

TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ



**BÁO CÁO ĐỒ ÁN HỌC PHẦN
KIỂM THỬ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

**ỨNG DỤNG KIỂM THỬ HIỆU NĂNG WEBSITE QUẢN LÝ
NHÀ HÀNG BẰNG JMETER**

NHÓM: 04

GVHD: TS. Cao Thanh Sơn

SV thực hiện: Võ Văn Hòa, 215748010310021

Trần Văn Hoàng, 215748010310012

Nghệ An, 06/2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn đến **TS. Cao Thanh Sơn** đã dành thời gian và công sức để hướng dẫn, giảng dạy và truyền đạt kiến thức cho chúng em trong suốt quá trình học tập. Sự am hiểu, sự tận tâm và sự chia sẻ của thầy đã giúp chúng em tiếp thu và hiểu bài học một cách dễ dàng hơn.

Chúng em rất biết ơn thầy vì cách thức giảng dạy tận tâm và sự chia sẻ kiến thức thực tế của mình. Thầy đã tạo ra một môi trường học tập tích cực, nơi chúng em có thể thảo luận, trao đổi ý kiến và phát triển khả năng của mình. Thầy cung cấp cho chúng tôi những bài giảng thú vị và đa dạng, giúp chúng em hiểu rõ và ứng dụng kiến thức vào thực tế.

Chúng em cũng nhận thức rằng nhóm còn nhiều hạn chế và khó khăn trong việc tiếp thu và áp dụng kiến thức. Tuy nhiên, sự hướng dẫn và những lời khuyên của thầy đã giúp chúng em vượt qua những thách thức đó. Chúng em cam kết sẽ tiếp tục nỗ lực hơn nữa để phát triển và áp dụng kiến thức một cách hiệu quả.

Cuối cùng, chúng em xin chân thành cảm ơn thầy vì sự tận tâm và mong muốn giúp đỡ chúng em trong quá trình học tập. Chúng em hứa sẽ tiếp thu và áp dụng những kiến thức đã học để đóng góp vào sự phát triển của xã hội và đạt được thành công trong sự nghiệp sau này.

Chúng em xin chân thành cảm ơn thầy!

Nghệ An, ngày 13 tháng 06 năm 2025

Nhóm Sinh Viên

Võ Văn Hòa

Trần Văn Hoàng

MỤC LỤC

LỜI CẢM ƠN.....	1
DANH SÁCH CÁC HÌNH.....	3
1. Lý do lựa chọn đề tài.....	4
2. Mục tiêu nghiên cứu.....	5
3. Phương pháp thực hiện.....	6
4. Cấu trúc đồ án.....	7
CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM.....	8
1.1. Khái niệm kiểm thử phần mềm.....	8
1.2. Các kỹ thuật cơ bản của kiểm thử.....	9
1.3. Quy trình kiểm thử phần mềm.....	11
1.4. Các loại kiểm thử hiệu năng.....	12
1.5. Công cụ kiểm thử hiệu năng.....	14
1.6. Một số nghiên cứu liên quan đến đề tài.....	17
CHƯƠNG 2. CÔNG CỤ KIỂM THỬ HIỆU NĂNG JMETER.....	18
2.1. Giới thiệu về Jmeter.....	18
2.2. Ưu và nhược điểm của Jmeter.....	19
2.3. Cài đặt và thiết lập Jmeter.....	21
2.4. Các thành phần chính của Jmeter.....	22
2.5. Ví dụ kiểm thử hiệu năng với Jmeter.....	26
CHƯƠNG 3. KIỂM THỬ HIỆU NĂNG HỆ THỐNG QUẢN LÝ NHÀ HÀNG SỬ DỤNG JMETER.....	27
3.1. Kiểm thử website nhà hàng ResOnline.....	27
TÀI LIỆU THAM KHẢO.....	57

DANH SÁCH CÁC HÌNH

Fig 1.1. Software testing.....	3
Fig 1.2. Software testing.....	3
Fig 2.1. Software testing.....	5
Fig 3.1. Software testing.....	6

MỞ ĐẦU

1. Lý do lựa chọn đề tài

Sự gia tăng về nhu cầu ăn uống, đặt món trực tuyến và quản lý nhà hàng một cách chuyên nghiệp đã đặt ra yêu cầu cao hơn cho các hệ thống phần mềm phục vụ lĩnh vực này. Việc xây dựng website quản lý nhà hàng không còn là điều mới mẻ, tuy nhiên, để hệ thống hoạt động ổn định, đáp ứng tốt lượng người dùng truy cập lớn trong giờ cao điểm, thì kiểm thử hiệu năng là một yếu tố then chốt không thể bỏ qua.

Kiểm thử hiệu năng là bước giúp đánh giá khả năng đáp ứng của hệ thống trong các điều kiện tải khác nhau như số lượng người dùng tăng cao, thời gian xử lý các yêu cầu, mức độ ổn định của hệ thống khi hoạt động liên tục. Trong đó, Apache JMeter là một công cụ kiểm thử mã nguồn mở mạnh mẽ, phổ biến và được tin dùng trong nhiều dự án thực tế.

Trong quá trình tìm hiểu và thực hiện xây dựng website quản lý nhà hàng bằng Laravel framework, chúng em nhận thấy rằng hiệu năng hệ thống là yếu tố có ảnh hưởng lớn đến trải nghiệm của người dùng, đặc biệt trong những khung giờ cao điểm như trưa và tối. Một website chậm, thường xuyên lỗi tải trang hay xử lý chậm sẽ gây khó chịu và ảnh hưởng đến hoạt động kinh doanh. Chính vì vậy, chúng em quyết định lựa chọn đề tài **“Ứng dụng kiểm thử hiệu năng website quản lý nhà hàng bằng JMeter”** với mong muốn kiểm chứng chất lượng hoạt động của hệ thống, đánh giá khả năng chịu tải và phát hiện ra những điểm còn hạn chế để tối ưu trước khi đưa vào sử dụng thực tế.

Việc lựa chọn đề tài này không chỉ giúp chúng em rèn luyện kỹ năng sử dụng công cụ kiểm thử chuyên dụng mà còn nâng cao nhận thức về tầm quan trọng của kiểm thử phần mềm trong quy trình phát triển sản phẩm, mong muốn góp phần hoàn thiện hơn hệ thống quản lý nhà hàng cũng như tích lũy thêm kinh nghiệm thực tế cho bản thân trong lĩnh vực kiểm thử và đảm bảo chất lượng phần mềm.

2. Mục tiêu nghiên cứu

Đề tài được thực hiện nhằm hướng đến việc nâng cao chất lượng và độ ổn định của hệ thống quản lý nhà hàng hoạt động trên nền tảng web. Một hệ thống không chỉ cần giao diện thân thiện, chức năng đầy đủ mà còn phải có khả năng đáp ứng tốt với lưu lượng truy cập cao, đảm bảo hiệu quả vận hành trong môi trường thực tế. Do đó, mục tiêu đầu tiên của đề tài là đánh giá toàn diện hiệu năng của hệ thống, bao gồm các yếu tố như tốc độ phản hồi, khả năng xử lý đồng thời nhiều người dùng, mức độ ổn định của hệ thống khi hoạt động trong thời gian dài.

Bên cạnh đó, đề tài cũng nhằm mục tiêu tìm hiểu và ứng dụng công cụ Apache JMeter, một công cụ kiểm thử mã nguồn mở phổ biến để tiến hành kiểm thử hiệu năng một cách chuyên nghiệp. Thông qua việc xây dựng các kịch bản mô phỏng hành vi người dùng thực tế như đăng nhập, đặt bàn, gọi món, xem thực đơn,... hệ thống sẽ được kiểm tra trong các tình huống khác nhau để đánh giá mức độ chịu tải và phát hiện ra các điểm nghẽn về hiệu năng.

Đề tài còn góp phần nâng cao tư duy phân tích và giải quyết vấn đề trong quá trình phát triển phần mềm thực tế. Việc tiếp cận bài bản từ khâu xây dựng kịch bản kiểm thử đến đánh giá kết quả giúp người thực hiện hình thành quy trình làm việc khoa học, chuyên nghiệp.

Ngoài ra kết quả đạt được cũng hướng đến việc phân tích kết quả kiểm thử để đưa ra nhận định khách quan về hiệu suất của hệ thống. Từ những dữ liệu thu được, người phát triển có thể điều chỉnh mã nguồn, tối ưu cơ sở dữ liệu, hoặc cải tiến kiến trúc hệ thống nhằm nâng cao hiệu quả vận hành. Đây là bước quan trọng nhằm hoàn thiện sản phẩm trước khi đưa vào triển khai thực tế.

Thông qua việc thực hiện đề tài này, người thực hiện không chỉ củng cố kiến thức về kiểm thử phần mềm, mà còn nâng cao kỹ năng sử dụng các công cụ kiểm thử chuyên dụng, đồng thời nhận thức rõ hơn vai trò của kiểm thử hiệu năng trong quy trình phát triển phần mềm.

3. Phương pháp thực hiện

Để thực hiện hiệu quả đề tài trước hết cần tiến hành khảo sát và tìm hiểu thực tế nhu cầu quản lý tại các nhà hàng hiện nay. Việc này giúp xác định rõ những chức năng cốt lõi mà một hệ thống quản lý nhà hàng cần có, từ đó làm cơ sở xây dựng các kịch bản kiểm thử phù hợp. Một số nội dung cần khảo sát bao gồm: quy trình đặt bàn, gọi món, thanh toán, quản lý thực đơn, quản lý khách hàng, thống kê doanh thu.

Song song với việc khảo sát thực tế, đề tài tiến hành tham khảo và phân tích một số trang mạng quản lý nhà hàng phổ biến hiện nay như PasGo, Sushi World, Vị Quê để rút ra bài học kinh nghiệm. Qua đó, những ưu điểm như giao diện đơn giản, thao tác nhanh, quản lý dễ dàng sẽ được tiếp thu để định hướng xây dựng hệ thống tối ưu. Đồng thời, những hạn chế như khả năng mở rộng thấp, tốc độ xử lý chậm khi có nhiều người truy cập, hoặc chi phí duy trì cao sẽ được lưu ý để từ đó đề xuất giải pháp phù hợp hơn.

Sau giai đoạn khảo sát và phân tích, hệ thống website quản lý nhà hàng sẽ được xây dựng trên nền tảng Laravel. Từ đây, đề tài sẽ sử dụng công cụ Apache JMeter để tiến hành kiểm thử hiệu năng cho website. Việc kiểm thử được thực hiện thông qua các kịch bản mô phỏng hành vi người dùng thực tế như truy cập website, đăng nhập, đặt món, thanh toán,... Các kịch bản này sẽ được cấu hình theo các mức tải khác nhau để đánh giá hiệu suất xử lý và khả năng chịu tải của hệ thống.

Cuối cùng thì kết quả kiểm thử sẽ được phân tích dựa trên các chỉ số như: thời gian phản hồi, tỉ lệ lỗi, số lượng yêu cầu xử lý mỗi giây, và mức sử dụng tài nguyên hệ thống. Từ đó, đề tài sẽ đưa ra các đánh giá, nhận xét khách quan về khả năng hoạt động của website và đề xuất các phương án tối ưu nếu cần thiết. Kết quả phân tích này sẽ là cơ sở quan trọng để cải thiện hiệu suất hệ thống và đảm bảo khả năng mở rộng trong tương lai.

4. Cấu trúc đề án

Chương 1: Tổng quan về kiểm thử phần mềm

Chương này trình bày tổng quan kiến thức cơ bản về kiểm thử phần mềm, bao gồm khái niệm, vai trò và tầm quan trọng của kiểm thử trong quy trình phát triển phần mềm. Đồng thời, chương cũng phân loại các loại kiểm thử phần mềm phổ biến như kiểm thử chức năng, kiểm thử phi chức năng, kiểm thử hiệu năng, kiểm thử bảo mật,... và tập trung làm rõ khái niệm, mục tiêu cũng như các chỉ số quan trọng trong kiểm thử hiệu năng. Qua đó, chương này sẽ giúp người đọc hiểu rõ nền tảng lý thuyết để làm cơ sở cho các nội dung sẽ được thực hiện trong các chương tiếp theo.

Chương 2: Công cụ kiểm thử hiệu năng Jmeter

Chương này giới thiệu về công cụ Apache JMeter, là một trong những công cụ kiểm thử hiệu năng mã nguồn mở phổ biến nhất hiện nay. Nội dung chương bao gồm tổng quan về JMeter, các tính năng nổi bật, cách cài đặt và cấu hình công cụ, các thành phần cơ bản như Thread Group, Sampler, Listener, Assertion, Timer,... Bên cạnh đó, chương này cũng trình bày quy trình xây dựng kịch bản kiểm thử hiệu năng và hướng dẫn sử dụng JMeter để mô phỏng hành vi người dùng truy cập hệ thống. Qua đó, chương này đóng vai trò cầu nối giữa lý thuyết kiểm thử và ứng dụng thực tế vào hệ thống quản lý nhà hàng.

Chương 3: Kiểm thử hiệu năng hệ thống quản lý nhà hàng sử dụng JMeter

Chương này đi sâu vào việc kiểm thử hiệu năng thực tế trên hệ thống website quản lý nhà hàng được xây dựng bằng Laravel. Đầu tiên, chương trình bày sơ lược về hệ thống và các chức năng chính cần kiểm thử. Tiếp theo, các kịch bản kiểm thử hiệu năng sẽ được xây dựng và triển khai thông qua JMeter, với các hành vi phổ biến như đăng nhập, đặt bàn, gọi món, và xử lý đơn hàng. Sau khi thực hiện kiểm thử, chương sẽ phân tích kết quả dựa trên các chỉ số như thời gian phản hồi, tỉ lệ lỗi, thông lượng và mức sử dụng tài nguyên.

CHƯƠNG 1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

Kiểm thử phần mềm đóng vai trò quan trọng trong quy trình phát triển phần mềm, góp phần đảm bảo chất lượng sản phẩm cuối cùng trước khi được đưa vào sử dụng. Việc thực hiện kiểm thử không chỉ giúp phát hiện lỗi và vấn đề tiềm ẩn trong hệ thống mà còn nâng cao độ tin cậy và hiệu suất của phần mềm.

1.1. Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm là hoạt động nhằm tìm kiếm và phát hiện ra các lỗi của phần mềm, đảm bảo phần mềm chính xác, đúng và đầy đủ theo yêu cầu của khách hàng, yêu cầu của sản phẩm đã đặt ra. Kiểm thử phần mềm cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm điều này cho phép đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

Quá trình kiểm thử này giúp phát hiện lỗi và sai sót trong quá trình phát triển, từ đó khắc phục kịp thời để tránh rủi ro trong thực tế. Ngoài ra, kiểm thử còn giúp đảm bảo rằng phần mềm đáp ứng các yêu cầu chức năng và phi chức năng đã đề ra. Bằng cách kiểm tra toàn diện, kiểm thử giúp cải thiện độ ổn định, tăng tính bảo mật và nâng cao trải nghiệm người dùng.

Có nhiều phương pháp kiểm thử phần mềm khác nhau, trong đó phổ biến nhất là kiểm thử hộp đen, kiểm thử hộp trắng và kiểm thử hộp xám. Kiểm thử hộp đen tập trung vào việc kiểm tra các đầu vào và đầu ra của hệ thống mà không quan tâm đến cấu trúc bên trong. Trong khi đó, kiểm thử hộp trắng đi sâu vào mã nguồn, thuật toán để kiểm tra logic hoạt động của chương trình. Kiểm thử hộp xám là sự kết hợp giữa hai phương pháp trên, cho phép người kiểm thử có một phần thông tin về hệ thống nhưng không cần nắm rõ toàn bộ mã nguồn.

Chú trọng vào kiểm thử và đảm bảo chất lượng phần mềm giúp giảm thiểu rủi ro, tiết kiệm chi phí sửa lỗi trong giai đoạn sau và nâng cao sự hài lòng của khách hàng. Bên cạnh đó, kiểm thử còn giúp đảm bảo hiệu suất và khả năng mở rộng của phần mềm, đặc biệt trong các hệ thống lớn có nhiều người dùng.

1.2. Các kỹ thuật cơ bản của kiểm thử

Kiểm thử chức năng là quá trình đánh giá phần mềm nhằm xác định xem nó có hoạt động đúng theo các yêu cầu chức năng đã được xác định hay không. Quá trình này tập trung vào việc kiểm tra từng chức năng cụ thể của phần mềm để đảm bảo rằng chúng đáp ứng đầy đủ các tiêu chí đã đặt ra. Trong kiểm thử chức năng, người kiểm thử sẽ tiến hành kiểm tra đầu vào, xử lý dữ liệu và đầu ra để xác minh rằng hệ thống hoạt động theo mong đợi. Các bước chính của kiểm thử chức năng bao gồm xác định yêu cầu, lập kế hoạch kiểm thử, thực hiện kiểm thử và ghi nhận kết quả. Việc kiểm thử chức năng đóng vai trò quan trọng trong việc đảm bảo phần mềm vận hành chính xác, giảm thiểu lỗi và nâng cao trải nghiệm người dùng.

Kiểm thử phi chức năng là quá trình đánh giá các yếu tố không liên quan trực tiếp đến chức năng của phần mềm, như hiệu suất, bảo mật, khả năng mở rộng, khả năng khôi phục, khả năng tương thích và tính dễ sử dụng. Mục tiêu của kiểm thử không chức năng là đảm bảo rằng phần mềm không chỉ hoạt động đúng theo yêu cầu chức năng mà còn phải đáp ứng được các tiêu chí về hiệu suất ổn định, an toàn và thân thiện với người dùng.

Một số loại kiểm thử không chức năng phổ biến bao gồm kiểm thử hiệu năng, kiểm thử bảo mật, kiểm thử khả năng sử dụng và kiểm thử khả năng khôi phục. Kiểm thử hiệu năng đánh giá khả năng chịu tải của hệ thống trong các điều kiện cụ thể để đảm bảo hoạt động mượt mà. Kiểm thử bảo mật tập trung vào việc phát hiện và giảm thiểu các lỗ hổng có thể bị khai thác, đảm bảo hệ thống không bị tấn công. Kiểm thử khả năng sử dụng giúp đánh giá mức độ dễ sử dụng của phần mềm từ góc độ người dùng cuối, cải thiện trải nghiệm tổng thể. Cuối cùng, kiểm thử khả năng khôi phục đảm bảo hệ thống có thể phục hồi nhanh chóng sau sự cố mà không gây mất dữ liệu quan trọng.

Kiểm thử đơn vị là quá trình kiểm tra từng phần nhỏ nhất của mã nguồn, chẳng hạn như hàm, phương thức hoặc lớp, nhằm đảm bảo rằng mỗi thành phần hoạt động đúng theo yêu cầu. Mục tiêu chính của kiểm thử đơn vị là phát hiện lỗi

sớm trong giai đoạn phát triển phần mềm, giúp tăng tính ổn định và tin cậy của mã nguồn. Ngoài ra, kiểm thử đơn vị còn giúp lập trình viên dễ dàng bảo trì và cải thiện mã nguồn trong tương lai, đồng thời giảm thiểu chi phí sửa lỗi trong các giai đoạn sau.

Kiểm thử tích hợp là quá trình kiểm tra sự tương tác giữa các thành phần phần mềm sau khi chúng được kết hợp để tạo thành một hệ thống hoàn chỉnh. Mục tiêu của kiểm thử tích hợp là đảm bảo rằng các thành phần có thể làm việc cùng nhau một cách đúng đắn, liên kết chặt chẽ và không xảy ra lỗi khi trao đổi dữ liệu. Nhờ đó, kiểm thử tích hợp giúp đảm bảo hệ thống hoạt động chính xác, ổn định và đáp ứng đúng yêu cầu của người dùng.

Kiểm thử hệ thống là quá trình kiểm tra toàn bộ hệ thống phần mềm nhằm đảm bảo rằng nó hoạt động đúng theo yêu cầu đã được xác định và đáp ứng đầy đủ các yêu cầu chức năng cũng như phi chức năng của người dùng. Trong kiểm thử hệ thống, có nhiều loại kiểm thử khác nhau nhằm đánh giá các khía cạnh quan trọng của phần mềm. Kiểm thử chấp nhận giúp xác định xem hệ thống có đáp ứng được yêu cầu của người dùng hay không. Kiểm thử hệ thống hoàn chỉnh kiểm tra toàn bộ hệ thống từ đầu đến cuối để đảm bảo tất cả các thành phần có thể hoạt động chính xác khi kết hợp với nhau. Kiểm thử hiệu suất đánh giá khả năng của hệ thống trong điều kiện áp lực, đảm bảo nó có thể hoạt động ổn định với yêu cầu về tốc độ và khả năng chịu tải.

Kiểm thử hồi quy là một quá trình kiểm tra lại các phần của phần mềm đã được kiểm thử trước đó nhằm đảm bảo rằng các thay đổi mới hoặc sửa đổi không gây ảnh hưởng tiêu cực đến các tính năng hiện có của hệ thống. Việc thực hiện kiểm thử hồi quy là cần thiết sau mỗi lần có sự thay đổi trong mã nguồn, cập nhật phần mềm hoặc bổ sung tính năng mới vào hệ thống. Điều này giúp đảm bảo rằng hệ thống vẫn hoạt động ổn định và không phát sinh lỗi ngoài ý muốn sau khi có sự điều chỉnh hoặc mở rộng.

1.3. Quy trình kiểm thử phần mềm

Là tập hợp các bước có hệ thống để đảm bảo phần mềm hoạt động đúng theo yêu cầu, không có lỗi và đáp ứng tốt các tiêu chí về hiệu suất, bảo mật và trải nghiệm người dùng.



Hình 1.1. Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm bắt đầu với việc xác định yêu cầu kiểm thử. Đây là bước quan trọng để đảm bảo việc kiểm tra được thực hiện đúng hướng, tập trung vào những tiêu chí cần thiết theo yêu cầu của bài toán và mong muốn của người dùng. Việc xác định chính xác định nghĩa yêu cầu và tiêu chuẩn kiểm thử giúp đảm bảo kết quả kiểm thử đáp ứng các mục tiêu đề ra, đồng thời giảm thiểu rủi ro có thể phát sinh.

