

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC SAO ĐỎ



ĐỒ ÁN TỐT NGHIỆP

Ngành: Công nghệ thông tin

TÊN ĐỀ TÀI: NGHIÊN CỨU, LẬP TRÌNH ỨNG DỤNG QUÉT BÀI VIẾT,
COMMENT TRONG FACEBOOK GROUP

Họ tên sinh viên: Phạm Quốc Hùng - Đỗ Văn Hoàn

Lớp: DK10-CNTT

Giảng viên hướng dẫn: ThS. Phạm Văn Kiên

MỤC LỤC

MỤC LỤC	i
DANH MỤC CÁC TỪ VIẾT TẮT	iii
DANH MỤC HÌNH ẢNH	iv
LỜI CAM ĐOAN	v
Chương 1: CƠ SỞ LÝ THUYẾT	3
1.1. Tổng quan API	3
1.1.1. API là gì.....	3
1.1.2. API hoạt động ra sao	3
1.1.3. Các thuật ngữ quan trọng trong API.....	4
1.1.4. Một số đặc tính của REST	4
1.1.5. Các ràng buộc	5
1.1.6. Ưu điểm và nhược điểm của API	5
1.1.7. API thường ứng dụng vào đâu.....	7
1.1.8. Một số cách để ngăn chặn rò rỉ thông tin khi dùng API.....	7
1.2. Tổng quan về ngôn ngữ lập trình C#	8
1.2.1. Ngôn ngữ lập trình C# dùng để làm gì	8
1.2.2. Đặc trưng của ngôn ngữ C#.....	9
1.2.3. Mục tiêu khi phát triển C#	10
1.2.4. Tính ứng dụng của ngôn ngữ C#	11
1.3. Sql server	12
1.3.1. SQL là gì.....	12
1.3.2. Tại sao sử dụng SQL	13
1.3.3. Tóm tắt lịch sử của SQL.....	14
1.3.4. Ưu điểm của SQL	15
1.3.5. Nhược điểm của SQL	16
1.3.6. Phân loại câu lệnh SQL	16
1.4. Công cụ sử dụng	19
1.4.1. Visual studio	19
1.4.2. Dev express.....	20
1.4.3. Sql server	21
Chương 2: Kỹ thuật tương tác với facebook	23
2.1. Api facebook.....	23
2.1.1. Facebook Graph API	23
2.1.2. Đăng ký ứng dụng graph api	24
2.1.3. Phương thức get.....	26
2.1.4. Phương thức post	26
2.2. Thư viện seleclum	27
2.3. Thư viện Facebook SDK	29
2.4. Tương tác với facebook	30
Chương 3: Xây dựng ứng dụng	33
3.1. Cơ sở dữ liệu.....	33
3.2. Kết hợp API Facebook vào ứng dụng	35

3.3. Thiết kế giao diện người dùng(UI).....	36
3.3.1. Giao diện chính của đồ án	36
3.3.2. Màn hình tài khoản	37
3.3.3. Màn hình quét bài viết	37
3.3.4. Màn hình tải video từ trang	39
3.3.5. Màn hình quét comment bài viết	40
3.4. Xây dựng chương trình.....	42
3.5. Thử nghiệm.....	51
KẾT LUẬN	52
TÀI LIỆU THAM KHẢO	56

Đại học Sao Đỏ

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Giải nghĩa
API	Application Programming Interface
SQL	Structured Query Language
URL	Uniform Resource Locator
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol

Đại học Sao Đỏ

DANH MỤC HÌNH ẢNH

Hình 1.1: Cấu trúc api	4
Hình 1.2: Giao tiếp IPA.....	6
Hình 1.3: Khả năng tương tác của api	7
Hình 1.4: Ngôn ngữ C#	12
Hình 1.5: Sql server	12
Hình 1.6: hình ảnh minh họa	14
Hình 1.7: Chức năng của SQL.....	14
Hình 1.8: Nhóm SQL	16
Hình 2.1: Giao diện trang web Facebook Develop	24
Hình 2.2: Tạo ứng dụng facebook	24
Hình 2.3: Hướng dẫn tạo app	26
Hình 2.4: Phân tích và Lưu trữ Dữ liệu	31
Hình 2.5: Xử lý Tương tác người dùng	31
Hình 2.6: Gửi Bài viết và Tương tác ngược lại	32
Hình 2.7: Đăng bài viết lên nhóm	32
Hình 3.1: Bảng tài khoản	33
Hình 3.2: Bảng danh sách bài viết.....	33
Hình 3.3: Bảng danh sách comment.....	33
Hình 3.4: Danh sách stored	34
Hình 3.5: Stored lấy toàn bộ bảng tài khoản	34
Hình 3.6: Stored thêm ACCOUNT	35
Hình 3.7: Giao diện chính	36
Hình 3.8: Màn hình tài khoản.....	37
Hình 3.9: Màn hình quét bài viết.....	37
Hình 3.10: Xuất file excel	38
Hình 3.11: Giao diện khi tải ảnh	39
Hình 3.12: Màn hình tải video từ trang	39
Hình 3.13: Màn hình quét comment bài viết.....	40
Hình 3.14: Thông báo khi thiếu thông tin	40
Hình 3.15: Thông báo khi click button xoá tài khoản.	41
Hình 3.16: Thông báo xoá thành công	41
Hình 3.17: Danh sách các thư viện cần thiết	42
Hình 3.18: Hàm export file excel.	44
Hình 3.19: Hàm đăng nhập.....	44
Hình 3.20: Lấy accesstoken.....	45

LỜI CAM ĐOAN

Tôi xin cam đoan các kết quả đưa ra trong đồ án tốt nghiệp này là các kết quả thu được trong quá trình nghiên cứu, thực nghiệm của tôi dưới sự hướng dẫn của ThS. Phạm Văn Kiên, không sao chép bất kỳ kết quả nghiên cứu nào của các tác giả khác.

Nội dung nghiên cứu có tham khảo và sử dụng một số thông tin, tài liệu từ các nguồn tài liệu đã được liệt kê trong danh mục các tài liệu tham khảo.

Nếu sai tôi xin chịu mọi hình thức kỷ luật theo quy định.

Hải Dương, ngày.....tháng.....năm.....

Sinh viên thực hiện

(Ký, ghi rõ họ và tên)

Đại học Sao Đỏ

MỞ ĐẦU**1. Lý do chọn đề tài**

Lý do chọn đề tài này xuất phát từ nhu cầu thực tế của người quản lý nội dung trên Facebook Group. Với sự phổ biến ngày càng tăng của Facebook và sự đa dạng trong việc chia sẻ thông tin trong nhóm, việc theo dõi và quản lý nội dung trở nên quan trọng hơn bao giờ hết. Sự kết hợp giữa sức mạnh của API Facebook và tiện ích của WinForms C# làm giao diện người dùng không chỉ giúp đơn giản hóa quy trình này mà còn tăng cường khả năng tương tác và quản lý.

Chúng em quan tâm đặc biệt đến khả năng tự động hóa việc quét thông tin từ bài viết, giúp người quản lý tiết kiệm thời gian và nỗ lực. Việc này không chỉ giúp họ theo dõi các sự kiện quan trọng mà còn mang lại lợi ích trong việc phân tích và hiểu rõ ngữ cảnh cộng đồng.

2. Mục tiêu nghiên cứu

Mục tiêu chính của dự án là phát triển một ứng dụng có khả năng tự động quét bài viết trong các nhóm Facebook, đồng thời cung cấp một giao diện người dùng thân thiện và dễ sử dụng. Chúng em muốn tối ưu hóa quá trình thu thập thông tin, đảm bảo tính chính xác và hiệu suất của ứng dụng.

Một mục tiêu phụ khác là tối ưu hóa trải nghiệm người dùng thông qua sự tích hợp linh hoạt và thân thiện của WinForms C#. Chúng em hi vọng rằng ứng dụng sẽ trở thành một công cụ hữu ích không chỉ cho việc quản lý nội dung mà còn cho những người muốn thăm dò thông tin trong các nhóm trên Facebook.

3. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của chúng em là những người quản lý và theo dõi nội dung trong các nhóm Facebook, đặc biệt là những người muốn tự động hóa quá trình quét bài viết. Các người này có thể là quản trị viên cộng đồng, nhà quảng cáo, hoặc bất kỳ ai muốn nắm bắt thông tin và xu hướng trong cộng đồng một cách thuận tiện.

4. Phạm vi nghiên cứu

Phạm vi của dự án được giới hạn vào việc phát triển và triển khai một ứng dụng có khả năng quét bài viết trong một nhóm Facebook cụ thể. Chúng em sẽ tập trung vào việc thu thập thông tin cơ bản như nội dung bài viết, người đăng, và thời gian đăng. Các tính năng mở rộng như phân tích nội dung hoặc theo dõi sự tương tác có thể được xem xét trong các giai đoạn phát triển sau này.

5. Phương pháp nghiên cứu

Chúng em sẽ áp dụng phương pháp phát triển phần mềm Agile để linh hoạt và đáp ứng linh hoạt với sự thay đổi trong quá trình phát triển. Điều này giúp Chúng em nhanh chóng đưa ra các phiên bản thử nghiệm và thu nhận phản hồi từ người dùng, từ đó cải thiện và điều chỉnh ứng dụng một cách hiệu quả. Phương pháp thử nghiệm liên tục cũng sẽ được áp dụng để đảm bảo tính ổn định và hiệu suất của ứng dụng.

6. Ý nghĩa khoa học và thực tiễn

Dự án này không chỉ mang lại giá trị thực tiễn cho người dùng mục tiêu mà còn đóng góp vào lĩnh vực phát triển ứng dụng sử dụng API mạng xã hội. Nó cũng mở ra cánh cửa cho nhiều nghiên cứu hơn về cách tối ưu hóa việc sử dụng API Facebook và phát triển ứng dụng thân thiện với người dùng

7. Kết cấu đề tài

Chương 1: Cơ sở lý thuyết tổng quan về API, ngôn ngữ C#, Sql server.

Chương 2: Kỹ thuật tương tác với facebook.

Chương 3: Xây dựng ứng dụng.

Chương 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan API

API là thuật ngữ không còn quá xa lạ, đặc biệt là trong xu hướng chuyển đổi số đang diễn ra mạnh mẽ như hiện nay. API có tính ứng dụng thực tế cao trong lĩnh vực công nghệ thông tin, tạo sự “tương tác” dễ dàng hơn giữa các hệ thống với nhau, từ đó thúc đẩy sự đổi mới và tăng tính hiệu quả cho nhiều hoạt động từ đời sống, xã hội cho đến hoạt động kinh doanh. Thông qua bài viết này, hãy cùng BAC's Blog tìm hiểu rõ hơn về API, cách hoạt động cũng như tính ứng dụng trong thực tế.

1.1.1. API là gì

API là viết tắt của cụm từ Application Programming Interface - giao diện lập trình ứng dụng, là các giao thức kết nối các thư viện và ứng dụng khác. Với khả năng truy xuất đến một tập các hàm hay dùng, API có thể trao đổi dữ liệu giữa các ứng dụng.

Thực tế API không phải ngôn ngữ lập trình, API chỉ là các hàm thủ tục thông thường. Các hàm này sẽ được viết nhiều dạng ngôn ngữ lập trình khác nhau.

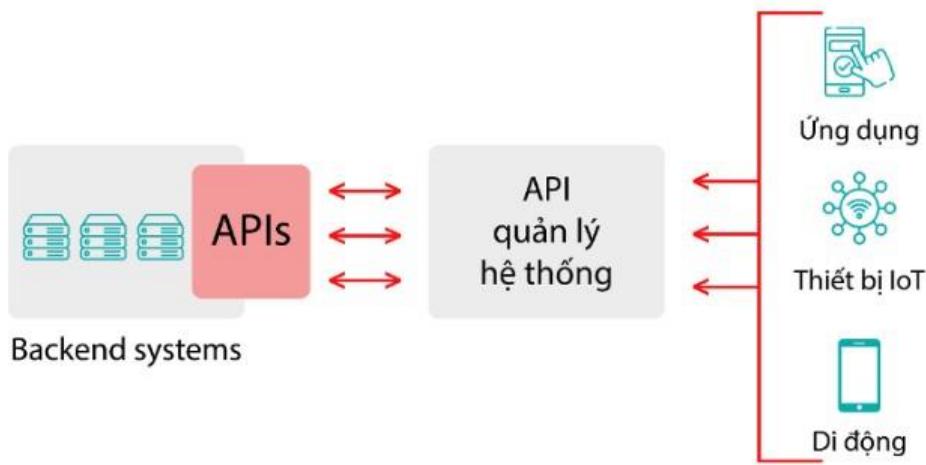
API có thể sử dụng cho web-based-system, operating system, database system, computer hardware hoặc software library. Các doanh nghiệp lớn hầu như đều sử dụng API cho khách hàng hay hệ thống dữ liệu nội bộ.

Để hiểu được bản chất của API, hãy tưởng tượng mình đang ngồi trong một nhà hàng và trước mặt là menu. Nhà bếp là một phần của “hệ thống” nơi mà sẽ chuẩn bị thức ăn để phục vụ. Nhưng để biết những món ăn mà order thì nhà bếp cần người phục vụ - người truyền thông tin giữa và nhà bếp. Trong trường hợp này, người phục vụ chính là API là cầu nối giúp truyền thông tin và phản hồi ngược lại cho khi món ăn sẵn sàng.

1.1.2. API hoạt động ra sao

API giao tiếp thông qua một tập hợp các quy tắc để xác định phương thức mà các máy tính, ứng dụng hoặc máy móc có thể tương tác với nhau. API hoạt động như một người trung gian giữa hai thiết bị bất kỳ muốn kết nối với nhau phục vụ cho một tác vụ được chỉ định.

