BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

THU THẬP VÀ PHÂN TÍCH DỮ LIỆU TỪ TRANG CÁ NHÂN TRÊN MẠNG XÃ HỘI X.COM (TWITTER.COM) SỬ DỤNG CÔNG CỤ SELENIUM 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 :

2286400022 - Hồ Nguyễn Hoàng Phát

2286400042 - Trần Lê Vân

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2024

HUTECH
Đại học Công nghệ Tp.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

THU THẬP VÀ PHÂN TÍCH DỮ LIỆU TỪ TRANG CÁ NHÂN TRÊN MẠNG XÃ HỘI X.COM (TWITTER.COM) SỬ DỤNG CÔNG CỤ SELENIUM 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 :

2286400022 - Hồ Nguyễn Hoàng Phát

2286400042 - Trần Lê Vân

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2024

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TPHCM, ngày……tháng……năm 2024

Giáo viên hướng dẫn

(Ký tên, đóng dấu)

LỜI CAM ĐOAN

Chúng tôi, Hồ Nguyễn Hoàng Phát, Trần Lê Vân xin cam đoan rằng:

Tất cả thông tin và phân tích trình bày trong báo cáo này được thực hiện một cách chính xác và trung thực.

Mọi dữ liệu, nhận định hoặc ý kiến được trích dẫn từ các nguồn khác đều đã được nêu rõ nguồn gốc và trích dẫn đúng quy định. Tôi cam đoan rằng không có bất kỳ hành vi sao chép hoặc sử dụng thông tin không hợp pháp nào từ các nguồn khác.

Bài báo cáo này là kết quả của công trình nghiên cứu độc lập của chúng tôi và chưa từng được công bố tại bất kỳ nơi nào khác. Tôi cam đoan đã tuân thủ nghiêm ngặt các quy tắc và quy định của môn học, bao gồm việc tham khảo và áp dụng các công cụ nghiên cứu một cách hợp lệ. Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình.

Tôi hy vọng rằng bài báo cáo này sẽ cung cấp những thông tin hữu ích cho các nhà nghiên cứu , doanh nghiệp. Góp phần vào việc hiểu rõ hơn về mạng xã hội ngày nay.

TPHCM, ngày……tháng 10 năm 2024

Sinh viên

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU , CÁC CHỮ VIẾT TẮT

Web Scraping: Trích xuất dữ liệu web tự động

NoSQL: Hệ quản trị cơ sở dữ liệu không quan hệ

GDPR: Quy định chung về bảo vệ dữ liệu

ToS: Điều khoản dịch vụ

MongoDB: Hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở

HTML (Hypertext Markup Language): Ngôn ngữ đánh dấu siêu văn bản dùng để cấu tạo web

Numpy, Pandas, Scikit-learn, Django, Flask: Thư viện Python

Global Interpreter Lock (GIL): là một cơ chế trong Python ngăn nhiều luồng thực thi mã Python cùng một lúc. Điều này hạn chế tính song song và đồng thời của một số ứng dụng.

Framework: Là code được viết sẵn và tạo thành bộ khung, hay khung phần mềm.

GUI: Giao diện đồ họa người dùng

Tkinter, PyQt, Kivy, wxWidgets: Các thư viện của Python

Automation testing framework: Bộ khung kiểm thử tự động

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu đề tài

Mạng xã hội đã trở thành một phần không thể thiếu trong cuộc sống hiện nay của chúng ta. Mỗi ngày có hàng triệu dữ liệu được tạo ra và chia sẻ trên các nền tảng trực tuyến. Dữ liệu trên mạng xã hội như một kho tàng vô giá, đóng vai trò quan trọng trong việc phân tích hành vi người dùng và dự đoán xu thế. Đề tài tập trung vào phát triển và áp dụng kỹ thuật thu thập dữ liệu (Web Scraping) bằng Selenium, một công cụ giúp hỗ trợ và cho phép mô phỏng các hành động của người dùng trên trình duyệt, thu thập dữ liệu từ các trang web một cách tự động. Từ đó, dữ liệu thu thập sẽ được phân tích tìm thấy các yếu tố cần thiết như hành vi người dùng, mức độ tương tác cùng các yếu tố khác.

1.2. Nhiệm vụ đồ án

Nhiệm vụ của đề tài là hướng tới nghiên cứu và ứng dụng kĩ thuật Web Scraping nhằm trích xuất dữ liệu từ các trang cá nhân trên mạng xã hội X(Twitter). Qua đó, phục vụ cho việc phân tích và phát hiện các xu thế, hiểu rõ hành vi của người tiêu dùng trên mạng xã hội.

1.2.1. Tính cấp thiết và lý do hình thành đề tài

Trong kỷ nguyên kỹ thuật số, mạng xã hội X(Twitter) đã trở thành một diễn đàn sôi động, nơi hàng triệu người dùng chia sẻ thông tin, ý kiến và tương tác với nhau mỗi ngày. Những nội dung như bài đăng, bình luận, lượt thích, và các hành động trực tuyến khác không chỉ phản ánh quan điểm cá nhân mà còn gợi ra những vấn đề xã hội, thị hiếu tiêu dùng và tình cảm cộng động. Những dữ liệu này ẩn chứa nguồn thông tin quý giá về hành vi, thái độ và khuynh hướng của người tiêu dùng, là nguồn tài nguyên phong phú cho các nhà nghiên cứu và nhà hoạch định chính sách trong việc hiểu sâu sắc hơn về thị hiếu, hành vi và xu hướng thay đổi của con người.

Tuy nhiên, việc thu thập và phân tích dữ liệu trên mạng xã hội X(Twitter) cũng còn nhiều khó khăn bởi sự biến đổi chóng mặt của dữ liệu. Quá trình khai thác dữ liệu thủ công từ các trang cá nhân không những tốn nhiều thời gian và công sức mà còn có thể vấp phải các rào cản kỹ thuật như việc xây dựng trang web với nhiều lớp mã JavaScript phức tạp. Hơn nữa, các công cụ phân tích dữ liệu truyền thống khó có thể xử lý và khai thác hiệu quả lượng thông tin không có cấu trúc và thay đổi liên tục này.

Do đó, kỹ thuật Web Scraping sử dụng công cụ Selenium, một công cụ tự động hóa trình duyệt web, mang lại giải pháp hữu hiệu giúp vượt qua những trở ngại này. Selenium cho phép mô phỏng tất cả các thao tác của người dùng trên trình duyệt, từ việc đăng nhập cho đến tương tác với các phần tử trên trang web, từ đó có thể lấy thông tin từ những trang web động mà các phương pháp truyền thống không thể tiếp cận.

Nghiên cứu này được tiện hành nhằm thỏa mãn nhu cầu cấp bách về một phương pháp hiệu quả để thu thập và phân tích dữ liệu trên mạng xã hội X(Twitter), từ đó khám phá được giá trị tiềm ẩn trong dữ liệu này. Thông qua việc sử dụng Selenium, đề tài hướng tới khám phá những giá trị tiềm ẩn trong dữ liệu giúp các nhà nghiên cứu có thể dự đoán xu thế thị trường, và các tổ chức có thể đưa ra các quyết định đúng đắn hơn dựa trên dữ liệu. Với sự tiến bộ của kĩ thuật thu thập dữ liệu và công nghệ tự động hoá qui trình, đề tài đã mở ra cơ hội mới cho việc sử dụng dữ liệu mạng xã hội trong nhiều lĩnh vực khác nhau, từ tiếp thị trực tuyến đến nghiên cứu khoa học và quản lí xã hội.

1.2.2. Ý nghĩa khoa học và thực tiễn của đề tài

Ý nghĩa khoa học: Đề tài này đóng góp vào việc củng cố và mở rộng kiến thức về kỹ thuật Web Scraping và cách ứng dụng Selenium trong thu thập dữ liệu từ các trang web động, đặc biệt là trên mạng xã hội. Bằng cách nghiên cứu quá trình thu thập dữ liệu tự động, đề tài giúp làm rõ các phương pháp và công cụ hiện đại, đồng thời có cái nhìn sâu sắc hơn về những vấn đề và giải pháp đối với việc xử lý dữ liệu lớn từ môi trường internet. Kết quả của nghiên cứu sẽ tạo tiền đề thúc đẩy

các công trình kế tiếp trong việc phân tích dữ liệu, nhất là việc áp dụng công nghệ thông tin để phân tích hành vi người dùng và phân tích dữ liệu mạng xã hội.

Ý nghĩa thực tiễn: Nghiên cứu này mang lại ý nghĩa thực tiễn sâu sắc trên nhiều phương diện. Đối với các nhà bán lẻ, kết quả thu thập và phân tích thông tin trên mạng xã hội cung cấp dữ liệu hữu ích về hành vi tiêu dùng, thói quen mua sắm, và cảm xúc xã hội. Điều này giúp họ hoạch định chiến lược marketing hiệu quả hơn, tối ưu chiến dịch quảng cáo. Ngoài ra, nghiên cứu còn hỗ trợ các nhà nghiên cứu xã hội trong việc phân tích xu hướng, hành vi cộng đồng và các vấn đề xã hội, góp phần vào quá trình ra quyết định dựa trên dữ liệu chính xác.

1.3. Mục tiêu nghiên cứu

1.3.1. Mục tiêu tổng quát

Đề tài nhằm xây dựng một hệ thống Web Scraping tự động, linh hoạt và hiệu quả sử dụng Selenium để thu thập thông tin từ trang cá nhân trên mạng xã hội X(Twitter). Sau đó, sử dụng dữ liệu thu thập được để khám phá những xu hướng và hành vi của người dùng trên nền tảng X.

1.3.2. Mục tiêu cụ thể

Trong bài viết này, chúng tôi xây dựng một kích bản tự động dùng Selenium để thu thập dữ liệu từ trang cá nhân trên mạng xã hội X(Twitter), bao gồm các dữ liệu công khai như bài đăng, ngày đăng bài, số reaction, bình luận, lượt chia sẻ. Tổ chức dữ liệu lại theo cấu trúc có thể phân tích. Lưu trữ dữ liệu đã thu thập vào cơ sở dữ liệu NoSQL, nhằm khai thác tính năng lưu trữ dữ liệu linh động và hiệu quả của MongoDB. Sử dụng truy vấn NoSQL trên MongoDB để thực hiện phân tích dữ liệu, khám phá xu hướng, hành vi của người dùng và các yếu tố như tương tác và nội dụng. Đánh giá hiệu quả của hệ thống Web Scraping dựa trên tốc độ thu thập, tính chính xác của dữ liệu, và khả năng mở rộng hệ thống trong việc xử lý các tập dữ liệu lớn từ mạng xã hội X(Twitter).

1.4. Đối tượng và phạm vi giới hạn

1.4.1. Đối tượng

Đối tượng nghiên cứu của đề tài là các trang cá nhân trên mạng xã hội X(Twitter), bao gồm các tài khoản cá nhân, doanh nghiệp. Các tài khoản được chọn phải có hoạt động thường xuyên, cụ thể là các tương tác như bài đăng, bình luận và lượt thích phải công khai để cung cấp dữ liệu phong phú phục vụ quá trình nghiên cứu.

1.4.2. Phạm vi giới hạn

Đề tài này tập trung vào việc thu thập dữ liệu tự động từ các trang cá nhân trên mạng xã hội X(Twitter) bằng việc áp dụng kỹ thuật Web Scraping bằng Selenium. Dữ liệu thu thập bao gồm các thông tin về bài viết, bình luận, lượt thích, và các tương tác công khai khác. Sau khi thu thập, dữ liệu sẽ được lưu trữ và xử lý trên MongoDB, để thuận tiện cho quá trình phân tích. Từ đó, ta thấy được các xu hướng, hành vi người dùng.

1.5. Phương pháp nghiên cứu

1.5.1. Phương pháp nghiên cứu sơ bộ

Chúng tôi tiến hành nghiên cứu sơ bộ về các kỹ thuật Web Scraping và khả năng thu thập dữ liệu từ mạng xã hội X(Twitter). Nghiên cứu sơ bộ sẽ giúp xác định các vấn đề quan trọng cần giải quyết, bao gồm loại dữ liệu cần thu thập, giới hạn kỹ thuật của Selenium và các vấn đề về quyền riêng tư. Điều này giúp xác định các hướng tiếp cần cần thiết cho quá trình nghiên cứu chuyên sâu tiếp theo.

1.5.2. Phương pháp nghiên cứu tài liệu

Phương pháp nghiên cứu liên quan đến việc thu thập tài liệu, báo cáo nghiên cứu liên quan đến Web Scraping, Selenium, phân tích dữ liệu và NoSQL. Việc nghiên cứu tài liệu sẽ cung cấp các kiến thức nền tảng và khinh nghiệm thực tế trong việc sử dụng các công cụ này. Thông qua việc nghiên cứu tài liệu, chúng tôi sẽ có căn cứ để lựa chọn công cụ thích hợp cho đề tài.

1.5.3. Phương pháp nghiên cứu thống kê

Phương pháp thống kê sẽ được sử dụng để phân tích dữ liệu sau khi đã thu thập được. Các phương pháp thống kê mô tả sẽ giúp tóm tắt các đặc điểm chính của dữ liệu, như tần suất xuất hiện của các tương tác, xu hướng nội dung, rút ra các kết luận về hành vi người dùng và tương tác trên mạng xã hội X(Twitter) dựa trên mẫu dữ liệu đã thu thập.

1.5.4. Phương pháp nghiên cứu thực nghiệm

Phương pháp thử nghiệm sẽ được thực hiện trong việc phát triển hệ thống Web Scraping sử dụng Selenium. Nhóm nghiên cứu sẽ thử nghiệm các tình huống trích xuất dữ liệu ngẫu nhiên từ các trang cá nhân, kiểm tra hiệu năng của hệ thống và chỉnh sửa khi cần thiết. Việc lưu trữ và truy xuất dữ liệu trong cơ sở dữ liệu NoSQL cũng sẽ được thử nghiệm nhằm đánh giá việc lưu trữ và truy xuất dữ liệu.

