TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC
MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU

ĐỀ TÀI

THU THẬP DỮ LIỆU CHỨNG KHOÁN TỪ WEBSITE SIMPLIZE.VN BẰNG CÔNG CỤ SELENIUM WEBDRIVER VÀ MONGODB

Giảng viên hướng dẫn: ThS. Lê Nhật Tùng

Nhóm sinh viên thực hiện:

Họ và tên	*MSSV*	*Mã lớp*
Hồ Gia Thành	2286400029	22DKHA1
Trương Minh Khoa	2286400011	22DKHA1
Huỳnh Thái Linh	2286400015	22DKHA1

TP. Hồ Chí Minh, 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN MÔN HỌC
MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU

ĐỀ TÀI

THU THẬP DỮ LIỆU CHỨNG KHOÁN TỪ WEBSITE SIMPLIZE.VN BẰNG CÔNG CỤ SELENIUM WEBDRIVER VÀ MONGODB

Giảng viên hướng dẫn: ThS. Lê Nhật Tùng

Nhóm sinh viên thực hiện:

Họ và tên	*MSSV*	*Mã lớp*
Hồ Gia Thành	2286400029	22DKHA1
Trương Minh Khoa	2286400011	22DKHA1
Huỳnh Thái Linh	2286400015	22DKHA1

TP. Hồ Chí Minh, 2024

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

TP.HCM, Ngày.....tháng.....năm 2024

Giảng viên hướng dẫn

(Ký tên, đóng dấu)

LỜI CAM ĐOAN

Chúng tôi, Hồ Gia Thành, Trương Minh Khoa và Huỳnh Thái Linh, xin cam đoan rằng:

Tất cả nội dung của bài báo cáo này là kết quả từ quá trình nghiên cứu và làm việc chung của cả ba chúng tôi. Các thông tin được trình bày trong báo cáo đều được thu thập từ những nguồn đáng tin cậy và đã được xử lý cẩn thận.

Chúng tôi đảm bảo rằng không có bất kỳ hành vi sao chép hay sử dụng thông tin không chính xác nào từ các nguồn khác. Mọi tài liệu tham khảo đã được ghi nguồn rõ ràng và tuân thủ đúng các quy định về trích dẫn học thuật.

Bài báo cáo này là sản phẩm nghiên cứu chung của chúng tôi và chưa từng được nộp hoặc công bố ở bất kỳ đâu trước đây. Chúng tôi hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của nội dung báo cáo này.

Chúng tôi hy vọng rằng bài báo cáo này sẽ mang đến một góc nhìn tổng quát và chi tiết về chủ đề "Thu thập dữ liệu chứng khoán từ website simplize.vn bằng công cụ Selenium WebDriver và MongoDB", bên cạnh đó cũng đồng thời đóng góp phần nhỏ vào việc nghiên cứu trong lĩnh vực này.

TP.HCM, Ngày.....tháng.....năm 2024

Sinh viên

Hồ Gia Thành
Trương Minh Khoa
Huỳnh Thái Linh

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT VÀ TỪ KHÓA

.....

Mục lục

Danh sách bảng

Danh sách hình vẽ

Chương 1

TỔNG QUAN

1.1 Giới thiệu đề tài

Trong bối cảnh công nghệ ngày càng phát triển, việc tự động hóa quy trình thu thập và phân tích dữ liệu đã trở thành một yếu tố quan trọng trong nhiều lĩnh vực, đặc biệt là tài chính. Đối với các nhà đầu tư, việc hiểu rõ sự biến động của cổ phiếu và các chỉ số thị trường giúp họ đưa ra quyết định đúng đắn. Đề tài "Thu thập dữ liệu về chứng khoán từ website simplize.vn[1] bằng công cụ Selenium WebDriver và MongoDB" được chúng tôi chọn nhằm mục tiêu áp dụng công cụ tự động hóa mạnh mẽ trong việc thu thập dữ liệu chứng khoán. Điều này giúp người nghiên cứu cung cấp các thông số quan trọng và đưa ra nhận định về xu hướng chứng khoán, hỗ trợ quyết định đầu tư hiệu quả hơn.

1.2 Nhiệm vụ của đồ án

Nhiệm vụ của đề tài "Thu thập dữ liệu về chứng khoán từ website simplize.vn" là áp dụng các kỹ thuật tự động thu thập dữ liệu từ web nhằm khám phá thông tin liên quan đến cổ phiếu, chứng khoán. Quá trình này giúp cho việc tự động thu thập dữ liệu của cổ phiếu nhanh chóng . Từ đó, nhà đầu tư có thể đưa ra quyết định đầu tư dựa trên thông tin thu thập được.

[1]Simplize.vn, là một trang web cung cấp thông tin phân tích và đánh giá về thị trường chứng khoán tại Việt Nam, giúp người dùng theo dõi và nắm bắt tình hình cổ phiếu. Trang web cung cấp dữ liệu từ nhiều nguồn tin cậy nhằm hỗ trợ nhà đầu tư trong việc ra quyết định

1.2.1 Tính cấp thiết của đề tài

Trong thời đại số hóa hiện nay, việc thu thập và dữ liệu trở thành một công cụ đắc lực giúp các nhà đầu tư và chuyên gia tài chính nắm bắt xu hướng thị trường. Đặc biệt trong lĩnh vực chứng khoán, việc theo dõi và phân tích các mã chứng khoán là điều cần thiết để đưa ra các quyết định đầu tư chính xác. Sự phát triển mạnh mẽ của công nghệ đã mở ra khả năng tự động hóa quá trình này, và công cụ mã nguồn mở Selenium là một trong những công cụ mạnh mẽ để hỗ trợ việc thu thập dữ liệu tự động từ các trang web tài chính.

Thu thập dữ liệu về chứng khoán từ website simplize.vn là một bước tiến quan trọng trong việc ứng dụng công nghệ vào đầu tư tài chính. Điều này không chỉ giúp nhà đầu tư tiết kiệm thời gian trong việc thu thập dữ liệu, mà còn cung cấp các thông tin kịp thời và chính xác, giúp đưa ra các quyết định hiệu quả hơn. Cụ thể, việc tự động thu thập dữ liệu chứng khoán có thể mang lại những lợi ích sau:

- **Tối ưu hóa chiến lược đầu tư**: Bằng cách thu thập dữ liệu liên tục, nhà đầu tư có thể dự đoán xu hướng biến động của mã chứng khoán , từ đó xây dựng các chiến lược đầu tư phù hợp hơn với tình hình thị trường.

- **Cải thiện khả năng ra quyết định dựa trên dữ liệu**: Thay vì dựa vào cảm tính hoặc thông tin không đầy đủ, các quyết định đầu tư sẽ dựa trên dữ liệu đã được phân tích kỹ lưỡng, giảm thiểu rủi ro và tăng hiệu quả.

- **Tiết kiệm thời gian và nguồn lực**: Việc tự động hóa việc thu thập dữ liệu giúp giảm thiểu thời gian, đồng thời cho phép các nhà đầu tư tập trung vào việc phân tích sâu hơn và đưa ra các chiến lược đầu tư tối ưu.

- **Nâng cao khả năng cạnh tranh trên thị trường**: Trong thị trường tài chính cạnh tranh khốc liệt, việc sử dụng công nghệ để cập nhật thông tin nhanh chóng và chính xác giúp nhà đầu tư đưa ra các quyết định nhanh chóng, từ đó giữ vững và nâng cao lợi thế cạnh tranh.

Với những lý do trên, có thể thấy rằng việc ứng dụng công nghệ Selenium để thu thập và phân tích dữ liệu các mã chứng khoán không chỉ mang lại nhiều lợi ích thiết thực cho nhà đầu tư mà còn đóng góp quan trọng vào việc nâng cao hiệu quả của hoạt động đầu tư trên thị trường chứng khoán.

1.2.2 Ý nghĩa khoa học và thực tiễn của đề tài

Ý nghĩa khoa học: Đề tài "Thu thập dữ liệu về chứng khoán từ website simplize.vn bằng công cụ Selenium WebDriver và MongoDB" đóng góp vào lĩnh vực phân tích dữ liệu bằng cách áp dụng các công cụ và phương pháp tự động hóa mạnh mẽ, giúp đơn giản hóa quá trình thu thập dữ liệu từ web. Qua đó, đề tài làm phong phú thêm lý thuyết về tự động hóa trong thu thập dữ liệu, đồng thời ứng dụng các phương pháp phân tích cơ bản để xử lý và khai thác thông tin từ các nguồn dữ liệu chứng khoán. Những kết quả này giúp các nhà nghiên cứu và chuyên gia trong lĩnh vực tài chính có thêm công cụ hỗ trợ trong việc nghiên cứu sâu hơn về xu hướng thị trường và dự đoán biến động cổ phiếu.

Ý nghĩa thực tiễn: Đề tài mang lại nhiều lợi ích cho các nhà đầu tư bằng việc cung cấp một phương pháp thu thập dữ liệu tự động và nhanh chóng. Kết quả thu thập dữ liệu về mã chứng khoán sẽ hỗ trợ các nhà đầu tư đưa ra quyết định dựa trên dữ liệu chính xác hơn, từ đó cải thiện hiệu quả đầu tư. Điều này không chỉ giúp tiết kiệm thời gian và nguồn lực mà còn nâng cao khả năng cạnh tranh của nhà đầu tư trên thị trường chứng khoán. Bằng cách tự động hóa việc thu thập dữ liệu, nhà đầu tư có thể nhanh chóng phát hiện các cơ hội đầu tư tiềm năng và quản lý rủi ro một cách hiệu quả. Phương pháp thu thập dữ liệu từ đề tài còn có thể được ứng dụng rộng rãi trong nhiều lĩnh vực khác, giúp nâng cao chất lượng và hiệu quả phân tích dữ liệu.

Hơn nữa, kết quả nghiên cứu này còn hỗ trợ các nhà hoạch định chính sách trong việc đưa ra các quy định và chính sách hỗ trợ sự phát triển bền vững của thị trường tài chính. Bằng cách kết hợp giữa lý thuyết về tự động hóa và ứng dụng thực tiễn trong lĩnh vực chứng khoán, đề tài không chỉ mang lại giá trị khoa học mà còn đóng góp tích cực vào sự phát triển của ngành tài chính, mở ra nhiều cơ hội mới cho các nhà đầu tư và góp phần vào sự phát triển bền vững của thị trường.

1.3 Mục tiêu

1.3.1 Mục tiêu tổng quan

Đề tài nhằm cung cấp một phương pháp thu thập dữ liệu tự động từ website simplize.vn đối với các mã chứng khoán, giúp nhận diện các xu hướng và biến động của chứng khoán. Từ đó, hỗ trợ nhà đầu tư đưa ra các chiến lược đầu tư hiệu quả và tối ưu hóa quyết định tài chính, nâng cao khả năng cạnh tranh trên thị trường.

1.3.2 Mục tiêu cụ thể

Mục tiêu của đề tài này là xây dựng một hệ thống tự động hóa sử dụng Selenium để thu thập dữ liệu từ website simplize.vn, tập trung vào các mã chứng khoán cụ thể. Chúng tôi sẽ thu thập các thông tin quan trọng như giá cổ phiếu, khối lượng giao dịch, và các chỉ số tài chính liên quan. Dữ liệu thu thập sẽ được tổ chức lại theo cấu trúc phù hợp để dễ dàng phân tích.

Tiếp theo, chúng tôi sẽ lưu trữ dữ liệu này trong cơ sở dữ liệu NoSQL, cụ thể là MongoDB, nhằm tận dụng tính linh hoạt và hiệu quả trong việc lưu trữ dữ liệu. Các truy vấn NoSQL sẽ được sử dụng để thực hiện phân tích dữ liệu, khám phá các xu hướng và biến động của cổ phiếu, cũng như đánh giá ảnh hưởng của các yếu tố khác đến giá cổ phiếu. Cuối cùng, chúng tôi sẽ đánh giá hiệu quả của hệ thống thu thập dữ liệu tự động dựa trên tốc độ thu thập dữ liệu, độ chính xác của dữ liệu và khả năng mở rộng của hệ thống khi xử lý các tập dữ liệu lớn từ trang web simplize.vn.

1.4 Đối tượng và phạm vi

1.4.1 Đối tượng

Đối tượng nghiên cứu của đề tài là các mã chứng khoán và dữ liệu liên quan được thu thập từ website simplize.vn. Các mã chứng khoán có sự biến động đáng kể trong thị trường tài chính, và việc thu thập dữ liệu liên quan sẽ giúp hiểu rõ hơn về các yếu tố ảnh hưởng đến giá cổ phiếu. Đề tài nhằm cung cấp cái nhìn sâu sắc về xu hướng biến động của mã chứng khoán, từ đó đưa ra những đề xuất giúp các nhà đầu tư tối ưu hóa được quyết định của mình.

1.4.2 Phạm vi

Đề tài tập trung vào thu thập dữ liệu về các mã chứng khoán từ website simplize.vn. Chúng tôi sẽ áp dụng phương pháp thu thập dữ liệu và kỹ thuật tự động hóa để nhận diện các xu hướng và biến động của cổ phiếu. Dựa trên kết quả này, chúng tôi sẽ đề xuất các chiến lược đầu tư hợp lý và đánh giá hiệu quả của chúng. Điều này giúp mang lại giá trị lý thuyết và thực tiễn cho các nhà đầu tư trên thị trường tài chính.

1.5 Phương pháp nghiên cứu

1.5.1 Phương pháp nghiên cứu sơ bộ

Trước khi tiến hành thu thập dữ liệu, chúng tôi sẽ thực hiện một nghiên cứu sơ bộ để hiểu rõ hơn về lĩnh vực nghiên cứu và các yếu tố quan trọng liên quan đến các mã chứng khoán. Nghiên cứu này sẽ bao gồm việc tìm hiểu về thị trường tài chính, các yếu tố ảnh hưởng đến biến động của cổ phiếu. Thông qua nghiên cứu sơ bộ, chúng tôi sẽ xác định các vấn đề cụ thể cần giải quyết và đề xuất các phương pháp nghiên cứu phù hợp để đạt được kết quả tối ưu.

1.5.2 Phương pháp nghiên cứu tài liệu

Phương pháp nghiên cứu này bao gồm việc tổng hợp tài liệu và các báo cáo nghiên cứu liên quan đến web scraping, Selenium, phân tích dữ liệu và cơ sở dữ liệu NoSQL. Qua việc nghiên cứu các nguồn tài liệu hiện có, nhóm sẽ thu thập kiến thức nền tảng và các ví dụ thực tiễn trong việc áp dụng các công cụ này vào thực tế. Nhóm sẽ đánh giá và phân tích các phương pháp thu thập và phân tích dữ liệu, đặc biệt là các kỹ thuật sử dụng Selenium trong lĩnh vực tài chính. Những thông tin này sẽ là cơ sở vững chắc để phát triển công cụ thu thập và phân tích dữ liệu phù hợp cho đề tài, từ đó đưa ra các chiến lược đầu tư hiệu quả dựa trên dữ liệu thu thập được từ trang web simplize.vn.

1.5.3 Phương pháp nghiên cứu thống kê

Trong quá trình phân tích dữ liệu, nhóm sẽ áp dụng các phương pháp thống kê để mô tả và phân tích các biến số quan trọng liên quan đến các mã chứng khoán được thu thập từ trang web simplize.vn. Các phương pháp này sẽ cho phép nhóm đánh giá mối quan hệ giữa các biến số như giá cổ phiếu, đánh giá của nhà phân tích và khối lượng giao dịch. Thông qua việc phân tích này, nhóm có thể xác định những yếu tố ảnh hưởng đến xu hướng đầu tư của nhà đầu tư và hiệu quả kinh doanh của từng mã chứng

khoán. Việc sử dụng các phương pháp thống kê sẽ cung cấp cái nhìn sâu sắc về các mô hình hành vi của nhà đầu tư, giúp tối ưu hóa chiến lược đầu tư dựa trên dữ liệu thu thập được

1.5.4 Phương pháp thực nghiệm

Nhóm sẽ tiến hành thực nghiệm thông qua việc xây dựng hệ thống thu thập dữ liệu tự động bằng Selenium. Quá trình này bao gồm việc thử nghiệm các trường hợp trích xuất dữ liệu từ trang web simplize.vn, đánh giá hiệu suất của hệ thống và thực hiện các điều chỉnh cần thiết. Đồng thời, việc lưu trữ và xử lý dữ liệu trong cơ sở dữ liệu NoSQL cũng sẽ được thử nghiệm nhằm đảm bảo khả năng lưu trữ và truy xuất hiệu quả, phục vụ cho việc phân tích dữ liệu và đưa ra quyết định trong đầu tư chứng khoán.

1.5.5 Phương pháp đánh giá

Sau khi đã thu thập dữ liệu, chúng tôi sẽ thực hiện phương pháp đánh giá để đo lường hiệu quả của quá trình tự động thu thập dữ liệu và phân tích dữ liệu. Quá trình này sẽ xem xét tốc độ thu thập, tính chính xác và đầy đủ của dữ liệu, cùng với hiệu quả của quy trình phân tích thông qua các tiêu chí về độ tin cậy và các giá trị khác từ kết quả phân tích hệ thống. Nhóm sẽ sử dụng các chỉ số như tỷ lệ lỗi trong dữ liệu thu thập, thời gian cần thiết để hoàn thành quá trình thu thập và phân tích, cũng như mức độ chính xác của các dự đoán từ mô hình phân tích. Các kết quả này sẽ giúp nhóm cải thiện quy trình và tối ưu hóa hệ thống để nâng cao hiệu suất trong việc hỗ trợ quyết định đầu tư.