Ví dụ đơn giản: Khi muốn đăng nhập Facebook thông qua ứng dụng trên điện thoại bằng tài khoản của . Lúc này, ứng dụng Facebook sẽ thực hiện một lệnh tới API để truy xuất tài khoản và thông tin đăng nhập của . Sau đó, Facebook sẽ truy cập thông tin này từ một trong các máy chủ của mình và trả dữ liệu về ứng dụng di động.



Hình 1.1: Cấu trúc api

1.1.3. Các thuật ngữ quan trọng trong API

Request và Response

Request có nghĩa là yêu cầu, phía Client sẽ gửi "request" (yêu cầu) đến phía Server về những mong muốn kết quả của mình. Khi nhận được yêu cầu, phía Server sẽ trả lại kết quả thì đó gọi là Response.

REST

REST hay RESTful là viết tắt của Representational State Transfer là quy tắc về ngôn ngữ giao tiếp giữa người và máy để máy có thể hiểu được những ý định của con người.

1.1.4. Một số đặc tính của REST

Resource-based

Resource-based dịch nghĩa tiếng Việt là dựa trên nguồn dữ liệu. Mỗi một Server mà chúng ta sử dụng API đều là một nguồn.

Ví dụ, xây dựng một trang web và liên kết với API của Tiki để lấy dữ liệu, thì lúc này dữ liệu Tiki là resource của .

Representation

Trong cuộc sống đời thường, khi cần nhắn tin trao đổi bằng các loại ngôn ngữ khác nhau sẽ dùng các phương thức khác nhau như chữ tượng hình hoặc kiểu bảng chữ cái. Đối với máy tính cũng vậy, chúng cũng cần cách thức biểu diễn với các nguyên tắc riêng để có thể giao tiếp với nhau. Hai phương thức phổ biến nhất là JSON và XML.

1.1.5. Các ràng buộc

Uniform interface (Uniform Resource Identifier / Locator, hay URI / URL): Thuật ngữ này có nghĩa cả hai sẽ thống nhất về các thông tin và quyền hạn từ ban đầu, mọi thứ sẽ không thay đổi trong cả quá trình sử dụng. Vậy nên với vai trò là một BA, cần suy nghĩ thấu đáo và lường trước nhu cầu của hệ thống về sự phát triển trong tương lai để đưa ra các Uniform interface từ ban đầu. Cụ thể, 2 tháng trước trang web của cần lấy thông tin về tên và giá sản phẩm tại Shopee, tuy nhiên hiện tại cần thêm mô tả và đánh giá về sản phẩm. Điều này sẽ rất khó khăn cho nhóm của để có thể lấy thêm các trường dữ liệu và thêm nhiều thời gian. Mặt khác, đừng nên tham lam quá khi mong muốn quyền hạn quá nhiều vì nên nhớ một số API cần trả phí khi sử dụng!

Client-server architecture: Đây là sự tuân thủ kiến trúc Client - Server. Vai trò và trách nhiệm của hai bên được quy định từ ban đầu và sẽ không bất kỳ sự phá vỡ nào. Client luôn là người sử dụng tài nguyên và Server luôn phải là người cung cấp, nắm giữ.

Stateless: Từ này khi dịch ra tiếng Việt có vẻ khá trừu tượng, khó hiểu nhưng thực tế là đòi hỏi các thông tin về yêu cầu cần được đưa đầy đủ. Ví dụ, mỗi lần gửi request cho Twitter để lấy thông tin về các bài viết được xuất bản cuối tuần, thì cần thông tin chính xác là thứ bảy hay chủ nhật hay cả hai.

Cacheable: Có thể lưu vào bộ nhớ đệm là ý nghĩa của thuật ngữ trên. Mặc dù Stateless giúp thông tin được chính xác hơn nhưng đồng nghĩa với việc trang web hoặc ứng dụng sẽ mất nhiều thời gian hơn rất nhiều để truy xuất thông tin, dẫn đến việc giảm hiệu suất trang web. Vậy thì bộ nhớ đệm chính là giải pháp cho trường hợp trên, với những request đã được yêu cầu từ trước thì những phản hồi sẽ nhanh hơn.

Layered system: Trong một số trường hợp Client và Server không giao tiếp trực tiếp với nhau mà thông qua các lớp trung gian thì chúng được gọi là layered system. Nhiệm vụ của layered system dùng để “gánh bớt” một lượng cache cho hệ thống, từ đó có thể tăng hiệu suất.

1.1.6. Ưu điểm và nhược điểm của API

API có những ưu điểm nổi bật có thể kể đến như:

Có thể dùng API trên đa dạng các nền tảng của desktop, điện thoại di động mobile hay website.

Với mỗi nền tảng thì API sẽ linh hoạt các định dạng dữ liệu sao cho phù hợp với các ứng dụng trên các hệ điều hành đó.

Người dùng có thể thuận tiện xây dựng HTTP server khi dùng API để việc lập trình trở nên dễ dàng hơn.

API sử dụng các mã nguồn mở với chức năng RESTFUL nên có thể ứng dụng vào bất cứ hệ điều hành nào để hỗ trợ JSON và XML.

Có thể dùng API để trao đổi thông tin 2 chiều, đảm bảo được độ tin cậy cho các bên.

Các thành phần MVC như Model Binder, Action, Unit Test, Controller,... đều có thể được API hỗ trợ đầy đủ các tính năng.



Hình 1.2: Giao tiếp IPA

Bên cạnh những ưu điểm trên thì API vẫn còn một số điểm hạn chế cần khắc phục trong thời gian tới. Cụ thể là:

API Website chưa hỗ trợ tất cả các server, nó chỉ mới hỗ trợ POST và GET mà thôi.

Người dùng cần có hiểu biết kỹ càng và có kinh nghiệm backend thì mới có thể tận dụng triệt để các tính năng mà API mang lại.

Việc phát triển, vận hành hay nâng cấp API là không hề dễ dàng và cần một thời gian dài để thực hiện. Thậm chí, người dùng còn phải trả một chi phí khá lớn cho việc đó.

Nếu chủ sở hữu API chưa giới hạn kỹ các điều kiện thì vẫn có thể bị tấn công hệ thống và không đảm bảo được việc bảo mật.

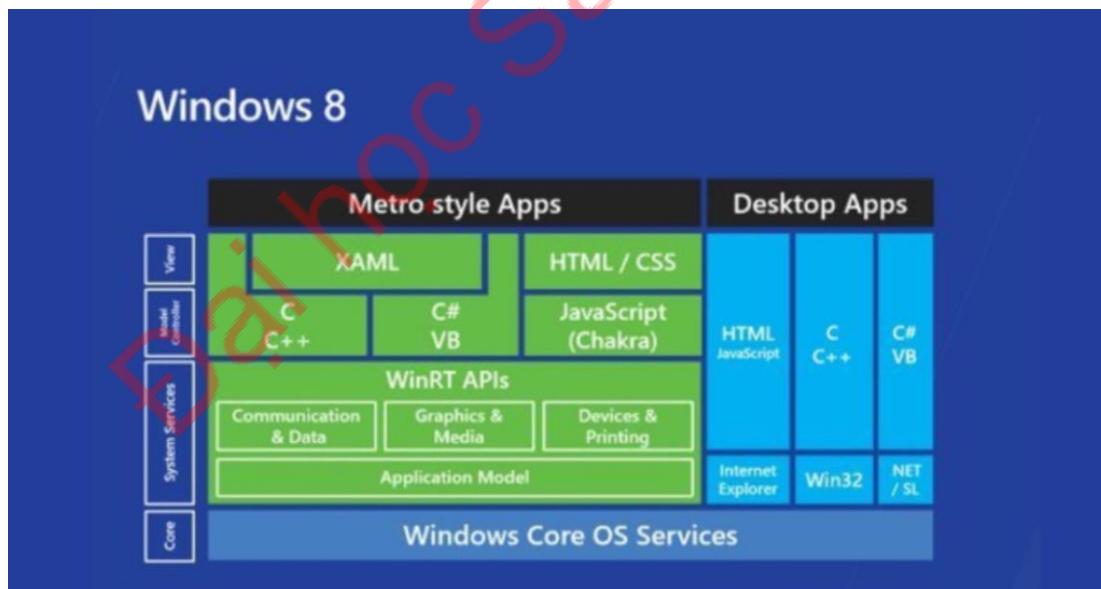
1.1.7. API thường ứng dụng vào đâu

Việc không phụ thuộc vào ngôn ngữ lập trình cố định đã khiến API xử lý gọn nhẹ nhiều tương tác kết nối giữa các giao diện và phần mềm. Do đó, API đã được ứng dụng rộng rãi hiện nay vào các nền tảng sau:

Web API: Đây là ứng dụng thường gặp nhất của API, cho phép người dùng truy cập, lấy dữ liệu, cập nhật cơ sở dữ liệu. Web API là giao diện xử lý tác vụ giữa trình duyệt web và máy chủ web. Chẳng hạn, có thể dễ dàng tạo mới chức năng login thông qua Twitter, Youtube, Facebook, Google,... đồng nghĩa với việc đang truy xuất đến API.

API của framework hay thư viện phần mềm: API quy định và mô tả các tác vụ mà thư viện phần mềm cung cấp. API có thể được viết bằng nhiều ngôn ngữ khác nhau, triển khai hoạt động bằng nhiều cách khác nhau. Chẳng hạn, có thể tạo file PDF bằng ngôn ngữ C++ bằng cách dùng PHP yêu cầu thư viện phần mềm.

API trên hệ điều hành: Hệ điều hành Linux hay Windows sở hữu đa dạng các API khác nhau, cung cấp cơ sở dữ liệu cho API để mô tả các hàm và cách thức giao tiếp thường dùng. Từ đó, góp phần hỗ trợ lập trình viên tạo ra các ứng dụng phần mềm kết nối và tương tác trực tiếp với hệ điều hành.



Hình 1.3: Khả năng tương tác của API

1.1.8. Một số cách để ngăn chặn rò rỉ thông tin khi dùng API

Xác thực cẩn thận thẩm quyền của người dùng cũng như ứng dụng

cần xác thực kỹ càng xem ai là người cuối cùng sử dụng cũng như sự tin cậy của ứng dụng. Chẳng hạn như khi dùng các ứng dụng như Uber hay AirBnB thì chúng sẽ thực hiện quy trình gọi API. Khi đó thì ứng dụng sẽ được xác thực chính xác hơn.

Mã hóa các dữ liệu sẽ được truyền tải từ các nền tảng

có thể dùng chứng chỉ SSL để thực hiện việc mã hóa này. Khi các dữ liệu đã được mã hóa, sẽ ngăn chặn được các tình trạng như bị đột nhập, tấn công và ăn cắp các dữ liệu quan trọng trong quá trình truyền tải thông tin.

Hạn chế đặt mật khẩu cố định hoặc quá dễ đoán

Nhiều người vì lười biếng hoặc vì hay quên nên thường chọn mật khẩu cố định hoặc lấy những mật khẩu được lưu cục bộ ở trên các ứng dụng trong điện thoại di động của mình. Thói quen này có thể sẽ làm cho dễ dàng bị đánh cắp thông tin. Vì thế, hãy đặt một mật khẩu có độ khó và không phổ biến nhé.

Dùng chữ ký số để bảo mật

Chữ ký số là chuỗi ký tự đặc trưng và duy nhất của một người dùng. Nó sẽ giúp lưu trữ các chuỗi này ở trong cơ sở dữ liệu và chỉ khi nhập đúng tên và mật khẩu thì mới được cung cấp chữ ký số.

Như vậy, bài viết này đã giúp biết được API là gì và cách thức vận hành của nó. API là một công cụ quan trọng mà Chúng em nghĩ rằng bất cứ ngành nghề hay doanh nghiệp nào cũng nên sử dụng. Bởi vì nhờ nó mà doanh nghiệp của sẽ có cơ hội để mở rộng quy mô phát triển, tiếp cận được nhiều khách hàng hơn.

1.2. Tổng quan về ngôn ngữ lập trình C#

Ngôn ngữ lập trình C# hay còn được gọi là C Sharp là một ngôn ngữ lập trình thuần hướng đối tượng rất đa năng, mạnh mẽ được rất nhiều lập trình viên tin dùng.

Được ra đời vào năm 2000, thiết kế bởi Anders Hejlsberg rất nổi tiếng với việc tạo ra các sản phẩm Turbo Pascal, Delphi, J++, WFC và do Microsoft phát triển thì qua nhiều năm phát triển cùng nâng cấp C# đang có phiên bản mới nhất là 9.0, được phát hành vào năm 2020 cùng với Visual Studio 2019 phiên bản 16.8.

C# được phát triển dựa trên ngôn ngữ lập trình C++ và ngôn ngữ lập trình java. Do đó C# có được những ưu điểm của hai ngôn ngữ trên đồng thời khắc phục những hạn chế của hai ngôn ngữ nền tảng của nó.

Cái tên C# của ngôn ngữ lập trình này bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường.

1.2.1. Ngôn ngữ lập trình C# dùng để làm gì

Ngôn ngữ lập trình C# là một ngôn ngữ rất đa dụng và mạnh mẽ, tùy theo ý muốn của lập trình viên ngôn ngữ này có thể làm được những công việc sau đây:

Phát triển web backend (ASP.NET MVC, ASP.NET core, Web API, Graph API, gRPC, Blazor server, Uno platform, Mono)

Phát triển web front end (Blazor WebAssembly, Uno platform)

Phát triển desktop app (Winform, WPF, UWP, WinUI, Mono, Uno, MAUI, Blazor desktop...)

Phát triển game 2D, 3D đa nền tảng (Game engine: Unity, Monogame, Godot, Stride, CryEngine, Flax Engine, Evergine, NeoAxis, XNA ...)

Phát triển thực tế ảo (VR), thực tế tăng cường (AR), thực tế hỗn hợp (MR) (HoloLens, Unity, CryEngine, Oculus quest..)

Phát triển ứng dụng đồ họa 2D, 3D đa nền tảng (2D: SkiaSharp, ImageSharp...; 3D: OpenTK, SharpDX, SharpVulkan, Vulkan.NET, Veldrid, Silk.NET, Helix Toolkit, Aspose..)