Sau khi xác định yêu cầu, bước tiếp theo là kiểm soát và lập kế hoạch kiểm thử phần mềm. Việc lên kế hoạch giúp xác định nguồn lực, phương pháp kiểm thử phù hợp, lịch trình thực hiện cũng như các tiêu chí đánh giá kết quả. Một kế hoạch kiểm thử hiệu quả sẽ bao gồm việc xác định phạm vi kiểm thử, mục tiêu cần đạt, phương pháp tiếp cận, tiến độ thực hiện và các yếu tố liên quan khác.

Bước phân tích và thiết kế kịch bản kiểm thử giúp người kiểm thử xây dựng các ca kiểm thử và dữ liệu kiểm thử phù hợp. Mục tiêu là đảm bảo bao quát đầy đủ các chức năng, tính năng của phần mềm và phát hiện rủi ro tiềm ẩn. Quá trình này bao gồm phân tích tài liệu, xác định các tính năng cần kiểm thử, ưu tiên kiểm tra và lựa chọn môi trường, công cụ phù hợp.

Sau khi thiết kế ca kiểm thử và dữ liệu kiểm thử, người kiểm thử tiến hành

kiểm tra tính năng phần mềm. Giai đoạn này có thể bao gồm kiểm tra API hoặc kiểm thử giao diện nếu đã lập trình xong. Mục tiêu là tìm ra lỗi dựa trên các ca kiểm thử đã chuẩn bị trước đó. Hoạt động kiểm thử gồm so sánh kết quả thực tế với mong đợi, ghi nhận lỗi và báo cáo lên hệ thống quản lý lỗi.

Sau khi kiểm tra xong, nhóm kiểm thử ghi nhận lỗi, đánh giá kết quả kiểm thử và lập báo cáo chi tiết. Nếu phát hiện lỗi, cần báo lại để lập trình viên sửa kịp thời, tránh ảnh hưởng đến khách hàng. Việc đánh giá rủi ro và chất lượng phần mềm giúp xác định tiêu chí hoàn thành kiểm thử. Mỗi dự án có tiêu chí đánh giá khác nhau tùy theo đặc thù.

Cuối cùng là đánh giá kết quả kiểm thử để đảm bảo lỗi đã được khắc phục và phần mềm đáp ứng yêu cầu. Nếu cần, nhóm kiểm thử tiếp tục kiểm tra để đảm bảo tính ổn định, sau đó báo cáo với quản lý dự án để quyết định có cần sửa tiếp sau khi triển khai hay không.

1.4. Các loại kiểm thử hiệu năng

Kiểm thử hiệu năng là một quy trình được sử dụng để đánh giá hiệu suất và khả năng chịu tải của một ứng dụng hoặc hệ thống. Có nhiều loại kiểm thử hiệu năng khác nhau, dưới đây là một số loại kiểm thử phổ biến.



Hình 1.2. Kiểm thử hiệu năng

Kiểm thử tải giúp xác định hiệu suất của hệ thống khi hoạt động dưới tải bình thường và tải cao. Mục tiêu là kiểm tra tốc độ phản hồi, khả năng xử lý số lượng người dùng đồng thời và tài nguyên hệ thống. Nếu hệ thống chậm hoặc không đáp ứng được yêu cầu, cần tối ưu hóa. Đây là một trong những bước quan trọng để đảm bảo trải nghiệm người dùng tốt.

Kiểm thử căng thẳng đẩy hệ thống vượt quá giới hạn chịu tải để kiểm tra khả năng chịu đựng. Mục tiêu là xác định điểm sập hệ thống và đánh giá khả năng phục hồi sau khi bị quá tải. Nếu hệ thống bị crash, cần tối ưu để cải thiện độ ổn định. Điều này giúp tránh gián đoạn dịch vụ trong môi trường thực tế.

Kiểm thử đột biến đánh giá hệ thống khi có sự gia tăng đột ngột về số lượng người dùng hoặc dữ liệu. Hệ thống phải có khả năng xử lý và phản hồi nhanh mà không bị gián đoạn. Nếu xảy ra lỗi, cần tối ưu kiến trúc hoặc mở rộng tài nguyên. Loại này quan trọng với các ứng dụng có lưu lượng truy cập không ổn định.

Kiểm thử sức chịu đựng đánh giá hiệu suất hệ thống khi chạy liên tục trong thời gian dài. Điều này giúp phát hiện các vấn đề như rò rỉ bộ nhớ, suy giảm hiệu suất hoặc lỗi tích lũy theo thời gian. Nếu hệ thống xuống cấp dần, cần cải thiện bộ nhớ, xử lý dữ liệu hoặc tối ưu tài nguyên.

Kiểm thử khả năng mở rộng kiểm tra xem hệ thống có thể tăng tài nguyên để đáp ứng số lượng người dùng ngày càng lớn hay không. Hệ thống cần duy trì hiệu suất ổn định khi mở rộng server, CPU hoặc RAM. Nếu hiệu suất giảm khi mở rộng, cần điều chỉnh kiến trúc hoặc cân bằng tải. Điều này giúp hệ thống phát triển linh hoạt theo nhu cầu.

Kiểm thử tính đồng thời đánh giá hệ thống khi có nhiều người dùng thực hiện thao tác cùng lúc. Nó giúp phát hiện các vấn đề tranh chấp tài nguyên, khóa dữ liệu hoặc tắc nghẽn. Nếu hệ thống bị chậm hoặc treo, cần tối ưu xử lý song song và cơ chế đồng bộ dữ liệu. Kiểm thử này quan trọng với các ứng dụng có lượng truy cập lớn cùng thời điểm.

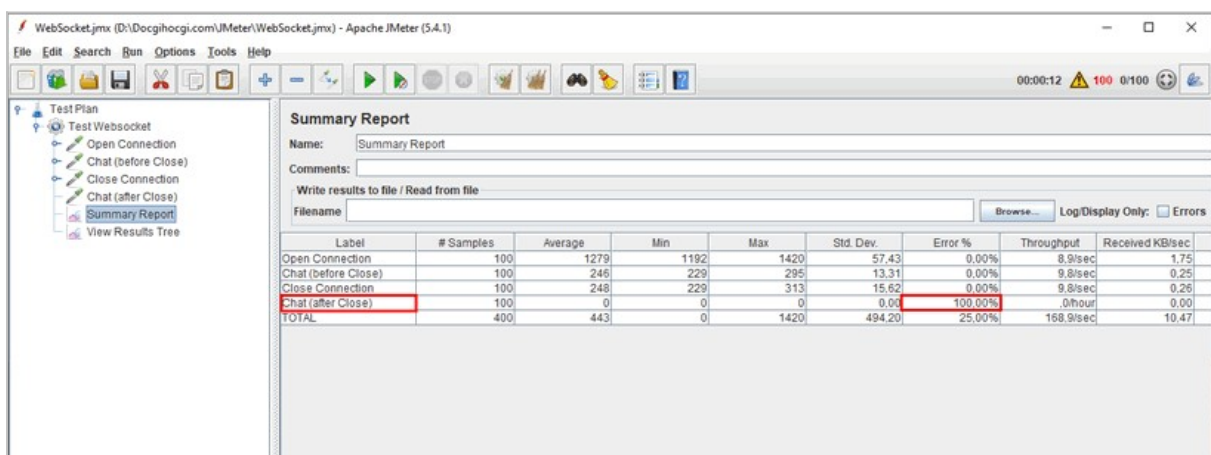
1.5. Công cụ kiểm thử hiệu năng

1.5.1. Công cụ kiểm thử Jmeter

Apache JMeter là một công cụ mã nguồn mở được sử dụng để kiểm thử hiệu năng của các ứng dụng web và nhiều loại dịch vụ khác. Ban đầu, JMeter được phát triển để kiểm thử ứng dụng web, nhưng hiện nay nó hỗ trợ nhiều giao thức khác nhau như HTTP, HTTPS, FTP, JDBC, SOAP, REST, và MQTT. Công cụ này giúp giả lập nhiều người dùng truy cập đồng thời vào hệ thống để đánh giá hiệu suất, khả năng chịu tải và độ ổn định của ứng dụng dưới áp lực cao.

JMeter cung cấp nhiều tính năng mạnh mẽ như tạo và thực thi kịch bản kiểm thử, phân tích dữ liệu phản hồi, đo lường thời gian phản hồi của hệ thống và báo cáo kết quả trực quan. Người dùng có thể thiết lập nhiều tham số kiểm thử như số lượng người dùng ảo, tần suất gửi yêu cầu, thời gian chờ giữa các yêu cầu và điều kiện dừng kiểm thử. Ngoài ra, JMeter hỗ trợ tích hợp với các công cụ giám sát như Grafana, Prometheus để theo dõi hiệu suất hệ thống theo thời gian thực.

Một trong những ưu điểm lớn của JMeter là tính miễn phí, giao diện thân thiện và khả năng mở rộng mạnh mẽ nhờ hệ thống plugin phong phú. Công cụ này có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux mà không cần cài đặt phức tạp. JMeter cũng cho phép xuất báo cáo chi tiết về hiệu suất hệ thống dưới dạng biểu đồ, giúp người kiểm thử dễ dàng phân tích kết quả.



WebSocket.jmx (D:\Docgihocgi.com\JMeter\WebSocket.jmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:00:12 100 0/100

Test Plan

- Test WebSocket
 - Open Connection
 - Chat (before Close)
 - Close Connection
 - Chat (after Close)
 - Summary Report
 - View Results Tree

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors

Label	# Samples	Average	Min	Max	Std. Dev	Error %	Throughput	Received Kbytes
Open Connection	100	1279	1192	1420	57.43	0.00%	8.8/sec	1.75
Chat (before Close)	100	246	229	295	13.31	0.00%	9.8/sec	0.25
Close Connection	100	248	229	313	15.62	0.00%	9.8/sec	0.26
Chat (after Close)	100	0	0	0	0.00	100.00%	0.0/sec	0.00
TOTAL	400	443	0	1420	494.20	25.00%	168.9/sec	10.47

Hình 1.3. Công cụ kiểm thử Jmeter

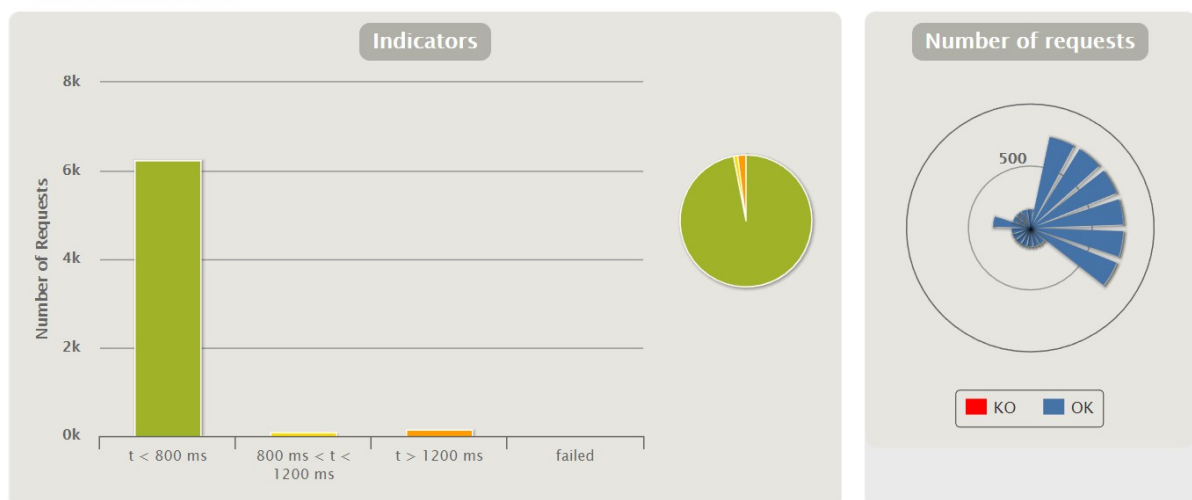
1.5.2. Công cụ kiểm thử Gatling

Gatling là một công cụ kiểm thử hiệu năng mã nguồn mở được viết bằng ngôn ngữ Scala, giúp đánh giá khả năng chịu tải của hệ thống. Công cụ này chuyên dùng để kiểm thử ứng dụng web và hỗ trợ các giao thức như HTTP, WebSockets, JMS và JDBC. Gatling giúp mô phỏng nhiều người dùng truy cập đồng thời vào hệ thống để kiểm tra hiệu suất và độ ổn định của ứng dụng.

Một trong những tính năng nổi bật của Gatling là khả năng tự động tạo báo cáo chi tiết dưới dạng đồ họa, giúp người kiểm thử dễ dàng phân tích kết quả kiểm thử. Công cụ này cũng hỗ trợ viết kịch bản kiểm thử bằng mã, giúp linh hoạt hơn trong việc tùy chỉnh các kịch bản kiểm thử. Ngoài ra, Gatling có thể tích hợp với các công cụ DevOps như Jenkins, Kubernetes, giúp tự động hóa kiểm thử hiệu năng.

Ưu điểm lớn nhất của Gatling là hiệu suất cao, sử dụng ít tài nguyên hệ thống hơn so với JMeter khi chạy kiểm thử với số lượng lớn người dùng ảo. Công cụ này có thể chạy trên nhiều nền tảng khác nhau mà không yêu cầu cấu hình phức tạp. Ngoài ra, Gatling hỗ trợ tạo kịch bản kiểm thử bằng code, giúp kiểm thử linh hoạt và có thể tái sử dụng.

> Global Information



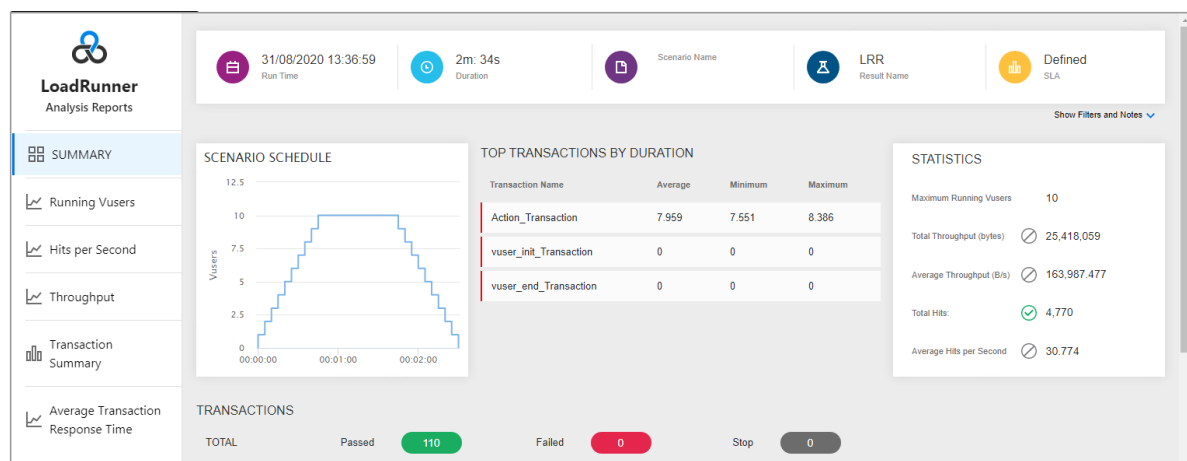
Hình 1.4. Công cụ kiểm thử Gatling

1.5.3. Công cụ kiểm thử LoadRunner

LoadRunner là một công cụ kiểm thử hiệu năng thương mại do Micro Focus phát triển, được sử dụng rộng rãi trong các doanh nghiệp lớn. Công cụ này giúp mô phỏng hàng nghìn người dùng ảo truy cập đồng thời vào hệ thống, giúp đánh giá hiệu suất, khả năng chịu tải và độ ổn định của phần mềm. LoadRunner hỗ trợ nhiều giao thức khác nhau như HTTP, HTTPS, SAP, Citrix, Oracle, Mobile, giúp kiểm thử hiệu năng trên nhiều loại hệ thống khác nhau.

Một tính năng quan trọng của LoadRunner là khả năng ghi lại và phát lại kịch bản kiểm thử, giúp giảm thời gian tạo test case và nâng cao hiệu quả kiểm thử. Công cụ này cũng cung cấp khả năng giám sát hiệu suất hệ thống theo thời gian thực, giúp phát hiện nhanh các vấn đề về hiệu năng và tối ưu hóa hệ thống trước khi triển khai chính thức. Ngoài ra, LoadRunner có thể tích hợp với các công cụ giám sát như Dynatrace, AppDynamics để theo dõi hiệu suất ứng dụng một cách toàn diện.

Ưu điểm lớn của LoadRunner là khả năng hỗ trợ nhiều giao thức phức tạp, có giao diện thân thiện và cung cấp báo cáo phân tích chi tiết về hiệu suất hệ thống. Công cụ này cũng cho phép mô phỏng hàng trăm nghìn người dùng mà không ảnh hưởng đến hiệu suất của máy kiểm thử, giúp kiểm tra hệ thống ở quy mô lớn.



Hình 1.5. Công cụ kiểm thử LoadRunner

1.6. Một số nghiên cứu liên quan đến đề tài

[Performance Testing](#) nhấn mạnh tầm quan trọng của việc kiểm thử hiệu suất trong giai đoạn phát triển phần mềm, nhằm đảm bảo rằng ứng dụng đáp ứng được các yêu cầu về hiệu suất và có thể chịu được tải trọng thực tế. Ngoài ra, bài giảng cũng đề cập đến các khía cạnh khác như độ tin cậy các phương pháp, kỹ thuật kiểm thử hiệu suất bao gồm kiểm thử tải, kiểm thử stress và kiểm thử độ bền nhằm xác định giới hạn và khả năng chịu đựng của hệ thống.

[ApacheJmeter](#) cung cấp một hướng dẫn chi tiết và toàn diện về cách sử dụng Apache JMeter, công cụ mã nguồn mở nổi bật trong lĩnh vực kiểm thử hiệu suất và tải cho các ứng dụng web, giới thiệu lịch sử và mục đích của JMeter, nhấn mạnh khả năng của nó trong việc mô phỏng tải nặng để đánh giá hiệu suất ứng dụng. Tài liệu còn hướng dẫn người dùng từng bước cài đặt và cấu hình JMeter trên các hệ điều hành khác nhau, đảm bảo rằng người dùng có thể dễ dàng thiết lập môi trường làm việc.

[Performance Testing with Jmeter Second Edition](#) của tác giả Bayo Erinle cung cấp hướng dẫn chi tiết về cách sử dụng Apache JMeter để thực hiện kiểm thử hiệu suất cho các ứng dụng web. Sách bao gồm các chủ đề như cài đặt JMeter, thiết lập các kịch bản kiểm thử, phân tích kết quả và tối ưu hóa hiệu suất. Phiên bản thứ hai này cập nhật các tính năng mới của JMeter và cung cấp các ví dụ thực tiễn để người đọc áp dụng vào công việc kiểm thử của mình.

[Performance testing \(load\) of web applications](#) tập trung vào việc kiểm thử hiệu suất tải của các ứng dụng web thông qua việc ưu tiên hóa các trường hợp kiểm thử. Bài viết nhấn mạnh tầm quan trọng của kiểm thử hiệu suất đối với các ứng dụng web và phân tích các điểm nghẽn của ứng dụng dựa trên phản ứng, phần mềm và các yếu tố khác. Phương pháp được đề xuất nhằm tối ưu hóa quy trình kiểm thử bằng cách tập trung vào các trường hợp kiểm thử có ảnh hưởng lớn nhất đến hiệu suất của ứng dụng.

CHƯƠNG 2. CÔNG CỤ KIỂM THỬ HIỆU NĂNG JMETER

Nội dung chương bao gồm giới thiệu tổng quan về JMeter, ưu nhược điểm của công cụ, cùng hướng dẫn cài đặt và thiết lập. Ngoài ra, chương cũng phân tích các thành phần chính và giới thiệu các chức năng quan trọng của JMeter trong kiểm thử hiệu năng. Cuối cùng, chương cung cấp một số ví dụ thực tế để minh họa cách áp dụng JMeter trong kiểm thử hiệu năng.

2.1. Giới thiệu về Jmeter

Apache JMeter là một công cụ kiểm thử hiệu năng mã nguồn mở, được phát triển bởi Apache Software Foundation. Ban đầu, JMeter được thiết kế để kiểm thử hiệu suất của các ứng dụng web, nhưng hiện nay nó hỗ trợ nhiều giao thức khác nhau như HTTP, HTTPS,... Công cụ này giúp mô phỏng nhiều người dùng truy cập đồng thời vào hệ thống để đánh giá khả năng chịu tải và độ ổn định của ứng dụng.

Công cụ này được sử dụng rộng rãi trong kiểm thử hiệu năng nhờ khả năng tạo và thực thi kịch bản kiểm thử với số lượng người dùng lớn. Nó cung cấp các báo cáo chi tiết về thời gian phản hồi, tỷ lệ lỗi, số lượng yêu cầu mỗi giây và các chỉ số quan trọng khác giúp đánh giá hiệu suất hệ thống. Bên cạnh đó, nó còn hỗ trợ tích hợp với nhiều công cụ giám sát và phân tích dữ liệu như Grafana, Prometheus để theo dõi hiệu năng theo thời gian thực.

Với giao diện đồ họa trực quan và khả năng mở rộng linh hoạt, JMeter giúp người kiểm thử dễ dàng thiết lập và thực hiện các bài kiểm thử phức tạp. Nó có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux mà không yêu cầu cài đặt phức tạp. Ngoài ra, JMeter cũng hỗ trợ kiểm thử dựa trên dòng lệnh, giúp tự động hóa và tích hợp vào quy trình DevOps.

Tuy nhiên, JMeter cũng có một số hạn chế như tiêu tốn tài nguyên khi chạy kiểm thử với số lượng lớn người dùng ảo. Việc tạo kịch bản kiểm thử phức tạp đòi

hỏi người dùng phải có hiểu biết nhất định về kiểm thử hiệu năng và cách thức hoạt động của giao thức mạng. Dù vậy, với những tính năng mạnh mẽ và khả năng mở rộng cao, JMeter vẫn là một trong những công cụ kiểm thử hiệu năng phổ biến nhất hiện nay.