1.6 Những đóng góp nghiên cứu của đề tài

1.6.1 Trong lĩnh vực học thuật

Công cụ mã nguồn mở Selenium đóng vai trò quan trọng trong lĩnh vực học thuật bằng cách tự động hóa việc thu thập dữ liệu từ trang web chứng khoán simplize.vn. Nhờ đó, các nhà nghiên cứu có thể trích xuất thông tin về giá cổ phiếu và biến động thị trường, xây dựng cơ sở dữ liệu chi tiết. Điều này tạo nền tảng cho các nghiên cứu phân tích định lượng và mô hình dự đoán, giúp nâng cao độ chính xác trong dự báo tài chính và ra quyết định đầu tư.

1.6.2 Trong thực tiễn kinh doanh

Việc ứng dụng Selenium trong thực tiễn kinh doanh, đặc biệt với các doanh nghiệp trong lĩnh vực tài chính, giúp tối ưu hóa quy trình thu thập và phân tích dữ liệu từ trang web chứng khoán như simplize.vn. Điều này giúp các doanh nghiệp dễ dàng theo dõi biến động thị trường và dự đoán xu hướng đầu tư, từ đó xây dựng chiến lược hiệu quả hơn. Việc ra quyết định nhanh chóng dựa trên dữ liệu cập nhật không chỉ cải thiện hiệu suất kinh doanh mà còn nâng cao khả năng cạnh tranh trong môi trường tài chính đầy biến động.

1.7 Cấu trúc đồ án

1.7.1 Trình bày cấu trúc đồ án

Chương 1: Tổng quan
Chương 2: Cơ sở lý thuyết
Chương 3: Phương pháp thực nghiệm
Chương 4: Kết quả thực nghiệm

1.7.2 Tóm tắt từng chương

Chương 1: Tổng quan

Trình bày lý do chọn đề tài, tính cấp thiết, mục tiêu và phương pháp nghiên cứu bên cạnh đó là những đóng góp trong học thuật và thực tiễn kinh doanh của đề tài. Từ đó, cho ta một nền tảng lý thuyết cần thiết để có thể hiểu rõ về tầm quan trọng của kỹ thuật web scraping.

Chương 2: Cơ sở lý thuyết

Cung cấp các kiến thức liên quan về web scraping, Selenium, NoSQL, MongoDB, bên cạnh đó trình bày các khái niệm về dữ liệu và phân tích dữ liệu từ website simplize. Các tài liệu lý thuyết và nghiên cứu liên quan sẽ được tham khảo để làm cơ sở cho phát triển đề tài.

Chương 3: Phương pháp thực nghiệm

Mô tả chi tiết cụ thể về quá trình thu thập và xử lý dữ liệu,gồm việc xây dựng hệ thống web scraing, lưu trữ và phân tích dữ liệu, bên cạnh đó là cách thức thực nghiệm và đánh giá hiệu quả của hệ thống.

Chương 4: Kết quả thực nghiệm

Phần này sẽ tập trung vào phân tích dữ liệu thu thập được từ trang web simplize.vn, đưa ra các đánh giá các xu hướng thị trường và biến động của các mã cổ phiếu. Các kết quả phân tích sẽ được trình bày dưới dạng số liệu cụ thể, kèm theo những kết luận quan trọng về tình hình thị trường.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1 WEB SCRAPING

2.1.1 Giới thiệu Web scraping

Web Scraping [1] là một phương pháp trích xuất dữ liệu từ các trang web bằng phần mềm tự động lưu thông tin ở định dạng có tổ chức.

Thay vì sao chép và dán thủ công từng phần thông tin như cách truyền thống, phương pháp này cho phép chúng ta thu thập dữ liệu với số lượng lớn trong thời gian ngắn. Đặc biệt, với nhu cầu phân tích và xử lý dữ liệu ngày càng tăng, web scraping trở thành công cụ hữu ích để trích xuất thông tin từ nhiều nguồn khác nhau trên internet mà không cần tốn quá nhiều công sức.

2.1.2 Nguyên lý hoạt động

Quy trình hoạt động của web scraping gồm các bước:

Hình 2.1: Quy trình Web scarping. Nguồn [2]

- **Xác định website mục tiêu**: Đây là trang web mà bạn muốn thu thập dữ liệu từ đó.

- **Thu thập URL**: Lấy địa chỉ URL của website mà bạn muốn trích xuất thông tin.
- **Gửi yêu cầu**: Tạo yêu cầu HTTP để lấy HTML của website mục tiêu.
- **Tìm dữ liệu**: Sử dụng bộ định vị để xác định vị trí dữ liệu trong HTML.
- **Lưu dữ liệu**: Ghi lại dữ liệu đã trích xuất vào file JSON, CSV hoặc các định dạng cấu trúc khác.

2.1.3 Ưu điểm và hạn chế

Ưu điểm:

- **Tự động hóa quy trình thu thập dữ liệu**: Web scraping cho phép giảm thiểu thời gian và công sức so với phương pháp thu thập dữ liệu truyền thống.
- **Khả năng thu thập dữ liệu quy mô lớn**: Người dùng có thể lấy thông tin từ nhiều trang web khác nhau chỉ trong một khoảng thời gian ngắn.
- **Cập nhật dữ liệu liên tục**: Các công cụ web scraping có thể được cấu hình để tự động làm mới dữ liệu theo thời gian thực, đảm bảo thông tin luôn chính xác và cập nhật.
- **Hỗ trợ phân tích thông tin**: Cung cấp dữ liệu hữu ích cho việc đánh giá và nghiên cứu các lĩnh vực khác nhau.
- **Theo dõi biến động thông tin**: Giúp giám sát sự thay đổi của các yếu tố liên quan một cách hiệu quả.

Hạn chế:

- **Vấn đề pháp lý**: Việc thu thập dữ liệu từ các trang web có thể dẫn đến vi phạm điều khoản dịch vụ, gây ra rủi ro pháp lý cho người thực hiện.
- **Khó khăn với trang web động**: Các trang web sử dụng JavaScript hoặc công nghệ tải động có thể làm khó khăn trong quá trình thu thập dữ liệu.

- **Yêu cầu kiến thức kỹ thuật**: Để thực hiện web scraping hiệu quả, người dùng cần nắm vững lập trình, cấu trúc HTML và các giao thức web.
- **Đạo đức và trách nhiệm**: Quy trình thu thập và tổ chức thông tin từ các trang web có thể gây ra những lo ngại về tính hợp pháp và công bằng trong việc sử dụng dữ liệu. Do đó, người dùng cần thực hiện web scraping một cách có trách nhiệm, tôn trọng quyền riêng tư và tuân thủ các quy định của trang web để bảo vệ quyền lợi của các bên liên quan.[3]

2.2 SELENIUM

2.2.1 Tổng quan về công cụ mã nguồn mở Selenium

1. Khái niệm:

   Selenium là một bộ công cụ và thư viện mã nguồn mở tự động hóa việc thử nghiệm các trang web và ứng dụng web. Tính linh hoạt của nó trong việc thử nghiệm trên nhiều môi trường khác nhau là nhờ khả năng đa trình duyệt, đa ngôn ngữ và đa nền tảng. Selenium tích hợp liền mạch với các quy trình phát triển hiện có và hỗ trợ các ngôn ngữ lập trình như Java, JavaScript, C#, PHP, Python và Ruby. Hơn thế Selenium có thể tương thích với nhiều trình duyệt web phổ biến như Chrome, Firefox, Safari, Edge và Opera, đảm bảo khả năng kiểm thử trên nhiều trình duyệt. Tính linh hoạt của Selenium còn được nâng cao nhờ khả năng tương thích với các khung kiểm thử tự động khác như TestNG, JUnit, MSTest, pytest, WebdriverIO,...[4]

2. Lịch sử phát triển:

   Năm 2004, Jason Huggins, một kỹ sư làm việc tại ThoughtWorks[1] đã phát triển một chương trình với tên gọi là JavaScript Test Runner. Ông đã phát triển thư viện Javascript để chạy tự động các test trên nhiều trình duyệt. Đây chính là cơ sở để Selenium IDE và Selenium RC ra đời.

   Năm 2006, Simon Stewart - một nhân viên của Google tiếp tục phát triển Selenium với công việc được đặt tên là WebDriver. Nhờ có công cụ

[1]ThoughtWorks là công ty tư vấn công nghệ và phát triển phần mềm quốc tế, chuyên cung cấp dịch vụ phát triển phần mềm, tư vấn công nghệ và chuyển đổi số. Được thành lập vào năm 1993, ThoughtWorks nổi bật với cam kết sử dụng công nghệ để giải quyết các vấn đề phức tạp, khuyến khích đổi mới và áp dụng các phương pháp phát triển linh hoạt (Agile). Công ty có văn phòng tại nhiều quốc gia và thường xuyên tham gia các hoạt động xã hội nhằm cải thiện cộng đồng thông qua công nghệ.

Hình 2.2: Hình minh họa về công cụ mã nguồn mở Selenium. Nguồn [5]

này, Google đã nhận được một lượng người sử dụng Selenium rất lớn nhưng đứng trước những hạn chế của sản phẩm thì các tester vẫn phải làm việc rất vất vả.

Năm 2008, Selenium và WebDriver chính thức được kết hợp bởi Selenium đang dần lớn mạnh và WebDriver lại là công cụ của tương lai. Với sự kết hợp này, người dùng được cung cấp một tệp những tính năng lớn.[6]

3. Thành phần của Selenium

Selenium bao gồm 4 thành phần chính:

- **Selenium IDE**: Đây là công cụ ghi và phát lại, giúp dễ dàng tạo và chỉnh sửa các kịch bản kiểm thử mà không cần nhiều kiến thức lập trình. Nó phù hợp cho các kiểm thử đơn giản nhưng gặp hạn chế khi xử lý các trang web động hoặc tự động hóa phức tạp.
- **Selenium RC**: Đây là công cụ cũ, đã bị thay thế bởi WebDriver do tính phức tạp và hiệu suất thấp hơn. RC điều khiển trình duyệt thông qua JavaScript và hoạt động như một lớp trung gian, nhưng không còn được sử dụng rộng rãi.
- **Selenium WebDriver**: Thành phần chính của Selenium, cung cấp API linh hoạt để tương tác trực tiếp với các trình duyệt (Chrome, Firefox, Edge, Safari) mà không cần lớp trung gian. WebDriver có thể thực hiện các thao tác như nhấp chuột, cuộn trang và thu thập dữ liệu, hỗ trợ nhiều ngôn ngữ lập trình và ứng dụng web động.

- **Selenium Grid**: Công cụ này cho phép thực thi các kịch bản kiểm thử đồng thời trên nhiều trình duyệt và hệ thống khác nhau. Grid hỗ trợ phân phối các kịch bản kiểm thử, giúp tiết kiệm thời gian và tối ưu hóa tài nguyên kiểm thử. [7]

Hình 2.3: Các thành phần chính của Selenium. Nguồn [8]

2.2.2 Cách thức hoạt động của Selenium (WebDriver)

Selenium WebDriver hoạt động dựa trên việc giao tiếp giữa mã lập trình và trình duyệt thông qua các driver cụ thể. Quá trình này được thực hiện thông qua các bước sau:

1. **Thư viện khách hàng Selenium (Selenium Client Library)**: Người dùng viết mã điều khiển trình duyệt sử dụng các ngôn ngữ lập trình như Java, Python, C#, hoặc Ruby. Mỗi ngôn ngữ đều có thư viện Selenium tương ứng để hỗ trợ việc điều khiển trình duyệt thông

qua các phương thức được cung cấp sẵn.

2. Giao thức JSON Wire qua HTTP (JSON Wire Protocol Over HTTP):

Mã lệnh từ thư viện Selenium Client được gửi tới trình điều khiển trình duyệt (Browser Driver) dưới dạng các yêu cầu HTTP sử dụng giao thức JSON. Giao thức này giúp chuyển các lệnh như mở trang web, tương tác với phần tử trang, cuộn trang, v.v., tới trình duyệt.

3. Trình điều khiển trình duyệt (Browser Driver):

Mỗi trình duyệt (Chrome, Firefox, Internet Explorer, v.v.) có một trình điều khiển riêng, chẳng hạn như ChromeDriver cho Google Chrome hay GeckoDriver cho Firefox. Các trình điều khiển này nhận lệnh từ thư viện Selenium qua giao thức JSON và chuyển đổi chúng thành các lệnh mà trình duyệt có thể hiểu và thực thi.

4. Trình duyệt (Browser):
Sau khi nhận lệnh từ Browser Driver, trình duyệt thực hiện các thao tác điều khiển tương tự như hành động của người dùng, chẳng hạn như mở một trang web, nhấn nút, hoặc nhập dữ liệu.

5. Giao tiếp qua máy chủ HTTP (HTTP over HTTP Server):
Lệnh điều khiển giữa WebDriver và trình duyệt được truyền tải qua giao thức HTTP, giúp việc điều khiển trình duyệt từ xa trở nên hiệu quả và dễ dàng.[9]

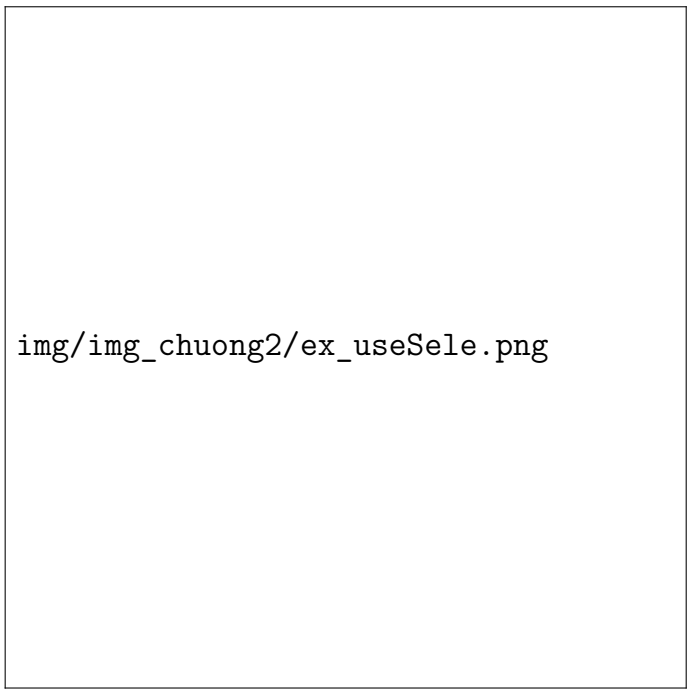
Hình 2.4: Cách thức hoạt động của Selenium WebDrive. Nguồn [10]

Ví dụ: Selenium có thể truy cập vào các trang web khác nhau, cụ thể trang web trong đề tài là trang "simplize.vn". Từ đó, có thể phát triển thêm để hệ thống tự động hóa lấy dữ liệu của trang web như: Mã chứng khoán, lịch sử giá, hồ sơ doanh nghiệp,...

2.2.3 Ưu điểm và hạn chế của Selenium

Ưu điểm:

- **Khả năng tương thích giữa các trình duyệt:** Selenium hỗ trợ nhiều trình duyệt như Chrome, Firefox, Safari và Internet Explorer,



Hình 2.5: Ví dụ minh họa cách sử dụng Selenium WebDrive.

giúp kiểm tra các ứng dụng web trên các nền tảng khác nhau dễ dàng hơn.

- **Độc lập nền tảng:** Selenium là một công cụ đa nền tảng, có nghĩa là nó có thể được sử dụng trên các hệ điều hành khác nhau như Windows, Linux và macOS.
- **Hỗ trợ nhiều ngôn ngữ lập trình:** Selenium hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Java, C#, Python, Ruby và các ngôn ngữ khác. Điều này cho phép các nhà phát triển và người thử nghiệm chọn ngôn ngữ mà họ cảm thấy thoải mái nhất.
- **Cộng đồng và tài nguyên lớn:** Selenium có một cộng đồng lớn và năng động. Điều này có nghĩa là có rất nhiều tài liệu, hướng dẫn và diễn đàn có sẵn, giúp người dùng dễ dàng tìm thấy trợ giúp và giải pháp cho các vấn đề phổ biến.
- **Tích hợp với các công cụ khác:** Selenium có thể dễ dàng tích hợp với các công cụ và khung công tác khác, chẳng hạn như TestNG[2], JUnit[3], Maven[4] và Jenkins[5], tăng cường khả năng của nó và làm cho

[2]TestNG là framework kiểm thử Java mạnh mẽ, hỗ trợ kiểm thử đơn vị, tích hợp và chức năng.

[3]JUnit là thư viện kiểm thử mã nguồn mở cho Java, phổ biến cho việc thực hiện kiểm thử đơn vị.

[4]Maven là công cụ quản lý dự án và tự động hóa xây dựng, giúp quản lý phụ thuộc và cấu hình dự án Java.

[5]Jenkins là máy chủ tự động hóa mã nguồn mở, hỗ trợ tích hợp liên tục (CI) và triển khai liên tục (CD).

nó phù hợp với các nhu cầu thử nghiệm khác nhau.