Phát triển mobile app, IOS native, Android native (Xamarin, MAUI, Uno platform)

Phát triển đám mây (Azure, AWS, Google Cloud...)

Học máy và trí tuệ nhân tạo (ML.NET, TensorFlow, csiSharp..)

Data science, bigdata (csiSharp, Apache Spark)

Blockchain (NEO, Stratis) Microservices and containers Internet of thing (IoT, 5G)

Hệ thống nhúng (Raspberry pi, PLC)

1.2.2. Đặc trưng của ngôn ngữ C#

C# là một ngôn ngữ lập trình mạnh mẽ và linh hoạt bao gồm một loạt các tính năng của ngôn ngữ hiện đại.

Một trong những đặc trưng nổi bật của C#, bao gồm:

Lập trình hướng đối tượng: C# là ngôn ngữ lập trình hướng đối tượng, có nghĩa là nó được thiết kế dựa trên khái niệm Object. Objects là các trường hợp của Classes, được sử dụng để định nghĩa hành vi và thuộc tính của một object.

Garbage Collection: C# bao gồm quản lý bộ nhớ tự động, có nghĩa là nó sẽ tự động giải phóng bộ nhớ khi không cần thiết. Điều này giúp giảm nguy cơ rò rỉ bộ nhớ và giúp viết mã an toàn và đáng tin cậy dễ dàng hơn.

Type Safety: C# là ngôn ngữ theo kiểu an toàn, có nghĩa là nó ngăn nhà phát triển mắc các lỗi lập trình phổ biến, chẳng hạn như ngoại lệ con trỏ null. Nhờ đó, khiến mã trở nên mạnh mẽ và đáng tin cậy hơn.

Lập trình không đồng bộ: C# hỗ trợ cả lập trình không đồng bộ, cho phép nhà phát triển viết mã có thể thực thi các tác vụ song song. Điều này giúp cải thiện hiệu suất của các ứng dụng cần thực hiện đồng thời nhiều tác vụ.

LINQ: C# bao gồm Truy vấn Tích hợp Ngôn ngữ (Language-Integrated Query - LINQ), đây là một công cụ mạnh hỗ trợ quá trình truy vấn và thao tác dữ liệu. LINQ cho phép nhà phát triển viết các truy vấn theo một cú pháp quen thuộc, giúp dễ dàng làm việc với dữ liệu trong một ứng dụng hơn.

Delegates và Events: Với tính năng này cho phép nhà phát triển viết mã có thể phản hồi các sự kiện, chẳng hạn như input của người dùng hoặc thay đổi dữ liệu. Góp phần tạo ra nhiều ứng dụng tương tác và đáp ứng hơn.

Iterators: C# cũng hỗ trợ cho trình vòng lặp, cho phép nhà phát triển viết mã có thể lặp qua một tập hợp các đối tượng. Từ đó, giúp mã trở nên ngắn gọn và dễ đọc hơn.

Nói ngắn gọn, thì C# là ngôn ngữ lập trình hiện đại và mạnh mẽ với nhiều tính năng giúp dễ dàng viết mã an toàn, đáng tin cậy và hiệu quả.

1.2.3. Mục tiêu khi phát triển C#

Khi Microsoft phát triển C#, họ đã nghĩ đến một số mục tiêu chính. Những mục tiêu có thể kể đến như:

Tính đơn giản: Microsoft mong muốn tạo ra một ngôn ngữ lập trình dễ học và dễ sử dụng. Nên C# được thiết kế tương tự như các ngôn ngữ lập trình phổ biến khác, chẳng hạn như C++ và Java, giúp các nhà phát triển dễ dàng chuyển đổi sang C#.

Năng suất: Microsoft muốn tạo ra một ngôn ngữ lập trình giúp các nhà phát triển viết mã nhanh hơn mà vẫn đảm bảo tính hiệu quả. C# bao gồm nhiều tính năng giúp dễ dàng viết mã nhanh chóng, chẳng hạn như quản lý bộ nhớ tự động, an toàn kiểu và hỗ trợ cho các mô hình lập trình hiện đại.

Tính linh hoạt: Microsoft muốn tạo ra một ngôn ngữ lập trình có thể được sử dụng để phát triển nhiều loại ứng dụng, bao gồm desktop applications, web applications, mobile applications và game. C# được thiết kế để trở thành một ngôn ngữ linh hoạt có thể được sử dụng cho nhiều loại ứng dụng khác nhau.

Khả năng tương tác: Microsoft muốn tạo ra một ngôn ngữ lập trình có thể được sử dụng để phát triển các ứng dụng chạy trên nhiều nền tảng. C# được thiết kế để trở thành ngôn ngữ nền tảng trung lập có thể được sử dụng để phát triển ứng dụng cho Windows, Linux và macOS.

Bảo mật: Microsoft muốn tạo ra một ngôn ngữ lập trình giúp các nhà phát triển viết mã an toàn hơn. C# bao gồm nhiều tính năng giúp ngăn ngừa các lỗ hổng bảo mật phổ biến, chẳng hạn như lỗi tràn bộ đệm và ngoại lệ con trả null.

1.2.4. Tính ứng dụng của ngôn ngữ C#

C# là một ngôn ngữ lập trình linh hoạt có thể được sử dụng trong quá trình xây dựng và phát triển của nhiều loại ứng dụng.

Một số ứng dụng phổ biến của C# bao gồm:

Phát triển desktop application

C# có thể được sử dụng để tạo desktop applications trên Windows bằng cách sử dụng Windows Presentation Foundation (WPF) hoặc Windows Forms.

Các ứng dụng này có giao diện người dùng phong phú và quyền truy cập vào nhiều loại tài nguyên hệ thống.

Phát triển web

C# có thể được sử dụng để tạo các ứng dụng web bằng ASP.NET Framework.

ASP.NET cho phép các nhà phát triển tạo các ứng dụng web động với khả năng xử lý phía máy chủ và kết nối cơ sở dữ liệu.

Phát triển game

C# được sử dụng trong các công cụ phát triển trò chơi như Unity, cho phép các nhà phát triển tạo trò chơi cho nhiều nền tảng bao gồm máy tính để bàn, thiết bị di động và bảng điều khiển.

Phát triển mobile application

C# có thể được sử dụng để phát triển các ứng dụng mobile cho Android và iOS với sự hỗ trợ của Framework Xamarin.

AI và machine learning

C# có thể được sử dụng trong các ứng dụng AI và machine learning thông qua Frameworks như ML.NET, cung cấp các công cụ để xây dựng và đào tạo các mô hình học máy.

Database programming

C# có thể được sử dụng để kết nối và thao tác với cơ sở dữ liệu bằng cách sử dụng Frameworks như Entity Framework hoặc ADO.NET.

Internet of Things (IoT)

C# có thể được sử dụng để phát triển các ứng dụng IoT, cũng như có khả năng tương tác với các thiết bị IoT bằng cách sử dụng Frameworks như .NET Core và Azure IoT.



Hình 1.4: Ngôn ngữ C#

1.3. Sql server

1.3.1. SQL là gì

SQL là loại ngôn ngữ máy tính, giúp cho thao tác lưu trữ và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. SQL là viết tắt của Structured Query Language là ngôn ngữ truy vấn có cấu trúc



Hình 1.5: Sql server

Tất cả RDBMS (hệ thống quản lý cơ sở dữ liệu quan hệ) như MySQL, Oracle, MS Access, Sybase, Informix, Postgres và SQL Server đều sử dụng SQL như là ngôn ngữ cơ sở dữ liệu chuẩn.

1.3.2. Tại sao sử dụng SQL

Thế mạnh của ngôn ngữ SQL phát huy trong nhiều trường hợp sử dụng khác nhau.

Ngoài việc là một ngôn ngữ máy tính phổ biến, SQL còn là ngôn ngữ máy tính hữu dụng. Vì vậy, người ta thường sử dụng SQL cho các mục đích:

Tạo cơ sở dữ liệu, bảng và view mới.

Để chèn các bản ghi vào trong một cơ sở dữ liệu.

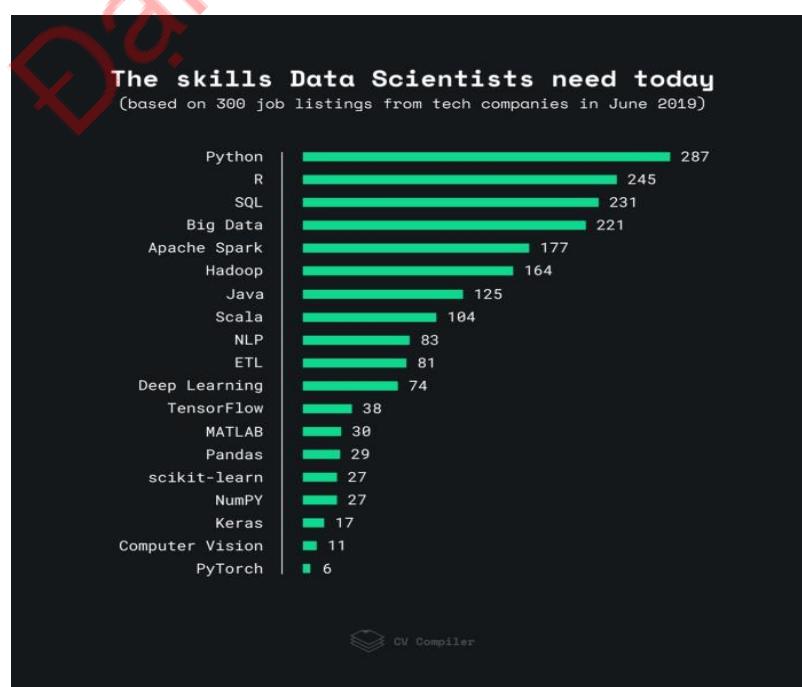
Để xóa các bản ghi từ một cơ sở dữ liệu.

Để lấy dữ liệu từ một cơ sở dữ liệu.

Hiện nay hầu hết các website đều sử dụng MySQL làm cơ sở quản lý dữ liệu cho website trên hosting. Việc hiểu biết về ngôn ngữ lập trình SQL sẽ giúp thao tác dữ liệu tốt hơn mà không cần nhờ đến hỗ trợ từ các dịch vụ cho thuê hosting.

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của Relational Database Management System. MySQL được tích hợp apache, PHP. MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL.

Hơn nữa, SQL còn được liệt kê là một trong những kỹ năng hàng đầu mà Data Scientist cần sở hữu. Một trong những ưu điểm chính của việc sử dụng SQL là khi thực hiện các thao tác với dữ liệu thì dữ liệu được truy cập trực tiếp (không cần sao chép trước). Điều này có thể tăng tốc đáng kể việc thực thi quy trình làm việc.



Hình 1.6: hình ảnh minh họa

1.3.3. Tóm tắt lịch sử của SQL

Đến nay, SQL đã hình thành và phát triển hơn 40 năm.

Nhìn chung, để phát triển được như ngày nay, SQL đã phải trải qua giai đoạn hình thành và thay đổi suốt nhiều năm:

1970 – Tiến sĩ Edgar F. “Ted” Codd của IBM được biết đến như là cha đẻ của các cơ sở dữ liệu quan hệ mô tả một mô hình quan hệ cho cơ sở dữ liệu trong quyển sách của mình.

1974 – Structured Query Language đầu tiên được phát triển.

1978 – IBM đã làm việc để phát triển các ý tưởng của Codd và đưa ra một sản phẩm có tên SystChung em / R. Cùng thời điểm đó Relational Software, Inc cũng phát triển cơ sở dữ liệu của riêng họ.

1979 – Relational Software, Inc. giới thiệu Oracle V2 và nhận được nhiều sự chú ý từ công chúng.

1986 – IBM đã phát triển nguyên mẫu đầu tiên của cơ sở dữ liệu quan hệ và được tiêu chuẩn hóa bởi ANSI. Cơ sở dữ liệu quan hệ đầu tiên được phát hành bởi Relational Software sau này được gọi là Oracle.

Chức năng của SQL là gì?



Hình 1.7: Chức năng của SQL

SQL cho phép người dùng thực hiện đa dạng các chức năng

Một trong những lý do khiến cho SQL được sử dụng phổ biến, chính là nó đã cho phép người dùng thực hiện đa dạng các chức năng sau:

Cho phép người dùng truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.

Cho phép người dùng mô tả dữ liệu.

Cho phép người dùng xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.

Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện và trình biên dịch trước.

Cho phép người dùng tạo và thay đổi các cơ sở dữ liệu và bảng.

Cho phép người dùng tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.

Cho phép người dùng thiết lập quyền trên các bảng, thủ tục và view.

1.3.4. *Ưu điểm của SQL*

Ưu điểm của SQL giúp nó an toàn duy trì đến hôm nay.

Tại sao cần sử dụng SQL? Với SQL, người dùng thường như khai thác được khá nhiều lợi ích. Dưới đây là một số ưu điểm của SQL:

Dữ liệu có ở mọi nơi: Dữ liệu xuất hiện ở mọi nơi trên màn hình từ laptop đến điện thoại của . Việc học tập và tìm hiểu SQL sẽ giúp biết được cách thức hoạt động của những dữ liệu này.

Thêm, sửa, đọc và xóa dữ liệu dễ dàng: với SQL, các thao tác xử lý dữ liệu trở nên dễ dàng hơn bao giờ hết. chỉ cần thực hiện một số thao tác với dữ liệu đơn giản trên SQL thay vì phải dùng nhiều câu lệnh phức tạp trên các loại ngôn ngữ khác.

SQL giúp công việc lập trình dễ dàng hơn: có thể lưu nhiều dữ liệu cho nhiều ứng dụng khác nhau trên cùng một cơ sở dữ liệu và việc truy cập các cơ sở dữ liệu này trở lên đơn giản hơn nhờ một cách thức giống nhau.

