|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC**

**THU THẬP VÀ PHÂN TÍCH DỮ LIỆU DƯỢC PHẨM TỪ WEBSITE PHARMACITY.VN SỬ DỤNG CÔNG CỤ MÃ NGUỒN MỞ SELENIUM WEBDRIVER VÀ MONGODB**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : THS. LÊ NHẬT TÙNG

Sinh viên thực hiện :

Hà Thế Anh MSSV: 2286400002

Nguyễn Nhật Nam MSSV: 2286400019

Hoàng Quang Minh MSSV: 2286400017

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2024

|  |  |
| --- | --- |
| logo (CMYK)-01 | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM** |

**ĐỒ ÁN MÔN HỌC**

**THU THẬP VÀ PHÂN TÍCH DỮ LIỆU DƯỢC PHẨM TỪ WEBSITE PHARMACITY.VN SỬ DỤNG CÔNG CỤ MÃ NGUỒN MỞ SELENIUM WEBDRIVER VÀ MONGODB**

Ngành: **KHOA HỌC DỮ LIỆU**

Môn học: **MÃ NGUỒN MỞ TRONG KHOA HỌC DỮ LIỆU**

Giảng viên hướng dẫn : THS. LÊ NHẬT TÙNG

Sinh viên thực hiện :

Hà Thế Anh MSSV: 2286400002

Nguyễn Nhật Nam MSSV: 2286400019

Hoàng Quang Minh MSSV: 2286400017

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2024

# LỜI CAM ĐOAN

Chúng tôi, Hà Thế Anh, Hoàng Quang Minh và Nguyễn Nhật Nam xin cam đoan rằng:

Mọi thông tin và nghiên cứu được trình bày trong bài báo cáo này là trung thực và khách quan được thu thập và phân tích một cách cẩn thận dựa trên các nguồn chính thống và đáng tin cậy.

Bất kỳ thông tin hoặc ý kiến nào được trích dẫn từ các nguồn khác đều được nêu rõ nguồn gốc và được trích dẫn theo đúng quy định. Chúng tôi xin cam đoan rằng không có bất kỳ sự sao chép hoặc sử dụng thông tin không đúng đắn nào từ các nguồn khác.

Bài báo cáo này là công trình nghiên cứu độc lập của chúng tôi chưa từng được công bố ở bất kỳ nơi nào khác. Tôi cam đoan rằng đã tuân thủ đầy đủ các quy tắc và quy định của môn học bao gồm cả việc tham khảo và sử dụng công cụ nghiên cứu.

Tôi hy vọng rằng bài báo cáo này sẽ cung cấp một cái nhìn tổng quan rõ ràng và toàn diện về chủ đề “Tìm hiểu Selenium và MongoDB, thu thập dữ liệu sản phẩm nhà thuốc Pharmacity" và sẽ đóng góp một phần nhỏ vào lĩnh vực nghiên cứu này.

TP.HCM, Ngày.....tháng.....năm 2024

**Sinh viên**

Hà Thế Anh

Hoàng Quang Minh

Nguyễn Nhật Nam

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

*TP. Hồ Chí Minh, ngày tháng năm 2023*

**Giáo viên hướng dẫn**

**(**Ký tên,đóng dấu**)**

# MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc181215129)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN ii](#_Toc181215130)

[MỤC LỤC iii](#_Toc181215131)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT VÀ TỪ KHÓA vi](#_Toc181215132)

[DANH MỤC HÌNH VẼ vii](#_Toc181215133)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc181215134)

[1.1. Giới thiệu đề tài 1](#_Toc181215135)

[1.2. Nhiệm vụ của đồ án 1](#_Toc181215136)

[*1.2.1.* *Tính cấp thiết của đề tài* 1](#_Toc181215137)

[*2.2.1.* *Ý nghĩa khoa học và thực tiễn của đề tài* 2](#_Toc181215138)

[1.3. Mục tiêu 3](#_Toc181215139)

[*1.3.1.* *Mục tiêu tổng quan* 3](#_Toc181215140)

[*1.3.1.* *Mục tiêu cụ thể* 4](#_Toc181215141)

[1.4. Đối tượng và phạm vi 4](#_Toc181215142)

[*1.4.1.* *Đối tượng* 4](#_Toc181215143)

[*1.4.2.* *Phạm vi* 5](#_Toc181215144)

[1.5. Phương pháp nghiên cứu 5](#_Toc181215145)

[*1.5.1.* *Phương pháp nghiên cứu sơ bộ* 5](#_Toc181215146)

[*1.5.2.* *Phương pháp nghiên cứu tài liệu* 5](#_Toc181215147)

[*1.5.3.* *Phương pháp nghiên cứu thống kê* 6](#_Toc181215148)

[*1.5.4.* *Phương pháp thực nghiệm* 6](#_Toc181215149)

[*1.5.5.* *Phương pháp đánh giá* 6](#_Toc181215150)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7](#_Toc181215151)

[2.1. Selenium 7](#_Toc181215152)

[*2.1.1.* *Giới thiệu và trích xuất dữ liệu từ Selenium.* 7](#_Toc181215153)

[*2.1.2.* *Ưu điểm và nhược điểm* 7](#_Toc181215154)

[*2.1.3.* *Thành phần chính của Selenium* 9](#_Toc181215155)

[*2.1.4.* *Cách thức Selenium WebDriver hoạt động.Nguồn* [4] 11](#_Toc181215156)

[*2.1.5.* *Ứng dụng của Selenium* 12](#_Toc181215157)

[2.2. MongoDB 13](#_Toc181215158)

[*2.2.1.* *Tổng quan về MongoDB* 13](#_Toc181215159)

[*2.2.2.* *Các thành phần của MongoDB* 14](#_Toc181215160)

[*2.2.3.* *Các cấu trúc của mô hình dữ liệu* 15](#_Toc181215161)

[*2.2.4.* *Các mẫu thiết kế của mô hình dữ liệu* 17](#_Toc181215162)

[*2.2.5.* Các phương thức CRUD trong MongoDB 20](#_Toc181215163)

[*2.2.6.* *Các tính năng nổi bật của MongoDB* 22](#_Toc181215164)

[*2.2.7.* *So sánh MongoDB và các cơ sở dữ liệu khác* 23](#_Toc181215165)

[*2.2.8.* Công cụ hỗ trợ MongoDB 24](#_Toc181215166)

[*2.2.9.* *Ưu điểm và nhược điểm* 24](#_Toc181215167)

[*2.2.10.* *Hướng phát triển* 26](#_Toc181215168)

[CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM 27](#_Toc181215169)

[3.1. Mục tiêu thực nghiệm 27](#_Toc181215170)

[*3.1.1.* *Tìm hiểu và thành thạo sử dụng các công cụ mã nguồn mở* 27](#_Toc181215171)

[*3.1.2.* *Tự động hóa quá trình thu thập dữ liệu dược phẩm* 27](#_Toc181215172)

[*3.1.3.* *Lưu trữ và xử lý dữ liệu hiệu quả thông qua MongoDB* 27](#_Toc181215173)

[3.2. Quá trình thực nghiệm 28](#_Toc181215174)

[*3.2.1.* *Định nghĩa các hàm thu thập dữ liệu* 28](#_Toc181215175)

[*3.2.2.* *Thu thập dữ liệu* 33](#_Toc181215176)

[*3.2.3.* *Chuẩn hóa dữ liệu* 34](#_Toc181215177)

[3.3. Mô tả dữ liệu 34](#_Toc181215178)

[3.4. Kết quả thực nghiệm 35](#_Toc181215179)

[*3.4.1.* *Kết quả thu thập* 35](#_Toc181215180)

[*3.4.2.* *Phân tích dữ liệu* 36](#_Toc181215181)

[*3.4.3.* *Đánh giá ưu và nhược điểm của việc thu thập dữ liệu bằng công cụ mã nguồn mở Selenium WebDriver* 43](#_Toc181215182)

[3.5. Kết luận 43](#_Toc181215183)

[CHƯƠNG 4. KẾT LUẬN VÀ KIẾN NGHỊ 44](#_Toc181215184)

[4.1. Kết luận 44](#_Toc181215185)

[4.2. Kiến nghị 44](#_Toc181215186)

[*4.2.1.* *Tích hợp các công cụ hỗ trợ xử lý trang động* 45](#_Toc181215187)

[*4.2.2.* *Tối ưu hóa hiệu suất thu thập dữ liệu* 45](#_Toc181215188)

[*4.2.3.* *Phát triển mô hình phân tích xu hướng* 45](#_Toc181215189)

[*4.2.4.* *Sử dụng cơ sở dữ liệu linh hoạt* 45](#_Toc181215190)

[*4.2.5.* *Đào tạo nhân sự chuyên môn* 46](#_Toc181215191)

[TÀI LIỆU THAM KHẢO 47](#_Toc181215192)

[PHỤ LỤC 49](#_Toc181215193)

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

# DANH MỤC HÌNH VẼ

[Hình 2.1: Thành phần của Selenium 9](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215194)

[Hình 2.2: Cách thức hoạt động của Selenium 11](#_Toc181215195)

[Hình 2.3: Ví dụ về mở một trang web tự động 12](#_Toc181215196)

[Hình 2.4: Ví dụ về nhúng dữ liệu 16](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215197)

[Hình 2.5: Ví dụ về tham chiếu 17](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215198)

[Hình 2.6: Ví dụ quan hệ một - một 17](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215199)

[Hình 2.7: Ví dụ về quan hệ một - nhiều 18](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215200)

[Hình 2.8: Ví dụng quan hệ một – nhiều sử dụng tham số 19](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215201)

[Hình 2.9: Ví dụ hệ thuống danh mục sách được mô tả theo cấu trúc cây 19](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215202)

[Hình 2.10: Thêm dữ liệu 20](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215203)

[Hình 2.11: Chèn nhiều tài liệu Documents vào một Collection 20](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215204)

[Hình 2.12: Truy xuất tất cả Documents trong Collection “students” 21](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215205)

[Hình 2.13: So sánh MongoDB và My SQL 23](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215206)

[Hình 2.14: So sánh với Apache Cassandra 24](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215207)

[Hình 3.1: Hàm load\_all\_products 28](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215208)

[Hình 3.2: Hàm get\_product\_links 29](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215209)

[Hình 3.3: Lấy mã sản phẩm 29](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215210)

[Hình 3.4: Lấy tên sản phẩm 30](#_Toc181215211)

[Hình 3.5: Lấy hình ảnh 30](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215212)

[Hình 3.6: Lấy thương hiệu 30](#_Toc181215213)

[Hình 3.7: Lấy giá bán 30](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215214)

[Hình 3.8: Lấy lượt yêu thích 31](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215215)

[Hình 3.9: Lấy số lượng bán 31](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215216)

[Hình 3.10: Lấy loại sản phẩm 31](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215217)

[Hình 3.11: Lấy quy cách 31](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215218)

[Hình 3.12: Lấy nơi sản xuất 31](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215219)

[Hình 3.13: Lấy hoạt tính 32](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215220)

[Hình 3.14: Lấy chỉ định 32](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215221)

[Hình 3.15: Tạo từ điển lưu dữ liệu thu thập được sau đó lưu vào MongoDB 32](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215222)

[Hình 3.16: Kết nối tới MongoDB, khởi tạo web driver, mở trang web thu thập dữ liệu 33](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215223)

[Hình 3.17: Thu thập dữ liệu 33](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215224)

[Hình 3.18: Chuẩn hóa dữ liệu 34](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215225)

[Hình 3.19: Hiện tất cả các sản phẩm và số lượng sản phẩm trong collection 'products' 36](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215226)

[Hình 3.20: Tìm sản phẩm không kê đơn có giá cao nhất 37](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215227)

[Hình 3.21: Tìm sản phẩm không kê đơn có giá thấp nhất 37](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215228)

[Hình 3.22: Lấy sản phẩm có thành phần hoạt tính chứa "Levocetirizin" 37](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215229)

[Hình 3.23: Đếm số sản phẩm có nguồn gốc từ "Việt Nam" 38](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215230)

[Hình 3.24: Đếm số sản phẩm không có nguồn gốc từ "Việt Nam" 38](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215231)

[Hình 3.25: Tìm sản phẩm có giá bán hơn 100k 38](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215232)

[Hình 3.26: Tìm sản phẩm có số lượng bán hơn 5000 39](file:///C:\Users\NAM\OneDrive\Documents\Doan2\repo_nhoms4\report\demobaocao.docx#_Toc181215233)

[Hình 3.27: Lấy thông tin chi tiết sản phẩm và với thông tin bán hàng 39](#_Toc181215234)

[Hình 3.28: Tìm sản phẩm có số lượt thích cao nhất và trả về tên, giá bán, like số lượng bán ra sản phẩm 41](#_Toc181215235)

[Hình 3.29: Tính tổng số lượng sản phẩm bán được từ collection 'sales' 41](#_Toc181215236)

[Hình 3.30: Tính tổng số tiền bán thuốc không kê đơn 42](#_Toc181215237)

# TỔNG QUAN

## Giới thiệu đề tài

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc thu thập và quản lý dữ liệu sản phẩm một cách hiệu quả là yếu tố then chốt đối với các doanh nghiệp, đặc biệt là trong lĩnh vực dược phẩm. Để tối ưu hóa được quy trình này, công cụ tự động như Selenium được sử dụng với công việc thu thập dữ liệu từ các website một cách tự động, trong khi MongoDB đóng vai trò là một kho lưu trữ và quản lý được khối lượng thông tin sản phẩm một cách linh hoạt. Áp dụng hai công cụ này lại với nhau giúp các doanh nghiệp và trong đó có nhà thuốc Pharmacity có thể dễ dàng theo dõi và quản lý sản phẩm, nhanh chóng cập nhật được thông tin, từ đó nâng cao hiệu quả kinh doanh và cải thiện khả năng cạnh tranh.