- **Tính linh hoạt và khả năng mở rộng:** Selenium có thể được mở rộng cho các chức năng khác nhau thông qua các API của nó, làm cho nó có thể thích ứng với các kịch bản thử nghiệm khác nhau.
- **Hỗ trợ thực hiện song song:** Selenium cho phép thực hiện song song các kịch bản kiểm tra, có thể làm giảm đáng kể thời gian thực hiện kiểm tra tổng thể.

Hạn chế:

- **Hỗ trợ cho ứng dụng máy tính để bàn còn hạn chế:** Selenium chủ yếu được thiết kế để kiểm tra các ứng dụng web, nên khả năng hỗ trợ cho việc kiểm tra ứng dụng máy tính để bàn còn rất hạn chế.
- **Không có báo cáo tích hợp:** Selenium thiếu khả năng báo cáo tích hợp. Kết quả kiểm tra cần được quản lý bằng các công cụ bên ngoài hoặc bằng cách tích hợp nó với các khung báo cáo khác.
- **Xử lý các yếu tố động:** Selenium đôi khi có thể phải đối mặt với những thách thức khi xử lý các phần tử web động thay đổi thường xuyên, đòi hỏi những nỗ lực bổ sung trong việc bảo trì tập lệnh.
- **Không hỗ trợ đầu đọc CAPTCHA và mã vạch:** Selenium không thể tự động hóa các thử nghiệm liên quan trực tiếp đến CAPTCHA hoặc đầu đọc mã vạch, vì các tính năng này được thiết kế để ngăn chặn các tương tác tự động.
- **Khó khăn ban đầu:** Đối với người mới bắt đầu, có thể có gặp khó khăn trong việc học tập liên quan đến Selenium, đặc biệt nếu họ không quen thuộc với các ngôn ngữ lập trình mà nó hỗ trợ.[11]

2.2.4 Ứng dụng của Selenium

- **Kiểm thử tự động:** Sử dụng Selenium để kiểm tra chức năng và hiệu suất của ứng dụng web trên nhiều trình duyệt khác nhau mà không cần can thiệp thủ công.
- **Thu thập dữ liệu web:** Tự động truy cập và thu thập thông tin từ các trang web, đặc biệt là từ các trang không cung cấp API.
- **Giả lập hành vi người dùng:** Mô phỏng các tương tác của người dùng trên ứng dụng web, như đăng nhập, thêm sản phẩm vào giỏ hàng, hoặc điền thông tin vào biểu mẫu.
- **Kiểm thử trên nhiều nền tảng:** Đảm bảo rằng ứng dụng hoạt động chính xác trên nhiều trình duyệt khác nhau, giúp phát hiện các vấn đề tương thích.
- **Tích hợp với quy trình phát triển (CI/CD)[6]:** Kết hợp Selenium với các công cụ CI/CD như Jenkins để tự động hóa các kịch bản kiểm thử trong quy trình phát triển phần mềm.[12]

[6]CI/CD là viết tắt của Continuous Integration (Tích hợp liên tục) và Continuous Deployment/Delivery (Triển khai/Phát hành liên tục). Đây là một phương pháp phát triển phần mềm nhằm tự động hóa quy trình phát triển, kiểm thử và phát hành ứng dụng.

2.2.5 So sánh Selenium với Scrapy

Đặc điểm	Scrapy	Selenium
Mục đích	Quét và thu thập dữ liệu web.	Kiểm tra web và tự động hóa; cũng có thể dùng để cạo.
Hỗ trợ ngôn ngữ	Chỉ hỗ trợ Python.	Hỗ trợ nhiều ngôn ngữ: Java, JavaScript, Python, C#, PHP, Ruby.
Tốc độ khớp lệnh	Thực hiện nhanh, phù hợp với dự án lớn.	Chậm hơn do tương tác với trình duyệt.
Sự phù hợp của dự án	Lý tưởng cho dự án cạo quy mô nhỏ và lớn.	Phù hợp cho dự án quy mô vừa và nhỏ, nhất là với nội dung động.
Khả năng mở rộng	Mở rộng cao, xử lý nhiều yêu cầu đồng thời.	Hạn chế mở rộng do sử dụng nhiều tài nguyên.
Hỗ trợ proxy	Có hỗ trợ proxy.	Cũng hỗ trợ proxy.
Khả năng không đồng bộ	Không đồng bộ theo thiết kế.	Thiếu khả năng không đồng bộ gốc.
Selectors	Sử dụng CSS và XPath để chọn HTML.	Cũng sử dụng CSS và XPath, linh hoạt trong lựa chọn.
Kết xuất động	Không thể hiển thị nội dung động; cần thư viện bổ sung.	Có khả năng hiển thị đầy đủ trang JavaScript và AJAX.
Hỗ trợ trình duyệt	Không tương tác trình duyệt; chỉ xử lý yêu cầu HTTP.	Hỗ trợ Chrome, Edge, Firefox, và Safari.
Thực hiện không đầu	SKhông hỗ trợ thực thi không đầu.	Hỗ trợ thực thi không đầu, không hiển thị giao diện đồ họa.
Tương tác trình duyệt	SKhông có tương tác trình duyệt trực tiếp.	Cho phép nhấp, cuộn và điền biểu mẫu.

Bảng 2.1: So sánh Selenium với Scrapy.[13]

2.3 NOSQL

2.3.1 Giới thiệu cơ sở dữ liệu NoSQL

NoSQL là một loại hệ thống quản lý cơ sở dữ liệu (DBMS)[7] được thiết kế để xử lý và lưu trữ khối lượng lớn dữ liệu phi cấu trúc và bán cấu trúc. Không giống như các cơ sở dữ liệu quan hệ truyền thống sử dụng các bảng có lược đồ được xác định trước để lưu trữ dữ liệu, cơ sở dữ liệu NoSQL sử dụng các mô hình dữ liệu linh hoạt có thể thích ứng với các thay đổi trong cấu trúc dữ liệu và có khả năng mở rộng theo chiều ngang để xử lý lượng dữ liệu ngày càng tăng.[14]

2.3.2 Phân loại cơ sở dữ liệu NoSQL

- **Cơ sở dữ liệu**
  document: Lưu trữ dữ liệu dưới dạng tài liệu (document), thường ở định dạng JSON hoặc BSON. Ví dụ: MongoDB, Couchbase.
- **Cơ sở dữ liệu key-value:** Lưu trữ dữ liệu dưới dạng cặp khóa-giá trị (key-value pairs). Ví dụ: Redis, DynamoDB.
- **Cơ sở dữ liệu column-family:** Lưu trữ dữ liệu theo các cột (columns) thay vì hàng, phù hợp với dữ liệu lớn. Ví dụ: Cassandra, HBase.
- **Cơ sở dữ liệu graph:** Lưu trữ và quản lý dữ liệu dưới dạng đồ thị (graph), với các nút (nodes) và cạnh (edges) để biểu diễn mối quan hệ. Ví dụ: Neo4j, ArangoDB.

2.3.3 Ưu điểm và nhược điểm

Ưu điểm:

- **Khả năng mở rộng:** NoSQL hỗ trợ mở rộng theo chiều ngang, giúp thêm nhiều máy chủ khi cần, thích hợp với dữ liệu lớn và khối lượng công việc tăng trưởng.
- **Linh hoạt về cấu trúc:** Hỗ trợ dữ liệu có cấu trúc, bán cấu trúc và không cấu trúc, thích hợp cho các ứng dụng không có cấu trúc dữ liệu cố định.
- **Hiệu suất cao trong truy vấn:** Được thiết kế để xử lý khối lượng lớn dữ liệu nhanh chóng.
- **Chi phí thấp:** Không yêu cầu hạ tầng phần cứng đắt tiền và có thể chạy trên các cụm phần cứng giá rẻ.

[7]DBMS (Database Management System) - Hệ thống Quản lý Cơ sở Dữ liệu.

Nhược điểm:

- **Thiếu nhất quán:** NoSQL thường dựa trên mô hình "eventual consistency," có thể dẫn đến dữ liệu không chính xác hoặc lỗi thời.
- **Thiếu chuẩn hóa:** Hiện chưa có chuẩn chung cho các hệ thống NoSQL, gây khó khăn cho việc chuyển đổi và tích hợp.
- **Hỗ trợ phân tích hạn chế:** So với SQL, NoSQL kém hơn trong việc phân tích dữ liệu phức tạp.

2.3.4 Các ứng dụng phổ biến của NoSQL

- **Dữ liệu lớn (Big Data):** Lưu trữ và xử lý khối lượng lớn dữ liệu từ nhiều nguồn khác nhau.
- **Mạng xã hội:** Quản lý và phân tích dữ liệu người dùng và các tương tác giữa họ.
- **Thương mại điện tử:** Quản lý thông tin sản phẩm, đơn hàng và dữ liệu khách hàng một cách linh hoạt.

2.4 MongoDB

2.4.1 Giới thiệu về MongoDB

1. Định nghĩa:

   MongoDB là một chương trình quản lý cơ sở dữ liệu NoSQL mã nguồn mở. NoSQL (Không chỉ SQL) được sử dụng thay thế cho cơ sở dữ liệu quan hệ truyền thống. Cơ sở dữ liệu NoSQL khá hữu ích để làm việc với các bộ dữ liệu phân tán lớn. MongoDB là một công cụ có thể quản lý thông tin định hướng tài liệu, lưu trữ hoặc truy xuất thông tin.[15]

2. Lịch sử phát triển:

   MongoDB được bắt đầu phát triển vào đầu năm 2007 khi công ty 10gen đang phát triển một nền tảng tương tự dịch vụ Azure[8] của Microsoft. Công ty 10gen là một công ty phần mềm có trụ sở tại New York, nay được đổi tên thành MongoDB Inc. Việc phát triển ban đầu tập trung vào xây dựng PaaS (một nền tảng dịch vụ) nhưng sau đó vào năm 2009, MongoDB đã

[8]Microsoft Azure là nền tảng điện toán đám mây của Microsoft, cung cấp các dịch vụ như cơ sở hạ tầng (IaaS), nền tảng (PaaS) và phần mềm (SaaS). Azure cho phép doanh nghiệp phát triển, triển khai và quản lý ứng dụng mà không cần quản lý cơ sở hạ tầng. Nền tảng này nổi bật với khả năng mở rộng, chi phí hiệu quả và tích hợp dễ dàng với các sản phẩm khác của Microsoft. Azure cũng cung cấp các dịch vụ phân tích, trí tuệ nhân tạo, và bảo mật để hỗ trợ doanh nghiệp trong việc tối ưu hóa hoạt động và bảo vệ dữ liệu.

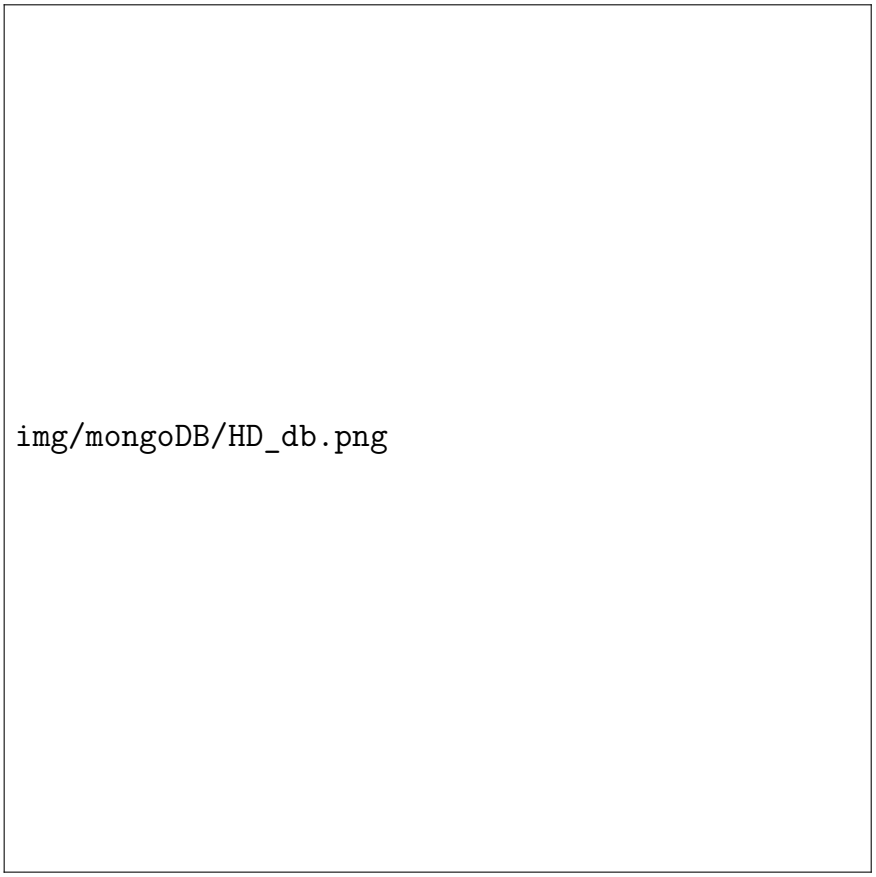
xuất hiện trên thị trường như một dự án mã nguồn mở máy chủ cơ sở dữ liệu và được duy trì bởi chính tổ chức này.[16]

Tháng 3 năm 2010, MongoDB Inc. đã tung ra sản phẩm sẵn sàng đầu tiên của mình là phiên bản 1.4. Phiên bản ổn định tiếp theo của MongoDB là phiên bản 2.4.9 được phát hành vào ngày 10 tháng 1 năm 2014.

Đầu năm 2015, phiên bản 3.0 được phát hành, cuối năm 2015 phiên 3.2 ra đời đi kèm với công cụ quản trị trên giao diện đồ họa MongoDB Compass.

2.4.2 Cách MongoDB hoạt động

MongoDB hoạt động dưới dạng một hệ thống cơ sở dữ liệu phi quan hệ, lưu trữ dữ liệu dưới dạng tài liệu (document) JSON. Dữ liệu được lưu trữ trong collections và documents. Do đó, database, collection và documents sẽ có liên quan với nhau như hình dưới đây:



Hình 2.6: Cách hoạt động của MongoDB. Nguồn [17]

- Cơ sở dữ liệu MongoDB lưu trữ tài liệu trong các collections, tương tự như bảng trong cơ sở dữ liệu quan hệ. Mỗi collection có thể chứa nhiều tài liệu (documents) có cấu trúc dữ liệu tùy ý.

- Bây giờ bên trong collection sẽ có tài liệu (documents). Các tài liệu này sẽ chứa dữ liệu mà bạn muốn lưu trữ trong MongoDB database. Mỗi document có thể chứa nhiều fields dữ liệu, mỗi field được định danh bằng tên và có giá trị tương ứng.

- Các tài liệu (documents) được tạo bằng cách sử dụng các field. Các field là các key-value pair trong tài liệu, nó giống như các cột trong cơ sở dữ liệu quan hệ. Giá trị của fields có thể thuộc bất kỳ loại dữ liệu BSON nào như double, string, boolean,...
- MongoDB hỗ trợ việc tạo index cho các field dữ liệu trong collection, giúp tăng tốc độ truy vấn. Chúng còn hỗ trợ sao chép dữ liệu giữa các node trong một cluster giúp đảm bảo tính khả dụng và độ tin cậy của hệ thống.
- MongoDB phân tán dữ liệu trên nhiều node, giúp tăng khả năng mở rộng của hệ thống, đồng thời chúng còn hỗ trợ tính toán phân tán bằng cách sử dụng MapReduce giúp xử lý dữ liệu lớn một cách hiệu quả.

Khi sử dụng MongoDB, bạn có thể sử dụng API và driver của MongoDB để truy cập và thao tác với dữ liệu trong hệ thống cơ sở dữ liệu này.[18]

2.4.3 Ưu điểm và hạn chế

Ưu điểm:

- **Khả năng mở rộng và linh hoạt:** MongoDB mở rộng tốt, xử lý dữ liệu lớn và dễ dàng mở rộng theo chiều ngang. Thiết kế lược đồ linh hoạt giúp điều chỉnh nhanh chóng cho ứng dụng phát triển nhanh.
- **Hiệu suất và tốc độ:** MongoDB tối ưu hóa cho đọc và ghi với thông lượng cao, lý tưởng cho ứng dụng cần xử lý dữ liệu nhanh, nhờ vào lập chỉ mục, sao chép và sharding.
- **Ngôn ngữ truy vấn phong phú:** Ngôn ngữ truy vấn mạnh mẽ cho phép thực hiện các truy vấn phức tạp và biến đổi dữ liệu dễ dàng, nâng cao khả năng phân tích.
- **Tài liệu giống như JSON:** Việc sử dụng BSON cho phép biểu diễn mối quan hệ phân cấp phức tạp, tích hợp tốt với ứng dụng web hiện đại.
- **Hỗ trợ dữ liệu không gian địa lý:** MongoDB hỗ trợ dữ liệu không gian địa lý, rất hữu ích cho ứng dụng dựa trên vị trí như lập bản đồ và truy vấn không gian.
- **Cộng đồng và hệ sinh thái mạnh mẽ:** MongoDB có cộng đồng năng động và hệ sinh thái phong phú với nhiều công cụ, hỗ trợ phát triển ứng dụng.