Được sử dụng và hỗ trợ bởi nhiều công ty lớn: tất cả các công ty lớn về công nghệ trên thế giới hiện nay như Microsoft, IBM, Oracle... đều hỗ trợ việc phát triển ngôn ngữ SQL.

Lịch sử hơn 40 năm: với lịch sử phát triển hơn 40 năm từ 1970, SQL vẫn tồn tại và trụ vững đến ngày nay. Điều này cho thấy vị trí của SQL hiện tại rất khó bị thay thế bởi bất kỳ một ngôn ngữ máy tính nào khác.

1.3.5. Nhược điểm của SQL

Một số phiên bản SQL đòi hỏi chi phí vận hành khá cao gây khó khăn cho người lập trình

Mặc dù là ngôn ngữ được sử dụng phổ biến, thế nhưng, không một ngôn ngữ nào là hoàn hảo. Vì vậy, SQL cũng tồn tại một số hạn chế dưới đây:

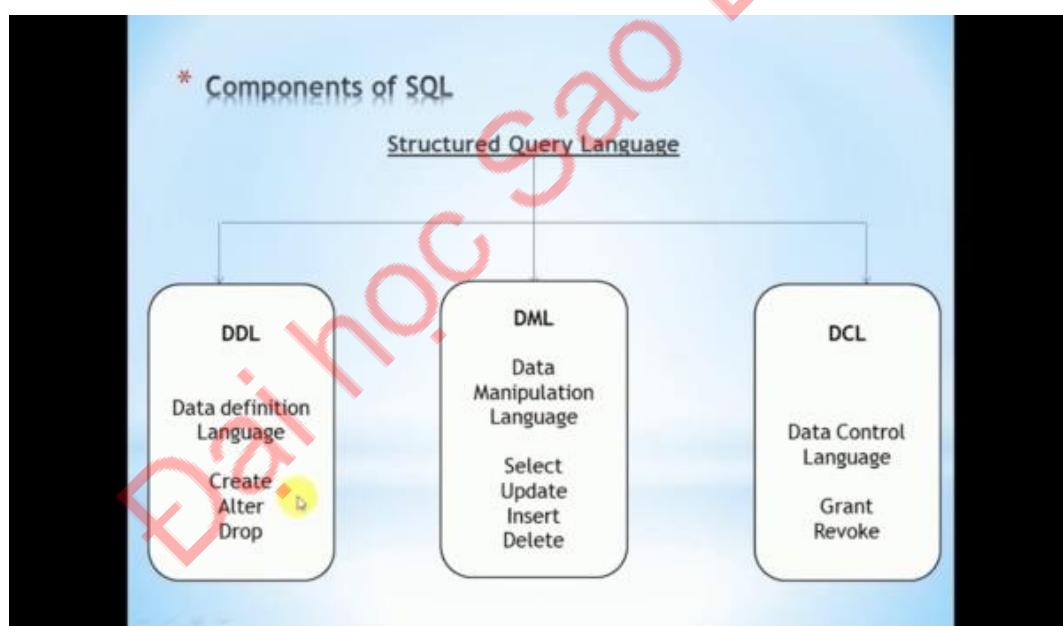
Giao diện khó dùng: SQL có giao diện phức tạp khiến một số người dùng khó truy cập.

Không được toàn quyền kiểm soát: Các lập trình viên sử dụng SQL không có toàn quyền kiểm soát cơ sở dữ liệu.

Thực thi: Hầu hết các chương trình cơ sở dữ liệu SQL đều có phần mở rộng độc quyền riêng.

Giá cả: Chi phí vận hành của một số phiên bản SQL khá cao khiến một số lập trình viên gặp khó khăn khi tiếp cận.

1.3.6. Phân loại câu lệnh SQL



Hình 1.8: Nhóm SQL

SQL có ba nhóm lệnh phổ biến

SQL DDL (Data Definition Language) là gì?

DDL là ngôn ngữ định nghĩa dữ liệu. Các lệnh DDL bao gồm:

- Lệnh CREATE: Tạo một bảng, một View của bảng, hoặc đối tượng khác trong Database.
- Lệnh ALTER: Sửa đổi một đối tượng Database đang tồn tại, ví dụ như một bảng.

- Lệnh DROP: Xóa toàn bộ một bảng, một View của bảng hoặc đối tượng khác trong một Database.
- SQL DML (Data Manipulation Language) là gì?
- DML là ngôn ngữ thao tác dữ liệu. Các lệnh DML bao gồm:
- Lệnh SELECT: Lấy các bản ghi cụ thể từ một hoặc nhiều bảng.
- Lệnh INSERT: Tạo một bản ghi.
- Lệnh UPDATE: Sửa đổi các bản ghi.
- Lệnh DELETE: Xóa các bản ghi.
- SQL DCL (Data Control Language) là gì?
- DCL là ngôn ngữ điều khiển dữ liệu. Các lệnh DCL bao gồm:
- Lệnh GRANT: Trao một quyền tới người dùng.
- Lệnh REVOKE: Thu hồi quyền đã trao cho người dùng.

Các câu lệnh SQL phổ biến

Trong SQL có khá nhiều câu lệnh, mệnh đề khác nhau. Tuy nhiên dưới đây là một số những truy vấn thường gặp nhất. Chúng ta sẽ cùng tìm hiểu chi tiết ngay bên dưới:

- Truy vấn Update
- Truy vấn Insert
- Truy vấn Select
- Mệnh đề Order By
- Mệnh đề Group By
- Từ khóa Distinct
- Mệnh đề Where
- Mệnh đề AND & OR

SQL có nhiều câu lệnh thể hiện sự cho phép thao tác đa dạng

Truy vấn Update trong SQL là gì?

Truy vấn UPDATE trong SQL được sử dụng để sửa đổi các bản ghi đang tồn tại trong một bảng. có thể sử dụng mệnh đề WHERE với truy vấn UPDATE sửa đổi các hàng đã lựa chọn, nếu không, hệ thống sẽ mặc định là tất cả các hàng đều bị tác động.

Cú pháp:

```
UPDATE          ten_bang  
SET    cot1    =    giatri1,    cot2    =    giatri2....,    cotN    =    giatriN  
WHERE [dieu_kien];
```

Truy vấn Insert trong SQL là gì?

Lệnh INSERT INTO trong SQL được sử dụng để thêm các hàng dữ liệu mới vào một bảng trong Database.

Cú pháp:

Xác định cột để chèn dữ liệu:

```
INSERT INTO TABLE_TEN (cot1, cot2, cot3,...cotN)]  
VALUES (giatri1, giatri2, giatri3,...giatriN);
```

Thao tác trên tất cả các cột trong bảng

```
INSERT INTO TABLE_TEN VALUES (giatri1,giatri2,giatri3,...giatriN);
```

Truy vấn Select trong SQL là gì?

Lệnh SELECT trong SQL được sử dụng để lấy dữ liệu từ một bảng trong Database mà trả về dữ liệu ở dạng bảng dữ liệu kết quả.

Cú pháp:

Xác định cột có giá trị muốn lấy:

```
SELECT cot1, cot2, cotN FROM ten_bang;
```

Thao tác trên tất cả các cột trong bảng:

```
SELECT * FROM ten_bang;
```

Mệnh đề Order By

Mệnh đề ORDER BY trong SQL được sử dụng để sắp xếp dữ liệu theo thứ tự tăng dần hoặc theo thứ tự giảm dần, trên một hoặc nhiều cột.

Cú pháp:

```
SELECT danh_sach_cot  
FROM ten_bang  
[WHERE dieu_kien]  
[ORDER BY cot1, cot2, .. cotN] [ASC | DESC];
```

Mệnh đề Group By

Mệnh đề GROUP BY trong SQL được sử dụng kết hợp với lệnh SELECT để sắp xếp dữ liệu đồng nhất vào trong các nhóm.

Cú pháp:

```
SELECT cot1, cot2  
FROM ten_bang  
WHERE [ dieu_kien ]  
GROUP BY cot1, cot2  
ORDER BY cot1, cot2
```

Từ khóa Distinct

Mệnh đề DISTINCT trong SQL được sử dụng kết hợp với lệnh SELECT để loại tất cả các bản sao của bản ghi và chỉ lấy các bản ghi duy nhất.

Cú pháp:

```
SELECT DISTINCT cot1, cot2,.....cotN
```

```
FROM ten_bang
```

```
WHERE [dieu_kien]
```

Mệnh đề Where

Mệnh đề WHERE trong SQL được sử dụng để xác định một điều kiện trong khi lấy dữ liệu từ bảng đơn hoặc nhiều bảng kết hợp. sẽ sử dụng mệnh đề WHERE để lọc các bản ghi và chỉ lấy lại các bản ghi cần thiết.

Cú pháp:

```
SELECT cot1, cot2, cotN
```

```
FROM ten_bang
```

```
WHERE [dieu_kien]
```

Mệnh đề AND & OR

Các toán tử AND và OR trong SQL được sử dụng để kết hợp nhiều điều kiện để thu hẹp phạm vi dữ liệu trong một lệnh SQL. Hai loại toán tử này được gọi là toán tử liên hợp. Nó cho phép tồn tại nhiều điều kiện trong mệnh đề WHERE của một lệnh SQL.

Cú pháp:

```
SELECT cot1, cot2, cotN
```

```
FROM ten_bang
```

```
WHERE [dieu_kien_1] AND [dieu_kien_2]...AND [dieu_kien_N];
```

1.4. Công cụ sử dụng

1.4.1. Visual studio

Visual Studio 2019 là một môi trường phát triển tích hợp (IDE - Integrated Development Environment) mạnh mẽ của Microsoft, được thiết kế để hỗ trợ nhiều ngôn ngữ lập trình và nền tảng phát triển khác nhau. Dưới đây là một số điểm nổi bật về công cụ lập trình Visual Studio 2019:

Đa Nền Tảng: Visual Studio 2019 hỗ trợ phát triển ứng dụng trên nhiều nền tảng như Windows, Android, iOS, Linux, và nền tảng web. Điều này cho phép lập trình viên phát triển ứng dụng đa nền tảng một cách thuận tiện.

Ngôn Ngữ Lập Trình Đa Dạng: Visual Studio 2019 hỗ trợ nhiều ngôn ngữ lập trình như C#, C++, Visual Basic, F#, Python, và nhiều ngôn ngữ khác. Điều này làm cho nó trở thành một lựa chọn phổ biến cho nhiều dự án phát triển.

Công Cụ Lập Trình .NET: Visual Studio 2019 là một công cụ mạnh mẽ cho phát triển ứng dụng dựa trên nền tảng .NET. Nó cung cấp môi trường lập trình cho .NET Core và .NET Framework.

Debugging và Profiling: IDE này có các công cụ mạnh mẽ để debug và profile ứng dụng. Người phát triển có thể theo dõi và phân tích mã nguồn của họ để tìm ra lỗi và tối ưu hiệu suất.

IntelliCode: Visual Studio 2019 tích hợp tính năng IntelliCode, sử dụng trí tuệ nhân tạo để đề xuất mã nguồn dựa trên các mô hình học máy được huấn luyện trên hàng triệu dòng mã nguồn từ cộng đồng lập trình viên.

Hỗ Trợ Git: Visual Studio 2019 tích hợp tính năng quản lý mã nguồn và sử dụng hệ thống kiểm soát phiên bản Git, giúp người phát triển làm việc nhóm một cách hiệu quả.

Azure DevOps: Nó tích hợp chặt chẽ với dịch vụ Azure DevOps của Microsoft, giúp quản lý và triển khai dự án một cách dễ dàng.

Visual Studio Marketplace: Cung cấp một cửa hàng trực tuyến, nơi có thể tìm kiếm và cài đặt các tiện ích mở rộng, mẫu mã, và công cụ khác nhau để mở rộng khả năng của môi trường lập trình.

Visual Studio 2019 mang lại nhiều công cụ và tính năng giúp người phát triển tối ưu hóa quá trình phát triển phần mềm và tăng cường hiệu suất.

1.4.2. Dev express

DevExpress là một tập đoàn phần mềm có trụ sở tại Hoa Kỳ, chuyên cung cấp các công cụ phát triển ứng dụng cho nhiều nền tảng khác nhau như .NET, HTML5, và nhiều hơn nữa. Công ty này nổi tiếng với các sản phẩm chất lượng cao và được sử dụng rộng rãi trong ngành công nghiệp phần mềm. DevExpress cung cấp các thành phần giao diện người dùng (UI), báo cáo, lập trình dữ liệu, và nhiều công cụ khác giúp nhà phát triển thiết kế và xây dựng ứng dụng một cách nhanh chóng và dễ dàng.

Xây dựng giao diện người dùng chuyên nghiệp

DevExpress cung cấp các thành phần giao diện người dùng (UI) giúp tạo ra giao diện đẹp mắt và chuyên nghiệp cho ứng dụng của mình. Với DevExpress, có thể tạo ra các điều khiển tùy chỉnh, biểu đồ, lưới dữ liệu, và nhiều thành phần UI khác để tăng trải nghiệm người dùng và nâng cao hiệu suất ứng dụng.

Tối ưu hóa hiệu suất ứng dụng

Với các công cụ DevExpress, có thể tối ưu hiệu suất ứng dụng của mình để đảm bảo hoạt động mượt mà và đáp ứng nhanh chóng cho người dùng. DevExpress cung cấp các công cụ giám sát hiệu suất, tối ưu hóa mã nguồn, và phân tích tải để có thể nâng cao hiệu suất ứng dụng một cách dễ dàng.

Xây dựng báo cáo chuyên nghiệp

DevExpress cung cấp các công cụ mạnh mẽ để thiết kế và xây dựng các báo cáo chuyên nghiệp cho ứng dụng của . có thể tạo ra các báo cáo đa dạng với đồ thị, biểu đồ, và các định dạng tùy chỉnh để hiển thị dữ liệu một cách rõ ràng và dễ hiểu.

