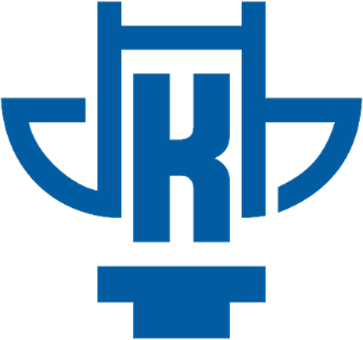
**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**



**KHOA CÔNG NGHỆ THÔNG TIN**

----------

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHOÁ: 2020 – 2024**

***ĐỀ TÀI:***

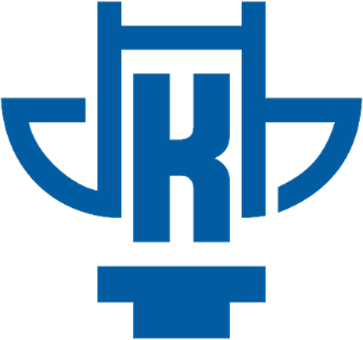
**ỨNG DỤNG JMETER TRONG**

**KIỂM THỬ HIỆU NĂNG CHO HỆ THỐNG WEB.**

**SINH VIÊN THỰC HIỆN: ĐỖ THỊ PHƯƠNG ANH**

**LỚP: 20CN2**

**HÀ NỘI 2-2024**



**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------

**ĐỒ ÁN TỐT NGHIỆP KỸ SƯ**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH: 7480201**

**KHOÁ: 2020 – 2024**

***ĐỀ TÀI:***

**ỨNG DỤNG JMETER TRONG**

**KIỂM THỬ HIỆU NĂNG CHO HỆ THỐNG WEB.**

**GIÁO VIÊN HƯỚNG DẪN: Th.S NGUYỄN THỊ HẠNH**

**SINH VIÊN THỰC HIỆN: ĐỖ THỊ PHƯƠNG ANH**

**LỚP: 20CN2**

**HÀ NỘI 2-2024**

#### LỜI CẢM ƠN

Lời đầu tiên em xin cảm ơn đấng sinh thành đã nuôi dưỡng, cố gắng để cho em được đi học, mở rộng tầm nhìn và tri thức, dõi theo em cho tới quá trình đồ án tốt nghiệp ngày hôm nay.

Tiếp theo, em xin cảm ơn những thầy cô bộ môn khoa Công nghệ thông tin trường Đại học Kiến trúc Hà Nội đã giảng dạy, cho em những bài học, hướng dẫn em từng bước hoàn thiện các kiến thức kĩ năng của ngành học. Ngoài ra, còn các thầy cô mặc dù chưa từng giảng dạy, nhưng gián tiếp giúp cho trải nghiệm quãng thời gian học tập của em tốt đẹp hơn. Em cảm thấy tự hào vì được học trong môi trường có chương trình học luôn cập nhật và theo kịp sự phát triển nhanh chóng của thời đại. Em muốn gửi lời cảm ơn chân thành nhất đến cô giáo Th.S Nguyễn Thị Hạnh – người đã đồng hành, hướng dẫn em khắc phục nhiều những thiếu sót, vượt qua những khó khăn trong quá trình thực hiện đồ án.

Em cũng vô cùng biết ơn thầy Bùi Hải Phong và công ty Cổ phần Tương lai NextX, bởi tại đây em từng được dìu dắt và học tập trong môi trường vô cùng lành mạnh, cầu tiến, sáng tạo, luôn nỗ lực hết mình. <https://nextcrm.vn/> hệ sinh thái web dành cho công việc quản lý kinh doanh mà em sử dụng trong đề tài, cũng là hệ thống được NextX dày công xây dựng và vô cùng tâm huyết.

Trong quá trình thực hiện đồ án này, em nhận thấy mình còn nhiều điều thiếu sót và cần học hỏi nhiều thêm. Em kính mong nhận được những ý kiến góp ý từ phía các thầy cô, bạn bè để em có thể hiểu biết thêm, mở rộng tri thức, trải nghiệm và hoàn thiện đồ án được tốt hơn.

Em xin chân thành cảm ơn!

Hà Nội, tháng 2 năm 2024

Sinh viên thực hiện

Đỗ Thị Phương Anh

#### LỜI CAM ĐOAN CỦA TÁC GIẢ

Em Đỗ Thị Phương Anh, sinh viên lớp 20CN2 khoa Công nghệ thông tin, hiện đang là người thực hiện đồ án tốt nghiệp với đề tài “Ứng dụng JMeter trong kiểm thử hiệu năng cho hệ thống web” tại Trường Đại học Kiến Trúc Hà Nội.

Em cam đoan rằng mọi nội dung và kết quả đều là công sức tự nhiên của bản thân em dưới sự hướng dẫn của ThS. Nguyễn Thị Hạnh – giảng viên hướng dẫn của em tại trường. Em xác nhận rằng không có sự gian lận hoặc làm giả dối trong quá trình thực hiện dự án này. Mọi quá trình làm đồ án được công khai minh bạch và kết quả dựa trên sự cố gắng nỗ lực của bản thân cũng như sự giúp đỡ của thầy cô trong khoa.

Kết quả thực hiện đồ án hoàn toàn không sao chép hoặc sử dụng từ bất kỳ kết quả nghiên cứu của đề tài nào tương tự.

Em hiểu rằng việc vi phạm bất kỳ nguyên tắc nào trong lời cam đoan này sẽ có thể dẫn đến hậu quả xử lý nghiêm túc từ phía nhà trường và em sẽ chịu trách nhiệm về mọi kết quả pháp lý liên quan.

#### LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc đảm bảo chất lượng và hiệu năng của các hệ thống web trở thành một trong những yêu cầu cấp thiết đối với các doanh nghiệp và tổ chức. Hiệu năng của hệ thống không chỉ ảnh hưởng trực tiếp đến trải nghiệm người dùng mà còn đóng vai trò quan trọng trong việc giữ chân khách hàng và tạo lợi thế cạnh tranh trên thị trường.

**Bối cảnh và lý do chọn đề tài:**

Trong quá trình phát triển các hệ thống web, một trong những thách thức lớn là kiểm tra và đánh giá khả năng chịu tải của hệ thống khi có số lượng người dùng lớn truy cập đồng thời. Việc thiếu đi các công cụ và phương pháp kiểm thử hiệu năng sẽ dễ dẫn đến tình trạng hệ thống chậm chạp, thậm chí sập hoàn toàn khi vận hành thực tế. Để giải quyết vấn đề này, Apache JMeter đã được lựa chọn như một công cụ kiểm thử hiệu năng mạnh mẽ và hiệu quả.

**Ý nghĩa khoa học và thực tiễn của đề tài:**

Ý nghĩa khoa học: Đề tài góp phần nghiên cứu và ứng dụng JMeter, một công cụ mã nguồn mở phổ biến, để thực hiện các bài kiểm thử hiệu năng. Việc tìm hiểu và triển khai JMeter giúp bổ sung thêm kiến thức về quy trình kiểm thử hệ thống và các phương pháp đo lường hiệu năng một cách khoa học.

Ý nghĩa thực tiễn: Thông qua việc áp dụng JMeter vào kiểm thử hiệu năng cho hệ thống web, đề tài cung cấp một giải pháp cụ thể giúp các nhà phát triển, kiểm thử viên đánh giá chính xác khả năng đáp ứng của hệ thống. Kết quả của kiểm thử giúp doanh nghiệp tối ưu hệ thống, cải thiện hiệu năng và tăng cường độ ổn định trước khi triển khai chính thức.

**Phạm vi và giới hạn của đề tài:**

Đề tài tập trung vào việc sử dụng Apache JMeter để kiểm thử hiệu năng cho hệ thống web, cụ thể là các bài kiểm thử như:

* Load Test: Đánh giá khả năng chịu tải dưới áp lực truy cập của nhiều người dùng đồng thời.
* Stress Test: Xác định ngưỡng giới hạn của hệ thống trước khi xảy ra lỗi.
* Soak Test: Kiểm tra hiệu năng của hệ thống trong thời gian dài.

Do giới hạn về thời gian và tài nguyên, đề tài không đi sâu vào các loại hình kiểm thử khác như kiểm thử chức năng, bảo mật, kiểm thử giao diện người dùng, hay kiểm thử hiệu năng của database.

**Kết quả đạt được:**

* Tìm hiểu và làm chủ các tính năng cơ bản của Apache JMeter.
* Xây dựng các kịch bản kiểm thử hiệu năng phù hợp với hệ thống web.
* Đánh giá và phân tích kết quả kiểm thử thông qua các báo cáo, từ đó đưa ra các đề xuất để cải thiện hiệu năng hệ thống.

#### MỤC LỤC

[LỜI CẢM ƠN i](#_Toc190475752)

[LỜI CAM ĐOAN CỦA TÁC GIẢ ii](#_Toc190475753)

[LỜI NÓI ĐẦU iii](#_Toc190475754)

[MỤC LỤC 1](#_Toc190475755)

[DANH MỤC CÁC BẢNG 5](#_Toc190475756)

[DANH MỤC CÁC HÌNH ẢNH 6](#_Toc190475757)

[KÝ HIỆU VÀ CHỮ VIẾT TẮT, THUẬT NGỮ 8](#_Toc190475758)

[CHƯƠNG I : ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 9](#_Toc190475759)

[1.1. Khảo sát về đề tài. 9](#_Toc190475760)

[1.1.1. .Lý do chọn đề tài. 9](#_Toc190475761)

[1.1.2. Sự kết hợp giữa JMeter và web. 9](#_Toc190475762)

[1.1.3. Các phương pháp nghiên cứu. 10](#_Toc190475763)

[1.1.4. Thực trạng của lĩnh vực nghiên cứu. 11](#_Toc190475764)

[1.1.5. Đối tượng và phạm vi. 11](#_Toc190475765)

[1.1.6. Kết quả mong đợi ở đề tài này. 11](#_Toc190475766)

[1.2. Tìm hiểu về kiểm thử hiệu năng. 12](#_Toc190475767)

[1.2.1. Định nghĩa. 12](#_Toc190475768)

[1.2.2. Các loại kiểm thử hiệu năng . 12](#_Toc190475769)

[1.2.3. Các công cụ có thể hỗ trợ kiểm thử hiệu năng. 14](#_Toc190475770)

[1.2.4. Các chỉ số cần đo lường. 14](#_Toc190475771)

[1.2.5. Quy trình chung về kiểm thử hiệu năng. 15](#_Toc190475772)

[1.2.6. Phân biệt giữa kiểm thử hiệu năng và các loại kiểm thử khác. 15](#_Toc190475773)

[1.2.7. Ứng dụng của kiểm thử hiệu năng trong thực tế. 16](#_Toc190475774)

[1.2.8. Một số tiêu chí chuyên nghành cho hiệu năng web: 18](#_Toc190475775)

[1.3. Tìm hiểu về công cụ JMeter. [7] 19](#_Toc190475776)

[1.3.1. Giao diện cơ bản của JMeter. 20](#_Toc190475777)

[1.3.2. Thao tác cơ bản trong JMeter. 22](#_Toc190475778)

[1.3.3. Các chức năng chính của JMeter. 23](#_Toc190475779)

[1.4. Các kỹ thuật kiểm thử. 24](#_Toc190475780)

[1.4.1. Kiểm thử hộp đen. 24](#_Toc190475781)

[1.4.2. Kiểm thử hộp trắng. 24](#_Toc190475782)

[1.4.3. Kiểm thử hộp xám. 24](#_Toc190475783)

[1.4.4. Kỹ thuật phân vùng tương đương. 24](#_Toc190475784)

[1.4.5. Phân tích giá trị biên. 24](#_Toc190475785)

[1.4.6. Kỹ thuật pairwise. 25](#_Toc190475786)

[1.4.7. .Đoán lỗi bằng kinh nghiệm tích luỹ. 25](#_Toc190475787)

[CHƯƠNG II : CÔNG CỤ KIỂM THỬ VÀ WEBSITE KIỂM THỬ 26](#_Toc190475788)

[2.1. Xác định chi tiết các bước thực hiện test. 26](#_Toc190475789)

[2.1.1. Bước 1: Xác định môi trường kiếm thử. 26](#_Toc190475790)

[2.1.2. Bước 2: Lập kế hoạch kiểm thử (test plan). 26](#_Toc190475791)

[2.1.3. Bước 3: Thực thi test. 26](#_Toc190475792)

[2.1.4. Bước 4: Tìm hiểu các chức năng nâng cao, 27](#_Toc190475793)

[2.2. Postman: 28](#_Toc190475794)

[2.3. Cài đặt và tìm hiểu giao diện JMeter. 29](#_Toc190475795)

[2.3.1. Kiểm tra phiên bản Java. 29](#_Toc190475796)

[2.3.2. Cài đặt JMeter. 32](#_Toc190475797)

[2.4. Công cụ DevTools trong trình duyệt. 34](#_Toc190475798)

[2.4.1. Giới thiệu chung. 34](#_Toc190475799)

[2.4.2. Tìm key. 34](#_Toc190475800)

[2.5. Về website cần kiểm thử. 36](#_Toc190475801)

[2.5.1. Giao diện web. 37](#_Toc190475802)

[2.5.2. Các chức năng chính. 39](#_Toc190475803)

[CHƯƠNG III : THỰC HIỆN KIỂM THỬ 44](#_Toc190475804)

[3.1. Đánh giá tính quan trọng của từng chức năng. 44](#_Toc190475805)

[3.1.1. Đánh giá các tiêu chí. 44](#_Toc190475806)

[3.1.2. Đánh giá mức độ quan trọng của các nhóm chức năng. 44](#_Toc190475807)

[3.1.3. Đánh giá chi tiết các loại kiểm thử hiệu năng. 46](#_Toc190475808)

[3.2. Tạo Test plan và các thành phần cơ bản. 47](#_Toc190475809)

[3.2.1. Thêm Test Plan. 47](#_Toc190475810)

[3.2.2. Thêm Config element. 47](#_Toc190475811)

[3.2.3. Thêm Thread group. 48](#_Toc190475812)

[3.2.4. Thêm HTTP Request. 48](#_Toc190475813)

[3.2.5. HTTP Header Manager. 49](#_Toc190475814)

[3.2.6. Graph result. 50](#_Toc190475815)

[3.2.7. Thêm Listener để xem kết quả. 51](#_Toc190475816)

[3.3. Testcase. 52](#_Toc190475817)

[3.3.1. Chức năng đăng nhập. 53](#_Toc190475818)

[3.3.2. Thêm đơn đặt hàng. 54](#_Toc190475819)

[3.3.3. Chức năng lên đơn. 54](#_Toc190475820)

[3.3.4. Đối tác giao hàng. 55](#_Toc190475821)

[3.3.5. Đặt hàng. 55](#_Toc190475822)

[3.3.6. Tạo phiếu thu. 56](#_Toc190475823)

[3.3.7. Đánh giá mức độ nghiêm trọng của lỗi. 57](#_Toc190475824)

[3.3.8. Đề xuất khắc phục. 57](#_Toc190475825)

[CHƯƠNG IV : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 57](#_Toc190475826)

[4.1. Kết luận. 57](#_Toc190475827)

[4.2. Định hướng phát triển. 59](#_Toc190475828)

[PHỤ LỤC 60](#_Toc190475829)

[TÀI LIỆU THAM KHẢO 61](#_Toc190475830)

#### DANH MỤC CÁC BẢNG

[CHƯƠNG I : ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 8](#_Toc190475679)

[Bảng I.1. Phân biệt các loại kiểm thử hiệu năng. 12](#_Toc190475680)

[Bảng I.2. Phân biệt kiểm thử hiệu năng và các loại kiểm thử khác. 14](#_Toc190475681)

[CHƯƠNG II : CÔNG CỤ KIỂM THỬ VÀ WEBSITE KIỂM THỬ 25](#_Toc190475682)

[CHƯƠNG III : THỰC HIỆN KIỂM THỬ 43](#_Toc190475683)

[Bảng III.1. Đánh giá chi tiết các loại kiểm thử hiệu năng. 45](#_Toc190475684)

[CHƯƠNG IV : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc190475685)

#### DANH MỤC CÁC HÌNH ẢNH

[CHƯƠNG I : ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 8](#_Toc190475696)

[Hình I.1. Màn hình chờ thanh toán vé concert Anh trai say hi- ảnh từ vov.vn 16](#_Toc190475697)

[Hình I.2. [6] ​​Google Website Speed Recommendations for 2025. 18](#_Toc190475698)

[Hình I.3. Testplan trong JMeter 19](#_Toc190475699)

[Hình I.4. Sampler trong Thread group: vị trí, các chức năng. 20](#_Toc190475700)

[Hình I.5. Config element: vị trí, các chức năng. 21](#_Toc190475701)

[Hình I.6. Sampler trong JMeter. 22](#_Toc190475702)

[CHƯƠNG II : CÔNG CỤ KIỂM THỬ VÀ WEBSITE KIỂM THỬ 25](#_Toc190475703)

[Hình II.1. Truy cập web postman install. 28](#_Toc190475704)

[Hình II.2. Mở cmd. 29](#_Toc190475705)

[Hình II.3. Đã có môi trường java thích hợp (trên laptop của em). 29](#_Toc190475706)

[Hình II.4. MacOS chưa có java [11] 29](#_Toc190475707)

[Hình II.5. Windows chưa có java [11] 30](#_Toc190475708)

[Hình II.6. Chọn phiên bản java mới nhất. 30](#_Toc190475709)

[Hình II.7. Chọn phiên bản phù hợp với laptop. 31](#_Toc190475710)

[Hình II.8. Chọn nơi lưu file và cài đặt. 31](#_Toc190475711)

[Hình II.9. Tải về file zip JMeter. 32](#_Toc190475712)

[Hình II.10. Chọn vị trí tải về trên laptop. 32](#_Toc190475713)

[Hình II.11. Giải nén file. 33](#_Toc190475714)

[Hình II.12. Dùng devtools tìm kiếm key. 34](#_Toc190475715)

[Hình II.13. Dùng postman kiểm tra key-value. 34](#_Toc190475716)

[Hình II.14. Hệ và phân hệ web NextX 35](#_Toc190475717)

[Hình II.15. Hệ sinh thái đa dạng. 35](#_Toc190475718)

[Hình II.16. Hệ thống khách hàng lên đến hơn 10.000 người dùng. 36](#_Toc190475719)

[Hình II.17. Giao diện đăng nhập. 36](#_Toc190475720)

[Hình II.18. Giao diện chọn Chi nhánh và Phân hệ. 37](#_Toc190475721)

[Hình II.19. Giao diện Tổng quan. 37](#_Toc190475722)

[Hình II.20. NextCRM: chức năng Tổng quan. 38](#_Toc190475723)

[Hình II.21. NextCRM: chức năng Khách hàng và nhánh các chức năng nhỏ. 39](#_Toc190475724)

[Hình II.22. NextCRM: Một số chức năng khác. 40](#_Toc190475725)

[Hình II.23. NextCRM: Chức năng quản lý Sổ quỹ, Nhân viên, Báo cáo. 41](#_Toc190475726)

[Hình II.24. Các chức năng hỗ trợ từ ban quản lý. 41](#_Toc190475727)

[Hình II.25. Chức năng trong bán online. 42](#_Toc190475728)

[Hình II.26. Chức năng lên đơn cùng hàng chục chức năng hỗ trợ từng nhu cầu nhỏ. 42](#_Toc190475729)

[CHƯƠNG III : THỰC HIỆN KIỂM THỬ 43](#_Toc190475730)

[Hình III.1. Testplan trong JMeter. 46](#_Toc190475731)

[Hình III.2. HTTP Request default. 46](#_Toc190475732)

[Hình III.3. Thread Group nơi giả lập thao tác người dùng. 47](#_Toc190475733)

[Hình III.4. HTTP Request nơi ghi các thông tin cần thiết để gửi request tới server. 47](#_Toc190475734)

[Hình III.5. HTTP Header Manager. 48](#_Toc190475735)

[Hình III.6. Graph Result nơi biểu diễn trực quan thông số phản hồi. 49](#_Toc190475736)

[Hình III.7. Công thức tính Deviation độ lệch chuẩn thời gian phản hồi so với Avegare Response Time. 49](#_Toc190475737)

[Hình III.8. View result tree nơi xem kết quả server trả cho request. 50](#_Toc190475738)

[Hình III.9. Summary Report. 51](#_Toc190475739)

[Hình III.11. Lấy phương thức request là GET. 52](#_Toc190475740)

[Hình III.12. Test login với postman 52](#_Toc190475741)

[Hình III.10. Thêm đơn đặt hàng. 53](#_Toc190475742)

[Hình III.13. Chức năng đặt hàng 54](#_Toc190475743)

[CHƯƠNG IV : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc190475744)

#### KÝ HIỆU VÀ CHỮ VIẾT TẮT, THUẬT NGỮ

\*Giải nghĩa trong đồ án này được dùng để giải thích cách dùng từ trong ngữ cảnh phù hợp.

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ | Giải nghĩa |

|  |  |  |
| --- | --- | --- |
|  | Client | Ý chỉ máy khách. |
|  | DevOps Engineer | Kỹ sư phát triển – vận hành hệ thống công nghệ thông tin. |
|  | Distributed testing | Kiểm tra phân tán |
|  | SLA | Service level agreement – thoả thuận mức độ dịch vụ. Đối với test hiệu năng thì nó được hiểu là điểm ước lượng ổn với trải nghiệm người dùng. |
|  | Throughtput | Đầu ra |
|  | Pass/passed | Vượt bài kiểm tra. |
|  | Plugin | Hay còn gọi là add-on hoặc extension, là một phần mở rộng bổ sung tính năng cho một phần mềm, ứng dụng hoặc hệ thống mà không cần thay đổi mã nguồn gốc. |

# ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT

## Khảo sát về đề tài.

### .Lý do chọn đề tài.

* Ý nghĩa đề tài :

Kiểm thử hiệu năng thường bị coi là công việc đòi hỏi kiến thức sâu về hạ tầng, mạng và tối ưu hóa hệ thống – công việc dành cho các kiến trúc sư hệ thống, DevOps Engineer. Dẫn đến việc ít người lựa chọn theo đuổi lĩnh vực này. Tuy nhiên, với sự hỗ trợ của các công cụ như JMeter, việc tiếp cận kiểm thử hiệu năng đã trở nên dễ dàng hơn, thậm chí đối với những người không chuyên sâu về kỹ thuật hạ tầng.

* Dựa trên ưu điểm của công cụ JMeter: [1]

JMeter là công cụ mã nguồn mở, miễn phí, phù hợp cho mọi quy mô dự án từ nhỏ đến lớn.

Tính linh hoạt và đa năng: JMeter hỗ trợ kiểm thử nhiều giao thức như HTTP, HTTPS, SOAP, REST, FTP và cả các ứng dụng web động, giúp dễ dàng áp dụng vào hầu hết các hệ thống web hiện nay.

Cộng đồng mạnh mẽ và tài liệu phong phú: JMeter có cộng đồng người dùng rộng lớn, tài liệu và hướng dẫn chi tiết, giúp người mới dễ dàng tiếp cận và triển khai kiểm thử hiệu năng nhanh chóng.

### Sự kết hợp giữa JMeter và web.

Việc nghiên cứu ứng dụng kiểm thử JMeter vào test hệ thống web mang lại lợi ích ứng dụng cao, bởi web không đòi hỏi phải cài đặt vào client, thuận lợi trong việc đưa web ra sử dụng thời gian ngắn với lưu lượng truy cập khổng lồ, khi không cần thì không nằm trong máy như app. Web có thể truy cập nhanh chóng vì vậy lượt truy cập có thể tăng cao đột biến đòi hỏi tính chịu tải của hệ thống hạ tầng máy chủ, hệ thống back end, hệ thống dịch vụ web.