2.2. Ưu và nhược điểm của Jmeter

2.2.1. Ưu điểm

JMeter là một công cụ mã nguồn mở và hoàn toàn miễn phí, giúp người dùng dễ dàng tiếp cận mà không cần chi trả phí bản quyền. Nhờ vào cộng đồng phát triển mạnh mẽ, JMeter liên tục được cập nhật với nhiều tính năng mới, giúp nâng cao hiệu quả kiểm thử hiệu năng phần mềm.

Một trong những điểm mạnh của JMeter là khả năng hỗ trợ nhiều giao thức khác nhau, không chỉ giới hạn ở ứng dụng web, nhờ đó còn có thể áp dụng để kiểm thử hiệu suất cho nhiều loại hệ thống khác nhau, từ trang web đến dịch vụ lập trình giao diện ứng dụng, cơ sở dữ liệu và các ứng dụng nhúng.

Công cụ cũng được đánh giá cao nhờ khả năng mở rộng mạnh mẽ thông qua hệ thống phân bổ trợ đa dạng. Người dùng có thể cài đặt các phân bổ trợ để bổ sung thêm chức năng, giúp kiểm thử hiệu suất hiệu quả hơn. Ngoài ra, JMeter có thể tích hợp với các công cụ giám sát hiệu suất như Grafana và Prometheus để theo dõi trạng thái hệ thống theo thời gian thực.

Giao diện trực quan của JMeter giúp người dùng dễ dàng thiết lập kịch bản kiểm thử mà không cần phải viết mã. Điều này đặc biệt hữu ích cho những người mới làm quen với kiểm thử hiệu năng. Bên cạnh đó, JMeter còn hỗ trợ chế độ dòng lệnh, cho phép thực hiện kiểm thử với quy mô lớn mà không cần mở giao diện đồ họa, giúp tiết kiệm tài nguyên hệ thống.

JMeter cung cấp hệ thống báo cáo chi tiết và trực quan, giúp người dùng dễ dàng phân tích kết quả kiểm thử. Sau khi hoàn thành kiểm thử, JMeter sẽ tạo ra các biểu đồ, bảng thống kê và báo cáo hiệu suất của hệ thống, giúp đội ngũ kiểm thử

nhANH chóng xác định các điểm nghẽn hiệu năng và tối ưu hóa ứng dụng.

2.2.2. Nhược điểm

Mặc dù mạnh mẽ, JMeter vẫn có một số hạn chế, đặc biệt là về tiêu thụ tài nguyên hệ thống. Khi thực hiện kiểm thử với số lượng lớn người dùng ảo, JMeter có thể chiếm dụng nhiều bộ nhớ truy cập ngẫu nhiên và bộ xử lý trung tâm, làm giảm hiệu suất của máy kiểm thử. Điều này gây khó khăn khi thực hiện các bài kiểm thử quy mô lớn mà không có sự hỗ trợ của nhiều máy chủ.

Một nhược điểm khác của JMeter là giao diện đồ họa có thể trở nên chậm khi làm việc với kịch bản kiểm thử phức tạp. Khi số lượng yêu cầu tăng lên, hiệu suất của công cụ có thể bị ảnh hưởng, gây khó khăn trong việc thao tác và phân tích dữ liệu. Điều này khiến nhiều người dùng phải chuyển sang chế độ dòng lệnh để tối ưu hóa quá trình kiểm thử.

Công cụ này không hỗ trợ kiểm thử hiệu suất theo thời gian thực một cách đầy đủ. Mặc dù có thể kết hợp với các công cụ giám sát, JMeter không cung cấp khả năng giám sát hệ thống ngay trong quá trình kiểm thử. Điều này đòi hỏi người dùng phải thiết lập các công cụ bên ngoài để theo dõi tài nguyên hệ thống. Điều này có thể gây bất tiện khi cần theo dõi và phân tích hiệu suất hệ thống ngay lập tức để kịp thời điều chỉnh kịch bản kiểm thử.

Việc thiết lập kịch bản kiểm thử trong JMeter có thể phức tạp đối với người mới bắt đầu. Mặc dù giao diện kéo thả giúp đơn giản hóa quá trình tạo kịch bản, nhưng để kiểm thử nâng cao, người dùng cần hiểu rõ về cách hoạt động của các giao thức mạng và cấu trúc hệ thống. Điều này đòi hỏi một lượng kiến thức kỹ thuật nhất định để khai thác tối đa khả năng của JMeter.

Cuối cùng, JMeter không phải là công cụ lý tưởng để kiểm thử ứng dụng có giao diện người dùng đồ họa. Công cụ này chủ yếu tập trung vào kiểm thử hiệu suất phía máy chủ, do đó không thể mô phỏng hành vi của người dùng thực tế trên giao diện ứng dụng. Nếu cần kiểm thử hiệu suất của ứng dụng có giao diện đồ họa,

người dùng có thể phải kết hợp JMeter với các công cụ khác như Selenium.

2.3. Cài đặt và thiết lập Jmeter

Là một công cụ mã nguồn mở, không yêu cầu cài đặt phức tạp, chỉ cần giải nén tệp tải về và chạy tệp thực thi là có thể sử dụng ngay. Tuy nhiên, do JMeter được viết bằng Java, nên máy tính cần có sẵn môi trường chạy Java với phiên bản phù hợp. Truy cập <https://www.oracle.com/java/technologies/downloads/> để cài đặt JDK, môi trường chạy cho Java.

Sau khi cài đặt môi trường chạy Java thành công, hãy truy cập vào đường dẫn https://jmeter.apache.org/download_jmeter.cgi, đây là trang chủ chính thức của Jmeter cho phép cài đặt phiên bản kiểm thử phù hợp với nhu cầu, chọn phiên bản mới nhất và mục Binaries, tùy vào thiết bị sử dụng để chọn tải mục tgz cho Linux hay zip cho máy tính Window.

Thư mục được tải về có thể đưa vào ổ C để dễ dàng quản lý, sau khi giải nén thành công thì mở file /bin/ApacheJmeter.rar để chạy công cụ. Người dùng có thể cài đặt thêm các plugin để mở rộng tính năng của JMeter. Apache JMeter có một hệ thống plugin phong phú, hỗ trợ nhiều loại kiểm thử khác nhau như kiểm thử cơ sở dữ liệu, kiểm thử dịch vụ web hoặc kiểm thử giao thức mạng. Các plugin này có thể được cài đặt thông qua JMeter Plugins Manager, giúp đơn giản hóa quá trình mở rộng công cụ.

Apache JMeter 5.6.3 (Requires Java 8+)

Binaries

[apache-jmeter-5.6.3.tgz sha512 pgp](#)
[apache-jmeter-5.6.3.zip sha512 pgp](#)

Source

[apache-jmeter-5.6.3_src.tgz sha512 pgp](#)
[apache-jmeter-5.6.3_src.zip sha512 pgp](#)

Hình 2.1. Giao diện tải về công cụ kiểm thử Jmeter

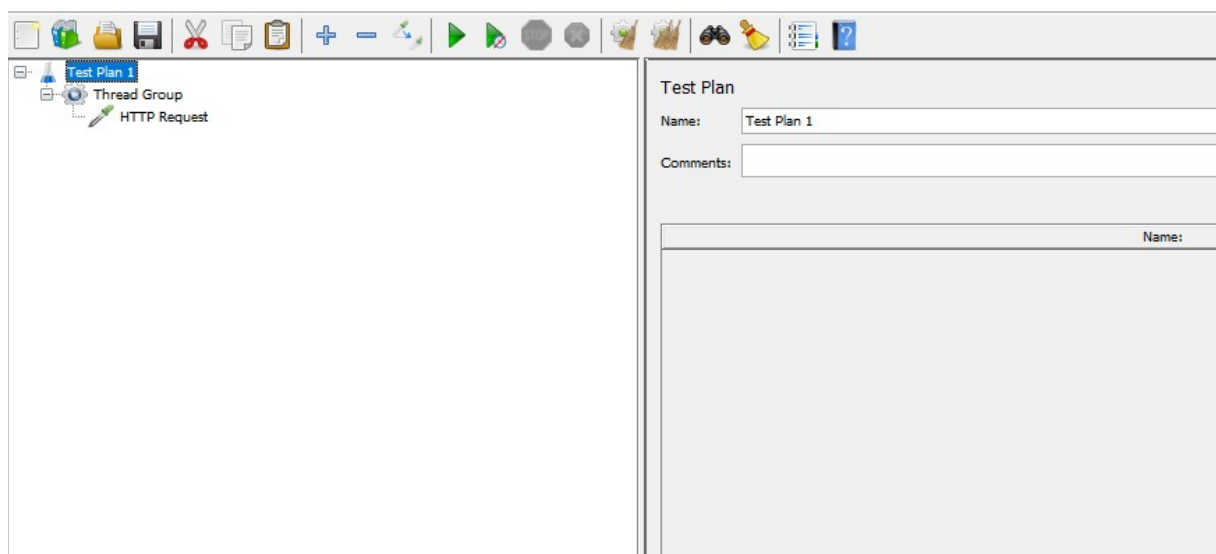
2.4. Các thành phần chính của Jmeter

2.4.1. Test Plan

Test Plan là thành phần quan trọng nhất trong JMeter, đóng vai trò như một tập hợp chứa tất cả các phần tử cần thiết để thực hiện một kịch bản kiểm thử hiệu năng. Một Test Plan bao gồm các nhóm luồng, các bộ lấy mẫu, bộ lắng nghe, bộ hẹn giờ, bộ xác nhận và các thành phần cấu hình khác. Đây là nơi người dùng thiết lập các thông số kiểm thử, định nghĩa cách thức mô phỏng tải lên hệ thống, và kiểm soát toàn bộ quá trình kiểm thử.

Khi tạo một Test Plan, người dùng cần xác định rõ mục tiêu kiểm thử, lựa chọn số lượng người dùng ảo, thiết lập thời gian chạy, tần suất gửi yêu cầu và cách xử lý kết quả phản hồi từ hệ thống. Ngoài ra có thể được cấu hình để sử dụng các biến số, giúp tái sử dụng dữ liệu linh hoạt và giảm thiểu việc nhập thủ công.

Một Test Plan trong JMeter có thể được lưu lại dưới dạng tệp XML giúp người dùng dễ dàng chỉnh sửa hoặc chia sẻ với các nhóm kiểm thử khác. Việc thiết lập một kế hoạch kiểm thử hợp lý sẽ giúp đánh giá chính xác hiệu suất hệ thống, tối ưu tài nguyên và phát hiện sớm các vấn đề về hiệu năng trước khi triển khai thực tế.



Hình 2.1. Giao diện của một Test Plan

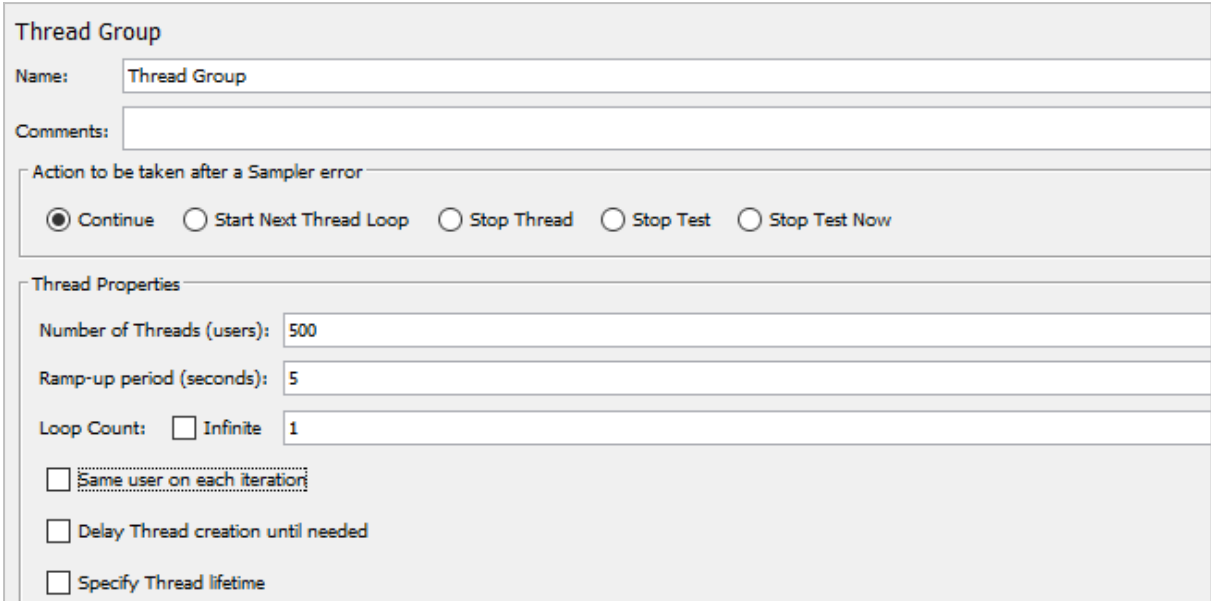
2.4.2. Thread Group

Thread Group trong JMeter là một thành phần quan trọng giúp mô phỏng số lượng người dùng ảo gửi yêu cầu đến hệ thống để kiểm thử hiệu suất. Mỗi Thread Group đại diện cho một tập hợp các luồng, trong đó mỗi luồng hoạt động như một người dùng độc lập truy cập vào ứng dụng.

Số lượng luồng trong Thread Group xác định số lượng người dùng ảo sẽ thực hiện các yêu cầu đến hệ thống trong quá trình kiểm thử. Mỗi luồng hoạt động như một người dùng riêng biệt, gửi yêu cầu đồng thời đến máy chủ mục tiêu.

Thời gian tăng dần là khoảng thời gian mà JMeter sử dụng để khởi tạo tất cả các luồng. Thông số này giúp kiểm soát tốc độ tải lên hệ thống, tránh tình trạng tất cả người dùng ảo gửi yêu cầu cùng một lúc. Ví dụ, nếu có 100 luồng và thời gian tăng dần là 10 giây, JMeter sẽ tạo một luồng mới sau mỗi 0,1 giây.

Số lần lặp xác định số lần mỗi luồng sẽ lặp lại các bước kiểm thử trong quá trình chạy. Nếu đặt giá trị cụ thể, mỗi luồng sẽ thực hiện đúng số lần lặp đó rồi kết thúc. Ngược lại, nếu chọn chế độ vô hạn, luồng sẽ tiếp tục gửi yêu cầu cho đến khi kiểm thử được dừng thủ công.



The screenshot shows the 'Thread Group' configuration window in JMeter. It includes a 'Name' field set to 'Thread Group', a 'Comments' field, and a section for 'Action to be taken after a Sampler error' with radio buttons for 'Continue' (selected), 'Start Next Thread Loop', 'Stop Thread', 'Stop Test', and 'Stop Test Now'. Below this is the 'Thread Properties' section with fields for 'Number of Threads (users)' (500), 'Ramp-up period (seconds)' (5), and 'Loop Count' (1, with an 'Infinite' checkbox). There are also three checkboxes: 'Same user on each iteration' (checked), 'Delay Thread creation until needed', and 'Specify Thread lifetime'.

Hình 2.2. Giao diện của một Thread Group

2.4.3. Sampler

Sampler trong JMeter là thành phần chịu trách nhiệm gửi các yêu cầu đến máy chủ và nhận phản hồi từ hệ thống đích. Mỗi Sampler mô phỏng một loại yêu cầu cụ thể, chẳng hạn như HTTP Request để kiểm thử website, JDBC Request để truy vấn cơ sở dữ liệu, hay FTP Request để kiểm tra truyền tải tập tin. Sampler giúp người dùng mô phỏng nhiều kịch bản khác nhau, từ việc tải trang web đến thực hiện giao dịch phức tạp trên hệ thống, nhằm đánh giá hiệu suất ứng dụng một cách toàn diện.

Server Name hoặc IP trong JMeter là thông tin xác định địa chỉ của máy chủ mà bạn muốn gửi yêu cầu kiểm thử. Khi kiểm thử một ứng dụng web chạy trên máy tính cá nhân có thể sử dụng IP “127.0.0.1”.

Port Number là số xác định dịch vụ chạy trên máy chủ. Các ứng dụng web thường chạy trên các cổng như 8000, 8080 hoặc 3000, tùy vào cấu hình thiết bị.

Path là đường dẫn cụ thể của tài nguyên trên máy chủ mà bạn muốn kiểm thử và Method là cách mà JMeter gửi yêu cầu đến máy chủ. Có bốn phương thức phổ biến: GET, POST, PUT và DELETE.

The screenshot displays the configuration for an HTTP Request sampler in JMeter. The 'Name' field is filled with 'Login'. Below it, the 'Web Server' section includes a 'Protocol' dropdown set to 'http:' and a 'Server Name or IP' field containing '127.0.0.1'. The 'HTTP Request' section shows the 'Method' dropdown set to 'GET' and the 'Path' field set to '/login'. There are checkboxes for 'Redirect Automatically', 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data', and 'Browser-compatible headers'. The 'Parameters' tab is selected, showing a table with two parameters: 'user_name' with value 'vanhoa123' and 'password' with value '123456'.

Name:	
user_name	vanhoa123
password	123456

Hình 2.3. Giao diện của một Sampler

2.4.4. Listener

Listener trong JMeter là một thành phần quan trọng giúp thu thập và hiển thị kết quả kiểm thử hiệu suất. Nó cho phép người dùng theo dõi phản hồi từ máy chủ, phân tích dữ liệu và trực quan hóa kết quả dưới nhiều dạng khác nhau như bảng, biểu đồ hoặc tệp nhật ký. Listener giúp đánh giá hiệu suất hệ thống dựa trên các thông số như thời gian phản hồi, số lượng yêu cầu thành công hay thất bại và thông lượng hệ thống. Có nhiều loại Listener nhưng thông dụng là 3 loại sau đây.

View Results Tree là một Listener hiển thị chi tiết phản hồi từ máy chủ theo dạng cây. Nó giúp người kiểm thử theo dõi từng yêu cầu HTTP, hiển thị cả nội dung phản hồi và mã trạng thái để xác định lỗi hoặc kiểm tra nội dung trả về. Đây là công cụ hữu ích khi cần kiểm tra tính chính xác của phản hồi từ máy chủ.

Summary Report cung cấp cái nhìn tổng quan về kết quả kiểm thử, bao gồm số lượng yêu cầu gửi đi, tỷ lệ lỗi, thời gian phản hồi trung bình và thông lượng hệ thống. Báo cáo này giúp đánh giá nhanh hiệu suất tổng thể của hệ thống mà không đi vào quá nhiều chi tiết, phù hợp cho các bài kiểm thử có quy mô lớn.

Aggregate Report hiển thị dữ liệu kiểm thử dưới dạng bảng, bao gồm các chỉ số quan trọng như thời gian trung bình, thời gian phản hồi tối thiểu/tối đa, độ lệch chuẩn và phần trăm lỗi.

Summary Report

Name:

Comments:

☐ Write results to file / ☐ Read from file

Filename

Label	# Samples	Average	Min	Max
Login	100	3829	199	7088
TOTAL	100	3829	199	7088

Hình 2.4. Giao diện phản hồi trạng thái toàn bộ yêu cầu

2.5. Ví dụ kiểm thử hiệu năng với Jmeter

Thực hiện kiểm thử hiệu năng đăng nhập cho website quản lý nhà hàng, đầu tiên cần tạo một Test Plan mới quản lý quá trình kiểm thử. Sau đó tạo một Thread Group đại diện cho nhóm người dùng ảo thực hiện chức năng đăng nhập. Cấu hình số lượng người dùng ảo là 500 người, thời gian là 5s với vòng lặp là 1 được thể hiện như Hình 2.2. Giao diện của một Thread Group.

Sau khi cấu hình được Thread Group, tiến đến thiết lập Sampler, vì chức năng đăng nhập yêu cầu HTTP Request và tiến hành kiểm thử trên cổng của máy tính nên cấu hình địa chỉ IP, cổng, đường dẫn, liên kết và hai thông số truyền vào là tên đăng nhập và nhập khẩu được thể hiện như Hình 2.3. Giao diện của một Sampler.

Tạo Listener để hiển thị kết quả phản hồi từ hệ thống, mã trạng thái HTTP và nội dung trả về.

Hình dưới đây là một Listener trong JMeter dùng để theo dõi chi tiết kết quả kiểm thử từng yêu cầu. Danh sách các phản hồi được hiển thị với các biểu tượng màu xanh đại diện cho thành công và màu đỏ là thất bại. Điều này giúp người kiểm thử dễ dàng nhận biết những yêu cầu gặp lỗi.

Sample #	Start Time	Thread Name	Label	Sample Time(...	Status	Bytes	Sent Bytes	Latency	Connect Time...
4	08:43:59.639	HomePage_Us...	HTTP Request ...	682		108186	120	681	0
5	08:43:59.654	HomePage_Us...	HTTP Request ...	824		108186	120	823	1
6	08:43:59.655	HomePage_Us...	HTTP Request ...	995		108186	120	994	1
7	08:43:59.672	HomePage_Us...	HTTP Request ...	1132		108186	120	1130	0
8	08:43:59.672	HomePage_Us...	HTTP Request ...	1287		108186	120	1285	1
9	08:43:59.683	HomePage_Us...	HTTP Request ...	1442		108186	120	1440	0
10	08:43:59.693	HomePage_Us...	HTTP Request ...	1585		108186	120	1583	1
11	08:43:59.703	HomePage_Us...	HTTP Request ...	1744		108186	120	1743	0
12	08:43:59.712	HomePage_Us...	HTTP Request ...	1891		108186	120	1890	1
13	08:43:59.722	HomePage_Us...	HTTP Request ...	2027		108186	120	2026	1
14	08:44:01.768	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
15	08:44:01.778	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
16	08:44:01.790	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
17	08:44:01.801	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
18	08:44:01.806	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
19	08:44:01.816	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
20	08:44:01.828	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
21	08:44:01.842	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
22	08:44:01.848	HomePage_Us...	HTTP Request ...	0		2595	0	0	0
23	08:44:01.858	HomePage_Us...	HTTP Request ...	0		2595	0	0	0
24	08:44:01.869	HomePage_Us...	HTTP Request ...	1		2595	0	0	1
25	08:44:01.886	HomePage_Us...	HTTP Request ...	0		2595	0	0	0
26	08:44:01.892	HomePage_Us...	HTTP Request ...	0		2595	0	0	0
27	08:43:59.732	HomePage_Us...	HTTP Request ...	2161		108186	120	2160	1
28	08:44:01.897	HomePage_Us...	HTTP Request ...	0		2595	0	0	0

Hình 2.5. Giao diện phản hồi của từng đợt yêu cầu

CHƯƠNG 3. KIỂM THỬ HIỆU NĂNG HỆ THỐNG QUẢN LÝ NHÀ HÀNG SỬ DỤNG JMETER

Với mục đích trình bày quy trình kiểm thử hiệu năng cho hệ thống quản lý nhà hàng sử dụng Apache JMeter với các nhiệm vụ như giới thiệu hệ thống quản lý nhà hàng, xây dựng kế hoạch kiểm thử, thiết lập các kịch bản kiểm thử và thực hiện quy trình kiểm thử. Qua đó, chương sẽ phân tích kết quả thu được để đánh giá hiệu suất của hệ thống, từ đó đưa ra những nhận định và khuyến nghị nhằm cải thiện chất lượng dịch vụ và trải nghiệm của khách hàng.