1.5.5. Phương pháp đánh giá

Cuối cùng, 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, cũng như hiệu quả của quá trình phân tích dữ liệu thông qua các tiêu chí về độ tin cậy và các giá trị khác của các kết quả phân tích từ hệ thống.

1.6. Những đóng góp nghiên cứu của đề tài

1.6.1. Trong lĩnh vực học thuật

Đề tài này đóng góp bằng cách mở rộng kiến thức về kỹ thuật Web Scraping và tự động thu thập dữ liệu từ các nền tảng mạng xã hội, đặc biệt là sử dụng Selenium. Điều này giúp bổ sung cho các tài liệu về phân tích dữ liệu phi cấu trúc và MongoDB trong việc lưu trữ và quản lý dữ liệu. Ngoài ra, còn góp phần vào nghiên cứu hành vi người dùng trên mạng xã hội, mở ra tiềm năng cho những nghiên cứu sâu hơn về lĩnh vực này.

1.6.2. Trong thực tiễn kinh doanh

Cung cấp một hệ thống có thể giúp doanh nghiệp thu thập dữ liệu về hành vi và xu hướng người dùng trên mạng xã hội, từ đó xây dựng các chiến lược phát triển trong tương lai. Ngoài ra, các doanh nghiệp có thể tận dụng hệ thống để hiểu rõ hơn về sự tương tác của khách hàng và cải thiện trải nghiệm người dùng.

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 nghiên cứu

Chương 4: Kết quả thực nghiệm

1.7.2. Tóm tắt từng chương

Chương 1: Tổng quan

Trình bày về lý do chọn đề tài , tính cấp thiết, mục tiêu nghiên cứu, phạm vi và giới hạn nghiên cứu, cũng như đóng góp khoa học và thực tiễn của đề tài. Từ đó, đưa ta các nền tảng lý thuyết cần thiết để giải thích về tầm quan trọng của kỹ thuật Web Scraping và phương pháp phân tích dữ liệu từ mạng xã hội.

Chương 2: Cơ sở lý thuyết

Cung cấp các kiến thức liên quan đến Web Scraping, Selenium, NoSQL và MongoDB, đồng thời trình bày các khái niệm về dữ liệu phi cấu trúc và phân tích dữ liệu từ mạng xã hội. Các lý thuyết và nghiên cứu trước đây cũng sẽ được tham khảo để làm cơ sở cho việc phát triển hệ thống.

Chương 3: Phương pháp nghiên cứu

Chương này mô tả chi tiết về các phương pháp thu thập dữ liệu và xử lý dữ liệu, bao gồm việc xây dựng hệ thống Web Scraping, lưu trữ và phân tích dữ liệu , cũng như cách thức thực nghiệm và đánh giá hiệu quả của hệ thống.

Chương 4: Kết quả thực nghiệm

Chương này tập trung vào việc phân tích kết quả thu thập được, đưa ra các nhận xét về xu hướng và hành vi người dùng trên mạng xã hội. Các kết quả phân tích sẽ được trình bày dưới dạng số liệu, và những kết luận liên quan.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Web Scraping

2.1.1. Khái niệm

Web Scraping là quá trình tự động thu thập dữ liệu từ các trang web.

Có một cách phổ biến mà chúng ta vẫn hay dùng đó là sao chép và dán (Ctrl + C, Ctrl + V), nhưng đây là cách thủ công và khá rườm rà. Chúng ta chỉ có thể sử dụng nếu dữ liệu ta thu thập được ít, nhưng nếu cần thu thập nhiều dữ liệu dung cho mục đích khai thác và phân tích, ta không thể dùng theo cách phổ thông như vậy vì rất tốn thời gian.

Nhưng ngày nay, dữ liệu đang ngày càng trở nên quan trọng và phần lớn dữ liệu này đều có sẵn trên các trang web. Vậy làm thế nào để có thể thu thập số lượng lớn dữ liệu nhanh chóng? Khái niệm Web Scraping bắt đầu xuất hiện từ đây. Quá trình này được thực hiện với sự trợ giúp của các phần mềm trích xuất dữ liệu. Và nó tự động tải và thu thập dữ liệu từ các trang web dựa trên yêu cầu của người dung.[1]

2.1.2. Cách thức hoạt động

Quá trình thu thập dữ liệu bao gồm các bước sau:

Hình 2.1 Quá trình thu thập dữ liệu

- Gửi một yêu cầu HTTP đến server.
- Nhận HTML từ server.

- Phân tích HTML và trích xuất dữ liệu.

- Lưu trữ dữ liệu.

2.1.2. Công dụng

- Nghiên cứu thị trường: Web Scraping được sử dụng để lấy phản hồi từ khách hàng về dịch vụ, hoặc về sản phẩm thể xin ý kiến từ khách hàng. Hoặc nó có thể so sánh giá cả từ các đối thủ cạnh tranh.

- Phân tích dữ liệu lớn: Web Scraping có khả năng thu thập số lượng lớn dữ liệu và nhờ vào đó, có thể dự đoán xu hướng kinh doanh trong tương lai.

- Học máy: Máy được phép học và cải thiện khả năng của nó thay vì được lập trình sẵn. Mà để làm được điều đó, cần có một lượng dữ liệu cực kỳ lớn được thu thập và trích xuất từ hàng triệu trang web khác nhau thông qua Web Scraping.

- Phân tích dữ liệu tài chính: Web Scraping được sử dụng để tổng hợp thông tin từ các thị trường chứng khoán, tình hình biến động về kinh tế thế giới hoặc cụ thể hơn là khu vực, để có thể hổ trợ phân tích và nghiên cứu đầu tư.

- Tối ưu hóa công cụ tìm kiếm (SEO): Web Scraping giúp thu thập dữ liệu theo thời gian thực từ các trang mạng xã hội, nhật kí trực tuyến. Doanh nghiệp dựa vào các dữ liệu thu thập được để tìm ra xu hướng mới để tiếp cận khách hàng, nâng cao lượt hiển thị và thứ hạng của một trang web.

2.1.3. Ưu điểm

- Tự động hóa quy trình thu thập dữ liệu: Giúp tiết kiệm thời gian và công sức so với việc thu thập dữ liệu thủ công

- Thu thập lượng lớn dữ liệu: Có khả năng thu thập dữ liệu từ nhiều trang wed khác nhau trong một khoảng thời gian ngắn

- Cập nhập liên tục: Có thể thiết lập công cụ Web Scraping để tự động cập nhập dữ liệu theo thời gian thực, giúp thong tin luôn chính xác và mới nhất.

- Tăng hiệu quả phân tích: Cung cấp lượng thông tin luôn chính lớn cho các dự án phân tích , nghiên cứu thị trường và dự đoán xu hướng.

2.1.4. Nhược điểm

- Phụ thuộc vào cấu trúc HTML: Nếu trang thay đổi cấu trúc HTML, mã scraping cần được cập nhập để phù hợp với cấu trúc mới.

- Tính hợp pháp của Web Scraping: Đây không phải là vấn đề bất hợp pháp, nhưng nếu ta lạm dụng, có thể làm giảm hiệu suất máy chủ của trang web. Còn nhiều yếu tố liên quan đến đạo đức hoặc thậm chí là pháp luật. Một số quy định về bảo vệ dữ liệu (GDPR) ở châu Âu về thu thập dữ liệu cá nhân. Hoặc nhiều trang web có ToS (điều khoản dịch vụ) dẫn tới vi phạm pháp lý dù thực tế nó không phải là hành động bất hợp pháp.**[2]**

2.2. Python

2.2.1. Khái niệm

Python là một ngôn ngữ lập trình bậc cao, được thông dịch. Được Guido Van Rossum tạo ra và phát hành vào năm 1991. Ngôn ngữ này ngày càng trở nên phổ biến và phát triển cực kỳ mạnh mẽ bởi cấu trúc cú pháp đơn giản, dễ đọc. Và đặc biệt được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

2.2.2. Tính năng và lợi thế của python

- Đơn giản và dễ tiếp cận: Đây là một ngôn ngữ có cú pháp rất đơn giản, dễ học và cũng dễ đọc. Là ngôn ngữ phù hợp cho người mới học lập trình.

- Được thông dịch (interpreted language): Python là ngôn ngữ được thông dịch, nghĩa là mã được thực thi từng dòng một mà không cần biên dịch thành mã máy. Điều này giúp dễ dàng kiểm tra và sữa lỗi, không quá phức tạp. [3]

- Sự linh hoạt: Bạn không cần khai báo kiểu dữ liệu của biến một cách rõ ràng. Python có thể tự động suy ra kiểu dữ liệu dựa trên giá trị được gán cho biến. Kiểu dữ liệu của biến được áp dụng khi chạy chương trình. Giúp làm tăng tốc độ và giúp ngăn ngừa lỗi làm cho mã mạnh mẽ hơn.

- Ngôn ngữ đa nền tảng: Python có thể chạy trên nhiều hệ điều hành khác nhau mà không cần thay đổi. Giúp dễ dàng phát triển ứng dụng Python trên nhiều nền tảng khác nhau mà không gặp bất kì rắc rối nào.

- Đa dạng thư viện: Gồm các thư viện cho khoa học dữ liệu (Numpy, Pandas), học máy (Scikit-learn), web (Django, Flask) cung cấp các công cụ và mô-đun cho nhiều tác vụ khác nhau giúp xây dựng các ứng dụng mà không cần viết lại mọi thứ từ đầu.

- Đa năng: Đây là ngôn ngữ có thể sử dụng cho nhiều mục đích khác nhau ví dụ như phát triển web, trí tuệ nhân tạo, trò chơi, khoa học dữ liệu, học máy, học sâu,...

- Hệ sinh thái: Python có một cộng đồng lớn cực kỳ đông đảo và năng động, nhiều diễn đàn hỗ trợ cho người mới là điều khiến ngôn ngữ này phát triển rất nhanh.

2.2.3. Nhược điểm

- Hiệu suất: Tốc độ chạy chương trình chậm hơn so với những ngôn ngữ phổ biến khác.

- Global Interpreter Lock (GIL): là một cơ chế trong Python ngăn nhiều luồng thực thi mã Python cùng một lúc. Điều này hạn chế tính song song và đồng thời của một số ứng dụng.

- Tiêu thụ bộ nhớ: Python có thể tiêu tốn rất nhiều bộ nhớ, đặc biệt khi làm việc với các tập dữ liệu lớn hoặc các thuật toán phức tạp. [4]

2.2.4. Ứng dụng

- Web Scraping: Python được sử dụng để thu thập dữ liệu từ các trang web dựa vào phân tích cú pháp HTML.

- Phát triển web: Các framework phổ biến như Django và Flask để phát triển các ứng dụng web.

- Tính toán khoa học (khoa học dữ liệu và máy học): SciPy là một thư viện mã nguồn mở dùng để tính toán khoa học. Numpy là thư viện toán học chuyên xử lý dữ liệu ma trận và mảng. Pandas là thư viện phân tích và mô hình hóa dữ liệu.

- Giao diện đồ họa người dùng (GUI): Python có thể dùng để phát triển giao diện thông qua sự hỗ trợ của các thư viện như Tkinter, PyQt, Kivy, wxWidgets.

- Quản lý cơ sở dữ liệu: Python cho phép truy cập các phần mềm quản lý cơ sở dữ liệu như MySQL, MongoDB, PostgreSQL.

- Giáo dục: Các câu lệnh khá đơn giản nên rất thích hợp để trở thành một ngôn ngữ lập trình cho người mới bắt đầu học. Python ngày càng được nhiều giáo viên giảng dạy ở các cấp bậc trung học, đại học.[4], [5]

2.3. Selenium

2.3.1. Khái niệm

Selenium là một automation testing framework miễn phí (mã nguồn mở). Nó được sử dụng để kiểm thử các ứng dụng web trên các trình duyệt (chrome, firefox, ms edge, …) và nền tảng khác nhau (Windows, Mac, Linux, …)**. [6]** Cho phép người dùng mô phỏng các hoạt động của người dùng trên trình duyệt như nhấp chuột, điền biểu mẫu và điều hướng giữa các trang web.

2.3.2. Các thành phần của Selenium

Selenium IDE: Là một công cụ ghi lại các thao tác trên trình duyệt của người dùng và cho phép tạo các kịch bản kiểm thử mà không cần lập trình.

Selenium WebDriver: Là thành phần chính, cho phép điều khiển và tương tác trực tiếp với các thành phần của trình duyệt web một cách tự động.

Selenium Grid: Cho phép chạy các kịch bản kiểm thử song song trên nhiều máy tính và trình duyệt cùng một lúc, giúp tiết kiệm thời gian và tăng hiệu suất.

2.3.3. Ứng dụng

Kiểm thử ứng dụng web: là công cụ phổ biến nhất để thực hiện kiểm thử tự động cho các ứng dụng web , nhằm đảm bảo chúng hoạt động đúng trên các trình duyệt và hệ điều hành khác nhau.

Tự động hóa tác vụ web: Tự động hóa các tác vụ lặp đi lặp lại trên web, như thu thập dữ liệu, gửi email tự động,…

Phát triển phần mềm: Sử dụng Selenium để kiểm tra các tính năng mới của ứng dụng trước khi phát hành, giúp phát hiện lỗi sớm trong quá trình phát triển.

2.3.4. Các công trình công nghệ áp dụng Selenium

"Web Scraping Using Python and Selenium to Build E-commerce Price Comparison System"

Tác giả: J. K. Sharma, A. Kumar

Tạp chí: International Journal of Innovative Research in Technology (2020)

Mô tả: Bài báo mô tả cách Selenium được sử dụng để thu thập dữ liệu từ các trang web thương mại điện tử nhằm xây dựng hệ thống so sánh giá trực tuyến.

2.4. Giới thiệu về NoSQL

2.4.1. Khái niệm về NoSQL

NoSQL (Not Only SQL) là một loại cơ sở dữ liệu không quan hệ , được thiết kế để xử lý và lưu trữ dữ liệu lớn theo cách linh hoạt hơn so với cơ sở dữ liệu quan hệ truyền thống (SQL) và được ứng dụng rộng rãi. NoSQL cho phép lưu trữ dữ liệu có cấu trúc , bán cấu trúc, phi cấu trúc mà không cần một schema cố định.