Ưu điểm của việc sử dụng JMeter với web: [1]

* Mô phỏng tải thực tế cho hệ thống web: JMeter cho phép mô phỏng số lượng lớn người dùng truy cập đồng thời để kiểm tra khả năng chịu tải của website. Hữu ích để đo hiệu năng trước khi triển khai hệ thống lên môi trường thực tế.
* Hỗ trợ nhiều giao thức liên quan đến web, như HTTP/HTTPS, WebSocket, SOAP, REST API, v.v. Điều này giúp kiểm thử các ứng dụng web ở nhiều mức độ (frontend, backend, API).
* Tích hợp dễ dàng vào quy trình DevOps: JMeter có thể tích hợp vào CI/CD pipeline thông qua Jenkins, GitLab CI/CD, giúp kiểm thử hiệu năng liên tục khi phát triển ứng dụng web.
* Mã nguồn mở và miễn phí: Là giải pháp hiệu quả về chi phí so với các công cụ thương mại như LoadRunner hoặc NeoLoad.
* Khả năng mở rộng: Cho phép thiết lập distributed testing để mô phỏng lưu lượng lớn hơn bằng cách sử dụng nhiều máy tính.
* Phân tích hiệu năng chi tiết: JMeter có thể tạo báo cáo hiệu quả. Kết quả kiểm tra có thể được xem lại bằng cách sử dụng Graph, Chart, and Tree View. JMeter hỗ trợ các định dạng khác nhau của báo cáo như text, XML, HTML and JSON.
* Dễ sử dụng: Dễ sử dụng với các thành phần như Thread Groups, Samplers và Listeners để tạo kịch bản kiểm thử mà không cần nhiều kiến thức lập trình.

### Các phương pháp nghiên cứu.

Để nghiên cứu cách kiểm thử hiệu năng : Tra cứu tài liệu, tự đặt câu hỏi và thực hành, nhờ sự hướng dẫn của giảng viên hướng dẫn, ghi chép lại quá trình và dữ liệu, thực hiện phân tích dữ liệu.

* Phân tích , tổng hợp thông tin: Phân tích chủ đề đồ án tốt nghiệp, phân tích từng chữ và cụm từ, để tìm ra hướng đi, hướng phát triển của đồ án. Phân tích các bài báo, nguồn thông tin khổng lồ tham khảo được trên mạng, sau đó chắt lọc và tổng hợp, tút gọn lại để nội dung cô đọng. Phân tích các khía cạnh xã hội để tìm ra sự liên kết của đề tài với thực tiễn.
* Phương pháp so sánh: so sánh về nhu cầu người dùng, sử dụng so sánh trong testcase để tìm ra ưu nhược điểm của sản phẩm. Việc so sánh và tham chiếu cũng đem lại hiệu quả thuyết phục người dùng cuối.
* Phương pháp liệt kê thông tin: để làm rõ các bằng chứng về kết quả đồ án, giúp người đọc hình dung cụ thể giá trị của sản phẩm, cụ thể ý nghĩa cần truyền đạt.
* Phương pháp thống kê số liệu : từ số liệu được ghi nhận thực tế để tìm ra vấn đề.

### Thực trạng của lĩnh vực nghiên cứu.

Người thực hiện : nên là người có kỹ thuật và hiểu biết về hạ tầng mạng, thiết bị, biết cách tối ưu hoá hệ thống.

Không có số liệu nào thống kê có bao nhiêu người làm công việc kiểm thử hiệu năng, nhưng qua thời gian thực tập, bản thân em nhận thấy có ít các tester làm về kiểm thử hiệu năng trong dự án thực tế, dù đã được đào tạo hay chưa. Nhưng đây vẫn là điều cần thiết để làm nếu bất kỳ công ty nào cần vận hành hệ thống hạ tầng thiết bị và internet.

Tiêu chuẩn ngành hoặc Best Practices: Các các công ty công nghiệp lớn như Google, Amazon, hoặc Netflix, có các yêu cầu rất cụ thể về thời gian phản hồi và throughput trong các bài kiểm thử hiệu năng.

### Đối tượng và phạm vi.

* CPU: Xử lý hiệu quả các yêu cầu từ người dùng, khả năng xử lý đồng thời.
* Mức tối ưu của phương pháp thiết kế: Đánh giá tính tối ưu trong cấu trúc thiết kế hệ thống, như phân phối tài nguyên, cách thức phân phối tải và tối ưu các thuật toán.
* Dung lượng tiêu thụ: Xem xét lượng băng thông và tài nguyên cần thiết để duy trì hiệu suất khi có lượng truy cập lớn, từ đó phát hiện sự tắc nghẽn mạng, hạn chế tài nguyên hoặc cấu hình không tối ưu.

### Kết quả mong đợi ở đề tài này.

* Xác định khả năng chịu tải của hệ thống:

Hệ thống có thể chịu đựng một số lượng người dùng đồng thời nhất định mà không gặp phải sự cố hoặc giảm hiệu suất nghiêm trọng.

Đánh giá rõ ràng giới hạn của hệ thống về số lượng yêu cầu có thể xử lý trong một khoảng thời gian cụ thể.

* Phát hiện điểm nghẽn (Bottlenecks):

Xác định các yếu tố ảnh hưởng đến hiệu suất như CPU, bộ nhớ, băng thông mạng hoặc các thành phần khác trong hệ thống khi lượng truy cập tăng cao.

Đưa ra các điểm yếu trong cấu trúc thiết kế hoặc hạ tầng, từ đó giúp cải thiện hiệu suất của hệ thống.

* Hiệu suất và khả năng mở rộng của hệ thống:

Xác định mức độ hiệu quả của việc mở rộng hệ thống khi tải trọng tăng (ví dụ: tăng số lượng người dùng hoặc dữ liệu) và kiểm tra tính ổn định của hệ thống khi mở rộng.

Đánh giá khả năng tối ưu hóa các thành phần hệ thống như máy chủ, cơ sở dữ liệu, hoặc mạng.

* Sự ổn định và bền vững của hệ thống:

Xác định khả năng duy trì hiệu suất trong thời gian dài (kiểm thử độ bền) mà không gặp phải sự cố như rò rỉ bộ nhớ, tắc nghẽn mạng hay mất dữ liệu.

Phát hiện các vấn đề tiềm ẩn trong hệ thống có thể gây ra sự cố sau một thời gian hoạt động liên tục.

* Khuyến nghị cải thiện hiệu suất:

Dựa trên kết quả kiểm thử, đưa ra các khuyến nghị và biện pháp cải thiện hiệu suất hệ thống, bao gồm nâng cấp phần cứng, tối ưu hóa mã nguồn hoặc thay đổi cấu trúc thiết kế.

* Đánh giá mức độ tối ưu của phương pháp thiết kế:

Xác định mức độ tối ưu trong thiết kế hệ thống, từ đó đưa ra các đề xuất về cách thức triển khai và phân phối tài nguyên sao cho hiệu quả nhất.

## Tìm hiểu về kiểm thử hiệu năng.

### Định nghĩa.

Kiểm thử hiệu năng( Performance Test) là một quy trình hoặc phương pháp được sử dụng để đánh giá hiệu suất và khả năng chịu tải của một ứng dụng hoặc hệ thống. Mục đích của việc thực hiện Performance Test nhằm xác định các chỉ số hiệu suất như thời gian phản hồi, tải trọng tối đa mà hệ thống có thể chịu được.

### Các loại kiểm thử hiệu năng .

Load test( kiểm thử tải): Đánh giá khả năng của hệ thống khi có một lượng lớn người dùng truy cập cùng lúc, nhưng vẫn ở mức tải bình thường.

Stress test (kiểm thử sức chịu đựng): Xác định giới hạn tối đa của hệ thống (khi vượt ngưỡng tải cho phép) và quan sát cách hệ thống xử lý khi quá tải. Break point test là một dạng mở rộng của Stress test là mức tải cụ thể mà hệ thống bắt đầu suy giảm nghiêm trọng hoặc sụp đổ hoàn toàn, cần quan sát tài nguyên hệ thống (CPU, RAM, băng thông, kết nối DB) để tìm ra điểm nghẽn, kiểm tra hệ thống phục hồi như thế nào sau khi đạt đến điểm phá vỡ. Đồ án này sẽ không test break point vì khả năng gây hậu quả nghiêm trọng của nó tới hệ thống đang hoạt động với nhiều khách hàng.

Soak test/Endurance Test (Kiểm thử độ bền): Kiểm tra xem hệ thống có duy trì hiệu năng ổn định khi hoạt động liên tục trong thời gian dài.

Spike Test (Kiểm thử tăng đột biến): Đánh giá cách hệ thống phản ứng khi tải tăng đột ngột trong một thời gian ngắn.

Scalability Test (Kiểm thử khả năng mở rộng): Đánh giá khả năng của hệ thống khi tài nguyên (CPU, RAM, băng thông) được mở rộng.

Volume test: kiểm tra phần mềm hoặc ứng dụng về hiệu năng của nó đối với dữ liệu khổng lồ của cơ sở dữ liệu. VD báo cáo, đồng bộ...Kiểu kiểm thử này tập trung vào Speed

Concurrency Test (Kiểm thử đồng thời): kiểm tra cách hệ thống xử lý nhiều yêu cầu đồng thời từ nhiều user tại cùng một thời điểm( giống Load test). Chủ yếu kiểm tra đồng bộ dữ liệu, khóa dữ liệu, tranh chấp tài nguyên, deadlock và hiệu năng khi nhiều user cùng thao tác trên một chức năng cụ thể.

Phân biệt các loại kiểm thử hiệu năng : [2]



Bảng I.1. Phân biệt các loại kiểm thử hiệu năng.

### Các công cụ có thể hỗ trợ kiểm thử hiệu năng.

Apache JMeter: Công cụ phổ biến, miễn phí, mạnh mẽ để kiểm thử hiệu năng các hệ thống web.

LoadRunner (Micro Focus): Công cụ chuyên nghiệp, nhưng thường dùng trong các dự án lớn do chi phí cao.

K6: Công cụ kiểm thử hiệu năng mã nguồn mở, nhẹ và dễ sử dụng.

Gatling: Phù hợp cho các bài kiểm tra hiệu năng ứng dụng web với kịch bản phức tạp.

### Các chỉ số cần đo lường.

Response Time (Thời gian phản hồi): Thời gian để hệ thống xử lý và trả về kết quả cho một yêu cầu.

Throughput (Lưu lượng): Số lượng yêu cầu mà hệ thống xử lý trong một giây.

Error Rate (Tỷ lệ lỗi): Tỷ lệ phần trăm yêu cầu bị lỗi khi tải cao.

CPU và Memory Usage (Sử dụng tài nguyên): Kiểm tra mức tiêu thụ tài nguyên của hệ thống trong các tình huống tải khác nhau.

Các chỉ số trong Garph result:

* Median(trung vị - đơn vị : ms): Là giá trị ở giữa danh sách thời gian phản hồi khi đã sắp xếp tăng dần. Trong trường hợp các giá trị lẻ thì median là giá trị response time của response có thứ tự chính giữa. Trong trường hợp các giá trị chẵn thì median là giá trị response time trung bình của 2 response có thứ tự chính giữa trong chuỗi các response. Median không quan tâm các response time có được trả về đồng thời hay không, nó chỉ sắp xếp tất cả response time và lấy giá trị giữa.
* Average(trung bình- đơn vị: ms): Là thời gian phản hồi trung bình (response time) của tất cả các response trong test. Công thức:

Các chỉ số trong summary report:

* Std.Dev:
* %Error:
* Receipt(KB/sec):
* Sent (KB/sec):

### Quy trình chung về kiểm thử hiệu năng.

Bước 1. Xác định, phân tích yêu cầu.

Bước 2. Thiết lập môi trường kiểm thử: Cài đặt JMeter, chuẩn bị môi trường sử dụng web, xây dựng kịch bản kiểm thử.

Bước 3. Thực hiện kiểm thử: Chạy các bài kiểm tra theo các mức tải khác nhau.

Bước 4. Phân tích kết quả: So sánh với yêu cầu (ví dụ: thời gian phản hồi tối đa là 2 giây).

Bước 5. Tối ưu hóa và kiểm tra lại: Điều chỉnh cấu hình server, cơ sở dữ liệu hoặc mã nguồn để cải thiện hiệu năng.

### Phân biệt giữa kiểm thử hiệu năng và các loại kiểm thử khác.



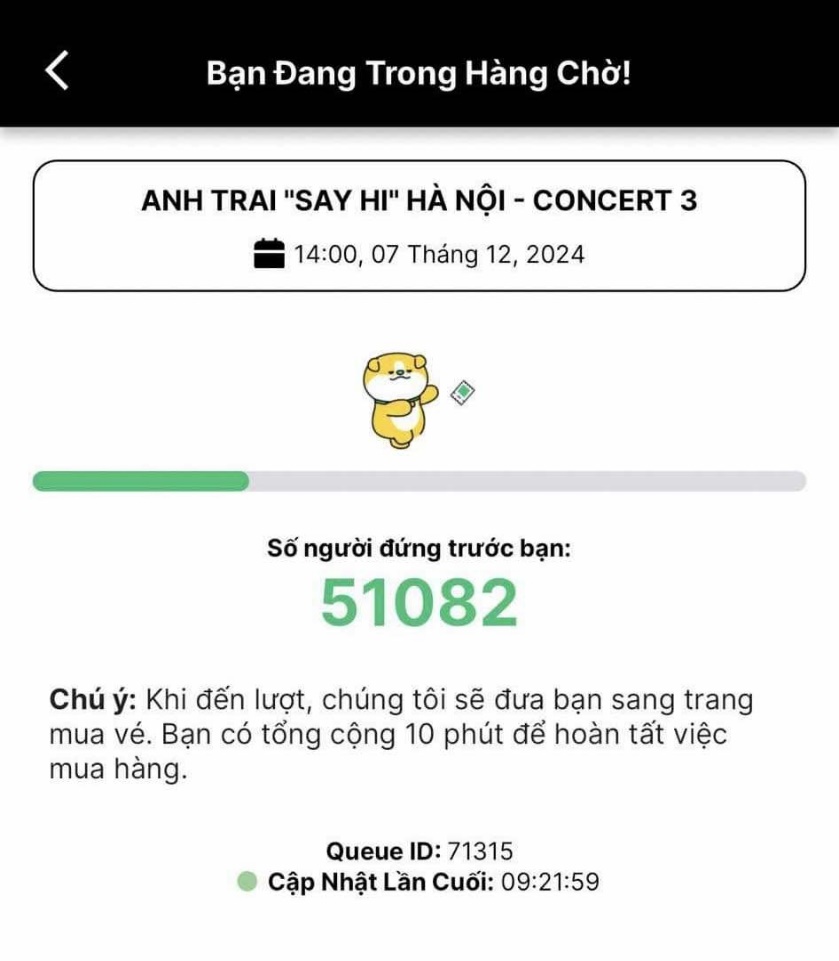
Bảng I.2. Phân biệt kiểm thử hiệu năng và các loại kiểm thử khác.

### Ứng dụng của kiểm thử hiệu năng trong thực tế.

Vai trò của kiểm thử hiệu năng trong hệ thống web:

* Đảm bảo trải nghiệm người dùng tốt: Người dùng sẽ rời bỏ hệ thống nếu website tải chậm hoặc không ổn định. Kiểm thử hiệu năng giúp cải thiện tốc độ phản hồi và độ ổn định.
* Đo lường khả năng mở rộng: Giúp đánh giá khả năng xử lý khi số lượng người dùng tăng lên.
* Phát hiện nút cổ chai (Bottleneck): Tìm ra các thành phần trong hệ thống (database, server, mạng) gây chậm trễ để tối ưu hóa.
* Hỗ trợ các loại kiểm thử khác: Một hệ thống bảo mật và chức năng đầy đủ vẫn cần đảm bảo hiệu năng để đáp ứng nhu cầu thực tế.

Ví dụ thực tế: Các sự kiện giải trí cần bán vé online từ website như Born Pink World tour Hà Nội 2023, concert Anh trai “say hi”, chỉ trong thời gian ngắn đã có hàng chục nghìn lượt truy cập đăng ký mua vé [3], vì vậy đòi hỏi back end của web cần logic và tối ưu trước khi đưa tới tay người dùng.



Hình I.1. Màn hình chờ thanh toán vé concert Anh trai say hi- ảnh từ vov.vn

Ứng dụng kiểm thử hiệu năng trong việc giả định các cuộc tấn công của hacker: Các hacker có thể lợi dụng khả năng chịu tải kém để phá hoại hệ thống, đó là cuộc tấn công mạng DoS và DDoS, ảnh hưởng quá trình vận hành hệ thống và trải nghiệm người dùng, đồng thời lợi dụng thời gian hệ thống sụp đổ để gửi các dữ liệu xấu tới các thiết bị khác thông qua web hoặc email [4]

Thời gian tải nhanh hơn cải thiện trải nghiệm người dùng , dẫn đến nhiều lượt xem trang hơn, thời lượng phiên dài hơn và thứ hạng công cụ tìm kiếm tốt hơn. Ví dụ: một trang tải trong 1 giây có thể có tỷ lệ chuyển đổi cao hơn tới 5 lần so với trang mất 10 giây để tải. [5]

### Một số tiêu chí chuyên nghành cho hiệu năng web:

[5]Thời gian tải trang lý tưởng được coi là dưới 2 giây. Các tiêu chuẩn công nghiệp, như Google, Amazon**, hoặc Netflix, có các yêu cầu cụ thể về thời gian phản hồi và throughput trong các bài kiểm thử hiệu năng. Ví dụ:**

* **Google yêu cầu thời gian tải trang nhanh dưới 3 giây.**
* **Amazon báo cáo rằng mỗi giây trễ có thể làm giảm 7% doanh thu.**

Trên thực tế thời gian tải trung bình thay đổi tùy thuộc vào nền tảng:

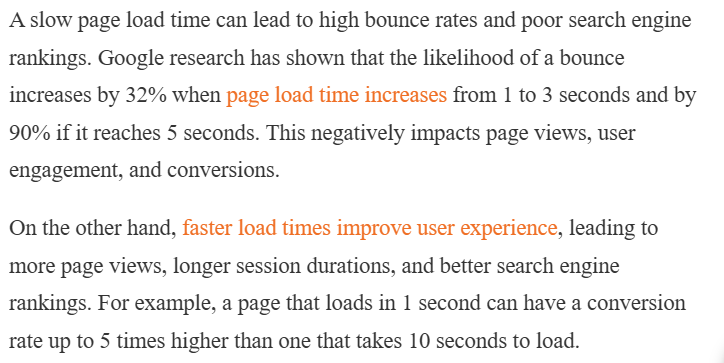
* Máy tính để bàn: 2,5 giây
* Di động: 8,6 giây
* Các trang web công nghệ: 6,8 giây
* Các trang web du lịch: 6,7 giây
* Các trang web bán lẻ: 6 giây
* Các trang web chăm sóc sức khỏe: 5,6 giây

Các yếu tố chính ảnh hưởng đến thời gian tải trang:

* Tối ưu hóa hình ảnh: Nén hình ảnh và sử dụng định dạng tệp phù hợp để giảm thời gian tải.
* Giảm thiểu các plugin và tập lệnh: Hạn chế các tập lệnh của bên thứ ba có thể làm chậm trang web.
* Bộ nhớ đệm của trình duyệt: Bật bộ nhớ đệm để tăng tốc độ tải trang cho khách truy cập quay lại.
* Mạng phân phối nội dung (CDN): Sử dụng CDN để rút ngắn khoảng cách dữ liệu phải truyền đi trên toàn cầu.

Thời gian tải nhanh hơn cải thiện trải nghiệm người dùng , dẫn đến nhiều lượt xem trang hơn, thời lượng phiên dài hơn và thứ hạng công cụ tìm kiếm tốt hơn. Ví dụ: một trang tải trong 1 giây có thể có tỷ lệ chuyển đổi cao hơn tới 5 lần so với trang mất 10 giây để tải.

[12]



Hình I.2. [6] ​​Google Website Speed Recommendations for 2025.

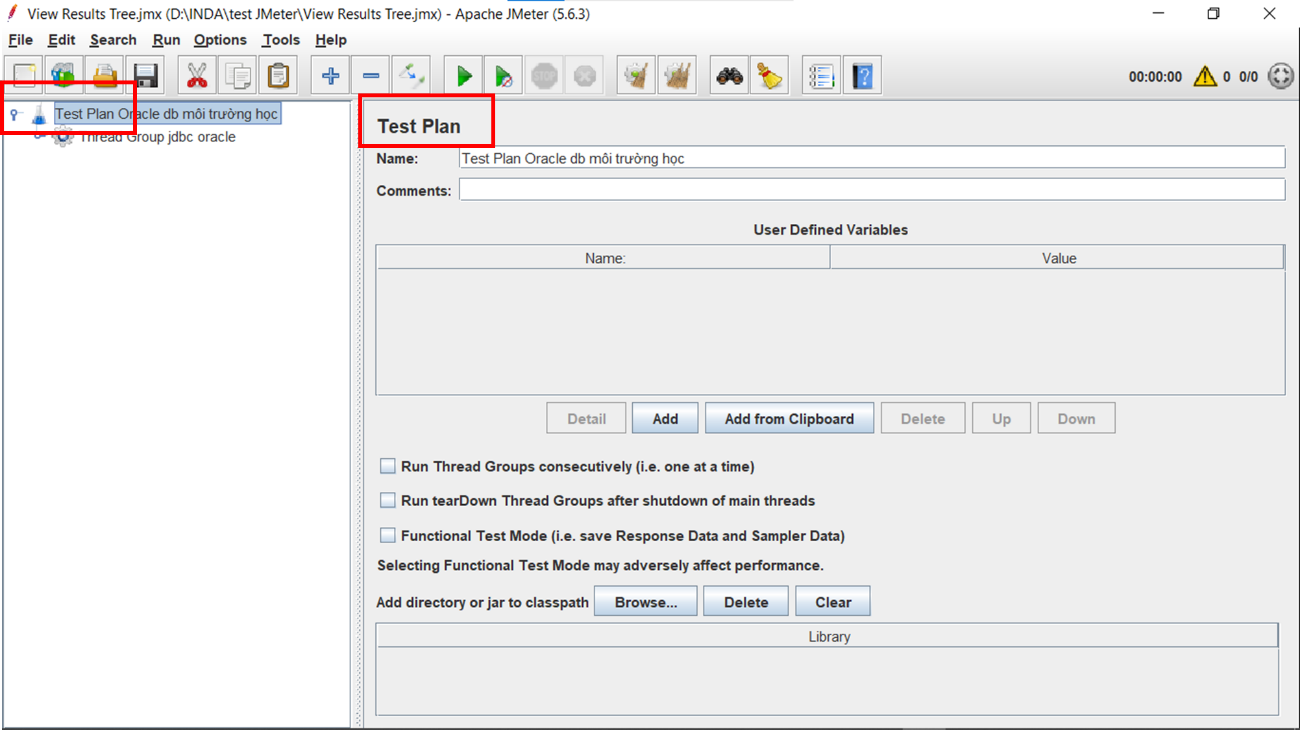
Đối với website tiêu chí được xác định dựa trên 5 yếu tố:

* Tài liệu yêu cầu của hệ thống( đây là phần em không có, cho nên không thể dựa vào tài liệu để đánh giá kết quả hệ thống).
* Đặc điểm của website.
* So sánh với tiêu chuẩn ngành.
* Kết quả kiểm tra hiệu năng.
* Kỳ vọng của người dùng( có thể xây dựng tiêu chuẩn dựa trên tiêu chí này).