Hạn chế:

- **Sử dụng bộ nhớ:** Định dạng BSON của MongoDB có thể tiêu tốn nhiều bộ nhớ hơn so với các định dạng lưu trữ khác, gây bất lợi cho những ứng dụng có tài nguyên hạn chế.
- **Giao dịch phức tạp:** Dù hỗ trợ giao dịch ACID đa tài liệu, nhưng tính năng này trong MongoDB còn khá mới và có thể chưa hoàn thiện như trong các hệ quản trị cơ sở dữ liệu quan hệ truyền thống.
- **Vấn đề về tính nhất quán:** MongoDB thiết kế với tính nhất quán cuối cùng, điều này có thể không phù hợp cho các ứng dụng đòi hỏi tính nhất quán mạnh mẽ và chính xác tức thời.
- **Giới hạn lập chỉ mục:** Một số loại cấu trúc dữ liệu và truy vấn có thể không được lập chỉ mục hiệu quả, gây ảnh hưởng đến hiệu suất.

2.4.4 Các tính năng nổi bật của MongoDB

- **Định hướng tài liệu:** MongoDB lưu trữ dữ liệu dưới dạng tài liệu với cặp khóa-giá trị, giúp tăng tính linh hoạt so với các bảng trong RDBMS. Mỗi tài liệu có một ID duy nhất.
- **Cơ sở dữ liệu không có lược đồ:** MongoDB cho phép một bộ sưu tập chứa nhiều loại tài liệu khác nhau mà không yêu cầu phải có cấu trúc giống nhau, mang lại tính linh hoạt cao.
- **Khả năng mở rộng:** MongoDB hỗ trợ mở rộng theo chiều ngang thông qua sharding, cho phép phân phối dữ liệu trên nhiều máy chủ, giúp dễ dàng thêm máy mới vào hệ thống.
- **Lập chỉ mục:** MongoDB tự động lập chỉ mục mọi trường trong tài liệu, giúp tăng tốc độ truy xuất và tìm kiếm dữ liệu so với việc tìm kiếm từng tài liệu không được lập chỉ mục.
- **Tổng hợp:** MongoDB hỗ trợ thực hiện các phép toán tổng hợp trên dữ liệu với ba phương pháp: ống tổng hợp, chức năng giảm ánh xạ, và tổng hợp đơn mục đích.
- **Hiệu suất cao:** Với các tính năng như mở rộng, lập chỉ mục và sao chép, MongoDB có hiệu suất cao và độ bền dữ liệu vượt trội so với các cơ sở dữ liệu khác.[19]

2.4.5 So sánh MongoDB và RDBMS

1.Cấu trúc dữ liệu:

- **MongoDB:** Lưu trữ dữ liệu dưới dạng tài liệu JSON (BSON), cho phép linh hoạt trong cấu trúc và kiểu dữ liệu.
- **RDBMS:** Sử dụng bảng với các hàng và cột có cấu trúc cố định.

2.Tính mở rộng:

- **MongoDB:** Mở rộng ngang (sharding).
- **RDBMS:** Thường mở rộng dọc (vertical scaling).

3.Khả năng xử lý giao dịch:

- **MongoDB:** Hỗ trợ giao dịch tài liệu.
- **RDBMS:** Tính năng ACID[9] mạnh mẽ.

4.Truy vấn dữ liệu:

- **MongoDB:** Ngôn ngữ truy vấn linh hoạt.
- **RDBMS:** Ngôn ngữ SQL chuẩn.

5.Mục đích sử dụng:

- **MongoDB:** Dữ liệu phi cấu trúc, ứng dụng web.
- **RDBMS:** Tính toàn vẹn dữ liệu cao, ứng dụng tài chính.[20]

[9]ACID là các thuộc tính đảm bảo tính toàn vẹn giao dịch trong cơ sở dữ liệu: Atomicity (giao dịch hoàn thành toàn bộ hoặc không), Consistency (dữ liệu luôn ở trạng thái hợp lệ), Isolation (giao dịch hoạt động độc lập), và Durability (thay đổi được lưu trữ vĩnh viễn sau khi hoàn tất).

2.4.6 Công cụ hỗ trợ MongoDB

MongoDB có nhiều công cụ hỗ trợ giúp người dùng dễ dàng quản lý và làm việc với cơ sở dữ liệu. Hai công cụ phổ biến nhất là Robo 3T và MongoDB Compass.[21]

- **Robo 3T**(trước đây là Robomongo) là một công cụ miễn phí mã nguồn mở, cung cấp giao diện đồ họa thân thiện giúp người dùng tương tác với MongoDB. Nó hỗ trợ các chức năng như kết nối, duyệt và chỉnh sửa cơ sở dữ liệu trực quan, phù hợp cho những người cần một công cụ đơn giản và nhẹ nhàng.
- **MongoDB Compass** là công cụ chính thức do MongoDB phát triển. Đây là một ứng dụng GUI (Giao diện người dùng đồ họa) đầy đủ tính năng, cho phép người dùng không chỉ duyệt và chỉnh sửa dữ liệu mà còn phân tích hiệu suất truy vấn và tối ưu hóa cấu trúc cơ sở dữ liệu. Ngoài ra, Compass còn cung cấp khả năng hiển thị trực quan về dạng dữ liệu và mối quan hệ giữa các đối tượng trong MongoDB, hỗ trợ người dùng trong việc kiểm tra và xác thực dữ liệu.

2.4.7 Cách thức thao tác với MongoDB

1. Cài Đặt MongoDB

MongoDB có thể cài đặt trên Windows, macOS và Linux. Người dùng tải bản cài đặt từ trang chủ MongoDB và làm theo hướng dẫn. Sau khi cài đặt, có thể cấu hình MongoDB chạy tự động khi khởi động hệ thống.

2. Kết Nối Tới MongoDB

- MongoDB Shell (mongosh): Sử dụng lệnh mongosh trong terminal để quản lý và thao tác với cơ sở dữ liệu.
- Kết Nối Từ Ứng Dụng: Kết nối qua các ngôn ngữ lập trình như Python (thư viện pymongo) hoặc Node.js (thư viện mongodb) để tương tác với cơ sở dữ liệu.

3. Tạo Cơ Sở Dữ Liệu

Trong Trong MongoDB, không cần lệnh riêng để tạo cơ sở dữ liệu; khi người dùng thêm dữ liệu vào cơ sở dữ liệu chưa tồn tại, MongoDB sẽ tự động tạo nó. Để chọn hoặc tạo cơ sở dữ liệu mới, sử dụng lệnh:

use database_name

Lệnh này sẽ chuyển đến cơ sở dữ liệu database_name và tự động tạo nếu nó chưa tồn tại.

4. Tạo Collection

Collection trong MongoDB lưu trữ các tài liệu. Người dùng có thể tạo Collection bằng lệnh:

db.createCollection("collection_name")

Tuy nhiên, Collection cũng sẽ tự động được tạo khi người dùng chèn dữ liệu vào.

5. Chèn Dữ Liệu

Dữ liệu trong MongoDB được lưu trữ dưới dạng tài liệu BSON. Để chèn tài liệu vào Collection, sử dụng lệnh:

db.collection_name.insert(name: "Linh", age: 20)

Lệnh này sẽ lưu thông tin của một người tên "Linh" và tuổi 20.

6. Truy Vấn Dữ Liệu

MongoDB cho phép truy vấn dữ liệu với nhiều điều kiện khác nhau. Sử dụng lệnh find() để tìm kiếm:

db.collection_name.find(name: "Linh")

Lệnh trên sẽ trả về tất cả các tài liệu có trường name là "Linh".

7. Cập Nhật Dữ Liệu

Người dùng có thể cập nhật tài liệu bằng lệnh *updateOne()* hoặc *updateMany()*:

db.collection_name.updateOne(name: "Linh", $set: age: 22)

Lệnh này sẽ cập nhật tuổi của "Linh" thành 22.

8. Xóa Dữ Liệu

Để xóa dữ liệu, sử dụng lệnh *deleteOne()* hoặc *deleteMany()*:

db.collection_name.deleteOne(name: "Linh")

Lệnh này sẽ xóa tài liệu có tên "Linh" trong Collection.[22]

Chương 3

PHƯƠNG PHÁP THỰC NGHIỆM

3.1 Phương pháp thu thập và lưu trữ dữ liệu

Chúng tôi đã thực hiện quá trình thu thập dữ liệu từ trang web chứng khoán simpilze.vn thông qua việc sử dụng công cụ mã nguồn mở **Selenium** và lưu trữ toàn bộ dữ liệu của trang web vào **MongoDB**. Quá trình này được thực hiện theo các bước sau:

3.1.1 Truy xuất thông tin của trang web

Chúng tôi dùng công cụ mã nguồn mở **Selenium Webdriver** để truy xuất dữ liệu từ trang web chứng khoán simplize.vn.

Bắt đầu bằng việc xác định đường dẫn chứa thông tin dữ liệu lịch sử giá và hồ sơ doanh nghiệp của các mã chứng khoán trên simplize.vn, sử dụng XPath để thu thập dữ liệu từ các cột, hàng chứa thông tin về giá, hồ sơ doanh nghiệp của cổ phiếu.

Các dữ liệu cần thu thập từ trang web bao gồm: ngày, giá cao nhất, giá thấp nhất, giá mở cửa, giá đóng cửa,thay đổi giá, phần trăm thay đổi, và khối lượng giao dịch của cổ phiếu bên cạnh đó là hồ sơ doanh nghiệp.

Sau khi thu thập dữ liệu, thông tin được lưu vào các tệp JSON và import vào MongoDB để quản lý và phân tích.

3.1.2 Thu thập dữ liệu chứng khoán, cổ phiếu

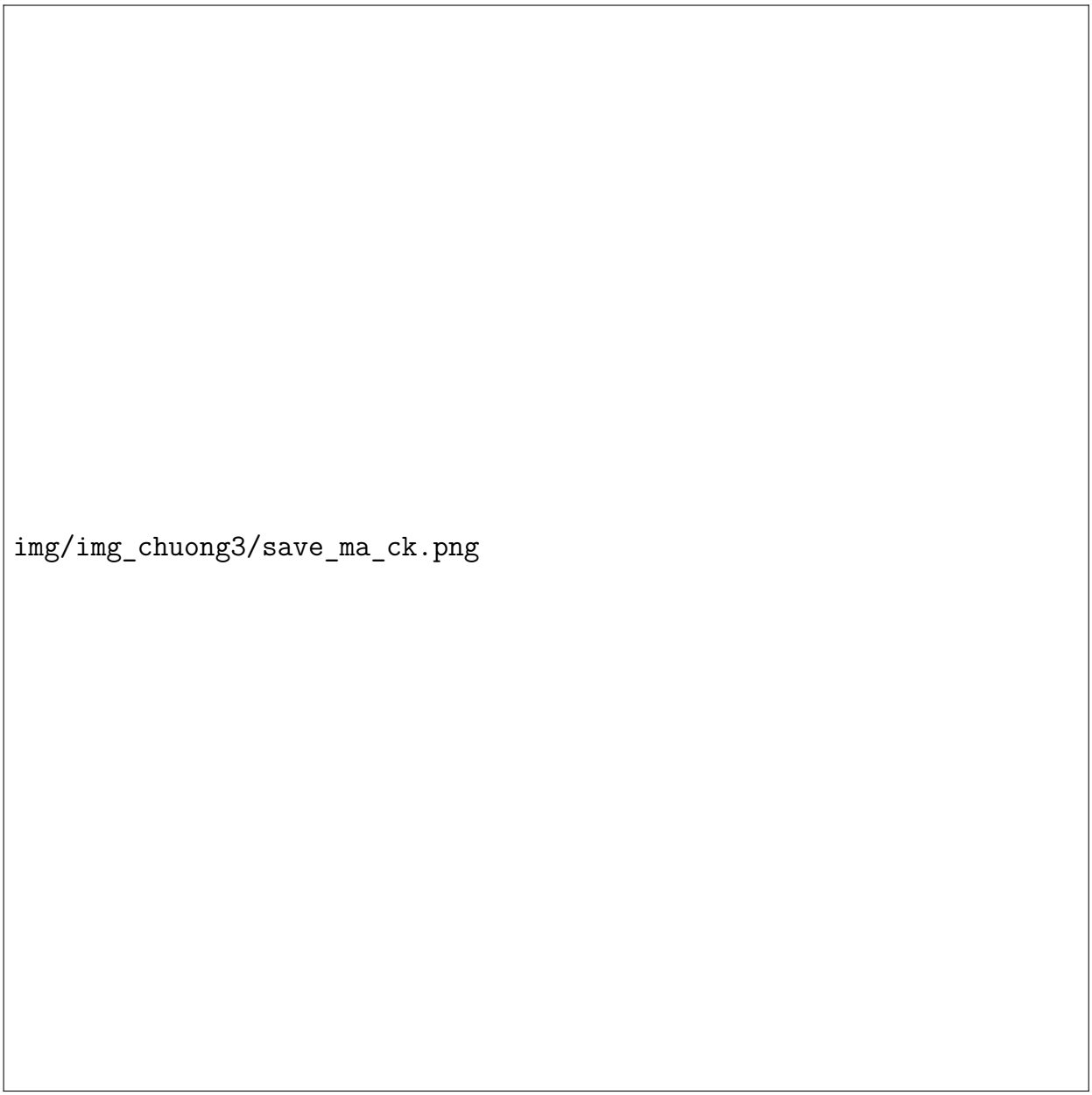
Truy xuất link từng lĩnh vực

Đoạn mã sử dụng Selenium để thu thập tên các lĩnh vực từ trang web và lưu vào MongoDB. Cụ thể,lấy danh sách các lĩnh vực, tạo collection tương ứng trong MongoDB, sau đó chèn dữ liệu tên lĩnh vực vào. Tiếp theo, đoạn mã truy cập vào từng lĩnh vực để thu thập thêm thông tin chi tiết, với mỗi lân truy cập được thực hiện tuần tự.

Hình 3.1: Đoạn mã dùng để truy xuất và thu thập từng lĩnh vực

Truy xuất link từng mã trong từng lĩnh vực cụ thể

Sau khi lấy được các thông tin các lĩnh vực và thêm các lĩnh vực đó làm collection, sau đó truy cập vào từng lĩnh vực và thu thập danh sách mã từ trang web tương ứng với các lĩnh vực bằng cách tìm các phần tử chứa mã, sau đó thêm chúng vào danh sách **ma_list**. Mỗi danh sách mã được in ra và lưu vào danh sách lớn **ma_lists**. Sau khi thu thập xong, mã quay lại trang trước để tiếp tục lấy dữ liệu cho lĩnh vực tiếp theo.



Hình 3.2: Đoạn mã dùng để lưu thông tin danh sách mã chứng khoán.

Thu thập dữ liệu lịch sử giá của các mã

Sau khi thu thập danh sách mã cổ phiếu, đoạn mã tiếp tục duyệt qua từng danh sách mã cổ phiếu trong **ma_lists**, sau đó tạo URL tương ứng cho mỗi mã cổ phiếu để truy cập vào trang lịch sử giá. Sau khi trang được tải, mã thực hiện cuộn xuống để đảm bảo toàn bộ dữ liệu đã được tải. Tiếp theo, dữ liệu giá cổ phiếu như ngày, giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, thay đổi giá, phần trăm thay đổi và khối lượng giao dịch được thu thập từ các hàng trong bảng hiển thị trên trang và lưu vào danh sách **ls_gia_list**. Sau khi thu thập dữ liệu từ một trang, mã tự động nhấn nút "next" để chuyển sang trang tiếp theo, tiếp tục thu thập thêm dữ liệu. Nếu gặp lỗi hoặc không tìm thấy nút chuyển trang, quá trình sẽ dừng lại. Nhờ đó, đoạn mã tự động hóa việc lấy dữ liệu lịch sử giá của nhiều mã cổ phiếu để phục vụ cho phân tích sau này.

Lưu dữ liệu lịch sử giá vào JSON

Sau khi lấy lịch sử giá của các mã thì tiếp tục lưu trữ danh sách dữ liệu lịch sử giá cổ phiếu vào một tệp JSON và quay lại trang trước để tiếp tục thu thập dữ liệu cho mã cổ phiếu tiếp theo. Cụ thể, sau khi thu thập đủ dữ liệu lịch sử giá từ các trang, danh sách **ls_gia_list** sẽ được lưu vào một tệp JSON với tên định dạng **ma_lich_su_gia.json**, trong đó **ma** là mã cổ phiếu tương ứng. Sau khi lưu dữ liệu thành công, lệnh **driver.back()** được thực hiện để quay lại trang trước và tiếp tục quá trình thu thập dữ liệu cho mã cổ phiếu tiếp theo trong danh sách.

Thu thập dữ liệu hồ sơ doanh nghiệp của các mã

Cũng tương tự cách lấy lịch sử giá thì đoạn mã này dùng để thu thập thông tin hồ sơ doanh nghiệp cho từng mã cổ phiếu và lưu trữ chúng vào file JSON. Đầu tiên, mã khởi tạo một danh sách rỗng **ho_so_data_list** để chứa dữ liệu. Sau đó, nó lặp qua từng danh sách mã cổ phiếu trong **ma_lists** và tạo URL cho hồ sơ doanh nghiệp dựa trên mã cổ phiếu. Tiếp theo, mã tìm kiếm các phần tử chứa thông tin hồ sơ doanh nghiệp bằng cách sử dụng **find_elements** với XPath tương ứng. Dữ liệu được lưu vào danh sách **ho_so_list** bằng cách lặp qua các phần tử và thêm vào danh sách. Cuối cùng, danh sách **ho_so_list** được lưu vào file JSON với tên định dạng **ma_ho_so_doanh_nghiep.json**, trong đó **ma** là mã cổ phiếu tương ứng. Nhờ đó, tự động hóa quy trình thu thập và lưu trữ thông tin hồ sơ doanh nghiệp cho nhiều mã cổ phiếu khác nhau.