## Nhiệm vụ của đồ án

Nhiệm vụ của đề tài “Thu thập và phân tích dữ liệu dược phẩm từ website Pharmacity.vn sử dụng công cụ mã nguồn mở Selenium WebDriver và MongoDB” là áp dụng được Selenium để tự động quá trình thu thập dữ liệu sản phẩm từ website của nhà thuốc Pharmacity và lưu trữ, quản lý bằng MongoDB. Thông qua quá trình thu thập và quản lý dữ liệu này, doanh nghiệp sẽ có cái nhìn sâu sắc hơn về cái sản phẩm từ đó tối ưu hóa chiến lược bán hàng, nâng cao chất lượng dịch vụ và tăng khả năng cạnh tranh trên thị trường.

### *Tính cấp thiết của đề tài*

Trong thời đại số hóa, việc thu thập và quản lý dữ liệu hiệu quả là yếu tố then chốt để doanh nghiệp tồn tại và phát triển. Nhà thuốc Pharmacity là một chuỗi cửa hàng bán thuốc lẻ lớn đang phải đối mặt với sự cạnh tranh khốc liệt từ nhiều đối thủ. Để đáp ứng được nhu cầu của thị trường và duy trì cạnh tranh thì doanh nghiệp cần phải ứng dụng được công nghệ tự động hóa quy thu thập và quản lý dữ liệu sản phẩm.

Dự án “Thu thập và phân tích dữ liệu dược phẩm từ website Pharmacity.vn sử dụng công cụ mã nguồn mở Selenium WebDriver và MongoDB” được triển khai nhằm giải quyết được bài toán thu thập và quản lý dữ liệu sản phẩm. Cụ thể việc thu thập và quản lý dữ liệu mang lại cho nhà thuốc Pharmacity những lợi ích sau:

* Tối ưu hóa được chiến lược kinh doanh: Việc thu thập dữ liệu sẽ giúp doanh nghiệp hiểu rõ hơn về thị trường, từ đó đưa ra được chiến lược marketing và bán hàng chính xác tối ưu hóa hiệu quả kinh doanh.
* Cải thiện dịch vụ và trải nghiệm của người dùng: Với dữ liệu được cập nhật tự động hóa liên tục, doanh nghiệp có thể nhanh chóng bắt kịp xu hướng và đáp ứng nhu cầu của khách hàng, nâng cao sự trải nghiệm dịch vụ của người dùng.
* Tối ưu hóa vận hành: Dữ liệu chính xác về sản phẩm giúp doanh nghiệp quản lý kho hàng hiệu quả hơn, giảm lãng phí thời gian và tối ưu hóa quá trình phân phối.
* Đưa ra quyết định dựa trên dữ liệu: Doanh nghiệp có thể dựa trên dữ liệu chính xác để đưa ra quyết định thay đổi chiến lược một cách đúng đắn.
* Nâng cao khả năng cạnh tranh: Ứng dụng công nghệ vào thu thập dữ liệu, doanh nghiệp nhanh chóng thích ứng được xự thay đổi của thị trường và thay đổi chiến lược cho phù hợp, tăng cường vị thế cạnh tranh.

Với những lợi ích trên dự án không chỉ giúp doanh nghiệp nâng cao hiệu quả mà còn tăng cường khả năng và phát triển bền vững trong lĩnh vực dược phẩm.

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

**Ý nghĩa khoa học:** Dự án của nhóm chúng tôi sẽ đóng góp vào lĩnh vực tự động hóa thu thập dữ liệu bằng cách sử dụng công cụ Selenium và MongoDB. Việc áp dụng hai công cụ này giúp đơn giản hóa quá trình thu thập dữ liệu từ website, đặc biệt là dữ liệu về sản phẩm từ hệ thống nhà thuốc. Dự án không chỉ mở rộng kiến thức về cách tự động hóa trong thu thập dữ liệu mà còn giúp tích hợp những phương pháp xử lý dữ liệu cơ bản để khai thác thông tin hữu ích từ các nguồn dữ liệu trực tuyến. Điều này giúp các nhà phân tích và chuyên gia những công cụ hiện đại hơn trong công việc xử lý dữ liệu, tiềm năng ứng dụng được trong các dự án lớn.

**Ý nghĩa thực tiễn:** Dự án này mang lại nhiều lợi ích cho nhà thuốc Pharmacity trong việc tự động hóa thu thập và quản lý dữ liệu sản phẩm. Nhờ đó, dữ liệu sản phẩm được cập nhật liên tục giúp doanh nghiệp hiểu rõ hơn về xu hướng tiêu thụ trên thị trường, đẩy nhanh được chiến lược marketing và cải thiện dịch vụ đối với khách hàng. Việc áp dụng công cụ này giúp doanh nghiệp giảm bớt được thời gian và nguồn nhân lực, nắm bắt nhanh chóng trước những thay đổi của thị trường. Điều này đặc biệt quan trọng khi doanh nghiệp cần phải duy trì được lợi thế cạnh tranh trong môi trường phát triển hiện nay. Phương pháp thu thập và quản lý dữ liệu từ dự án có thể được mở rộng ra nhiều ngành hàng khác . Giúp nâng cao hiệu quả quản lý sản phẩm trong kinh doanh.

Ngoài ra, kết quả của dự án mang lại giúp doanh nghiệp có thể đưa ra những chiến lược dựa trên dữ liệu một cách chính xác thay vì dựa trên cảm tính. Điều này góp phần hỗ trở các nhà quản lý trong việc phát triển bền vững trong hoạt động kinh doanh. Nhờ sự kết hợp của lý thuyết và tự động hóa vào trong ứng dụng thực tiễn, dự án không chỉ mang lại giá trị khoa học mà còn đóng góp quan trọng vào việc nâng cao hiệu và phát triển bền vững của doanh nghiệp.

## Mục tiêu

### *Mục tiêu tổng quan*

Dự án này nhằm phát triển phương pháp tự động thu thập dữ liệu và quản lý dữ liệu sản phẩm nhà thuốc Pharmacity bằng công cụ Selenium để thu thập tự động dữ liệu, lưu trữ và quản lý dữ bằng MongoDB. Mục tiêu giúp doanh nghiệp nắm bắt nhanh chóng và chính xác thông tin sản phẩm, từ đó đưa ra được chiến lược kinh doanh hiệu quả, cải thiện hoạt động và tăng cường khả năng cạnh tranh trên thị trường.

### *Mục tiêu cụ thể*

Trong dự án này, nhóm tôi sẽ sử dụng các dữ liệu dược phẩm đã thu thập được từ website trực tuyến của nhà thuốc Pharmacity. Bao gồm thông tin sản phẩm, giá cả và đánh giá của khách hàng. Quá trình thực hiện bao gồm: xác định dữ liệu cần được thu thập và thu thập bằng Selenium, lưu trữ dữ liệu trong MongoDB, áp dụng phương pháp phân tích dữ liệu và đánh giá hiệu quả của hệ thống quản lý dữ liệu.

Cụ thể hơn, nhóm tôi sẽ triển khai công cụ Selenium để thu thập dữ liệu tự động và dùng MongoDB để quản lý và lưu trữ dữ liệu đã thu thập được. Sau đó, xử lý dữ liệu và phân tích dữ liệu để hỗ trợ doanh nghiệp đưa ra được các chiến lược tối ưu hóa vận hành, nâng cao chất lượng dịch vụ. Kết quả mang lại mong đợi là thu thập và quản lý dữ liệu một cách hiệu quả, giúp doanh nghiệp dễ dàng theo dõi và quản lý thông tin sản phẩm từ đó cải thiện hiệu suất kinh doanh và tăng cường khả năng cạnh tranh.

## Đối tượng và phạm vi

### *Đối tượng*

Đối tượng nghiên cứu của dự án là các sản phẩm đang được bán trên hệ thống trực tuyến của nhà thuốc Pharmacity. Dữ liệu sẽ bao gồm thông tin về sản phẩm, giá cả đánh giá của khách hàng. Mục tiêu của dự án là thu thập và phân tích dữ liệu để hiểu rõ hơn về những yếu tố ảnh hưởng đến hoạt động kinh doanh của Pharmacity, từ đó để xuất các giải pháp giúp cải thiện hiệu suất và tối ưu hóa quá trình vận hành.

### *Phạm vi*

Phạm vi phân tích tập trung vào công việc thu thập dữ liệu sản phẩm từ website của nhà thuốc Pharmacity thông qua công cụ Selenium và lưu trữ, quản lý bằng MongoDB. Dự án tập trung vào phân tích đánh giá thông tin sản phẩm, phản hồi và xu hướng tiêu thụ của khách hàng, tối ưu hóa chiến lược bán hàng và cải thiện trải nghiệm khách hàng.

## Phương pháp nghiên cứu

### *Phương pháp nghiên cứu sơ bộ*

Trước khi tiến hành thu thập dữ liệu, chúng tôi sẽ thực hiện nghiên cứu sơ bộ để hiểu rõ hơn về ngành dược phẩm trực tuyến và các yếu tố liên quan. Nghiên cứu sơ bộ sẽ bao gồm việc tìm hiểu về hoạt động kinh doanh trực tuyến của nhà thuốc Pharmacity, các yếu tố ảnh hưởng đến doanh thu và đánh giá sản phẩm của khách hàng, cũng như các phương pháp thu thập dữ diện tự động và phân tích được một cách hiệu quả. Thông qua quá trình này, chúng tôi sẽ xác định những vấn đề cần được giải quyết và lựa chọn các phương pháp nghiên cứu phù hợp cho dự án.

### *Phương pháp nghiên cứu tài liệu*

Chúng tôi sẽ tiến hành nghiên cứu tài liệu để thu thập thông tin về các công cụ và phương pháp tự động hóa thu thập dữ liệu và quản lý dữ liệu bằng công cụ Selenium và MongoDB. Qua việc đánh giá các nghiên cứu trước đây và các công trình khóa học liên quan, chúng tôi xác định kĩ thuật phù hợp nhất để triển khai việc thu thập dữ liệu dược phẩm từ website của Pharmacity và quản lý chúng một cách hiệu quả. Từ đó, những công cụ này sẽ xây dựng được hệ thống tự động hóa thu thập dữ liệu và phân tích dữ liệu cho dự án.

### *Phương pháp nghiên cứu thống kê*

Trong quá trình phân tích dữ liệu thu thập từ website của Pharmacity, chúng tôi sẽ sử dụng các phương pháp thống kê để mô tả và phân tích các yếu tố liên quan đến sản phẩm. Nhóm tôi sẽ đánh giá số lượng khách mua hàng của từng sản phẩm và đánh giá của khách hàng. Từ đó, chúng tôi sẽ đánh giá được các yếu tố ảnh hưởng đến hoạt động kinh doanh của Pharmacity đồng thời đề xuất các giải pháp để nâng cao năng suất kinh doanh.

### *Phương pháp thực nghiệm*

Nhóm sẽ thực hiện quá trình thực nghiệm dựa trên dữ liệu dược phẩm được thu thập từ website Pharmacity. Bap gồm các bước xử lý dữ liệu sản phẩm, áp dụng các phương phân tích để đánh giá xu hướng tiêu thụ của khách hàng, cũng như đánh giá những chiến lược kinh doanh đã đề ra. Thông qua thực nghiệm này, nhóm sẽ đưa ra được tính khả thi và hiệu quả của phương pháp nghiên cứu, nhằm tối ưu hóa quy trình quản lý sản phẩm, nâng cao hiệu suất cạnh tranh của Pharmacity trên thị trường trực tuyến.

### *Phương pháp đánh giá*

Sau khi thu thập và phân tích dữ liệu, nhóm sẽ tiến hành đánh giá và đo lường hiệu quả của quá trình phân tích mang lại. Bước này sẽ bao gồm so sánh về thông tin của sản phẩm, giá cả, phản hồi của khách hàng và xu hướng tiêu thụ để đánh giá hiệu suất của kinh doanh. Nhóm đối chiếu kết quả này với thực tế hoạt động của Pharmacity để đưa ra giải pháp cụ thể giúp tối ưu hóa quá trình kinh doanh, nâng cao hiệu quả quản lý sản phẩm và cải thiện tổng thể hoạt động của doanh nghiệp.