## Tìm hiểu về công cụ JMeter. [7]

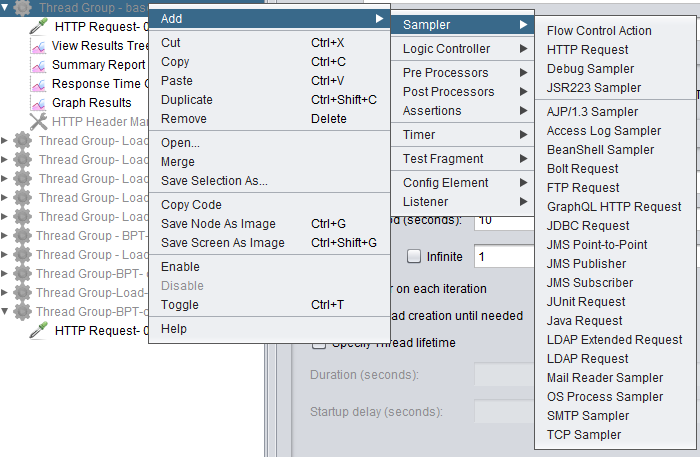
Phần này cung cấp cơ sở ban đầu để ứng dụng JMeter một cách hiệu quả trong kiểm thử hiệu năng của hệ thống web:

### Giao diện cơ bản của JMeter.



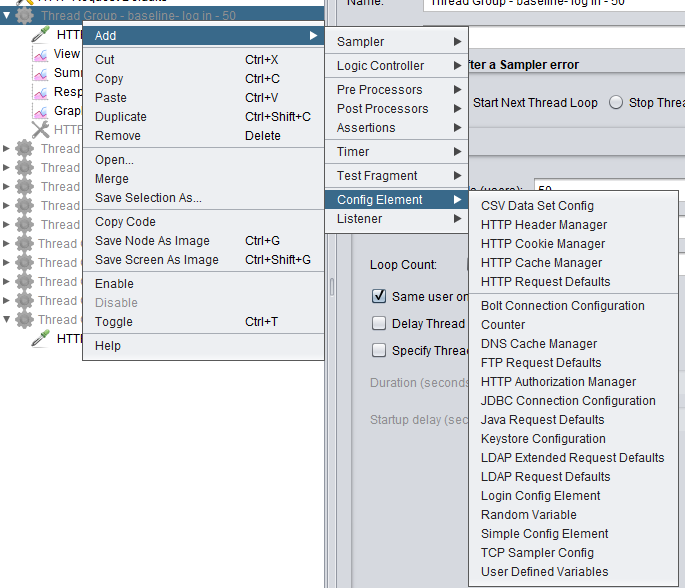
Hình I.3. Testplan trong JMeter

* Test Plan: Tạo kế hoạch kiểm thử bao gồm các bước và thủ tục.
* Thread Group: Thiết lập số luồng user ảo, tần suất request, thời gian chạy.
* Sampler : gồm các chức năng( http request, giúp mô phỏng các loại yêu cầu của người dùng đến máy chủ. Đây là thành phần bắt buộc phải có nếu không test plan sẽ không chạy được.



Hình I.4. Sampler trong Thread group: vị trí, các chức năng.

* Listeners: Hiển thị kết quả test (Graph, Table, Log...)
* Config element : là các đối tượng cấu hình request, một cấu trúc cung cấp khả năng tạo ra những giá trị mặc định cho các request ở bên trong testplan, hoặc sẽ tạo ra các biến với các gía trị cụ thể cho trước mà sau đó sẽ được gọi bên trong test plan để sử dụng 1 hoặc nhiều lần; cũng hỗ trợ cấu hình để đọc đọc nội dung của 1 file data, truyền các data đó từ file bên ngoài vào bên trong testplan để sử dụng, Sampler có thể sử dụng dữ liệu thông qua các biến này.



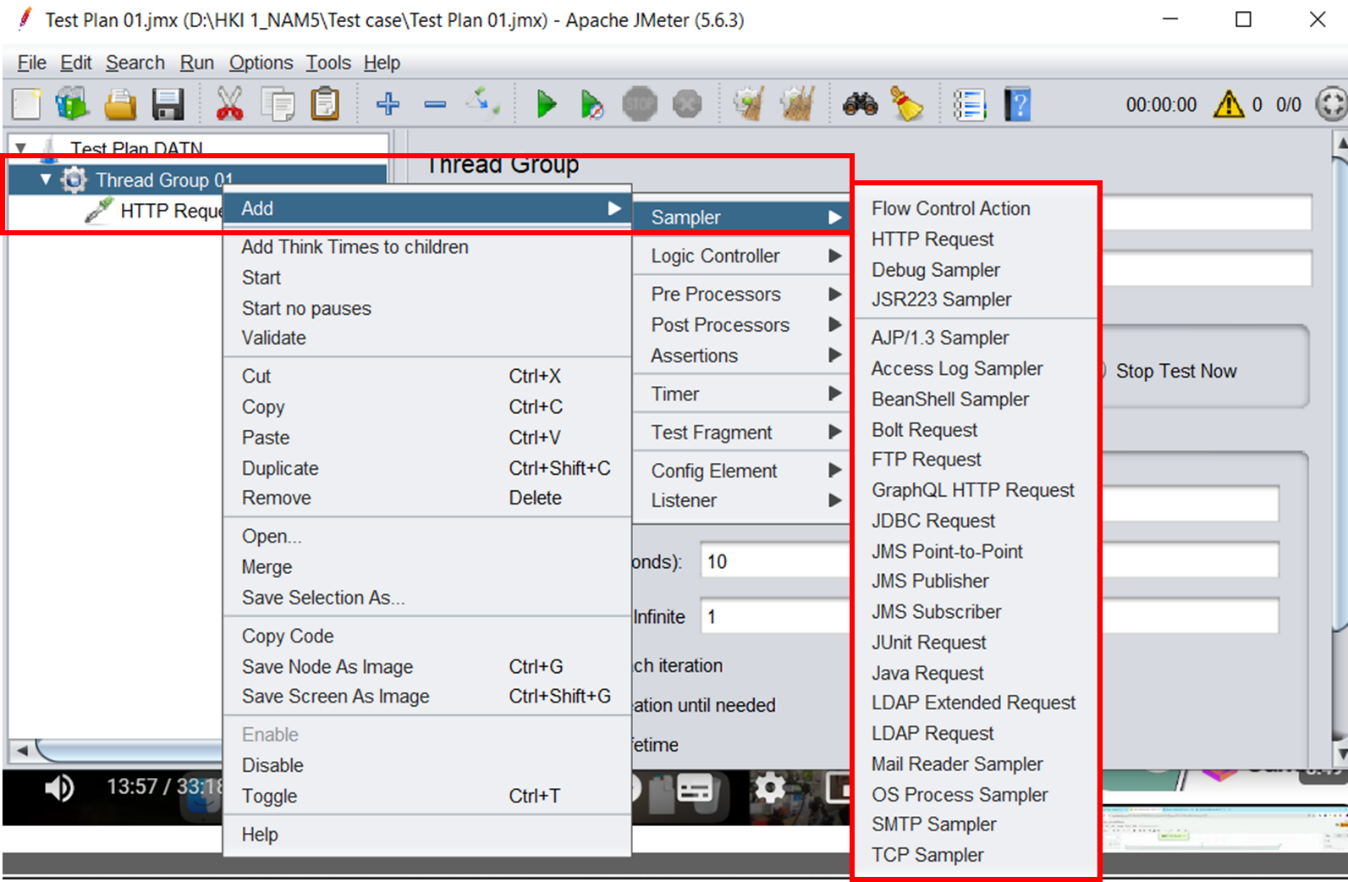
Hình I.5. Config element: vị trí, các chức năng.

### Thao tác cơ bản trong JMeter.

* Tạo Test Plan và cài đặt các Thread Group.
* Thêm HTTP Request để kiểm thử các API hoặc trang web.
* Sử dụng Assertions để xác minh kết quả test (vd: HTTP Code, text trên trang).
* Số lượng Thread: Mỗi Thread đại diện cho một người dùng ảo, JMeter cho phép thay đổi số lượng người dùng không hạn chế để thực hiện các thử nghiệm.
* Ram-Up Period: Thời gian để bắt đầu tất cả những Thread.
* Loop Count: Số lần lặp lại những yêu cầu của người dùng. Ngoài ra còn có những tùy chọn khác như việc chạy các Thread vào lịch biểu định sẵn, xác định hành động sẽ thực hiện khi xảy ra lỗi…
* Controller: HTTP Request Defaults. Tạo HTTP Request Defaults: Add > Config Element > HTTP Request Defaults.
* HTTP Request Defaults: Định nghĩa trang web mà mình sẽ thực hiện xuyên suốt kịch bản test. 1 kịch bản test có thể có nhiều Http request khác nhau ở các URI (Path) khác nhau, nhưng đều xuất phát ở cùng 1 domain được định nghĩa ở HTTP Request Defaults.
* Bảng HTTP Request Defaults, nhập tên trang web, port cần được kiểm tra
* Http request: Định nghĩa 1 request mô phỏng cho 1 chức năng/thao tác của user trên hệ thống.
* Để thêm mới Http request: Add -> Sampler -> HTTP Request.
* Throughput(request/s) [8]

### Các chức năng chính của JMeter.

* Kiểm thử hiệu năng: Load Test, Stress Test, Soak Test,…
* Hỗ trợ đa giao thức: HTTP, FTP, JDBC, SOAP/REST API...



Hình I.6. Sampler trong JMeter.

* Báo cáo: Tự động tổng hợp và hiển thị kết quả qua đồ thị hoặc file, ví dụ : Summary Report, Graph Results,…

## Các kỹ thuật kiểm thử.

### Kiểm thử hộp đen.

Kiểm thử hộp đen (Black Box Testing): Kiểm tra đầu vào và đầu ra mà không cần quan tâm đến cách hoạt động bên trong của hệ thống. Mục tiêu là kiểm tra các chức năng của phần mềm dựa trên các yêu cầu đầu vào.

### Kiểm thử hộp trắng.

Kiểm thử hộp trắng (White Box Testing) : Test Data được sử dụng để kiểm tra logic bên trong và các dòng mã của phần mềm, các luồng xử lý, và hiệu suất của các thuật toán, cấu trúc bảng, ràng buộc.

### Kiểm thử hộp xám.

Kiểm thử hộp xám : là một phương pháp kiểm thử phần mềm mà trong đó người kiểm thử có một hiểu biết hạn chế về cấu trúc nội bộ của hệ thống, nhưng vẫn có khả năng truy cập vào mã nguồn hoặc cơ sở dữ liệu. Phương pháp này kết hợp giữa kiểm thử hộp trắng (white box testing) và kiểm thử hộp đen (black box testing).

### Kỹ thuật phân vùng tương đương.

Định nghĩa : Phân vùng tương đương là phương pháp chia các điều kiện đầu vào thành những vùng tương đương nhau. Tất cả các giá trị trong một vùng tương đương sẽ cho một kết quả đầu ra giống nhau. Vì vậy chúng ta có thể test một giá trị đại diện trong vùng tương đương. Các lớp tương đương được xác định bằng bằng cách lấy mỗi trạng thái đầu vào (thường là 1 câu hay 1 cụm từ trong đặc tả) và phân chia nó thành 2 hay nhiều nhóm. [9]

### Phân tích giá trị biên.

Kỹ thuật phân tích giá trị biên (Boundary Value Analysis - BVA) là Bảng quyết định: trong đó các testcase được thiết kế bao gồm các giá trị tại các biên. Nếu dữ liệu đầu vào được sử dụng là trong giới hạn giá trị biên, nó được cho là Positive testing. Nếu dữ liệu đầu vào được sử dụng là ngoài giới hạn giá trị biên, nó được cho là Negative testing. Mục tiêu là lựa chọn các test case để thực thi giá trị biên. [9]

Các bước cụ thể sử dụng bảng quyết định:

Bước 1: Xác định các lớp tương đương theo đầu vào hoặc đầu ra. Với mỗi điều kiện đầu vào hoặc đầu ra được mô tả trong đặc tả yêu cầu thì lấy ra

* Lớp thoả mãn điều kiện
* Lớp không thoả mãn điều kiện

Bước 2: Thiết kế testcase và lựa chọn test dựa vào các lớp tương đương trên.

### Kỹ thuật pairwise.

Là một phương pháp test kết hợp mỗi cặp 2 tham số đầu vào của 1 bộ các đối tượng có liên quan đến nhau, tạo ra bộ giá trị kiểm thử: Ta sẽ kiểm tra tất cả các khả năng có thể kết hợp các giá trị của cặp 2 tham số đó với nhau. Thực hiện kiểm tra theo cặp như vậy sẽ giúp làm giảm thời gian hơn rất nhiều so với việc phải kiểm tra đầy đủ mọi khả năng kết hợp của tất cả các giá trị của bộ nhiều các thông số với nhau. [10]

Thiết kế testcase bằng kỹ thuật paiwise:

Bước 1: Xác định tất cả các trạng thái có thể của ứng dụng hoặc hệ thống.

Bước 2: Xác định các sự kiện hoặc hành động có thể xảy ra ở mỗi trạng thái.

Bước 3: Vẽ State Transition Diagram để biểu diễn các trạng thái và chuyển đổi giữa chúng.

Bước 4 : Tạo ra các test case cho mỗi chuyển đổi trạng thái để đảm bảo rằng chuyển đổi giữa các trạng thái hoạt động như mong đợi.

### .Đoán lỗi bằng kinh nghiệm tích luỹ.

Phương pháp dựa vào kinh nghiệm, trực quan của người kiểm thử để đưa ra các trường hợp kiểm thử. Trong quá trình kiểm thử, người kiểm thử có thể áp dụng nhiều kỹ thuật khác nhau, kết hợp các phương pháp kiểm thử với nhau để có thể tìm các lỗi của phần mềm một cách tối đa.

# CÔNG CỤ KIỂM THỬ VÀ WEBSITE KIỂM THỬ

## Xác định chi tiết các bước thực hiện test.

### Bước 1: Xác định môi trường kiếm thử.

Xác định được các yếu tố này để thiết kế và lập kế hoạch test hiệu quả hơn, xác định được những thách thức trong quá trình test.

* Web là môi trường đã đưa vào sử dụng thực tế cho công ty và các doanh nghiệp khác nhau.
* Hệ thống web gồm những thành phần nào ? Hệ thống cấu thành nên web là cả 1 hệ sinh thái, do hạn chế về điều kiện tiếp cận nên không thể tìm hiểu kỹ về phần cứng, các câu lệnh, xử lý logic sâu bên trong back end, hệ thống data.
* Cấu hình mạng như thế nào?
* Xác định các tiêu chí chấp nhận: Bao gồm số CCU, %Error, Response time, TPS, %RAM, %CPU. Các tiêu chí này được đưa ra trong bộ KPI của hệ thống, hoặc dựa trên hợp đồng vớikhách hàng ...

### Bước 2: Lập kế hoạch kiểm thử (test plan).

* Thiết lập kịch bản kiểm thử (Test Plan) trên google document: Chọn lựa các kịch bản chính, các chức năng chính, hay sử dụng, chức năng cần xử lý lượng dữ liệu lớn ...Xác định các biến đổi giữa các người dùng (phân quyền khác nhau với các role khác nhau), xác định dữ liệu test.
* Tạo Test plan bằng các chức năng testplan, thread group, view tree, …. Trong JMeter, nơi có thể thêm các bước tương tác với web. Ví dụ, mô phỏng các yêu cầu HTTP, các truy vấn API, hoặc các tương tác với server.
* Cấu hình môi trường: Chuẩn bị công cụ test, tài nguyên test. VD: tool giả lập tải, máy client đủ mạnh để đẩy tải, cần bao nhiêu máy client ...
* Cấu hình các yêu cầu HTTP (HTTP Request Sampler).
* Thực hiện thiết kế các script test theo kế hoạch test.

### Bước 3: Thực thi test.

* Cấu hình các yêu cầu HTTP gửi đến server đó. Điều này bao gồm việc thiết lập URL, method (GET, POST...), và các dữ liệu cần thiết (headers, body).
* Tạo dữ liệu ảo: thêm data. Còn username và password thì không thể tạo được, phải xác nhận qua người quản lý hệ thống.
* Theo dõi hiệu suất và thu thập dữ liệu, đánh giá dữ liệu.
* Sử dụng các loại listeners (như Summary Report, View Results Tree, Aggregate Graph) giúp thu thập và phân tích dữ liệu về thời gian phản hồi, tỉ lệ lỗi, và hiệu suất tổng thể của ứng dụng .
* Kiểm thử tải và khả năng mở rộng (Load Testing & Scalability), kiểm thử stress (Stress Testing), …

### Bước 4: Tìm hiểu các chức năng nâng cao,

Các chức năng nâng cao giúp ích trong việc phân tích tối ưu hoá kịch bản kiểm thử trong testplan JMeter.

Logic Controller : Cho phép kiểm soát luồng thực thi của các request. Các controller phổ biến:

* If Controller: Thực thi điều kiện dựa trên logic "nếu-đúng" (If-Else).
* Loop Controller: Lặp lại một nhóm các request với số lần cố định.
* While Controller: Lặp lại dựa trên điều kiện cho đến khi sai.
* Switch Controller: Thực thi một nhánh request cụ thể dựa trên giá trị.
* Runtime Controller: Thực thi request trong khoảng thời gian cố định.

Pre-Processor và Post-Processor:

* Pre-Processor: Thực hiện thao tác trước khi gửi request.

Ví dụ:

User Parameters: Tùy chỉnh giá trị trước khi request chạy.

HTTP URL Re-writing Modifier: Thêm token session hoặc thông tin vào URL.

* Post-Processor: Xử lý kết quả sau khi nhận response.

Ví dụ:

Regular Expression Extractor: Trích xuất dữ liệu từ response (token, ID).

JSON Extractor: Trích xuất giá trị từ JSON response.

Assertions (Xác nhận): Xác minh tính đúng đắn của response.

* Response Assertion: Kiểm tra response có chứa cụm từ, mã trạng thái, hoặc giá trị mong muốn.
* JSON Assertion: Xác minh cấu trúc JSON.
* XPath Assertion: Kiểm tra response theo XPath (XML).
* Size Assertion: Kiểm tra kích thước dữ liệu nhận được.

Timers (Bộ đếm thời gian): Dùng để kiểm soát khoảng thời gian giữa các request.

* Constant Timer: Thêm một khoảng dừng cố định.
* Gaussian Random Timer: Thêm khoảng dừng ngẫu nhiên dựa trên giá trị trung bình và độ lệch chuẩn.
* Uniform Random Timer: Thêm khoảng dừng ngẫu nhiên trong một khoảng nhất định.

Data Parameterization (Tham số hóa dữ liệu): Kịch bản kiểm thử sử dụng dữ liệu từ file hoặc biến động.

* CSV Data Set Config: Tải dữ liệu từ file CSV để tạo các request độc lập.
* User Defined Variables: Định nghĩa biến dùng trong toàn bộ kịch bản.

Cách dùng :

Bước 1 : Add

Bước 2: Đặt tên biết bất kỳ (name), đặt value là IP address, description để mô tả chúc năng, vị trí cần dùng biến.

Cách gọi biến (user defined variable) :

${Variable\_name}gọi tại “Server name or IP” của “HTTP Request Defaults”.

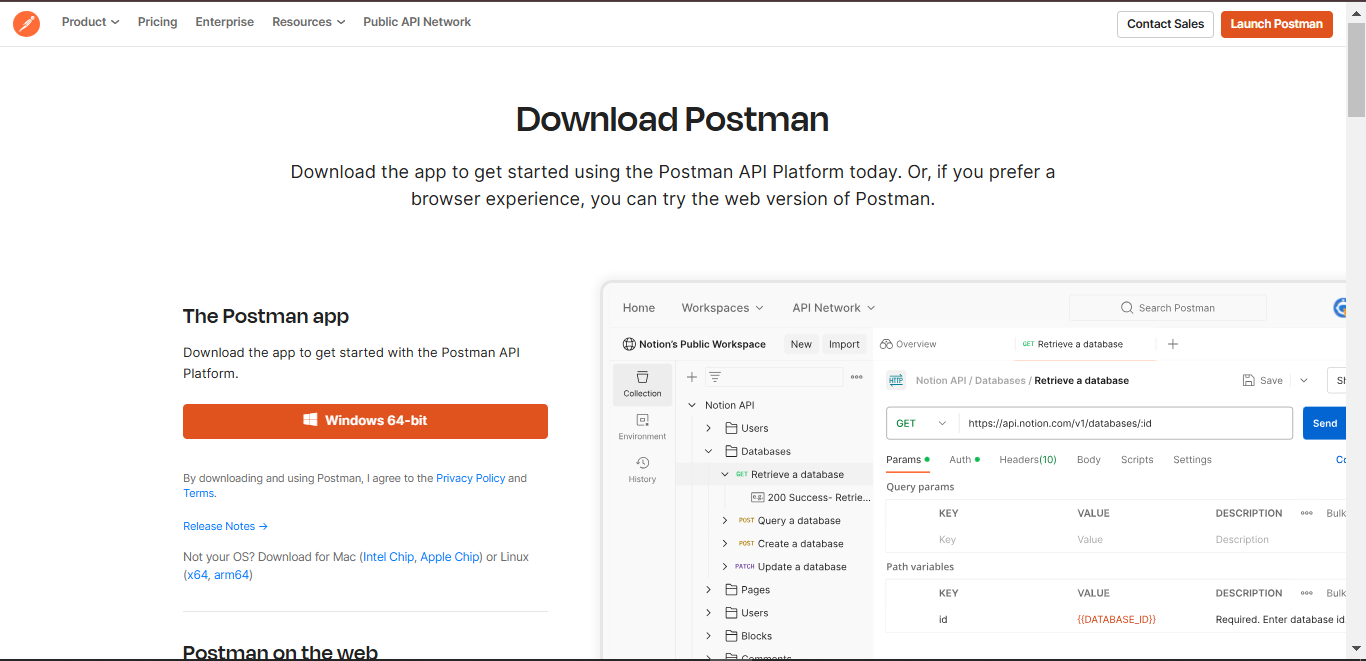
Sampler nâng cao:

* Beanshell Sampler: Viết code Java để thực hiện thao tác đặc biệt.
* JDBC Request: Kiểm thử truy vấn SQL đối với cơ sở dữ liệu.
* JMS Publisher/Subscriber: Kiểm thử hệ thống nhắn tin JMS.
* TCP Sampler: Kiểm tra giao tiếp TCP giữa client và server.

## Postman:

Postman đã được cài sẵn trong máy từ trước. Sau đây là các bước cơ bản để tải.

* Bước 1: Truy cập đường dẫn sau: <https://www.postman.com/downloads/>
* Bước 2: Bấm chọn nút window64 bit.
* Bước 3: Chọn vị trí lưu file, bấm OK và chờ tải.
* Bước 4 : Mở bằng cách chuột phải vào biểu tượng app Postman, bấm Open.



Hình II.1. Truy cập web postman install.