Hình 3.3: Đoạn mã dùng để thu thập thông tin lịch sử giá của các mã.

Cập nhật dữ liệu lịch sử giá và hồ sơ doanh nghiệp cho từng mã

Cuối cùng đoạn mà dùng để thực hiện việc cập nhật dữ liệu lịch sử giá và hồ sơ doanh nghiệp cho từng mã cổ phiếu trong các collection. Đầu tiên, mã lặp qua từng danh sách mã cổ phiếu trong **ma_lists** và xác định tên lĩnh vực tương ứng từ **collection_list**, sau đó truy cập vào collection này. Tiếp theo, mã kiểm tra sự tồn tại của các file JSON chứa dữ liệu lịch sử giá **(lich_su_gia.json)** và hồ sơ doanh nghiệp **(ho_so_doanh_nghiep.json)**. Nếu các file này tồn tại, dữ liệu sẽ được tải vào các biến tương ứng là **ls_gia_list** và **ho_so_list**. Cuối cùng,

Hình 3.4: Đoạn mã dùng lưu dữ liệu lịch sử giá của mã vào file JSON

đoạn mã sử dụng phương thức **update_one** để cập nhật thông tin trong collection với các trường mới cho lịch sử giá và hồ sơ doanh nghiệp của từng mã cổ phiếu.

3.1.3 Lưu trữ dữ liệu với MongoDB

Sau khi đã hoàn thành việc thu thập dữ liệu, chúng tôi sử dụng MongoDB để lưu trữ cũng như truy vấn dữ liệu cho tương lai, mỗi mã chứng khoán sẽ được lưu vào các collection tương ứng trong cơ sở dữ liệu.

Khi hoàn tất thu thập thì thực hiện đóng kết nối với MongoDB:

Hình 3.5: Đoạn mã dùng để lấy dữ liệu hồ sơ doanh ngiệp của các mã và lưu vào JSON.

client.close()

3.2 Mô tả dữ liệu

Dữ liệu thu thập được từ trang web simplize.vn gồm lịch sử giá của các mã chứng khoán, cổ phiếu. Với các thuộc tính sau:

Hình 3.6: Đoạn mã dùng để cập nhật dữ liệu cho các mã.

3.3 Sơ đổ quá trình thu thập dữ liệu

Tên biến	Mô tả	Kiểu dữ liệu
Ngày	Ngày ghi nhận giao dịch	Date
Giá mở cửa	Giá đầu tiên trong phiên giao dịch	Float
Giá cao nhất	Giá cao nhất trong phiên giao dịch	Float
Giá thấp nhất	Giá thấp nhất trong phiên giao dịch	Float
Giá đóng cửa	Giá cuối cùng trong phiên giao dịch	Float
Thay đổi giá	Sự thay đổi của giá đóng cửa so với ngày trước	Integer
% Thay đổi	Tỷ lệ phần trăm thay đổi so với ngày trước	Float
Khối lượng	Số lượng cổ phiếu giao dịch	Integer

Bảng 3.1: Bảng mô tả các biến và kiểu dữ liệu của chúng.

Hình 3.7: Sơ đồ quá trình thu thập và xử lý dữ liệu

Chương 4

KẾT QUẢ THỰC NGHIỆM

4.1 Giới thiệu

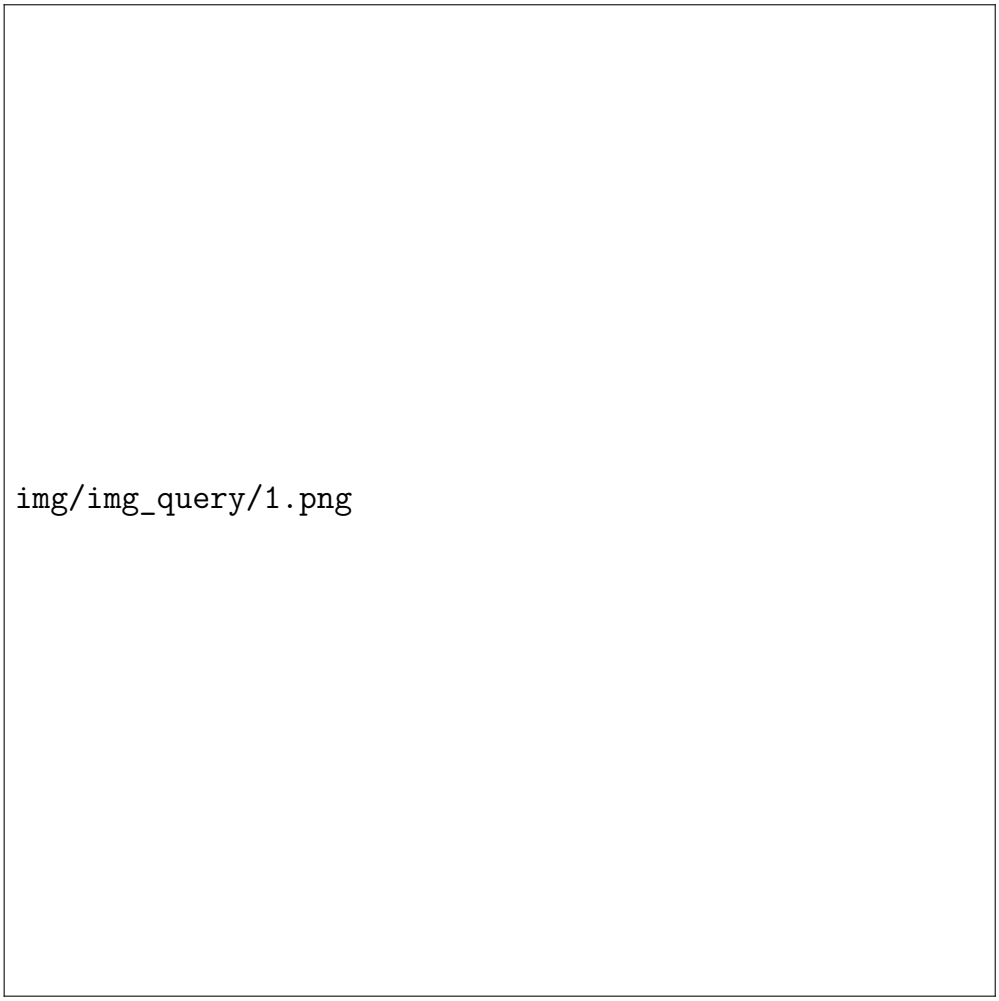
Trong phần này, chúng tôi sẽ mô tả chi tiết các kết quả đạt được từ việc áp dụng công cụ Selenium để thu thập dữ liệu từ trang web chứng khoán simplize.vn. Quá trình thực hiện, từ thu thập đến phân tích dữ liệu, được trình bày cụ thể cùng với các kết quả thu thập được. Nhờ các dữ liệu này, nhóm nghiên cứu đã phân tích xu hướng cổ phiếu và đánh giá mức độ hiệu quả của việc sử dụng Selenium trong việc tự động hóa quy trình thu thập dữ liệu.

4.2 Kết quả thu thập dữ liệu

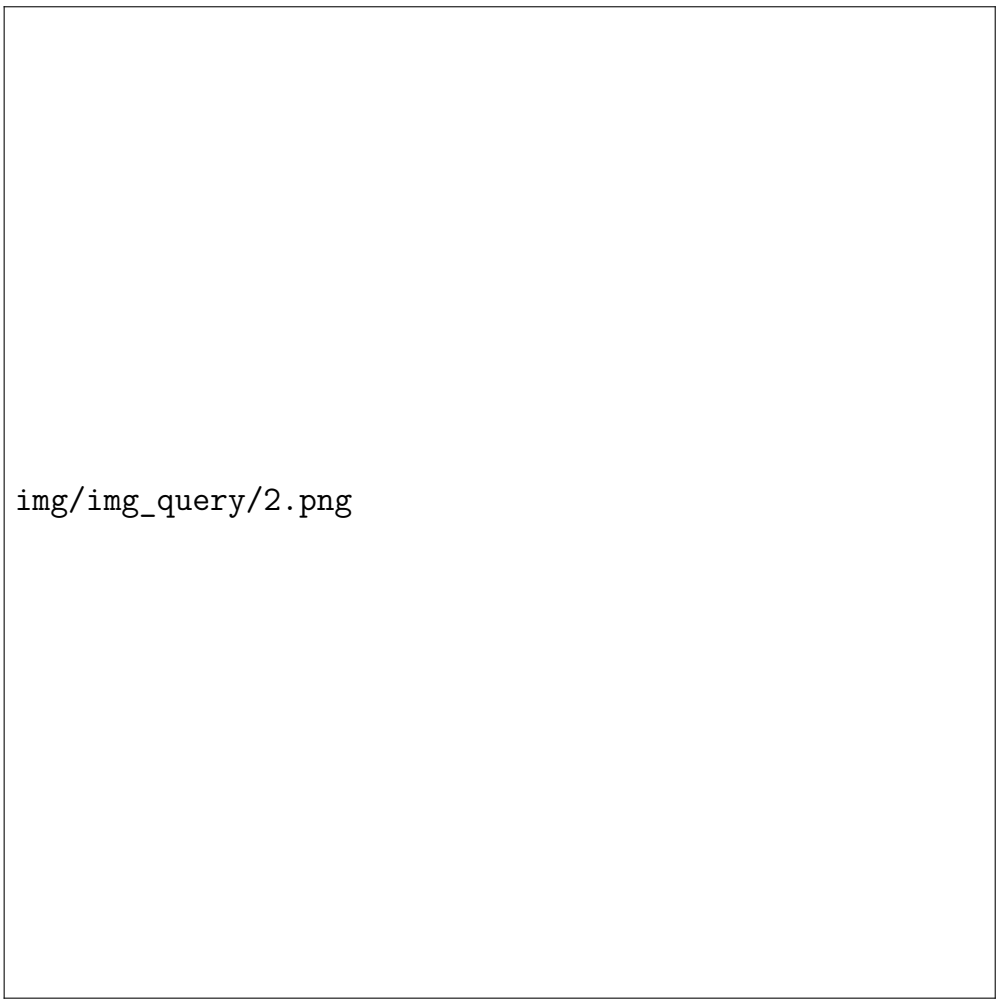
Dữ liệu được thu thập từ trang web simpile.vn bao gồm các thông tin như: ngày, giá mở cửa, giá cao nhất, giá thấp nhất, giá đóng cửa, thay đổi giá, % thay đổi, khối lượng và hồ sơ doanh nghiệp. Sau khi hoàn tất quá trình thu thập, tổng số mã lấy được là **80 mã** trong nhóm ngành tài chính và số ngày lấy được của từng mã là **60 ngày**.

Dữ liệu sẽ được đưa vào MongoDB và được lưu trữ để phục vụ các bước phân tích cho tương lai.