1.4.3. *Sql server*

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ được phát triển bởi Microsoft. Dưới đây là một tổng quan về SQL Server:

Kiến Trúc và Tính Năng Chính

Kiến Trúc Đa Tầng: SQL Server sử dụng kiến trúc đa tầng, bao gồm các thành phần như Database Engine, Analysis Services, Reporting Services, và Integration Services.

Database Engine: Chịu trách nhiệm quản lý cơ sở dữ liệu, bao gồm việc lưu trữ, truy xuất dữ liệu, và thực hiện các tác vụ quản lý.

Analysis Services: Cung cấp các dịch vụ phân tích và khám phá dữ liệu.

Reporting Services: Cho phép tạo và quản lý báo cáo.

Integration Services: Dùng để tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ngôn Ngữ Truy Vấn

SQL Server sử dụng ngôn ngữ truy vấn Transact-SQL (T-SQL) để thực hiện các thao tác truy vấn dữ liệu.

Quản Lý Bảo Mật và Quyền Truy Cập

SQL Server cung cấp các tính năng bảo mật như quản lý quyền truy cập, xác thực Windows, và kỹ thuật mã hóa dữ liệu.

Khả Năng Mở Rộng và Hiệu Suất

SQL Server hỗ trợ các tính năng như partitioning, indexing, và tối ưu hóa truy vấn để đảm bảo hiệu suất tốt đối với cơ sở dữ liệu lớn.

Hỗ Trợ Các Chuẩn và Giao Thức

SQL Server hỗ trợ nhiều chuẩn và giao thức như SQL, ODBC, JDBC, XML, và ADO.NET.

Công Cụ Quản Lý và Giám Sát

SQL Server Management Studio (SSMS) là công cụ chính để quản lý cơ sở dữ liệu và thực hiện các tác vụ quản trị.

SQL Server Profiler cung cấp công cụ giám sát và xác định xem xét các hoạt động trong cơ sở dữ liệu.

Dự Án Lớn và Doanh Nghiệp

SQL Server thường được sử dụng cho các dự án lớn và hệ thống doanh nghiệp, với khả năng xử lý cơ sở dữ liệu lớn và đáp ứng yêu cầu về hiệu suất và bảo mật.

Hỗ Trợ Tích Hợp với Công Nghệ Microsoft Khác

SQL Server tích hợp chặt chẽ với các sản phẩm và dịch vụ khác của Microsoft như Azure, Power BI, và Visual Studio.

Phiên Bản SQL Server

SQL Server có nhiều phiên bản khác nhau, từ phiên bản Express (miễn phí) cho đến các phiên bản Enterprise với nhiều tính năng cao cấp.

SQL Server là một hệ quản trị cơ sở dữ liệu đầy đủ tính năng, linh hoạt, và mạnh mẽ, được sử dụng rộng rãi trên toàn cầu cho các ứng dụng doanh nghiệp và dự án phần mềm lớn.

Chương 2: Kỹ thuật tương tác với facebook

2.1. Api facebook

Facebook API là nền tảng do Facebook cung cấp cho người viết ứng dụng để dễ dàng trong việc tạo ứng dụng và đảm bảo người viết ứng dụng không can thiệp quá sâu vào hệ thống của Facebook.

Thông qua Facebook API, ta có thể lấy được thông tin về người dùng như là thông tin cá nhân, ảnh profile.... nếu như được cấp quyền cho ta truy cập trang cá nhân.

Facebook sẽ gửi một phương thức POST đến máy chủ Facebook API. Nó bao gồm một số các thông số yêu cầu như api_key của ứng dụng, session_key của người dùng đưa ra yêu cầu. Bên cạnh đó Facebook còn thêm vào tham số fb_sig để thông báo ứng dụng đưa ra yêu cầu. Bằng cách này tất cả các lời gọi API sẽ được đảm bảo, Facebook có thể xác minh các yêu cầu được gửi từ một ứng dụng đã được chấp thuận.

2.1.1. Facebook Graph API

Định Nghĩa

Từ Graph dịch ra có nghĩa là từ đồ thị cho nên Facebook Graph API là cách để lấy dữ liệu vào và ra khỏi đồ thị xã hội của Facebook. có thể sử dụng để truy vấn dữ liệu, gửi những câu chuyện mới, tải lên hình ảnh và một loạt các nhiệm vụ khác mà một ứng dụng có thể làm.

Nó bao gồm:

node (nút): Những thứ xuất hiện trên facebook có tương tác như ảnh, post...

edge (cạnh): những kết nối, liên hệ giữa nó.

field (trường): thông tin của nó.

Graph API là dựa trên HTTP, do đó, làm việc với bất kỳ ngôn ngữ nào có một thư viện HTTP, như cURL, urllib - sẽ lấy dữ liệu thông qua các http request.

Truy cập giao diện Graph API của Facebook

Đầu tiên truy cập vào trang <https://developers.facebook.com/>

Sau đó phải tạo 1 ứng dụng facebook, nếu không sẽ không thể sử dụng api. (cách tạo thì các google nhé).

Vào More -> Tool -> Graph API Explorer các sẽ ra được giao diện bên dưới.

Các thành phần chính

Nơi cấp quyền truy cập vào dữ liệu của , có thể quy định quyền truy cập của API

Tạo ra Access_Token để truy cập vào API

Nơi để gửi request

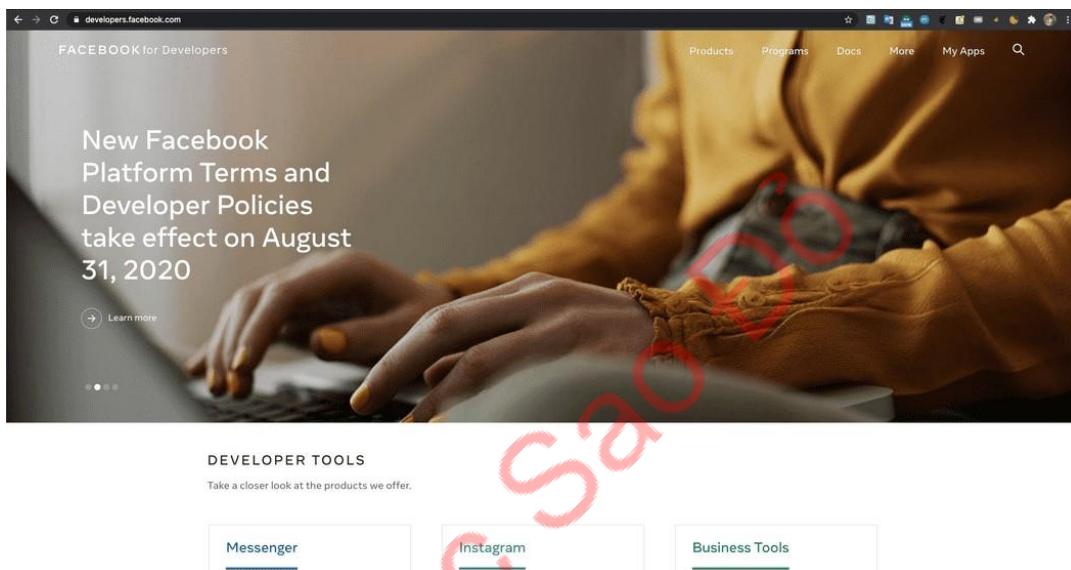
Nhập vào các thông tin mà muốn lấy ra

Nơi hiển thị dữ liệu (dữ liệu được hiển thị dưới dạng json).

2.1.2. Đăng ký ứng dụng graph api

Để có thể lấy được dữ liệu người dùng của Facebook hay Instagram, thì điều đầu tiên chúng ta bắt buộc phải đăng ký app của Graph Facebook.

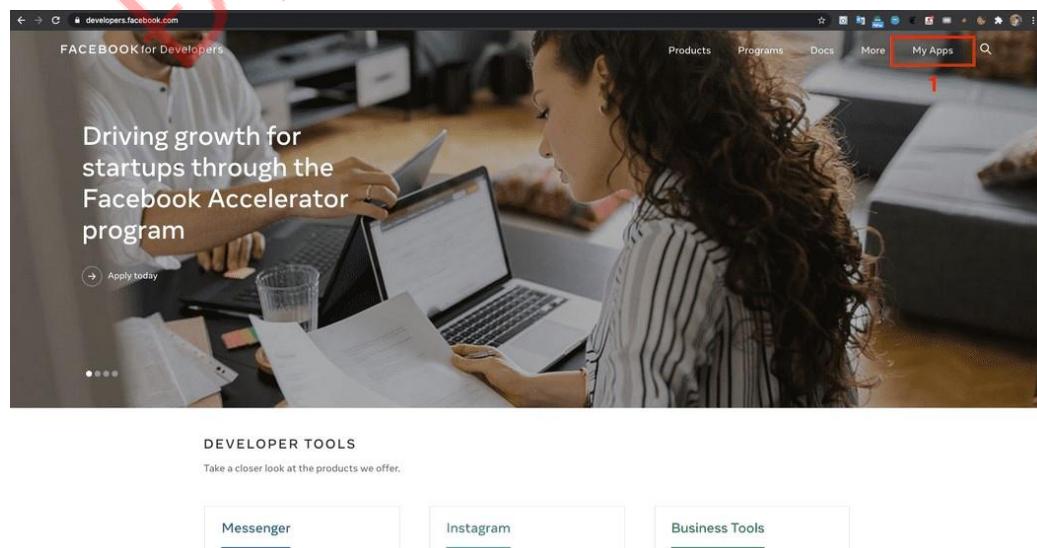
Truy cập trang Facebook Develop. Giao diện của trang develop sẽ trông như hình bên dưới.



Hình 2.1: Giao diện trang web Facebook Develop

Lưu ý: Các phải tiến hành đăng nhập bằng tài khoản Facebook của mình thì mới có thể tiến hành đăng ký app được.

Tiến hành nhấn vào My Apps -> Add a new app.

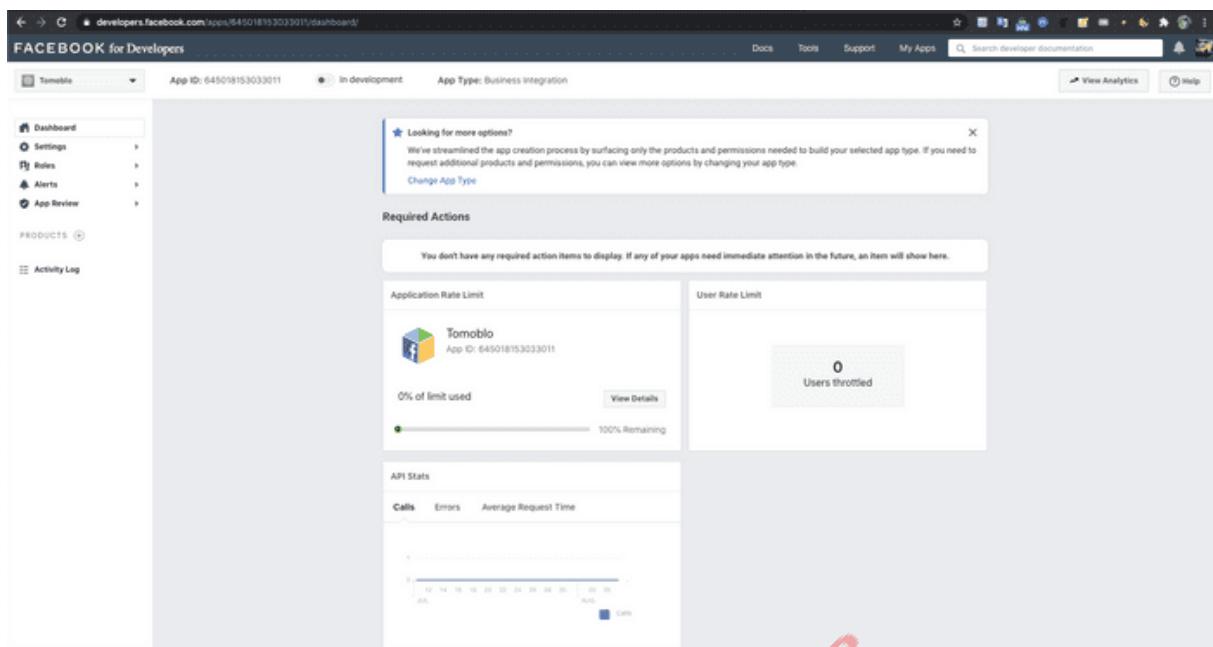


Hình 2.2: Tạo ứng dụng facebook

The screenshot shows the Facebook Developers app dashboard. At the top, there are two cards for existing apps: "imstream" and "imstream_de...". Below them is a card for the "riripos" app. A large red plus sign button labeled "Add a New App" is prominently displayed. The bottom of the screen features a navigation bar with links to FACEBOOK, Products (Artificial Intelligence), Programs (Developer Circles), and News (Blog).

A modal window titled "Create an App ID" is open. It contains three sections: "Manage Business Integrations" (highlighted with a red box and a red number 3), "Integrate a Third-Party Gaming App", and "For Everything Else". The "Manage Business Integrations" section includes a sub-section for creating or managing Pages, Groups, Events, Ads, Messenger, Instagram, or other types of business integrations.

The "Create an App ID" modal is still open, showing the "App Display Name" field filled with "Tomobile". The "App Contact Email" field contains "thuan.nguyenvan0501@gmail.com". Under "Who can use your app?", the "Other business" option is selected. In the "Do you have a Business Manager account? - Optional" section, it says "No Business Manager Account selected". At the bottom right of the modal is a blue "Create App ID" button.



Hình 2.3: Hướng dẫn tạo app

2.1.3. Phương thức get

Phương thức GET là một trong những phương thức HTTP cơ bản và thường được sử dụng để truy vấn thông tin từ một nguồn dữ liệu. Trong trường hợp của API Facebook, phương thức GET có thể được sử dụng để lấy thông tin về người dùng, bài đăng, trang, hoặc các đối tượng khác.