3.1. Kiểm thử hiệu năng website nhà hàng ResOnline

3.1.1. Giới thiệu website nhà hàng ResOnline

Nhà hàng ResOnline là một hệ thống nhà hàng chuyên cung cấp các món ăn nhanh chuẩn phong cách Châu Âu, với đa dạng các danh mục món ăn như gà rán, mì ý, burger với giá cả phù hợp với mọi đối tượng thực khách.



Hình 3.1. Giao diện thực đơn nhà hàng ResOnline

Khách hàng có thể đặt bàn trực tuyến với nhà hàng với các món ăn muốn dùng bữa giúp nhà hàng chuẩn bị nhanh chóng, phục vụ đúng giờ và mang lại trải nghiệm ẩm thực trọn vẹn cho thực khách.

THÔNG TIN ĐẶT BÀN

Chọn cơ sở

Cơ sở 1 - 121 - Lê Hồng Phong - Hưng Bình - Tp.Vinh - Nghệ An

Họ tên *

Số điện thoại *

Số lượng khách *

Chọn ngày *

Khung giờ

Ghi chú

THÔNG TIN MÓN ĂN

MÓN ĂN	THÀNH TIỀN
Hot dog x 1	25.000 đ
Burger gà giòn x 1	30.000 đ
Burger tôm x 1	40.000 đ
Gà sốt cay x 2	70.000 đ
Thành tiền	165.000 đ
Khuyến mãi	0 đ
Tổng tiền	165.000 đ

THANH TOÁN

Hình 3.2. Giao diện đặt bàn nhà hàng ResOnline

Một tính năng hiện đại của nhà hàng là thực khách có thể trò chuyện với AI thông minh để hỏi về các thông tin liên hệ, thực đơn món ăn, combo ưu đãi của nhà hàng, giúp nâng cao trải nghiệm khách hàng, tạo sự tiện lợi và chuyên nghiệp khi tương tác với nhà hàng mọi lúc, mọi nơi.

Trò chuyện với AI Restaurant

Thực đơn nhà hàng có gì

Chào bạn! Thực đơn của nhà hàng mình có:

* **Món tráng miệng:** Kem dâu

* **Món chính:** Hot dog, Burger gà giòn, Burger tôm, Gà sốt cay, Mỳ ý sốt kem, Gà giòn vui vẻ

* **Món phụ:** Cơm trắng, Khoai tây lắc, Pepsi

Bạn muốn thử món nào ạ? Mình gợi ý món Burger gà giòn đang được nhiều người yêu thích đó!

Nhập tin nhắn...

Gửi

Hình 3.3. Giao diện chatbot AI nhà hàng ResOnline

3.1.2. Kế hoạch kiểm thử

Kế hoạch kiểm thử nhằm đánh giá hiệu năng, khả năng chịu tải và độ ổn định của hệ thống website trong các tình huống sử dụng thực tế với nhiều người dùng đồng thời. Các bài kiểm thử được xây dựng theo từng chức năng quan trọng của hệ thống như: trang chủ, trang thực đơn, giỏ hàng, đặt bàn, thanh toán, tìm kiếm và chatbot. Mỗi trường hợp kiểm thử được xác định rõ mục tiêu, đối tượng và loại kiểm thử phù hợp.

Kiểm thử	Đối tượng	Loại kiểm thử	Mô tả
Trang chủ (TC01)	Giao diện trang chủ	Kiểm thử tải	Kiểm tra khả năng tải trang chủ khi có nhiều người truy cập đồng thời
Thực đơn (TC02)	Giao diện trang thực đơn	Kiểm thử tải	Kiểm tra khả năng tải trang thực đơn khi có nhiều người truy cập đồng thời
Tìm kiếm món ăn (TC03)	Hệ thống chức năng tìm kiếm món ăn trên trang thực đơn	Kiểm thử hiệu suất	Kiểm tra khả năng tải danh sách thực đơn khi có nhiều truy vấn tìm kiếm dữ liệu đồng thời
Gợi ý món ăn liên quan (TC04)	Hệ thống xử lý gợi ý món ăn liên quan	Kiểm thử tải, độ trễ	Kiểm tra khả năng phản hồi gợi ý hệ thống xử lý gợi ý món ăn nếu có nhiều khách hàng xem thông tin món ăn
Tìm kiếm bài viết (TC05)	Hệ thống chức năng tìm kiếm bài viết trên trang bài viết	Kiểm thử hiệu suất	Kiểm tra khả năng tải danh sách bài viết khi có nhiều truy vấn tìm kiếm dữ liệu đồng thời

Giỏ hàng đặt bàn (TC06)	Hệ thống quản lý thao tác giỏ hàng	Kiểm thử tải, chịu tải	Kiểm tra khả năng hoạt động ổn định vào giờ cao điểm với nhiều người dùng thao tác đồng thời
Thanh toán (TC07)	Trang thanh toán đơn đặt bàn	Kiểm thử tải, hiệu suất	Kiểm tra khả năng tải trang thanh toán khi có nhiều người dùng thanh toán
Đặt bàn (TC08)	Giao diện đặt bàn và xử lý đặt bàn	Kiểm thử tải, chịu tải, độ bền, hiệu suất	Đánh giá hiệu suất xử lý đặt bàn, thời gian phản hồi, khả năng chịu tải khi có nhiều người đặt bàn cùng lúc
Chatbot (TC09)	Giao diện phản hồi khách hàng	Kiểm thử tải, chịu tải, hiệu suất	Kiểm tra xử lý phản hồi, chịu tải, tốc độ phản hồi khi có nhiều khách hàng hỏi
Đăng nhập (TC10)	Hệ thống xử lý đăng nhập tài khoản	Kiểm thử hiệu suất	Kiểm tra tốc độ xử lý đăng nhập tài khoản người dùng

Bảng 3.1. Kế hoạch kiểm thử nhà hàng ResOnline

3.1.3. Kịch bản kiểm thử

- Kịch bản kiểm thử trang chủ

Kịch bản kiểm thử trang chủ	
Mục tiêu	Đánh giá khả năng phản hồi của giao diện trang chủ khi có lượng lớn người dùng truy cập đồng thời, tượng trưng cho lượng khách hàng vào giờ cao điểm.
Dữ liệu đầu vào	- Số lượt truy cập đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000

	- Địa chỉ truy cập: http://127.0.0.1:8000/home
Các bước thực hiện	<p>Bước 1: Tạo một kịch bản trong Jmeter để thực hiện kiểm thử</p> <p>Bước 2: Cấu hình lần lượt với các số lượng người dùng truy cập cùng lúc vào trang chủ</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi, hiệu suất</p> <p>Bước 4: Lặp lại các bước trên với số lượng người dùng truy cập tăng dần</p>
Kết quả mong đợi	Tất cả phản hồi với thời gian thấp, tỉ lệ lỗi dưới 5%, giao diện trả về đầy đủ, không lỗi với hệ thống ổn định, không treo.

Bảng 3.2. Kịch bản kiểm thử trang chủ

- Kịch bản kiểm thử thực đơn

Kịch bản kiểm thử thực đơn	
Mục tiêu	Kiểm tra tốc độ xử lý và phản hồi của hệ thống khi có nhiều người dùng truy cập trang thực đơn cùng lúc.
Dữ liệu đầu vào	<p>- Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000</p> <p>- Địa chỉ kiểm thử thực đơn: http://127.0.0.1:8000/item</p>
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng truy cập với số lượng tương ứng</p> <p>Bước 2: Tạo yêu cầu đến máy chủ truy cập trang thực đơn và cấu hình kiểm thử với lượng người dùng tương ứng</p> <p>Bước 3: Ghi nhận thời gian xử lý, kết quả trả về</p> <p>Bước 4: Lặp lại các bước trên với lượng người dùng tăng dần</p>
Kết quả mong đợi	Tất cả phản hồi với tỉ lệ lỗi dưới 3%, giao diện trả về đầy đủ, không lỗi với hệ thống hoạt động ổn định, không treo.

Bảng 3.3. Kịch bản kiểm thử thực đơn

- Kịch bản kiểm thử tìm kiếm món ăn

Kịch bản kiểm thử tìm kiếm món ăn	
Mục tiêu	Đánh giá khả năng xử lý truy vấn tìm kiếm món ăn khi có nhiều người dùng nhập từ khóa tìm kiếm đồng thời. Đảm bảo kết quả tìm kiếm trả về nhanh chóng và hệ thống phản hồi ổn định.
Dữ liệu đầu vào	<ul style="list-style-type: none"> - Từ khóa: "gà", "com", "bò", "salad"... - Số lượng truy vấn đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000 - Địa chỉ truy vấn: http://127.0.0.1:8000/item?search=
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng</p> <p>Bước 2: Thêm yêu cầu máy chủ sử dụng các từ khóa tìm kiếm</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi, kết quả trả về</p> <p>Bước 4: Lặp lại quy trình với số lượng người dùng tăng dần</p>
Kết quả mong đợi	Hệ thống phản hồi với tỉ lệ lỗi dưới 3%, dữ liệu trả về đầy đủ và chính xác theo từ khóa tìm kiếm.

Bảng 3.4. Kịch bản kiểm thử tìm kiếm món ăn

- Kịch bản kiểm thử gợi ý món ăn liên quan

Kịch bản kiểm thử gợi ý món ăn liên quan	
Mục tiêu	Đánh giá khả năng xử lý gợi ý món ăn khi có nhiều người dùng truy cập đồng thời trang chi tiết món ăn. Đảm bảo hệ thống phản hồi nhanh chóng, chính xác và hệ thống hoạt động ổn định.
Dữ liệu đầu vào	<ul style="list-style-type: none"> - Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000 - Địa chỉ truy vấn: http://127.0.0.1:8000/api/recommendation - Các mã món ăn có thể thay đổi: 1, 2, 3,...
Các bước	Bước 1: Tạo nhóm người dùng với số lượng người tương ứng

thực hiện	<p>Bước 2: Thêm yêu cầu máy chủ với hệ thống gợi ý món ăn</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi, dữ liệu phản hồi</p> <p>Bước 4: Tăng dần số lượng người dùng để kiểm tra hiệu năng</p>
Kết quả mong đợi	API phản hồi với tỉ lệ lỗi dưới 3%, dữ liệu gợi ý trả về đầy đủ và chính xác theo mã món ăn, hệ thống hoạt động ổn định, không treo khi có nhiều người truy cập đồng thời.

Bảng 3.5. Kịch bản kiểm thử gợi ý món ăn liên quan

- Kịch bản kiểm thử tìm kiếm bài viết

Kịch bản kiểm thử tìm kiếm bài viết	
Mục tiêu	Đánh giá khả năng phản hồi và hiệu suất của hệ thống khi nhiều người dùng thực hiện tìm kiếm bài viết cùng lúc. Đảm bảo kết quả tìm kiếm được trả về chính xác và hệ thống hoạt động ổn định, không trả về phản hồi lỗi.
Dữ liệu đầu vào	<p>- Từ khóa: “ẩm thực”, “sự kiện”, “khuyến mãi”, “món ngon”,...</p> <p>- Số lượng truy vấn đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000</p> <p>- Địa chỉ truy vấn: http://127.0.0.1:8000/posts/search?k=</p>
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng</p> <p>Bước 2: Thêm yêu cầu máy chủ với các từ khóa khác nhau</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi, dữ liệu phản hồi</p> <p>Bước 4: Tăng dần số lượng người dùng để kiểm tra hiệu năng</p>
Kết quả mong đợi	Hệ thống trả về kết quả với tỉ lệ lỗi dưới 3%, kết quả đúng với từ khóa tìm kiếm, không thiếu hoặc sai nội dung. Giao diện vẫn hoạt động ổn định, không bị treo hoặc sập.

Bảng 3.6. Kịch bản kiểm thử tìm kiếm bài viết

- Kịch bản kiểm thử giỏ hàng đặt bàn

Kịch bản kiểm thử giỏ hàng đặt bàn	
Mục tiêu	Đánh giá khả năng xử lý thao tác thêm, xóa, cập nhật giỏ hàng khi có nhiều người dùng thao tác đồng thời vào giờ cao điểm.
Dữ liệu đầu vào	<ul style="list-style-type: none"> Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000 Hành động: Thêm món vào giỏ, cập nhật số lượng, xóa món Địa chỉ truy vấn: http://127.0.0.1:8000/cart/update
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng</p> <p>Bước 2: Thực hiện thao tác cập nhật giỏ hàng đặt bàn</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi và độ ổn định</p> <p>Bước 4: Lặp lại với số lượng người dùng tăng dần</p>
Kết quả mong đợi	Thao tác hoàn thành với tỷ lệ lỗi dưới 3%, hệ thống không bị treo hoặc mất dữ liệu, cập nhật giỏ hàng chính xác.

Bảng 3.7. Kịch bản kiểm thử giỏ hàng đặt bàn

- Kịch bản kiểm thử thanh toán

Kịch bản kiểm thử thanh toán	
Mục tiêu	Kiểm tra hiệu suất xử lý và khả năng phản hồi của trang thanh toán khi có nhiều người thanh toán cùng lúc.
Dữ liệu đầu vào	<ul style="list-style-type: none"> Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000 Địa chỉ truy vấn: http://127.0.0.1:8000/checkout
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng</p> <p>Bước 2: Gửi yêu cầu thanh toán qua địa chỉ truy vấn</p> <p>Bước 3: Ghi nhận thời gian xử lý và phản hồi hệ thống</p> <p>Bước 4: Lặp lại với số lượng người dùng tăng dần</p>

Kết quả mong đợi	Trang phản hồi với tỷ lệ lỗi dưới 3%, hệ thống không bị quá tải, đơn hàng được xử lý đúng.
------------------	--

Bảng 3.8. Kịch bản kiểm thử thanh toán

- Kịch bản kiểm thử đặt bàn

Kịch bản kiểm thử đặt bàn	
Mục tiêu	Đánh giá hiệu suất, độ bền và khả năng xử lý đồng thời của hệ thống khi nhiều người dùng đặt bàn cùng lúc.
Dữ liệu đầu vào	- Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000 - Địa chỉ truy vấn: http://127.0.0.1:8000/booking
Các bước thực hiện	Bước 1: Tạo nhóm người dùng với số lượng người tương ứng Bước 2: Gửi xác nhận đặt bàn với thông tin hợp lệ Bước 3: Ghi nhận thời gian xử lý, độ chính xác và phản hồi Bước 4: Kiểm tra trạng thái hệ thống sau mỗi đợt
Kết quả mong đợi	Hệ thống xử lý đơn đặt bàn với tỉ lệ lỗi dưới 3%, không mất dữ liệu, không bị trùng lặp, chịu tải tốt trong thời gian dài.

Bảng 3.9. Kịch bản kiểm thử đặt bàn

- Kịch bản kiểm thử chatbot

Kịch bản kiểm thử chatbot	
Mục tiêu	Kiểm tra hiệu suất và khả năng chịu tải của chatbot khi có nhiều người dùng gửi tin nhắn cùng lúc.
Dữ liệu đầu vào	- Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000 - Địa chỉ truy vấn: http://127.0.0.1:8000/checkout - Tin nhắn: “Giờ mở cửa?”, “Có món gà không?”,...
Các bước	Bước 1: Tạo nhóm người dùng với số lượng người tương ứng

Thực hiện	<p>Bước 2: Gửi thông tin câu hỏi đến hệ thống chatbot</p> <p>Bước 3: Ghi nhận tốc độ phản hồi, độ chính xác</p> <p>Bước 4: Lặp lại với số lượng người dùng tăng dần</p>
Kết quả mong đợi	Chatbot phản hồi đúng nội dung, tỉ lệ lỗi dưới 3%, không bị trễ hoặc treo.

Bảng 3.10. Kịch bản kiểm thử chatbot

- Kịch bản kiểm thử đăng nhập

Kịch bản kiểm thử đăng nhập	
Mục tiêu	Kiểm tra khả năng xử lý đăng nhập tài khoản người dùng khi có nhiều lượt yêu cầu đăng nhập hệ thống cùng lúc vào giờ cao điểm của nhà hàng.
Dữ liệu đầu vào	<p>- Số lượng yêu cầu đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000</p> <p>- Địa chỉ đăng nhập: http://127.0.0.1:8000/login</p> <p>- Dữ liệu: Tài khoản hệ thống hợp lệ</p>
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng để tiến hành thử nghiệm cùng lúc</p> <p>Bước 2: Gửi yêu cầu đăng nhập hệ thống</p> <p>Bước 3: Ghi nhận thời gian xử lý, kết quả trả về</p> <p>Bước 4: Lặp lại các bước trên với lượng người dùng tăng dần</p>
Kết quả mong đợi	Hệ thống phản hồi dưới 3 giây, xác thực chính xác, không bị treo hoặc xảy ra quá tải.

Bảng 3.11. Kịch bản kiểm thử đăng nhập

3.1.4. Thực hiện và kết quả kiểm thử

- Kiểm thử trang chủ

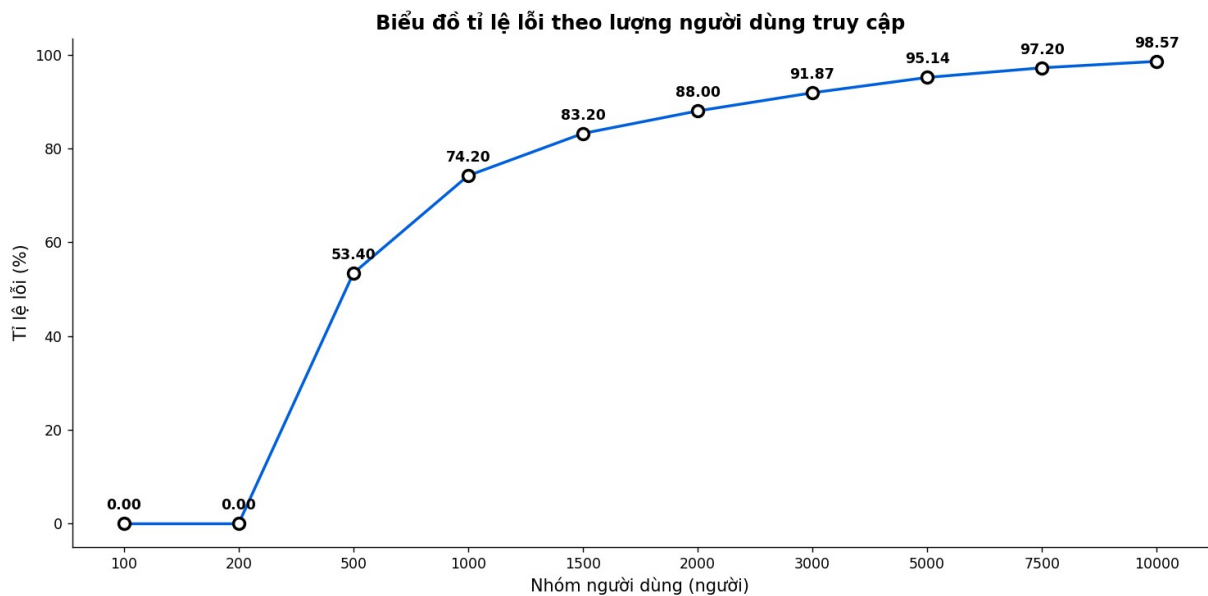
Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử trang chủ với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.2 với kết quả trả về được thể hiện như bảng bên dưới đây.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	3.60	0.00	Không xảy ra lỗi, phản hồi trong thời gian tốt	Đạt
200	9.51	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	6.79	53.40	Hệ thống quá tải nhiều thời điểm làm thời gian phản hồi trung bình giảm	Không đạt
1000	4.37	74.20	Tình trạng quá tải tăng	Không đạt
1500	2.83	83.20	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
2000	2.20	88.00	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.37	91.87	Hệ thống chịu tải lớn	Không đạt
5000	0.86	95.14	Hệ thống gần như quá tải	Không đạt
7500	0.62	97.20	Hệ thống quá tải	Không đạt
10000	0.45	98.57	Hệ thống quá tải	Không đạt

Bảng 3.12. Kết quả kiểm thử trang chủ

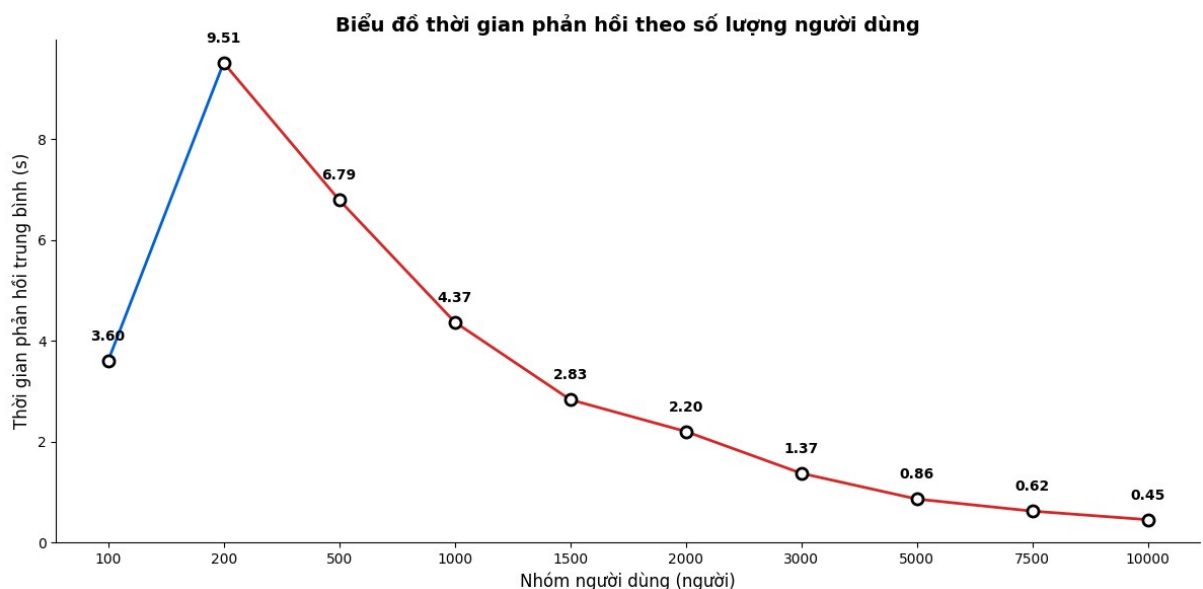
Từ kết quả kiểm thử, có thể thấy tỉ lệ lỗi tăng đột biến khi số lượng người dùng vượt mốc 500. Cụ thể, với 500 người dùng, hệ thống bắt đầu ghi nhận tỉ lệ lỗi

53.4%, và con số này tiếp tục tăng mạnh, đạt đến 98.57% ở mức 10.000 người.