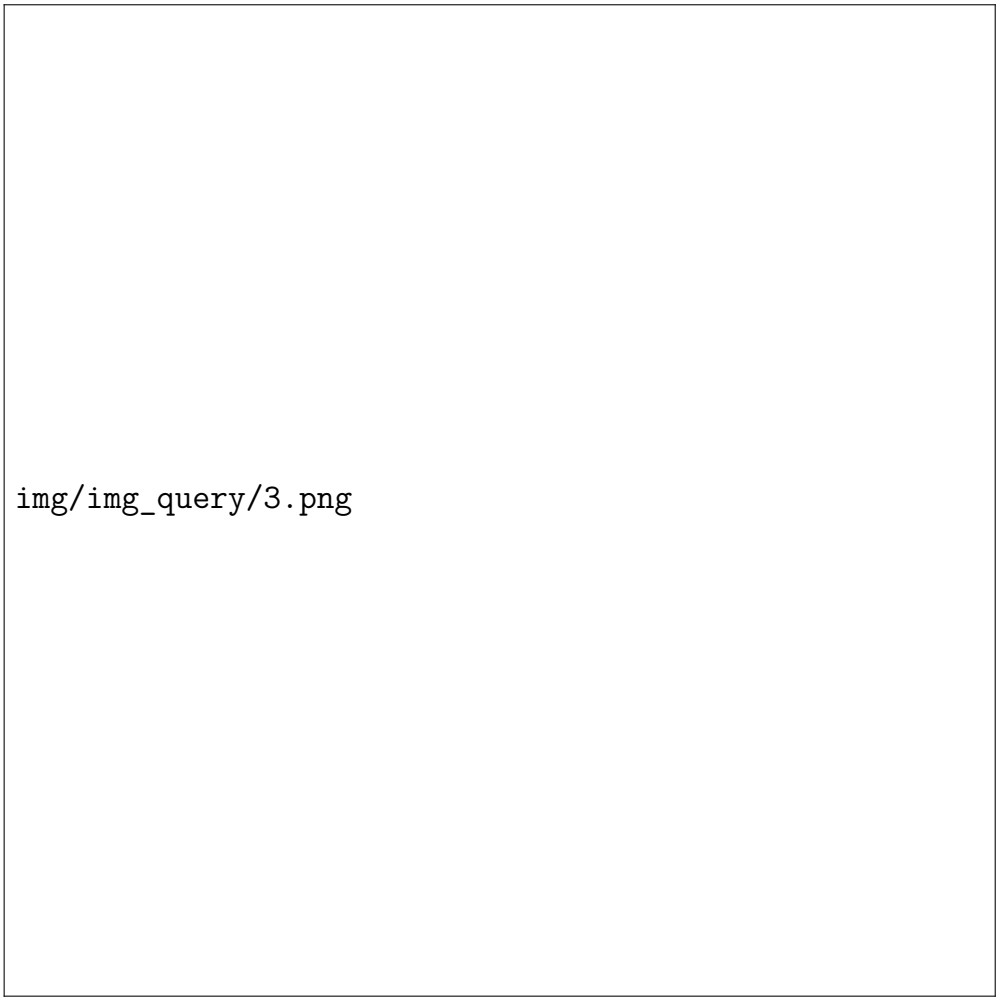
4.3 Phân tích dữ liệu



Hình 4.1: In ra toàn bộ

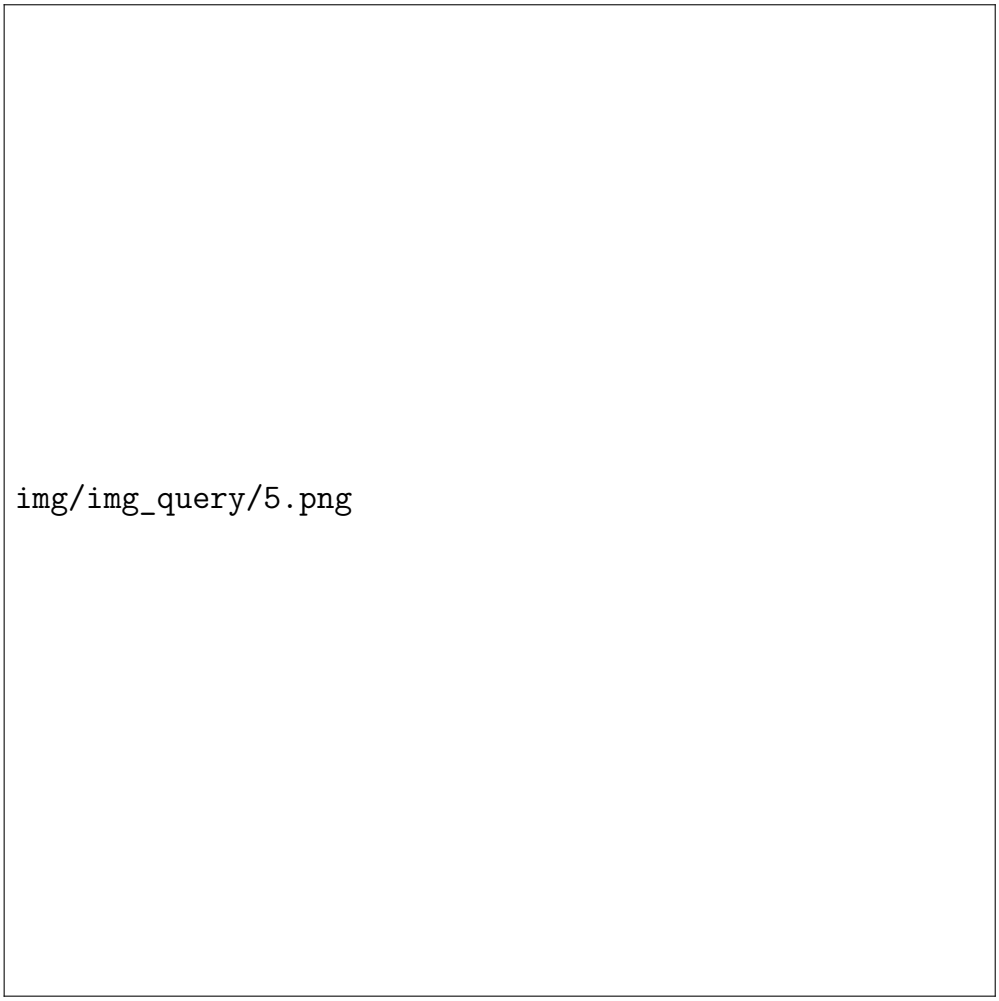


Hình 4.2: in ra tên của tất cả các mã trong tài chính ngân hàng

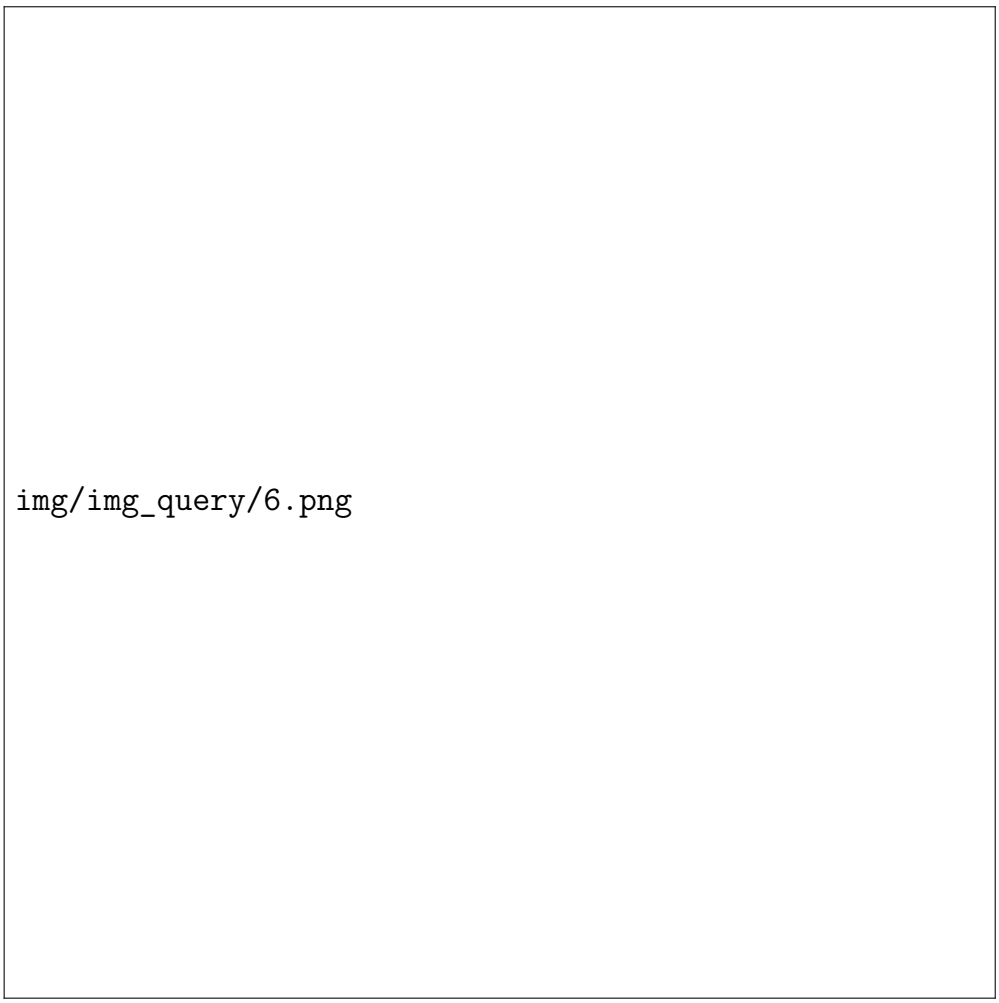


Hình 4.3: lấy dữ liệu của giá cổ phiếu của các mã trong Tài chính ngân hàng

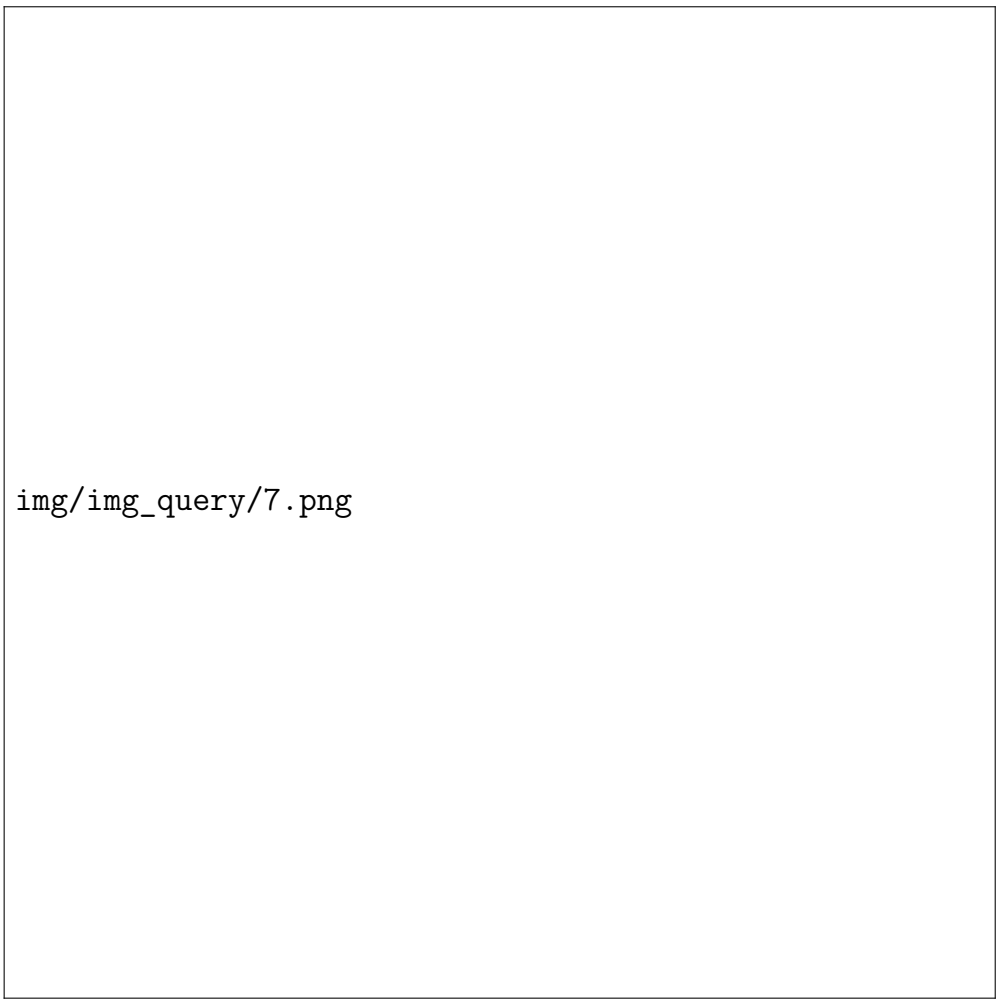
Hình 4.4: đếm số lượng data có trong tài chính ngân hàng



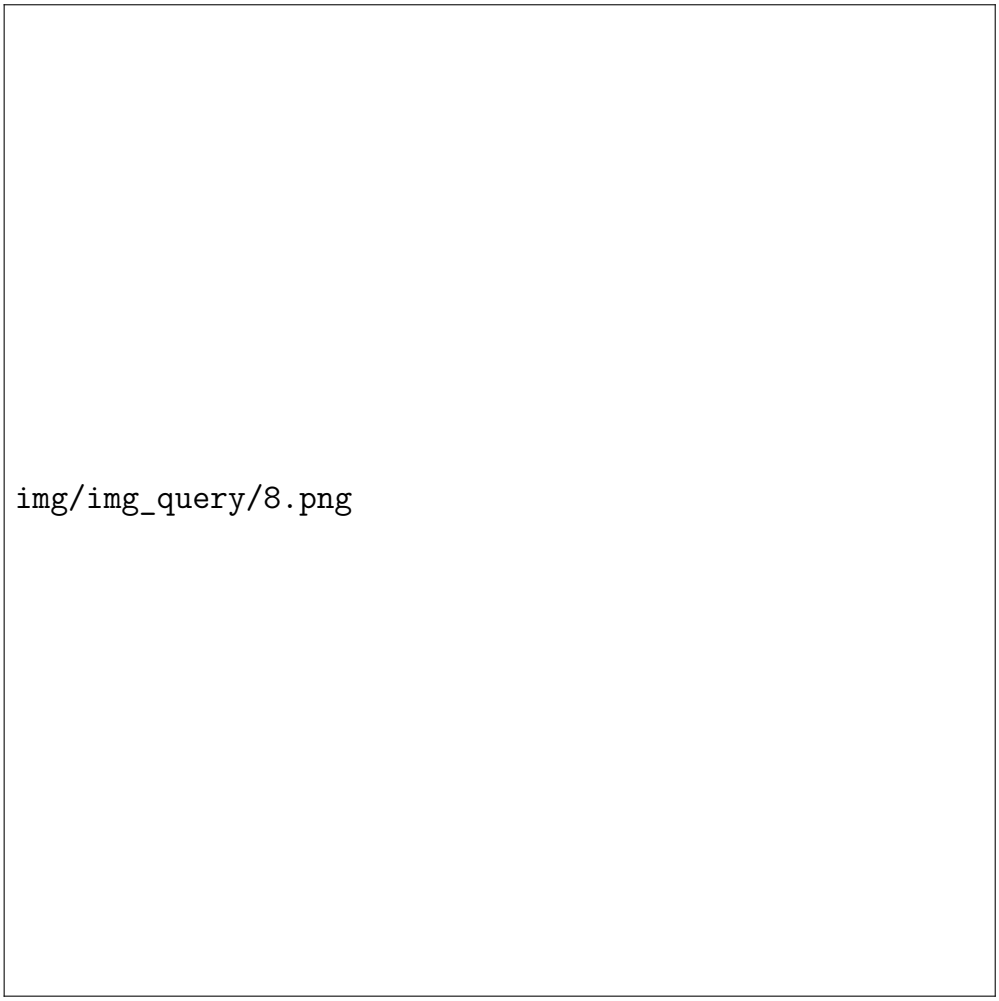
Hình 4.5: đếm tổng volume của một mã bất kì



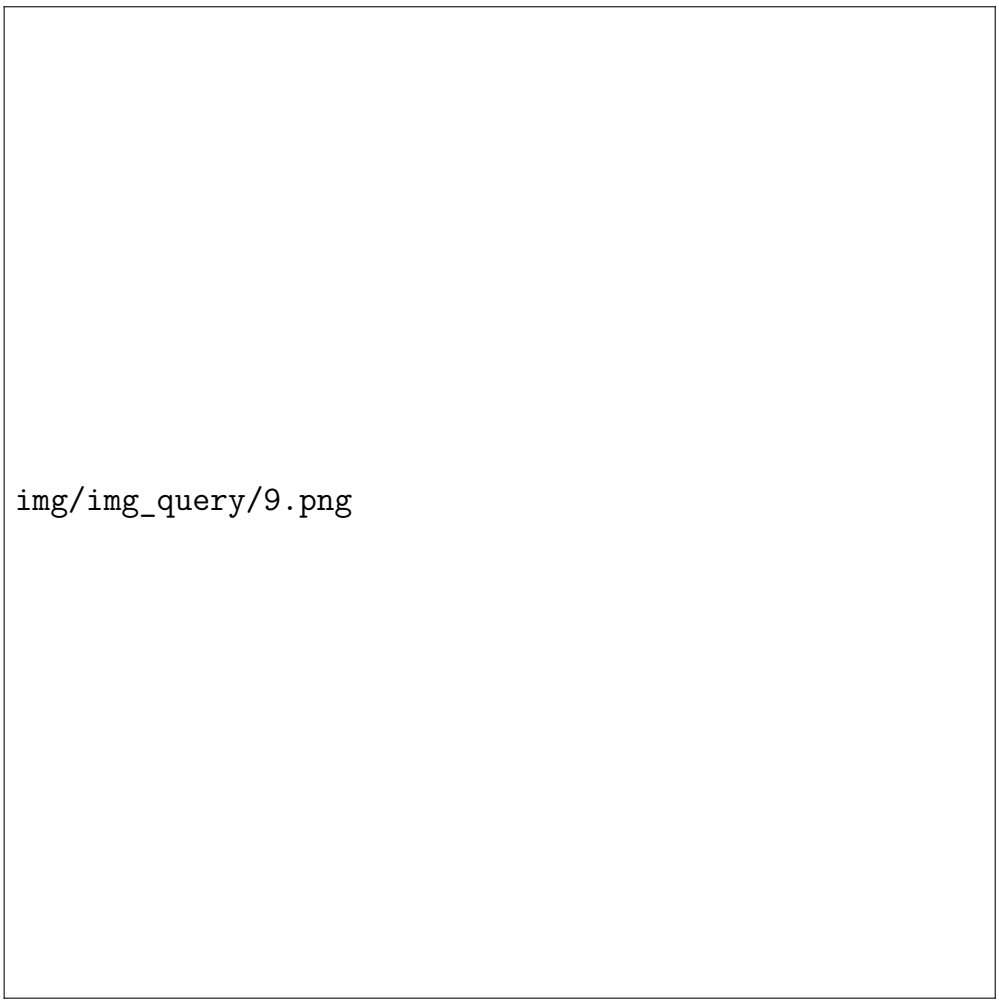
Hình 4.6: đếm tổng volume của một mã bất kì



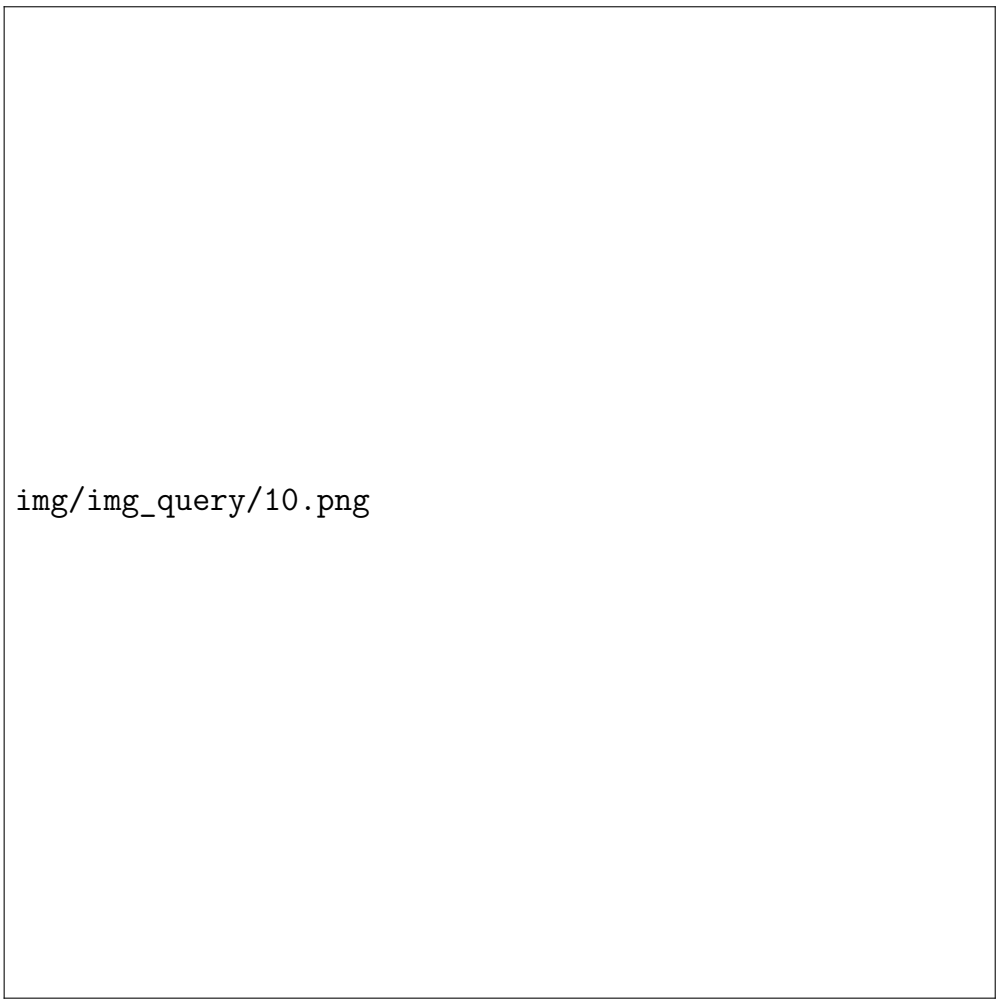
Hình 4.7: tìm giá mở cửa lớn nhất trong tất cả các mã(mã bị lỗi khi thay thành giá mở cửa do có rỗng)



