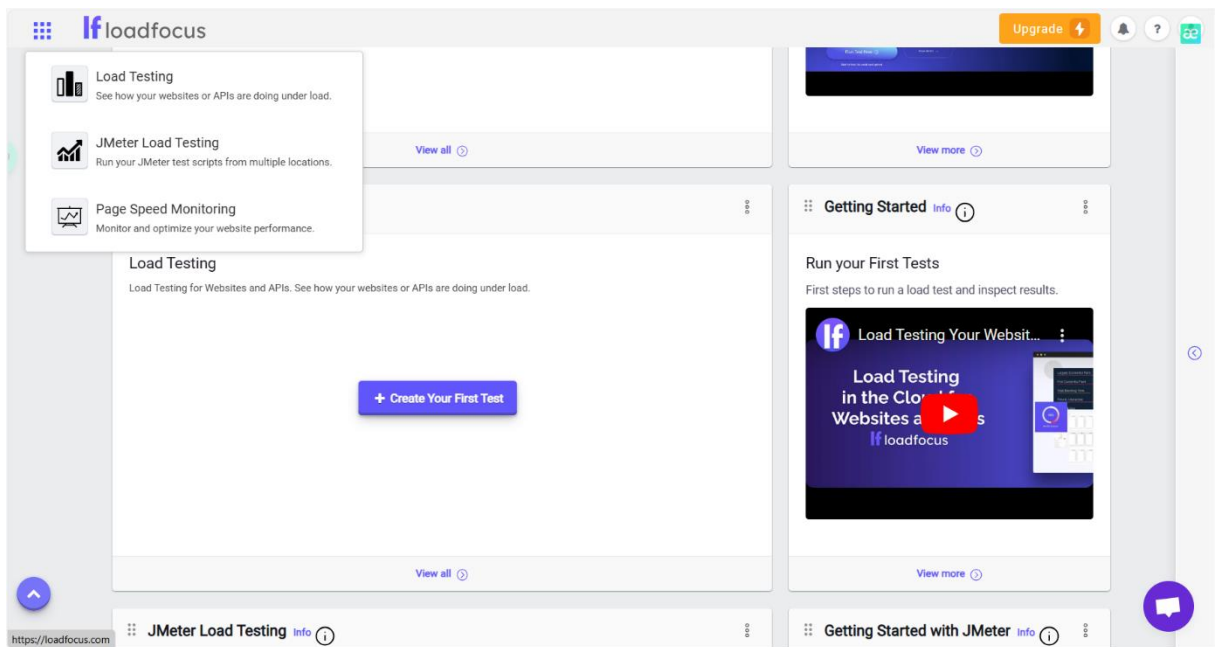
2.4.3.2. Thực hiện kiểm thử

Đầu tiên tạo tài khoản công cụ kiểm thử LoadFocus trên trang web loadfocus.com.