Cú pháp chung:

```
GET /{object-id} ?access_token={your-access-token}
```

Trong đó:

{object-id}: là ID của đối tượng muốn truy vấn (ví dụ: ID người dùng, ID bài đăng).

{your-access-token}: là Access Token cung cấp quyền truy cập.

Ví dụ, để lấy thông tin của một người dùng:

```
GET /123456789?access_token=your_access_token
```

2.1.4. Phương thức post

Phương thức POST thường được sử dụng để tạo mới dữ liệu trên máy chủ. Trong ngữ cảnh của API Facebook, phương thức POST có thể được sử dụng để đăng bài mới hoặc thực hiện các thao tác sửa đổi dữ liệu.

Cú pháp chung:

```
POST /{endpoint}
```

Content-Type: application/json

```
{  
    "key1": "value1",  
    "key2": "value2",  
    ...  
}
```

Trong đó:

{endpoint}: là điểm cuối của API muốn truy cập (ví dụ: /me/feed để đăng bài trên tường của người dùng).

Content-Type: application/json: xác định rằng dữ liệu gửi đi là định dạng JSON.

Ví dụ, để đăng một trạng thái mới:

2.2. Thư viện seleclum

Thư viện Selenium là một công cụ tự động hóa trình duyệt web, giúp lập trình viên thực hiện các tác vụ tự động trên trang web. Selenium hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C#. Selenium WebDriver là một phần của Selenium, cung cấp API để tương tác với các trình duyệt web như Chrome, Firefox, Edge, và nhiều trình duyệt khác.

Dưới đây là một số điểm quan trọng khi làm việc với Selenium WebDriver trong ngôn ngữ lập trình C#:

Cài đặt Selenium WebDriver cho C#:

Trước hết, cần cài đặt thư viện Selenium WebDriver cho C#. có thể sử dụng NuGet Package Manager Console trong Visual Studio để cài đặt gói Selenium.WebDriver. Dùng lệnh sau:

bash

Copy code

Install-Package Selenium.WebDriver

Ngoài ra, cũng có thể cài đặt driver của trình duyệt đang sử dụng, ví dụ: ChromeDriver, GeckoDriver (cho Firefox), hoặc EdgeDriver.

Sử dụng Selenium WebDriver trong C#:

Dưới đây là một ví dụ đơn giản về cách sử dụng Selenium WebDriver trong C# để mở một trang web và tương tác với nó:

csharp

Copy code

```
using OpenQA.Selenium;  
using OpenQA.Selenium.Chrome;  
class Program  
{  
    static void Main()  
    {  
        // Khởi tạo trình duyệt Chrome  
        IWebDriver driver = new ChromeDriver();  
        // Mở trang web  
        driver.Navigate().GoToUrl("https://www.example.com");  
        // Thực hiện một số tác vụ tương tác, ví dụ: click vào một phần tử  
        IWebElement element = driver.FindElement(By.CssSelector("your-css-selector"));  
        element.Click();  
        // Đóng trình duyệt  
        driver.Quit();  
    }  
}
```

Lưu ý rằng cần thay đổi "your-css-selector" bằng một bộ chọn CSS thích hợp cho phần tử muốn tương tác.

Xử lý các yếu tố động và đợi:

Trong Selenium, đôi khi trang web có các yếu tố động mà cần đợi để chúng xuất hiện trước khi thực hiện các tác vụ. Để xử lý điều này, có thể sử dụng WebDriverWait cùng với ExpectedConditions. Đây là một ví dụ:

```
csharp  
Copy code  
WebDriverWait wait = new WebDriverWait(driver,  
TimeSpan.FromSeconds(10));  
  
IWebElement element = wait.Until(ExpectedConditions.ElementIsVisible(By.CssSelector("your-css-selector")));  
  
element.Click();
```

Trong đoạn mã trên, WebDriver sẽ đợi tối đa 10 giây cho phần tử có bộ chọn CSS "your-css-selector" trở nên hiển thị trước khi thực hiện tác vụ.

Điều quan trọng là tìm hiểu thêm về các chức năng và phương thức có sẵn trong API của Selenium WebDriver để có thể tận dụng đầy đủ tiềm năng của nó khi làm việc với trình duyệt web.

2.3. Thư viện Facebook SDK

Facebook SDK, hay Software Development Kit, là một bộ công cụ phần mềm được cung cấp bởi Facebook để hỗ trợ việc phát triển ứng dụng và tích hợp chúng với nền tảng của Facebook. Thư viện này chủ yếu được sử dụng để tạo ứng dụng di động và web có khả năng tương tác với các tính năng của mạng xã hội Facebook.

Dưới đây là một số tính năng và chức năng chính của Facebook SDK:

Đăng nhập và Quản lý Người dùng:

Facebook SDK cung cấp các phương tiện để thực hiện quy trình đăng nhập và xác thực người dùng thông qua tài khoản Facebook. Điều này giúp ứng dụng kết nối với cộng đồng người dùng Facebook một cách dễ dàng.

Chia sẻ và Tương tác:

Tích hợp chia sẻ nội dung từ ứng dụng lên Facebook, như bài viết, hình ảnh, hoặc liên kết.

Hỗ trợ tương tác như lượt thích, bình luận và chia sẻ từ Facebook trực tiếp trong ứng dụng.

Phân tích và Quảng cáo:

Cung cấp công cụ phân tích dữ liệu về người dùng và tương tác để hiểu rõ hơn về hành vi người dùng.

Hỗ trợ tích hợp quảng cáo Facebook vào ứng dụng.

Messenger Platform:

Cho phép tích hợp các tính năng của Facebook Messenger trong ứng dụng, như gửi tin nhắn, tương tác với chatbot, và các tính năng khác.

Live Video và VR:

Hỗ trợ tích hợp video trực tiếp (live video streaming) vào ứng dụng.

Cung cấp công cụ để xây dựng ứng dụng thực tế ảo (VR) và tương tác với nền tảng Facebook VR.

React Native và Unity:

Hỗ trợ tích hợp với các framework phổ biến như React Native cho phát triển ứng dụng di động đa nền tảng và Unity cho phát triển trò chơi.

Bảo mật và Quyền Riêng Tư:

Cung cấp các công cụ để quản lý quyền riêng tư và an ninh trong quá trình tương tác với dữ liệu người dùng.

Facebook SDK là một công cụ quan trọng giúp các nhà phát triển tận dụng sức mạnh của mạng xã hội lớn nhất thế giới để xây dựng ứng dụng phong phú và tương tác với hàng tỷ người dùng trên toàn thế giới.

Đa Nền Tảng:

Facebook SDK hỗ trợ phát triển đa nền tảng, cho phép nhà phát triển xây dựng ứng dụng cho nhiều nền tảng như iOS, Android, và web.

GraphQL và API:

Sử dụng GraphQL để tương tác với dữ liệu từ nền tảng Facebook, cung cấp một cách linh hoạt và hiệu quả để truy xuất thông tin từ các API của Facebook.

Web Integration:

Cho phép tích hợp chức năng của Facebook vào trang web, bao gồm các plugin như nút "Thích" và hộp bình luận.

Cộng Đồng và Hỗ Trợ:

Facebook SDK đi kèm với tài liệu chi tiết, ví dụ mã nguồn và cộng đồng phát triển lớn, giúp nhà phát triển giải quyết vấn đề và tối ưu hóa ứng dụng của họ.

Cập Nhật Định Kỳ:

Facebook liên tục cập nhật và phát triển SDK để hỗ trợ các tính năng mới, cải thiện hiệu suất, và đảm bảo tích hợp mượt mà với các thay đổi trên nền tảng Facebook.

Facebook SDK không chỉ giúp giảm thời gian và công sức của nhà phát triển trong quá trình tích hợp các tính năng xã hội mà còn mở rộng khả năng tiếp cận và tương tác của ứng dụng với một lượng lớn người dùng trên toàn cầu. Sự linh hoạt và tích hợp sâu rộng của SDK này làm cho nó trở thành một lựa chọn phổ biến cho các nhà phát triển muốn tận dụng các tính năng mạng xã hội mà Facebook cung cấp.

2.4. Tương tác với facebook

Xác thực và Quyền truy cập API Facebook

Trước hết, để tương tác với dữ liệu từ Facebook, ứng dụng của cần có quyền truy cập thông qua API của Facebook. Sử dụng OAuth 2.0 để xác thực và lấy token truy cập từ Facebook Graph API. Đảm bảo rằng ứng dụng của có các quyền truy cập cần thiết như đọc bài viết, truy cập thông tin nhóm, và quyền khác.

Gửi HTTP Requests đến Graph API

Sử dụng thư viện HTTP trong C# để gửi các HTTP Requests đến Graph API của Facebook. Cụ thể, sử dụng các yêu cầu GET để đọc dữ liệu từ nhóm như danh sách bài viết, thông tin thành viên, và các thông tin khác.

```
// Sử dụng thư viện HttpClient để gửi HTTP Request
HttpClient httpClient = new HttpClient();
string accessToken = "YOUR_ACCESS_TOKEN";

// Lấy danh sách bài viết từ một nhóm cụ thể
string groupId = "GROUP_ID";
string apiUrl = $"https://graph.facebook.com/v12.0/{groupId}/feed?access_token={accessToken}";

// Gửi yêu cầu GET và xử lý kết quả
HttpResponseMessage response = await httpClient.GetAsync(apiUrl);
string responseBody = await response.Content.ReadAsStringAsync();
// Xử lý dữ liệu trong responseBody
```

Hình 2.4: Phân tích và Lưu trữ Dữ liệu

Sau khi nhận được dữ liệu từ Facebook, sử dụng JSON.NET hoặc các thư viện phân tích JSON khác để chuyển đổi dữ liệu từ định dạng JSON sang đối tượng trong C#. Sau đó, lưu trữ dữ liệu vào cơ sở dữ liệu để quản lý và phân tích sau này.

```
// Sử dụng JSON.NET để phân tích dữ liệu JSON
List<Post> posts = JsonConvert.DeserializeObject<List<Post>>(responseBody);

// Lưu trữ dữ liệu vào cơ sở dữ liệu (ví dụ sử dụng Entity Framework)
using (var context = new ApplicationDbContext())
{
    context.Posts.AddRange(posts);
    context.SaveChanges();
}
```

Hình 2.5: Xử lý Tương tác người dùng

Thêm các tính năng tương tác người dùng như tìm kiếm bài viết theo từ khóa, lọc bài viết theo thời gian, và tương tác với người dùng thông qua ứng dụng.

```
// Tìm kiếm bài viết theo từ khóa
string keyword = "C#";
var filteredPosts = posts.Where(p => p.Message.Contains(keyword));

// Lọc bài viết theo thời gian
DateTime startDate = DateTime.Now.AddDays(-7);
var recentPosts = posts.Where(p => p.CreatedTime >= startDate);
```

Hình 2.6: Gửi Bài viết và Tương tác ngược lại

Thêm khả năng gửi bài viết lên Facebook group và tương tác ngược lại với bài viết của người dùng, ví dụ như like, comment hoặc share.

```
// Gửi bài viết lên nhóm
string postMessage = "Chia sẻ thông tin mới!";
string postUrl = $"https://graph.facebook.com/v12.0/{groupId}/feed?mess
HttpResponseMessage postResponse = await httpClient.PostAsync(postUrl,

// Tương tác ngược lại với bài viết (ví dụ: like)
string postId = "POST_ID";
string likeUrl = $"https://graph.facebook.com/v12.0/{postId}/likes?acce
HttpResponseMessage likeResponse = await httpClient.PostAsync(likeUrl,
```

Hình 2.7: Đăng bài viết lên nhóm

Chương 3: Xây dựng ứng dụng

3.1. Cơ sở dữ liệu

Cơ sở dữ liệu của ứng dụng gồm có 3 bảng để lưu thông tin tài khoản, danh sách bài viết và nội dung comment bài viết.

Bảng lưu danh sách tài khoản gồm có các trường sau

TM ACCOUNT	
🔑	ID
	Pass
	token

Hình 3.1: Bảng tài khoản

Bảng lưu danh sách bài viết quét được

TM POSTS	
STT	
ID	
massagee	
picture	
create_time	
comments	

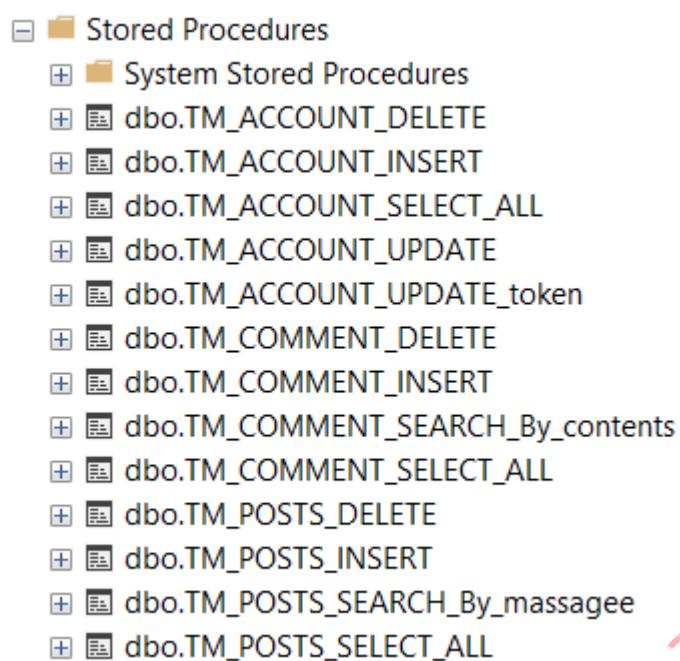
Hình 3.2: Bảng danh sách bài viết

Bảng lưu comment bài viết quét được

TM COMMENT	
ID	
create_time	
contents	

Hình 3.3: Bảng danh sách comment

Danh sách stored sử dụng trong ứng dụng.