2.4.2. Tính năng nổi bật của NoSQL

Xử lý dữ liệu lớn: được thiết kế để xử lý các tập dữ liệu khổng lồ.

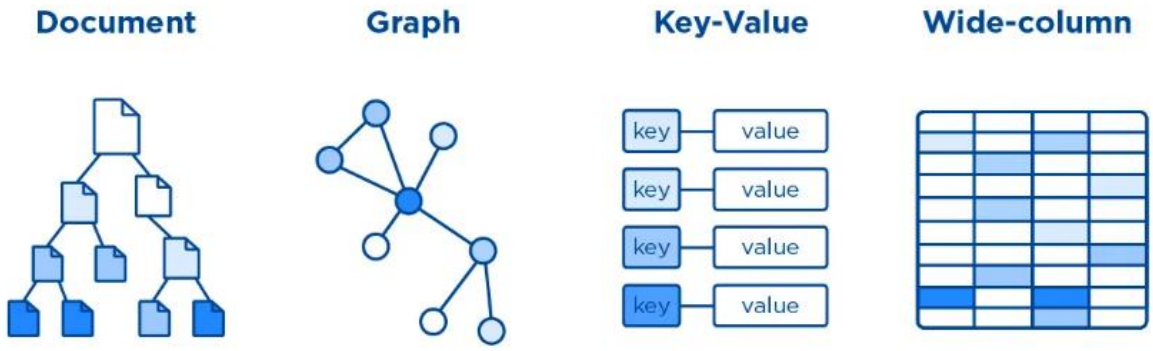
Khả năng mở rộng và phân phối: Hệ thống NoSQL có thể mở rộng dễ dàng, có khả năng chạy trên hàng nghìn máy chủ.

Truy vấn nhanh: được tối ưu hóa để trả về kết quả truy vấn một cách nhanh chóng, do không bị giới hạn bởi các mối quan hệ phức tạp như trong SQL.

Không cần cấu trúc(Schema-less): NoSQL cho phép lưu trữ dữ kiệu mà không yêu cầu cấu trúc cố định, mang lại sự linh hoạt trong việc quản lý dữ liệu.

Phát triển mã nguồn mở: nhiều giải pháp NoSQL được phát triển dưới dạng mã nguồn mở, cho phép cộng đồng tham gia vào việc cải tiến và phát triển công nghệ.

2.4.3. Các loại dữ liệu NoSQL



Hình 2.2 Phân loại cơ sở dữ liệu NoSQL[7]

Cơ sở dữ liệu document: Lưu trữ dữ liệu dưới dạng tài liệu (documents), thường là JSON hoặc XML, cho phép linh hoạt trong cấu trúc dữ liệu. Mỗi tài liệu có thể chứa các cặp khóa-giá trị và hỗ trợ cấu trúc lồng nhau.

Cơ sở dữ liệu Wide-column: Lưu trữ dữ liệu theo cột , tối ưu hóa các truy vấn đề có tính toán trên các cột cụ thể.

Cơ sở dữ liệu Key-Value: Lưu trữ dưới dạng cặp khóa giá trị, mỗi khóa là duy nhất và phù hợp cho các ứng dụng cần truy cập nhanh.

Cơ sở dữ liệu Graph: Dữ liệu được lưu trữ dưới dạng các nút và cạnh, tập trung vào việc lưu trữ và truy vấn dữ liệu lên quan đến mối quan hệ, lý tưởng cho các ứng dụng yêu cầu phân tích mối quan hệ phức tạp.

2.4.4.Ứng dụng của NoSQL

Mạng xã hội: Lưu trữ các bài đăng, bình luận và tương tác của người dùng trên các nền tảng như Facebook, Instagram.

Thương mại điện tử: Quản lý dữ liệu khách hàng, đơn hàng và tương tác người dùng trên các trang web như Amazon, eBay.

Phân tích dữ liệu lớn (Big Data): Thu thập và phân tích dữ liệu phi cấu trúc từ các nguồn khác nhau nhằm hỗ trợ các quyết định kinh doanh.

Phân tích thời gian thực: Theo dõi và phân tích dữ liệu theo thời gian thực từ cảm biến IoT hoặc ứng dụng trực tuyến.

2.5. MongoDB

2.5.1.Khái niệm

MongoDB là một hệ quản trị cơ sở dữ liệu (DBMS) thuộc loại NoSQL, được thiết kế để lưu trữ và truy xuất dữ liệu theo cách linh hoạt và không yêu cầu sự chuẩn bị cấu trúc cố định trước. Nó lưu trữ dữ liệu dưới dạng tài liệu (document) JSON**[8].** Là môt hệ quản trị cơ sở dữ liệu phổ biến cho các ứng dụng web và các dự án phát triển nhanh, đặc biệt là khi có sự thay đổi thường xuyên trong cấu trúc dữ liệu.

2.5.2. Đặc điểm quan trọng của MongoDB

Các ad hoc query: MongoDB hỗ trợ các truy vấn linh hoạt, cho phép tìm kiếm theo trường dữ liệu, thực hiện các truy vấn thông thường, tìm kiếm theo biểu thức chính quy (regular expression), và truy vấn theo khoảng giá trị.

Indexing: Bất kỳ trường nào trong tài liệu BSON cũng có thể được tạo chỉ mục, giúp tăng tốc quá trình tìm kiếm và truy xuất dữ liệu.

Replication (Nhân bản): MongoDB hỗ trợ chức năng nhân bản, tạo ra một bản sao đồng nhất với phiên bản đang hoạt động. Điều này giúp bảo vệ dữ liệu khỏi mất mát và đảm bảo tính toàn vẹn của cơ sở dữ liệu trong trường hợp sự cố.

Aggregation: Các phép toán tập hợp trong MongoDB xử lý và trả về kết quả được tính toán. Chúng có thể nhóm các giá trị từ nhiều tài liệu lại với nhau và thực hiện nhiều phép toán đa dạng để trả về kết quả duy nhất. Điều này giống với GROUP BY và các hàm tổng hợp trong SQL.

Lưu trữ file: MongoDB có thể được sử dụng như một hệ thống lưu trữ file, tận dụng các chức năng của nó và hoạt động như một cách phân phối thông qua sharding, giúp quản lý và truy xuất dữ liệu lớn.

2.5.3. Ưu điểm

Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.

Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

2.5.4. Nhược điểm

Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.

Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.

Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

2.5.5. Ứng dụng của MongoDB

Phát triển ứng dụng web: Nhiều công ty như Facebook, eBay, và Adobe sử dụng MongoDB để lưu trữ và quản lý lượng lớn dữ liệu người dùng.

Khoa học dữ liệu: MongoDB hỗ trợ phân tích dữ liệu lớn và có thể được sử dụng trong các ứng dụng học máy để lưu trữ và xử lý khối lượng lớn thông tin.

Hệ thống quản lý nội dung: MongoDB rất phù hợp cho việc quản lý nội dung động như blog hoặc bài viết trên trang web.

Ứng dụng IoT: Với khả năng xử lý dữ liệu thời gian thực, MongoDB là lựa chọn tốt cho các ứng dụng IoT cần lưu trữ và phân tích dữ liệu từ cảm biến.

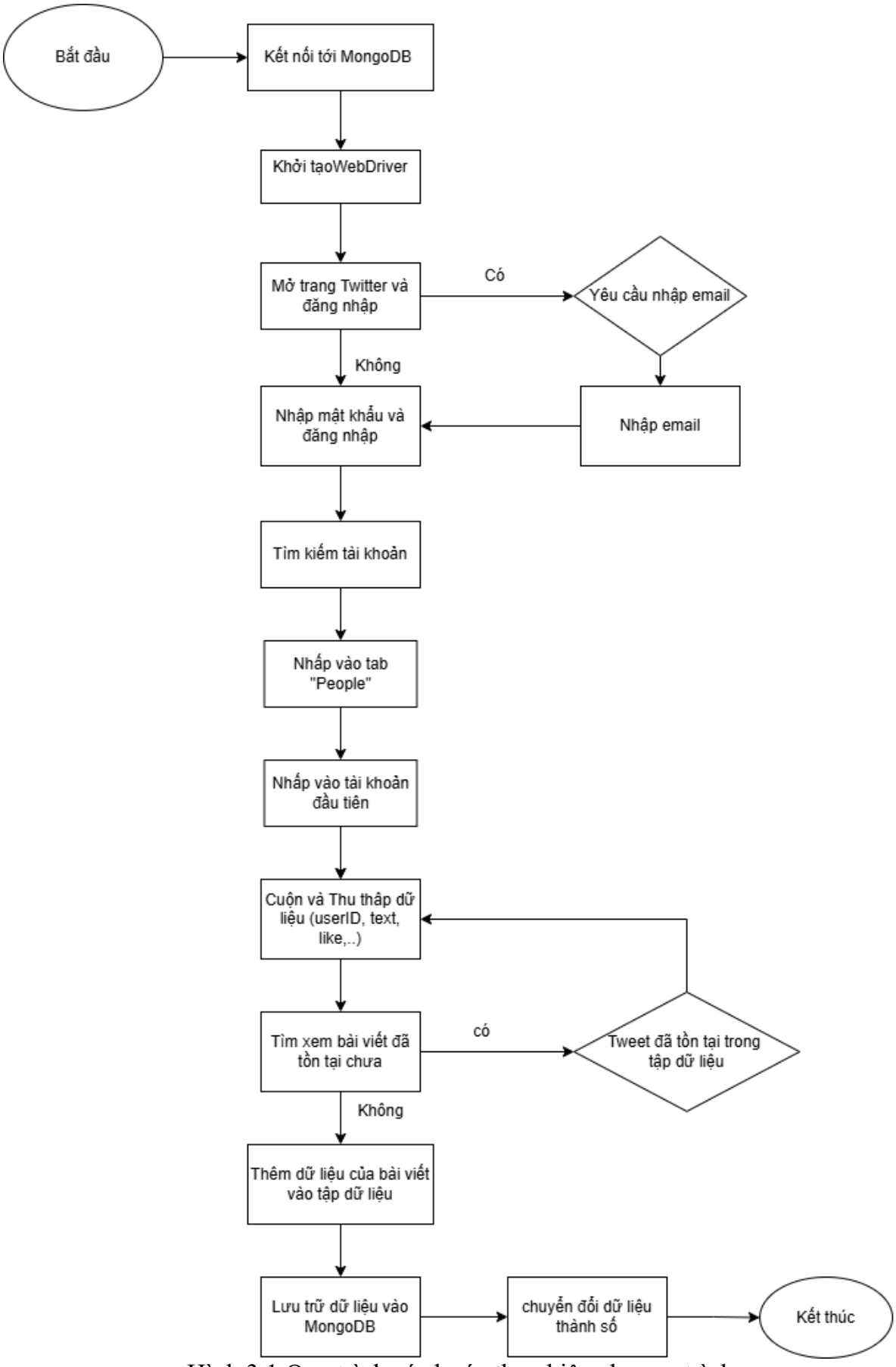
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

3.1. Trình bày công việc thực nghiệm

3.1.1. Mục tiêu

Truy cập trang cá nhân của một người nổi tiếng (có sức ảnh hưởng trên mạng xã hội) để thu thập dữ liệu các bài đăng, lượt tương tác, lượt xem,… Dựa vào đó để xác định được họ đăng những cái gì? Vấn đề họ thường bàn luận thuộc chủ đề gì? Và mọi người có quan tâm đến vấn đề đó hay không?

3.1.2. Quy trình thực nghiệm



Hình 3.1 Quy trình các bước thực hiện chương trình

3.2. Kết quả thực nghiệm

3.2.1. Chi tiết quá trình thu thập dữ liệu

- **Bước 1**: Kết nối đến cơ sở dữ liệu MongoDB

```
# #kết nối
client = MongoClient("mongodb://localhost:27017/")
client.drop_database('demo')
db = client['demo']  # chon csdl Facebookdata1
collection = db['tweets']
```

Hình 3.2 Câu lệnh kết nối cơ sở dữ liệu MongoDB

- **Bước 2**: Đường dẫn đến geckodriver

```
# Đường dẫn đến file thực thi geckodriver
gecko_path = r"D:/DoAnNhom10d/Nhom-10d/pythonProject/geckodriver.exe"
ser = Service(gecko_path) # Khởi tạo đối tượng dịch vụ với geckodriver

# Tạo tùy chọn
options = webdriver.firefox.options.Options()
options.binary_location ="C:/Program Files/Mozilla Firefox/firefox.exe"
# Thiết lập firefox chỉ hiện thị giao diện
options.headless = False
```

Hình 3.3 Câu lệnh kết nối trình duyệt FireFox và thiết lập tùy chọn cho trình duyệt

-- Khởi tạo đối tượng "Service" giúp tương tác với driver trong quá trình tự động hóa điều khiển trình duyệt.

-- Cấu hình trình duyệt: Khởi tạo đối tượng Options để thiết lập cấu hình của trình duyệt trước khi khởi động:

+ options.binary_location: Chỉ định vị trí của file thực thi chương trình.

+ options.headless: Chạy chương trình ở chế độ hiển thị giao diện (False) giúp cho phép bạn quan sát quá trình hoạt động của chương trình, đặt (True) thì trình duyệt chạy ngầm giúp bạn tiết kiệm tài nguyên.