Hình 3.4. Biểu đồ tỉ lệ lỗi kiểm thử trang chủ

Dựa vào Hình 3.5 có thể nhận thấy thời gian phản hồi trung bình tăng mỗi khi số lượng người dùng tăng, tuy nhiên giảm mạnh khi tỉ lệ phản hồi lỗi xuất hiện vì hệ thống phản hồi lỗi rất nhanh, cụ thể và bắt đầu ở 500 người dùng thì thời gian phản hồi bắt đầu giảm.



Hình 3.5. Biểu đồ thời gian phản hồi kiểm thử trang chủ

- Kiểm thử thực đơn

Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử trang thực đơn với các

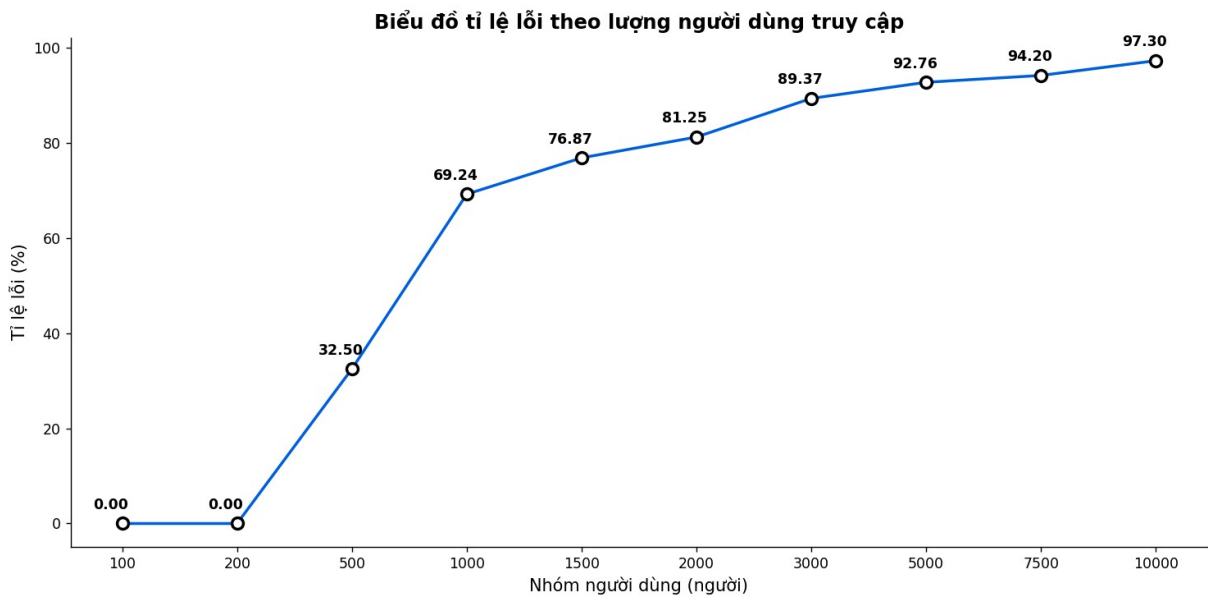
cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.3 với kết quả trả về được thể hiện như bảng bên dưới đây.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	1.31	0.00	Không xảy ra lỗi, phản hồi nhanh và ổn định	Đạt
200	7.23	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	6.54	32.50	Hệ thống quá tải nhiều thời điểm và không trả về kết quả mong muốn	Không đạt
1000	3.43	69.24	Tình trạng quá tải tăng mạnh, phản hồi lỗi nhiều	Không đạt
1500	2.51	76.87	Tỉ lệ lỗi cao, hệ thống chỉ còn phản hồi ít thời điểm	Không đạt
2000	1.92	81.25	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.30	89.37	Hệ thống quá tải	Không đạt
5000	0.97	92.76	Hệ thống quá tải	Không đạt
7500	0.63	94.20	Hệ thống quá tải	Không đạt
10000	0.43	97.30	Hệ thống quá tải	Không đạt

Bảng 3.13. Kết quả kiểm thử test case thực đơn

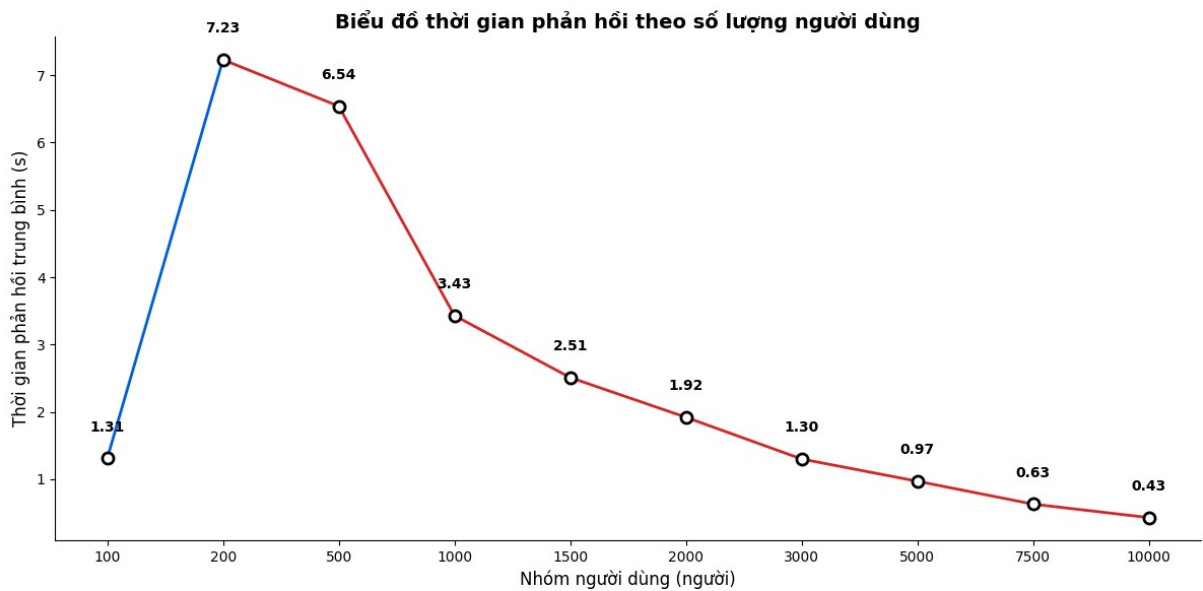
Qua kết quả kiểm thử hiệu suất, có thể thấy rằng hệ thống hoạt động ổn định và không xảy ra lỗi khi số lượng người dùng ở mức 100 đến 200. Tuy nhiên, bắt đầu từ mốc 500 người dùng, tỉ lệ lỗi đã tăng lên 32.5%, cho thấy hệ thống bắt đầu

xuất hiện hiện tượng quá tải.



Hình 3.6. Biểu đồ tỉ lệ lỗi kiểm thử thực đơn

Có thể nhận thấy thời gian phản hồi trung bình tăng khi số lượng người dùng gia tăng, nhưng giảm rõ rệt từ 500 người dùng trở đi do xử lý lỗi nhanh hơn, và ở 10000 người dùng, thời gian gần 0 cho thấy hệ thống gần như quá tải.



Hình 3.7. Biểu đồ thời gian phản hồi kiểm thử thực đơn

- Kiểm thử tìm kiếm món ăn

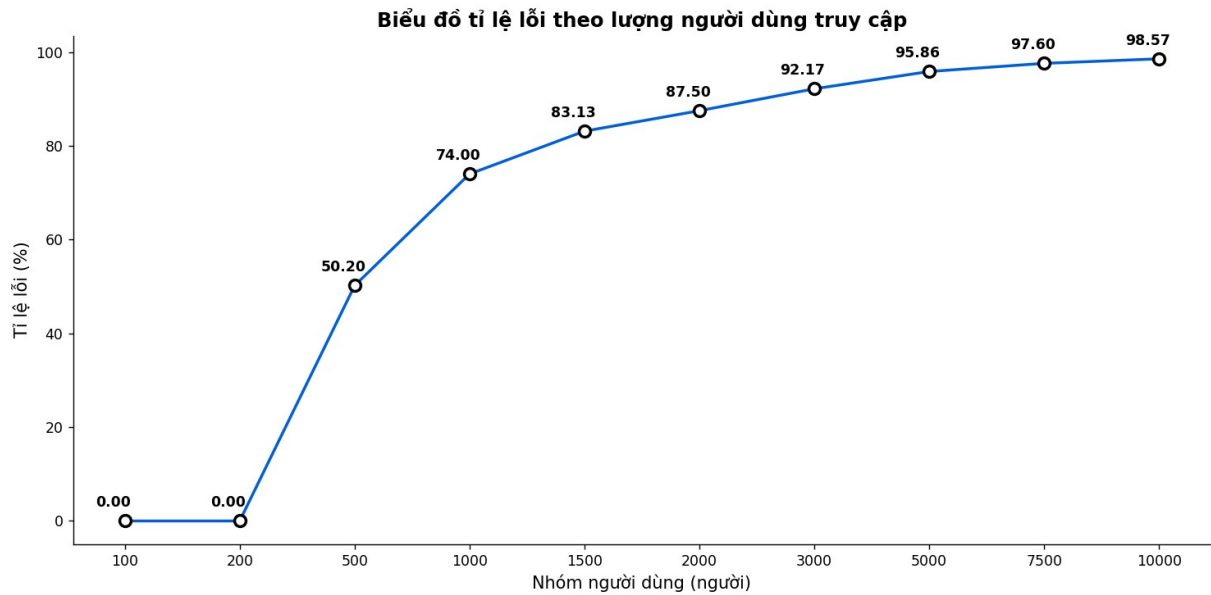
Cấu hình một kế hoạch kiểm thử để thực hiện kiểm thử tìm kiếm món ăn với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.4 với kết quả trả về được

thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	1.73	0.00	Không xảy ra lỗi, phản hồi nhanh và ổn định	Đạt
200	8.78	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	6.73	50.20	Hệ thống quá tải nhiều thời điểm và không trả về kết quả mong muốn	Không đạt
1000	3.90	74.00	Tình trạng quá tải tăng mạnh, phản hồi lỗi nhiều	Không đạt
1500	2.81	83.13	Tỉ lệ lỗi cao, hệ thống chỉ còn phản hồi ít thời điểm	Không đạt
2000	1.98	87.50	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.37	92.17	Hệ thống chịu tải rất lớn	Không đạt
5000	0.77	95.86	Hệ thống gần như quá tải	Không đạt
7500	0.62	97.60	Tỉ lệ lỗi gần tới ngưỡng	Không đạt
10000	0.49	98.57	Hệ thống quá tải	Không đạt

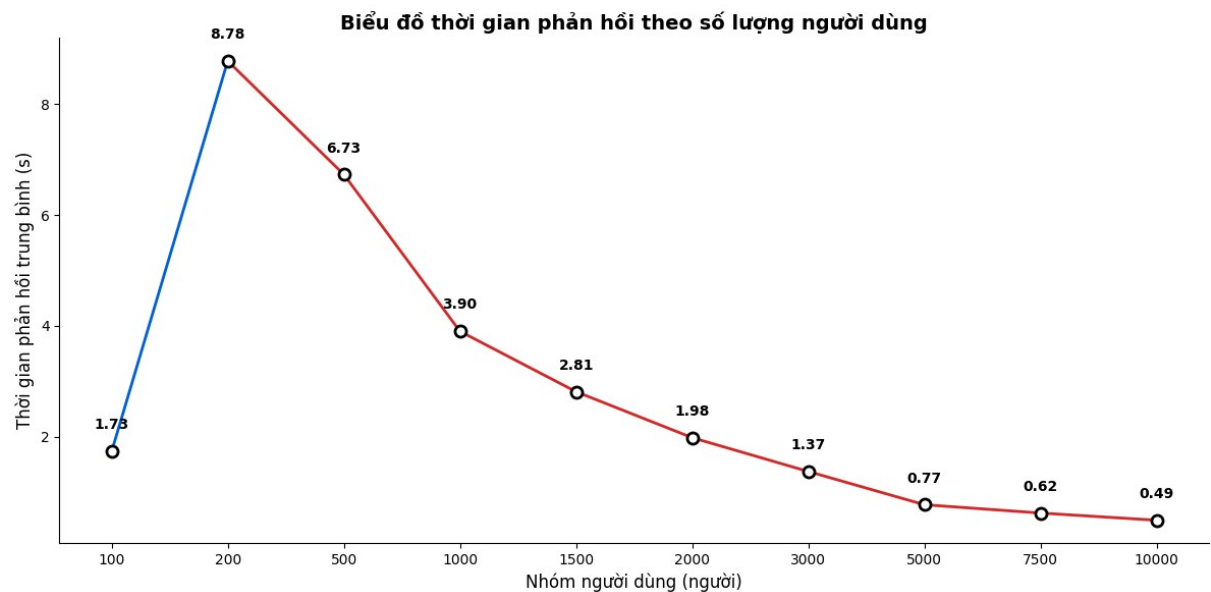
Bảng 3.14. Kết quả kiểm thử tìm kiếm món ăn

Từ dữ liệu kiểm thử hiệu suất, có thể thấy rằng hệ thống hoạt động ổn định ở ngưỡng 100 – 200 người dùng với tỉ lệ lỗi bằng 0%. Tuy nhiên, khi số lượng người dùng đạt 500 trở lên, tỉ lệ lỗi bắt đầu tăng mạnh. Điều này cho thấy hệ thống không thể xử lý hiệu quả khi bị tải cao, dẫn đến hiện tượng quá tải.



Hình 3.8. Biểu đồ tỉ lệ lỗi kiểm thử tìm kiếm món ăn

Biểu đồ sau cho thấy thời gian phản hồi trung bình tăng khi số lượng người dùng tăng, nhưng giảm đáng kể từ 500 người dùng trở đi do tốc độ xử lý lỗi nhanh hơn và đến 10000 người dùng thời gian phản hồi gần 0, hệ thống gần như quá tải.



Hình 3.9. Biểu đồ thời gian phản hồi kiểm thử tìm kiếm món ăn

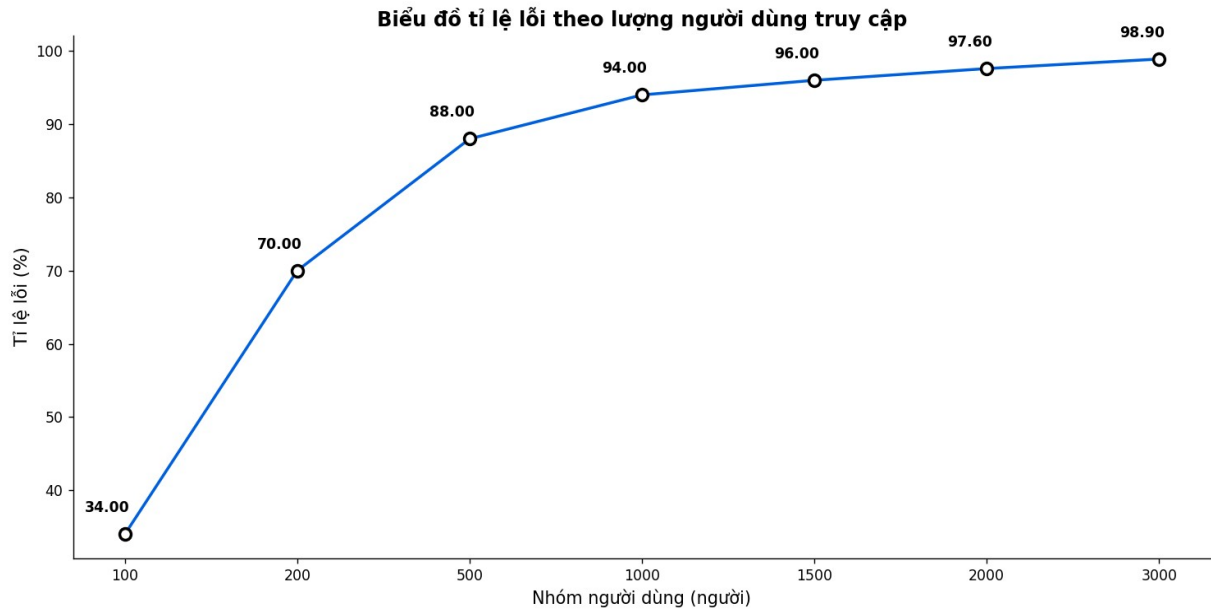
- Kiểm thử gợi ý món ăn liên quan

Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử gợi ý món ăn liên quan khi vào xem một món ăn với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.5 với kết quả trả về được biểu diễn bởi Bảng 3.15 dưới đây.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	11.74	34.00	Hệ thống xảy ra lỗi tại nhiều thời điểm, thời gian phản hồi lâu	Không đạt
200	19.00	70.00	Hệ thống quá tải nhiều thời điểm, tỉ lệ phản hồi lỗi cao	Không đạt
500	9.46	88.00	Hệ thống quá tải nhiều thời điểm và không trả về kết quả mong muốn	Không đạt
1000	4.76	94.00	Tình trạng quá tải tăng mạnh, tỉ lệ phản hồi báo lỗi cao	Không đạt
1500	3.20	96.00	Hệ thống gần như đạt giới hạn chịu tải	Không đạt
2000	2.42	97.60	Tỉ lệ lỗi vẫn có xu hướng tăng, thời gian phản hồi giảm mạnh	Không đạt
3000	1.95	98.90	Hệ thống quá tải	Không đạt

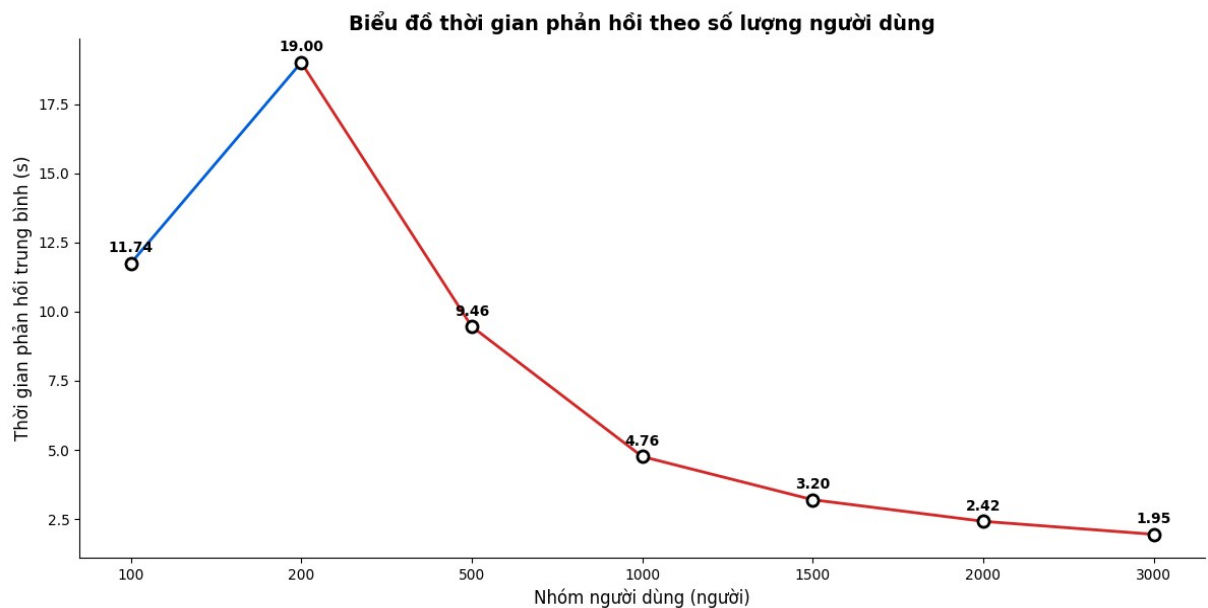
Bảng 3.15. Kết quả kiểm thử gợi ý món ăn liên quan

Từ 100 người dùng trở lên, hệ thống đã bắt đầu xuất hiện 34% lỗi và tình trạng này tăng rất nhanh, cho thấy hệ thống rơi vào trạng thái quá tải ở hầu hết các thời điểm. Từ 1500 người trở đi, tỉ lệ lỗi gần như đạt mức tối đa, cho thấy hệ thống đạt điểm quá tải nhanh.