## Cài đặt và tìm hiểu giao diện JMeter.

### Kiểm tra phiên bản Java.

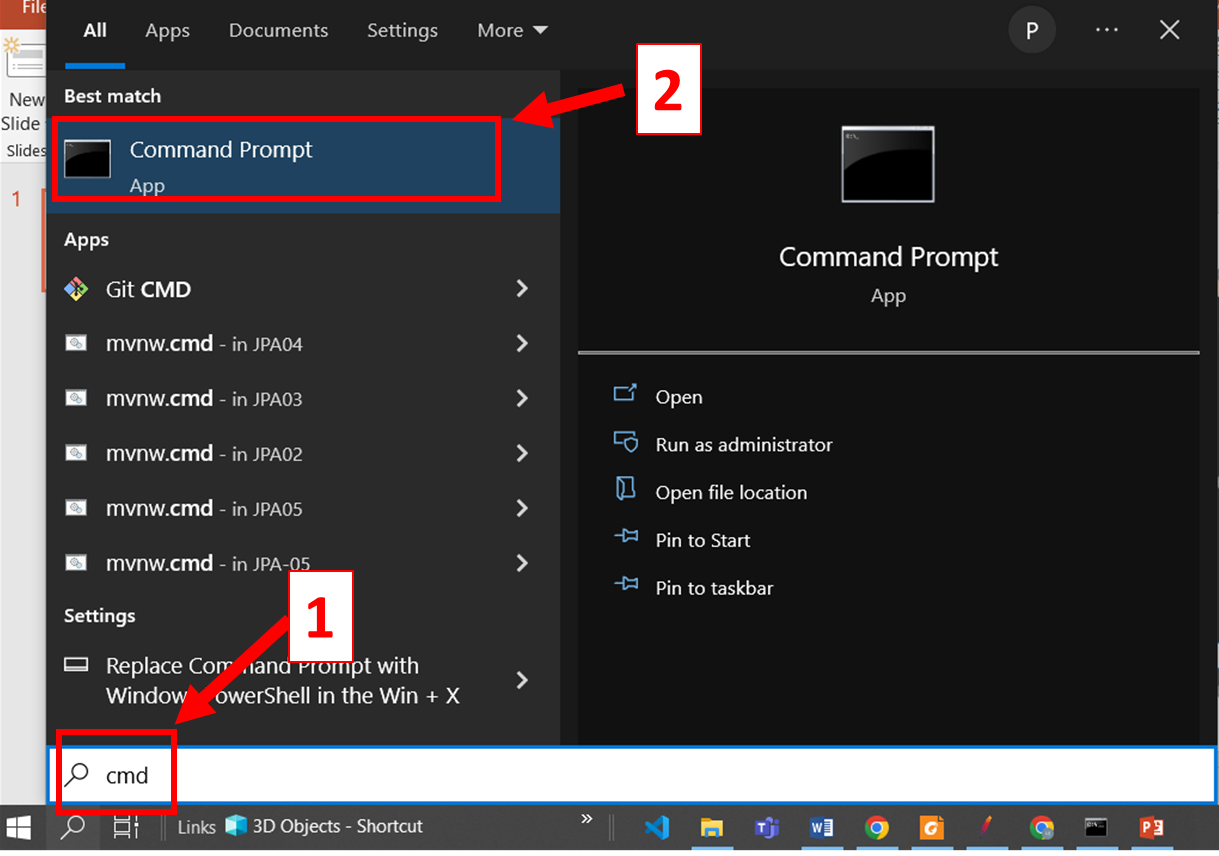
JMeter được viết hoàn toàn bằng ngôn ngữ lập trình Java, do đó nó yêu cầu một môi trường Java Runtime Environment (JRE) hoặc Java Development Kit (JDK) được cài đặt trên máy tính, giúp JMeter thực thi các tác vụ. JMeter không thể hoạt động độc lập mà cần môi trường Java để chuyển mã nguồn Java thành các lệnh máy tính có thể hiểu và thực hiện.

Phiên bản Java phù hợp: JMeter thường yêu cầu phiên bản Java từ 8 trở lên. Khi cài đặt JMeter, cần đảm bảo rằng Java đã được cấu hình đúng và biến môi trường là file ojdbc8.jar hoặc ojdbc11.jar đã được thiết lập để JMeter nhận diện Java.

Tương tác với hệ điều hành: Nhờ Java, JMeter có thể chạy trên nhiều nền tảng khác nhau (Windows, macOS, Linux) mà không cần thay đổi mã nguồn.

Thực hiện kiểm tra phiên bản Java trên máy tính cá nhân như sau:

Bước 1 : Mở cmd



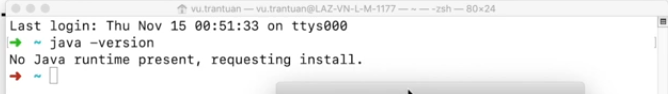
Hình II.2. Mở cmd.

Bước 2: gõ “java -version” → enter. Hiển thị như hình dưới nghĩa là môi trường đã có java, version từ 8 trở lên là dùng được JMeter. Chuyển luôn tới mục “2.3.2. Cài đặt JMeter”.



Hình II.3. Đã có môi trường java thích hợp (trên laptop của em).

* Hiển thị như này nghĩa là môi trường chưa có java, chuyển qua “bước 3: tải JDK” :



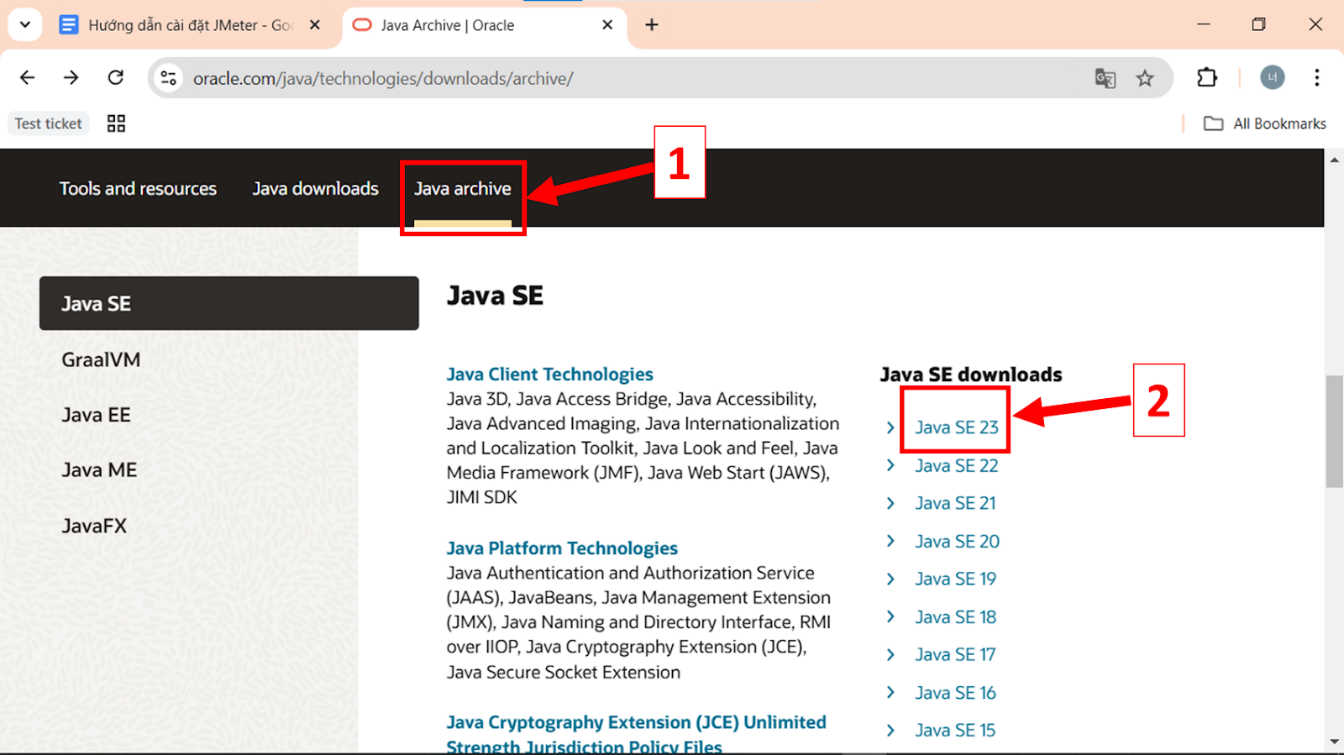
Hình II.4. MacOS chưa có java [11]



Hình II.5. Windows chưa có java [11]

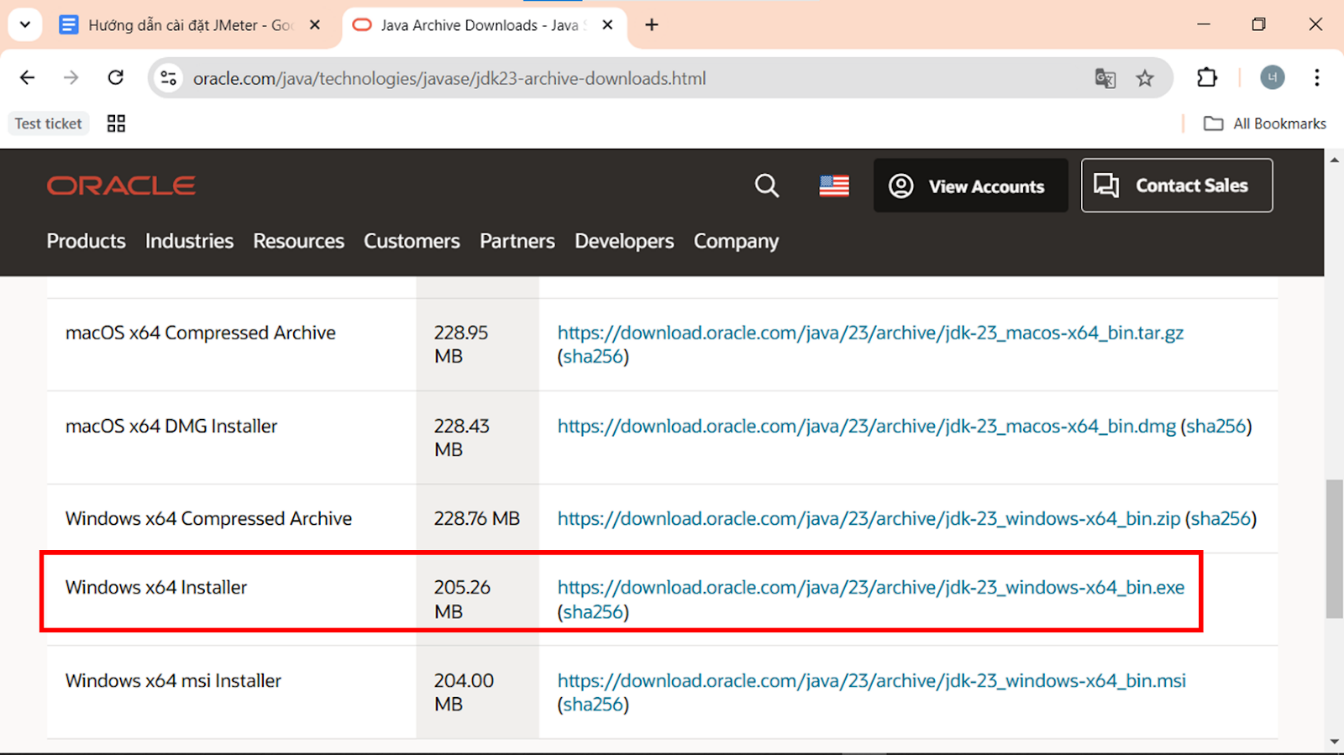
Bước 3 : Tải JDK.

* Truy cập link sau: <https://www.oracle.com/java/technologies/downloads/#java8>
* Chọn phiên bản JDK mới nhất:



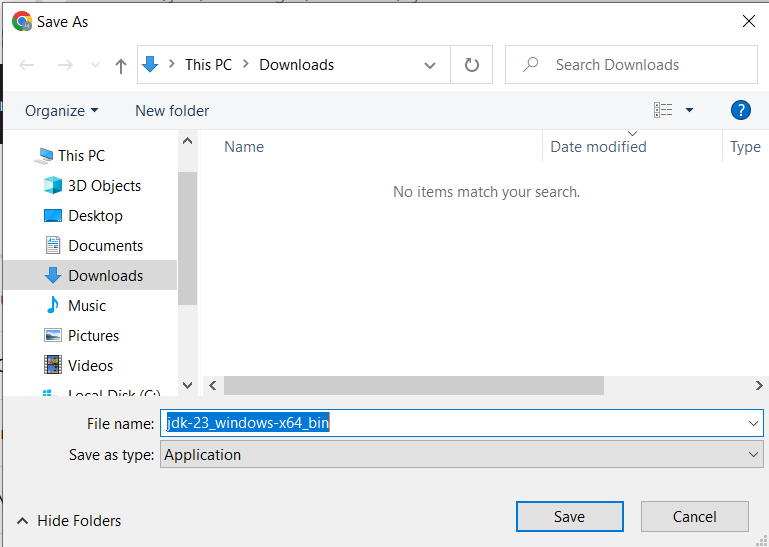
Hình II.6. Chọn phiên bản java mới nhất.

* Tiếp theo, chọn phiên bản phù hợp với máy của bạn và tải về, máy mình dùng là hệ điều hành window 64bit, nên mình sẽ chọn liên kết này:



Hình II.7. Chọn phiên bản phù hợp với laptop.

* Bấm save:

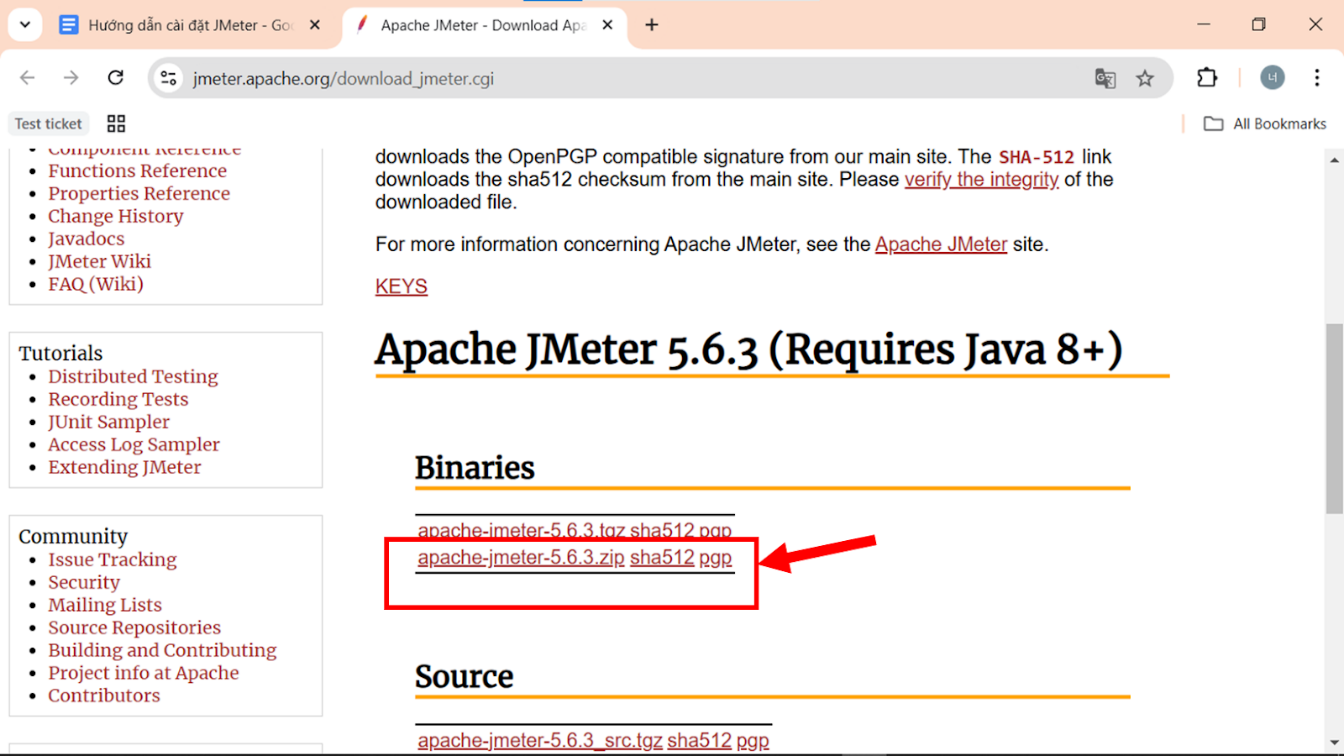


Hình II.8. Chọn nơi lưu file và cài đặt.

### Cài đặt JMeter.

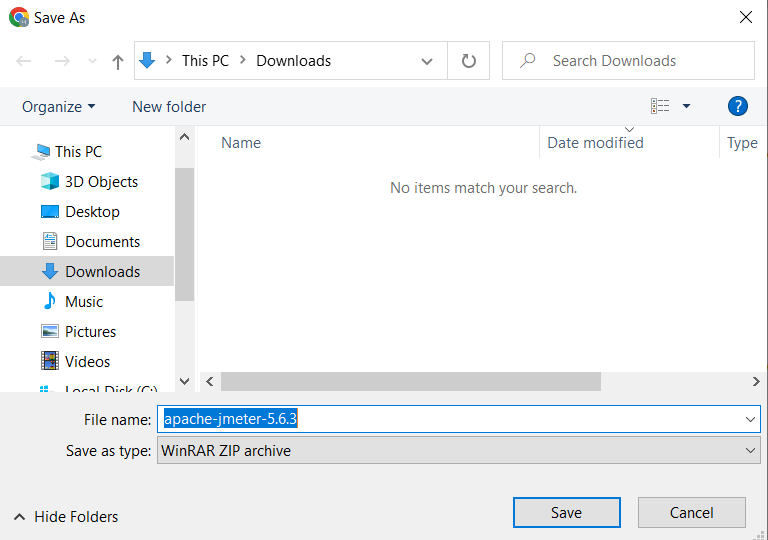
Bước 1: truy cập link sau - trang chính thức của JMeter. <https://jmeter.apache.org/download_jmeter.cgi>

Bước 2: chọn tải về file zip.



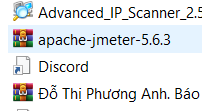
Hình II.9. Tải về file zip JMeter.

Bước 3 : Chọn vị trí đặt file và chờ hoàn tất là xong.



Hình II.10. Chọn vị trí tải về trên laptop.

Bước 4: Giải nén file.



Hình II.11. Giải nén file.

## Công cụ DevTools trong trình duyệt.

### Giới thiệu chung.

* Cách mở tool này trên web : phím Ctrl + Shift + I
* Giới thiệu các tab và chức năng trong devtools:

Element ghi lại thành phần html, console,…

Network ghi lại trạng thái của web hiện tại, lịch sử thao tác thời gian phản hồi, cookies, …

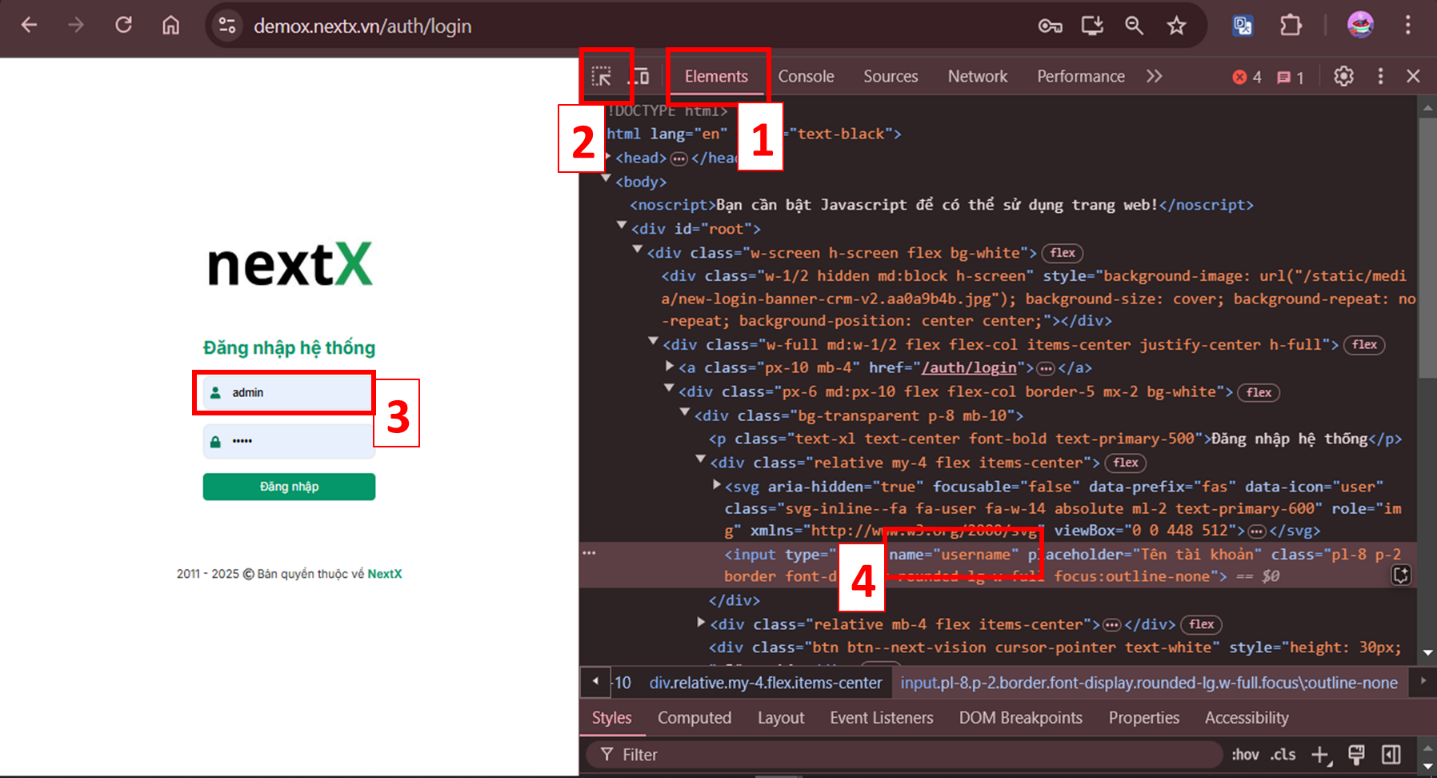
Bộ lọc( Filter) và tìm kiếm.

* Devtools( hay developer tools) là công cụ hữu ích hỗ trợ cho test web và các web developer

### Tìm key.

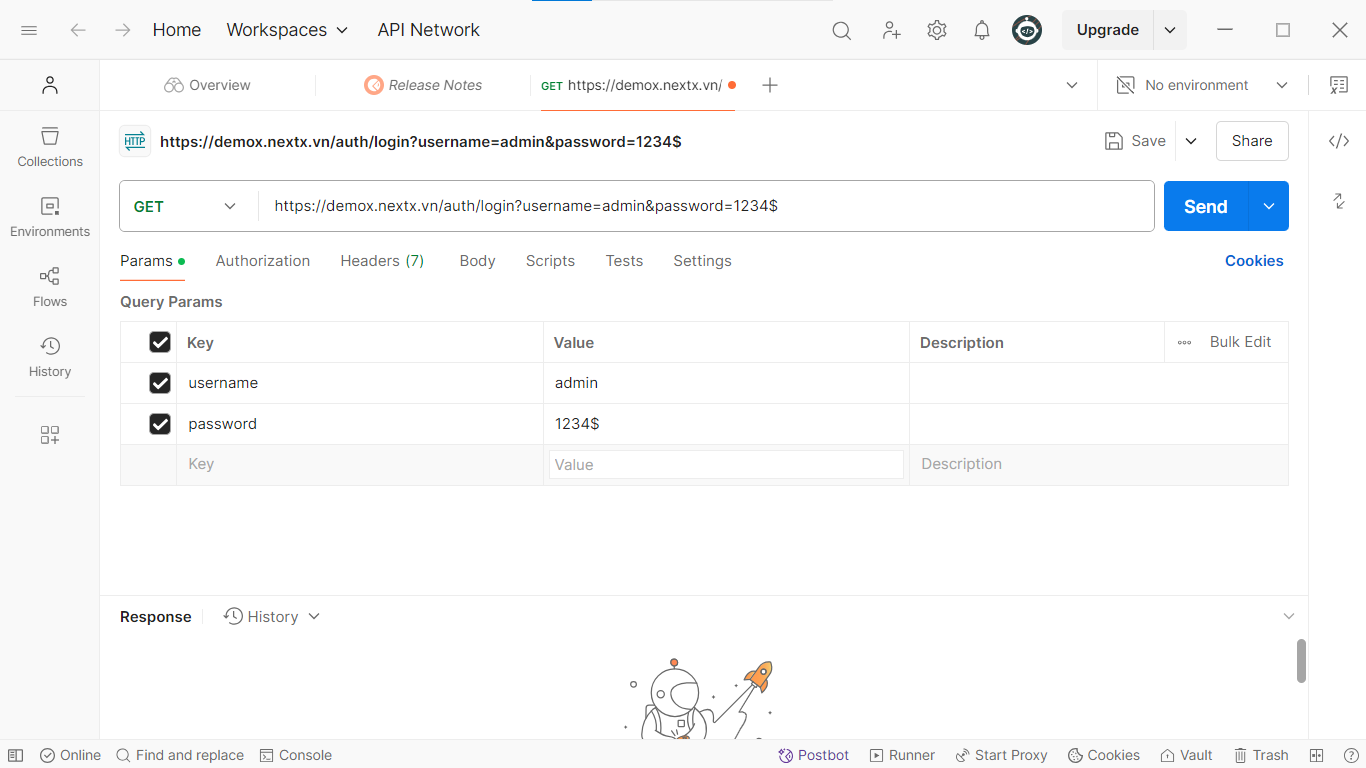
Để điền key, value trong parameter JMeter, sử dụng devtool tìm key như sau :

* Bước 1: Tại Hình II.12, click vào Element vị trí số (1), sau đó click vị trí số (2) chức năng này giúp nhìn rõ element nào tương ứng với nhau ( code và thành phần giao diện tương ứng).
* Bước 2: Trên giao diện, trỏ chuột vào vị trí cần tìm key(3).
* Bước 3: Phân tích thành phần code(4). Thuộc tính id được tham chiếu đến javascript/css cung cấp 1 định danh duy nhất cho 1 phần tử thẻ input trong html. Dữ liệu gửi đến sever có khoá là “username” và “password”, giá trị 2 key lần lượt là“admin ” và “1234$” (key - value)



Hình II.12. Dùng devtools tìm kiếm key.