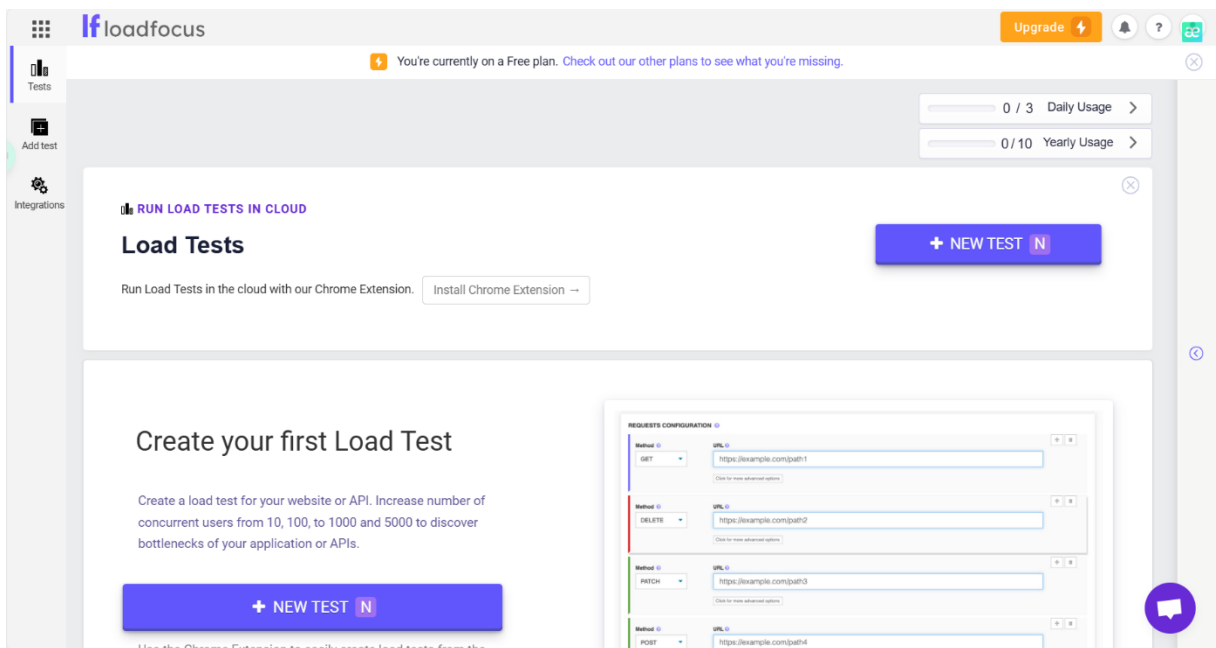
Bước 1: Đăng nhập vào trang web, sau khi đăng nhập xong chúng ta sẽ có giao diện sau.



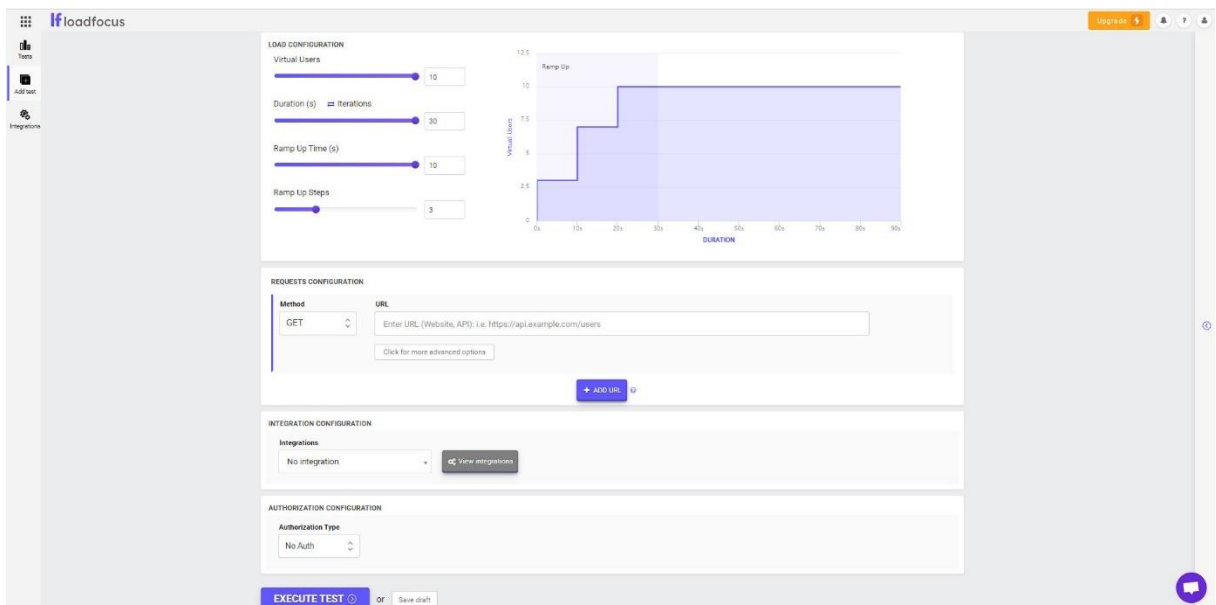
Bước 2: Chọn mục Load Testing như trong hình.