Hình 3.10. Biểu đồ tỉ lệ lỗi kiểm thử gợi ý món ăn

Nhìn vào dữ liệu, thời gian phản hồi trung bình tăng khi số lượng người dùng tăng, nhưng giảm rõ rệt từ 500 người dùng trở đi do xử lý phản hồi nhanh hơn, và đến 3000 người dùng, thời gian giảm còn 1.95, cho thấy hệ thống chịu tải lớn.



Hình 3.11. Biểu đồ thời gian phản hồi kiểm thử gợi ý món ăn

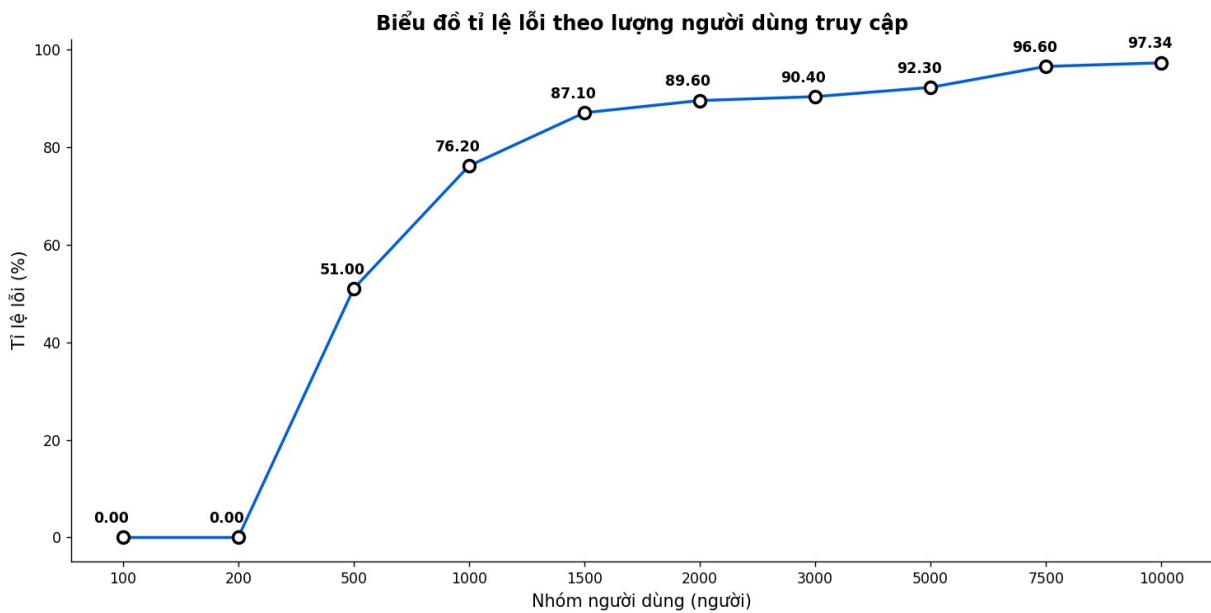
- Kiểm thử tìm kiếm bài viết

Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử tìm kiếm bài viết với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.6 với kết quả trả về được thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	2.08	0.00	Không xảy ra lỗi, phản hồi nhanh và ổn định	Đạt
200	8.63	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	6.73	51.00	Hệ thống quá tải nhiều thời điểm và không trả về kết quả mong muốn	Không đạt
1000	4.05	76.20	Tình trạng quá tải tăng mạnh, tỉ lệ lỗi tăng cao	Không đạt
1500	3.02	87.10	Tỉ lệ lỗi cao, hệ thống chỉ còn phản hồi ít thời điểm	Không đạt
2000	1.56	89.60	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.32	90.40	Hệ thống chịu tải lớn	Không đạt
5000	0.52	92.30	Hệ thống gần như quá tải	Không đạt
7500	0.42	96.60	Hệ thống quá tải	Không đạt
10000	0.39	97.34	Hệ thống quá tải	Không đạt

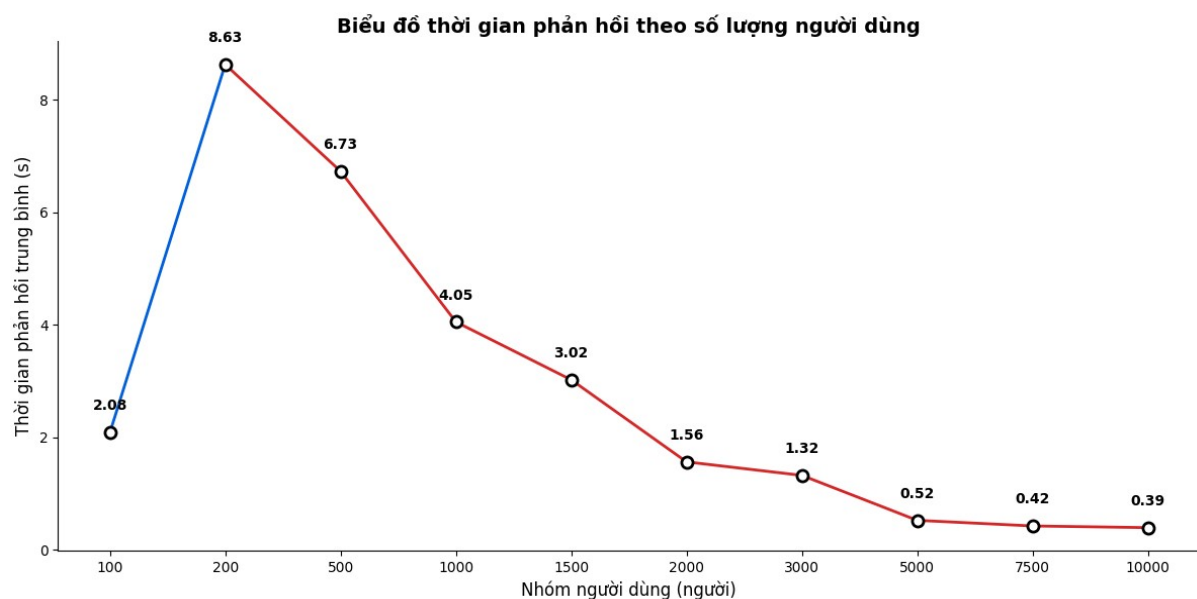
Bảng 3.16. Kết quả kiểm thử tìm kiếm bài viết

Hệ thống hoạt động ổn định với 100 và 200 người dùng khi không ghi nhận lỗi. Từ 500 người dùng trở lên, tỉ lệ lỗi bắt đầu tăng mạnh, đạt 51% tại 500 người. Khi số lượng người dùng tiếp tục tăng, tỉ lệ lỗi cũng tăng tương ứng, cho thấy hệ thống không hoạt động ổn định nào lượng người dùng tăng mạnh.



Hình 3.12. Biểu đồ tỉ lệ lỗi kiểm thử tìm kiếm bài viết

Dữ liệu cho thấy thời gian phản hồi trung bình tăng khi số lượng người dùng tăng, nhưng giảm mạnh từ 500 người dùng trở đi do phản hồi được xử lý nhanh hơn, và đến 10000 người dùng, thời gian phản hồi càng về 0.



Hình 3.13. Biểu đồ thời gian phản hồi kiểm thử tìm kiếm bài viết

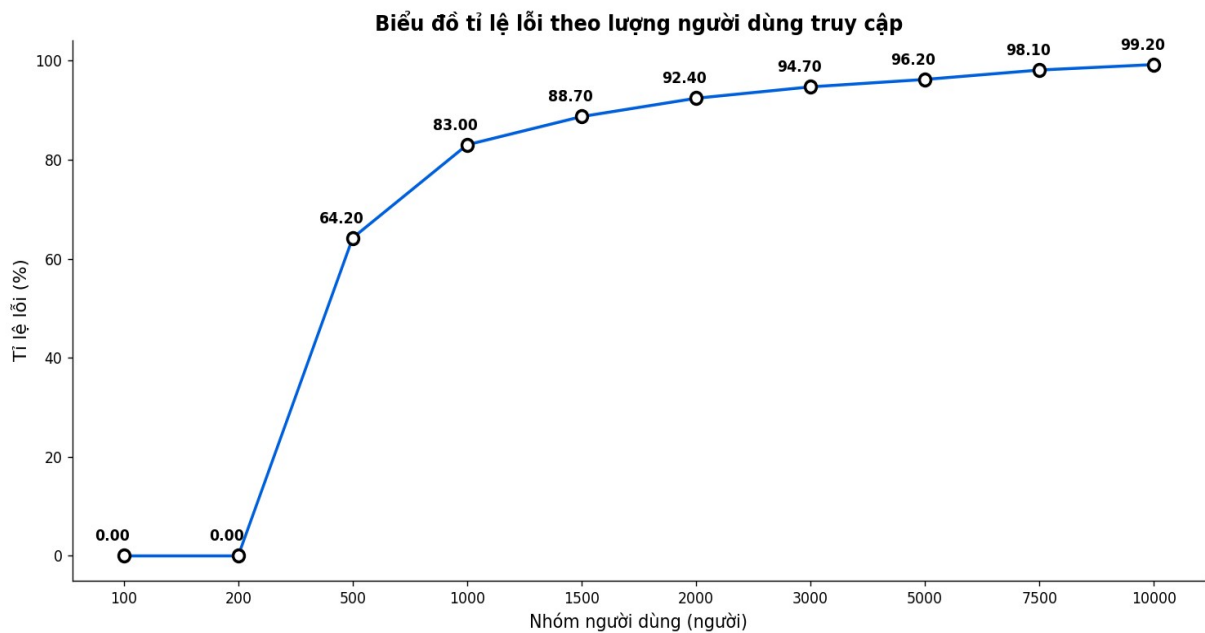
- Kiểm thử giỏ hàng đặt bàn

Cấu hình một kế hoạch kiểm thử để thực hiện kiểm thử giỏ hàng tại những giờ cao điểm nhiều khách hàng với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.7 với kết quả trả về được thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	5.70	0.00	Không xảy ra lỗi, tốc độ phản hồi ổn định	Đạt
200	15.41	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	5.60	64.20	Hệ thống quá tải nhiều thời điểm, tỉ lệ lỗi cao	Không đạt
1000	2.90	83.00	Tình trạng quá tải tăng mạnh, tỉ lệ phản hồi báo lỗi cao	Không đạt
1500	1.89	88.70	Tỉ lệ lỗi cao, hệ thống chỉ còn phản hồi ít thời điểm	Không đạt
2000	1.32	92.40	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.18	94.70	Hệ thống chịu tải lớn	Không đạt
5000	0.65	96.20	Hệ thống gần như quá tải	Không đạt
7500	0.25	98.10	Hệ thống quá tải	Không đạt
10000	0.39	99.20	Hệ thống quá tải	Không đạt

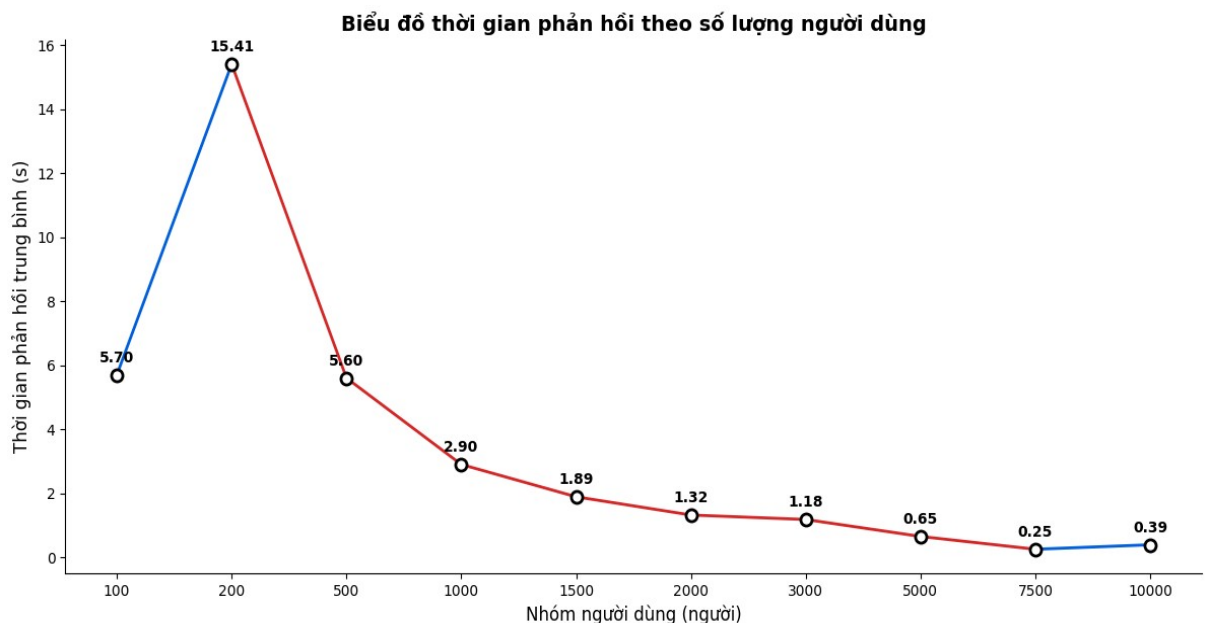
Bảng 3.17. Kết quả kiểm thử giỏ hàng đặt bàn

Hệ thống vận hành ổn định với 100 và 200 người dùng khi không xuất hiện lỗi. Tuy nhiên, từ 500 người dùng trở đi, hệ thống bắt đầu ghi nhận lỗi với tỉ lệ lên đến 64.%. Khi số lượng người dùng tiếp tục tăng, tỉ lệ lỗi cũng tăng dần đều.



Hình 3.14. Biểu đồ tỉ lệ lỗi kiểm thử giỏ hàng đặt bàn

Có thể thấy thời gian phản hồi trung bình tăng khi số lượng người dùng tăng, nhưng giảm mạnh từ 500 người dùng trở đi do xử lý nhanh hơn, và đến 10,000 người dùng, thời gian gần 0, cho thấy hệ thống gần như không thể xử lý thêm.



Hình 3.15. Biểu đồ thời gian phản hồi kiểm thử giỏ hàng đặt bàn

- Kiểm thử thanh toán

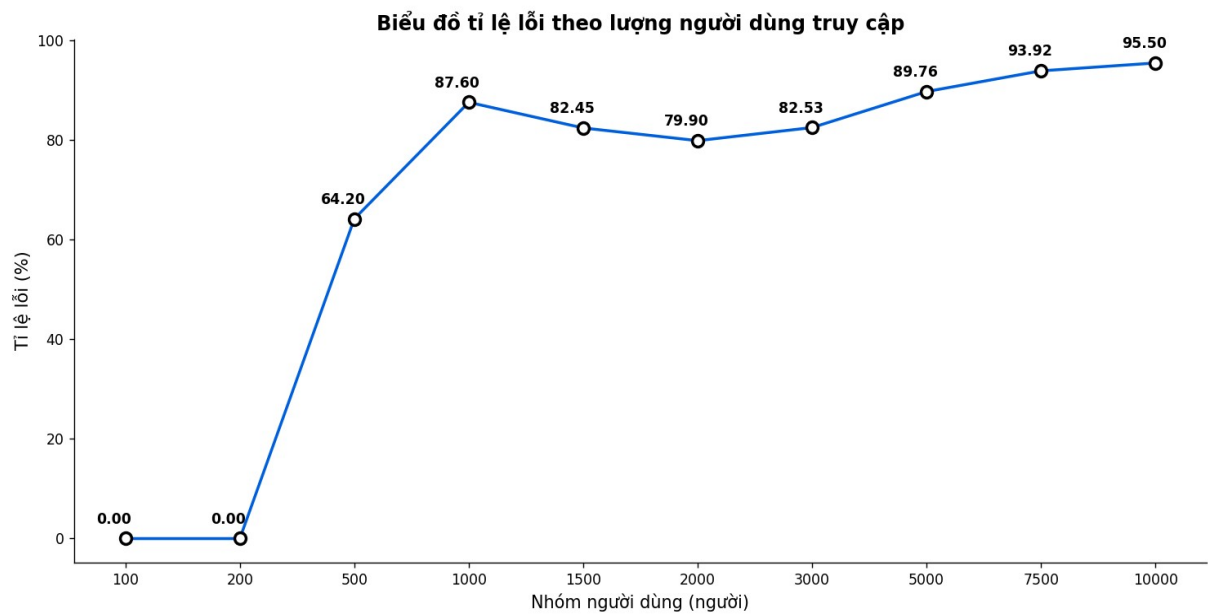
Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử khả năng tải trang thanh toán trực tuyến với các cài đặt và dữ liệu đầu vào được thể hiện chi tiết như

Bảng 3.8 với kết quả kiểm thử được thể hiện bởi bảng đánh giá dưới đây.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỷ lệ lỗi (%)	Nhận xét	Đánh giá
100	0.24	0.00	Không xảy ra lỗi, tốc độ phản hồi nhanh	Đạt
200	0.57	0.00	Không xảy ra lỗi, phản hồi nhanh	Đạt
500	0.94	64.20	Hệ thống xảy ra quá tải, tỷ lệ lỗi cao	Không đạt
1000	2.42	87.60	Tình trạng quá tải tăng mạnh, phản hồi nhiều lỗi	Không đạt
1500	1.89	82.45	Tỷ lệ lỗi cao, hệ thống chỉ còn phản hồi ít thời điểm	Không đạt
2000	1.32	79.90	Tỷ lệ lỗi giảm nhẹ nhưng vẫn quá cao	Không đạt
3000	1.18	82.53	Tỷ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
5000	0.65	89.76	Hệ thống hầu như quá tải	Không đạt
7500	0.25	93.92	Tình trạng lỗi vẫn tăng	Không đạt
10000	0.39	95.50	Hệ thống quá tải	Không đạt

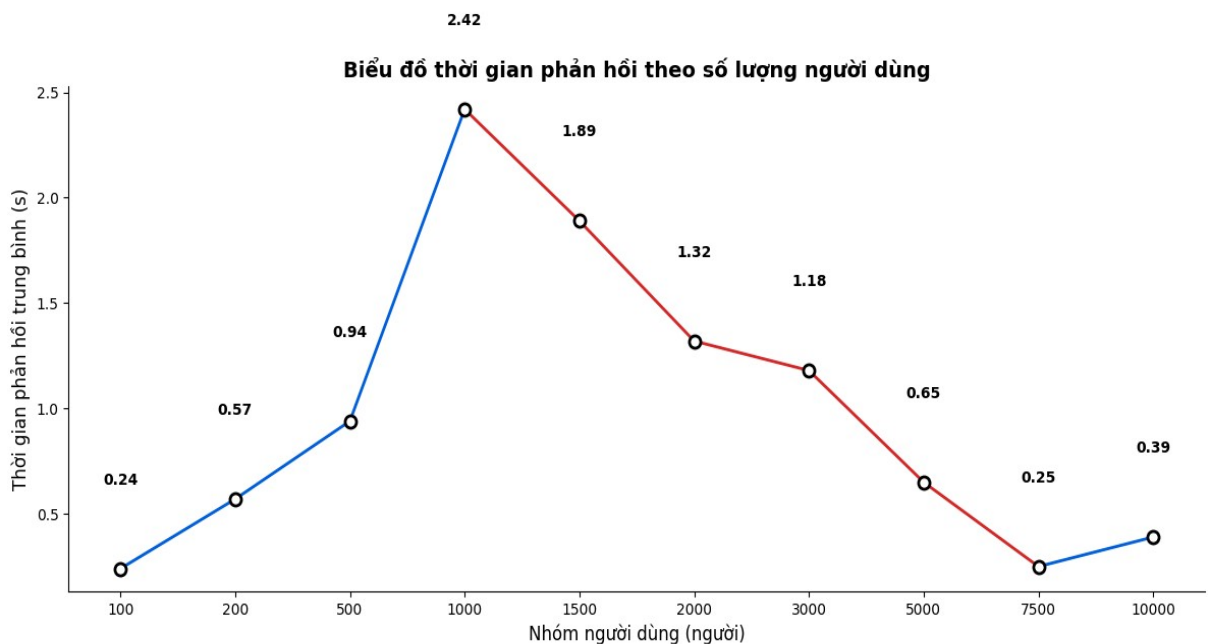
Bảng 3.18. Kết quả kiểm thử thanh toán

Hệ thống hoạt động ổn định với số lượng người dùng từ 100 đến 200, khi tỷ lệ lỗi là 0%. Tuy nhiên, từ mốc 500 người dùng trở đi, tỷ lệ lỗi bắt đầu tăng mạnh và không có dấu hiệu giảm xuống, cho thấy hệ thống chịu tải rất kém.



Hình 3.16. Biểu đồ tỉ lệ lỗi kiểm thử thanh toán

Nhìn vào biểu đồ dưới đây có thể nhận thấy thời gian phản hồi trung bình tăng từ 100 đến 1000 người dùng, rồi giảm dần từ 1500 người dùng và chủ yếu ở mức rất thấp từ 5000 người dùng trở đi.