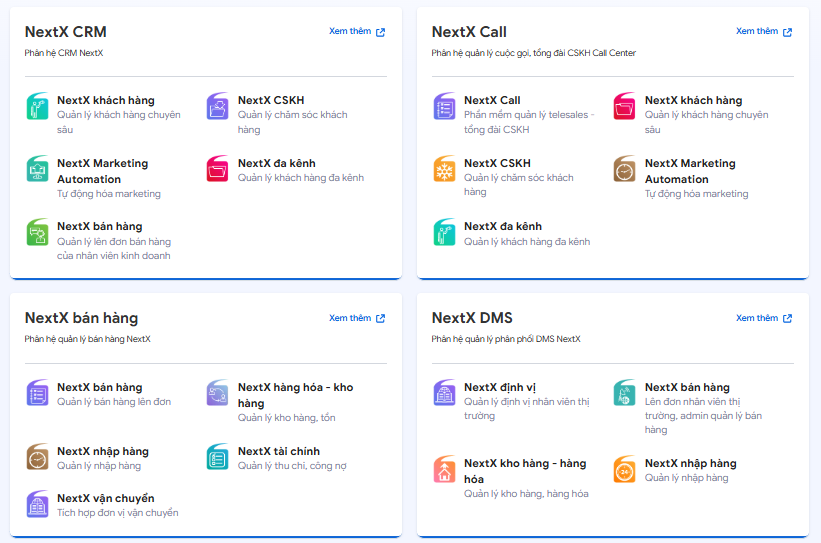
Sau đó sử dụng postman để kiểm tra có đúng key – value và request method không:



Hình II.13. Dùng postman kiểm tra key-value.

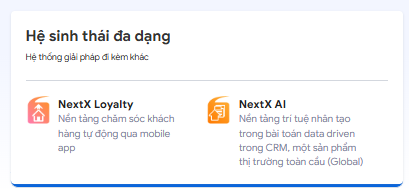
* Ngoài ra, devtool còn được sử dụng để theo dõi các request khác khi thao tác web, hỗ trợ giả lập request người dùng trên JMeter.

## Về website cần kiểm thử.

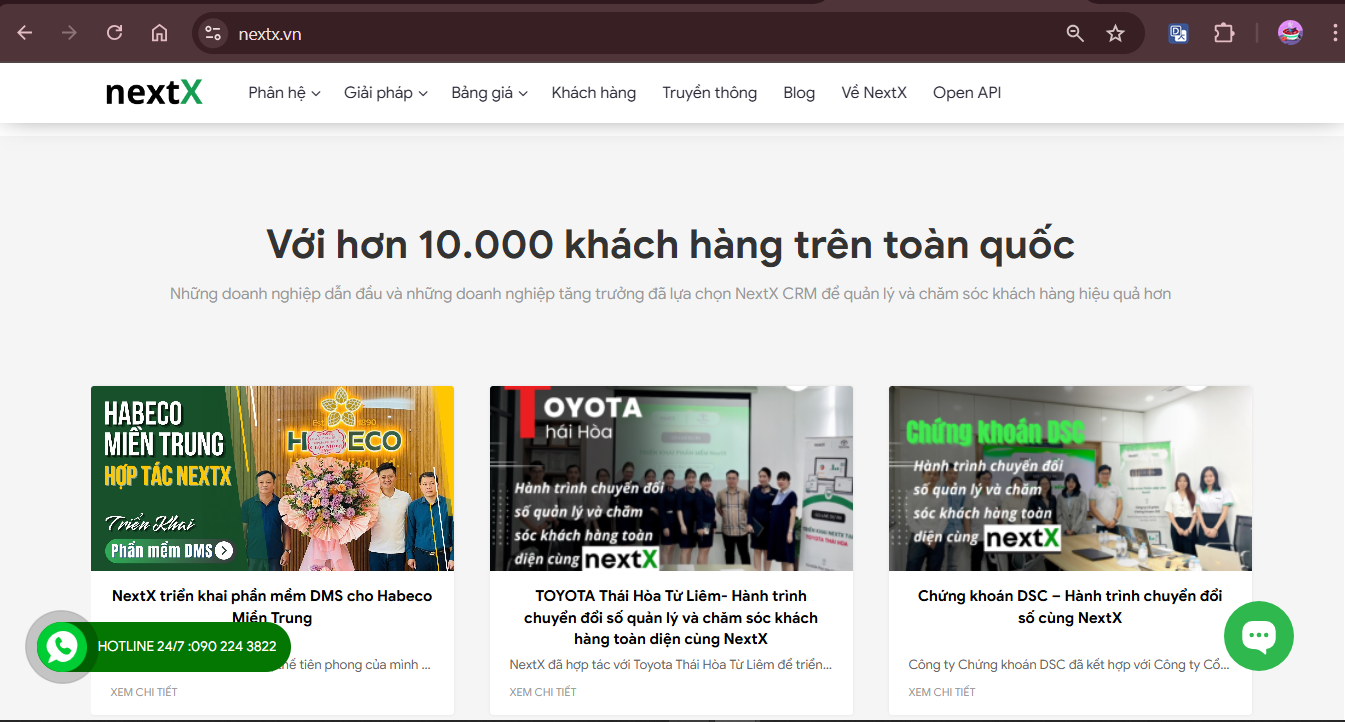


Hình II.14. Hệ và phân hệ web NextX

Qua trang khách của NextX có thể thấy công ty đã xây dựng một hệ sinh thái vô cùng đa dạng, đầy đủ chức năng, dịch vụ.



Hình II.15. Hệ sinh thái đa dạng.

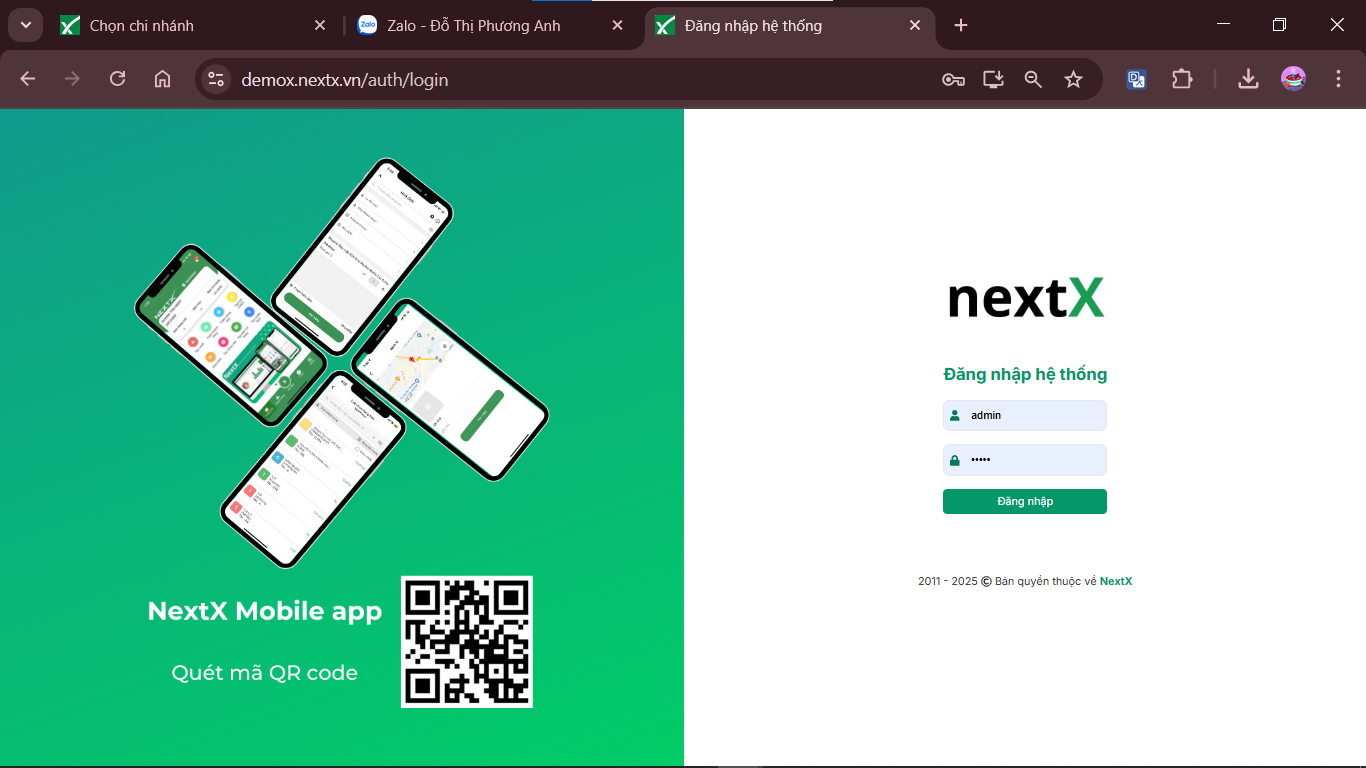


Hình II.16. Hệ thống khách hàng lên đến hơn 10.000 người dùng.

* Dựa vào hệ thống người dùng, mục đích sử dụng, lên kế hoạch test số lượng user trong testcase.
* Đặt ra giả thiết: Nếu tính 10000 người đều dùng hệ thống một lần trong 1 ngày(tính cả giờ hành chính, ca tối, ca đêm) thì trung bình có khoảng 7 user/phút. Nếu mỗi người gửi 10 request(đăng nhập, chọn chi nhánh, chọn phân hệ, tìm, lọc, thêm, sửa, chuyển trang,…) thì trung bình có khoảng 70 request/ phút. -> Giả thiết này đặt ra để dễ dàng định hướng testcase và hình dung thực tế yêu cầu của testcase. Giải thích cho câu hỏi tại sao có 10000 user mà mới có 2000 request đã có error, đó là bởi vì 2000request/5s.

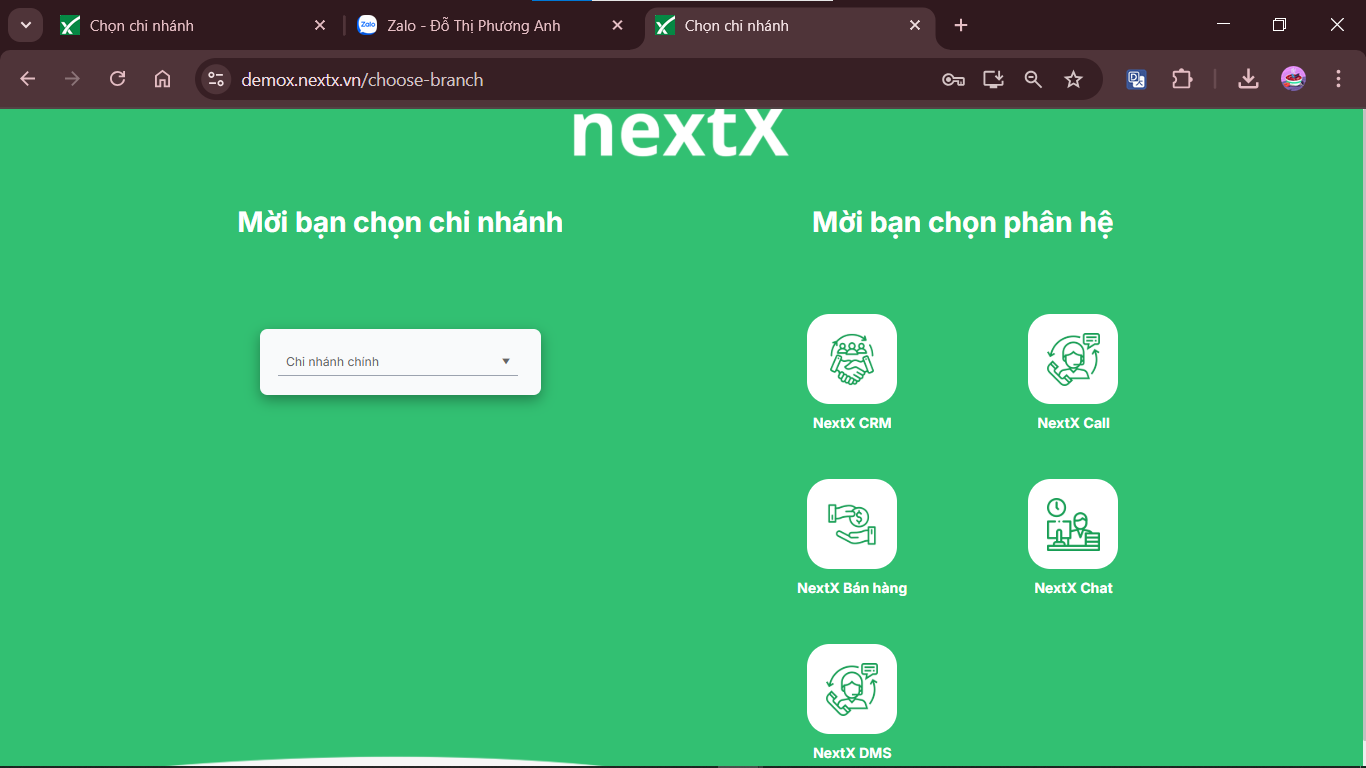
### Giao diện web.

Link web : <https://demox.nextx.vn/auth/login>



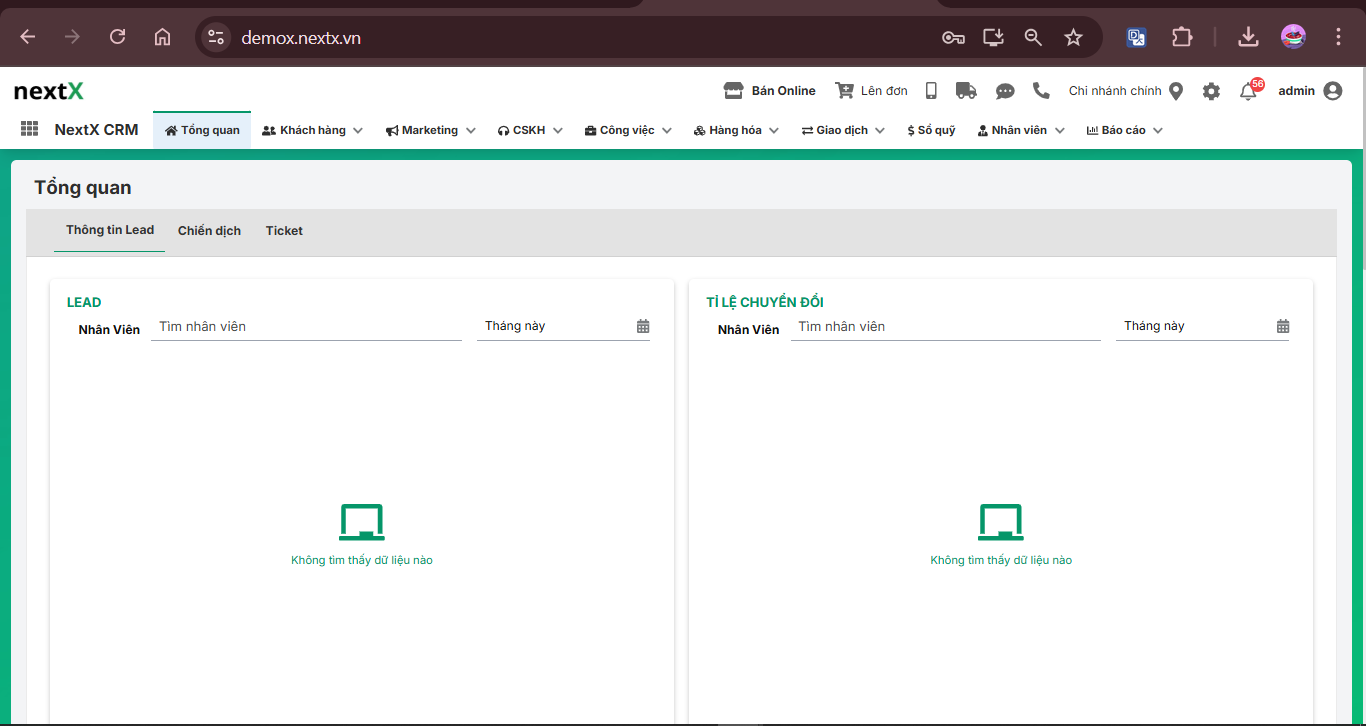
Hình II.17. Giao diện đăng nhập.

* Giao diện đăng nhập bao gồm tên đăng nhập và mật khẩu.



Hình II.18. Giao diện chọn Chi nhánh và Phân hệ.

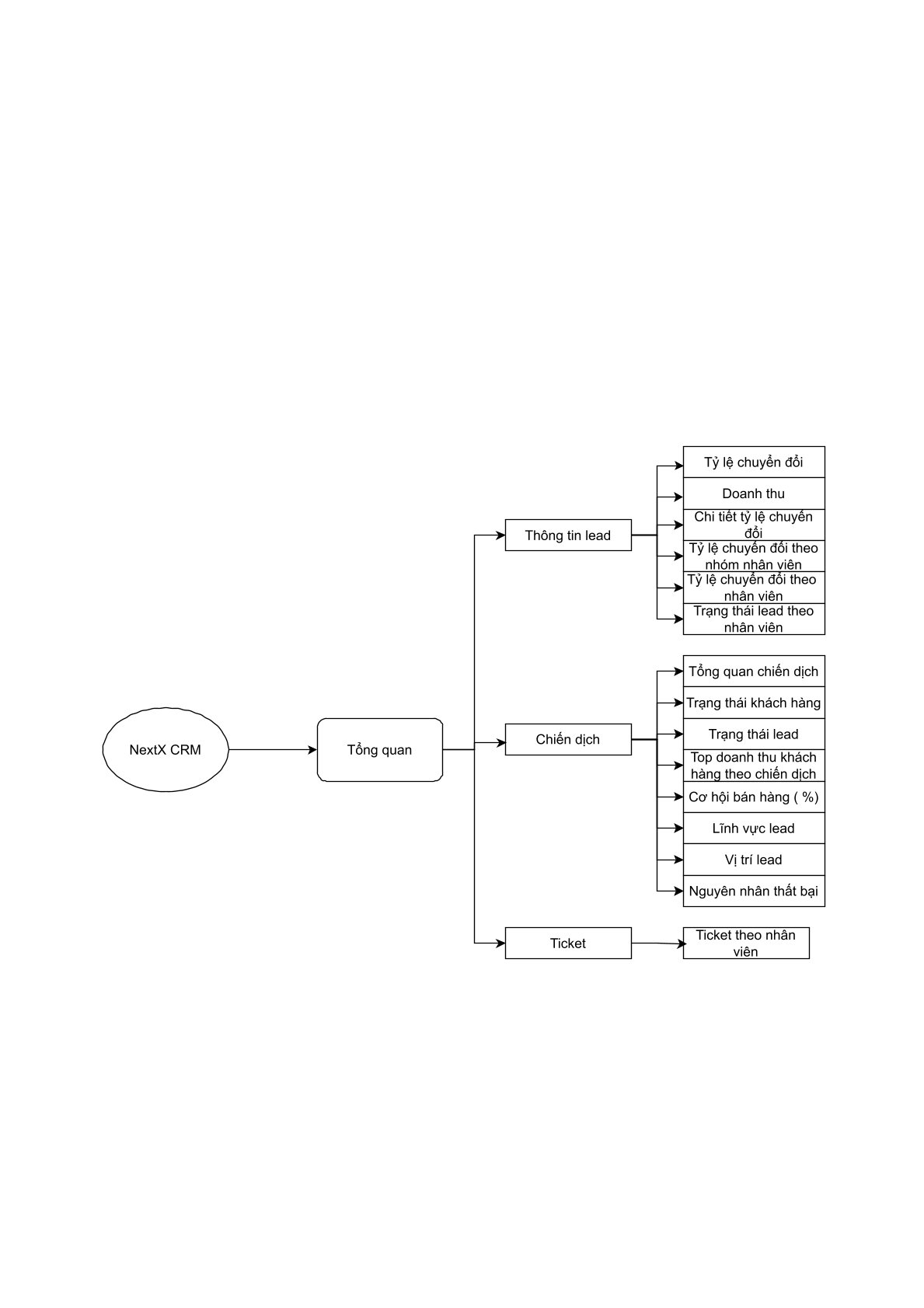
* Giao diện sau khi đăng nhập là 2 chức năng chọn chi nhánh và phân hệ. Bởi vì chi nhánh chính có nhiều dữ liệu đầy đủ hơn tiện cho công việc test nên em sẽ test tại chi nhánh chính. Và account được giao test chỉ duy nhất 1 account admin. Phân hệ NextCRM là phân hệ đa dạng và đầy đủ chức năng quan trọng nên em xin phép rút ngắn phạm vi test cho phù hợp với thời gian đồ án.
* Có 5 hệ chính: NextX CRM(CRM : Client relationship management), NextX Call, NextX Bán hàng, NextX Chat, NextX DMS(Distribute management system- hệ thống quản lý kênh phân phối).



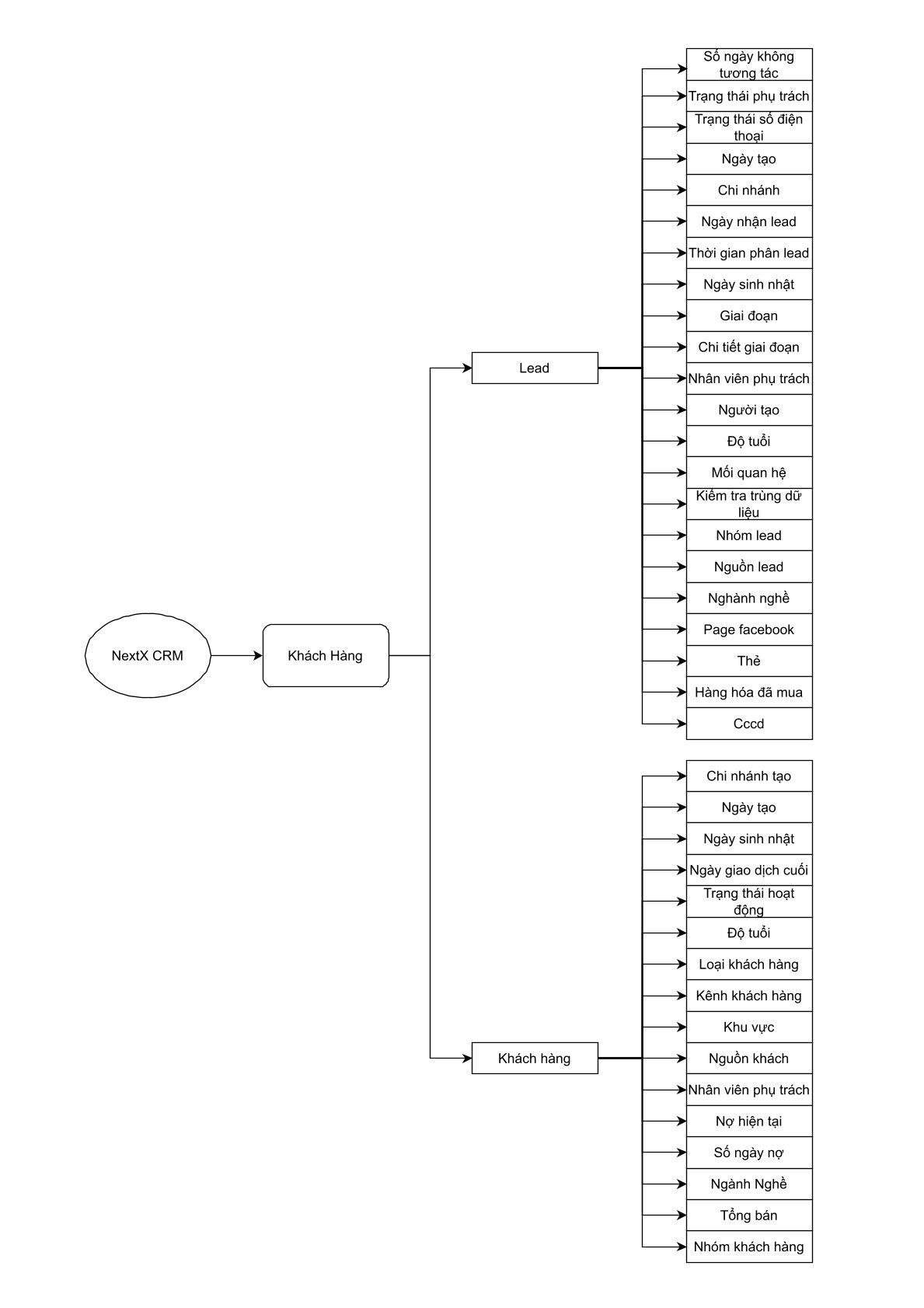
Hình II.19. Giao diện Tổng quan.