- **Bước 3**: Khởi tạo trình duyệt theo đối tượng cài đặt từ trước và truy cập trang:

```
# Khởi tạo driver
driver = webdriver.Firefox(options=options, service=ser)
driver.get("https://x.com/i/flow/login")
```

Hình 3.4 Khởi tạo trình duyệt theo đối tượng được cài và truy cập trang web

- **Bước 4**: Đăng nhập tài khoản:

```
#Đăng nhập
username = driver.find_element(By.XPATH, value: "//input[@name='text']")
username.send_keys("hnhp113")
time.sleep(1)
name = driver.find_element(By.XPATH, value: "//span[contains(text(),'Tiếp theo')]")
name.click()

time.sleep(2)
#Xác thực thông tin
try:
    email = driver.find_element(By.XPATH, value: "//input[@name='text']")
    email.send_keys("hnhp113114115@gmail.com")
    tt = driver.find_element(By.XPATH, value: "//span[contains(text(),'Tiếp theo')]")
    tt.click()
except:
    pass
time.sleep(2)
#Mật khẩu
password = driver.find_element(By.XPATH, value: "//input[@name='password']")
password.send_keys("phatho0317")
pw = driver.find_element(By.XPATH, value: "//span[contains(text(),'Đăng nhập')]")
pw.click()
```

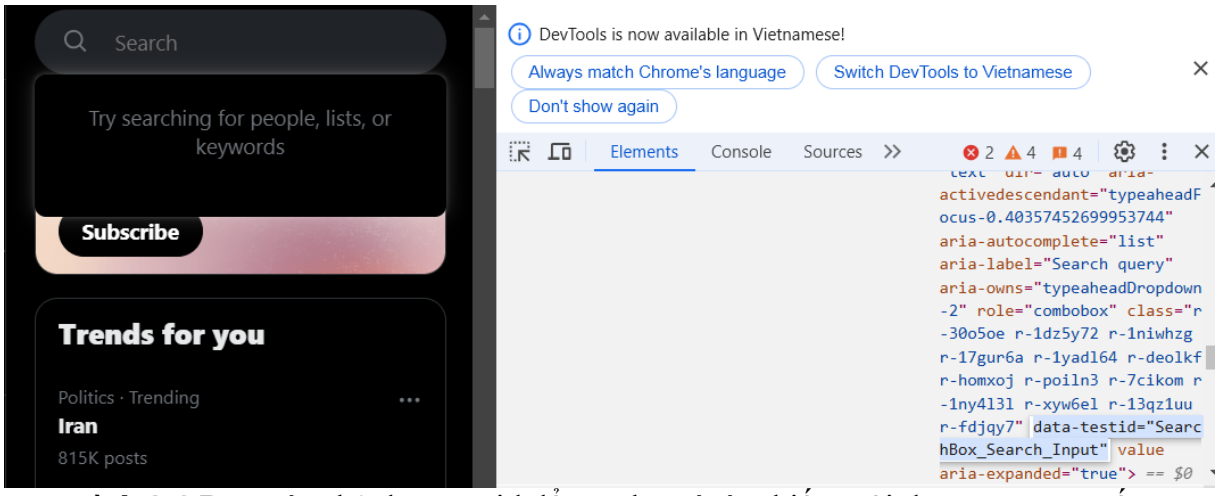
Hình 3.5 Đăng nhập trang X

-- Tài khoản: Sử dụng thẻ XPATH để tìm và gửi thông tin đăng nhập.

-- Xác nhận thông tin: Thỉnh thoảng, trang này sẽ hỏi xem gmail hoặc số điện thoại của bạn là gì để xác thực xem đó có phải là bạn không. Nhưng rất ít gặp, nếu không có thì bỏ qua.

-- Mật khẩu: Tương tự với bước nhập tài khoản.

- **Bước 5**: Tìm thanh tìm kiếm để nhập trang cá nhân bạn muốn truy cập:



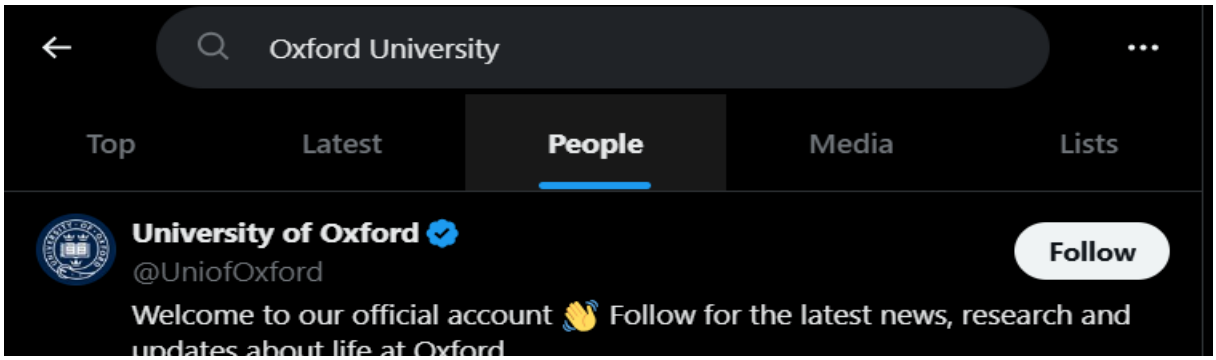
Hình 3.6 Dựa vào thẻ data-testid để gõ chữ và tìm kiếm nội dung mong muốn

```
#Tìm trang cá nhân bạn muốn quét dữ liệu
search = driver.find_element(By.XPATH, value: "//input[@data-testid='SearchBox_Search_Input']")
idol = "Oxford University"
search.send_keys(idol)
search.send_keys(Keys.ENTER)
time.sleep(5)
```

Hình 3.7 Câu lệnh để nhập và tìm kiếm

Hình 3.8 Nội dung tìm kiếm được

- **Bước 6**: Nhấn vào tab People để tìm người:



Hình 3.9 Qua tab People và click vào trang bạn chọn

```
#Nhấp vao người bạn muốn tìm
name_idol = driver.find_element(By.XPATH, value: "//*[@id='react-root']/div/div/div[2]/main/div/div/div/div[1]/div/div[3]
name_idol.click()
time.sleep(2)
```

Hình 3.10 Câu lệnh truy cập trang cá nhân tìm được

-- Dựa vào thẻ XPATH để tìm và nhấn chuột để truy cập trang cá nhân.

- **Bước 7**: Hàm cào dữ liệu bài viết

```
def scrape_tweets(driver, max_tweets = 660):  1 usage   Levan-24714 +1 *
    #Luu du luu kiem tra bai viet trung lap:
    global document
    data_set = set()
    count = 0

    userIDs = []
    timePosts = []
    tweetTexts = []  # = Post status
    likes = []
    replys = []
    reposts = []
    views = []
    tweetIMG=[]
```

Hình 3.11 Tạo biến lưu trữ

```python
def scrape_tweets(driver, max_tweets = 660):  1 usage  Levan-24714 +1 *
    while count < max_tweets:
        articles = driver.find_elements(By.XPATH, "//article[@data-testid='tweet']")
        for article in articles:
            try:
                userID = article.find_element(By.XPATH, ".//div[@data-testid='User-Name']").text
            except:
                userID = ''
            try:
                timePost = driver.find_element(By.XPATH, ".//time").get_attribute("datetime")
            except:
                timePost = ''
            try:
                tweetText = driver.find_element(By.XPATH, ".//div[@data-testid='tweetText']").text
            except:
                tweetText = ''
            try:
                like_count = driver.find_element(By.XPATH, ".//button[contains(@aria-label,'Likes')]")
                like = like_count.get_attribute('aria-label').split(' ')[0]
            except:
                like = ''
            try:
                reply_count = driver.find_element(By.XPATH, ".//button[contains(@aria-label,'Replies')]")
                reply = reply_count.get_attribute('aria-label').split(' ')[0]
            except:
                reply = ''
            try:
                # resport = driver.find_element(By.XPATH, ".//button[@data-testid='retweet']").text
                repost_count = driver.find_element(By.XPATH, ".//button[contains(@aria-label,'reposts')]")
                repost = repost_count.get_attribute('aria-label').split(' ')[0]
            except:
                repost = ''
```

Hình 3.12 Tìm dữ liệu dựa vào thẻ html và thêm dữ liệu đó vào biến

```python
            try:
                view = driver.find_element(By.XPATH, ".//a[contains(@aria-label,'views')]")
                views_count = view.get_attribute('aria-label').split(' ')[0]
            except:
                views_count = ''
            try:
                images = article.find_elements(By.XPATH, ".//img[contains(@src, 'https://pbs.twimg.com') and not (contains(@src, 'profile_images'))]")
                tweetIMGs = [img.get_attribute('src') for img in images]
            except:
                tweetIMGs = ''
```

Hình 3.13 Tìm lượt xem và hình ảnh

```python
def scrape_tweets(driver, max_tweets = 660):  1 usage  Levan-24714 +1 *
        # Tạo document dữ liệu để lưu trữ trong MongoDB
        document = {
            "userID": userID,
            "timePost": timePost,
            "tweetText": tweetText,
            "like": like,
            "reply": reply,
            "repost": repost,
            "views": views_count,
            "tweetIMG": tweetIMGs

        }

        #Kiểm tra xem tweet có trùng lặp không
        if tweetText not in tweetTexts:
            data_set.add(tweetText)
            userIDs.append(userID)
            timePosts.append(timePost)
            tweetTexts.append(tweetText)
            likes.append(like)
            replys.append(reply)
            reposts.append(repost)
            views.append(views_count)
            tweetIMG.append(tweetIMGs)
            collection.insert_one(document) #Chèn data Mongo
            count += 1

            if count > max_tweets:
                break
```

Hình 3.14 Tạo biến document chứa tất cả các dữ liệu đã được thu thập, kiểm tra xem dữ liệu đó đã tồn tại chưa

-- Biến document chứa tất cả dữ liệu được thu thập ở một bài post.
-- Kiểm tra xem tweet có trùng lập không, nếu không thì thêm vào 'data_set.add(tweetText)', và thêm các dữ liệu khác vào.
-- collection.insert_one(document): Ta có biến document chứa tất cả dữ liệu, đây là câu lệnh thêm document vào file MongoDB.
-- Lệnh if phía dưới để có thể dừng chương trình nếu số dữ liệu bạn cần thu thập đã đủ.

```
def scrape_tweets(driver, max_tweets = 660):  1 usage  Levan-24714 +1 *
        #Cuộn chậm
        driver.execute_script("window.scrollBy(0,600);")
        time.sleep(3)
        #Lấy thêm tweets mới sau khi cuộn
        print(len(set(tweetTexts)))
    print(f"Dữ liệu đã được lưu vào MongoDB. Đã cào được {count} bài")
scrape_tweets(driver)
driver.quit()
```

Hình 3.15 Cuộn trang, gọi hàm để hàm chạy và đóng trình duyệt sau khi chạy xong

- **Bước 8:** Hàm chuyển đổ kiểu dữ liệu từ string sang số trong MongoDB

```
def convert_numeric_fields(collection):  1 usage  Phat *
    for document in collection.find():
        update_fields = dict()

        for field, value in document.items():
            # Kiểm tra nếu giá trị là chuỗi và có thể chuyển đổi thành số
            if isinstance(value, str):
                try:
                    num_value = int(value) if value.isdigit() else float(value)
                    update_fields[field] = num_value #Cap nhat gia tri
                except:
                    continue
        # Nếu có bất kỳ trường nào cần cập nhật, thực hiện cập nhật
        if update_fields:
            collection.update_one({'_id': document['_id']}, {'$set': update_fields})

# Kết nối lại với cơ sở dữ liệu MongoDB và gọi hàm chuyển đổi
convert_numeric_fields(collection)
```

Hình 3.16 Hàm chuyển đổi kiểu dữ liệu

-- for document in collection.find(): duyệt qua từng document thuộc collection.

+ update_fields = {} là từ điển lưu các trường và giá trị cần cập nhật. Nếu có giá trị cần cập nhật, từ điển này sẽ lưu giá trị mới.

-- for field, value in document.item(): Duyệt tất cả trường (field) và giá trị (value).

+ If isinstance(value, str): Kiểm tra giá trị này có phải là chuỗi không.

Nếu phải, chuyển đổi thành số nguyên trước nếu giá trị của nó là số (value.isdigit()), hoặc chuyển thành số thực float.

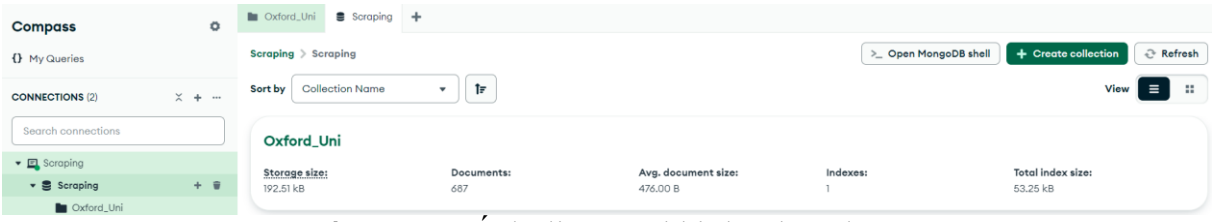
update_fields[field] = num_value: Cập nhật giá trị mới

Except: Bỏ qua các giá trị, tiếp tục chạy chương trình nếu nó không phải là ký tự số.

-- if update_fields: collection.update_one: Thực hiện cập nhật các dữ liệu nếu có sự thay đổi.

3.2.2. Kết quả thực nghiệm

Thu thập được file dữ liệu gồm 687 documents khác nhau



Hình 3.17 Số dữ liệu sau khi thu thập được

Hình 3.18 Các cột thu thập được ở một vài document

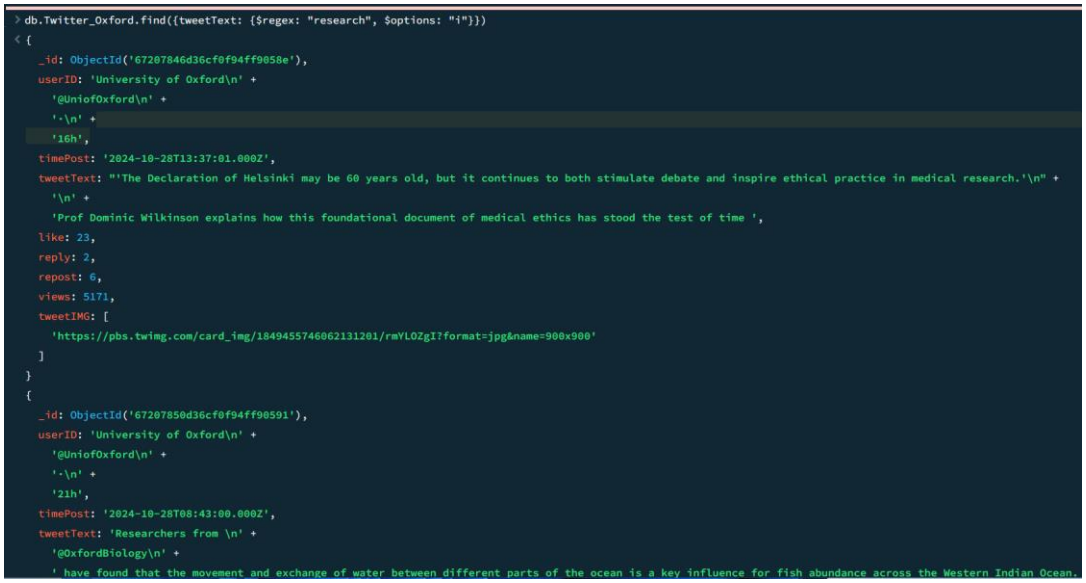
Thu thập được các documents gồm các cột dữ liệu: ‘userID’, ‘timePost’, ‘tweetText’ (bài viết), ‘like’, ‘reply’, ‘repost’ (bài đăng lại), ‘view’, ‘tweetIMG’ (ảnh trong bài viết).