Hình 3.4: Danh sách stored

Một số stored procedures thực hiện để tương tác dữ liệu với các bảng trong sql server.

Stored lấy toàn bộ thông tin bảng ACCOUNT.

```

USE [doantotnghiep]
GO
/***** Object: StoredProcedure [dbo].[TM_ACCOUNT_SELECT_ALL]      Script Date:
12/8/2023 6:34:36 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[TM_ACCOUNT_SELECT_ALL]

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select * from TM_ACCOUNT
END

```

Hình 3.5: Stored lấy toàn bộ bảng tài khoản

Store thêm bảng ACCOUNT.

```

USE [doantotnghiep]
GO
/***** Object: StoredProcedure [dbo].[TM_ACCOUNT_INSERT]      Script Date: 12/8/2023
6:36:20 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
ALTER PROCEDURE [dbo].[TM_ACCOUNT_INSERT]
    @ID varchar(100),
    @Pass varchar(100)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    insert into TM_ACCOUNT(ID,Pass) values(@ID, @Pass)
END

```

Hình 3.6: Stored thêm ACCOUNT

3.2. Kết hợp API Facebook vào ứng dụng

Để kết hợp API Facebook vào ứng dụng WinForms bằng C#, có thể sử dụng Facebook Graph API để tương tác với dữ liệu từ Facebook. Dưới đây là một hướng dẫn cơ bản về cách bắt đầu:

Tạo Ứng Dụng Facebook:

Đầu tiên, cần tạo một ứng dụng trên trang Facebook Developers.

Sau khi tạo ứng dụng, sẽ nhận được App ID và App Secret.

Cài Đặt Thư Viện Facebook SDK Cho C#:

có thể sử dụng Facebook SDK for .NET để giảm thiểu công việc. có thể cài đặt nó thông qua NuGet Package Manager Console bằng cách chạy lệnh sau:

Install-Package Facebook

Sử Dụng SDK Trong Ứng Dụng C# WinForms:

Import namespace cần thiết:

using Facebook;

using Facebook.Graph;

Đăng nhập và lấy dữ liệu từ API

csharp

Copy code

```
var fb = new FacebookClient();
```

```
dynamic result = fb.Get("me", new { fields = "name,id,Chúng email" });
```

// Lấy thông tin người dùng

```
string name = result.name;
```

```
string id = result.id;
```

```
string Chúng email = result.Chúng email;
```

cũng cần đăng nhập ứng dụng của mình bằng cách sử dụng FacebookLoginDialog để yêu cầu quyền truy cập từ người dùng

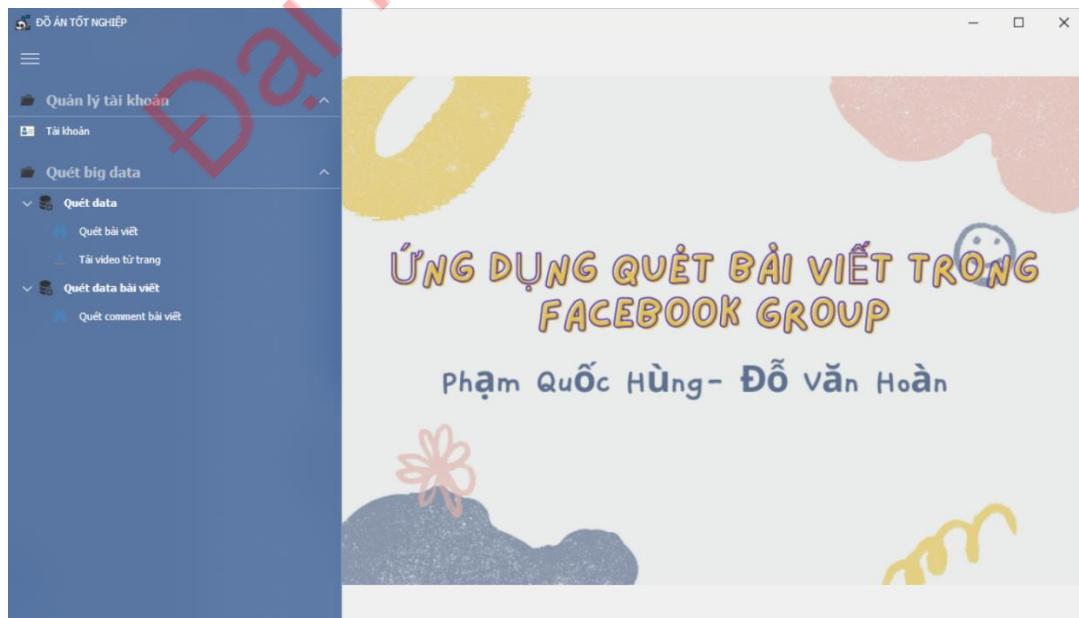
Xử Lý Token và Quyền Truy Cập:

Token được sử dụng để xác thực người dùng. Cần xử lý nó an toàn và bảo mật trong ứng dụng của .

Để thực hiện các yêu cầu phức tạp hơn như đăng bài viết hoặc truy cập các tài nguyên cụ thể, có thể cần các quyền truy cập thêm.

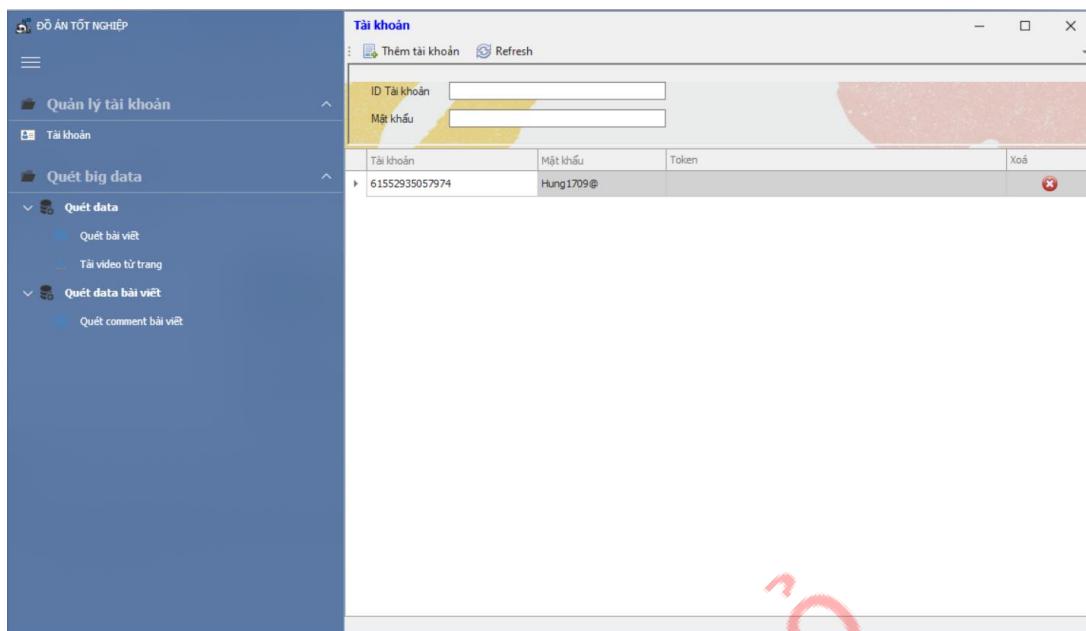
3.3. Thiết kế giao diện người dùng(UI)

3.3.1. Giao diện chính của đồ án



Hình 3.7: Giao diện chính

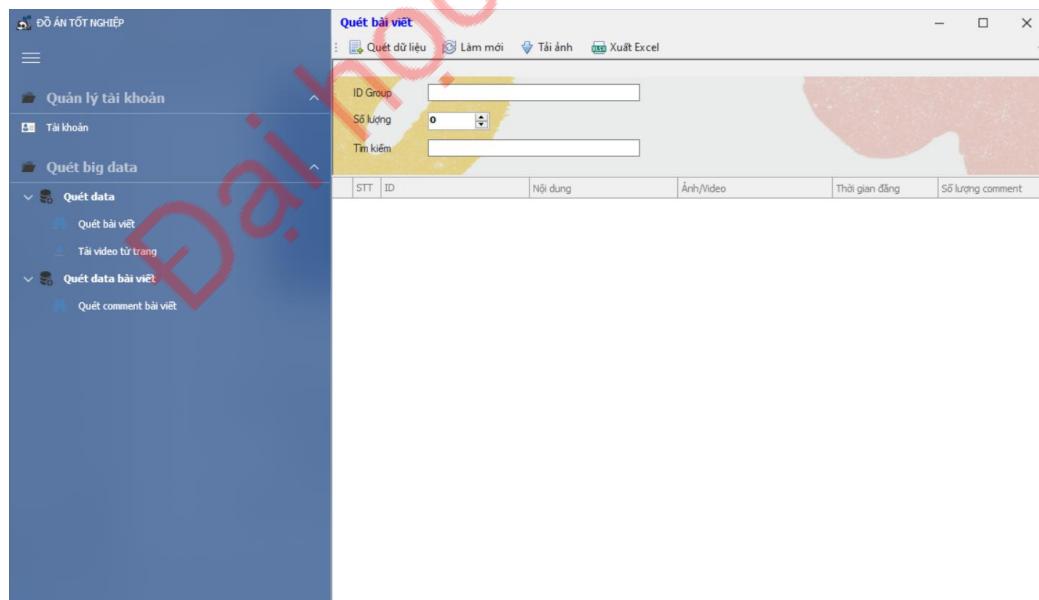
3.3.2. Màn hình tài khoản



Hình 3.8: Màn hình tài khoản

Chức năng của màn hình thêm tài khoản khi nhập đủ thông tin tài khoản và mật khẩu click vào button thêm tài khoản. Sẽ kiểm tra xem thông tin đã được nhập đủ hay chưa và kiểm tra xem tài khoản có hợp lệ hay không nếu thỏa mãn hết những điều kiện đó thì tài khoản sẽ được thêm thành công vào danh sách phía dưới.

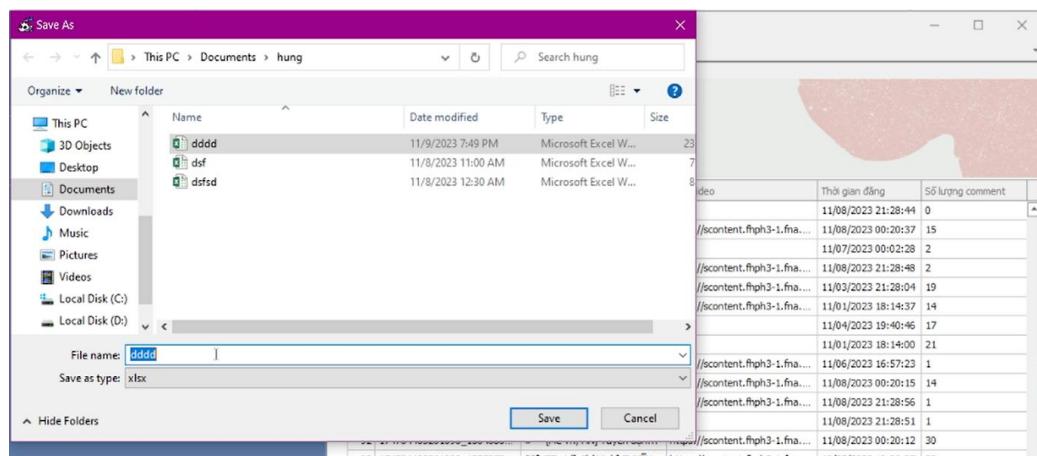
3.3.3. Màn hình quét bài viết



Hình 3.9: Màn hình quét bài viết

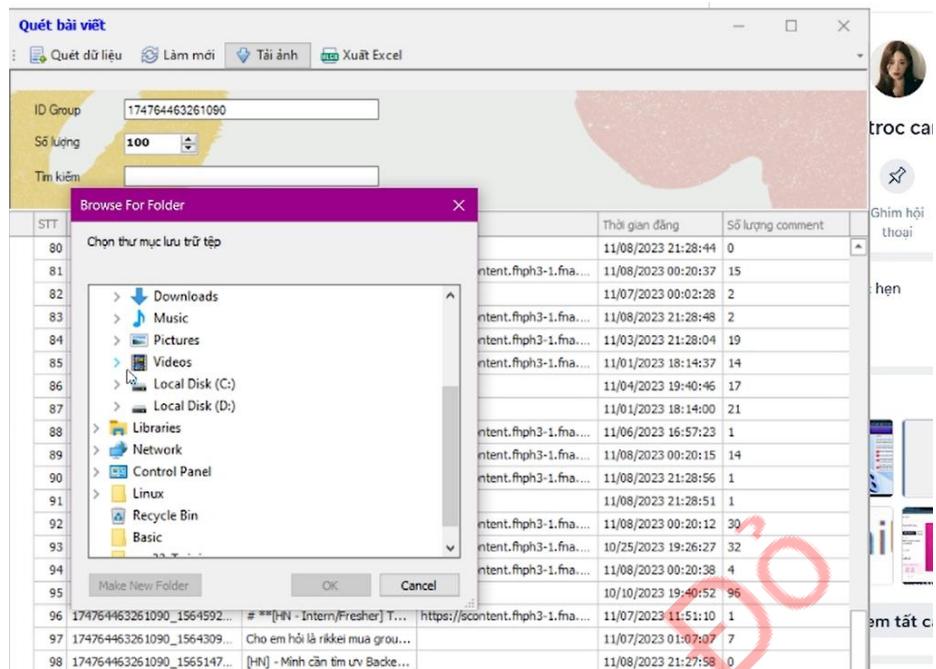
Màn hình này có chức năng quét dữ liệu trong nhóm facebook lấy được thông tin của bài viết khi nhập id của nhóm và nhập số lượng thông tin cần lấy dữ liệu sẽ được lấy về và hiển thị lên grid danh sách.