# CƠ SỞ LÝ THUYẾT

## Selenium

### *Giới thiệu và trích xuất dữ liệu từ Selenium.*

Selenium là một công cụ mã nguồn mở tự động hóa trình duyệt web, cho phép người dùng điều khiển các trình duyệt một cách tự động. Được phát triển bởi Jason Huggins vào năm 2004 và cho đến hiện nay trở thành một công cụ phổ biến nhất dùng để kiểm thử, tự động hóa và tương tác với trang web. Selenium được thực hiện thông qua các mã lệnh của người dùng, cho phép tự động các thao tác như nhấp chuột, nhập dữ liệu, cuộn trang và các hành động khác do người thực hiện các mã lệnh đưa ra. [1]

### *Ưu điểm và nhược điểm*

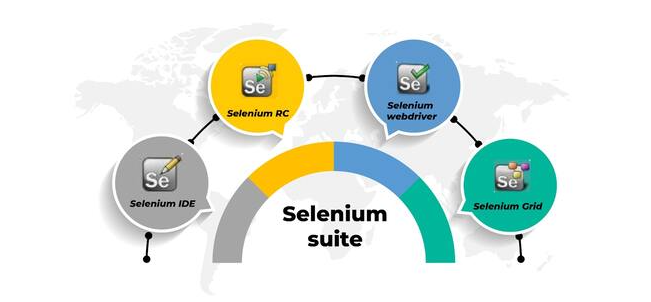
Ưu điểm:

* Hỗ trợ nhiều ngôn ngữ lập trình: Selenium đa dạng ngôn ngữ lập trình giúp cho người dùng dễ dàng viết các câu lệnh theo ngôn ngữ đã quen thuộc. Những ngôn ngữ được hỗ trợ như Python, Java, C# và JavaScript.
* Hỗ trợ nhiều hệ điều hành và trình duyệt: Selenium có thể tương tác được với các trình duyệt khác nhau như Google Chrome, Firefox, Safari và hoạt động trên các hệ điều hành Windows, macOS, Linux.
* Hỗ trợ thao tác cho người dùng: Công cụ mã nguồn mở này giúp cho người dùng tự động hóa được các thao tác như nhấp chuột, nhập dữ liệu, cuộn trang, đăng nhập vào một website và có thể điều hướng qua nhiều trang khác nhau. Điều này giúp cho phép thu thập dữ liệu một cách tự động mà các công cụ khách không thể thực hiện được.
* Tốc độ thực thi nhanh: Selenium WebDriver tận dụng khá tốt khả năng tự động hóa của mình trên các trình duyệt web được hỗ trợ. Mỗi trình duyệt khác nhau sẽ có một công cụ hỗ trợ trình duyệt khác nhau như ChromeDriver của Chrome hay Firefox của GeckoDriver. Do đó, tốc độ thực thi của Selenium WebDriver sẽ nhanh hơn nhiều với các công cụ khác.

Nhược điểm:

* Tiêu tốn nhiều tài nguyên: Do Selenium chạy và điều khiển trình duyệt, nó yêu cầu cần nhiều tài nguyên hệ thống hơn như CPU và RAM. Điều này gây ra tình trạng chậm hoặc treo máy khi khởi chạy đồng thời quá nhiều tác vụ làm giảm hiệu quả khi thu thập dữ liệu có quy mô lớn.
* Khó khăn trong việc xử lý nguồn dữ liệu lớn: Khi lựa chọn việc thu thập dữ liệu với số lượng lớn với nhiều trang web khác nhau thì Selenium không phải là lựa chọn hàng đầu do hiệu suất mang lại chậm và giới hạn về khả năng xử lý. Để xử lý nguồn dữ liệu lớn hơn người dùng thường sử dụng các công cụ chuyên dụng hơn cho web scraping như Scrapy hoặc BeautifulSoup hai công cụ này sẽ xử lý hiệu quả hơn trong tình huống này.
* Đòi hỏi kiến thực lập trình cao: Để sử dụng hiệu quả người dùng cần phải có kiến thức lập trình và hiểu các phần tử trên trang web. Điều này là rào cản đối với những người mới hoặc chưa có kiến thức về lập trình cấu trúc web.

### *Thành phần chính của Selenium*

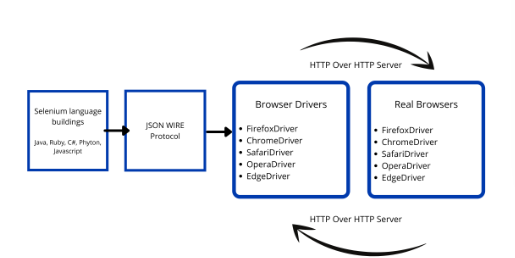


Hình 2.1: Thành phần của Selenium

Selenium gồm có 4 phần chính:[2]

* Selenium Integrated Development Environment (Selenium IDE): Là một công cụ dựa trên đồ họa GUI cho phép người dùng có thể ghi lại, chỉnh sửa và phát lại các kịch bản. Đây là một tiện ích mở rộng của trình duyệt, lần đầu được phát triển bởi phiên bản của hơn của Firefox. Người dùng có thể tìm kiếm và cài đặt Selenium IDE qua trạng thái bổ sung của Firefox. Đặc điểm nổi bật của Selenium là sử dụng một ngôn ngữ kịch bản riêng gọi là Selenese. Selenese rất dễ sử dụng không đòi hỏi về kiến thức lập trình chuyên sâu. Mặc dù dễ sử dụng nhưng nó cũng có một số hạn chế:
* Thực hiện các trường hợp thử nghiệm rất chậm và báo cáo không tốt khi so sánh với các thành phần khác.
* Không thể hỗ trợ thử nghiệm trên thiết bị di động.
* Không hỗ trợ thực hiện các trường hợp thử nghiệm song song hoặc từ xa.
* Selenium Webdriver: Đây là phiên bản cải tiến của Selenium RC, giúp khắc phục được những hạn chế của Selenium RC. Không giống như Selenium RC, WebDriver không yêu cầu khởi động máy chủ trung gian để thực hiện các bài kiểm thử. Thay vào đó thì Selenium WebDriver thao tác trực tiếp với trình duyệt giúp quá trình diễn ra nhanh hơn và hiệu quả hơn. WebDriver hỗ trợ nhiều ngôn ngữ lập trình như Java, C#, Python, PHP và Perl được hỗ trợ hoạt động trên nhiều hệ điều hành khác nhau như Windows, macOS, Linux. Nó cũng hỗ trợ quá trình kiểm thử song song và từ xa thông qua Selenium Grid. Mỗi trình duyệt để có trình điều khiển để thực hiện:
* ChromeDriver cho Chrome.
* GeckoDriver cho Firefox.
* SafariDriver cho Safari.
* IE Driver cho Internet Explorer.
* OperaDriver cho Opera.
* Selenium Grid: Công cụ có nhiệm vụ phục vụ cho việc thực thi đồng thời các kịch bản thử nghiệm trên nhiều trình duyệt và hệ thống khác nhau. Selenium Grid cho phép phân phối các kịch bản thử nghiệm trên các thiết bị và môi trường riêng biệt. Từ đó tăng hiệu suất và giảm thời gian kiểm thử.[3]
* Selenium RC (Remote Control): Đây là một trong những phiên bản đầu tiên của Selenium, được dùng để tự động hóa kiểm thử trình duyệt. Không giống như những công cụ Selenium khác mà Selenium RC không có giao diện đồ họa GUI thay vào đó nó chứa các thư viện và API để lập trình viên viết mã thử bằng nhiều ngôn ngữ khác nhau như C#, Java, PHP, Perl. Selenium RC hỗ trợ đa nền tảng và đa trình duyệt (Chrome, Firefox, Safari, Opera). Đồng thời hỗ trợ thực thi các kịch bản kiểm thử song song và từ xa thông qua Selenium Grid. Tuy nhiên, có những nhược điểm sau:
* Yêu cầu phải khởi động máy chủ thủ công: Trước khi chạy kiểm thử, người dùng phải khởi động Selenium Server để đóng vai trò trung gian giữa kiểm thử và trình duyệt.
* Không tương tác trực tiếp qua trình duyệt: Mã kiểm thử không giao tiếp trực tiếp được qua trình duyệt mà phải thông qua máy chủ Selenium, làm cho quá trình kiểm thử trở nên phức tạp và chậm hơn.

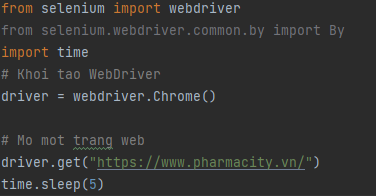
### *Cách thức Selenium WebDriver hoạt động.Nguồn* [4]



Hình 2.2: Cách thức hoạt động của Selenium

* Bước 1: Gửi câu lệnh từ mã đến WebDriver:
* Khi chạy các câu lệnh kiểm thử, Selenium WebDriver nhận chúng và sử dụng JSON Write Protocol để chuyển đổi các lệnh này thành yêu cầu HTTP.
* Bước 2: Quá trình điều khiển của trình duyệt và xử lý yêu cầu:
* Mỗi trình duyệt đều có trình điều khiển riêng (ChromeDriver của Chrome, GeckoDriver của Firefox,..) nhận các yêu cầu của WebDriver. Trình điều khiển dịch các câu lệnh này và thực hiện các thao tác trình duyệt có thể hiểu và thực hiện.
* Bước 3: Trình duyệt thực hiện và trả về:
* Trình điều khiển gửi các lệnh đến trình duyệt và thực hiện các thao tác như (mở trang, nhấp chuột, nhập dữ liệu, cuộn trang). Sau khi hoàn thành các thao tác trình điều khiển gửi đến, kết quả sẽ được trả về WebDriver và hiển thị chương trình kiểm thử.

Ví dụ: Selenium có thể truy cập vào trang web bán thuốc trực tuyến của nhà thuốc Pharmacity. Từ đó có thể phát triển thêm việc thu thập dữ liệu của website Pharmacity để lấy tên dược phẩm, giá bán, số lượng mua và đánh giá của khách hàng.



Hình 2.3: Ví dụ về mở một trang web tự động

### *Ứng dụng của Selenium*

* Kiểm thử tự động: Selenium là công cụ kiểm thử tự động cho các ứng dụng web. Nó giúp phát hiện ra lỗi và đảm bảo hoạt động ổn định trên nhiều trình duyệt và hệ điều hành. Đây là phần quan trọng trong phát triển phần mềm và đặc biệt với các dự án.[5]
* Tự động hóa các tác vụ hàng ngày: Selenium có thể tự động hóa các tác vụ như đăng nhập, điền form và tương tác với các trang web giúp tiết kiệm thời gian trong các quy hàng ngày.[6]
* Thu thập dữ liệu web: Dù đây không phải là công cụ chuyên dụng để thu thập dữ liệu nhưng Selenium vẫn rất hiệu quả trong quá trình thu thập dữ liệu từ các trang web động những nơi có chứa nội dung sử dụng JavaScript.
* Tích hợp quy trình CI/CD: Selenium hỗ trợ kiểm thử tự động trong các quá trình CI/CD, giúp kiểm tra liên tục trong quá trình phát triển phần mềm.[7]

## MongoDB

### *Tổng quan về MongoDB*

MongoDB là một trong những hệ quản trị cơ sở dữ liệu NoSQL nổi bật, được thiết kế dưới dạng mã nguồn mở và hướng tài liệu. Khác với các cơ sở dữ liệu quan hệ truyền thống dùng bảng để lưu trữ, MongoDB sử dụng Document để lưu trữ và truy xuất dữ liệu. Dữ liệu trong MongoDB được định dạng bằng BSON, tương tự như JSON.

Được ra mắt vào năm 2009, MongoDB được lập trình bằng ngôn ngữ C++, giúp nó vận hành với tốc độ nhanh hơn so với nhiều hệ quản trị cơ sở dữ liệu khác. Chính điều này đã nâng cao giá trị của MongoDB trong mắt các nhà phát triển.

Hệ thống này hỗ trợ đa nền tảng và cho phép lưu trữ các dữ liệu có cấu trúc phức tạp. Thay vì sử dụng bảng truyền thống, MongoDB lưu trữ dữ liệu theo cách Collections dựa trên tài liệu JSON. Đây là mô hình dữ liệu Key-Value, mang lại khả năng truy xuất nhanh chóng và khả năng mở rộng linh hoạt mà không cần các ràng buộc khóa ngoại hay khóa chính.

* NoSQL (Not Only SQL): MongoDB không sử dụng cấu trúc bảng truyền thống như các cơ sở dữ liệu quan hệ (SQL). Điều này cho phép MongoDB linh hoạt hơn trong việc xử lý dữ liệu phi cấu trúc hoặc bán cấu trúc.
* Tài liệu JSON: Dữ liệu trong MongoDB được lưu dưới dạng tài liệu JSON, giúp ta dễ dàng làm việc với dữ liệu có cấu trúc thay đổi theo thời gian.
* Schema-less: MongoDB là một cơ sở dữ liệu dựa trên Document, các tài liệu trong một Collection có thể có cấu trúc khác nhau.[8]

### *Các thành phần của MongoDB*

MongoDB gồm có 6 phần chính:

* Mongod:Là tiến trình lõi của MongoDB, đảm nhận nhiệm vụ lưu trữ và quản lý dữ liệu.
* Mongosh: Với giao diện Shell giúp người dùng có thể tương tác thực hiện lệnh với cơ sở dữ liệu.
* Collection: Một tập hợp các Document MongoDB. Tương tự như bảng trong cơ sở dữ liệu quan hệ, Collection có tính linh hoạt cao, không ràng buộc bởi cấu trúc cố định, cho phép chứa các tài liệu với những cấu trúc khác nhau.
* Document:
  + - * Đơn vị lưu trữ dữ liệu cơ bản trong cơ sở dữ liệu MongoDB.
      * Document trong MongoDB không cần phải có cùng trường hoặc cấu trúc với các Document khác trong cùng một Collection.
      * Đồng thời, các trường chung trong Document của một Collection có thể chứa các loại dữ liệu khác nhau.
* Database (Cơ sở dữ liệu):
  + - * Database là một thùng chứa vật lý chứa tập hợp các Collection. Một Database có thể chứa không Collection hoặc nhiều Collection.
      * Một phiên bản máy chủ MongoDB có thể lưu trữ nhiều Database và không có giới hạn về số lượng Database có thể được lưu trữ trên một phiên bản.
      * MongoDB sẽ tự động tạo cơ sở dữ liệu mới khi lưu trữ Document lần đầu tiên nếu như cơ sở dữ liệu không tồn tại.
* MongoDB Compass: Là công cụ giao diện đồ họa (GUI) cho phép người dùng tương tác trực quan với cơ sở dữ liệu MongoDB thay vì sử dụng giao diện dòng truyền thống. Với khả năng hiển thị trực quan các cơ sở dữ liệu, Collection, Document, MongoDB Compass hỗ trợ người dùng trong việc truy vấn, thực hiện các hàm tổng hợp và phân tích dữ liệu hiệu quả.[9]

### *Các cấu trúc của mô hình dữ liệu*

Một điểm mấu chốt trong việc tạo lập mô hình dữ liệu cho những ứng dụng sử dụng MongoDB liên quan đến cách tổ chức các Collection và cách thức mà ứng dụng thể hiện mối liên kết giữa các dữ liệu.