Mô tả dữ liệu:

Tên biến	Mô tả	Kiểu dữ liệu
userID	Tên tài khoản người dùng	String
timePost	Thời gian đăng bài	String
tweetText	Nội dung bài viết	String
like	Số lượng thích bài viết	Int
reply	Số lượng bình luận của bài viết	Int
report	Số lượng bài đăng lại	Int
view	Số lượng xem bài viết	Int
tweetIMG	Đường dẫn URL của hình ảnh bài viết	Array

3.3 Phân tích dữ liệu sản phẩm

-- Tìm tất cả bài đăng có chứa từ khóa tweetTexts chứa "research"



Hình 3.19 Tìm tất cả bài đăng có chứa từ khóa “research”

-- Lấy bài đăng trong thời gian cụ thể timePosts là "2024-09-03"

Hình 3.20 Lấy bài đăng với thời gian là 2024-09-03

-- Tìm tất cả bài đăng không có ảnh đính kèm

```
db.Twitter_Oxford.find({ tweetIMG: { $eq: [] } })
{
  _id: ObjectId('67207869d36cf0f94ff90597'),
  userID: 'Exeter College, Oxford\n' +
    '@ExeterCollegeOx\n' +
    '·\n' +
    'Oct 24',
  timePost: '2024-10-24T14:01:14.000Z',
  tweetText: ' How does it feel to be at Exeter College? \n' +
    '\n' +
    'Hear what some of our newly matriculated students think \n' +
    '\n' +
    '#ExeterCollegeOxford #OxfordUniversity',
  like: 28,
  reply: 3,
  repost: 3,
  views: 5995,
  tweetIMG: []
}
{
  _id: ObjectId('6720787ad36cf0f94ff9059c'),
  userID: 'Oxford Biology\n' +
    '@OxfordBiology\n' +
    '·\n' +
    'Oct 23',
  timePost: '2024-10-23T16:15:48.000Z',
```

Hình 3.21 Tìm tất cả bài đăng không có ảnh đính kèm

-- Tổng lượt thích của tất cả bài đăng

```
db.Twitter_Oxford.aggregate([{$group : {_id :null, totalLikes : {$sum: "$like"}}}])
{
  _id: null,
  totalLikes: 160947
}
```

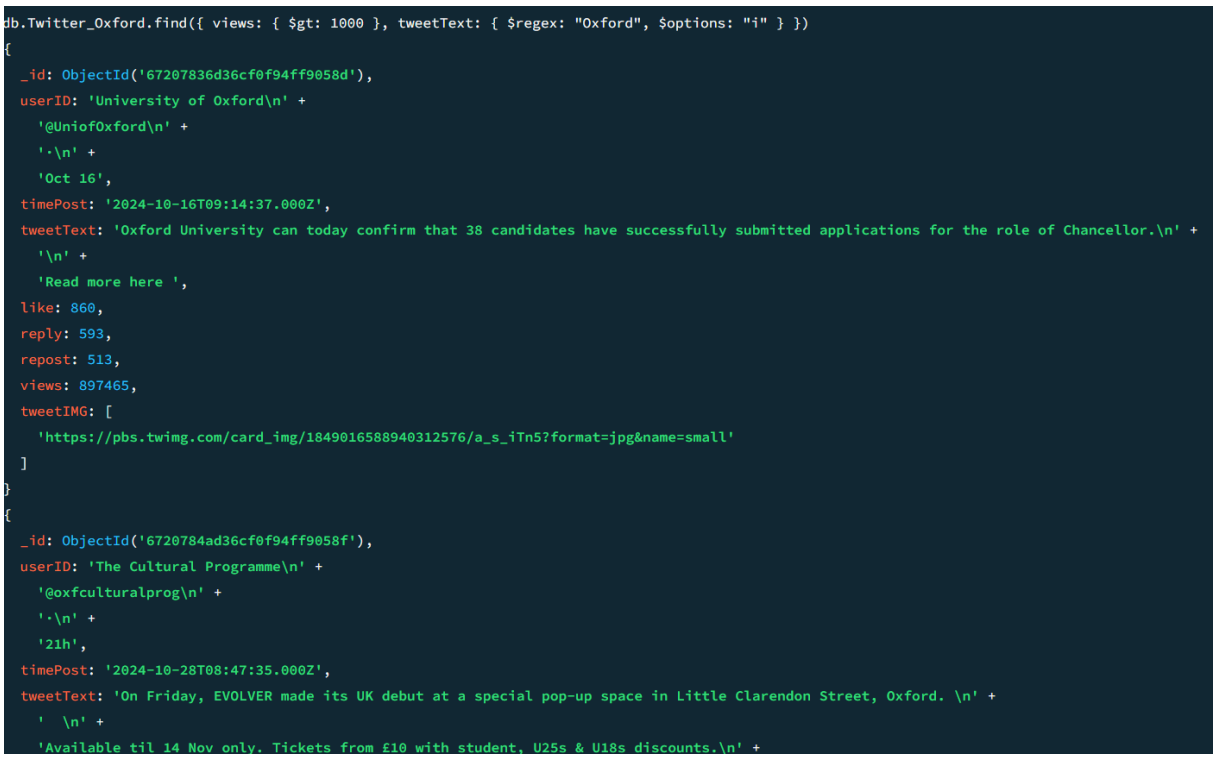
Hình 3.22 Tính tổng lượt thích tất cả bài đăng

-- Truy vấn tất cả bài đăng được xắp xếp theo lượt thích giảm dần

```
db.Twitter_Oxford.find().sort({like :-1})
{
  _id: ObjectId('67207d58d36cf0f94ff906e4'),
  userID: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    'Jun 16',
  timePost: '2024-06-15T18:35:41.000Z',
  tweetText: 'Eid Mubarak to everyone celebrating around the world \n' +
    '\n' +
    '#EidAlAdha',
  like: 7190,
  reply: 0,
  repost: 665,
  views: 173869,
  tweetIMG: [
    'https://pbs.twimg.com/media/GQIofe1WgAAeNGe?format=jpg&name=small'
  ]
}
{
  _id: ObjectId('67207f86d36cf0f94ff90776'),
  userID: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    'Apr 10',
  timePost: '2024-04-09T19:02:06.000Z',
  tweetText: 'Eid Mubarak \n' +
    '\n' +
```

Hình 3.23 Tìm bài đăng sắp xếp theo thứ tự giảm dần

-- Truy vấn bài đăng có hơn 1000 lượt xem và từ khóa "Oxford" trong nội dung



```
db.Twitter_Oxford.find({ views: { $gt: 1000 }, tweetText: { $regex: "Oxford", $options: "i" } })
{
  _id: ObjectId('67207836d36cf0f94ff9058d'),
  userID: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    'Oct 16',
  timePost: '2024-10-16T09:14:37.000Z',
  tweetText: 'Oxford University can today confirm that 38 candidates have successfully submitted applications for the role of Chancellor.\n' +
    '\n' +
    'Read more here ',
  like: 860,
  reply: 593,
  repost: 513,
  views: 897465,
  tweetIMG: [
    'https://pbs.twimg.com/card_img/1849016588940312576/a_s_iTn5?format=jpg&name=small'
  ]
}
{
  _id: ObjectId('6720784ad36cf0f94ff9058f'),
  userID: 'The Cultural Programme\n' +
    '@oxfculturalprog\n' +
    '·\n' +
    '21h',
  timePost: '2024-10-28T08:47:35.000Z',
  tweetText: 'On Friday, EVOLVER made its UK debut at a special pop-up space in Little Clarendon Street, Oxford. \n' +
    '  \n' +
    'Available til 14 Nov only. Tickets from £10 with student, U25s & U18s discounts.\n' +
```

Hình 3.24 Tìm bài đăng có hơn 1000 lượt xem và từ khóa "Oxford" trong nội dung

-- Lấy bài đăng có lượt thích và lượt chia sẻ thấp (nhỏ hơn 10) , có số lượt xem cao (lớn hơn 1000)

```
db.Twitter_Oxford.find({ like: { $lt: 50 }, repost: { $lt: 10 }, views: { $gt: 1000 } })
{
  _id: ObjectId('67207846d36cf0f94ff9058e'),
  userID: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    '16h',
  timePost: '2024-10-28T13:37:01.000Z',
  tweetText: "'The Declaration of Helsinki may be 60 years old, but it continues to both stimulate debate and inspire ethical practice in medical researc
    '\n' +
    'Prof Dominic Wilkinson explains how this foundational document of medical ethics has stood the test of time ',
  like: 23,
  reply: 2,
  repost: 6,
  views: 5171,
  tweetIMG: [
    'https://pbs.twimg.com/card_img/1849455746062131201/rmYLOZgI?format=jpg&name=900x900'
  ]
}
{
  _id: ObjectId('6720784ad36cf0f94ff9058f'),
  userID: 'The Cultural Programme\n' +
    '@oxfculturalprog\n' +
    '·\n' +
    '21h',
  timePost: '2024-10-28T08:47:35.000Z',
  tweetText: 'On Friday, EVOLVER made its UK debut at a special pop-up space in Little Clarendon Street, Oxford. \n' +
    '  \n' +
```

Hình 3.25 Tìm bài đăng có lượt thích nhỏ hơn 50, lượt chia sẻ nhỏ hơn 10 và lượt xem lớn hơn 1000

-- Tính trung bình số lượt thích cho các bài đăng của từng người dùng

```
db.Twitter_Oxford.aggregate([
    { $group: { _id: "$userID", avgLikes: { $avg: "$like" } } }
])
{
  _id: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    'Apr 18',
  avgLikes: 201
}
{
  _id: 'Oxford Anthropology\n' +
    '@oxford_anthro\n' +
    '·\n' +
    'Apr 10',
  avgLikes: 91
}
{
  _id: 'Oxford Giving\n' +
    '@OxfordGiving\n' +
    '·\n' +
    'Sep 26',
  avgLikes: 12
}
```

Hình 3.26 Tính trung bình số lượt thích theo userID

-- Lấy danh sách tất cả các bài đăng có nội dung chứa từ "Oxford" và có hình ảnh đi kèm

```
> db.Twitter_Oxford.find({
    tweetText: { $regex: "Oxford", $options: "i" },
    tweetIMG: { $ne: [] }
})
< {
  _id: ObjectId('67207836d36cf0f94ff9058d'),
  userID: 'University of Oxford\n' +
    '@UniofOxford\n' +
    '·\n' +
    'Oct 16',
  timePost: '2024-10-16T09:14:37.000Z',
  tweetText: 'Oxford University can today confirm that 38 candidates have successfully submi
    '\n' +
    'Read more here ',
  like: 860,
  reply: 593,
  repost: 513,
  views: 897465,
  tweetIMG: [
    'https://pbs.twimg.com/card_img/1849016588940312576/a_s_iTn5?format=jpg&name=small'
  ]
}
{
  _id: ObjectId('6720784ad36cf0f94ff9058f'),
  userID: 'The Cultural Programme\n' +
    '@oxfculturalprog\n' +
```

Hình 3.27 Tìm danh sách bài đăng có chứa từ “oxford” và có hình ảnh đi kèm

-- Tính số lượt xem trung bình của các bài đăng có chứa từ "Oxford" trong nội dung và có ít nhất 1000 lượt thích

```
> db.Twitter_Oxford.aggregate([
    { $match: { tweetText: { $regex: "Oxford", $options: "i" }, like: { $gte: 1000 } } },
    { $group: { _id: null, avgViews: { $avg: "$views" } } }
])
< {
  _id: null,
  avgViews: 96787.8
}
```

Hình 3.28 Tính lượt xem trung bình các bài đăng có chứa từ “Oxford” và có nhiều hơn 1000 lượt thích

-- Tìm bài đăng có lượt thích nhiều nhất và hiển thị thông tin chi tiết về bài đăng đó.

```
db.Twitter_Oxford.aggregate([
    {
        $sort: { like: -1 }
    },
    {
        $limit: 1
    },
    {
        $project: {
            tweetText: 1,  // Nội dung bài đăng
            like: 1,  // Lượt thích
            views: 1,  // Lượt xem
            tweetIMG: 1  // Hình ảnh
        }
    }
]);
{
  _id: ObjectId('67207d58d36cf0f94ff906e4'),
  tweetText: 'Eid Mubarak to everyone celebrating around the world \n' +
    '\n' +
    '#EidAlAdha',
  like: 7190,
  views: 173869,
  tweetIMG: [
    'https://pbs.twimg.com/media/GQIofe1WgAAeNGe?format=jpg&name=small'
  ]
}
```

Hình 3.29 Tìm bài đăng có lượt thích cao nhất và chỉ lấy một

-- Tính tổng số lượt thích và lượt chia sẻ của tất cả bài đăng trong tháng 10

```
> db.Twitter_Oxford.aggregate([{
        $addFields: {
            date: {
                $dateFromString: { dateString: "$timePost" }
            }
        }
    },
    {
        $project: {
            month: { $month: "$date" },
            like: 1,
            repost: 1
        }
    },
    {
        $match: { month: 10 }
    },
    {
        $group: {
            _id: null,
            totalLikes: { $sum: "$like" },
            totalReposts: { $sum: "$repost" }
        }
    }
])
< {
    _id: null,
    totalLikes: 19942,
    totalReposts: 4162
```