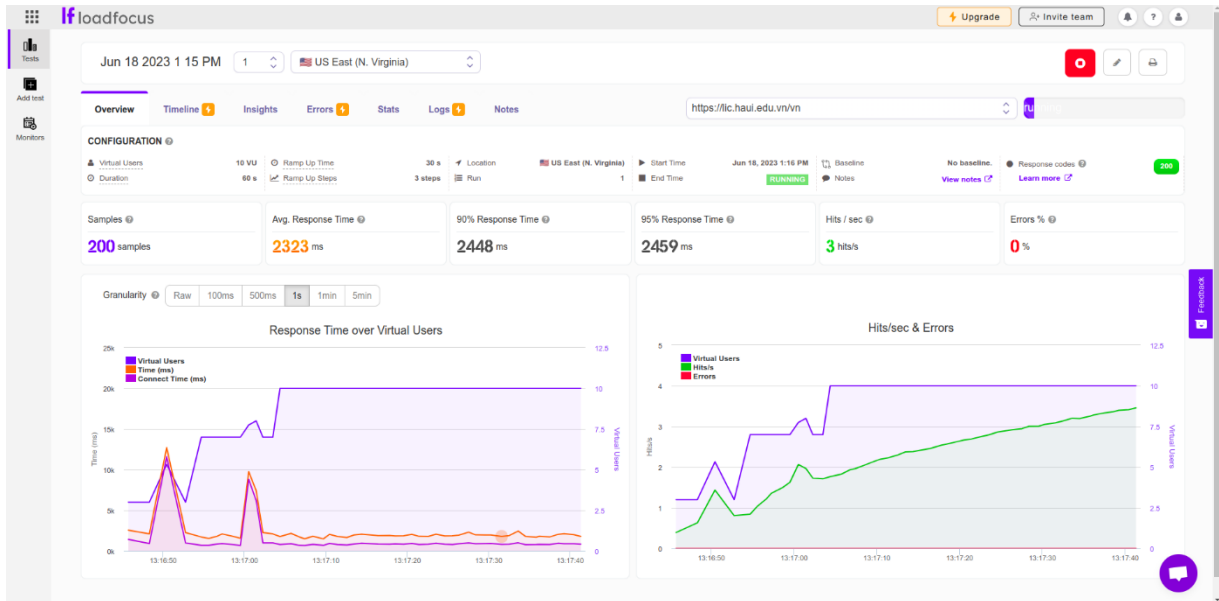
Bước 3: Sau khi chọn xong, trang web sẽ chuyển hướng đến trang web Load Testing



Bước 4: Chọn << New Test >>, chọn các thông số như số lượng user ảo, thời gian chạy, mức độ delay giữa các users,... Sau đó gán link trang web muốn kiểm thử vào vào URL và chọn EXECUTE TEST.



Bước 5: Đợi một khoảng thời gian để trang web bắt đầu chạy kiểm thử trang web đã gắn vào URL. Cuối cùng ta thu được một bảng báo cáo các thông số và kết quả của ca kiểm thử.



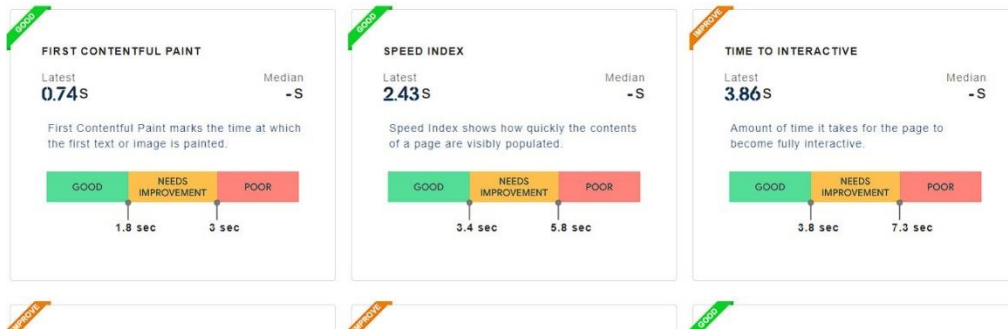
2.4.4. Kiểm tra tốc độ trang Speed Monitor Testing (Phan Anh Thắng)