### Các chức năng chính.

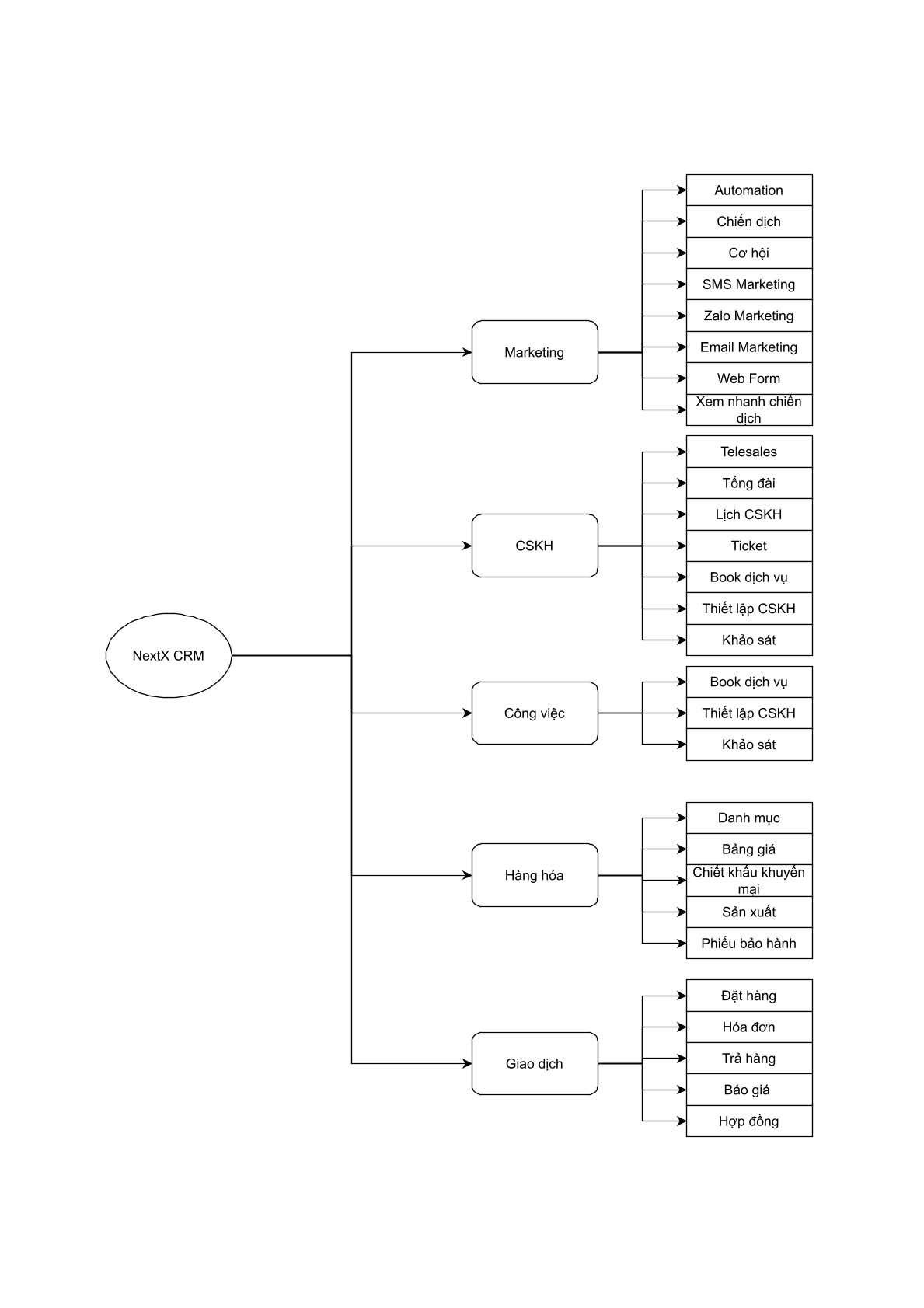
Trong website có 5 hệ, mỗi hệ có hàng trăm chức năng lớn nhỏ, dưới đây là các chức năng chính trong hệ NextX CRM, tài khoản admin, chi nhánh chính:



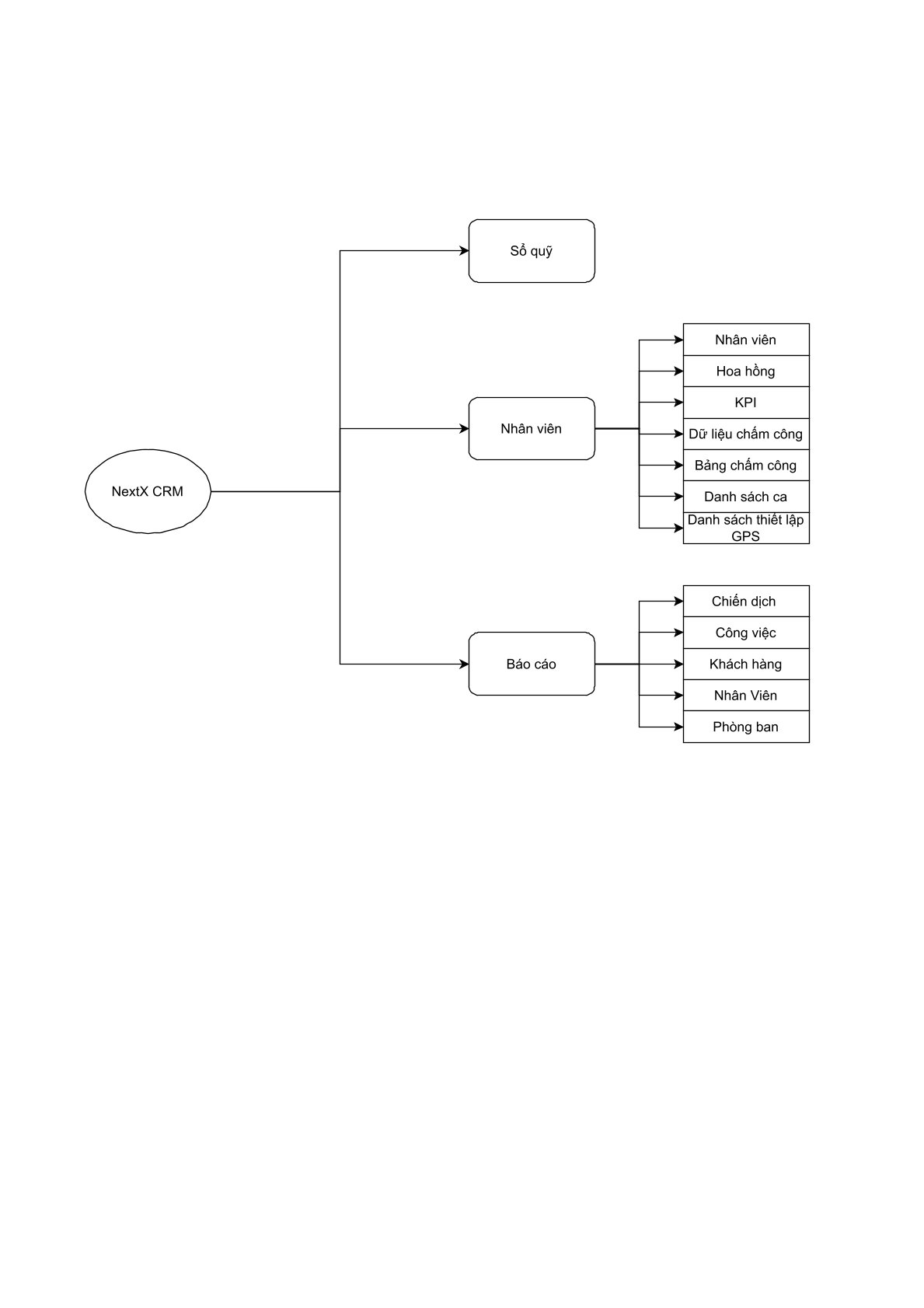
Hình II.20. NextCRM: chức năng Tổng quan.



Hình II.21. NextCRM: chức năng Khách hàng và nhánh các chức năng nhỏ.



Hình II.22. NextCRM: Một số chức năng khác.

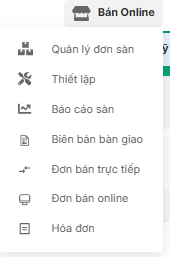


Hình II.23. NextCRM: Chức năng quản lý Sổ quỹ, Nhân viên, Báo cáo.

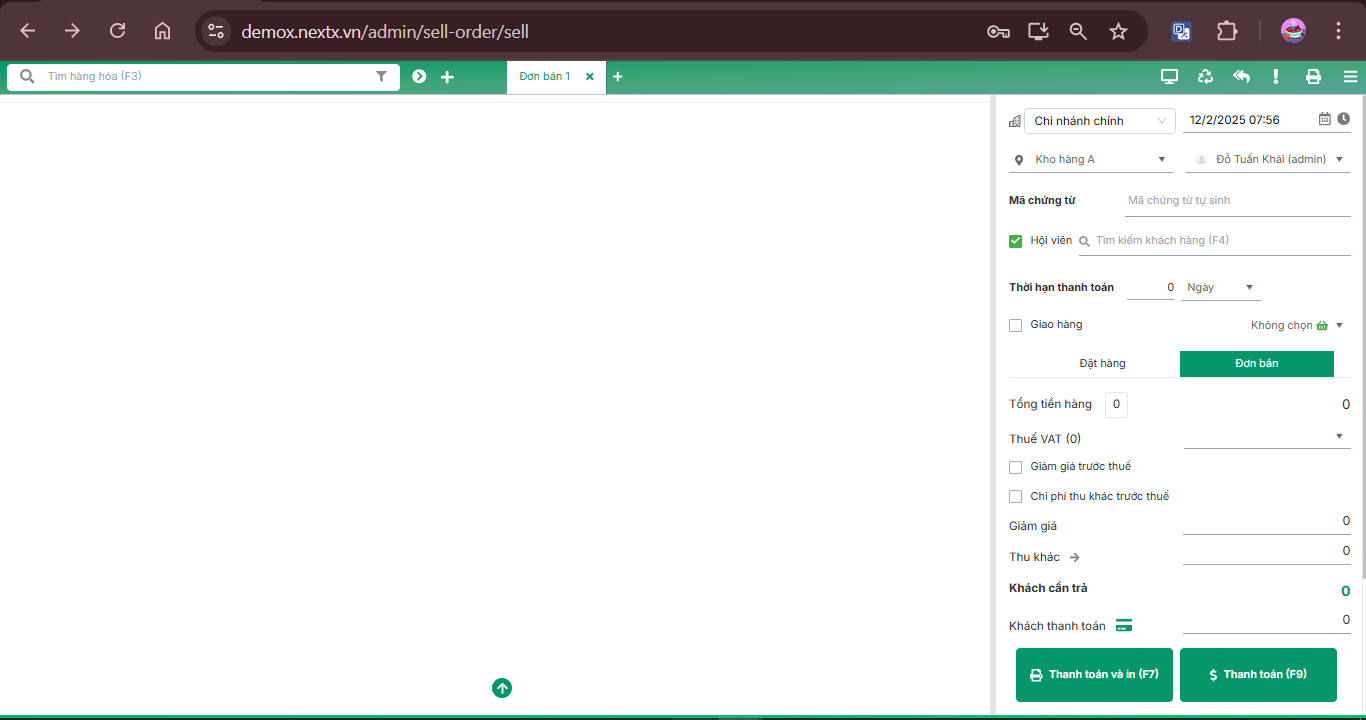
* Ngoài ra còn có các chức năng đăng nhập, đăng xuất, bộ lọc, tìm kiếm, nhập xuất dữ liệu. Còn các chức năng nhỏ như in, huỷ phiếu, mở rộng thông tin hiển thị,… sẽ được liệt kê chi tiết hơn ở testcase.
* Các chức năng tự động tạo biểu đồ, tính tổng dư nợ, thu chi,…
* Các chức năng : cài đặt, hòm thư, trực tiếp quay số, …



Hình II.24. Các chức năng khác:Bán online, Lên đơn, Hỗ trợ từ ban quản lý…



Hình II.25. Chức năng nhỏ hơn trong Bán online.



Hình II.26. Chức năng lên đơn cùng hàng chục chức năng hỗ trợ từng nhu cầu nhỏ.

# THỰC HIỆN KIỂM THỬ

## Đánh giá tính quan trọng của từng chức năng.

### Đánh giá các tiêu chí.

Bằng cách đánh giá xem chức năng nào dễ xảy ra lỗi trong quá trình vận hành hoặc dễ bị kẻ xấu lợi dụng tấn công giúp triển khai testcase nhanh và có ý nghĩa hơn, không bị lan man.

Đánh giá công việc cần làm trong testcase với các tiêu chí sau:

* Lượng người dùng đồng thời nhiều
* Các chức năng tốn ít thời gian thao tác hoặc có thể bị treo ở thao tác nào đó.
* Lượng data lớn.
* Chức năng mà có thể tăng đột biến lượng truy cập, lượng data( ví dụ : đặt hàng, sự kiện…)
* Chức năng này phù hợp với loại test hiệu năng nào, mục đích tìm ra lỗi, sử dụng kỹ thuật kiểm thử nào.
* Tìm ra kỹ thuật test phù hợp chức năng web.

Bởi vì các chức năng có thể trùng lặp trong các hệ và phân hệ nên quyết định test chức năng đại diện được đưa ra để phù hợp với thời gian đồ án và tránh trùng lặp gây thừa thãi, việc test các chức năng khác nhau cũng giúp em tích luỹ thêm kinh nghiệm.

Từ những tiêu chí đó, lọc được các chức năng có khả năng gây lỗi cao đối với một nền tảng CRM có nhiều phân hệ như quản lý khách hàng, bán hàng, tổng đài, quản lý công việc…, các chức năng quan trọng và dễ gặp lỗi về chịu tải, hiệu năng.

### Đánh giá mức độ quan trọng của các nhóm chức năng.

Đăng nhập vì đây là chức năng có thời gian thao tác rất ngắn, liên quan tới bảo mật. Đây là điểm vào của toàn bộ hệ thống. Nếu quá tải ở đây, toàn bộ hệ thống sẽ bị ảnh hưởng. Các vấn đề thường gặp:

* Thời gian phản hồi chậm khi xác thực tài khoản.
* Lỗi quá tải API login khi có quá nhiều yêu cầu trong thời gian ngắn.
* Cơ chế token không tối ưu, gây tình trạng đăng xuất bất ngờ hoặc quá tải bộ nhớ.

Quản lý quan hệ khách hàng (CRM) – Tìm kiếm và Lọc dữ liệu: Hệ thống CRM thường chứa lượng dữ liệu lớn (hàng trăm nghìn đến hàng triệu bản ghi). Khi nhiều người dùng tìm kiếm, lọc khách hàng theo điều kiện phức tạp, có thể gây:

* Truy vấn SQL chậm nếu không có index phù hợp.
* Timeout API khi truy vấn dữ liệu lớn không được tối ưu.
* Lỗi UI không phản hồi nếu client-side không xử lý tốt.

Chức năng Báo cáo & Dashboard (ví dụ **Xuất/Nhập Excel)**: Dashboard tổng hợp dữ liệu từ nhiều nguồn, yêu cầu tính toán tổng doanh thu, số lượng khách hàng, hiệu suất nhân viên. Nếu dữ liệu lớn, có thể gây chậm hoặc treo server. Các lỗi thường gặp:

* Truy vấn nặng trên database gây chậm, treo hệ thống.
* Lỗi tải biểu đồ và dữ liệu không hiển thị do API trả về chậm hoặc timeout.

Tổng đài chăm sóc khách hàng (Zalo, SMS, Email): Hệ thống có tích hợp tổng đài dễ gặp lỗi khi có quá nhiều cuộc gọi đồng thời, gửi tin hàng loạt, dễ bị quá tải, timeout hoặc spam detection. Các lỗi có thể xảy ra:

* Tổng đài không kết nối do quá tải.
* Mất gói tin, độ trễ cao làm gián đoạn cuộc gọi.
* Token bị mất, khách hàng đang gọi nhưng hệ thống ghi nhận đã kết thúc.

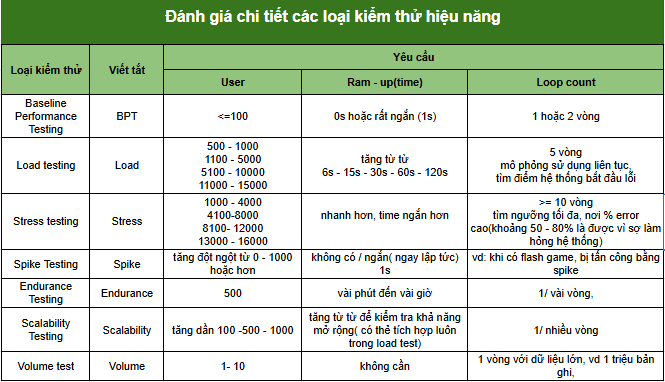
Giao dịch( Đặt hàng, Hóa đơn, Trả hàng): có hàng nghìn khách hàng và mỗi khách hàng có thể có 1 hoặc nhiều đơn hàng, vì vậy thanh toán có khả năng gặp quá tải. Đây là các nghiệp vụ cốt lõi của hệ thống. Nếu bị chậm hoặc lỗi, doanh nghiệp mất doanh thu.

Kết luận: Có thể tập trung vào các bài kiểm thử chịu tải (Load Test, Stress Test) với JMeter:

* Đăng nhập: Kiểm tra số lượng người dùng đăng nhập đồng thời mà hệ thống chịu được.
* Tìm kiếm – lọc dữ liệu: Đánh giá thời gian phản hồi khi thực hiện tìm kiếm với số lượng lớn.
* Báo cáo/xuất nhập excel: Chạy kiểm thử với nhiều người dùng truy vấn báo cáo đồng thời.
* Tổng đài( cuộc gọi, sms, …): Nếu có API, có thể mô phỏng nhiều cuộc gọi đồng thời.
* Giao dịch( đặt hàng).

### Đánh giá chi tiết các loại kiểm thử hiệu năng.

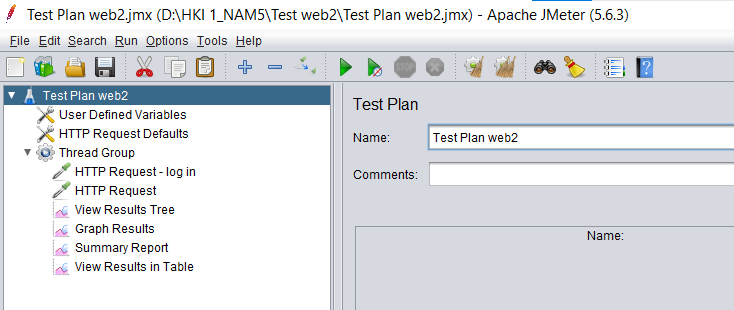
Phần này đánh giá các chỉ số nào quan trọng với loại kiểm thử hiệu năng tương ứng, cụ thể hoá mục đích sử dụng loại kiểm thử. Từ đó mới có thể đánh giá được chức năng nào cần dùng loại kiểm thử nào ở phần testcase.



Bảng III.1. Đánh giá chi tiết các loại kiểm thử hiệu năng.

## Tạo Test plan và các thành phần cơ bản.

### Thêm Test Plan.

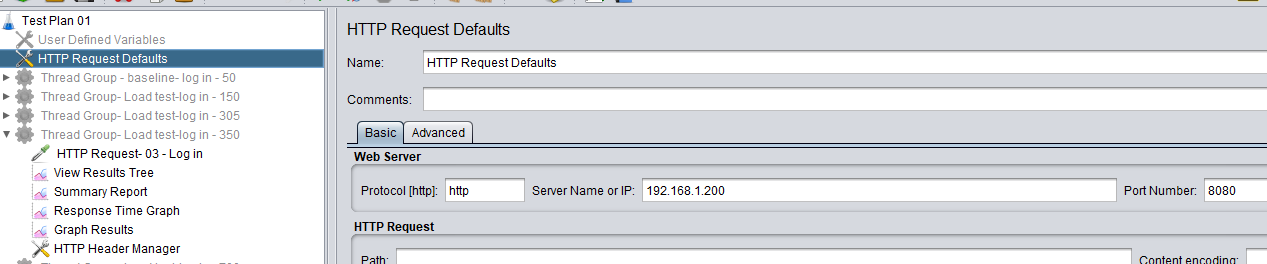


Hình III.1. Testplan trong JMeter.

* Test plan có thể đổi tên, thêm comments để ghi lại lưu ý, mục đích, công dụng của testplan.