3.30 Tính tổng số lượt thích và lượt chia sẻ trong tháng 10

-- Tìm 5 bài viết có lượt chia sẻ nhiều nhất và có lượt thích trung bình cao nhất

```
> db.Twitter_Oxford.aggregate([
    {
        $group: {
            _id: "$tweetText",
            avgLikes: { $avg: "$like" },
            reposts: { $first: "$repost" }
        }
    },
    {
        $sort: { reposts: -1, avgLikes: -1 }
    },
    {
        $limit: 5 // Lấy 5 bài đăng đầu tiên
    }
])
< {
    _id: 'NEW: A study has found that hospitals that are privatised typ
      '\n' +
      'Researchers analysed 13 long-term studies from different high-ir
      '\n' +
      'More info',
    avgLikes: 2324,
    reposts: 1936
  }
```

Hình 3.31 Tìm 5 bài viết có lượt chia sẻ nhiều nhất và lượt thích trung bình cao nhất

-- Tìm tất cả các bài đăng có hơn 6000 lượt thích và đếm tổng số từ trong nội dung của các bài đăng này

```
> db.Twitter_Oxford.aggregate([
      { $match: { like: { $gt: 6000 } } },
      { $project: { wordCount: { $size: { $split: ["$tweetText", " "] } } } },
      { $group: { _id: null, totalWords: { $sum: "$wordCount" } } }
  ])
< {
    _id: null,
    totalWords: 55
  }
```

Hình 3.32 Tìm bài đăng hơn 6000 lượt thích và đếm số từ của bài đăng

-- Tìm các bài đăng có từ 2 hình ảnh trở lên và tính tổng lượt xem của các bài này

```
db.Twitter_Oxford.aggregate([
    {
        $match: {
            tweetIMG: { $exists: true, $type: "array" }  // Kiểm tra nếu trường tweetIMG tồn tại và là mảng
        }
    },
    {
        $match: {
            $expr: { $gte: [{ $size: "$tweetIMG" }, 2] }  // Kiểm tra kích thước của tweetIMG
        }
    },
    {
        $group: {
            _id: null,
            totalViews: { $sum: "$views" }  // Tính tổng số lượt xem
        }
    }
])
{
  _id: null,
  totalViews: 1912108
}
```

Hình 3.33 Tìm bài đăng có 2 hình anh trở lên và tính tổng lượt xem các bài đăng này

-- Xác định số lượt thích trung bình của các bài đăng có ảnh và so sánh với các bài đăng không có ảnh

```
> db.Twitter_Oxford.aggregate([
    { $facet: {
        withImages: [
            { $match: { tweetIMG: { $ne: [] } } },
            { $group: { _id: null, avgLikes: { $avg: "$like" } } }
        ],
        withoutImages: [
            { $match: { tweetIMG: { $eq: [] } } },
            { $group: { _id: null, avgLikes: { $avg: "$like" } } }
        ]
    }}
])
< {
  withImages: [
    {
      _id: null,
      avgLikes: 265.229862475442
    }
  ],
  withoutImages: [
    {
      _id: null,
      avgLikes: 171.82119205298014
    }
  ]
}
```

Hình 3.34 Tìm bài đăng có ảnh và không ảnh rồi so sánh lượt thích trung bình

-- Tìm các bài đăng có tổng số lượt xem cao nhất và đếm số bài đăng có ít nhất 1000 lượt thích

```
db.Twitter_Oxford.aggregate([
    {
        $group: {
            _id: "$tweetText",  // Nhóm theo nội dung bài đăng
            totalViews: { $sum: "$views" },  // Tính tổng số lượt xem
            likeCount: { $sum: "$like" }  // Tính tổng số lượt thích
        }
    },
    {
        $sort: { totalViews: -1 }  // Sắp xếp theo tổng số lượt xem giảm dần
    },
    {
        $project: {
            tweetText: "$_id",  // Đặt tên trường cho nội dung bài đăng
            totalViews: 1,  // Giữ lại tổng số lượt xem
            likeCount: 1,  // Giữ lại tổng số lượt thích
            isPopular: { $gte: ["$likeCount", 1000] }  // Kiểm tra xem bài đăng có ít nhất 1000 lượt thích không
        }
    },
    {
        $match: { isPopular: true }  // Lọc các bài đăng có ít nhất 1000 lượt thích
    },
    {
        $count: "postCount"  // Đếm số bài đăng thỏa mãn điều kiện
    }
])
{
  postCount: 35
}
```

Hình 3.35 Tìm bài đăng có tổng lượt xem cao nhất và đếm số bài đăng trên 1000 lượt thích

-- Xác định bài đăng có lượt thích trung bình cao nhất có trên 500 lượt xem và có ít nhất 2 hình ảnh

```
db.Twitter_Oxford.aggregate([{
        $match: {
            views: { $gt: 500 },
            tweetIMG: { $exists: true, $type: "array" }  }
    },
    {   $match: {
            $expr: { $gte: [{ $size: "$tweetIMG" }, 2] }  }
    },
    {       $group: {
            _id: "$tweetText",  // Nhóm theo nội dung bài đăng
            totalLikes: { $sum: "$like" },
            postCount: { $sum: 1 }  // Đếm số bài đăng
        }
    },
    {       $project: {
            tweetText: "$_id",  // Đặt tên trường cho nội dung bài đăng
            avgLikes: { $divide: ["$totalLikes", "$postCount"] }  // Tính lượt thích trung bình
        }
```

Hình 3.36 Tìm bài đăng có lượt thích trung bình cao nhất và có ít nhất 2 hình

CHƯƠNG 4: KIẾN NGHỊ VÀ KẾT LUẬN

4.1. Kết luận

Thông qua quá trình thu thập dữ liệu, tuy vẫn gặp nhiều khó khăn nhưng cuối cùng cũng đạt được mục tiêu ban đầu đề ra là thu thập dữ liệu các bài đăng của trang cá nhân trên mạng xã hội X. Dựa vào đó, đã thu được các thông tin liên quan đến như nội dung bài đăng, lượt thích, lượt tương tác, các hình ảnh đi kèm,… Và lưu tất cả dữ liệu đã thu thập được vào cơ sở dữ liệu MongoDB, giúp dễ dàng phân tích và khai thác dữ liệu một cách nhanh chóng.

Ban đầu, mục tiêu của việc thu thập thông tin từ các bài báo có phần bình luận, hoặc phản hồi của người dùng. Tuy nhiên, do đặc thù của nền tảng X, người dùng thường tương tác bằng cách chia sẻ hơn là bình luận. Nhận thấy số lượng bình luận rất thấp mặc dù đây là tài khoản của một trường đại học rất danh giá ở Mỹ, và chúng không có giá trị phân tích. Vì vậy, nghiên cứu đã chuyển hướng sang phân tích lượt xem – là một chỉ số phản ánh rõ ràng hơn mức độ phổ biến của các bài đăng. Cuối cùng, đã đưa ra nhận đây rằng đây là dữ liệu có giá trị hơn, giúp làm rõ những vấn đề mà người dùng đang quan tâm đến.

- *Đánh giá hiệu suất và độ chính xác*

Đây là dữ liệu với độ chính xác cao bởi các yếu tố liên quan trực tiếp đến người dùng, các đối tượng mà những bài viết tiếp cận được. Tất cả đều được kiểm chứng và cho kết quả trùng khớp với dữ liệu trực tiếp từ trang này. Tuy nhiên, tốc độ thu thập dữ liệu chỉ ở mức từ thấp đến trung bình, một phần vì đây là lần đầu thử nghiệm trên một nền tảng khá mới. Và hiệu suất này cũng ảnh hưởng bởi các yếu tố như tốc độ đường truyền mạng, cộng với giới hạn về tốc độ. Vì đây là trang web mạng xã hội động nên không thể thu thập hết dữ liệu, chỉ có thể cuộn trang đến một mức độ nhất định, mặc dù có khả năng vẫn còn nội dung chưa hoặc không thể hiển thị vì một lí do nào đó.

4.2 Hạn chế và kiến nghị

4.2.1. Hạn chế của chương trình

Một vấn đề ảnh hưởng cực lớn đến quá trình thu thập dữ liệu đó là tốc độ đường truyền mạng bị giới hạn. Vì là một trang web động, nên không thể cuộn trang với tốc độ quá nhanh. Dường như trang web có một giới hạn nào đó đối với việc tải dữ liệu, bởi vì khi dữ liệu được thu thập tới hơn 600 dòng trang web ngừng hiển thị các tweet mới, mặc dù số lượt tweet của trang lên đến hơn 20.000 bài.

4.2.2 Kiến nghị

Đầu tiên, do sự thành công của việc thu thập dữ liệu từ trang trường đại học Oxford, nên việc áp dụng thu thập ở các trang khác cũng sẽ khả quan hơn mặc dù sẽ có một chút thay đổi nhất định ở các câu lệnh. Vì vậy, trong giai đoạn tới, hướng phát triển có thể chuyển dần sang thu thập dữ liệu từ những người có sức ảnh hưởng (Influencers) trong các lĩnh vực cụ thể, ví dụ như thể thao hoặc giải trí. Điều này giúp tệp dữ liệu thu thập được có giá trị hơn do nhắm vào một đối tượng cụ thể thay vì chung chung.

Thứ hai, dự án này đang hướng tới việc phát triển một công cụ tự động hóa quá trình thu thập dữ liệu hàng ngày từ các trang web. Việc này giúp làm tăng hiệu quả thu thập.

Thứ ba, mở rộng phạm vi nghiên cứu. Có thể thu thập dữ liệu từ các nền tảng lớn khác như Facebook, Instagram. Ngoài ra, việc phân tích dữ liệu theo thời gian cũng là một vấn đề cần quan tâm, giới trẻ thường hay dùng từ "xu hướng" để mô tả một sự quan tâm nào đó trong một khoảng thời gian nhất định. Việc cập nhật xu hướng theo thời gian có thể diễn ra hàng tháng, thậm chí là hàng tuần, hàng ngày.

Thứ tư, cân nhắc về vấn đề văn hóa. Giữa mỗi châu lục, hay mỗi khu vực sẽ có những chủ đề riêng mà họ quan tâm đến. Có thể chuyển hướng sang phân tích theo từng quốc gia, khu vực để có thể nhận được một kho dữ liệu trực quan, chi tiết và sát với thực tế hơn.

[1] "Introduction to Web Scraping," GeeksforGeeks. Accessed: Oct. 19, 2024. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-web-scraping/
[2] "Is Web Scraping Legal?" Accessed: Oct. 20, 2024. [Online]. Available: https://oxylabs.io/blog/is-web-scraping-legal
[3] "Getting Started with Python Programming," GeeksforGeeks. Accessed: Oct. 20, 2024. [Online]. Available: https://www.geeksforgeeks.org/getting-started-with-python-programming/
[4] "Python Language advantages and applications," GeeksforGeeks. Accessed: Oct. 23, 2024. [Online]. Available: https://www.geeksforgeeks.org/python-language-advantages-applications/
[5] "Applications for Python," Python.org. Accessed: Oct. 23, 2024. [Online]. Available: https://www.python.org/about/apps/
[6] "Tổng quan về Selenium và vai trò của các thành phần." Accessed: Oct. 22, 2024. [Online]. Available: https://tech.cybozu.vn/tong-quan-ve-selenium-va-vai-tro-cua-cac-thanh-phan-74a12/
[7] Tiến C. L. V., "NoSQL là gì? Các thông tin về cơ sở dữ liệu NoSQL." Accessed: Oct. 25, 2024. [Online]. Available: https://vietnix.vn/nosql-la-gi/
[8] "MONGODB LÀ GÌ? TÍNH NĂNG NỔI BẬT CỦA MONGODB MÀ BẠN CẦN BIẾT," 200Lab Blog. Accessed: Oct. 26, 2024. [Online]. Available: https://200lab.io/blog/mongodb-la-gi/

PHỤ LỤC 1

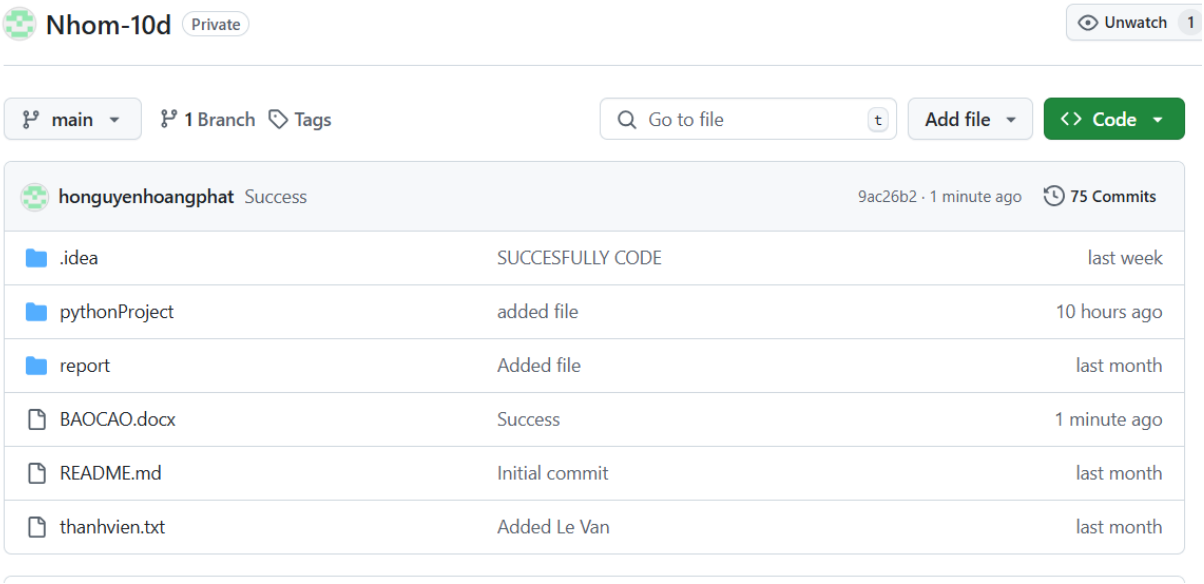
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
import pandas as pd
from selenium.webdriver.chrome.service import Service
from pymongo import MongoClient