Hình 3.17. Biểu đồ thời gian phản hồi kiểm thử thanh toán

- Kiểm thử đặt bàn

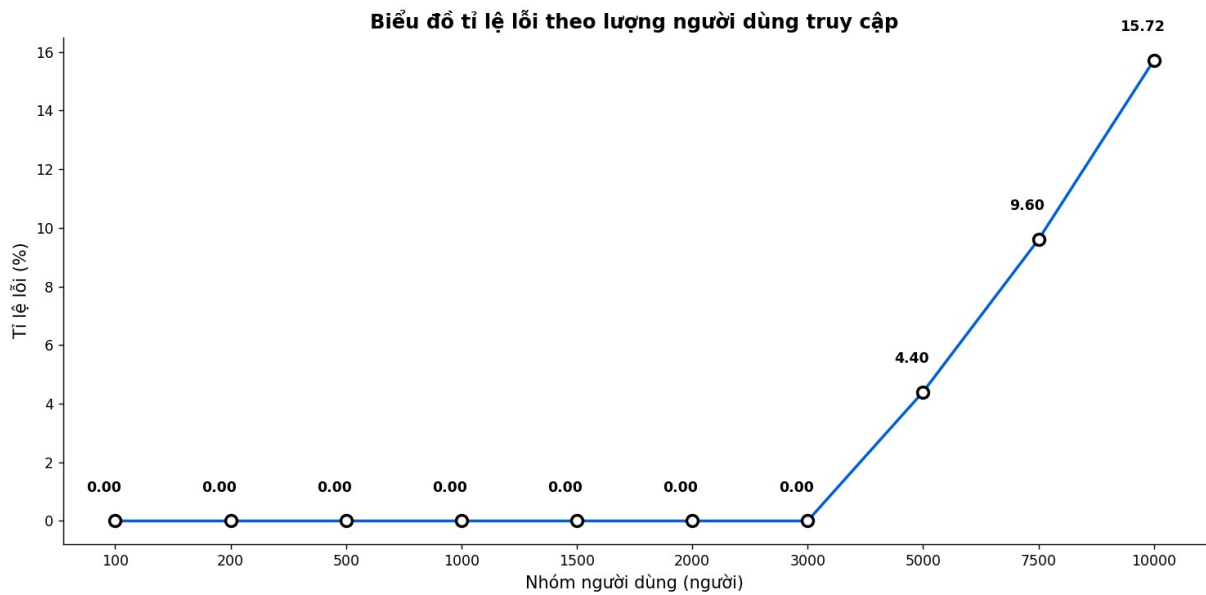
Cấu hình một kế hoạch kiểm thử để thực hiện kiểm thử hiệu suất đặt bàn trực tuyến với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.9 với kết quả trả

về được thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	0.02	0	Không có lỗi xảy ra	Đạt
200	0.04	0	Không xảy ra lỗi, phản hồi nhanh	Đạt
500	0.08	0	Tốc độ phản hồi ổn định	Đạt
1000	1.34	0	Không xảy ra lỗi, phản hồi nhanh	Đạt
1500	1.82	0	Tốc độ phản hồi ổn định	Đạt
2000	2.54	0	Không có lỗi xuất hiện, hệ thống không chịu tải	Đạt
3000	4.76	0	Hệ thống hoạt động ổn định, không có lỗi	Đạt
5000	10.43	4.40	Hệ thống xuất hiện lỗi, có tình trạng chịu tải	Không đạt
7500	17.20	9.60	Hệ thống chịu tải một số thời điểm, phản hồi chậm	Không đạt
10000	18.08	15.72	Hệ thống xảy ra quá tải một số thời điểm	Không đạt

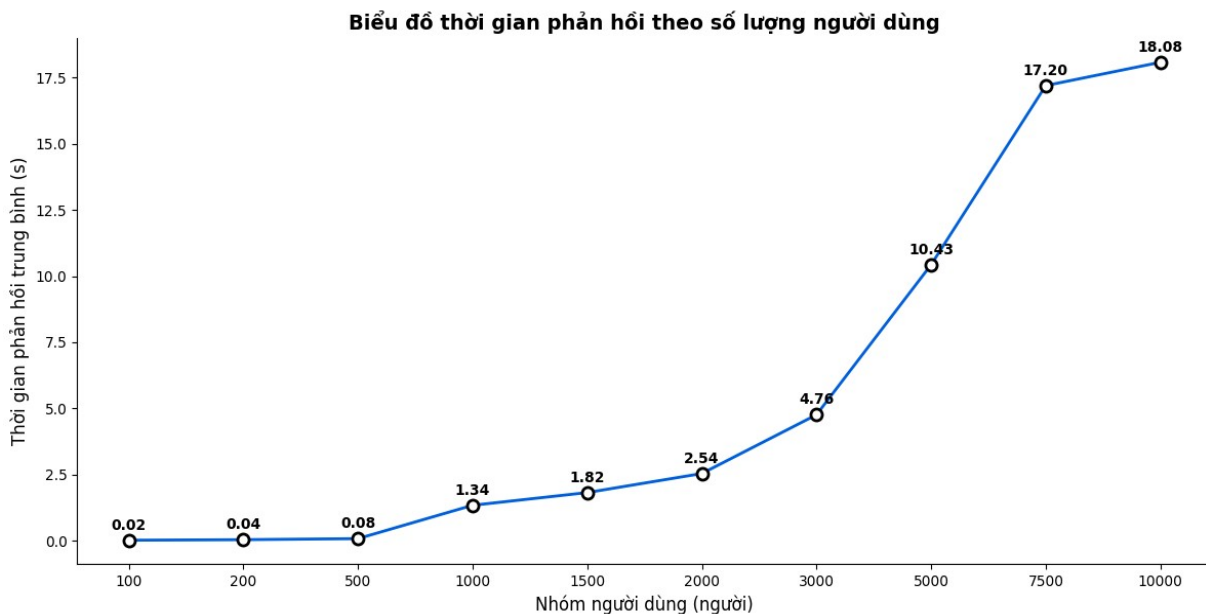
Bảng 3.19. Kết quả kiểm thử đặt bàn

Hệ thống hoạt động rất ổn định từ 100 đến 3000 người dùng với tỷ lệ lỗi là 0%, cho thấy khả năng xử lý tốt trong khoảng tải vừa phải. Tuy nhiên, khi số lượng người dùng tăng lên trên 5000, hệ thống bắt đầu ghi nhận lỗi và tăng dần với theo lượng người dùng truy cập.



Hình 3.18. Biểu đồ tỉ lệ lỗi kiểm thử đặt bàn

Dữ liệu cho thấy thời gian phản hồi trung bình tăng chậm từ 100 đến 500 người dùng, sau đó tăng mạnh từ 1000 người dùng và đạt đỉnh ở 10,000 người dùng với 18.08, ám chỉ hệ thống đang chịu áp lực nặng nề ở mức cao.



Hình 3.19. Biểu đồ thời gian phản hồi kiểm thử đặt bàn

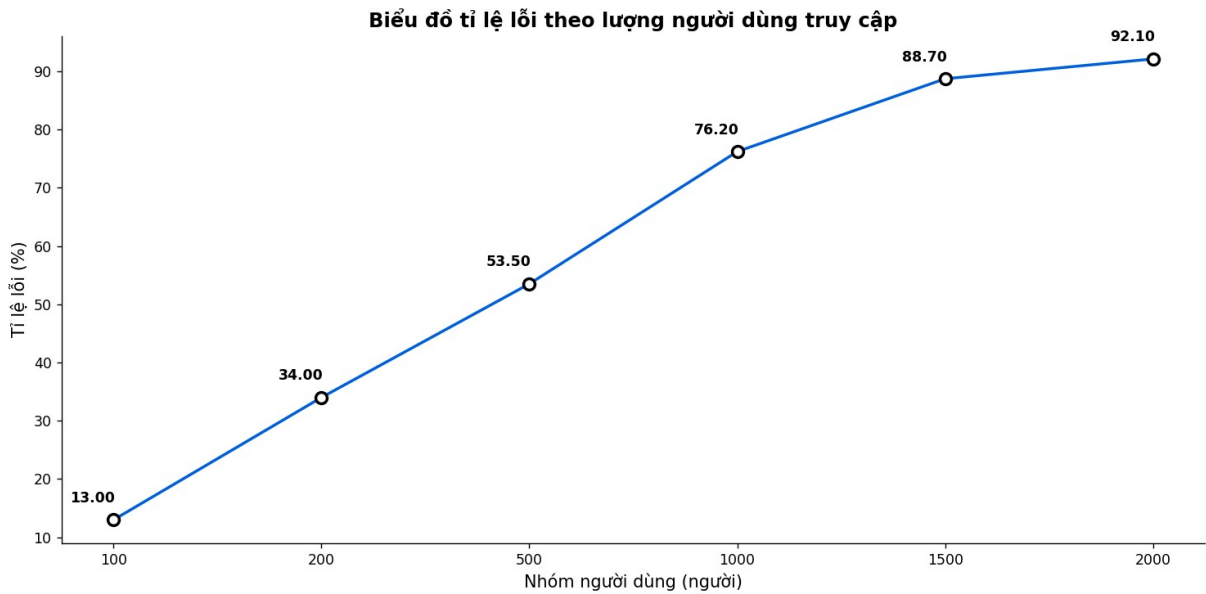
- Kiểm thử chatbot

Cấu hình một kế hoạch kiểm thử để thực hiện kiểm thử khả năng gửi phản hồi chatbot AI cho câu hỏi của khách hàng, với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.10 và kết quả trả về được thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	89.40	13.00	Không xử lý được nhiều người dùng một lúc và xảy ra lỗi, hệ thống đa phần treo không phản hồi	Không đạt
200	153.40	34.00	Thời gian phản hồi rất lâu, tỉ lệ lỗi tăng mạnh, hệ thống xử lý đa nhiệm kém	Không đạt
500	302.50	53.50	Hệ thống không xử lý được nhiều người dùng một lúc, tỉ lệ lỗi cao chưa áp dụng vào thực tế	Không đạt
1000	687.00	76.20	Hệ thống đa phần không phản hồi, khả năng chịu tải kém	Không đạt
1500	1276.00	88.70	Hệ thống không chịu được tải lớn	Không đạt
2000	2487.00	92.10	Hệ thống quá tải	Không đạt

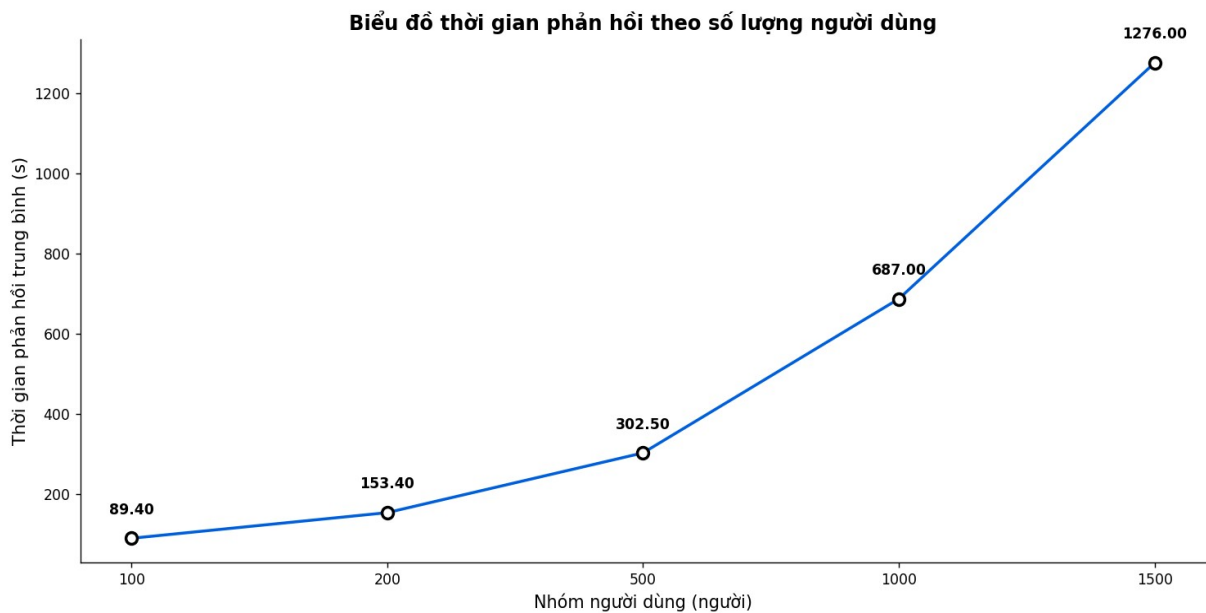
Bảng 3.20. Kết quả kiểm thử chatbot

Dữ liệu cho thấy khi số lượng người dùng tăng lên, hệ thống bắt đầu xuất hiện lỗi với tỉ lệ ngày càng cao. Ở mức 100 người dùng, hệ thống chỉ gặp 13% và cứ thế tăng mạnh ở lượng người dùng cao hơn. Điều này chứng tỏ hệ thống chưa được tối ưu hóa để xử lý lượng truy cập lớn và có nguy cơ sập khi quá tải.



Hình 3.20. Biểu đồ tỉ lệ lỗi kiểm thử chatbot

Quan sát dữ liệu từ biểu đồ dưới đây, có thể thấy thời gian phản hồi trung bình tăng liên tục từ 100 đến 2000 người dùng, đặc biệt tăng rất mạnh từ 1000 người dùng trở lên, đạt 2487 giây ở 2000 người dùng.



Hình 3.21. Biểu đồ thời gian phản hồi kiểm thử chatbot

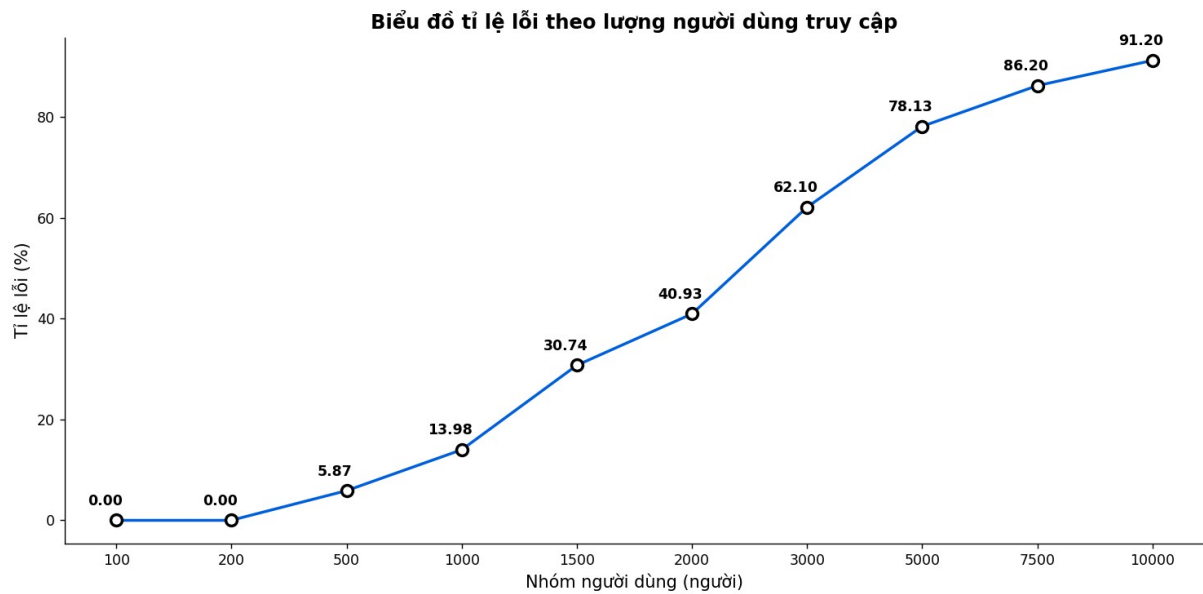
- Kiểm thử đăng nhập

Cấu hình một kế hoạch kiểm thử để thực hiện kiểm thử khả năng đăng nhập hệ thống tại giờ cao điểm với các cài đặt và dữ liệu đầu vào được thể hiện như Bảng 3.11 với kết quả trả về được thể hiện như bảng bên dưới.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	4.54	0.00	Không xảy ra lỗi, tốc độ phản hồi ổn định	Đạt
200	8.43	0.00	Không xảy ra lỗi, nhưng phản hồi chậm	Không đạt
500	17.88	5.87	Bắt đầu xuất hiện tình trạng quá tải, dẫn đến phản hồi lỗi	Không đạt
1000	20.43	13.98	Phản hồi lỗi có xu hướng tăng mạnh	Không đạt
1500	18.85	30.74	Tình trạng lỗi tăng mạnh	Không đạt
2000	15.85	40.93	Hệ thống chịu tải lớn	Không đạt
3000	10.53	62.10	Phản hồi lỗi vẫn tiếp tục tăng mạnh	Không đạt
5000	6.98	78.13	Hệ thống gần mức quá tải	Không đạt
7500	3.12	86.20	Hệ thống chịu tải đa số thời điểm, phản hồi chậm	Không đạt
10000	1.76	91.20	Hệ thống quá tải	Không đạt

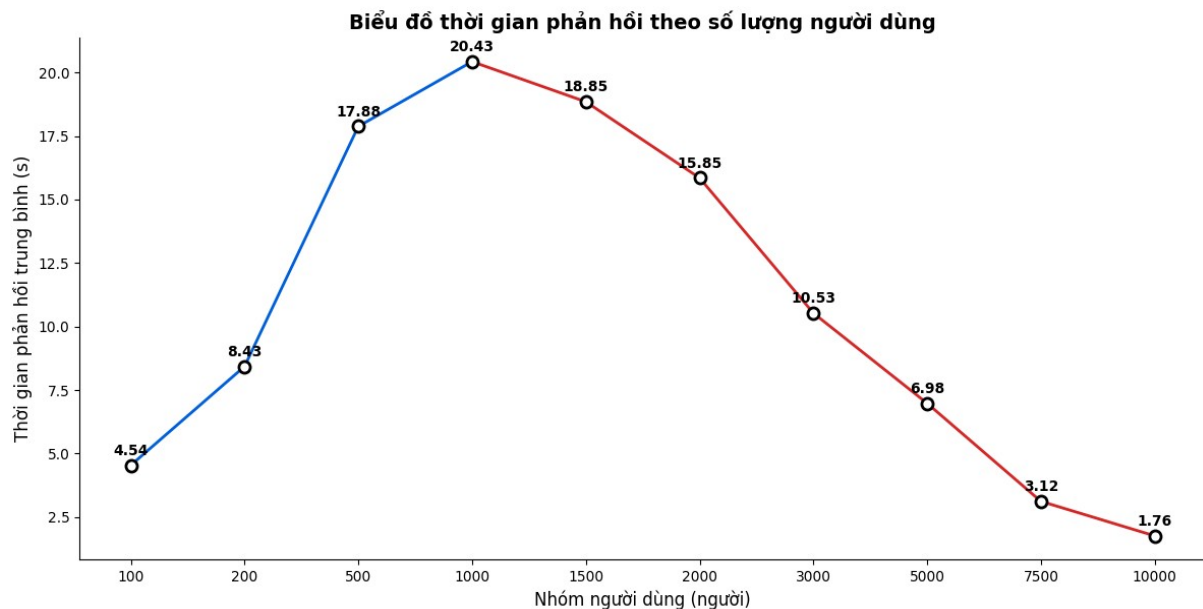
Bảng 3.21. Kết quả kiểm thử đăng nhập

Qua dữ liệu trên, có thể thấy rằng hệ thống hoạt động ổn định với 200 người dùng đầu tiên khi không có lỗi xảy ra. Tuy nhiên, từ mốc 500 người dùng trở đi, tỉ lệ lỗi bắt đầu xuất hiện và tăng nhanh theo cấp số nhân.



Hình 3.22. Biểu đồ tỉ lệ lỗi kiểm thử đăng nhập

Nhìn vào dữ liệu, thời gian phản hồi trung bình tăng từ 100 đến 1000 người dùng, đạt đỉnh 20.43 giây, sau đó giảm dần từ 1500 người dùng và xuống còn 1.76 giây ở 10,000 người dùng vì tỉ lệ lỗi tăng mạnh làm giảm thời gian phản hồi trung bình xuống.

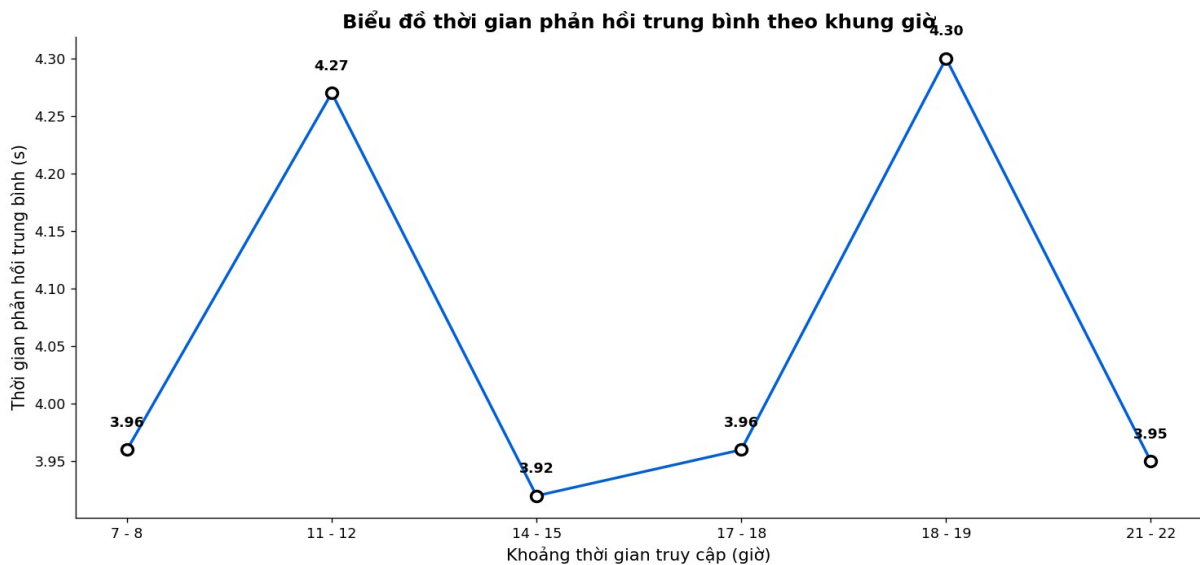


Hình 3.23. Biểu đồ thời gian phản hồi kiểm thử đăng nhập

- Kiểm thử tìm kiếm món ăn theo khung giờ

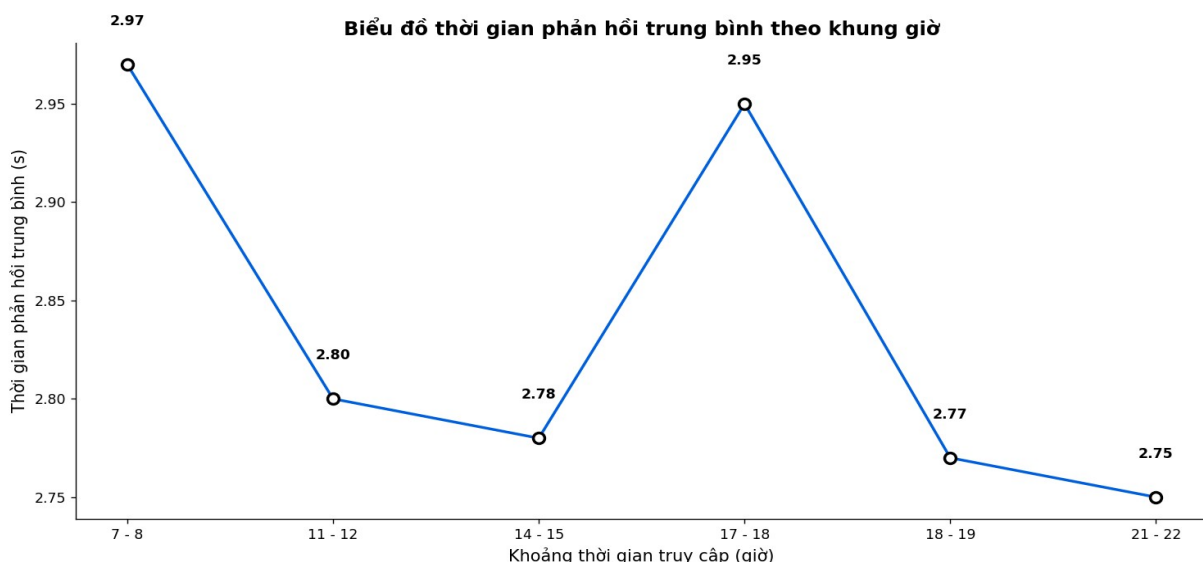
Tiến hành kiểm thử tìm kiếm món ăn theo khung giờ với mức người dùng truy cập là 1000 người, nhận thấy vào khoảng thời gian cao điểm như 11 giờ - 12

giờ và 18 giờ đến 19 giờ tăng nhẹ so với các khung giờ khác.