* Nhúng dữ liệu: MongoDB cho phép tích hợp mọi dữ liệu liên quan vào một tài liệu duy nhất. Dữ liệu này phản ánh mối quan hệ giữa các thành phần thông qua việc tổ chức chúng trong một cấu trúc tài liệu duy nhất. Nền tảng cũng hỗ trợ việc tích hợp các cấu trúc dữ liệu vào một trường hoặc mảng trong tài liệu. Các mô hình dữ liệu không theo tiêu chuẩn này tạo điều kiện cho ứng dụng thực hiện truy xuất và xử lý dữ liệu liên quan trong một thao tác cơ sở dữ liệu duy nhất.



Hình 2.4: Ví dụ về nhúng dữ liệu

* Tham chiếu: Được gọi là kiến trúc chuẩn hóa của mô hình dữ liệu, các tham chiếu lưu trữ các mối quan hệ giữa dữ liệu thông qua việc thêm liên kết hoặc tham chiếu giữa các tài liệu. Các ứng dụng có thể sử dụng những tham chiếu này để truy cập các dữ liệu liên quan. Đây là một dạng mô hình dữ liệu đã được chuẩn hóa. Trong ví dụ dưới đây, trường “user\_id” trong bộ dữ liệu “contact” và “access” được sử dụng để tham chiếu đến người dùng tương ứng.[10]

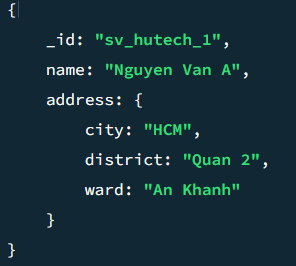
### *Các mẫu thiết kế của mô hình dữ liệu*

Hình 2.5: Ví dụ về tham chiếu

* Mô hình dữ liệu quan hệ giữa các Document:

1. Quan hệ một – một

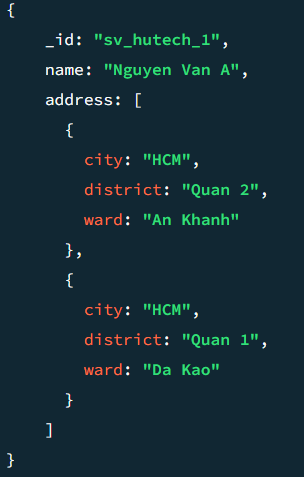
Ví dụ nếu xây dựng sơ đồ mối liên hệ một – một giữa sinh viên và địa chỉ của họ. Trong mô hình dữ liệu chuẩn hóa, tài liệu “DiaChi” sẽ bao gồm tham chiếu đến tài liệu “SinhVien”. Nếu ứng dụng thường xuyên cần truy cập dữ liệu địa chỉ song song với dữ liệu sinh viên, thì việc sử dụng tham chiếu sẽ yêu cầu thực hiện nhiều truy vấn hơn để xử lý các tham chiếu đó.[11]



Hình 2.6: Ví dụ quan hệ một - một

1. Quan hệ một – nhiều sử dụng nhúng dữ liệu

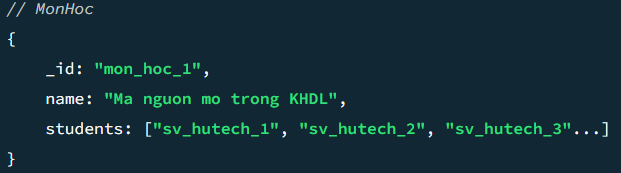
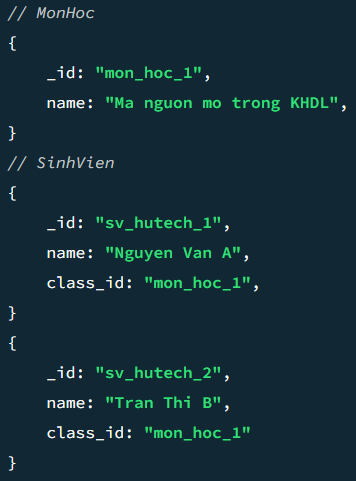
Mô hình dữ liệu quan hệ mang tính một – nhiều áp dụng phương pháp nhúng dữ liệu khi cần truy xuất nhiều thực thể liên quan đến một thực thể trong tập dữ liệu khác. Ví dụ như, một sinh viên có thể có nhiều địa chỉ liên lạc. Nếu việc truy cập thông tin địa chỉ cùng lúc với dữ liệu sinh viên diễn ra thường xuyên, thì phương án hiệu quả hơn là nhúng các thực thể địa chỉ vào trong dữ liệu sinh viên. [11]



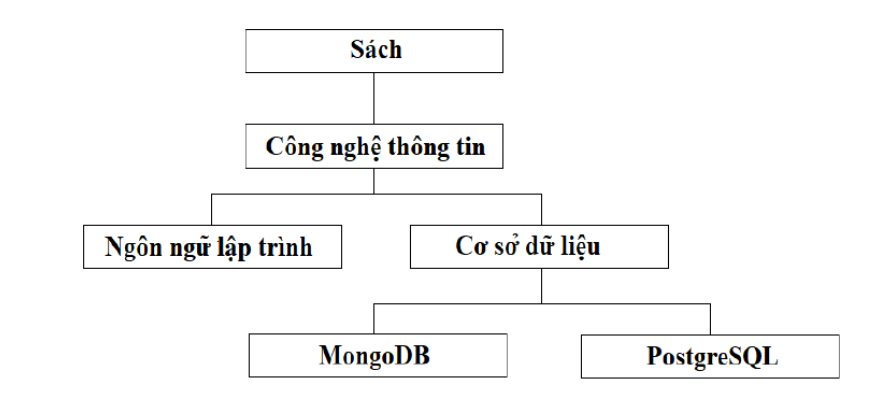
Hình 2.7: Ví dụ về quan hệ một - nhiều

1. Quan hệ một – nhiều sử dụng tham chiếu

Xem xét ví dụ về cấu trúc dữ liệu mô tả mối quan hệ giữa sinh viên và các khóa học đã đăng ký. Một môn học có thể thu hút nhiều sinh viên, vì vậy việc tích hợp thông tin môn học vào dữ liệu của từng sinh viên có thể dẫn đến tình trạng lặp lại thông tin môn học, gây lãng phí tài nguyên hệ thống. Để ngăn chặn việc lưu trữ trùng lặp thông tin môn học, có thể tạo ra một bộ sưu tập riêng biệt để lưu trữ dữ liệu về môn học. Cần ghi nhớ rằng trong trường hợp này, một môn học sẽ có rất nhiều sinh viên đăng ký, do đó, việc tham chiếu từ dữ liệu môn học sang dữ liệu sinh viên có thể khiến lượng thông tin cần lưu trữ trở nên khổng lồ.[11]

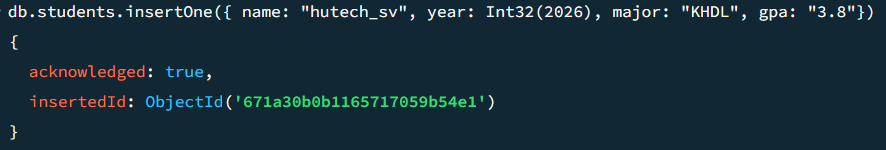


Hình 2.8: Ví dụng quan hệ một – nhiều sử dụng tham số

* Mô hình cây:[12]

Hình 2.9: Ví dụ hệ thuống danh mục sách được mô tả theo cấu trúc cây

### Các phương thức CRUD trong MongoDB

* Thêm dữ liệu: Sử dụng cú pháp *db.collection.insertOne()* để thêm một Document vào Collection.

Hình 2.10: Thêm dữ liệu

* Khi cần thêm nhiều Document cùng một lúc, sử dụng cú pháp ***db.collection.insertMany()*** và truyền vào một mảng gồm các Document.

Hình 2.11: Chèn nhiều tài liệu Documents vào một Collection

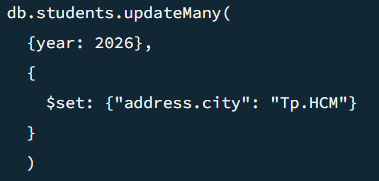
* Truy vấn dữ liệu: Để truy xuất tất cả tài liệu trong một Collection, bạn có thể áp dụng phương pháp ***db.collection.find()*** và cung cấp một đối tượng JSON rỗng {} làm tham số.



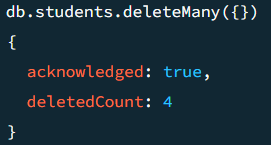
Hình 2.13: Truy xuất tất cả Documents trong Collection “students” với trường “year” bằng 2026

Hình 2.12: Truy xuất tất cả Documents trong Collection “students”

* Cập nhật dữ liệu: Cập nhật dữ liệu trên MongoDB được thực hiện thông qua các câu lệnh:
* db.collection.update()
* db.collection.updateMany()



Hình 2.14: Chèn thêm các Documents vào Collection “students”

* Xóa dữ liệu: Để xóa tất cả Document trong Collection ta sử dụng:
* db.collection.delete.Many() và truyền vào tham số là một đối tượng JSON rỗng.

Hình 2.15: Xóa tất cả Documents trong Collection

### *Các tính năng nổi bật của MongoDB*

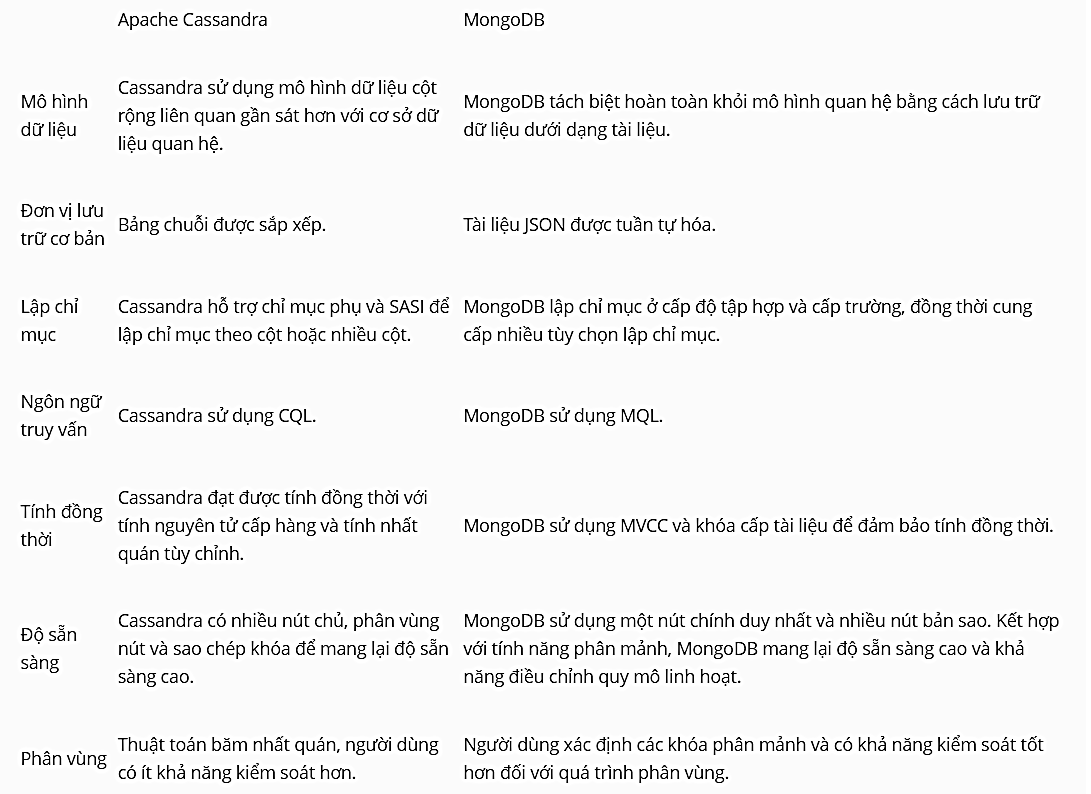
* Tìm kiếm và lập chỉ mục mạnh: MongoDB cung cấp khả năng tìm kiếm dữ liệu nhanh chóng nhờ hệ thống lập chỉ mục linh hoạt.
* Aggregation framework: Cho phép xử lý dữ liệu phức tạp và tạo các báo cáo chi tiết từ dữ liệu.
* Replication và Sharding**:** Đảm bảo tính sẵn sàng cao bằng cách tạo bản sao dữ liệu trên nhiều máy chủ. Ngoài ra, còn có khả năng tự động sao lưu dữ liệu, đảm bảo độ tin cậy và khả năng phục hồi của cơ sở dữ liệu.
* BSON: Là một định dạng dữ liệu nhị phân được sử dụng để lưu trữ tài liệu và thực hiện các cuộc gọi từ xa trong MongoDB. BSON bao gồm nhiều kiểu dữ liệu khác nhau, cho phép sử dụng số hoặc chuỗi định danh để xác định kiểu dữ liệu của một đối tượng.[13]

### *So sánh MongoDB và các cơ sở dữ liệu khác*

* *So với* SQ*L ( My*SQL*, Postgre*SQ*L)*: MongoDB có ưu điểm về tính linh hoạt và mở rộng, nhưng không mạnh trong việc xử lý các mối quan hệ phức tạp giữa các bảng.[14]