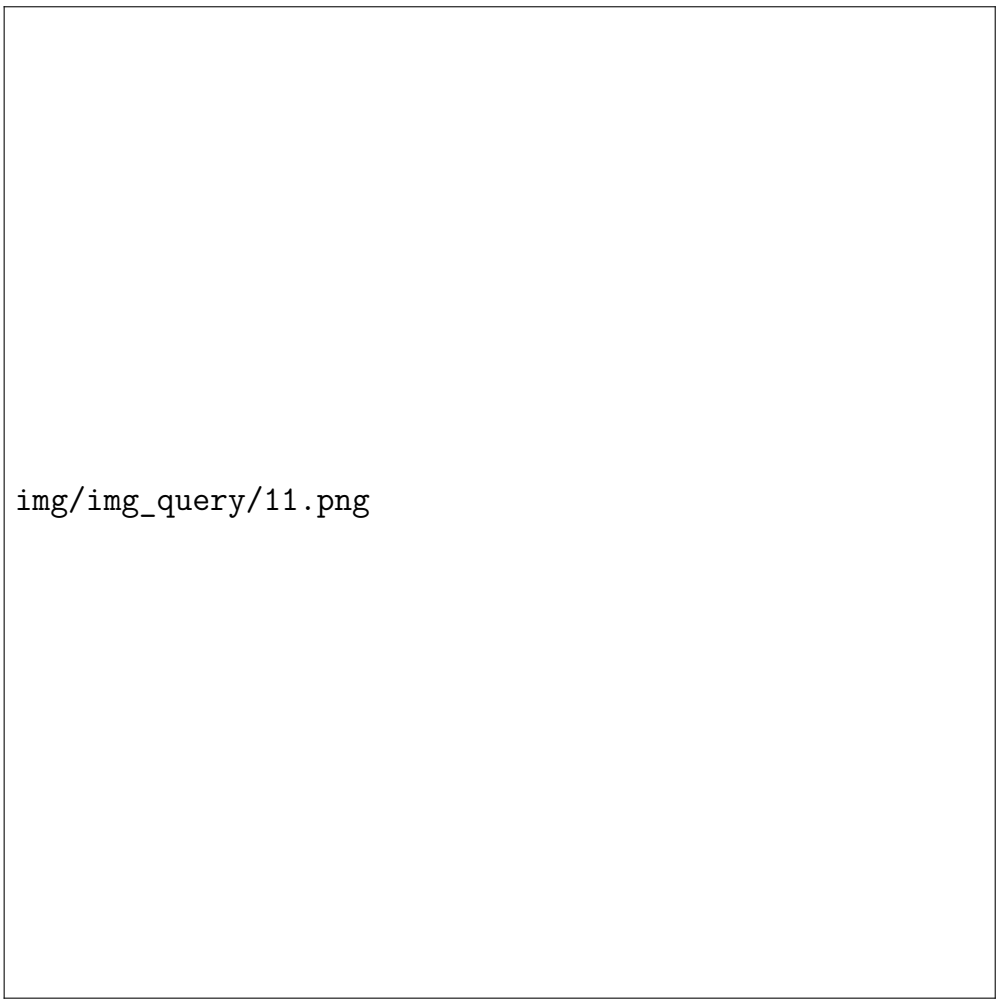
Hình 4.8: lấy % thay đổi nhỏ nhất của một mã bất kì



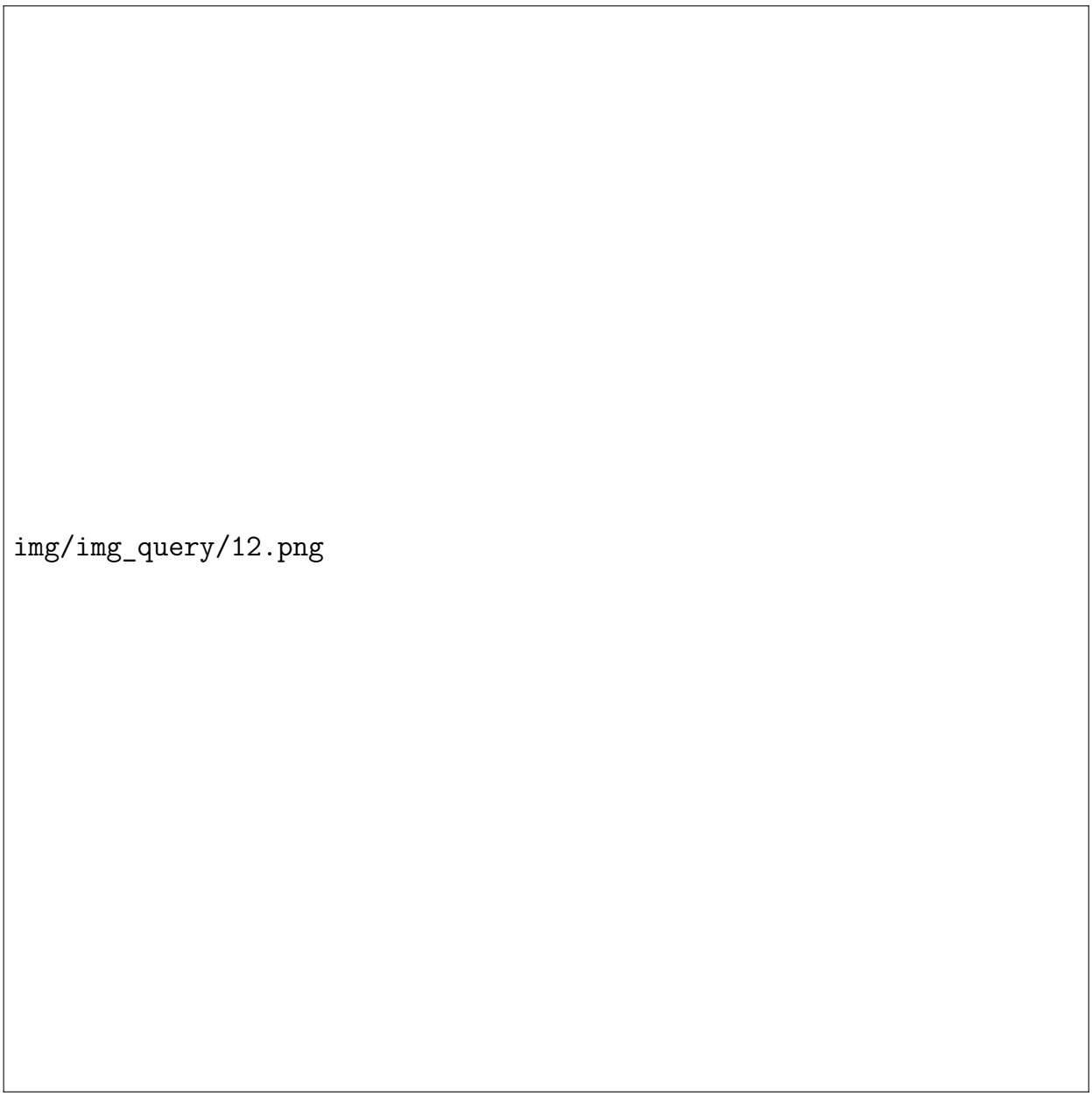
Hình 4.9: in ra sự chêch lệch khối lượng giữa 2 data có ngày gần nhất trong một cỗ phiếu



Hình 4.10: in ra sự chêch lệch khối lượng giữa 2 data có ngày bất kì trong một cỗ phiếu



Hình 4.11: lấy khối lượng lớn nhất cho hai mã bất kì và tính sự chêch lệch khối lượng



Hình 4.12: xuất tên các mã có dữ liệu từ ngày nào tới ngày nào

4.4 Đánh giá hiệu suất Selenium

Qua quá trình thực nghiệm, chúng tôi đã đánh giá hiệu quả của Selenium trong việc thu thập dữ liệu từ trang web simplize.vn. Kết quả cho thấy Selenium là một công cụ mạnh mẽ và linh hoạt, phù hợp cho việc thu thập dữ liệu từ các trang có nội dung động và yêu cầu tương tác với người dùng.

- **Hiệu suất:**Selenium có khả năng thu thập dữ liệu từ 80 mã chứng khóa với tốc độ khoảng 1 mã/phút, giúp tiết kiệm đáng kể thời gian so với phương pháp thu thập thủ công. Toàn bộ quá trình thu thập dữ liệu cho 80 mã tốn khoảng 70 phút.

- **Khả năng xử lý lỗi:**Trong quá trình thu thập, Selenium có thể tự động xử lý các sự cố về kết nối và các yêu cầu không thành công. Tỷ lệ lỗi phát sinh rất thấp, chỉ khoảng 0.1% trên tổng số yêu cầu được gửi đi.
- **Tính chính xác:**Dữ liệu được thu thập có độ chính xác cao, với tỷ lệ thành công vượt quá 98% mà không gặp phải lỗi nào.

4.5 Các khó khăn và hạn chế

Mặc dù Selenium đã mang lại hiệu quả cao trong việc thu thập dữ liệu, chúng tôi cũng gặp phải một số thách thức đáng lưu ý:

- **Tốc độ tải trang:** Do Selenium tương tác trực tiếp với trình duyệt và phải xử lý toàn bộ nội dung trang, quá trình thu thập dữ liệu từ những trang web có lượng nội dung lớn hoặc tốc độ tải chậm có thể kéo dài đáng kể, làm giảm hiệu suất so với các công cụ nhẹ hơn như Scrapy.
- **Quản lý phiên (session):** Một số trang web yêu cầu duy trì trạng thái phiên liên tục, và điều này đòi hỏi việc thiết lập phiên trong Selenium phải được thực hiện chính xác. Nếu không, quá trình thu thập có nguy cơ bị gián đoạn, dẫn đến mất mát hoặc sai sót dữ liệu.

4.6 Kết luận

Chương này đã trình bày các kết quả thực nghiệm thu thập và phân tích dữ liệu từ trang web chứng khoán simplize.vn bằng công cụ Selenium. Kết quả cho thấy Selenium là một công cụ mạnh mẽ cho việc tự động thu thập dữ liệu từ các trang web động, đặc biệt là khi xử lý nội dung được tải qua JavaScript. Mặc dù quá trình thu thập có thể chậm hơn do phải tải toàn bộ trang, nhưng lợi ích của Selenium nằm ở khả năng xử lý các trang web phức tạp và đảm bảo thu thập đầy đủ dữ liệu, điều mà các công cụ khác khó thực hiện. Vì vậy, mặc dù thời gian thực thi có thể dài hơn, Selenium vẫn mang lại hiệu quả cao trong các tình huống cần tương tác trực tiếp với nội dung trang.

Chương 5

KẾT LUẬN VÀ KIẾN NGHỊ

5.1 Kết luận

Trong thời đại công nghệ số, việc thu thập và phân tích dữ liệu là yếu tố cốt lõi giúp các doanh nghiệp đưa ra quyết định kinh doanh hiệu quả. Thông qua đề tài này, nhóm đã nghiên cứu và ứng dụng công cụ mã nguồn mở Scrapy để tự động thu thập và phân tích dữ liệu từ trang web bán hàng trực tuyến. Kết quả thực nghiệm đã cho thấy hiệu quả của Scrapy trong việc thu thập thông tin một cách nhanh chóng, chính xác và tiết kiệm nguồn lực.

Kết quả thực nghiệm cho thấy Selenium là một công cụ mạnh mẽ, cho phép thu thập dữ liệu từ các trang web động một cách nhanh chóng và chính xác, đồng thời tiết kiệm đáng kể nguồn lực. Dưới đây là những kết luận chính của đề tài:

- **Khả năng thu thập dữ liệu tự động:**Selenium đã chứng tỏ được tính hiệu quả trong việc tự động hóa quy trình thu thập dữ liệu từ các trang web có cấu trúc động, nơi mà nội dung có thể thay đổi theo thời gian. Việc sử dụng Selenium giúp giảm thiểu thời gian và công sức so với các phương pháp thu thập dữ liệu thủ công.
- **Phân tích xu hướng thị trường:** Dữ liệu thu thập từ trang web chứng khoán Simplize đã được xử lý và phân tích thành công. Nhóm nghiên cứu đã xác định được các xu hướng quan trọng trong thị trường chứng khoán dựa trên các chỉ số tài chính, biến động giá cổ phiếu và đánh giá từ người dùng. Kết quả này cung cấp cái nhìn sâu sắc về các cổ phiếu tiềm năng, từ đó hỗ trợ doanh nghiệp trong việc điều chỉnh chiến lược đầu tư.

- **Ứng dụng thực tiễn:**Dữ liệu thu thập được từ Selenium có thể được áp dụng trong nhiều lĩnh vực khác nhau, bao gồm phân tích tài chính, dự đoán xu hướng thị trường và phát triển các chiến lược đầu tư hiệu quả. Công cụ này không chỉ giúp các doanh nghiệp tiết kiệm chi phí mà còn nâng cao khả năng cạnh tranh trên thị trường tài chính.

Mặc dù Selenium mang lại nhiều lợi ích, nhưng nó cũng có một số hạn chế khi phải xử lý các trang web có cơ chế bảo mật phức tạp hoặc yêu cầu duy trì phiên đăng nhập lâu dài. Để giải quyết vấn đề này, cần kết hợp sử dụng với các công cụ khác như Beautiful Soup hoặc các phương pháp tối ưu hóa quy trình thu thập dữ liệu.

5.2 Kiến nghị

Từ những kết quả từ quá trình thực nghiệm và phân tích, chúng tôi đưa ra một số kiến nghị để nâng cao hiệu quả ứng dụng của Selenium trong việc thu thập dữ liệu như:

1. Kết hợp với Scrapy hoặc BeatufulSoup: Đối với các trang web động sử dụng JavaScript, nên tích hợp Selenium với Scrapy hoặc BeautifulSoup để thu thập dữ liệu một cách hiệu quả hơn, mở rộng khả năng thu thập từ các trang web động.

2. Tối ưu hóa hiệu suất: Cần tối ưu hóa mã và quy trình thu thập của Selenium, bao gồm việc cải thiện thời gian phản hồi và giảm thiểu lỗi để đảm bảo tốc độ và độ chính xác trong quá trình thu thập.

3. Phát triển mô hình dự báo:Sử dụng dữ liệu thu thập được để xây dựng các mô hình dự báo xu hướng tiêu dùng, từ đó hỗ trợ doanh nghiệp trong việc ra quyết định nhanh chóng và hiệu quả.

4. Nâng cao khả năng quản lý dữ liệu lớn: Khuyến khích sử dụng cơ sở dữ liệu phi cấu trúc như MongoDB để quản lý dữ liệu thu thập và áp dụng các công cụ phân tích mạnh mẽ như Pandas và NumPy để xử lý dữ liệu một cách nhanh chóng.

5. Đào tạo nhân lực: Đầu tư vào đào tạo nhân viên về lập trình Python và hiểu biết về cách thức hoạt động của các công cụ thu thập dữ liệu mã nguồn mở nhằm nâng cao khả năng cạnh tranh và phát triển bền vững trong lĩnh vực này.

Tài liệu tham khảo

[1] *Web scraping*. URL: https://en.wikipedia.org/w/index.php?title=Web_scraping&oldid=1250978982.

[2] *1920px-WebCrawlerArchitecture.svg.png (1920×1467)*. URL: https://upload.wikimedia.org/wikipedia/commons/thumb/d/df/WebCrawlerArchitecture.svg/1920px-WebCrawlerArchitecture.svg.png.

[3] *Web Scraping có hợp pháp không? 2023 - IPBurger.com*. URL: https://www.ipburger.com/vi/blog/is-web-scraping-legal/.

[4] *What is Selenium? A Complete Guide on Selenium Testing*. URL: https://www.lambdatest.com/selenium.

[5] *WebDriver.png (379×372)*. URL: https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjdcFpKOnInGqRiiL7Pl7yCzVsPODQ_TPEf23mM-Ds9d2U_71ek-K9D1vDdXSQTwhdzaeCHdiGDDrtnTnsnjdIn9_eqqEeHjjoT9GSx2blGxyCbis9qt-FFB7YmGty/s1600/WebDriver.png.

[6] *Selenium là gì? Tổng quan những thông tin cần biết về Selenium*. URL: https://bizflycloud.vn/tin-tuc/selenium-la-gi-20220328105303215.htm.

[7] *Selenium (software)*. en. URL: https://en.wikipedia.org/w/index.php?title=Selenium_(software)&oldid=1248044578.

[8] *0*aqchwFAvns_8ml69 (602×273)*. URL: https://miro.medium.com/v2/resize:fit:828/format:webp/0*aqchwFAvns_8ml69.

[9] *What is Selenium WebDriver? [Complete Guide]*. en. URL: https://hackr.io/blog/what-is-selenium-webdriver.

[10] *1570190913rXish5jdLA.jpg (910×542)*. URL: https://cdn.hackr.io/uploads/posts/attachments/1570190913rXish5jdLA.jpg.

[11] *(20) Selenium, Advantages & Disadvantages | LinkedIn*. URL: https://www.linkedin.com/pulse/selenium-advantages-disadvantages-mahenderkar-sandeep-ftqqc/.

[12] *The Selenium Browser Automation Project*. en. URL: https://www.selenium.dev/documentation/.

[13] WebscrapingAPI. *Scrapy vs. Selenium: A Comprehensive Guide to Choosing the Best Web Scraping Tool - WebScrapingAPI*. en. Aug. 2023. URL: https://www.webscrapingapi.com/scrapy-vs-selenium.

[14] *Introduction to NoSQL*. vi. Section: DBMS. 2018. URL: https://www.geeksforgeeks.org/introduction-to-nosql/.

[15] *What is MongoDB? Features and how it works – TechTarget Definition*. URL: https://www.techtarget.com/searchdatamanagement/definition/MongoDB.