2.4.4.1. Phân tích thiết kế kiểm thử

- Nội dung ca kiểm thử: Kiểm tra tốc độ hiển thị nội dung trang web trên những thiết bị khác nhau (Mobile và Desktop), xem tốc độ phản hồi và hiệu suất của trang web.
- Dưới đây là màn hình tổng quan sau khi chạy xong ca kiểm thử:
- Giao diện chi tiết khi chạy xong ca kiểm thử:



Performance Timing



- Các thông số trong giao diện chi tiết:
 - + Performance: Hiệu suất trang web (tối đa là 100)
 - + Accessibility: Khả năng tiếp cận
 - + Best practices: Thực hành tốt nhất
 - + SEO: Tối ưu hóa trên công cụ tìm kiếm
 - + Progressive Web App: Các tính năng khác của web
 - + Performance Timing: Thời gian thực hiện
 - + First Contentful Paint: Thời gian tải xong nội dung đầu tiên lên trang web
 - + Speed Index: Tốc độ hiển thị nội dung của trang
 - + Time To Interactive: Thời gian cần thiết để trang có thể tương tác được
 - + Largest Contentful Paint: Thời gian tải xong những nội dung có dung lượng lớn lên trang web
 - + Total Blocking Time: Tổng thời gian chặn
 - + Cumulative Layout Shift: Thay đổi của bố cục khung nhìn

- + Last Run Timeline: Dòng thời gian chạy lần cuối của trang (các giá trị có thể thay đổi)
- Giao diện thể hiện kết quả tổng quan của ca kiểm thử tốc độ trang web:



Các mức hiệu suất:

- 0 – 49: Hiệu suất thấp
 - 50 – 89: Hiệu suất trung bình
 - 90 – 100: Hiệu suất cao
- Đây là giao diện thể hiện các thông số và phương án giả thiết về hiệu suất: Từ những thông số trên có thể thấy rằng hiệu năng của trang web đang ở mức thấp khi chỉ đạt 40/100 điểm. Về mức độ SEO của trang web thì ở mức cao 92/100đ



Performance

Metrics



<div><div>●</div><div>First Contentful Paint</div><div>0.7 s</div></div> <div>First Contentful Paint marks the time at which the first text or image is painted. Learn more about the First Contentful Paint metric.</div>	<div><div>▲</div><div>Cumulative Layout Shift</div><div>0.462</div></div> <div>Cumulative Layout Shift measures the movement of visible elements within the viewport. Learn more about the Cumulative Layout Shift metric.</div>
<div><div>▲</div><div>Largest Contentful Paint</div><div>3.4 s</div></div> <div>Largest Contentful Paint marks the time at which the largest text or image is painted. Learn more about the Largest Contentful Paint metric</div>	<div><div>▲</div><div>Speed Index</div><div>2.4 s</div></div> <div>Speed Index shows how quickly the contents of a page are visibly populated. Learn more about the Speed Index metric.</div>
<div><div>▲</div><div>Total Blocking Time</div><div>360 ms</div></div> <div>Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. Learn more about the Total Blocking Time metric.</div>	

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



Ở đây có thể thấy được điểm hiệu suất là kém. Khi mất tới 3.4s để có thể tải được 1 nội dung lớn. Và mất tới 7.79s để load xong 1 bức ảnh. Chỉ số tốc độ của trang web cũng ở mức nhanh khi mất khoảng 2.4s.