Hình 2.13: So sánh MongoDB và My SQL

* So với các cơ sở dữ liệu NoSQL khác (Cassandra, Couchbase): MongoDB nổi bật nhờ mô hình dữ liệu tài liệu dễ tiếp cận và cộng đồng hỗ trợ rộng lớn.[15]



Hình 2.14: *So* sánh với Apache Cassandra

### Công cụ hỗ trợ MongoDB

* MongoDB Compass: **:** Một công cụ GUI (giao diện đồ họa) cho phép người dùng trực hóa, quản lý và thao tác với dữ liệu MongoDB.
* NoSQL Manager: Công cụ xuất sắc này kết hợp một cách hài hòa giữa giao diện người dùng và Shell mang lại hiệu suất vượt trội cùng với nhiều tính năng hỗ trợ hoàn hảo cho MongoDB.[16]

### *Ưu điểm và nhược điểm*

Ưu điểm:

* + Không Schema: Giống như các cơ sở dữ liệu NoSQL khác, MongoDB không yêu cầu các Schema được xác định trước/
  + Lưu trữ bất kỳ loại dữ liệu nào: Điều này cho phép người dùng linh hoạt tạo số lượng trường trong Document theo nhu cầu, và giúp việc mở rộng cơ sở dữ liệu MongoDB trở nên dễ dàng hơn so với cơ sở dữ liệu quan hệ truyền thống.
  + Hướng Document: Một trong những ưu điểm của việc sử dụng Document là các đối tượng này ánh xạ tới các kiểu dữ liệu gốc trong một số ngôn ngữ lập trình. Việc có các Document được nhúng cũng làm giảm nhu cầu kết nối cơ sở dữ liệu, điều này có thể làm giảm chi phí.
  + Khả năng mở rộng: MongoDB có khả năng mở rộng dễ dàng bằng cách phân tán dữ liệu trên nhiều máy chủ qua việc phân chia dữ liệu (sharding). Ngoài ra, MongoDB cũng hỗ trợ tạo vùng dữ liệu dựa trên Shard Key.
  + Hiệu suất cao: MongoDB đặc biệt tốt trong việc xử lý lượng lớn dữ liệu phi cấu trúc hoặc bán cấu trúc.
  + Sử dụng các tính năng tìm hiểu được trên một ứng dụng trên môi trường localhost.

Nhược điểm:

* + Tốn tài nguyên: MongoDB sử dụng nhiều tài nguyên hơn so với cơ sở dữ liệu quan hệ vì cần lưu thêm metadata trong tài liệu.
  + Tính nhất quán của dữ liệu: MongoDB không cung cấp tính toàn vẹn tham chiếu đầy đủ thông qua việc sử dụng các ràng buộc khóa ngoại (foreign-key), điều này có thể ảnh hưởng đến tính nhất quán của dữ liệu.
  + Tính liên tục: Với chiến lược chuyển đổi dự phòng tự động, người dùng chỉ có thể thiết lập một node master trong cụm MongoDB. Nếu node master bị lỗi, một node khác sẽ tự động chuyển đổi thành master mới. Quá trình chuyển đổi này đảm bảo tính liên tục, nhưng không diễn ra tức thời mà có thể mất tới một phút.[11]

### *Hướng phát triển*

* Tối ưu hóa việc sử dụng tài nguyên bằng cách nghiên cứu và phát triển thêm các thuật toán nén hiệu quả hơn để tiết kiệm dung lượng lưu trữ và tài nguyên. Đồng thời, cải tiến cách lưu trữ metadata bằng cách giảm bớt thông tin thừa hoặc tìm kiếm định dạng nhẹ hơn mà vẫn giữ được tính linh hoạt của cơ sở dữ liệu.
* Tăng cường tính nhất quán của dữ liệu bằng việc tích hợp hỗ trợ khóa ngoại hoặc thiết lập cơ chế kiểm tra mối quan hệ giữa các collection để bảo vệ dữ liệu tốt hơn.
* Phát triển chiến lược mở rộng tự động tài nguyên theo thời gian thực khi cần thiết, chẳng hạn như tự động bổ sung node để cải thiện hiệu suất và giảm tải cho hệ thống khi nhu cầu gia tăng.

# KẾT QUẢ THỰC NGHIỆM

## Mục tiêu thực nghiệm

Mục tiêu của việc thu thập và phân tích dữ liệu dược phẩm trên website Pharmacity.vn bằng **Selenium WebDriver** và **MongoDB** có thể bao gồm các điểm như sau:

### *Tìm hiểu và thành thạo sử dụng các công cụ mã nguồn mở*

Thông qua quá trình thu thập và lưu trữ dữ liệu dược phẩm từ website Pharmacity, giúp chúng ta hiểu rõ hơn về các công cụ như **Selenium** **webdriver** để tự động hóa các thao tác thu thập dữ liệu trên trình duyệt, công cụ lưu trữ và phân tích, xử lý dữ liệu như **MongoDB**, **SQLite** nhằm phục vụ cho các mục tiêu, yêu cầu khác nhau cũng như thành thạo cách sử dụng chúng.

### *Tự động hóa quá trình thu thập dữ liệu dược phẩm*

Sử dụng công cụ mã nguồn mở **Selenium WebDriver** để tự động hóa quá trình duyệt website, thu thập các dữ liệu dược phẩm như mã, tên sản phẩm, giá bán, mô tả, thành phần, công dụng giúp giảm thiểu thời gian và công sức so với việc thu thập dữ liệu thủ công, cập nhật dữ liệu nhanh chóng khi có sự thay đổi về dữ liệu.

### *Lưu trữ và xử lý dữ liệu hiệu quả thông qua MongoDB*

Sau quá trình thu thập dữ liệu ta cần phải lưu trữ dữ liệu để xử lý, thông qua **MongoDB**, một cơ sở dữ liệu NoSQL. Giúp chúng ta lưu trữ và quản lý dữ liệu một cách linh hoạt, thuận tiện cho bước phân tích dữ liệu.

Dựa trên các dữ liệu thu thập được tiến hành phân tích dữ liệu chẳng hạn như đặc điểm của các loại dược phẩm bán chạy, những loại thuốc có nhu cầu sử dụng cao, xu hướng mua hàng nhằm nhận diện các yếu tố ảnh hưởng đến quyết định mua hàng, mục tiêu nhằm đề xuất các chính sách quảng cáo hiệu quả, cải thiện trải nghiệm người dùng, phát triển sản phẩm.

## Quá trình thực nghiệm

Dữ liệu dược phẩm thu thập từ website Pharmacity.vn sử dụng công cụ mã nguồn mở **Selenium WebDriver** và **Xpath** để thu tự động hóa quá trình duyệt web và thu thập dữ liệu, trong quá trình thu thập sử dụng thư viện **Re** trong **Python** để xử lý dữ liệu. Toàn bộ dữ liệu sau khi được xử lý sơ bộ sẽ được lưu trữ vào **MongoDB**. Quá trình thu thập dữ liệu bao gồm các bước sau:

### *Định nghĩa các hàm thu thập dữ liệu*

Để thuận tiện cho quá trình thu thập dữ liệu, đồng thời thu thập dữ liệu một cách chính xác, tránh khả năng trùng lặp dữ liệu, chúng tôi đã viết lên ba hàm trong python để phục vụ các mục tiêu trên các hàm lần lượt là:

#### Tải toàn bộ dược phẩm có trong trang

Hàm **load\_all\_products**: Định nghĩa hàm load\_all\_products sử dụng Selenium WebDriver, khởi tạo một vòng lặp vô hạn thực hiện thao tác cuộn xuống và tải thêm sản phẩm trên website Pharmacity.vn cho đến khi không còn sản phẩm để tải nữa.

A computer screen shot of text

Description automatically generatedMục tiêu: Lấy hết toàn bộ dữ liệu dược phẩm để thuận tiện cho việc lấy đường liên kết dẫn tới website của từng loại dược phẩm.

Hình 3.1: Hàm load\_all\_products

Hình 3.2.1: Hàm load\_all\_products

#### Lấy toàn bộ các đường liên kết dẫn đến website từng dược phẩm

Hàm **get\_product\_links**: Định nghĩa hàm **get\_product\_links** khởi tạo danh sách chứa liên kết các sản phẩm trên trang, sử dụng **Selenium**, **CSS Selector** để lấy các phần tử <a> có chứa tên sản phẩm, khởi tạo vòng lặp duyệt qua từng phần tử tìm được lấy thuộc tính “href” chính là đường liên kết dẫn đến trang web dược phẩm, lưu vào danh sách chứa liên kết.

A computer screen shot of a computer code

Description automatically generatedMục tiêu: Việc thu thập liên kết của từng sản phẩm giúp thu thập dữ liệu hiệu quả hơn, tránh việc trùng lặp dữ liệu, bỏ sót dữ liệu ảnh hưởng đến quá trình phân tích dữ liệu.

Hình 3.2.2: Hàm get\_product\_links

Hình 3.2: Hàm get\_product\_links

#### Lấy thông tin sản phẩm

Hàm **scrape\_product**: Định nghĩa hàm **scrape\_product** truy cập vào các liên kết đã thu thập trước đó, sử dụng **Selenium**, **CSS Selector**, **Xpath** tìm và lấy các phần tử chứa các thông tin sau:

* **Mã sản phẩm**

Hình 3.3: Lấy mã sản phẩm

Hình 3.2.3: Lấy mã sản phẩm

* **Tên sản phẩm**

A screenshot of a computer

Description automatically generated

Hình 3.4: Lấy tên sản phẩm

* **A computer screen with text

  Description automatically generatedLấy hình ảnh**

Hình 3.5: Lấy hình ảnh

Hình 3.2.5: Lấy hình ảnh

* **Lấy thương hiệu**

A computer screen with green and blue text

Description automatically generated

Hình 3.6: Lấy thương hiệu

* **A computer screen shot of text

  Description automatically generatedLấy giá bán lẻ**

Hình 3.7: Lấy giá bán

Hình 3.2.7: Lấy giá bán

* **Lấy lượt yêu thích**



Hình 3.2.8: Lấy lượt yêu thích

Hình 3.8: Lấy lượt yêu thích

* **A computer screen shot of a code

  Description automatically generatedLấy số lượng bán ra**

Hình 3.2.9: Lấy số lượng bán

Hình 3.9: Lấy số lượng bán

* **Lấy loại sản phẩm**

A screenshot of a computer

Description automatically generated

Hình 3.2.10: Lấy loại sản phẩm

Hình 3.10: Lấy loại sản phẩm

* **Lấy quy cách**

A computer screen shot of a black background

Description automatically generated

Hình 3.2.11: Lấy quy cách

Hình 3.11: Lấy quy cách

* **Lấy nơi sản xuất**

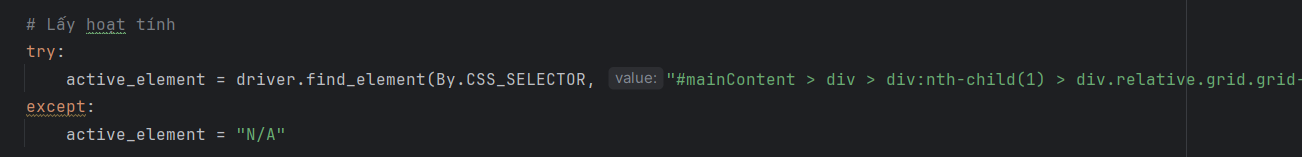
A computer screen shot of a program

Description automatically generated

Hình 3.12: Lấy nơi sản xuất

Hình 3.2.12: Lấy nơi sản xuất

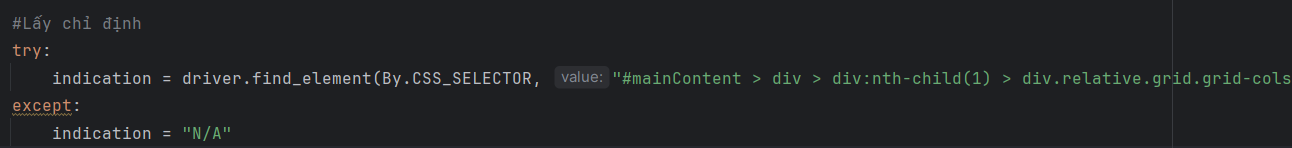
* **Lấy hoạt tính**



Hình 3.2.13: Lấy hoạt tính

Hình 3.13: Lấy hoạt tính

* **Lấy chỉ định**



Hình 3.14: Lấy chỉ định

Hình 3.2.14: Lấy chỉ định

Sau khi lấy các phần tử chứa thông tin của sản phẩm, khởi tạo từ điển lưu những thông tin đã thu thập được và lưu vào **MongoDB**, riêng bộ sưu tập Sales trong **MongoDB** ta sẽ chỉ lưu những dược phẩm thuộc loại thuốc không kê đơn:

A screen shot of a computer program

Description automatically generated

Hình 3.2.15: Tạo từ điển lưu dữ liệu thu thập được sau đó lưu vào mongoDB

Hình 3.15: Tạo từ điển lưu dữ liệu thu thập được sau đó lưu vào MongoDB

### *Thu thập dữ liệu*

A screen shot of a computer program

Description automatically generatedSau khi đã định nghĩa được các hàm cần thiết cho việc thu thập và lưu trữ dữ liệu,việc còn lại bao gồm kết nối tới **MongoDB**, khởi tạo các collection để lưu trữ dữ liệu. Ta sử dụng **webdriver-manager** để tự động cài đặt và quản lý phiên bản mới nhất của **GeckoDriver** , dùng **Selenium WebDriver** để mở website Pharmacity.vn để thu thập dữ liệu:

Hình 3.16: Kết nối tới MongoDB, khởi tạo web driver, mở trang web thu thập dữ liệu

A computer screen shot of a computer code

Description automatically generatedViệc còn lại sau khi đã thực hiện các bước trên là gọi hàm, đầu tiên ta gọi hàm **load\_all\_products** nhằm mục tiêu lấy toàn bộ sản phẩm cần thu thập dữ liệu, sau đó gọi hàm **get\_product\_links** thu thập toàn bộ các đường liên kết dẫn đến từng trang dược phẩm, cuối cùng khởi tạo vòng lặp duyệt qua từng đường liên kết đã thu thập được gọi hàm **scrape\_product** để tự động thu thập dữ liệu và lưu trữ vào **MongoDB**.