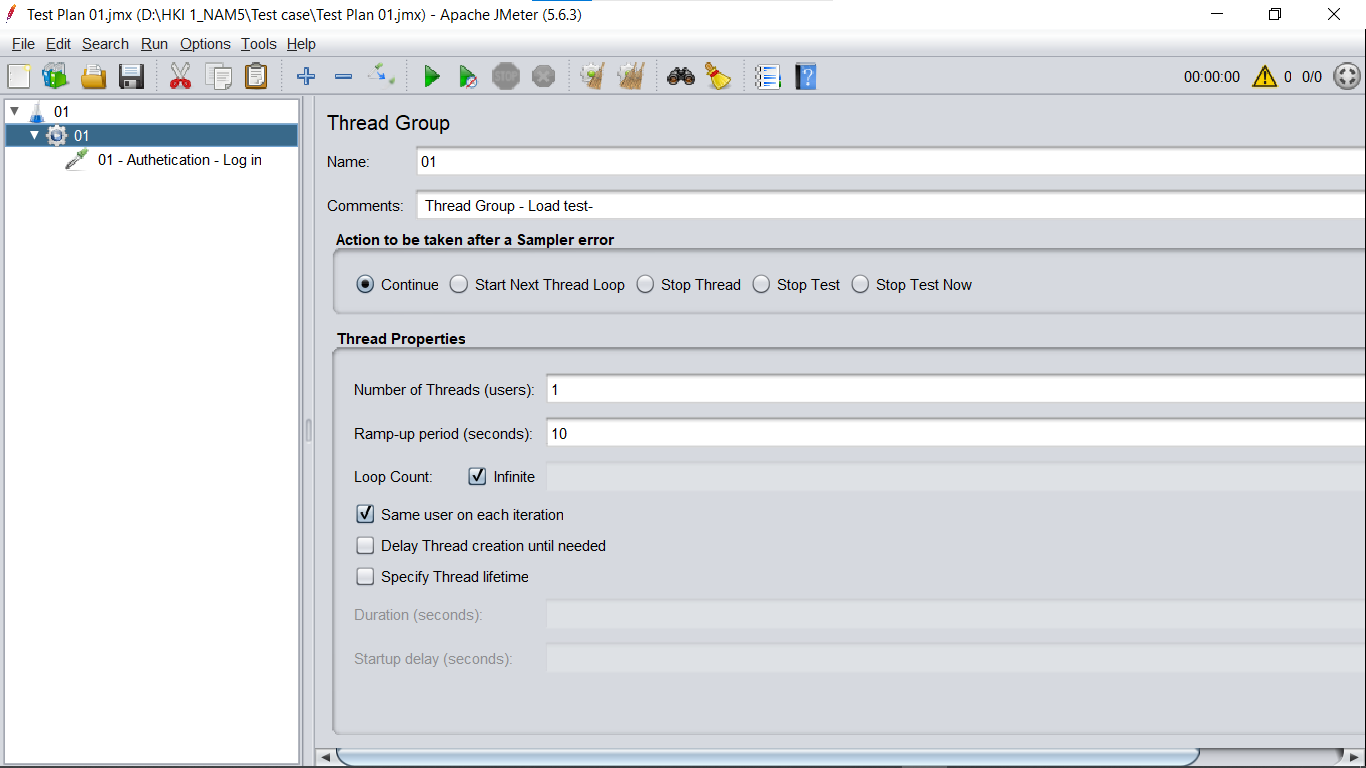
### Thêm Config element.

* Cách làm: Chuột phải tại testplan 🡪 Add 🡪 Config Elements
* Trong đó, được sử dụng để tối ưu testplan là : HTTP Request Defaults.



Hình III.2. HTTP Request default.

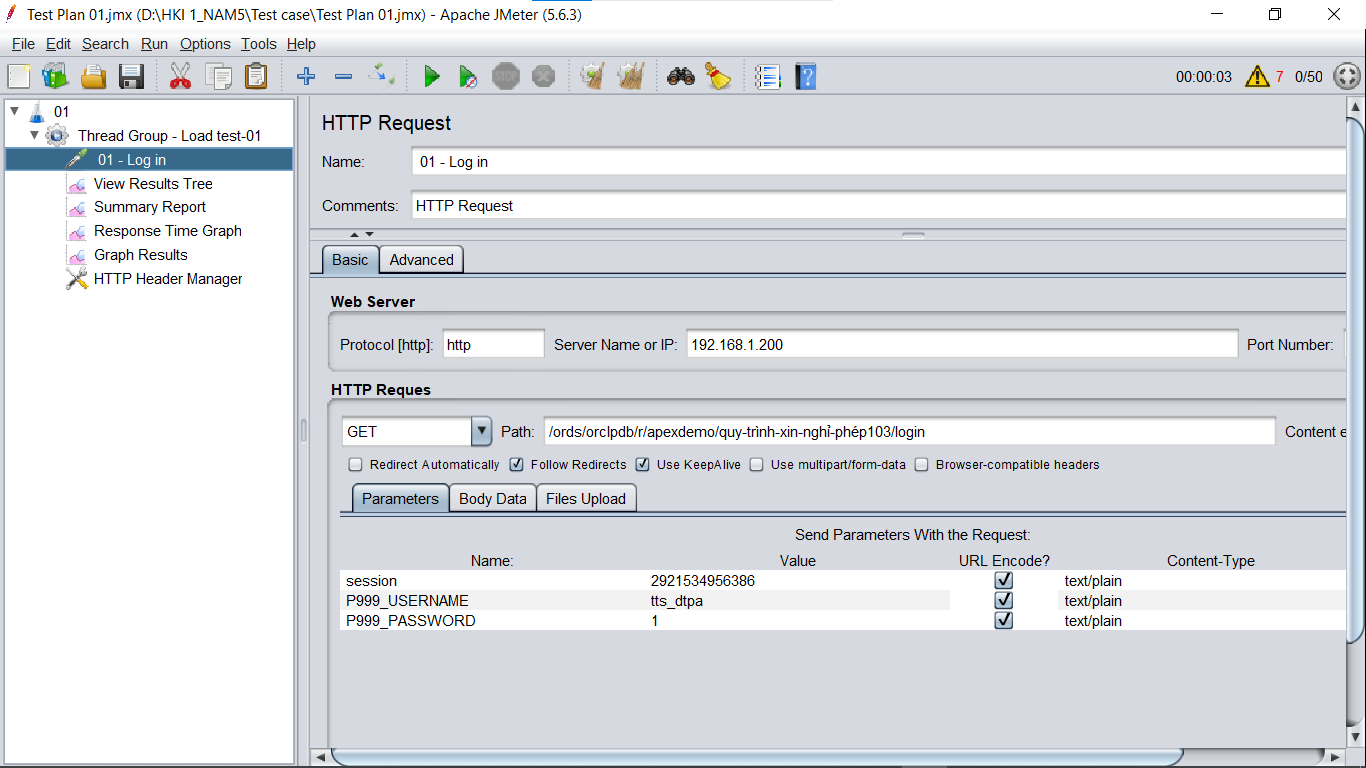
### Thêm Thread group.



Hình III.3. Thread Group nơi giả lập thao tác người dùng.

* Cách làm : Chuột phải tại test plan 🡪 add 🡪 Thread Group.

### Thêm HTTP Request.



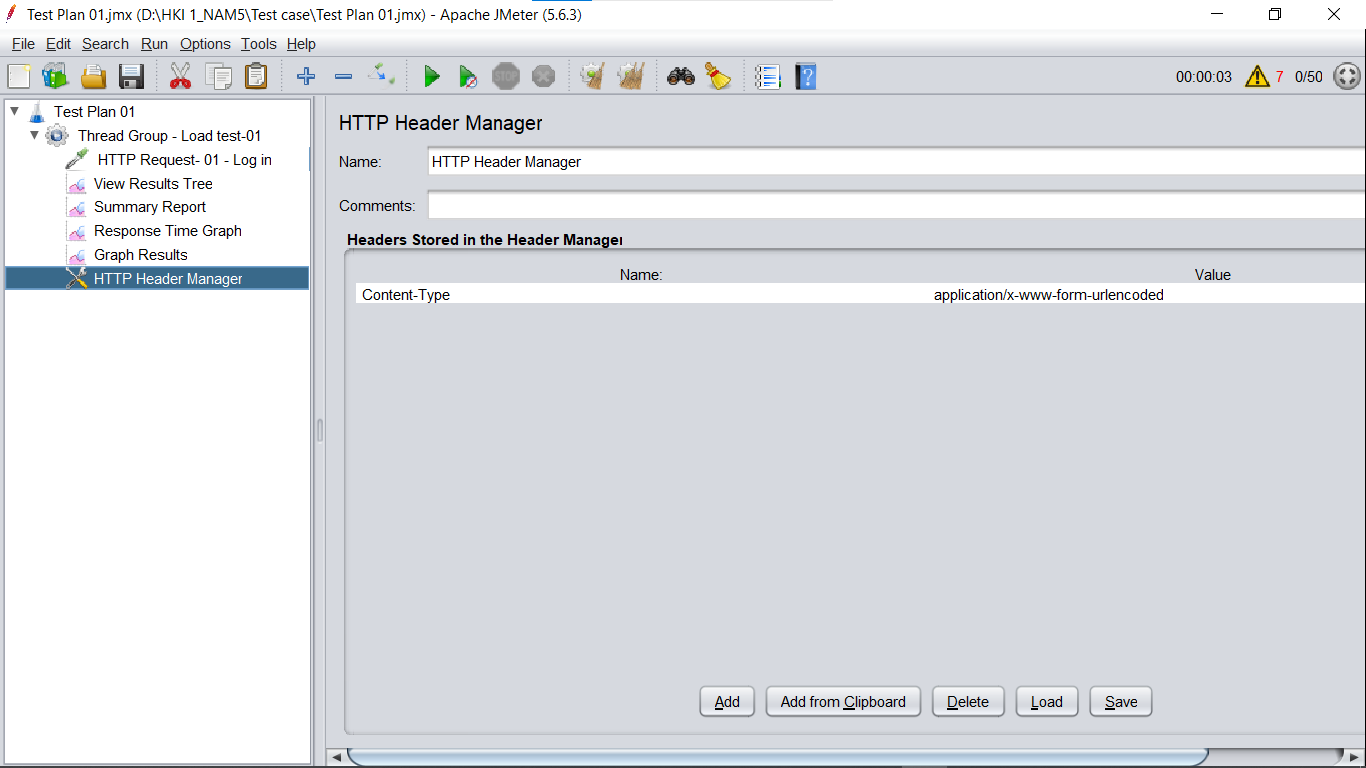
Hình III.4. HTTP Request nơi ghi các thông tin cần thiết để gửi request tới server.

### HTTP Header Manager.

* Là một thành phần trong JMeter dùng để thêm, sửa đổi hoặc gửi các tiêu đề (headers) trong HTTP request. Headers này thường chứa thông tin quan trọng để giúp server xử lý request đúng cách.
* Cách thêm: Chuột phải Thread group 01 🡪 Add 🡪 Config element 🡪 thêm HTTP Header Manager.

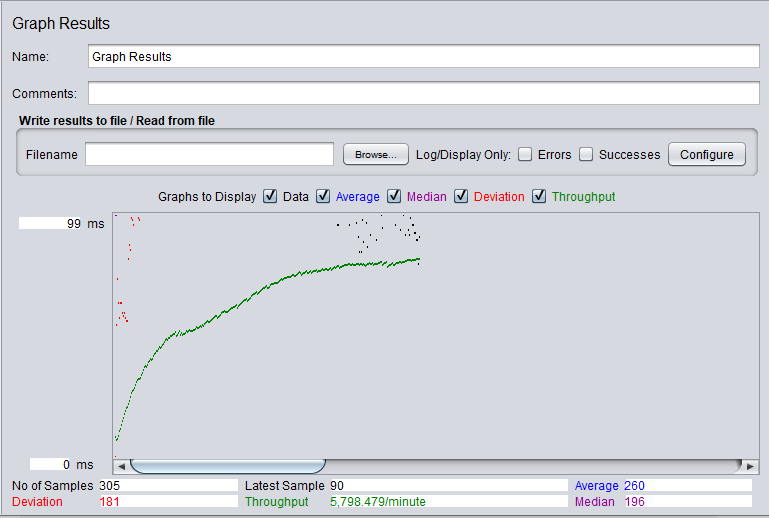
Lưu ý khi dùng:

* Dùng khi cần gửi API request có Authentication (JWT, OAuth, Basic Auth).
* Khi làm việc với REST API yêu cầu tiêu đề Content-Type và Accept.
* Khi cần giả lập trình duyệt để tránh bị chặn bởi server.
* Khi cần tắt cache để đảm bảo JMeter không lấy dữ liệu cũ.



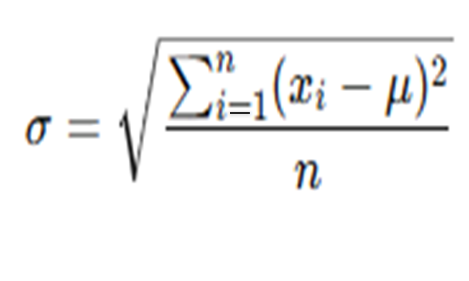
Hình III.5. HTTP Header Manager.

### Graph result.



Hình III.6. Graph Result nơi biểu diễn trực quan thông số phản hồi.

* Deviation : độ lệch chuẩn, đo lường sự **phân tán** của thời gian phản hồi của các yêu cầu so với giá trị trung bình. Công thức :



Hình III.7. Công thức tính Deviation độ lệch chuẩn thời gian phản hồi so với Avegare Response Time.

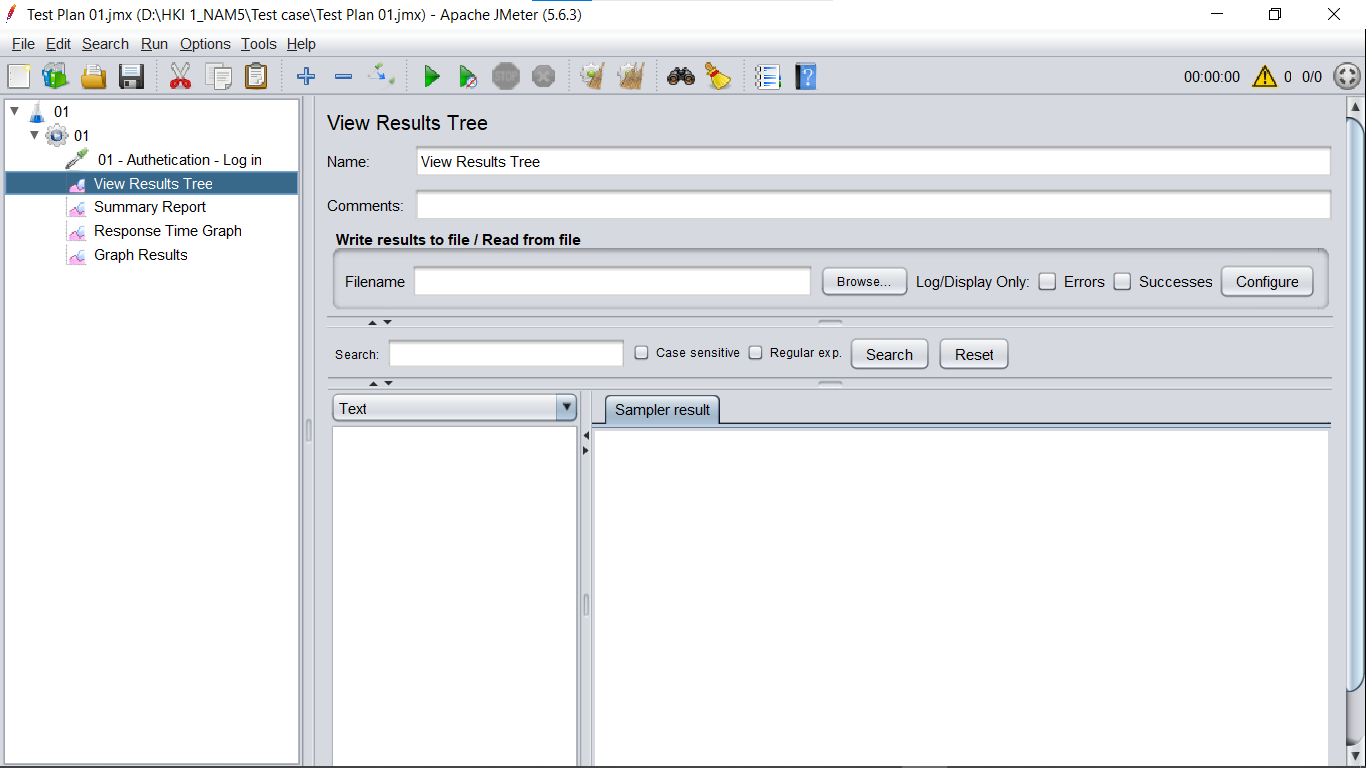
Trong đó:

* *σ* : Độ lệch chuẩn.
* *xi​:* Thời gian phản hồi của từng request.
* *μ*: Thời gian phản hồi trung bình (Average Response Time).
* *n*: Tổng số requests.
* Tổng từ *i=1* đến *n* mẫu thử.

### Thêm Listener để xem kết quả.

Để theo dõi và phân tích kết quả kiểm thử,cần thêm các Listener, chuột phải vào Thread Group → Add → Listener → Chọn:

* View Results Tree: Xem chi tiết từng HTTP Request (kết quả thành công/thất bại).
* Summary Report: Xem tổng quan hiệu suất (thời gian phản hồi, lỗi, v.v.).
* Graph Results: Biểu đồ kết quả kiểm thử theo thời gian.
* Response Time Graph: Phân tích thời gian phản hồi.

****

Hình III.8. View result tree nơi xem kết quả server trả cho request.

Kết quả trong Summary report:

****

Hình III.9. Summary Report.

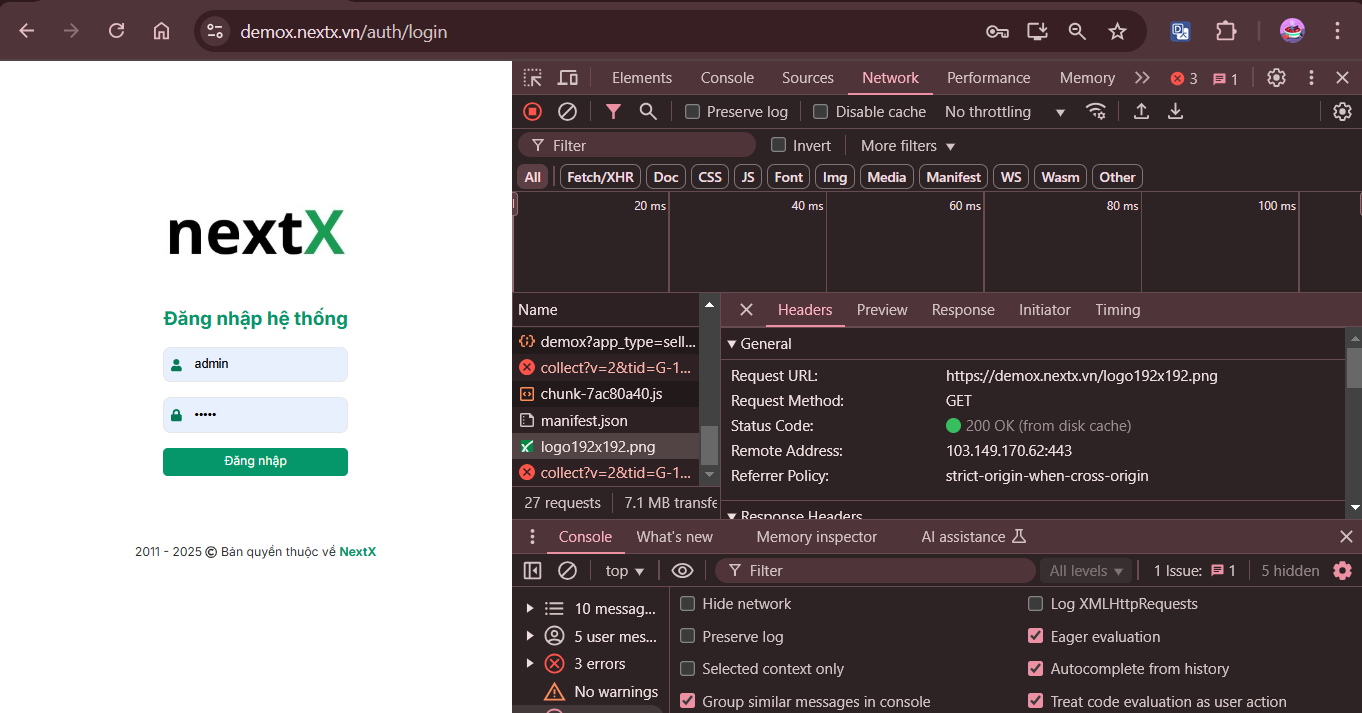
* Dùng để xem các thông số chi tiết của quá trình sent/received..

## Testcase.

Testcase sẽ ưu tiên thực hiện test các chức năng được đánh dấu mức độ quan trọng 3, 2 và các chức năng nào được đánh số ở cột ghi chú, module lặp chỉ cần test 1 lần duy nhất hoặc không cần thiết (tuỳ thuộc mức độ quan trọng) vì được đánh giá mức độ quan trọng thấp ít khả năng xảy ra vấn đề với hiệu suất.

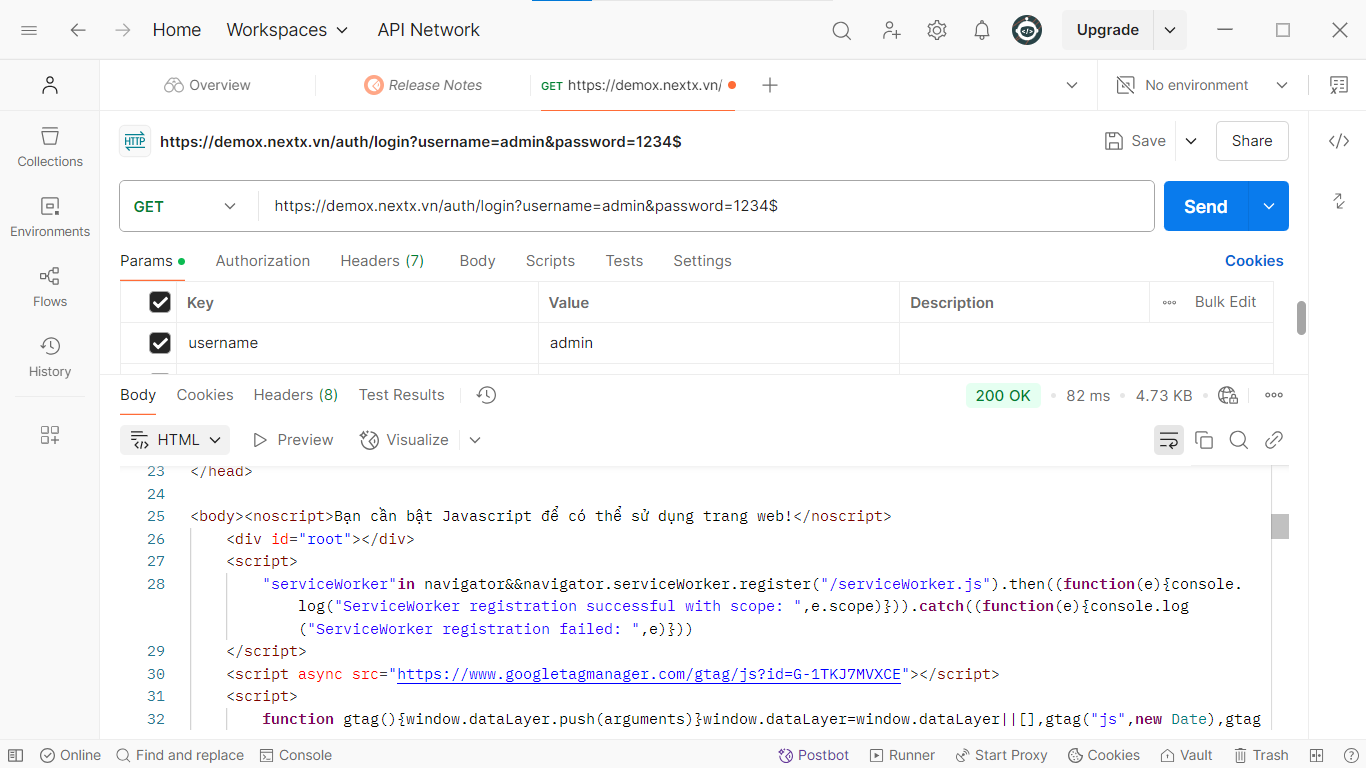
### Chức năng đăng nhập.

Bước 1: Xác nhận method request, thông tin đăng nhập.



Hình III.11. Lấy phương thức request là GET.