Best Practices

Trust and Safety

Ensure CSP is effective against XSS attacks

User Experience

▲ Displays images with incorrect aspect ratio

General

Detected JavaScript libraries

▲ Browser errors were logged to the console

Passed audits (11)

Not applicable (1)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

Content Best Practices — Format your HTML in a way that enables crawlers to better understand your app's content.

Về điểm SEO của trang web cho kết quả ở mức tốt. Điều này cho thấy trang web có thể dễ dàng được tìm thấy khi người dùng tìm trên các công cụ tìm kiếm.

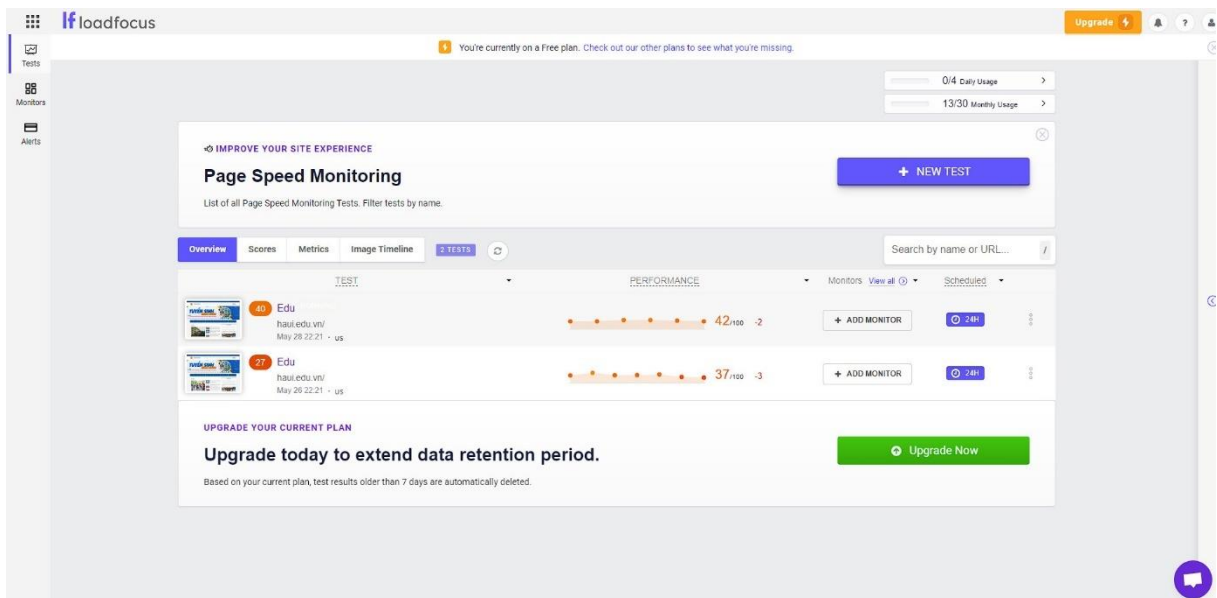
2.4.4.2. Thực hiện kiểm thử

Bước 1: Đầu tiên chúng ta cần tạo tài khoản hoặc đăng nhập trên công cụ kiểm thử Loadfocus trên trang web loadfocus.com

- Sau khi đăng nhập vào trang web sẽ xuất hiện giao diện sau:



Bước 2: Chọn mục Page Speed Monitoring trong hình, xuất hiện cửa sổ mới:



Bước 3: Nhấn vào << New Test >> để tạo một ca kiểm thử mới.