Hình 3.17: Thu thập dữ liệu

### *Chuẩn hóa dữ liệu*

Sử dụng thư viện **Re**, thực hiện việc chuẩn hóa các loại số liệu như giá bán, số lượng yêu thích số lượng bán về định dạng số thực, số nguyên. Ví dụ:

A screenshot of a computer program

Description automatically generated

Hình 3.18: Chuẩn hóa dữ liệu

Hình 3.2.18: Chuẩn hóa dữ liệu

Việc chuẩn hóa những loại dữ liệu như giá bán, lượt yêu thích, số lượng bán ra là cần thiết để phục vụ các bước quản lý và xử lý dữ liệu một cách hiệu quả, hợp lý.

## Mô tả dữ liệu

Các dữ liệu dược phẩm thu thập được từ website Pharmacity.vn sẽ bao gồm những thuộc tính sau đây:

|  |  |  |
| --- | --- | --- |
| Tên biến | Mô tả | Kiểu dữ liệu |
| Product\_ID | Mã sản phẩm | String |
| Product\_Name | Tên sản phẩm | String |
| Img | Đường liên kết hình ảnh SP | String |
| Brand | Thương hiệu sản phẩm | String |
| Price | Giá sản phẩm | Float |
| Link | Liên kết dẫn tới website sản phẩm | String |
| Likes | Lượt yêu thích sản phẩm | Integer |
| Sold | Số lượng sản phẩm bán ra | Integer |
| Type | Loại sản phẩm | String |
| Product\_Spec | Quy cách sản phẩm | String |
| Product\_origin | Nơi sản xuất | String |
| Active\_element | Hoạt tính sản phẩm | String |
| Indication | Chỉ định sử dụng | String |

## Kết quả thực nghiệm

### *Kết quả thu thập*

Dữ liệu thu thập được từ website Pharmacity.vn lưu vào **MongoDB** gồm các thuộc tính mã sản phẩm, tên sản phẩm, liên kết hình ảnh, thương hiệu, giá bán, liên kết, lượt yêu thích, số lượng bán, loại dược phẩm, quy cách, nới sản xuất, hoạt tính, chỉ định sử dụng.

* Dữ liệu dược phẩm thu thập được: **491**
* Dữ liệu dược phẩm thuộc loại thuốc kê đơn: **228**
* Dữ liệu dược phẩm thuộc loại thuốc không kê đơn: **263**
* Các dữ liệu thu thập được bao gồm các thuộc tính sau:
* Mã sản phẩm
* Tên sản phẩm
* Liên kết hình ảnh
* Thương hiệu
* Giá bán
* Liên kết
* Lượt yêu thích
* Số lượng bán
* Loại dược phẩm
* Quy cách
* Nơi sản xuất
* Hoạt tính
* Chỉ định sử dụng

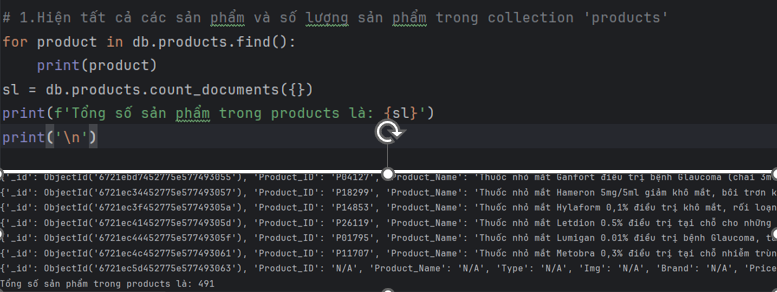
Một số thuộc tính của dữ liệu như thương hiệu thuốc vẫn chưa được thu thập hiệu quả, có nhiều dữ liệu bị thiếu hoạt lấy sai, cần cải thiện.

### *Phân tích dữ liệu*

Sau quá trình thu thập dữ liệu, ta tiến hành phân tích dữ liệu dược phẩm thu thập được nhằm tìm ra các thông tin như các loại thuốc được mua nhiều qua các tiêu chí nào, chức năng hay giá bán, nhãn hiệu thuốc được được tin dùng nhiều nhất vv..., để tìm và lọc các dữ liệu đã thu thập được cho quá trình phân tích ta thêm vào các query:

#### Các câu lệnh dùng để phân tích dữ liệu

* Hiện tất cả các sản phẩm và số lượng sản phẩm trong collection 'products'



*Hình 3.4 1: Hiện tất cả các sản phẩm và số lượng sản phẩm trong collection 'products'*

Hình 3.19: Hiện tất cả các sản phẩm và số lượng sản phẩm trong collection 'products'

* Tìm sản phẩm không kê đơn có giá cao nhất



Hình 3.20: Tìm sản phẩm không kê đơn có giá cao nhất

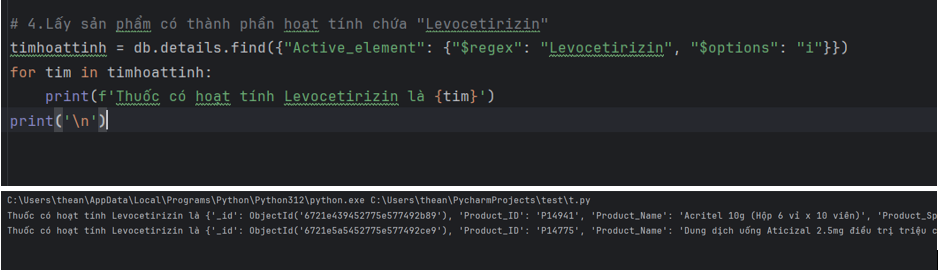
Hình 3.4 2: Tìm sản phẩm không kê đơn có giá cao nhất

* Tìm sản phẩm không kê đơn có giá thấp nhất



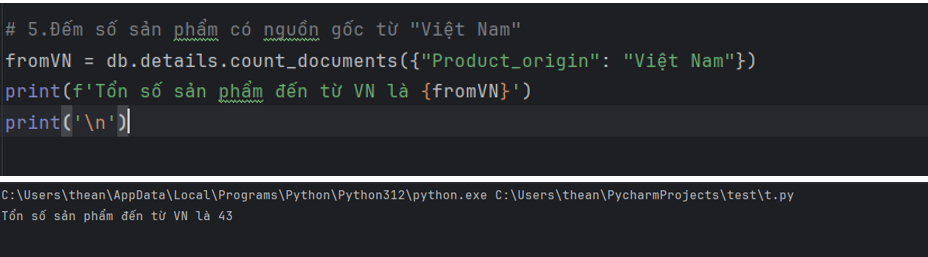
Hình 3.21: Tìm sản phẩm không kê đơn có giá thấp nhất

* Lấy sản phẩm có thành phần hoạt tính chứa "Levocetirizin"



Hình 3.22: Lấy sản phẩm có thành phần hoạt tính chứa "Levocetirizin"

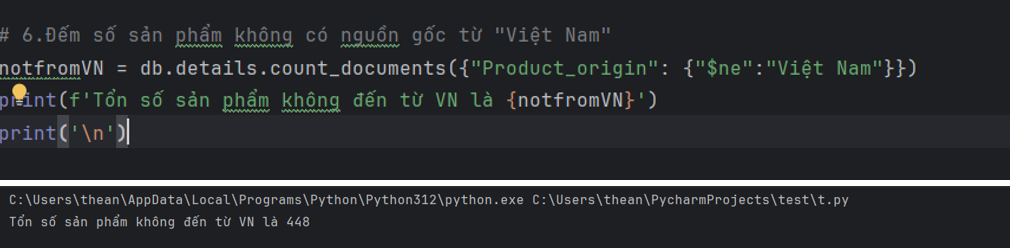
* Đếm số sản phẩm có nguồn gốc từ "Việt Nam"



Hình 3.4 4: Đếm số sản phẩm có nguồn gốc từ "Việt Nam"

Hình 3.23: Đếm số sản phẩm có nguồn gốc từ "Việt Nam"

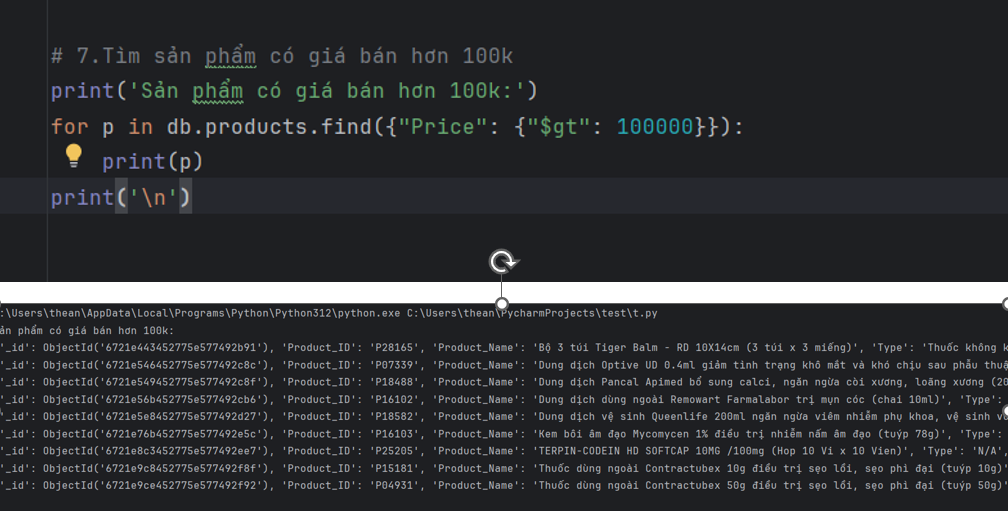
* Đếm số sản phẩm không có nguồn gốc từ "Việt Nam"



Hình 3.4 5: Đếm số sản phẩm không có nguồn gốc từ "Việt Nam"

Hình 3.24: Đếm số sản phẩm không có nguồn gốc từ "Việt Nam"