* Sử dụng Postman kiểm tra cặp key-value và phương thức sử dụng của web:



Hình III.12. Test login với postman

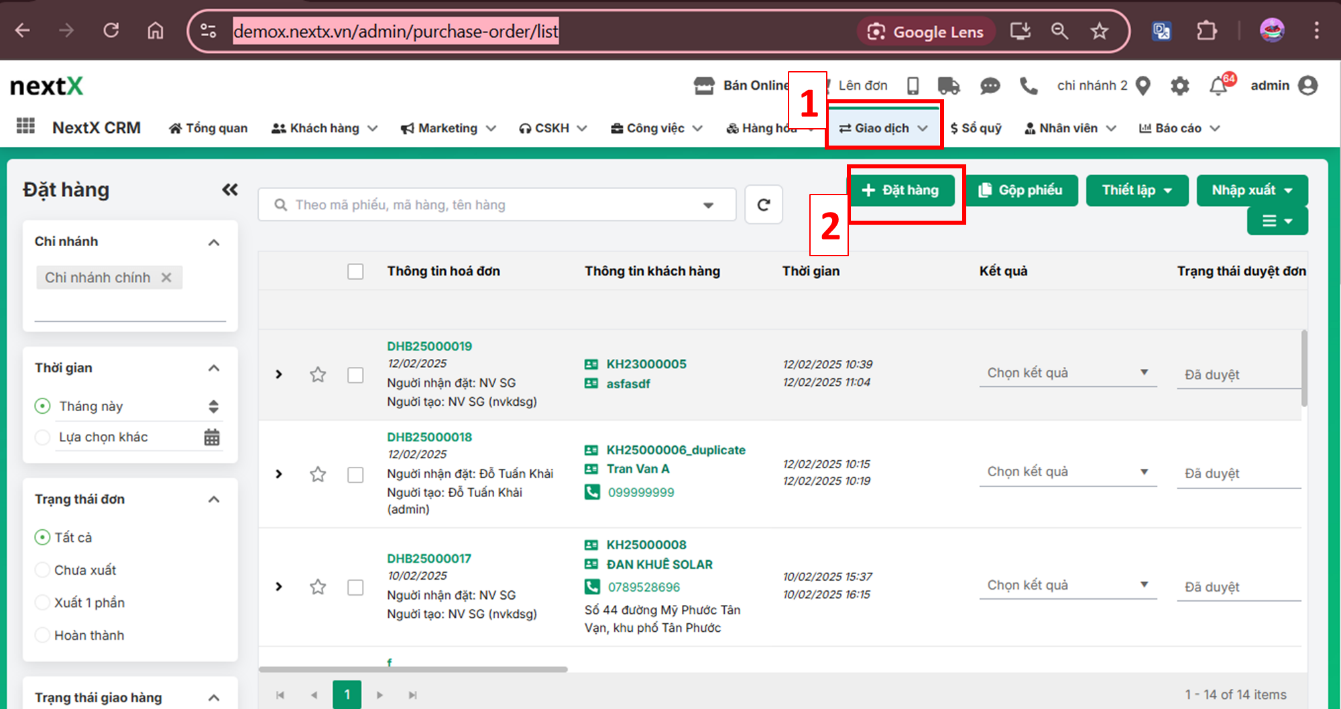
Kết luận key - value: username - password, phương thức là GET.

Bước 2: Chạy test case đăng nhập.

### Thêm đơn đặt hàng.

Giao dịch -> Đặt hàng -> Thêm đơn đặt hàng.

https://demox.nextx.vn/admin/purchase-order/list

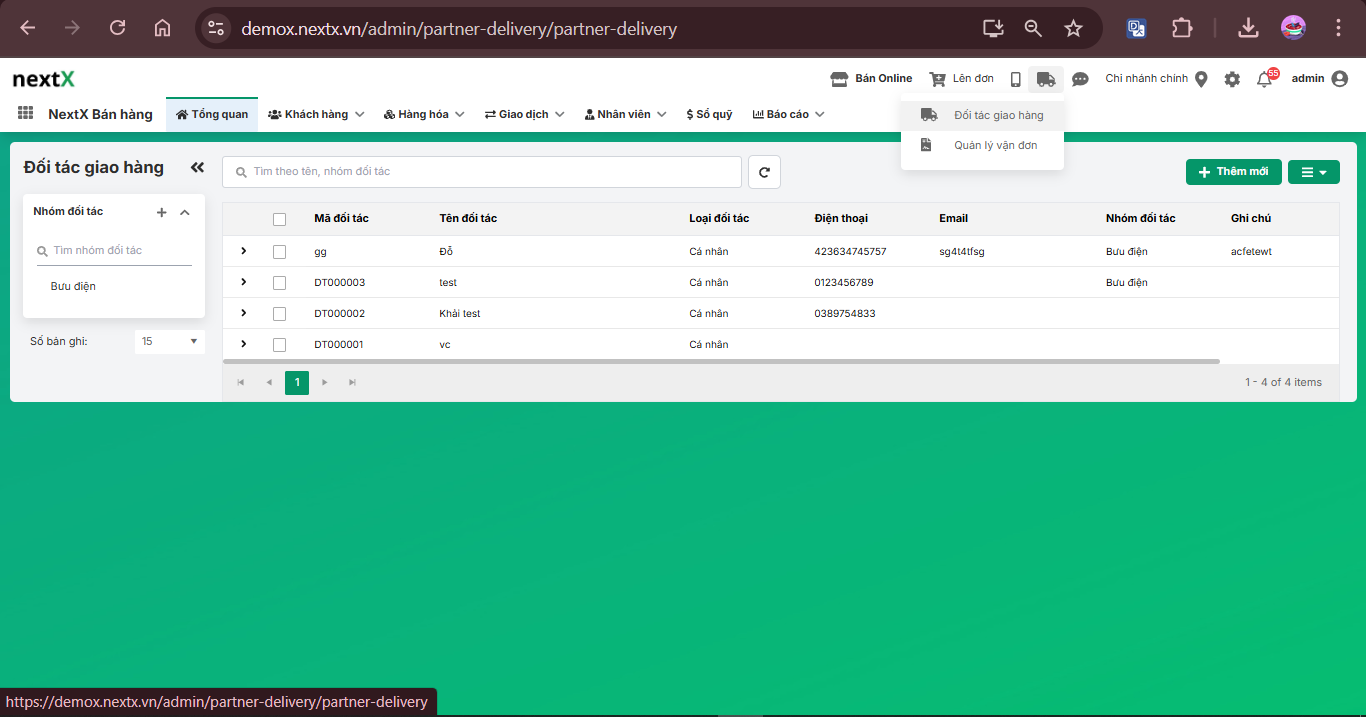


Hình III.10. Thêm đơn đặt hàng.

### Chức năng lên đơn.

https://demox.nextx.vn/admin/sell-order/sell

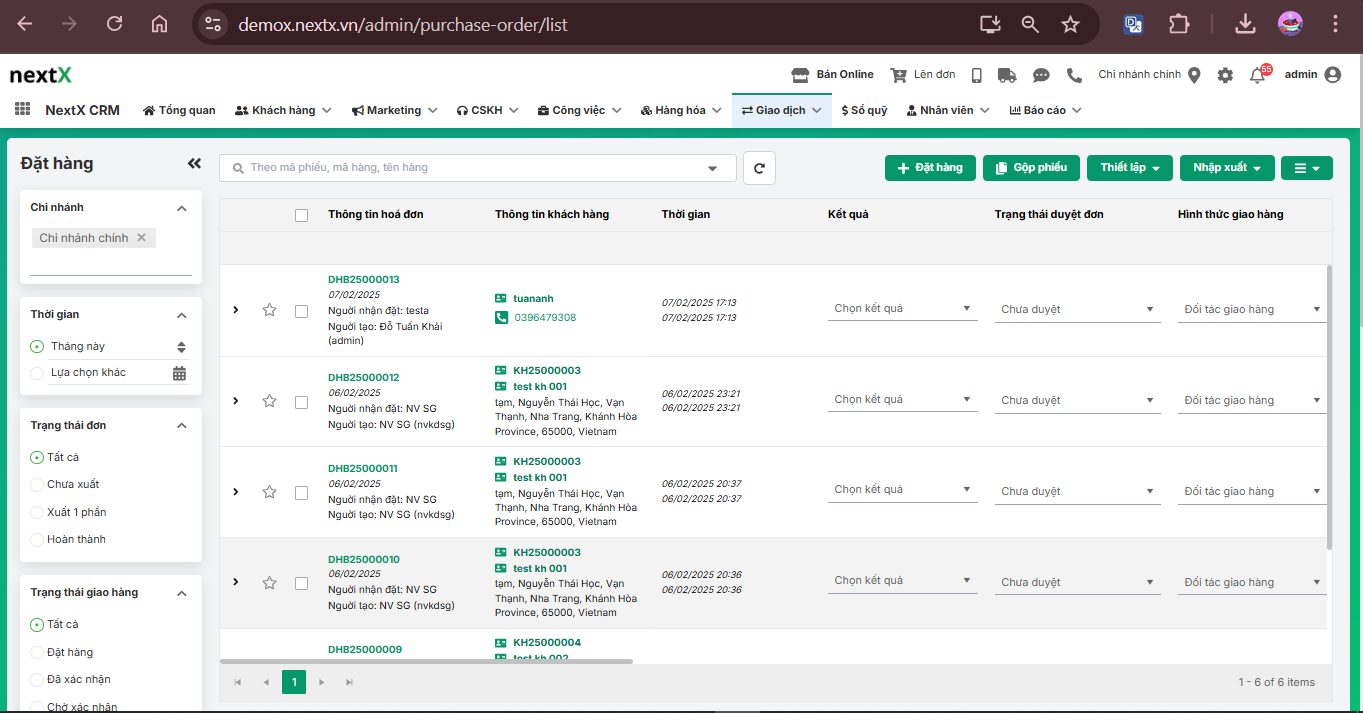
### Đối tác giao hàng.



### Đặt hàng.

https://demox.nextx.vn/admin/purchase-order/purchase

NextCRM : Giao dịch 🡪 Đặt hàng.



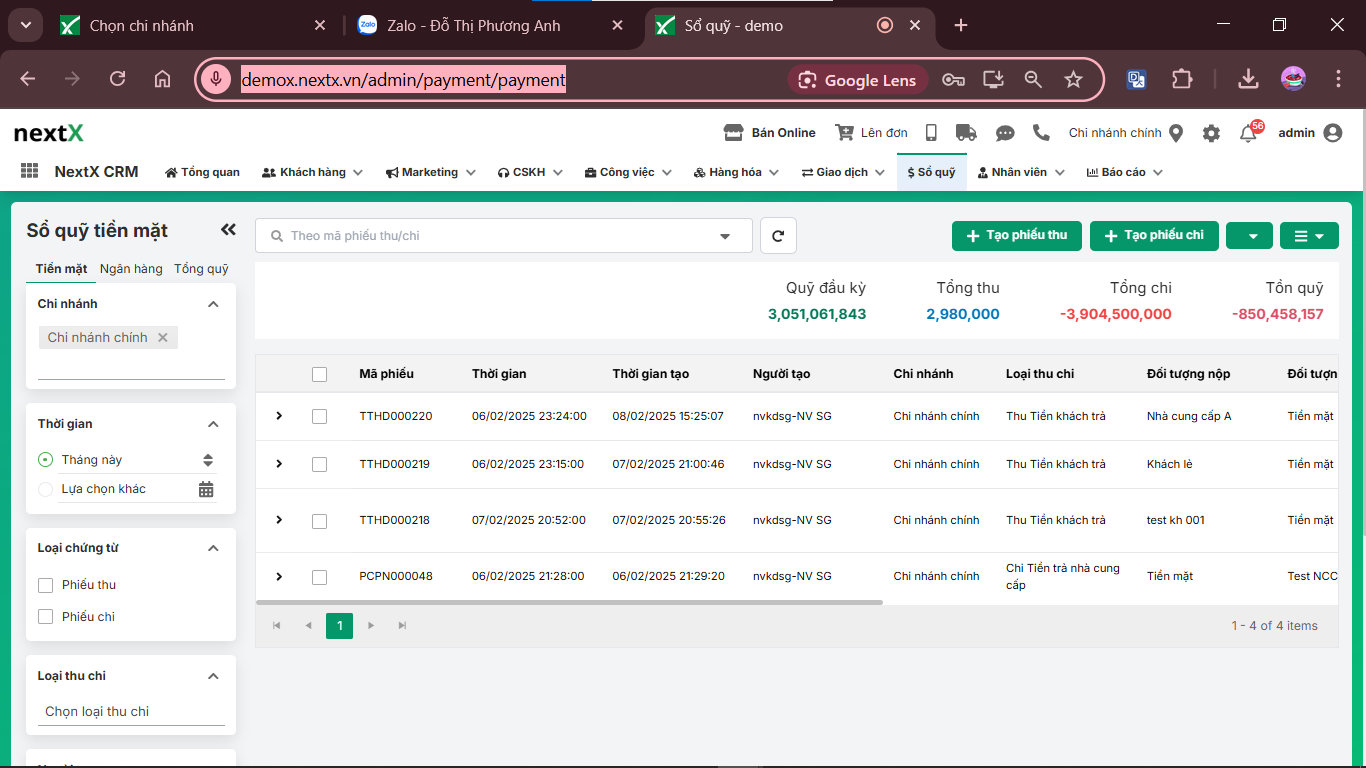
Hình III.13. Chức năng đặt hàng

BPT

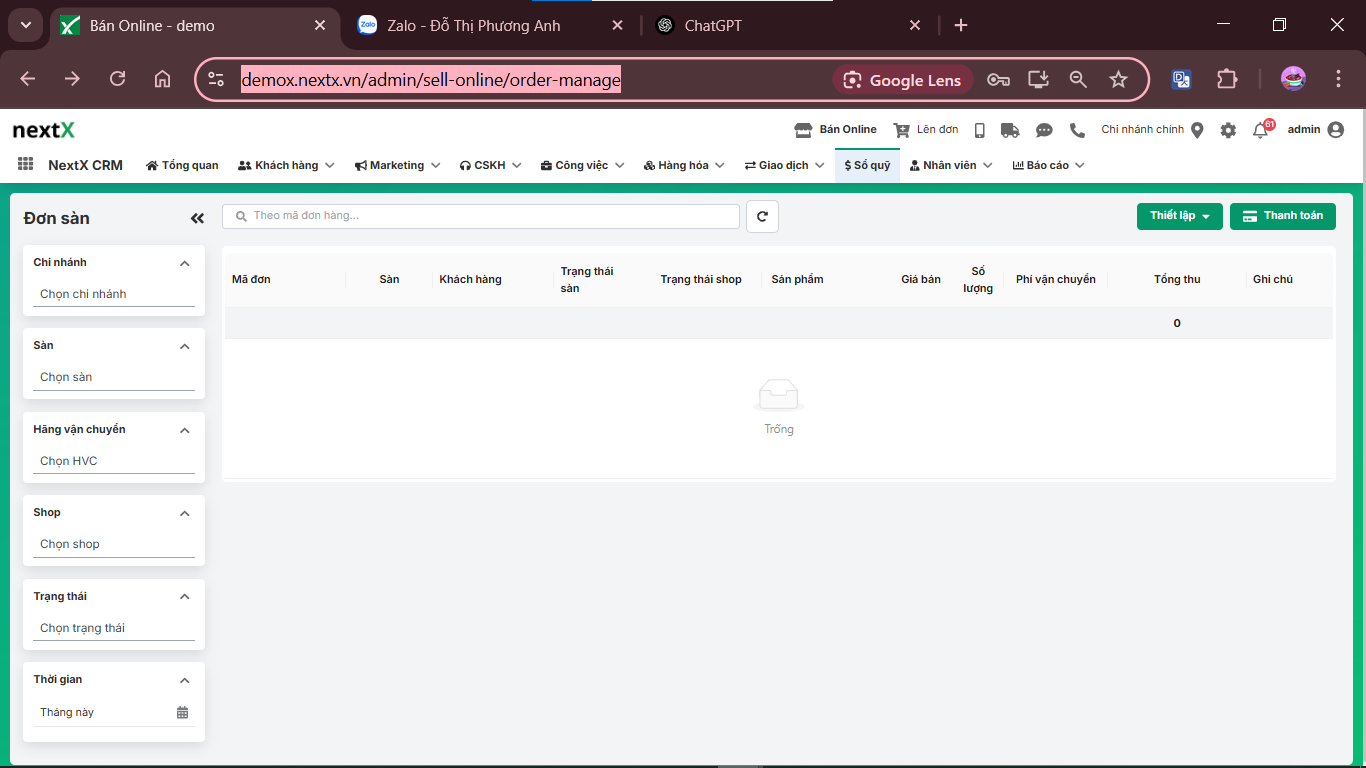
Load

### Tạo phiếu thu.

https://demox.nextx.vn/admin/payment/payment



https://demox.nextx.vn/admin/sell-online/order-manage



### Đánh giá mức độ nghiêm trọng của lỗi.

* Công việc này có thể thực hiện bởi tester hoặc người quản lý.
* Được thực hiện sau khi đã chuẩn bị xong các lệnh chạy và chạy thành công.

### Đề xuất khắc phục.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận.

Khó khăn

* Bình thường chỉ test UI UX và test data, lần đầu test hiệu năng nên chưa có kinh nghiệm, đặc biệt là về phần cứng, cách sửa lỗi, mất nhiều thời gian để sắp xếp, hoàn thiện chi tiết của testcase.
* Nguồn thông tin trên internet quá nhiều, cần rất cẩn thận trong việc lựa chọn nguồn tham khảo, sợ hiểu sai kiến thức, hiểu sai vấn đề. Hoặc kiến thức internet đem lại không đủ sâu, để hiểu tường tận và xử lý các thao tác đòi hỏi kỹ thuật khó.
* Khó để lấy số liệu chính xác vì mỗi lần test cũng với case giống nhau lại cho ra kết quả khác nhau, bởi còn phụ thuộc băng tần wifi, phụ thuộc khi đó có nhiều người dùng sử dụng tài nguyên hệ thống hay không.
* Khó để đưa ra kết luận kết quả test là thất bại hay thành công bởi cần so sánh với số liệu baseline, không có số liệu tuyệt đối mà phải dựa trên khoảng chấp nhân của người dùng.
* Khó để sửa lỗi khi mình không phải là người làm ra website.
* Chưa phân tích được mạng client có ảnh hưởng như nào tới việc JMeter gửi request.

Phương pháp khắc phục

* Sử dụng AI để giải đáp thắc mắc về các vấn đề kỹ thuật, các vấn đề về lý thuyết.
* Tham khảo các website có độ tin cậy từ các công ty uy tín.
* Tham khảo các bài báo chính thống.

Những gì đạt được:

* Tìm ra thêm nhiều cách để giải quyết vấn đề.
* Tường tận hơn và định hình quy trình kiểm thử hiệu năng, tích luỹ thêm kỹ năng kiểm thử.
* Độc lập hơn và hoàn thiện hơn trong công việc.
* Tự đánh giá khả năng và định hướng lại công việc.

## Định hướng phát triển.

#### PHỤ LỤC

**URL website kiểm thử:**

https://demox.nextx.vn/auth/login

**Google docs log bug :**

<https://docs.google.com/spreadsheets/d/12IZGaVyQ1F9jmvPgvGATONJz3LlI8rYriHDJLMv7lXM/edit?gid=1762909176#gid=1762909176>

#### TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | ThaoLy, "Jmeter là gì? Tại sao lại sử dụng Jmeter? Những ưu điểm và nhược điểm của nó," 2019. [Online]. Available: https://viblo.asia/p/jmeter-la-gi-tai-sao-lai-su-dung-jmeter-nhung-uu-diem-va-nhuoc-diem-cua-no-3P0lPe2m5ox. |
| [2] | AnhTester, "Phân biệt Performance Testing, Load Testing và Stress Testing," [Online]. Available: https://anhtester.com/blog/phan-biet-performance-testing-load-testing-va-stress-testing-b315.html. |
| [3] | M. Trang, ""Cháy vé" đợt đầu mở bán concert Anh trai "say hi" tại Hà Nội," VOV.VN, 2024. [Online]. Available: https://vov.vn/giai-tri/chay-ve-dot-dau-mo-ban-concert-anh-trai-say-hi-tai-ha-noi-post1133773.vov. |
| [4] | T. B. Hân, "DoS, DDoS là gì? Nhận biết, ngăn chặn tấn công từ chối dịch vụ," 2021. [Online]. Available: https://www.thegioididong.com/game-app/dos-ddos-la-gi-nhan-biet-ngan-chan-tan-cong-tu-choi-dich-vu-1392351#subqmenu2. |
| [5] | Dileep\_Thekkethil, “​​Khuyến nghị về tốc độ trang web của Google cho năm 2025,” 2024. [Trực tuyến]. Available: https://www.stanventures.com/blog/google-pagespeed-insights/google-recommended-speed/. |
| [6] | S. Ventures, "​​Google Website Speed Recommendations for 2025," [Online]. Available: https://stanventures.com/blog/google-pagespeed-insights/google-recommended-speed/. |
| [7] | V. T. Nga, “JMeter - Công cụ kiểm thử hiệu năng chuyên nghiệp,” viblo.asia, 2019. [Trực tuyến]. Available: https://viblo.asia/p/jmeter-cong-cu-kiem-thu-hieu-nang-chuyen-nghiep-924lJWGX5PM. |
| [8] | JMeterApache. [Online]. Available: https://jmeter.apache.org/usermanual/glossary.html. |
| [9] | @tranthihuongtrang, "Tìm hiểu về kỹ thuật phân tích giá trị biên và phân vùng tương đương trong kiểm thử hộp đen," viblo, 2017. [Online]. Available: https://viblo.asia/p/tim-hieu-ve-ky-thuat-phan-tich-gia-tri-bien-va-phan-vung-tuong-duong-trong-kiem-thu-hop-den-bWrZnBJrZxw. |
| [10] | TranLeKhuyen, "Pairwise testing và cách tự liệt kê test case thủ công.," 2016. [Online]. Available: https://viblo.asia/p/pairwise-testing-va-cach-tu-liet-ke-test-case-thu-cong-aKYMNBgEM83E. |
| [11] | *JMeter Vỡ Lòng 01: Setup/Install and Launching..* [Phim]. JMeter VN, 2019. |
| [12] | J. Vietnam, "Conversion Rate - 10 Sai Lầm Khiến Tỷ Lệ Chuyển Đổi Trên Website Ecommerce Kém," [Online]. Available: https://jamstackvietnam.com/blog/jamstack-experience/ty-le-chuyen-doi-tren-website-ecommerce. |
| [13] | J. VN, "JMeter Vỡ Lòng 07: Test Fragment," 2019. [Online]. Available: https://www.youtube.com/watch?v=KB\_yLHIytuM&list=PLiBNXaoW-wnjJs7220P-1T9avSVsqWXNW&index=8. |
| [14] | S. MEPF, "Bài 4: Trích Dẫn & Tạo Danh Mục Tài Liệu Tham Khảo Tự Động Trong Word 2013-2016-2019," 2021. [Online]. Available: https://www.youtube.com/watch?v=6h7y-CpdIlQ. |
| [15] | L. T. K. Hoà, Tối ưu hoá kiểm thử cổng thanh toán Merchant bằng phương pháp manual testing kết hợp với công cụ JMeter, 2024. |
| [16] | Đ. G. T. Education, “Thực hành test hiệu năng Login bằng Jmeter - ĐỨC GIANG TESTER,” 2022. [Trực tuyến]. Available: https://www.youtube.com/watch?v=Wxuwv4roGT8. |
| [17] | N. D. Giang, "JMeter script 1: Cách đẩy thông tin vào log," 2023. [Online]. Available: https://giangtester.com/jmeter-script-1-cach-day-thong-tin-vao-log/. |
| [18] | INDA, “Hướng dẫn kết nối môi trường học tập,” 2024. [Trực tuyến]. Available: https://docs.google.com/document/d/1Ih1Jl0-69c-02TpFSoESAQIdaZUoDNjk/edit. |
| [19] | N. Liên, “Performance Test – Tổng hợp các loại kiểm thử hiệu năng thông dụng nhất hiện nay,” FPT Shop, 2023. [Trực tuyến]. Available: https://fptshop.com.vn/tin-tuc/danh-gia/performance-test-165619. |
| [20] | TopDev, "Hướng dẫn sử dụng JMeter test hiệu năng website – Phần 1," [Online]. Available: https://topdev.vn/blog/huong-dan-su-dung-jmeter-test-hieu-nang-website-phan-1/. |

Đếm tổng số các chức năng

Làm testcase

Sửa word

Quay video màn hình chạy các test case