[16] *MongoDB*. vi. Apr. 2023. URL: https://vi.wikipedia.org/w/index.php?title=MongoDB&oldid=69892072#cite_note-6.

[17] *mongodb-la-gi-1.jpeg (852×491)*. URL: https://statics.cdn.200lab.io/2023/04/mongodb-la-gi-1.jpeg.

[18] *MONGODB LÀ GÌ? TÍNH NĂNG NỔI BẬT CỦA MONGODB MÀ BẠN CẦN BIẾT*. en. Apr. 2023. URL: https://200lab.io/blog/mongodb-la-gi/.

[19] *Top Features of MongoDB*. en. June 2021. URL: https://www.boardinfinity.com/blog/top-features-of-mongodb/.

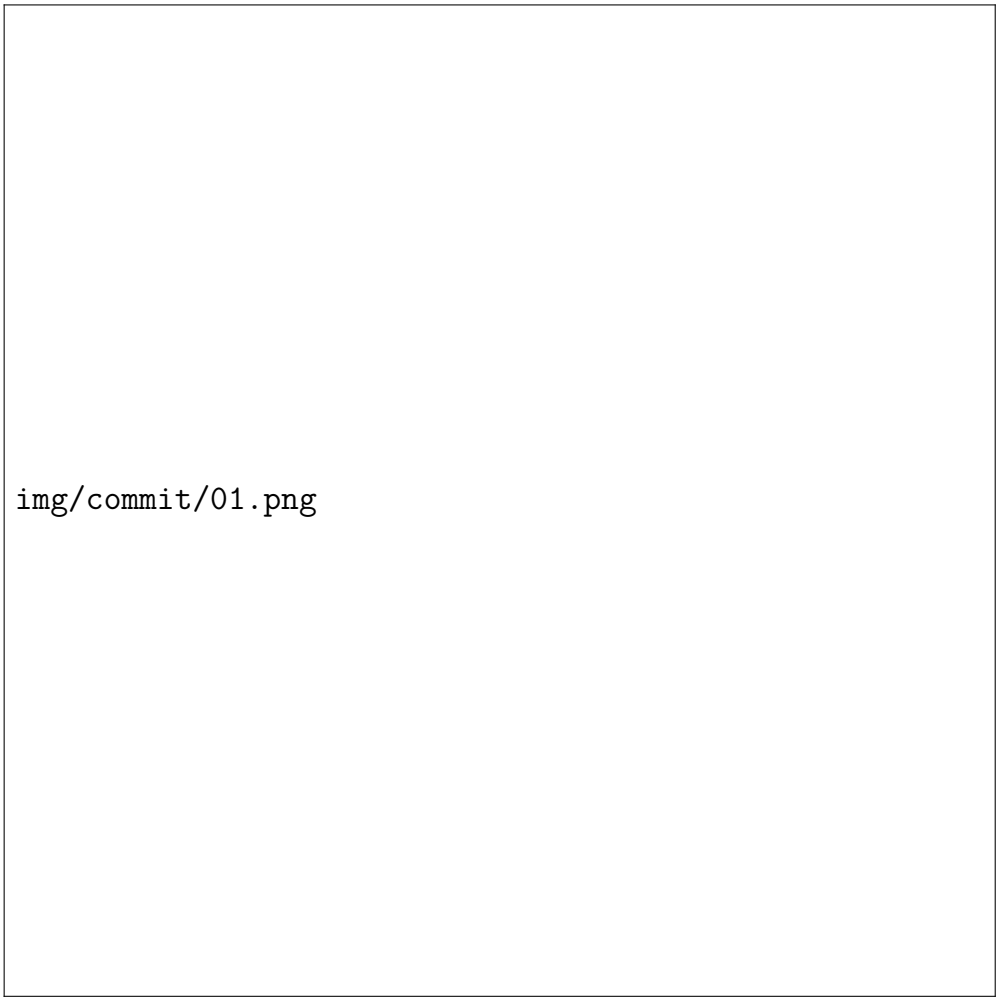
[20] *Difference between RDBMS and MongoDB*. vi. Section: DBMS. Sept. 2019. URL: https://www.geeksforgeeks.org/difference-between-rdbms-and-mongodb/.

[21] *MongoDB là gì? 9 Phần mềm quản trị Mongodb nên sử dụng 2024*. vi-VN. Section: Website. URL: https://prodima.vn/mongodb-la-gi/.

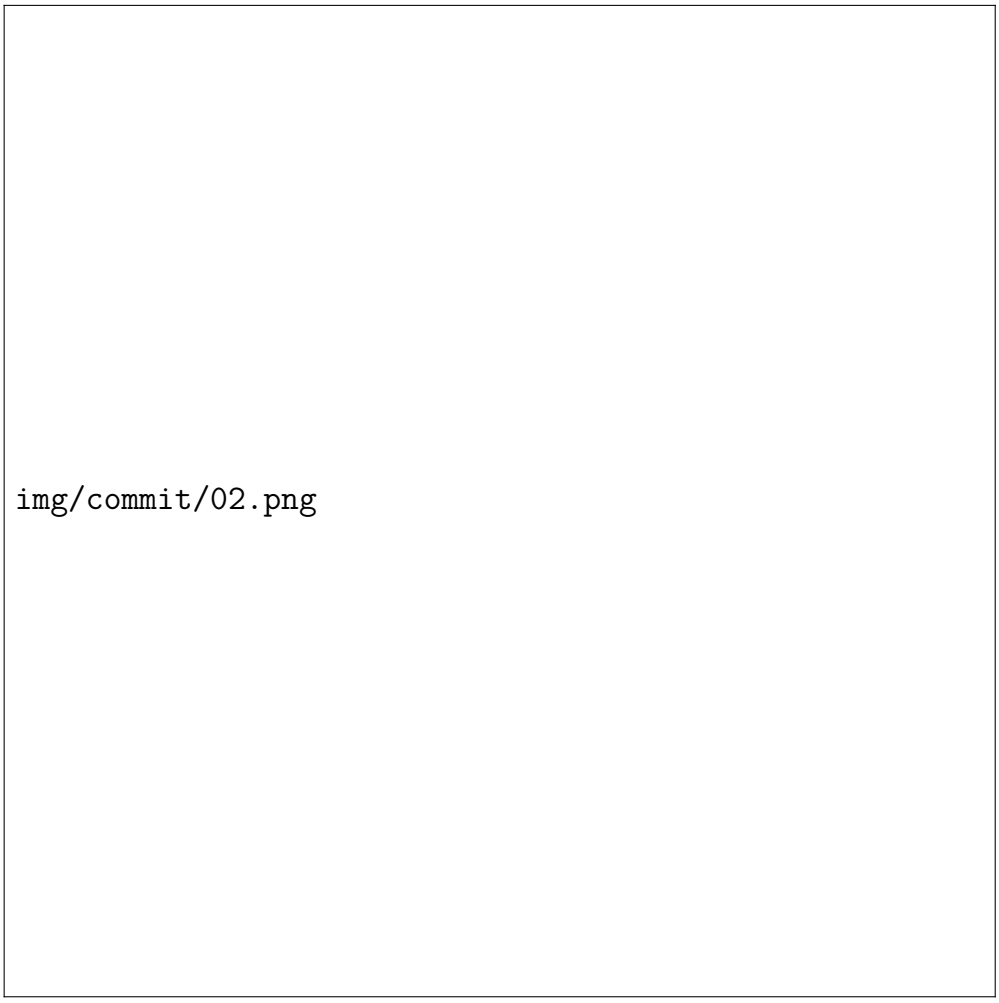
[22] *How to get started with MongoDB in 10 minutes*. URL: https://www.freecodecamp.org/news/learn-mongodb-a4ce205e7739/.

PHỤ LỤC

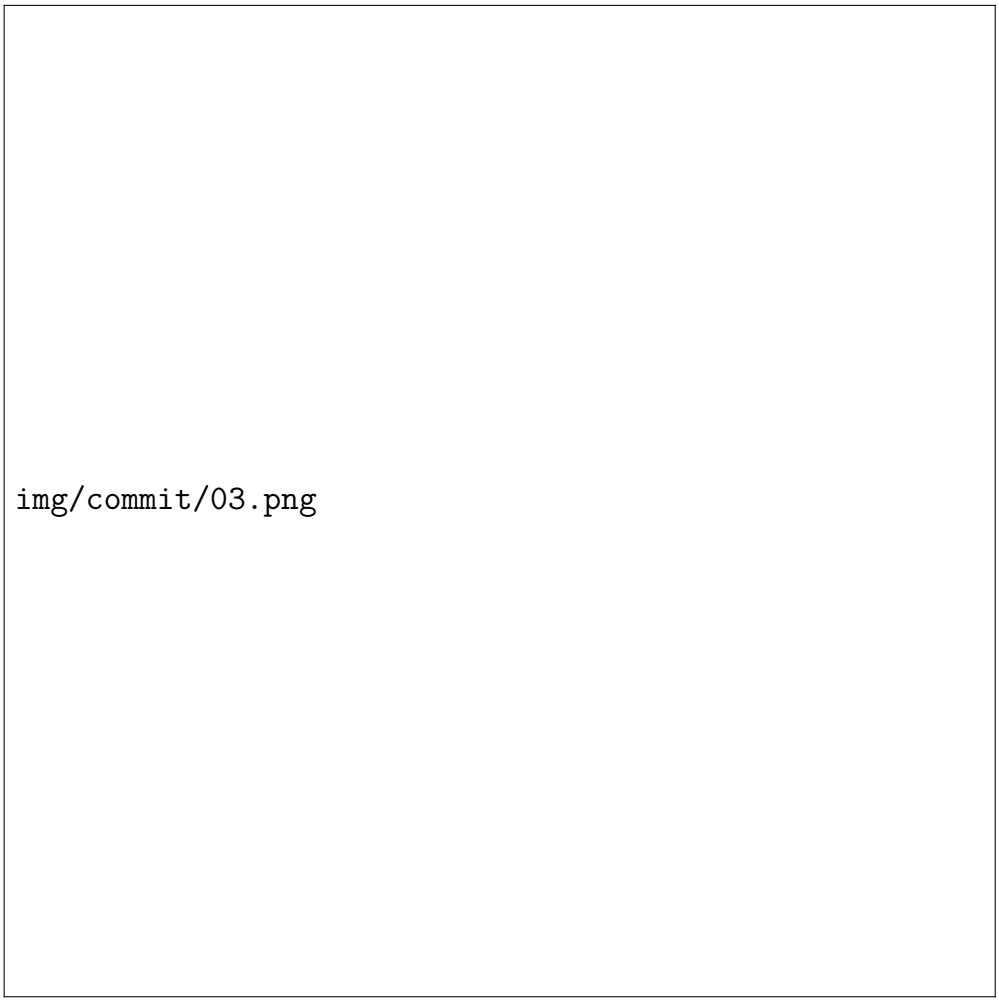
.1 History commit



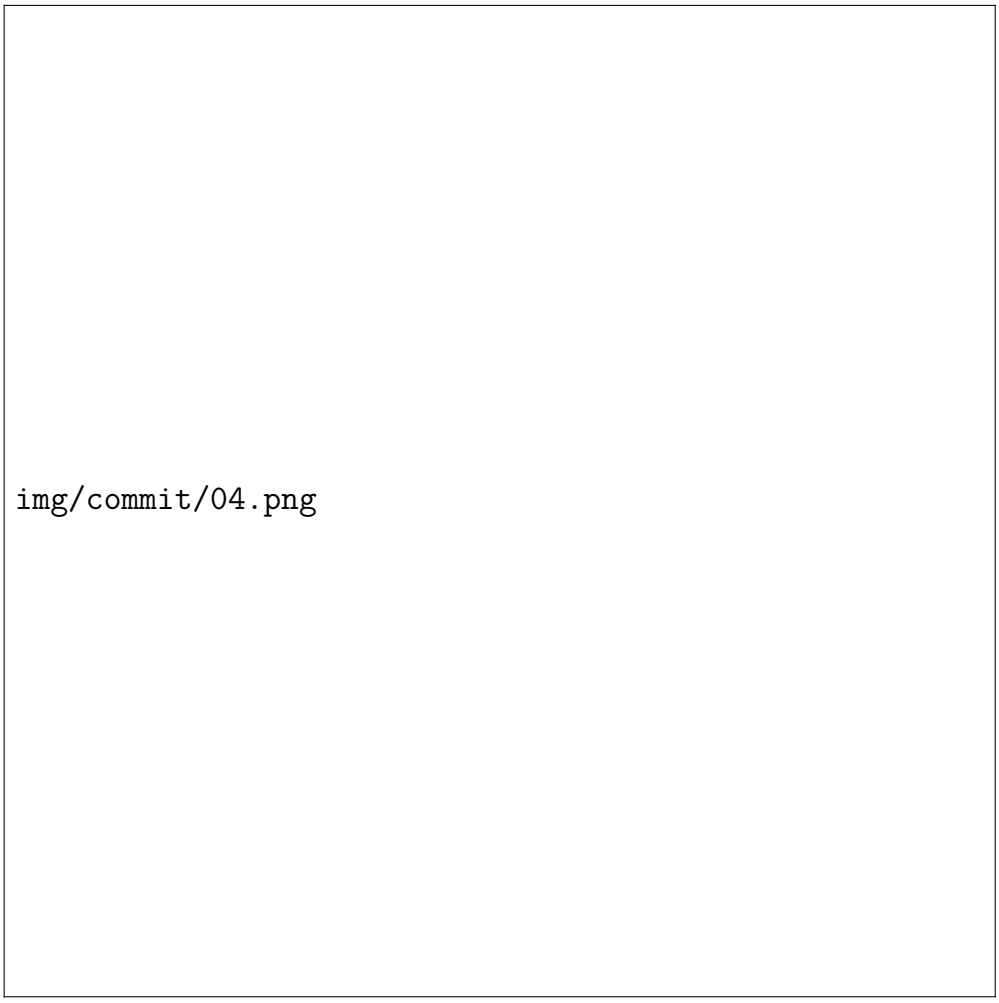
Hình 1: commit_01



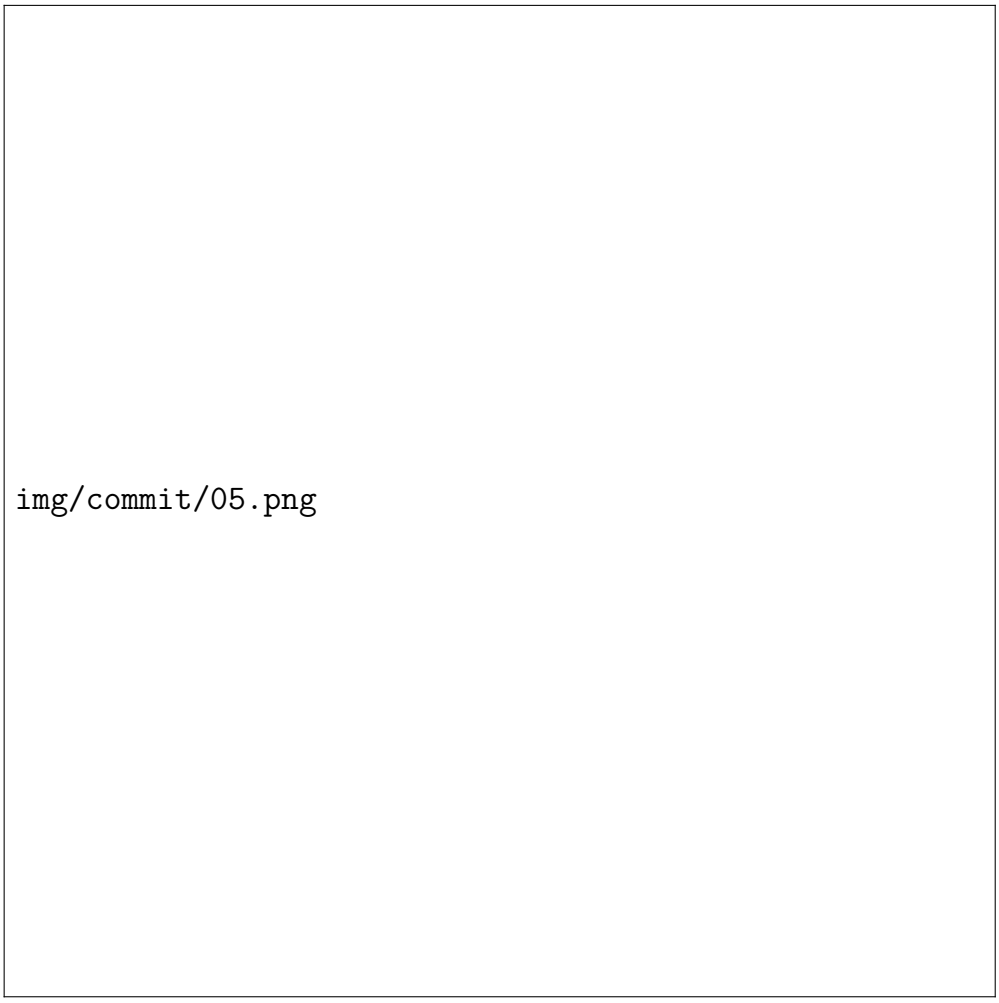
Hình 2: commit_02



Hình 3: commit_03



Hình 4: commit_04



Hình 5: commit_05