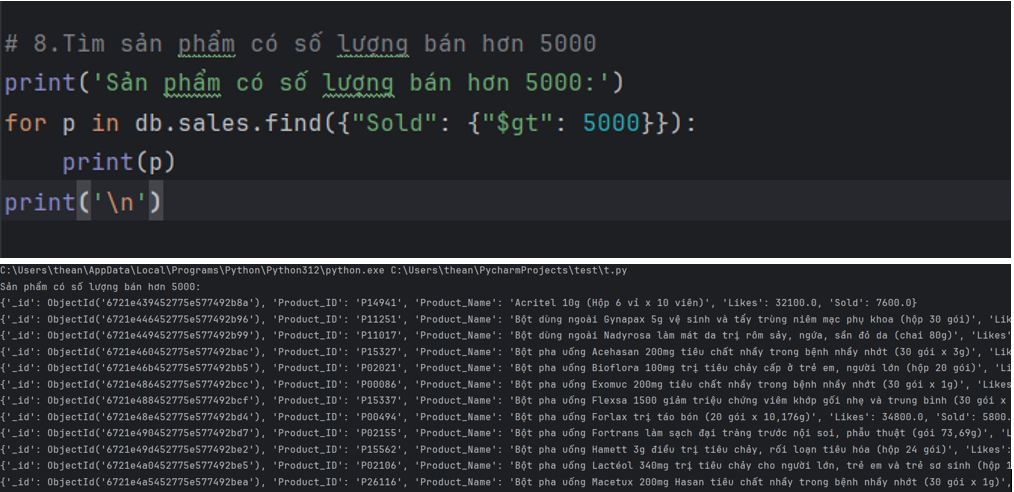
* Tìm sản phẩm có giá bán hơn 100k



Hình 3.4 6: Tìm sản phẩm có giá bán hơn 100k

Hình 3.25: Tìm sản phẩm có giá bán hơn 100k

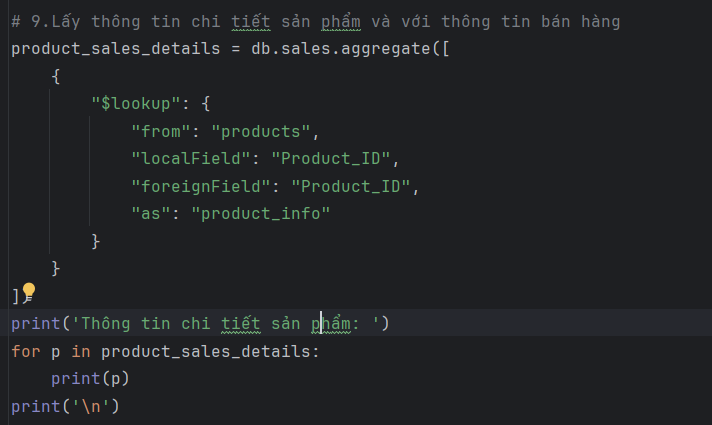
* Tìm sản phẩm có số lượng bán hơn 5000



Hình 3.4 7: Tìm sản phẩm có số lượng bán hơn 5000

Hình 3.26: Tìm sản phẩm có số lượng bán hơn 5000

* Lấy thông tin chi tiết sản phẩm và với thông tin bán hàng



Hình 3.27: Lấy thông tin chi tiết sản phẩm và với thông tin bán hàng

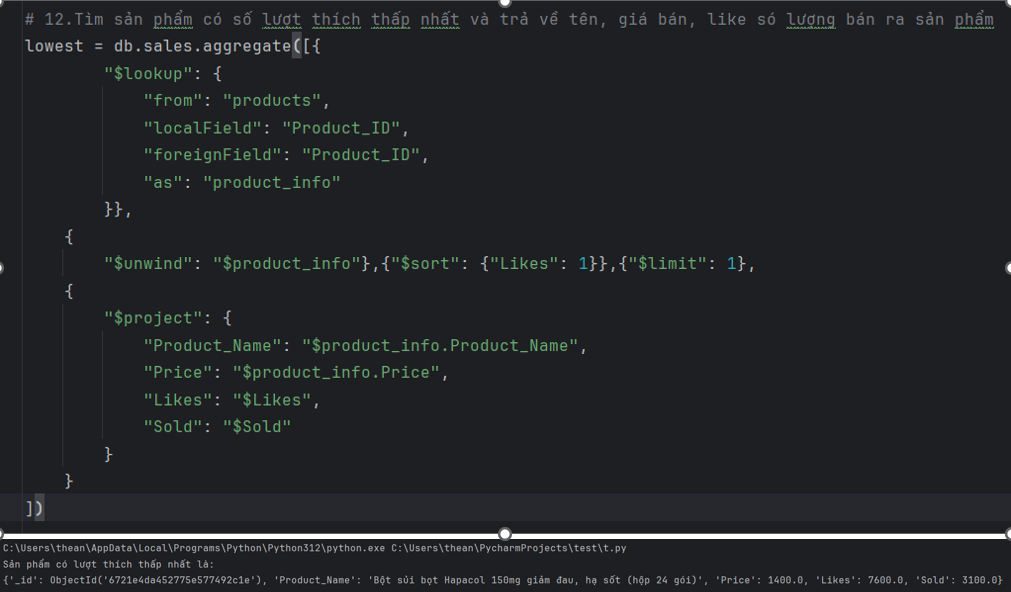
* Tìm sản phẩm có tên chứa từ khóa Eagle



Hình 3.4 9: Tìm sản phẩm có tên chứa từ khóa Eagle

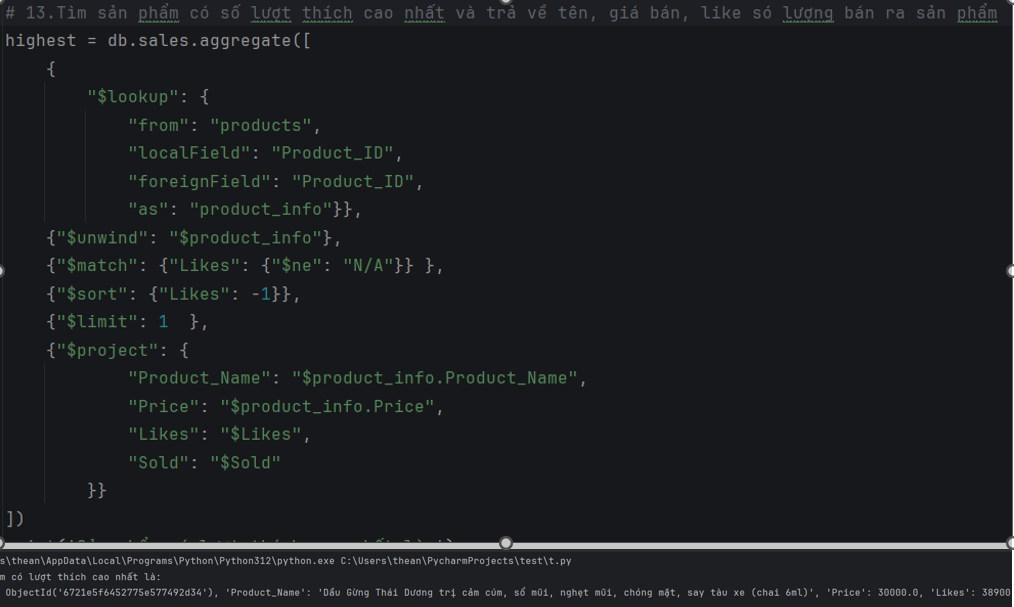
* Tìm sản phẩm theo Product\_ID

Hình 3.4 10: Tìm sản phẩm theo Product\_ID

* Tìm sản phẩm có số lượt thích thấp nhất và trả về tên, giá bán, like số lượng bán ra sản phẩm

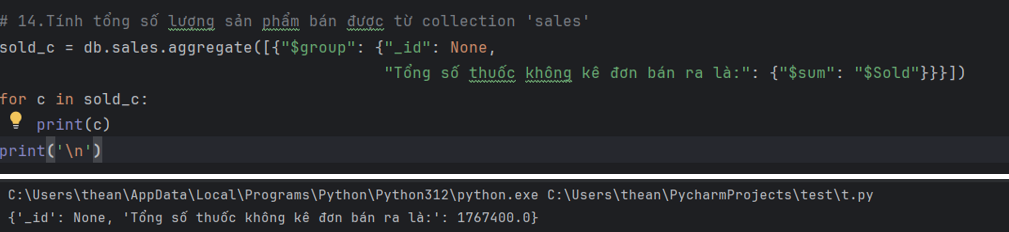
Hình 3.4 11: Tìm sản phẩm có số lượt thích thấp nhất và trả về tên, giá bán, like số lượng bán ra sản phẩm

* Tìm sản phẩm có số lượt thích cao nhất và trả về tên, giá bán, like số lượng bán ra sản phẩm



Hình 3.28: Tìm sản phẩm có số lượt thích cao nhất và trả về tên, giá bán, like số lượng bán ra sản phẩm

* Tính tổng số lượng sản phẩm bán được từ collection 'sales'



Hình 3.29: Tính tổng số lượng sản phẩm bán được từ collection 'sales'

* Tính tổng số tiền bán thuốc không kê đơn



Hình 3.30: Tính tổng số tiền bán thuốc không kê đơn

#### Kết luận phân tích

* Giá bán các loại dược phẩm nằm trong khoảng từ **1.000VND** đến **550.000VND**
* Tổng số lượng thuốc không kê đơn bán ra là: **1.767.400**
* Loại thuốc được mua nhiều trên 10000 được chỉ định sử dụng điều trị các triệu chứng viêm loét dạ dày tá tràng, viêm thực quản hay khô mắt, rửa mắt, sinh mũ đến từ các thương hiệu **Sanofi CHC import**, **Pharmedic**
* Tỉ lệ các loại thuốc có giá bán lẻ dưới và trên 100k lần lượt là **96.58%** và **3.42%** trên tổng số dữ liệu thuộc loại thuốc không kê đơn thu thập được
* Thuốc có doanh thu cao nhất trong các dữ liệu thu thập được là thuốc “Thuốc dùng ngoài Contractubex 50g điều trị sẹo lồi, sẹo phì đại (tuýp 50g)” với số tiền bán thuốc là **3.465.000.000VND**
* Thuốc có doanh thu thấp nhất trong các dữ liệu thu thập được là thuốc “Bột sủi bọt Hapacol 150mg giảm đau, hạ sốt (hộp 24 gói)” với số tiền bán thuốc là **4340000** VND

### *Đánh giá ưu và nhược điểm của việc thu thập dữ liệu bằng công cụ mã nguồn mở Selenium WebDriver*

Ưu điểm:

* Ưu điểm của **Selenium WebDriver** và **MongoDB** là giúp quá trình duyệt web và thu thập dữ liệu một cách tự động, hỗ trợ nhiều ngôn ngữ lập trình kết hợp với các thư viện mã nguồn mở khác giúp chuẩn hóa dữ liệu thu thập được và lưu trữ vào **MongoDB** một công cụ giúp lưu trữ và xử lý dữ liệu một cách linh hoạt, hiệu quả.
* Hiệu suất thu thập dữ liệu sử dụng Selenium nằm trong khoảng từ 0,5 đến 2s đối với một trang web tùy thuộc vào nhiều yếu tố như thời gian tải trang, các **Xpath**, CSS **Selector** được dùng hiệu quả hay không, độ phức tạp của trang web vv…
* **Selenium** giúp thực hiện nhiều tác vụ phức tạp một cách hiệu quả nhờ hỗ trợ nhiều thao tác duyệt web như điều hướng, click, nhập liệu.

Nhược điểm:

* **Selenium** **WebDriver** dựa trên trình duyệt để thao tác thu thập dữ liệu vì vậy tốc độ thu thập dữ liệu sẽ kém hiệu quả hơn nhiều so với **Scrapy** hay **BeautifulSoup** các công cụ thu thập dữ liệu làm việc ở tầng mạng
* Không thể thu thập dữ liệu dạng đồ họa như hình ảnh, video.

## Kết luận

Thông qua thực nghiệm thu thập và phân tích dữ liệu dược phẩm từ website Pharmacity.vn sử dụng công cụ mã nguồn mở **Selenium** **WebDriver** và **MongoDB** có thể kết luận **Selenium** và **MongoDB** là hai công cụ mã nguồn mở hiệu quả và hữu ích trong việc duyệt web để thu thập và lưu trữ xử lý dữ liệu, tuy còn nhiều hạn chế nếu phải thu thập dữ liệu từ những website phức tạp nhưng có thể khắc phục bằng cách kết hợp chung với các công cụ mã nguồn mở khác.

# KẾT LUẬN VÀ KIẾN NGHỊ

## Kết luận

Đối với thời kì công nghệ số đang phát triển không ngừng, công việc thu thập và xử lý dữ liệu trực tuyến đã trở thành yếu tố quyết định trong hoạt động phát triển của nhiều doanh nghiệp. Trong dự án của nhóm chúng tôi, nhóm đã sử dụng công cụ Selenium và MongoDB để tự động hóa thu thập dữ liệu đồng thời xử lý dữ liệu từ trang web nhà thuốc Pharmacity, giúp thu thập được thông tin một cách nhanh chóng và chính xác, tối ưu hóa được quy trình làm thủ công. Kết quả mà nhóm tôi đạt được trong dự án bao gồm:

* Tính tự động hóa cao: Selenium hỗ trợ tự động hóa thu thập dữ liệu, giảm thời gian và nhân lực so với quá trình thu thập dữ liệu thủ công, mang lại hiệu quá đáng kể trong việc lấy được các dữ liệu từ các trang web.
* Hiệu quả trong lưu trữ và phân tích: Dữ liệu sau khi thu thập được từ trang web trực tuyến của nhà thuốc Pharmacity được lưu trữ và quản lý trong MongoDB. Ở đây, cho phép phân tích dữ liệu một cách dễ dàng từ đó góp phần giúp doanh nghiệp hiểu rõ hơn về xu hướng thị trường để tối ưu hóa được chiến lược kinh doanh.

Mặc dù Selenium rất hữu ích trong việc thu thập dữ liệu nhưng công cụ này cũng gặp không ít khó khăn khi làm việc với các trang web yêu cầu phải đăng nhập duy trì. Điều này cho thấy được tiềm năng tích hợp thêm các công cụ hỗ trợ như Scrapy để mở rộng khả năng thu thập dữ liệu của hệ thống.

## Kiến nghị

Dựa trên quá trình thực nghiệm và kết quả phân tích, nhóm để xuất một số hướng phát triển để nâng cao hiệu quả của Selenium và MongoDB trong thu thập và phân tích dữ liệu:

### *Tích hợp các công cụ hỗ trợ xử lý trang động*

Đối với các trang web sử dụng JavaScript để tải nội dung ta có thể kết hợp thêm Scrapy giúp mở rộng được khả năng thu thập dữ liệu, không chỉ còn giới hạn bởi trang web tĩnh. Điều này sẽ giúp tăng tốc độ linh hoạt hơn trong việc thu thập dữ liệu từ các trang web động phổ biến.

### *Tối ưu hóa hiệu suất thu thập dữ liệu*

Tối ưu hóa mã Selenium và sử dụng middleware để kiểm soát yêu cầu và phản hồi. Điều này giúp quá trình thu thập dữ liệu nhanh hơn và giảm thiểu lỗi, đảm bảo được hệ thống hoạt động một cách hiệu quả.

### *Phát triển mô hình phân tích xu hướng*

Sử dụng các dữ liệu đã thu thập được và phát triển mô hình dự báo xu hướng thị trường nhờ trên dữ liệu. Áp dụng các thuật toán học máy và phân tích dữ liệu sẽ cho ra được dự đoán chính xác hơn, hỗ trợ đưa ra chiến lược quyết định cho doanh nghiệp.

### *Sử dụng cơ sở dữ liệu linh hoạt*

Để quản lý lượng dữ liệu lớn, ta dùng MongoDB hoặc các cơ sở dữ liệu phi cấu trúc khác sẽ giúp dữ liệu lưu trữ linh hoạt hơn. Kết hợp động thời các công cụ phân tích như Pandas hoặc Numpy để hỗ trợ xử lý và phân tích dữ liệu nhanh chóng và chính xác.

### *Đào tạo nhân sự chuyên môn*

Đào tạo đội ngũ nhân sự để đảm bảo nhân viên có kiến thức về Python và hiểu rõ cấu trúc web. Việc này sẽ tối ưu hóa quy trình thu thập dữ liệu và phân tích, từ đó sẽ đáp ứng tốt hơn nhu cầu của doanh nghiệp.