Hình 3.24. Biểu đồ thời gian phản hồi theo khung giờ kiểm thử tìm kiếm món ăn
- Kiểm thử giỏ hàng theo khung giờ

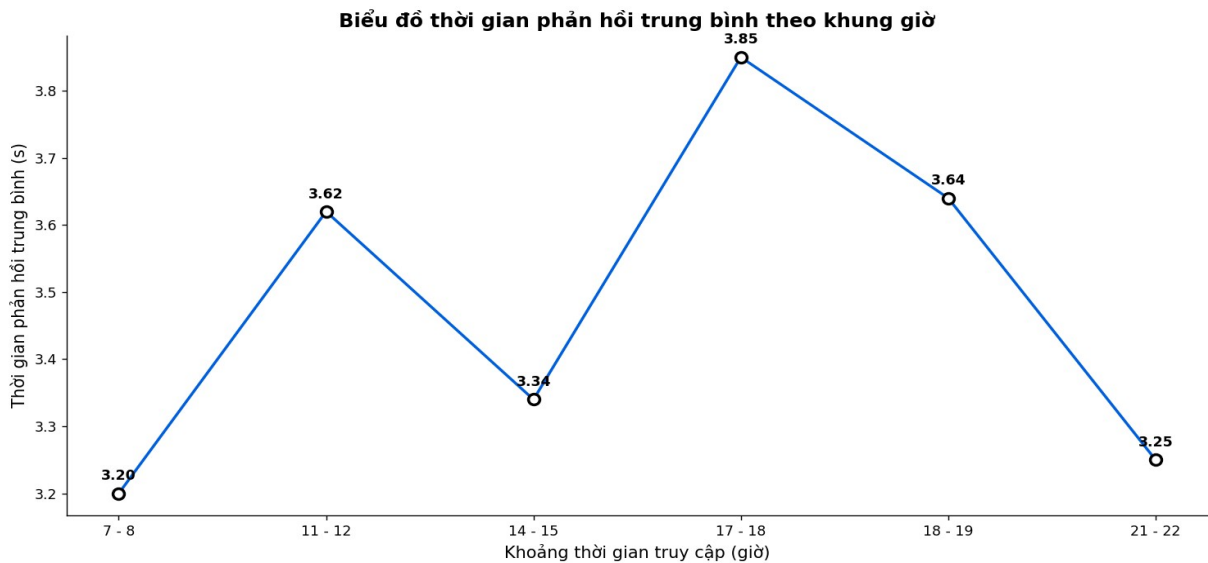
Tiến hành kiểm thử giỏ hàng theo khung giờ với cấu hình như kiểm thử tìm kiếm món ăn, nhận thấy vào khoảng thời gian cao điểm như 11 giờ - 12 giờ và 17 giờ đến 18 giờ tăng nhẹ so với các khung giờ khác.



Hình 3.25. Biểu đồ thời gian phản hồi theo khung giờ kiểm thử giỏ hàng đặt bàn
- Kiểm thử đặt bàn theo khung giờ

Tiến hành kiểm thử đặt bàn theo khung giờ với mức người dùng truy cập là 1000 người, nhận thấy vào khoảng thời gian cao điểm 17 giờ đến 18 giờ thì hệ

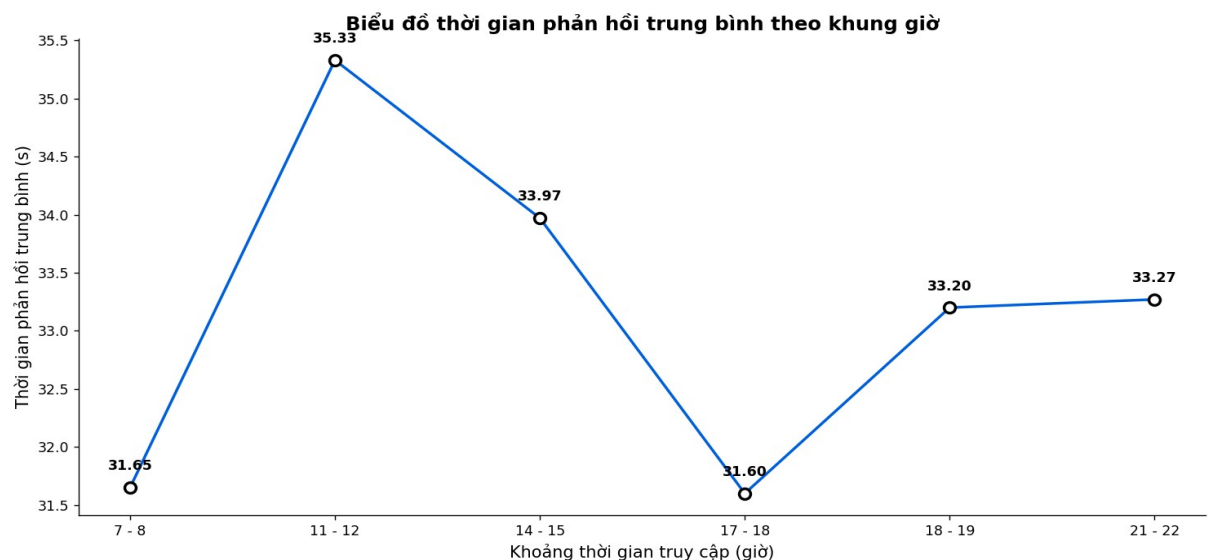
thống phản hồi chậm hơn nhiều các khung giờ khác.



Hình 3.26. Biểu đồ thời gian phản hồi theo khung giờ kiểm thử đặt bàn

- Kiểm thử chatbot theo khung giờ

Tiến hành kiểm thử chatbot theo khung giờ với mức người dùng truy cập là 1000 người, nhận thấy vào khoảng thời gian cao điểm 11 giờ đến 12 giờ thì hệ thống phản hồi chậm hơn nhiều các khung giờ khác.



Hình 3.26. Biểu đồ thời gian phản hồi theo khung giờ kiểm thử chatbot

3.1.5. Nhận xét và giải pháp xử lý

Kết quả kiểm thử hiệu suất cho thấy hệ thống hoạt động ổn định ở mức tải thấp, cụ thể trong khoảng 100–200 người dùng, với thời gian phản hồi trung bình

dao động từ 1–8 giây và tỉ lệ lỗi bằng 0%, đảm bảo hiệu năng như kỳ vọng. Tuy nhiên, khi lượng người dùng tăng từ 500 trở lên, hệ thống bắt đầu quá tải nghiêm trọng: thời gian phản hồi tăng mạnh, có lúc lên tới hơn 600 giây, đồng thời tỉ lệ lỗi cũng tăng vọt, nhiều trường hợp vượt 50–90%.

Ở một số lần kiểm thử, khi số lượng người dùng lên đến 1000–2000, thời gian phản hồi lại giảm đột ngột xuống dưới 2 giây, tuy nhiên đây không phải là dấu hiệu hệ thống xử lý tốt mà thực chất là do hệ thống trả lỗi nhanh thay vì xử lý thành công các yêu cầu, gây hiểu nhầm về hiệu năng thực tế.

Như vậy, tỉ lệ lỗi tăng cao chính là nguyên nhân khiến thời gian phản hồi giảm giả tạo ở các mức tải lớn. Những biểu hiện này cho thấy hệ thống hiện tại chưa có khả năng mở rộng tốt, dễ dàng rơi vào tình trạng nghẽn cổ chai, lỗi hàng loạt, đặc biệt khi vượt quá 1000 người dùng đồng thời.

Bên cạnh đó, khi kiểm thử theo khung giờ trong ngày, hệ thống ghi nhận thời gian phản hồi cao nhất vào các khung giờ cao điểm như 11–12h và 18–19h, thời điểm người dùng hoạt động nhiều. Ngược lại, các khung giờ như 7–8h và 14–15h có thời gian phản hồi thấp hơn, cho thấy sự phụ thuộc rõ rệt vào lưu lượng thực tế theo thời gian.

Giải pháp khuyến nghị bao gồm: tối ưu lại hệ thống xử lý và truy vấn cơ sở dữ liệu, triển khai cơ chế cân bằng tải, sử dụng bộ nhớ đệm để giảm tải cho server, đồng thời nâng cấp hạ tầng phần cứng và ứng dụng kiến trúc mở rộng ngang để tăng khả năng chịu tải.

Việc thiết lập hệ thống cảnh báo sớm, giám sát hiệu suất theo thời gian thực và thử nghiệm định kỳ cũng là bước quan trọng để đảm bảo tính ổn định lâu dài của hệ thống.

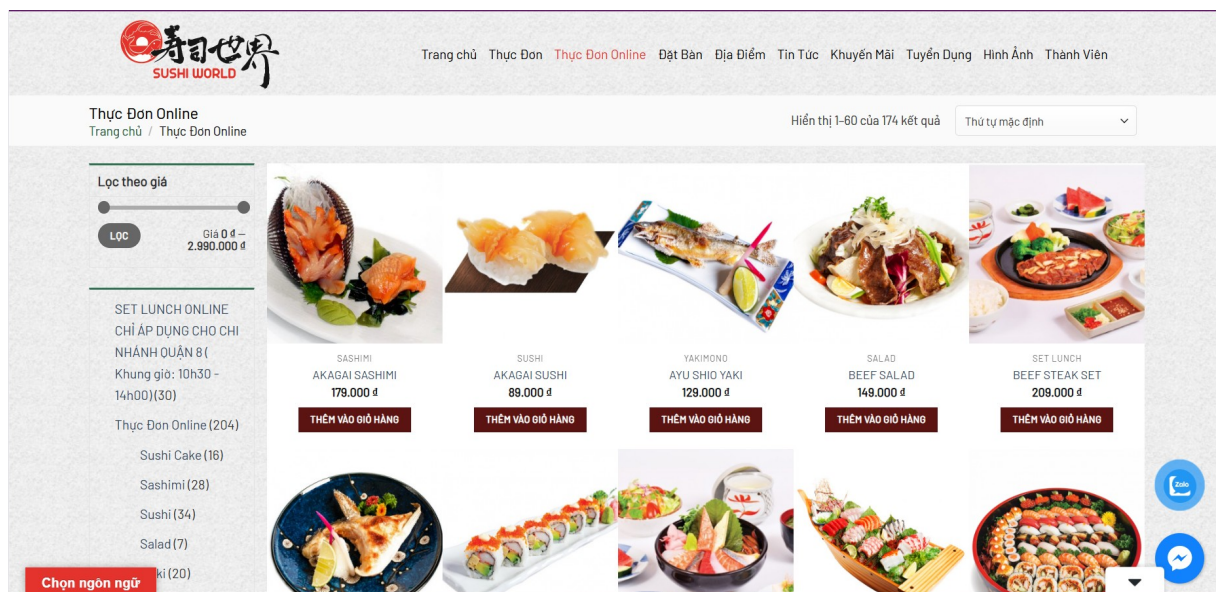
3.2. Kiểm thử hiệu năng website nhà hàng Sushi World

3.2.1. Giới thiệu website nhà hàng Sushi World

Sushi World là chuỗi nhà hàng Nhật Bản hàng đầu tại Việt Nam, nổi bật với

thực đơn phong phú và không gian đậm chất Nhật. Với sứ mệnh mang tinh hoa ẩm thực xứ Phù Tang đến gần hơn với thực khách Việt, Sushi World không ngừng nâng cao chất lượng dịch vụ và món ăn. Nhà hàng chinh phục thực khách bằng thực đơn phong phú với hơn 50 loại sushi, 18 loại sashimi tươi sống như cá hồi, cá ngừ, sò đỏ, bạch tuộc, cầu gai Nhật... . Đặc biệt, nguyên liệu được nhập khẩu trực tiếp từ các vùng biển Nhật Bản, đảm bảo độ tươi ngon và chất lượng cao.

Việc đặt bàn tại nhà hàng diễn ra vô cùng dễ dàng và thuận tiện, giúp thực khách nhanh chóng có được chỗ ngồi lý tưởng mà không cần mất nhiều thời gian. Chỉ với vài thao tác đơn giản trên website, bạn có thể lựa chọn thời gian, số lượng người và chi nhánh phù hợp, sau đó điền thông tin liên hệ để hoàn tất quá trình đặt bàn. Với rất nhiều chi nhánh tại TP.HCM như Trần Quang Khải, Lê Thánh Tôn, Tôn Thất Thiệp, Phú Nhuận, Củ Chi, quận 7, quận 8 và một chi nhánh tại Đà Nẵng, thuận tiện cho thực khách thưởng thức ẩm thực Nhật Bản ở nhiều địa điểm khác nhau.



Hình 3.23. Giao diện thực đơn trực tuyến nhà hàng Sushi World

3.2.2. Kế hoạch kiểm thử

Kiểm thử	Đối tượng	Loại kiểm thử	Mô tả
----------	-----------	---------------	-------

Trang chủ (TC01)	Giao diện trang chủ	Kiểm thử tải	Kiểm tra khả năng tải trang chủ khi có nhiều người truy cập đồng thời
Thực đơn (TC02)	Giao diện trang thực đơn	Kiểm thử tải	Kiểm tra khả năng tải trang thực đơn khi có nhiều người truy cập đồng thời
Tìm kiếm món ăn theo giá tiền (TC03)	Hệ thống chức năng tìm kiếm món ăn trên trang thực đơn	Kiểm thử hiệu suất	Kiểm tra khả năng tải danh sách thực đơn khi có nhiều truy vấn tìm kiếm món ăn theo giá
Bài viết (TC04)	Giao diện trang bài viết	Kiểm thử tải	Kiểm tra khả năng tải danh sách bài viết khi có nhiều người dùng truy cập đồng thời
Giỏ hàng đặt bàn (TC05)	Hệ thống quản lý thao tác giỏ hàng	Kiểm thử tải, chịu tải	Kiểm tra khả năng hoạt động ổn định vào giờ cao điểm với nhiều người dùng thao tác đồng thời
Đặt bàn (TC06)	Giao diện đặt bàn và xử lý đặt bàn	Kiểm thử tải, chịu tải, độ bền, hiệu suất	Đánh giá hiệu suất xử lý đặt bàn, thời gian phản hồi, khả năng chịu tải khi có nhiều người đặt bàn cùng lúc

Bảng 3.22. Kế hoạch kiểm thử nhà hàng Sushi World

3.2.3. Kịch bản kiểm thử

- Kịch bản kiểm thử trang chủ

Kịch bản kiểm thử trang chủ nhà hàng Sushi World cũng tương tự như của nhà hàng ResOnline được thể hiện ở Bảng 3.2 đảm bảo tính nhất quán trong phương pháp kiểm thử và giúp so sánh kết quả một cách khách quan, hiệu quả. Địa chỉ truy cập kiểm thử: <https://sushiworld.com.vn/>.

- Kịch bản kiểm thử thực đơn

Quy trình kiểm thử trang thực đơn của nhà hàng Sushi World được thiết kế dựa trên các tiêu chí tương đồng với kịch bản đã áp dụng cho nhà hàng ResOnline và được biểu diễn như Bảng 3.3 nhằm so sánh kết quả một cách khách hàng nhất. Địa chỉ truy cập kiểm thử: <https://sushiworld.com.vn/danh-muc/thuc-don-online/>.

- Kịch bản kiểm thử tìm kiếm món ăn theo giá tiền

Kịch bản kiểm thử tìm kiếm món ăn theo giá tiền	
Mục tiêu	Đánh giá khả năng xử lý truy vấn lọc món ăn theo giá khi có nhiều người dùng truy cập cùng lúc. Đảm bảo kết quả tìm kiếm trả về nhanh chóng và hệ thống phản hồi ổn định.
Dữ liệu đầu vào	<ul style="list-style-type: none"> - Mức giá tiền “10000” đến “500000” - Số lượng truy vấn đồng thời: 100, 200, 500, 1000, 1500, 2000, 3000, 5000, 7500, 10000 - Địa chỉ truy vấn: https://sushiworld.com.vn/danh-muc/thuc-don-online/?min_price=10000&max_price=500000
Các bước thực hiện	<p>Bước 1: Tạo nhóm người dùng với số lượng người tương ứng</p> <p>Bước 2: Thêm yêu cầu máy chủ sử dụng các từ khóa tìm kiếm</p> <p>Bước 3: Ghi nhận thời gian phản hồi, tỉ lệ lỗi, kết quả trả về</p> <p>Bước 4: Lặp lại quy trình với số lượng người dùng tăng dần</p>

Kết quả mong đợi	Hệ thống phản hồi với tỉ lệ lỗi dưới 3%, dữ liệu trả về đầy đủ và chính xác theo mức giá tìm kiếm, thời gian phản hồi nhanh và ổn định, không xảy ra tình trạng treo.
-------------------------	---

Bảng 3.23. Kịch bản kiểm thử tìm kiếm món ăn theo giá tiền

- Kịch bản kiểm thử bài viết

Việc kiểm thử bài viết trên trang Sushi World được thực hiện với cấu trúc và tiêu chí tương tự như kịch bản đã áp dụng cho hệ thống nhà hàng ResOnline. Mục tiêu là đảm bảo sự nhất quán trong phương pháp kiểm thử và đánh giá hiệu năng khi tải nội dung bài viết. Chi tiết quy trình được trình bày tại Bảng 3.6. Địa chỉ truy cập kiểm thử: <https://sushiworld.com.vn/danh-muc/thuc-don-online/>.

- Kịch bản kiểm thử giỏ hàng đặt bàn

Quy trình kiểm thử trang thực đơn của nhà hàng Sushi World được thiết kế dựa trên các tiêu chí tương đồng với kịch bản đã áp dụng cho nhà hàng ResOnline và được biểu diễn như Bảng 3.7 nhằm so sánh kết quả một cách khách hàng nhất. Địa chỉ truy cập kiểm thử: <https://sushiworld.com.vn/danh-muc/thuc-don-online/>.

- Kịch bản kiểm thử đặt bàn

Đối với chức năng đặt bàn trên website Sushi World, quy trình kiểm thử được triển khai theo một kịch bản tương đương với chức năng đặt bàn của ResOnline nhằm đảm bảo có thể so sánh hiệu suất xử lý yêu cầu đặt chỗ từ người dùng. Các chỉ số kỹ thuật được mô tả cụ thể trong Bảng 3.3.

Địa chỉ kiểm thử: <https://sushiworld.com.vn/danh-muc/thuc-don-online/>.

3.2.4. Kịch bản kiểm thử**- Kiểm thử trang chủ**

Cấu hình một kịch bản kiểm thử để thực hiện kiểm thử trang chủ với các cài đặt và dữ liệu đầu vào được thể hiện tương tự như Bảng 3.2 với kết quả trả về được thể hiện như bảng bên dưới đây.

Số lượng người dùng	Thời gian phản hồi trung bình (s)	Tỉ lệ lỗi (%)	Nhận xét	Đánh giá
100	3.60	0.00	Không xảy ra lỗi, phản hồi trong thời gian tốt	Đạt
200	9.51	0.00	Không xảy ra lỗi nhưng thời gian phản hồi lâu	Không đạt
500	6.79	53.40	Hệ thống quá tải nhiều thời điểm làm thời gian phản hồi trung bình giảm	Không đạt
1000	4.37	74.20	Tình trạng quá tải tăng	Không đạt
1500	2.83	83.20	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
2000	2.20	88.00	Tỉ lệ lỗi cao, hệ thống đa phần không phản hồi	Không đạt
3000	1.37	91.87	Hệ thống chịu tải lớn	Không đạt
5000	0.86	95.14	Hệ thống gần như quá tải	Không đạt
7500	0.62	97.20	Hệ thống quá tải	Không đạt
10000	0.45	98.57	Hệ thống quá tải	Không đạt

Bảng 3.24. Kết quả kiểm thử trang chủ

TÀI LIỆU THAM KHẢO

- [1] P.C. Jorgensen, B. DeVries, *Software Testing: A Craftsman's Approach*, Fifth Edition, CRC Press, Taylor & Francis Group, 2021.
- [2] A. Spillner, T. Linz, *Software Testing Foundations: A Study Guide for the Certified Tester Exam*, 5th Edition, GmbH, 2021.
- [3] Phạm Quang Huy, Phạm Quang Hiền, *Giáo trình thực hành kiểm thử phần mềm*, NXB Thanh niên, 2020.
- [4] Phạm Ngọc Hùng, Trương Anh Hoàng, Đặng Văn Hưng, *Giáo trình kiểm thử phần mềm*, NXB ĐHQGHN, 2014.