Bước 4: Nhập các thông số mới để tạo ca test mới:

New Test

×

URL

Switch to multiple URLs test

Name will autocomplete from the URL. Edit manually to set new name. We'll use this name in alerts.

https://www.hau.edu.vn/vn

✓

Name

HaUI

Location

us US East (N. Virginia) ×

Device

Desktop ×

Schedule

None

Hourly

Every 6h

Every 12h

Every 24h

Every 48h

Every week

RUN TEST NOW

Save for later

+ Name: Tên ca kiểm thử

+ Location: Vị trí thiết bị tạo mới

+ Web Page URL: Link trang web cần kiểm thử

+ Device: Thiết bị (Mobile/Desktop)

+ Schedule: Khung giờ chỉ định

+ Sau đó nhấn vào << Run Test Now >> để chạy ca kiểm thử

2.5. Báo cáo kiểm thử

Mục tiêu: Kiểm tra tốc độ load và kiểm tra tải của trang web

Phạm vi: Trang web hau.edu.vn và các trang web thuộc trang hau.edu.vn

Phương pháp kiểm thử: Kiểm thử hiệu năng

Kết quả:

Người thực hiện	Test case	Chức năng	Hành động	Kết quả mong muốn	Kết quả thực tế
Hồ Nam Tú	TC01	Kiểm tra tải	+ Tạo lệnh Jmeter + Tải tệp lệnh lên LoadFocus + Kiểm tra tải	Đáp ứng được tải cao	Success
Nguyễn Viết Vượng	TC02	Kiểm tra tải	+ Kiểm tra tải	Đáp ứng được tải cao	Success
Phan Anh Thắng	TC03	Kiểm tra tốc độ trang	+ Kiểm tra tốc độ tải trang	Thời gian tải nhanh	Unsuccessful



Kiểm tra tốc độ trang web



Performance

Metrics



<div>●</div> <div>First Contentful Paint</div> <div>First Contentful Paint marks the time at which the first text or image is painted. Learn more about the First Contentful Paint metric.</div> <div>0.7 s</div>	<div>▲</div> <div>Cumulative Layout Shift</div> <div>Cumulative Layout Shift measures the movement of visible elements within the viewport. Learn more about the Cumulative Layout Shift metric.</div> <div>0.462</div>
<div>▲</div> <div>Largest Contentful Paint</div> <div>Largest Contentful Paint marks the time at which the largest text or image is painted. Learn more about the Largest Contentful Paint metric.</div> <div>3.4 s</div>	<div>▲</div> <div>Speed Index</div> <div>Speed Index shows how quickly the contents of a page are visibly populated. Learn more about the Speed Index metric.</div> <div>2.4 s</div>
<div>▲</div> <div>Total Blocking Time</div> <div>Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. Learn more about the Total Blocking Time metric.</div> <div>360 ms</div>	

Values are estimated and may vary. The [performance score](#) is calculated directly from these metrics. [See calculator.](#)



- Hiệu năng khi load của trang web là ở mức trung bình. khi chỉ số hiệu năng chỉ đạt 40/100. Thời gian load xong 1 ảnh kích cỡ phù hợp lên màn hình nhanh mất 0.7s. Và thời gian load 1 nội dung lớn mất 3.4s. Đây là 1 tốc độ trung bình và nên nâng cấp để hiệu năng trang web được nâng cao hơn.

Overview

Timeline

Insights

Errors

Stats

Logs

Notes

completed

STATS REPORT

All labels requests executed during the test run.

Label	Samples	Response Time (ms)	Hits/sec	Min Time (ms)	Max Time (ms)	Perc 50%	Perc 90%	Perc 95%	Perc 99%	Total Errors	Errors %	Sent KBytes (KB/sec)	Received KBytes (KB/sec)
https://www.hauui.edu.vn/vn	229	1623	6	1405	2003	1653	1696	1724	1761	0	0	1	579

Kiểm tra tải

- Tốc độ tải trang chậm nhất là mất 2s và nhanh nhất là 1,4s, với lỗi là 0. Thời gian tải trung bình của trang là 1,62s

STATS REPORT														completed	
All labels requests executed during the test run.															
Label	Samples	Response Time (ms)	Hits/sec	Min Time (ms)	Max Time (ms)	Perc 50%	Perc 90%	Perc 95%	Perc 99%	Total Errors	Errors %	Sent KBytes (KB/sec)	Received KBytes (KB/sec)		
Homepage	130	2864	4	2618	3485	2880	2986	3014	3457	0	0	1	320		