# TÀI LIỆU THAM KHẢO

[1] “Selenium (software),” *Wikipedia*. Sep. 27, 2024. Accessed: Oct. 24, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Selenium\_(software)&oldid=1248044578

[2] “Components of the Selenium Automation Tool - DZone,” dzone.com. Accessed: Oct. 24, 2024. [Online]. Available: https://dzone.com/articles/components-of-selenium-automation-tool

[3] FPT C. ty C. phần B. lẻ K., “Selenium là gì? Tìm hiểu các tính năng nổi bật của Selenium trong lĩnh vực phần mềm.” Accessed: Oct. 22, 2024. [Online]. Available: https://fptshop.com.vn/tin-tuc/danh-gia/selenium-la-gi-167783

[4] “Selenium WebDriver: Nó là gì, hoạt động như thế nào và bạn có cần nó không.” Accessed: Oct. 24, 2024. [Online]. Available: https://inventorsoft.co/blog/selenium-webdriver-how-it-works

[5] Team C., “What is Selenium?,” Codecademy Blog. Accessed: Oct. 24, 2024. [Online]. Available: https://www.codecademy.com/resources/blog/what-is-selenium/

[6] Technology) S. K. (MSC in I., “Practical Applications of Selenium in IT,” Intelli Mindz. Accessed: Oct. 24, 2024. [Online]. Available: https://intellimindz.com/practical-applications-of-selenium/

[7] Kumar R., “What is Selenium and How it works? An Overview and Its Use Cases,” DevOpsSchool.com. Accessed: Oct. 24, 2024. [Online]. Available: https://www.devopsschool.com/blog/what-is-selenium-and-how-it-works-an-overview-and-its-use-cases/

[8] “MongoDB,” *Wikipedia tiếng Việt*. Apr. 17, 2023. Accessed: Oct. 28, 2024. [Online]. Available: https://vi.wikipedia.org/w/index.php?title=MongoDB&oldid=69892072

[9] L.B <hi@ngoclb.com> N. and Uyen T., “MongoDB là gì? Định nghĩa và Hiểu rõ A-Z về MongoDB,” ITviec Blog. Accessed: Oct. 28, 2024. [Online]. Available: https://itviec.com/blog/mongodb-la-gi/

[10] D. T. Được, “Thiết kế cơ sở dữ liệu bằng MongoDB sao cho chuẩn,” Dư Thanh Được. Accessed: Oct. 28, 2024. [Online]. Available: https://duthanhduoc.com/blog/thiet-ke-co-so-du-lieu-voi-mongodb

[11] “Nhom3 Tìm Hieu Mongo DB - ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN - Studocu.” Accessed: Oct. 28, 2024. [Online]. Available: https://www.studocu.com/vn/document/truong-dai-hoc-cong-nghe-thong-tin-dai-hoc-quoc-gia-thanh-pho-ho-chi-minh/nhap-mon-cong-nghe-phan-mem/nhom3-tim-hieu-mongo-db/46356795

[12] “Mô tả về Data Models trong MongoDB.” Accessed: Oct. 28, 2024. [Online]. Available: https://viblo.asia/p/mo-ta-ve-data-models-trong-mongodb-7ymwGXA0v4p1

[13] “MongoDB là gì? Các tính năng nổi bật của MongoDB - Viblo.” Accessed: Oct. 28, 2024. [Online]. Available: https://viblo.asia/p/mongodb-la-gi-cac-tinh-nang-noi-bat-cua-mongodb-PAoJex2N41j

[14] L.B <hi@ngoclb.com> N. and Uyen T., “MongoDB là gì? Định nghĩa và Hiểu rõ A-Z về MongoDB,” ITviec Blog. Accessed: Oct. 28, 2024. [Online]. Available: https://itviec.com/blog/mongodb-la-gi/

[15] “Cassandra so với MongoDB – Sự khác biệt giữa các cơ sở dữ liệu NoSQL – AWS,” Amazon Web Services, Inc. Accessed: Oct. 28, 2024. [Online]. Available: https://aws.amazon.com/vi/compare/the-difference-between-cassandra-and-mongodb/

[16] “MongoDB là gì? 9 Phần mềm quản trị Mongodb nên sử dụng 2024.” Accessed: Oct. 28, 2024. [Online]. Available: https://prodima.vn/mongodb-la-gi/

# PHỤ LỤC

* **Code của dự án:**
* from selenium import webdriver  
  from selenium.webdriver.firefox.service import Service  
  from webdriver\_manager.firefox import GeckoDriverManager  
  from selenium.webdriver.common.by import By  
  from pymongo import MongoClient  
  import time  
  import re  
    
  # Kết nối MongoDB  
  client = MongoClient("mongodb://localhost:27017/")  
  db = client['pharmacityDB']  
  client.drop\_database('pharmacityDB')  
    
  products\_collection = db['products']  
  sales\_collection = db['sales']  
  products\_detail = db['details']  
    
  # Khởi tạo WebDriver cho Firefox  
  driver = webdriver.Firefox(service=Service(GeckoDriverManager().install()))  
    
  # Truy cập vào trang dược phẩm  
  driver.get("https://www.pharmacity.vn/duoc-pham")  
  time.sleep(3)  
    
    
  # Hàm cào dữ liệu từng sản phẩm  
  def scrape\_product(product\_link):  
   # Mở link sản phẩm  
   driver.get(product\_link)  
   time.sleep(2)  
    
   # Lấy mã sản phẩm  
   try:  
   product\_code = driver.find\_element(By.CSS\_SELECTOR, "p.text-sm.leading-5.text-neutral-600").text  
   except:  
   product\_code = "N/A"  
    
   # Lấy tên sản phẩm  
   try:  
   product\_name = driver.find\_element(By.CSS\_SELECTOR, "h1.text-neutral-900.font-semibold").text  
   except:  
   product\_name = "N/A"  
    
   # Lấy hình ảnh  
   try:  
   product\_img = driver.find\_element(By.XPATH,  
   '//\*[@id="mainContent"]/div/div[1]/div[3]/div[1]/div[1]/div[1]/div/div[1]/div/div/div[1]/div/img').get\_attribute(  
   'src')  
   except:  
   product\_img = "N/A"  
    
   # Lấy nơi sản xuất  
   try:  
   product\_origin = driver.find\_element(By.CSS\_SELECTOR,  
   '#mainContent > div > div:nth-child(1) > div.relative.grid.grid-cols-1.gap-6.md\\:container.md\\:grid-cols-\\[min\\(60\\%\\,calc\\(555rem\\/16\\)\\)\\,1fr\\].md\\:pt-6.lg\\:grid-cols-\\[min\\(72\\%\\,calc\\(888rem\\/16\\)\\)\\,1fr\\] > div.grid.md\\:gap-6 > div.grid.grid-cols-1.items-start.md\\:gap-6.lg\\:grid-cols-2.xl\\:grid-cols-2 > div:nth-child(2) > div > div.flex.flex-col.px-4.md\\:px-0 > div.gap-3.md\\:gap-4.mb-3.grid.md\\:mb-4 > div.grid.gap-3.md\\:gap-2 > div:nth-child(5) > div').text  
   except:  
   product\_origin = "N/A"  
    
   # Lấy thương hiệu  
   try:  
   product\_brand = driver.find\_element(By.XPATH, '//\*[@id="mainContent"]/div/div[1]/div[3]/div[1]/div[1]/div[2]/div/div[3]/div[2]/div/a').text  
   except:  
   product\_brand = "N/A"  
    
   # Lấy giá bán  
   try:  
   product\_price = driver.find\_element(By.XPATH, '//\*[@id="mainContent"]/div/div[1]/div[3]/div[1]/div[1]/div[2]/div/div[3]/div[3]/h3').text  
   # Sử dụng regex để loại bỏ các ký tự không phải số và dấu phân cách thập phân  
   # Giữ lại số và dấu chấm (.)  
   cleaned\_price = re.search(r"\d+\.\d+", product\_price)  
   cleaned\_price\_str = cleaned\_price.group()  
   product\_price = float(cleaned\_price\_str)\*1000  
   except:  
   product\_price = "N/A"  
    
   # Lấy lượt yêu thích  
   try:  
   product\_likes = driver.find\_element(By.CSS\_SELECTOR, 'div.space-x-1:nth-child(2) > p:nth-child(1)').text  
   cleaned\_likes\_str = re.sub(r'[^\d.]', '', product\_likes)  
   product\_likes = float(cleaned\_likes\_str) \* 1000  
   except:  
   product\_likes = "N/A"  
    
   # Lấy số lượng bán  
   try:  
   product\_sold = driver.find\_element(By.CSS\_SELECTOR, 'p.text-sm:nth-child(3)').text  
   cleaned\_sold\_str = re.sub(r'[^\d.]', '', product\_sold)  
   product\_sold = float(cleaned\_sold\_str) \* 1000  
   except:  
   product\_sold = "N/A"  
    
   # Lấy loại thuốc  
   try:  
   product\_type = driver.find\_element(By.CSS\_SELECTOR, "div.md\\:text-base").text  
   except:  
   product\_type = "N/A"  
    
   # Lấy quy cách  
   try:  
   product\_spec = driver.find\_element(By.CSS\_SELECTOR, "h1.text-neutral-900.font-semibold").text  
   ps = re.search(r'\((.\*?)\)', product\_spec)  
   product\_spec = ps.group(1)  
   except:  
   product\_spec = "N/A"  
    
   # Lấy hoạt tính  
   try:  
   active\_element = driver.find\_element(By.CSS\_SELECTOR, "#mainContent > div > div:nth-child(1) > div.relative.grid.grid-cols-1.gap-6.md\\:container.md\\:grid-cols-\\[min\\(60\\%\\,calc\\(555rem\\/16\\)\\)\\,1fr\\].md\\:pt-6.lg\\:grid-cols-\\[min\\(72\\%\\,calc\\(888rem\\/16\\)\\)\\,1fr\\] > div.grid.md\\:gap-6 > div.grid.grid-cols-1.items-start.md\\:gap-6.lg\\:grid-cols-2.xl\\:grid-cols-2 > div:nth-child(2) > div > div.flex.flex-col.px-4.md\\:px-0 > div.gap-3.md\\:gap-4.mb-3.grid.md\\:mb-4 > div.grid.gap-3.md\\:gap-2 > div:nth-child(2) > div").text  
   except:  
   active\_element = "N/A"  
    
   #Lấy chỉ định  
   try:  
   indication = driver.find\_element(By.CSS\_SELECTOR, "#mainContent > div > div:nth-child(1) > div.relative.grid.grid-cols-1.gap-6.md\\:container.md\\:grid-cols-\\[min\\(60\\%\\,calc\\(555rem\\/16\\)\\)\\,1fr\\].md\\:pt-6.lg\\:grid-cols-\\[min\\(72\\%\\,calc\\(888rem\\/16\\)\\)\\,1fr\\] > div.grid.md\\:gap-6 > div.grid.grid-cols-1.items-start.md\\:gap-6.lg\\:grid-cols-2.xl\\:grid-cols-2 > div:nth-child(2) > div > div.flex.flex-col.px-4.md\\:px-0 > div.gap-3.md\\:gap-4.mb-3.grid.md\\:mb-4 > div.grid.gap-3.md\\:gap-2 > div:nth-child(3) > div").text  
   except:  
   indication = "N/A"  
    
   # Tạo từ điển lưu thông tin sản phẩm  
   product\_data = {  
   "Product\_ID": product\_code,  
   "Product\_Name": product\_name,  
   "Type": product\_type,  
   "Img": product\_img,  
   "Brand": product\_brand,  
   "Price": product\_price,  
   "Link": product\_link  
   }  
   sale\_data = {  
   "Product\_ID": product\_code,  
   "Product\_Name": product\_name,  
   "Likes": product\_likes,  
   "Sold": product\_sold  
   }  
    
   detail\_data = {  
   "Product\_ID": product\_code,  
   "Product\_Name": product\_name,  
   "Product\_Spec": product\_spec,  
   "Product\_origin": product\_origin,  
   "Active\_element": active\_element,  
   "Indication": indication  
   }  
    
   # Lưu vào MongoDB  
   products\_collection.insert\_one(product\_data)  
   products\_detail.insert\_one(detail\_data)  
   if product\_type == "Thuốc không kê đơn":  
   sales\_collection.insert\_one(sale\_data)  
    
   print(f"Đã lưu: {product\_name}")  
    
  # Hàm cuộn xuống cuối trang và nhấn nút xem thêm  
  def load\_all\_products():  
   while True:  
   driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")  
   time.sleep(2)  
   try:  
   # Tìm nút "Xem thêm" và nhấn vào nó  
   load\_more\_button = driver.find\_element(By.XPATH, "//button[span[contains(text(), 'Xem thêm')]]")  
   load\_more\_button.click()  
   time.sleep(2)  
   except:  
   break  
    
  # Lấy danh sách link sản phẩm  
  def get\_product\_links():  
   product\_links = []  
   try:  
   products = driver.find\_elements(By.CSS\_SELECTOR, "a:has(h3.line-clamp-2.h-10.text-sm.font-semibold)")  
   for product in products:  
   product\_link = product.get\_attribute("href")  
   product\_links.append(product\_link)  
   print(product\_link)  
   except Exception as e:  
   print(f"Error: {e}")  
   return product\_links  
    
    
    
    
  load\_all\_products()  
  links = get\_product\_links()  
  print(f'Tổng số link sản phẩm {len(links)} \n')  
    
  # Cào dữ liệu từ trang web  
  for link in links:  
   try:  
   scrape\_product(link)  
   except:  
   print("Lỗi")  
  driver.quit()
* **Lịch sử commit**