# #kết nối
client = MongoClient("mongodb://localhost:27017/")
client.drop_database('Twitters')
db = client['Twitters']  # chon csdl
collection = db['Twitter_Oxford']

# Đường dẫn đến file thực thi geckodriver
#gecko_path = r"D:/Nhom10/Nhom-10d/pythonProject/geckodriver.exe"
gecko_path = r"D:/DoAnNhom10d/Nhom-10d/pythonProject/geckodriver.exe"

ser = Service(gecko_path) # Khởi tạo đối tượng dịch vụ với geckodriver

# Tạo tùy chọn
options = webdriver.firefox.options.Options()
options.binary_location ="C:/Program Files/Mozilla Firefox/firefox.exe"
# Thiết lập firefox chỉ hiện thị giao diện
options.headless = True

# Khởi tạo driver
driver = webdriver.Firefox(options=options, service=ser)
driver.get("https://x.com/i/flow/login")

time.sleep(5)

#Đăng nhập
username = driver.find_element(By.XPATH, "//input[@name='text']")
username.send_keys("hnhp113")
time.sleep(1)
name = driver.find_element(By.XPATH, "//span[contains(text(),'Tiếp
theo')]")
name.click()

time.sleep(2)
#Xác thực thông tin
try:
    email = driver.find_element(By.XPATH, "//input[@name='text']")
    email.send_keys("hnhp113114115@gmail.com")
    tt = driver.find_element(By.XPATH, "//span[contains(text(),'Tiếp
theo')]")
    tt.click()
except:
    pass
time.sleep(2)
#Mật khẩu
password = driver.find_element(By.XPATH, "//input[@name='password']")
password.send_keys("phatho0317")
pw = driver.find_element(By.XPATH, "//span[contains(text(),'Đăng nhập')]")
pw.click()

time.sleep(10)
```

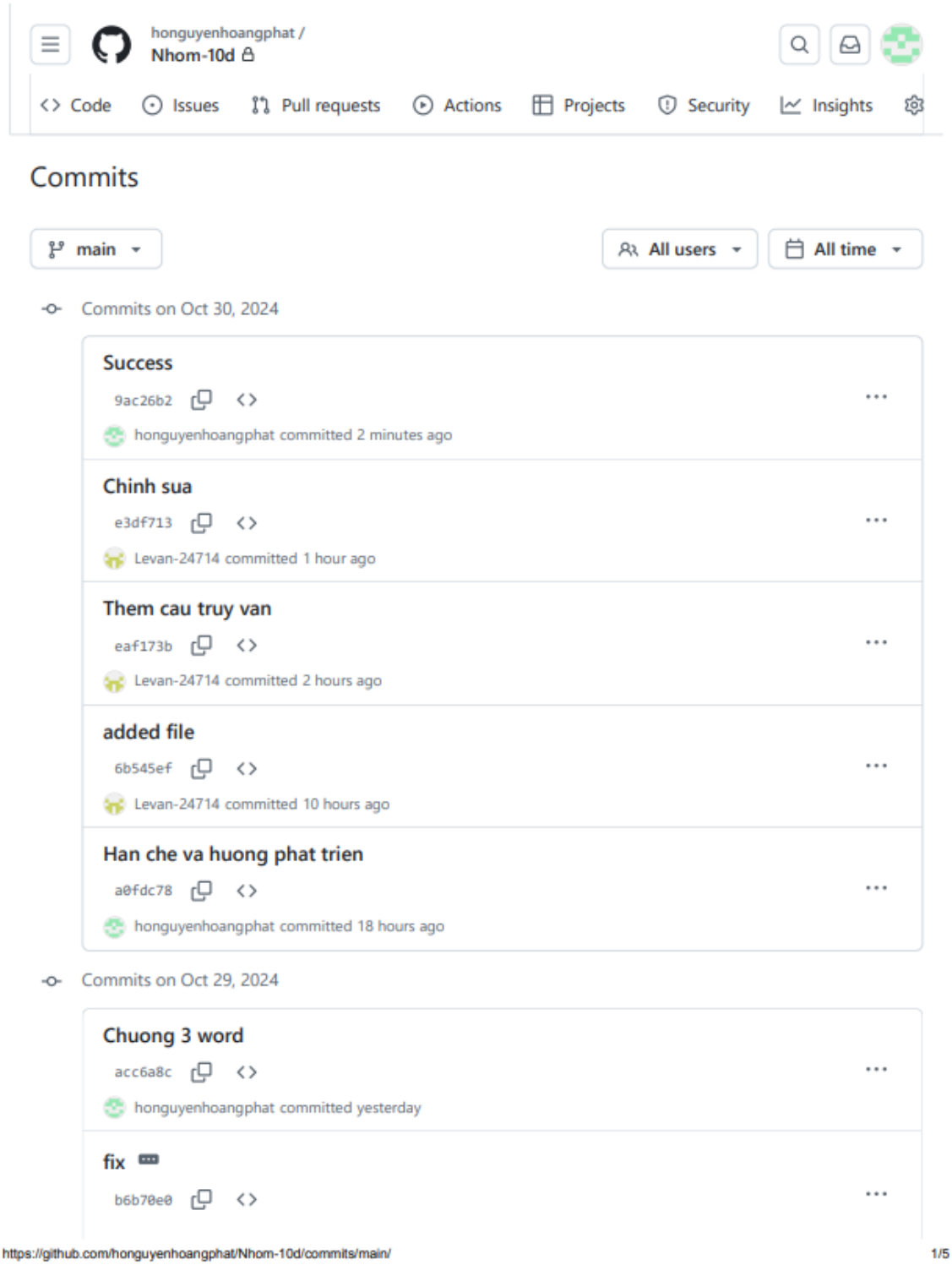
```
#Tìm trang cá nhân bạn muốn quét dữ liệu
search = driver.find_element(By.XPATH, "//input[@data-
testid='SearchBox_Search_Input']")
idol = "Oxford University"
search.send_keys(idol)
search.send_keys(Keys.ENTER)
time.sleep(5)

#Qua tab people để tìm, vì ở trang chính có khi k co
people = driver.find_element(By.XPATH, "//span[contains(text(),'People')]")
people.click()
time.sleep(7)

#Nhấp vao người bạn muốn tìm
name_idol = driver.find_element(By.XPATH, "//*[@id='react-
root']/div/div/div[2]/main/div/div/div/div[1]/div/div[3]/section/div/div/di
v[1]/div/div/button/div/div[2]/div[1]/div[1]/div/div[1]/a/div/div[1]/span/s
pan[1]")
name_idol.click()
time.sleep(2)

def scrape_tweets(driver, max_tweets = 660):
    #Luu du luu kiem tra bai viet trung lap:
    global document
    data_set = set()
    count = 0

    userIDs = []
    timePosts = []
    tweetTexts = []  # = Post status
    likes = []
    replys = []
    reposts = []
    views = []
    tweetIMG=[]

    while count < max_tweets:
        articles = driver.find_elements(By.XPATH, "//article[@data-
testid='tweet']")
        for article in articles:
            try:
                userID = article.find_element(By.XPATH, ".//div[@data-
testid='User-Name']").text
            except:
                userID = ''
            try:
                timePost = driver.find_element(By.XPATH,
".//time").get_attribute("datetime")
            except:
                timePost = ''
            try:
                tweetText = driver.find_element(By.XPATH, ".//div[@data-
testid='tweetText']").text
            except:
                tweetText = ''
            try:
                like_count = driver.find_element(By.XPATH,
".//button[contains(@aria-label,'Likes')]")
                like = like_count.get_attribute('aria-label').split(' ')[0]
            except:
                like = ''
            try:
                reply_count = driver.find_element(By.XPATH,
```

```
".//button[contains(@aria-label,'Replies')]")
                reply = reply_count.get_attribute('aria-label').split('
')[0]
            except:
                reply = ''
            try:
                # resport = driver.find_element(By.XPATH, ".//button[@data-
testid='retweet']").text
                repost_count = driver.find_element(By.XPATH,
".//button[contains(@aria-label,'reposts')]")
                repost = repost_count.get_attribute('aria-label').split('
')[0]
            except:
                repost = ''

            try:
                view = driver.find_element(By.XPATH, ".//a[contains(@aria-
label,'views')]")
                views_count = view.get_attribute('aria-label').split('
')[0]
            except:
                views_count = ''
            try:
                images = article.find_elements(By.XPATH,
".//img[contains(@src, 'https://pbs.twimg.com') and not (contains(@src,
'profile_images'))]")
                tweetIMGs = [img.get_attribute('src') for img in images]
            except:
                tweetIMGs = ''

            # Tạo document dữ liệu để lưu trữ trong MongoDB
            document = {
                "userID": userID,
                "timePost": timePost,
                "tweetText": tweetText,
                "like": like,
                "reply": reply,
                "repost": repost,
                "views": views_count,
                "tweetIMG": tweetIMGs

            }

            #Kiểm tra xem tweet có trùng lặp không
            if tweetText not in tweetTexts:
                data_set.add(tweetText)
                userIDs.append(userID)
                timePosts.append(timePost)
                tweetTexts.append(tweetText)
                likes.append(like)
                replys.append(reply)
                reposts.append(repost)
                views.append(views_count)
                tweetIMG.append(tweetIMGs)
                collection.insert_one(document) #Chèn data Mongo
                count += 1

                if count > max_tweets:
                    break


        #Cuộn chậm
        driver.execute_script("window.scrollBy(0,600);")
```

```
        time.sleep(3)
        #Lấy thêm tweets mới sau khi cuộn
        print(len(set(tweetTexts)))
    print(f"Dữ liệu đã được lưu vào MongoDB. Đã cào được {count} bài")
scrape_tweets(driver)
driver.quit()


def convert_numeric_fields(collection):
    for document in collection.find():
        update_fields = dict()

        for field, value in document.items():
            # Kiểm tra nếu giá trị là chuỗi và có thể chuyển đổi thành số
            if isinstance(value, str):
                try:
                    num_value = int(value) if value.isdigit() else
float(value)
                    update_fields[field] = num_value #Cap nhat gia tri
                except:
                    continue
        # Nếu có bất kỳ trường nào cần cập nhật, thực hiện cập nhật
        if update_fields:
            collection.update_one({'_id': document['_id']}, {'$set':
update_fields})