Kiểm tra tải bằng Jmeter

- Đối với việc kiểm tra tải bằng Jmeter. Kết quả khi tải trang gần như tương đương khi thực hiện kiểm tra tải mặc định trên LoadFocus. Với thời gian đáp ứng trung bình là 2.8s
- ⇒ Đối với việc tải trang. Trang web hoàn toàn có thể đáp ứng được nhu cầu của nhiều người truy cập vào trang web cùng 1 lúc. Mọi người có thể dễ dàng tìm thấy được trang web với điểm SEO 92/100. Tuy nhiên tốc độ load trang vẫn còn tương đối chậm. Đề xuất chủ trang web nên nâng cấp trang web để giúp cho trải nghiệm của người dùng được tốt hơn.

PHẦN 3: KIẾN THỨC VÀ BÀI HỌC KINH NGHIỆM

Kiến thức

- Kiến thức về quy trình kiểm thử phần mềm: Bạn sẽ hiểu rõ về các bước và giai đoạn trong quy trình kiểm thử phần mềm, từ lập kế hoạch, xác định yêu cầu kiểm thử, thiết kế và thực hiện các ca kiểm thử, đánh giá kết quả và báo cáo
- Kỹ năng phân tích và ghi nhận kết quả kiểm thử: Bạn sẽ học cách phân tích kết quả kiểm thử, xác định các lỗi và vấn đề, ghi nhận thông tin chi tiết về các ca kiểm thử và xây dựng báo cáo kết quả kiểm thử.
- Hiểu biết về các lỗi phổ biến: Bạn sẽ nhận thức về các lỗi phổ biến trong phần mềm và có khả năng xác định, mô tả và tái hiện các lỗi này trong quá trình kiểm thử.
- Kỹ năng tương tác và làm việc trong nhóm: Khi làm việc trên dự án kiểm thử phần mềm, bạn sẽ có cơ hội phát triển kỹ năng tương tác, làm việc nhóm, giao tiếp và cộng tác với các thành viên khác trong nhóm kiểm thử và nhóm phát triển.
- Kiến thức về quản lý kiểm thử: Bạn có thể hiểu cách quản lý kiểm thử phần mềm, từ lập kế hoạch, phân bổ tài nguyên, quản lý thời gian và đảm bảo chất lượng kiểm thử.

Chuẩn đầu ra

- Cơ bản hoàn thành được các chuẩn đầu ra của học phần, L1.1, L2.1, L2.2, L4.1

Bài học kinh nghiệm

- Về kiến thức: Chủ động tìm tài liệu, hướng dẫn. Biết cách vận dụng những kiến thức đã học được để xử lý vấn đề tốt hơn.
- Về kỹ năng: Phân chia thời gian làm việc cho từng đề mục kỹ hơn, hoàn thành thời gian theo đúng tiến độ, phân chia công việc cho các thành viên trong nhóm 1 cách hợp lý

Đề xuất

- Công cụ LoadFocus thật sự giúp cho kiểm tra tốc độ trang, kiểm tra tốc độ tải trở nên nhanh chóng và giảm thời gian hơn. Nó có công cụ thiết lập thời gian kiểm tra theo hàng giờ, hàng ngày 1 cách tự động giúp người kiểm thử giảm bớt gánh nặng trong việc kiểm thử. Nhóm

em nhận thấy có thể sử dụng công cụ này để ứng dụng trong kiểm thử cho các phần mềm và ứng dụng trong tương lai vì những tính năng ưu việt của nó.

TÀI LIỆU THAM KHẢO

- [1] Thạc Bình Cường và Nguyễn Đức Mẫn, *Giáo trình: Kiểm thử và đảm bảo chất lượng phần mềm* – Nhà xuất bản Bách khoa Hà Nội, xuất bản 2011
- [2] Jeffrey Rubin, *Handbook of usability testing: How to plan, design, and conduct effective tests*, John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [3] Loadfocus Documentation