# Kết nối lại với cơ sở dữ liệu MongoDB và gọi hàm chuyển đổi
convert_numeric_fields(collection)

print("Đã chuyển đổi tất cả các trường từ chuỗi thành số thành công.")
# print('Tổng số tệp:', collection.count_documents({}))
```

PHỤ LỤC 2


Nhom-10d
Private
Unwatch 1
main
1 Branch
Tags
Go to file
t
Add file
Code
honguyenhoangphat Success
9ac26b2 · 1 minute ago
75 Commits
.idea SUCCESFULLY CODE last week
pythonProject added file 10 hours ago
report Added file last month
BAOCAO.docx Success 1 minute ago
README.md Initial commit last month
thanhvien.txt Added Le Van last month


honguyenhoangphat /
Nhom-10d
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Commits
main
All users
All time
Commits on Oct 30, 2024
Success
9ac26b2
honguyenhoangphat committed 2 minutes ago
Chinh sua
e3df713
Levan-24714 committed 1 hour ago
Them cau truy van
eaf173b
Levan-24714 committed 2 hours ago
added file
6b545ef
Levan-24714 committed 10 hours ago
Han che va huong phat trien
a0fdc78
honguyenhoangphat committed 18 hours ago
Commits on Oct 29, 2024
Chuong 3 word
acc6a8c
honguyenhoangphat committed yesterday
fix
b6b70e0

Levan-24714 committed yesterday

added file

808a266

Levan-24714 committed yesterday

update

129ba03

Levan-24714 committed yesterday

Commits on Oct 28, 2024

.

f74c2a5

Levan-24714 committed 2 days ago

.

c674b58

Levan-24714 committed 2 days ago

Add data MongoDB

8aafdff

honguyenhoangphat committed 2 days ago

update

2c47a50

Levan-24714 committed 2 days ago

update

59fbf97

Levan-24714 committed 3 days ago

Commits on Oct 27, 2024

C3 do an

b7bebdc

honguyenhoangphat committed 3 days ago

update

8e625d0

Levan-24714 committed 3 days ago

added mongodb

b1c082f

Levan-24714 committed 4 days ago

Commits on Oct 26, 2024

update

ce85ce7

Levan-24714 committed 4 days ago

.

5dbcc54

Levan-24714 committed 5 days ago

update

82a868f

Levan-24714 committed 5 days ago

Commits on Oct 25, 2024

Firefox

24b4530

honguyenhoangphat committed 5 days ago

update

98dfc4d

Levan-24714 committed last week

Commits on Oct 24, 2024

update

2007ab5

Levan-24714 committed last week

Upd word

c288289

honguyenhoangphat committed last week

Upd word

bc7ac95

honguyenhoangphat committed last week

Upd word

303d01b

honguyenhoangphat committed last week

Commits on Oct 23, 2024

update

36441ef

Levan-24714 committed last week

delete

8a0f365

Levan-24714 committed last week

cap nhap

6fc4e97

Levan-24714 committed last week

update

caae454

Levan-24714 committed last week

update

d3f33a5

Levan-24714 committed last week

Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d

747a905

honguyenhoangphat committed last week

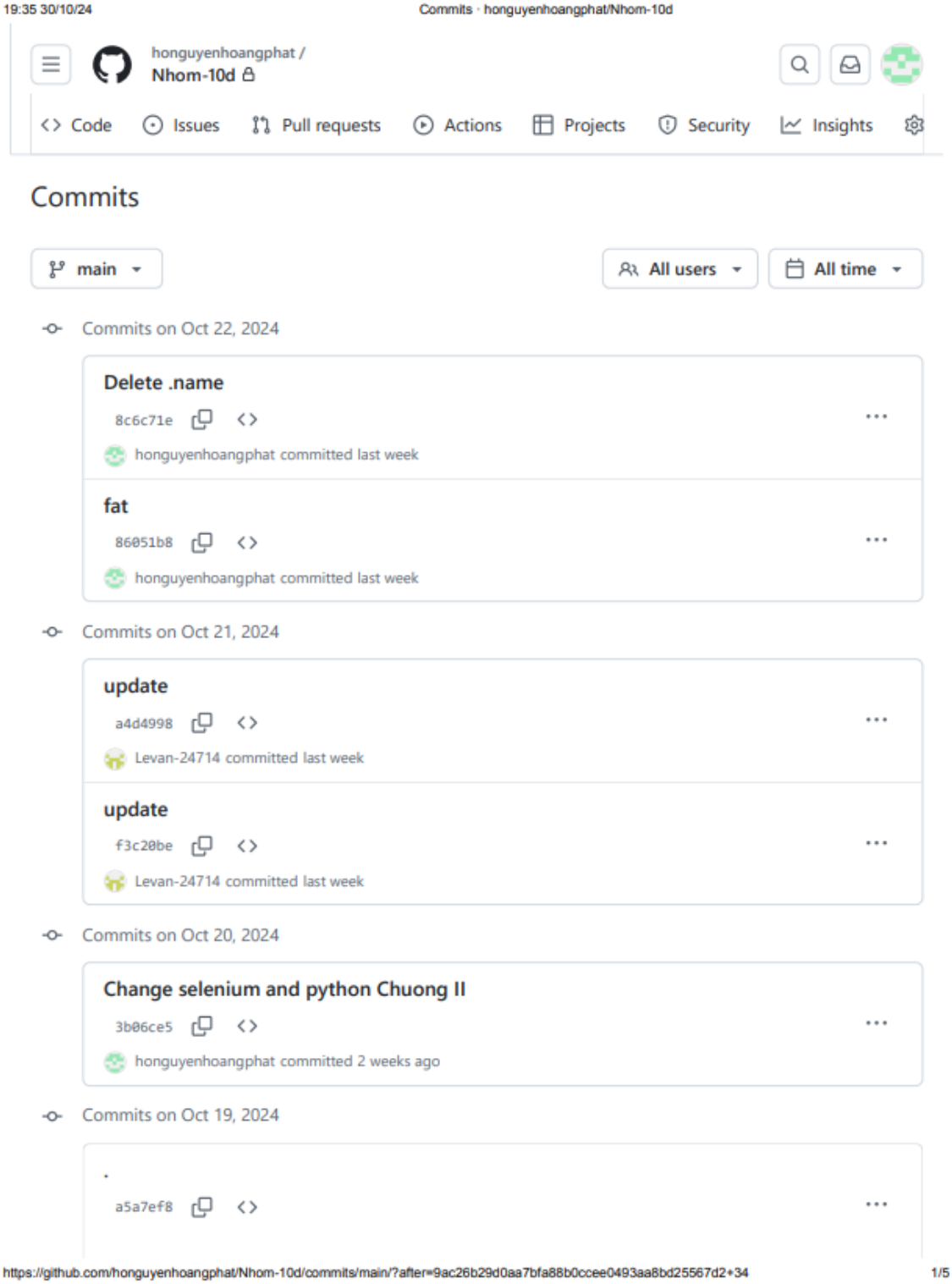
SUCCESFULLY CODE

259f196

honguyenhoangphat committed last week

Commits on Oct 22, 2024

update


honguyenhoangphat /
Nhom-10d
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Commits
main
All users
All time
Commits on Oct 22, 2024
Delete .name
8c6c71e
honguyenhoangphat committed last week
fat
86051b8
honguyenhoangphat committed last week
Commits on Oct 21, 2024
update
a4d4998
Levan-24714 committed last week
update
f3c20be
Levan-24714 committed last week
Commits on Oct 20, 2024
Change selenium and python Chuong II
3b06ce5
honguyenhoangphat committed 2 weeks ago
Commits on Oct 19, 2024
.
a5a7ef8

honguyenhoangphat committed 2 weeks ago

Conflic
23629b3
honguyenhoangphat committed 2 weeks ago

Upd
627b401
honguyenhoangphat committed 2 weeks ago

chuong 2
40f7eae
Levan-24714 committed 2 weeks ago

update
a9fe23c
Levan-24714 committed 2 weeks ago

Update
7c7e615
Levan-24714 committed 2 weeks ago

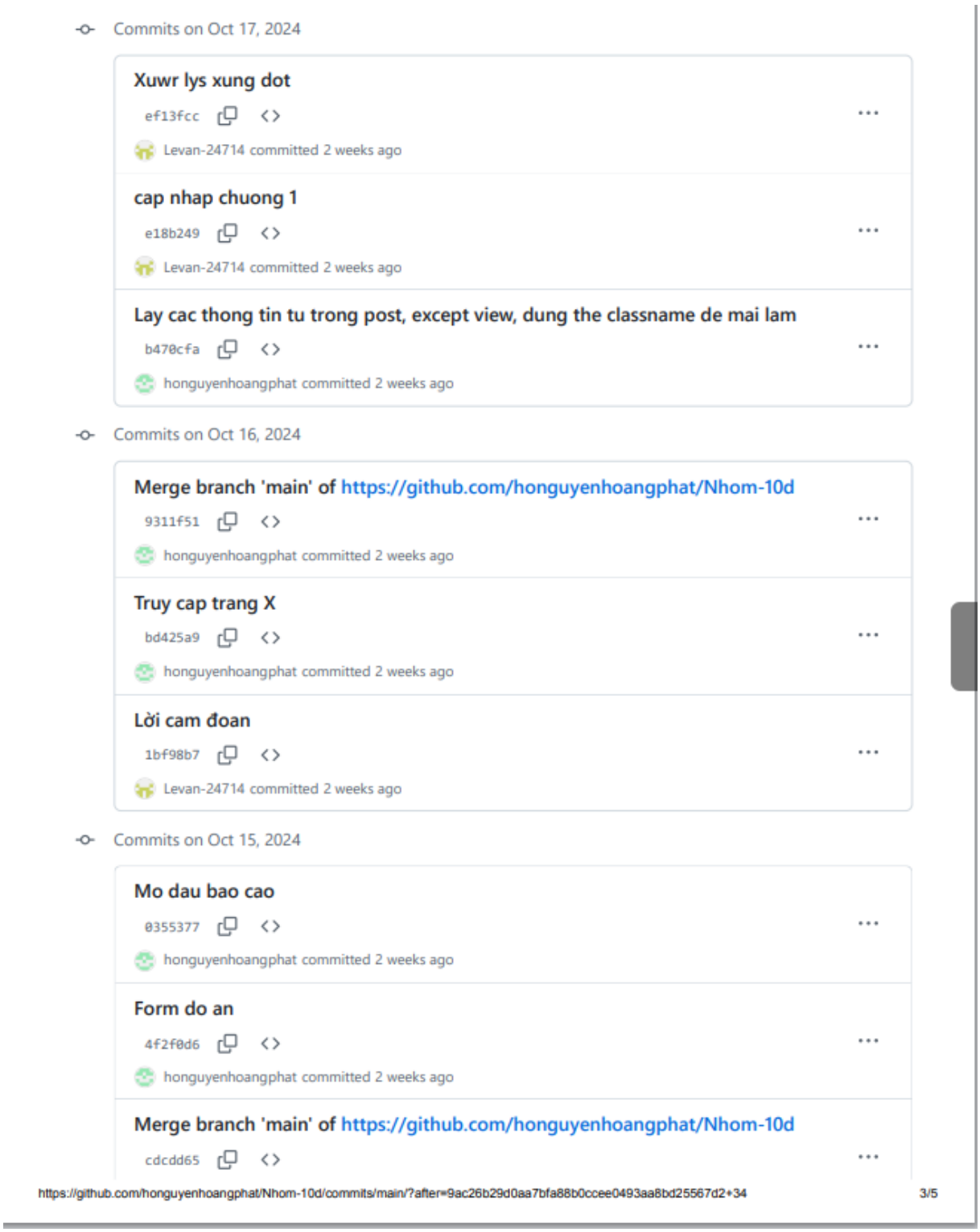
Commits on Oct 18, 2024

Xu ly trung lap data thanh cong
9df9228
honguyenhoangphat committed 2 weeks ago

Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d
b615fd4
honguyenhoangphat committed 2 weeks ago

Lay view thanh cong, xu ly loi tranh trung lap, mai se ghep vao bai 0
d81fa0f
honguyenhoangphat committed 2 weeks ago

Chuong 1
beddfaa
Levan-24714 committed 2 weeks ago


Commits on Oct 17, 2024
Xuwr lys xung dot
ef13fcc
Levan-24714 committed 2 weeks ago
cap nhap chuong 1
e18b249
Levan-24714 committed 2 weeks ago
Lay cac thong tin tu trong post, except view, dung the classname de mai lam
b470cfa
honguyenhoangphat committed 2 weeks ago
Commits on Oct 16, 2024
Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d
9311f51
honguyenhoangphat committed 2 weeks ago
Truy cap trang X
bd425a9
honguyenhoangphat committed 2 weeks ago
Lời cam đoan
1bf98b7
Levan-24714 committed 2 weeks ago
Commits on Oct 15, 2024
Mo dau bao cao
0355377
honguyenhoangphat committed 2 weeks ago
Form do an
4f2f0d6
honguyenhoangphat committed 2 weeks ago
Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d
cdcdd65
https://github.com/honguyenhoangphat/Nhom-10d/commits/main/?after=9ac26b29d0aa7bfa88b0ccee0493aa8bd25567d2+34
3/5

honguyenhoangphat committed 2 weeks ago

code python

2d3afe9

honguyenhoangphat committed 2 weeks ago

update file

5af3b60

Levan-24714 committed 2 weeks ago

Commits on Oct 14, 2024

added file BAOCAO.docx

e2a3816

Levan-24714 committed 2 weeks ago

Commits on Sep 30, 2024

Added file

83cc01e

Levan-24714 committed last month

sua trang bia

46ccbad

Levan-24714 committed on Sep 30

Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d

7c63b83

honguyenhoangphat committed on Sep 30

Add trang bia

c7e3a49

honguyenhoangphat committed on Sep 30

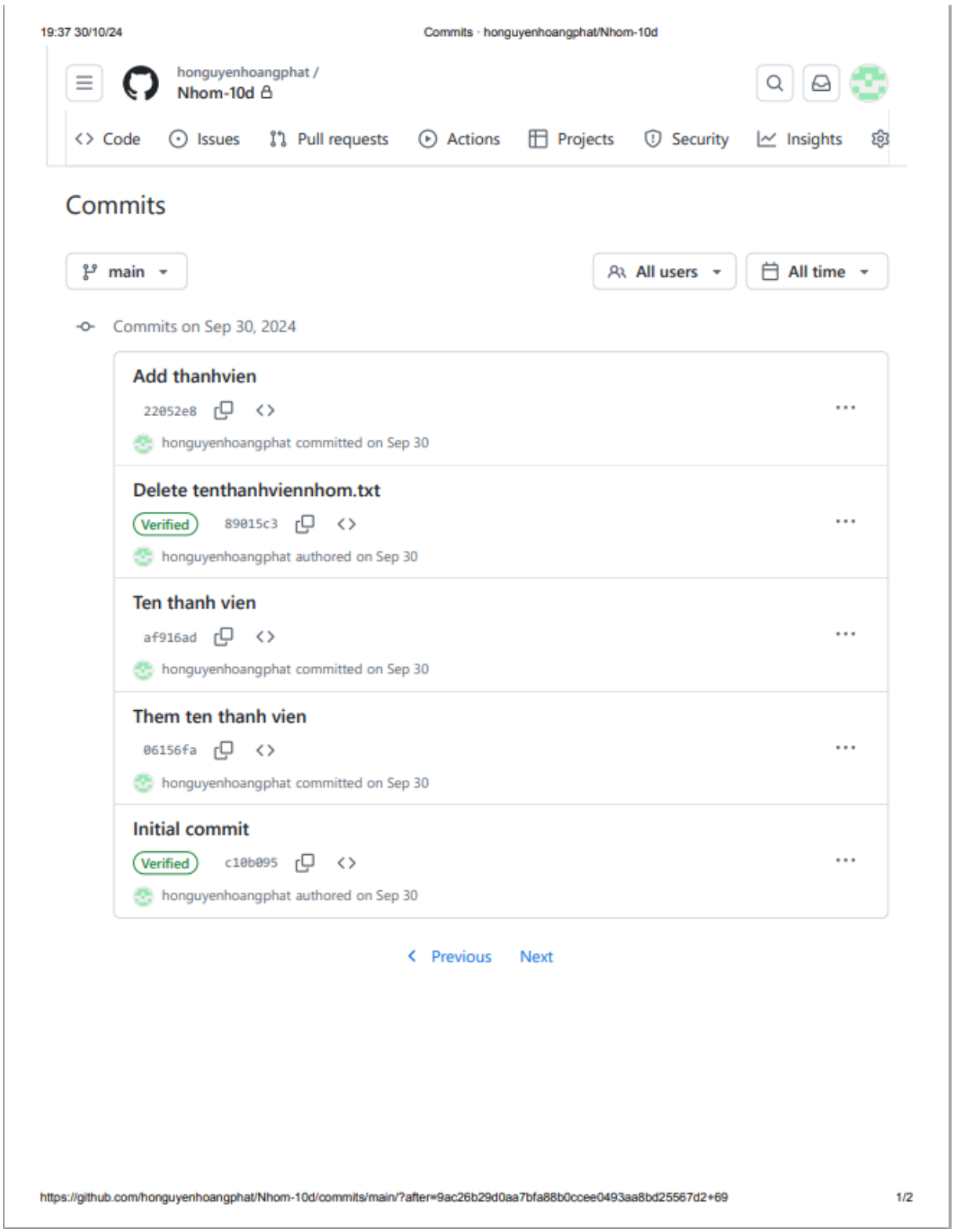
Added Le Van

d712c23

Levan-24714 committed on Sep 30

Merge branch 'main' of https://github.com/honguyenhoangphat/Nhom-10d

a578491



Link dự án github: https://github.com/honguyenhoangphat/Nhom-10d.git