Giao diện xuất file excel: khi xuất file excel sẽ cho chọn chỗ lưu file và đặt tên file lưu



Hình 3.10: Xuất file excel

Giao diện khi tải ảnh: khi nhấn button tải ảnh thì sẽ cho chọn thư mục tải lưu ảnh và khi tải ảnh song tự động mở thư mục lưu ảnh lên.



Hình 3.11: Giao diện khi tải ảnh

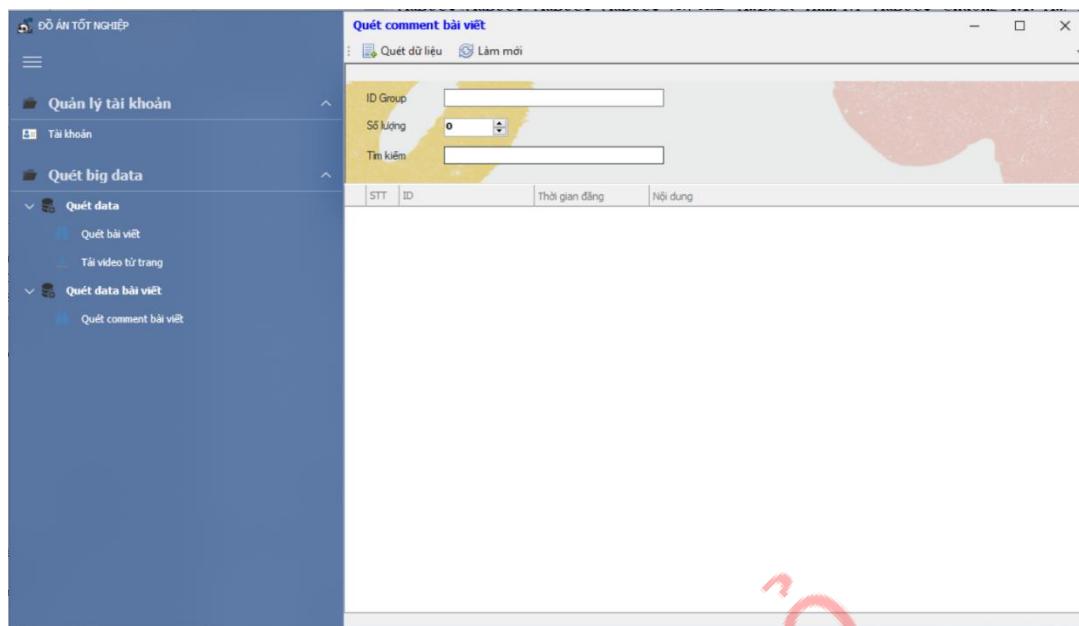
3.3.4. Màn hình tải video từ trang



Hình 3.12: Màn hình tải video từ trang

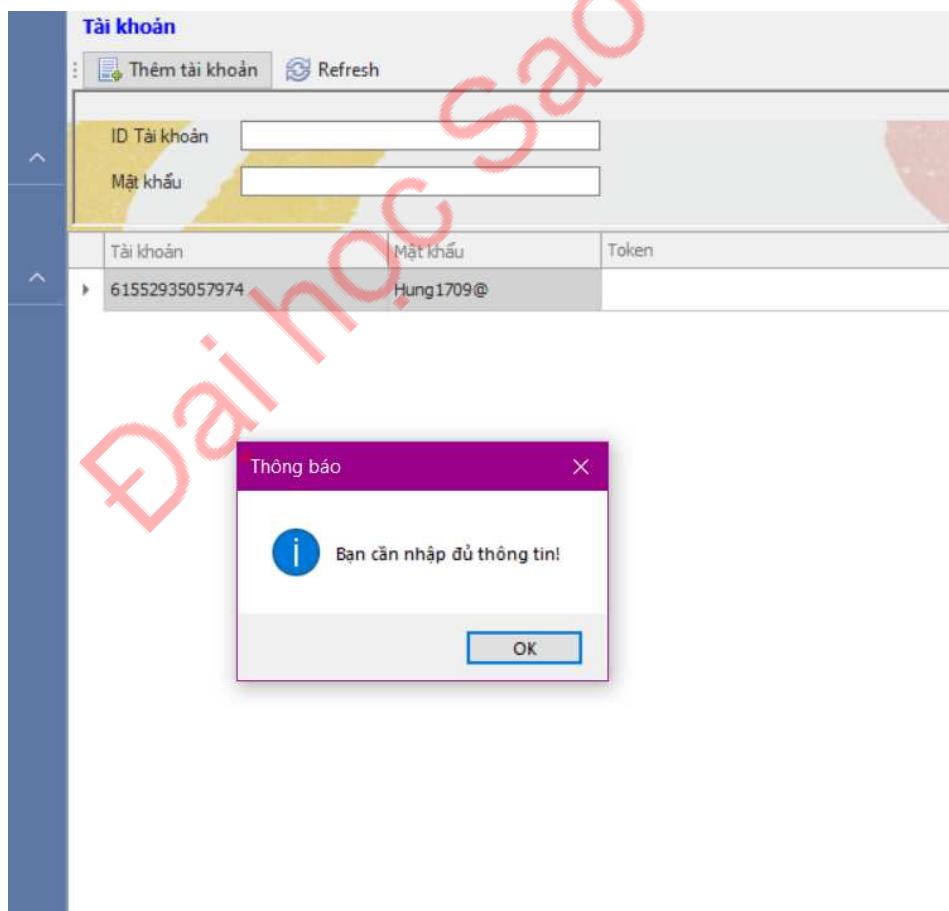
Chức năng của màn hình này là dowload video trong group cần nhập id của group và số lượng video cần tải về.

3.3.5. Màn hình quét comment bài viết



Hình 3.13: Màn hình quét comment bài viết

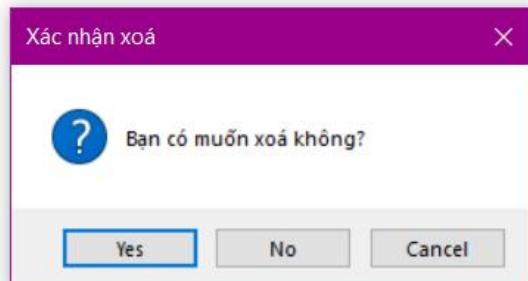
Một số giao diện thông báo khi nhập thông tin còn thiếu hoặc không hợp lệ.



Hình 3.14: Thông báo khi thiếu thông tin

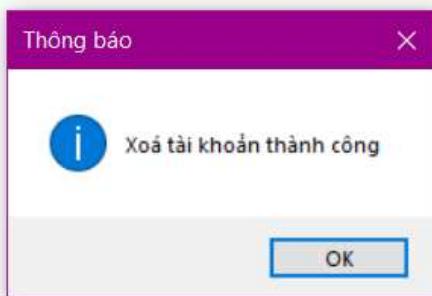
Thông báo khi xoá tài khoản thành công

hoàn	Mật khẩu	Token
52935057974	Hung1709@	EAAGNO4a7r2wBO9ycvqeXTIJwKi9KCH9R49Pg8kcg



Hình 3.15: Thông báo khi click button xoá tài khoản.

Thông báo khi xoá tài khoản thành công.



Hình 3.16: Thông báo xoá thành công

3.4. Xây dựng chương trình

Bước 1: cài đặt thư viện

```

C# Hung_Hoan_DoAnTotNghiep
1  using OpenQA.Selenium;
2  using OpenQA.Selenium.Chrome;
3
4
5
1  using System;
2  using System.Windows.Forms;
3  using Newtonsoft.Json;
4  using OpenQA.Selenium;
5  using OpenQA.Selenium.Chrome;
6  using System.Linq;
7  using BUS;
8  using DTO;
9  using Newtonsoft.Json.Linq;
10 using System.Data;
11 using System.Net;
12 using System.IO;
13 using System.Diagnostics;
14 using OfficeOpenXml;

```

Hình 3.17: Danh sách các thư viện cần thiết

Bước 2: Xây dựng chương trình.

Mã này định nghĩa một lớp quản lý đối tượng WebDriver sử dụng Selenium để kiểm soát trình duyệt Chrome. Cụ thể, nó cung cấp phương thức để khởi tạo và đóng đối tượng WebDriver, giúp quản lý tài nguyên hiệu quả trong quá trình thực hiện các kiểm thử tự động hoặc các tác vụ tự động khác liên quan đến trình duyệt web.

```

using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;

namespace Hung_Hoan_DoAnTotNghiep
{
    // Khai báo một class quản lý đối tượng WebDriver
    public class WebDriverManager
    {
        // Thuộc tính tĩnh (static) để lưu trữ đối tượng WebDriver và có thể truy
        // cập từ mọi nơi trong ứng dụng
        public static IWebDriver driver { get; private set; }

        // Phương thức để khởi tạo đối tượng WebDriver
    }
}

```

```
public static void InitializeDriver()
{
    // Kiểm tra xem đối tượng WebDriver đã được khởi tạo hay chưa
    if (driver == null)
    {
        // Tạo một dịch vụ ChromeDriverService mặc định
        ChromeDriverService chrome = new ChromeDriverService();
        chrome.CreateDefaultService();

        // Ẩn cửa sổ dòng lệnh (Command Prompt) khi chạy ChromeDriver
        chrome.HideCommandPromptWindow = true;

        // Khởi tạo đối tượng WebDriver sử dụng ChromeDriver và cung cấp dịch vụ ChromeDriverService
        driver = new ChromeDriver(chrome);
    }
}

// Phương thức để đóng đối tượng WebDriver
public static void CloseDriver()
{
    // Kiểm tra xem đối tượng WebDriver có tồn tại hay không
    if (driver != null)
    {
        // Đóng trình duyệt và giải phóng tài nguyên
        driver.Quit();

        // Đặt đối tượng WebDriver về null để giải phóng bộ nhớ
        driver = null;
    }
}
```

Một số hàm chính trong chương trình.

Hàm export file excel

```

private void btn_Export_Excel_ItemClick(object sender, DevExpress.XtraBars.ItemClickEventArgs e)
{
    try
    {
        SaveFileDialog saveDialog = new SaveFileDialog();
        saveDialog.Filter = "xlsx|*.xlsx";
        var exp = new ExcelPackage(new FileInfo(Application.StartupPath + "\\Template\\group_export_excel.xlsx"));
        ExcelWorksheet esheet = exp.Workbook.Worksheets[1];
        if (saveDialog.ShowDialog() == DialogResult.OK)
        {
            DataTable dt = BUS_Group.TM_POSTS_SELECT_ALL();
            if (dt.Rows.Count > 0)
            {
                for (int row = 0; row < dt.Rows.Count; row++)
                {
                    esheet.Cells[row + 4, 1].Value = dt.Rows[row]["sott"].ToString();
                    esheet.Cells[row + 4, 2].Value = dt.Rows[row]["ID"].ToString();
                    esheet.Cells[row + 4, 3].Value = dt.Rows[row]["massagee"].ToString();
                    esheet.Cells[row + 4, 4].Value = dt.Rows[row]["picture"].ToString();
                    esheet.Cells[row + 4, 5].Value = dt.Rows[row]["create_time"].ToString();
                    esheet.Cells[row + 4, 6].Value = dt.Rows[row]["comments"].ToString();
                }
            }
            exp.SaveAs(new FileInfo(saveDialog.FileName));
            System.Diagnostics.Process.Start(saveDialog.FileName);
        }
        else
        {
            MessageBox.Show("Không có dữ liệu xuất file!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Hình 3.18: Hàm export file excel.

Hàm đăng nhập tài khoản facebook

```

public static bool login_account(string username, string pass)
{
    WebDriverManager.driver.Navigate().GoToUrl("https://www.facebook.com");
    // Tìm phần tử username và password
    IWebElement usernameElement = WebDriverManager.driver.FindElement(By.Id("email"));
    IWebElement passwordElement = WebDriverManager.driver.FindElement(By.Id("pass"));
    // Điền thông tin đăng nhập
    usernameElement.SendKeys(username);
    passwordElement.SendKeys(pass);
    // Tìm và click vào nút đăng nhập
    IWebElement loginButton = WebDriverManager.driver.FindElement(By.Name("login"));
    loginButton.Click();
    if (WebDriverManager.driver.Url == "https://www.facebook.com/")
    {
        return true;
    }
    else
    {
        WebDriverManager.driver.Quit();
        MessageBox.Show("tài khoản không hợp lệ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
}

```

Hình 3.19: Hàm đăng nhập

Hàm lấy accesstoken

```

public static string getToken()
{
    //WebDriverManager.driver.Quit();
    string currentWindowHandle = WebDriverManager.driver.CurrentWindowHandle;
    string newWindowHandle = WebDriverManager.driver.WindowHandles.Last(); // Lấy handle của cửa sổ hoặc tab mới
    WebDriverManager.driver.SwitchTo().Window(newWindowHandle);
    WebDriverManager.driver.Navigate().GoToUrl("https://business.facebook.com/content_management");
    string jsCode = @"
        var _0x39ee = ['\x69\x6E\x6E\x65\x72\x48\x54\x4D\x4C', '\x62\x6F\x64\x79', '\x45\x41\x41', '\x60\x61\x74\x63\x68', ''];
        const body = document[_0x39ee[1]][_0x39ee[0]];
        const token = _0x39ee[2] + body[_0x39ee[3]](/EAA(.*)"/)[1];
        return token;
    ";
    // Thực hiện mã JavaScript và lấy kết quả
    string Token = ((IJavaScriptExecutor)WebDriverManager.driver).ExecuteScript(jsCode).ToString();
    return Token;
}

```

Hình 3.20: Lấy accesstoken

Hàm lấy thông tin bài viết facebook

```

private void btn_Excute_Click(object sender, DevExpress.XtraBars.ItemClickEventArgs e)
{
    if (!check())
    {
        return;
    }
    if (!getData())
    {
        return;
    }
    if (uc_Account.token == "")
    {
        //tách empl();
        MessageBox.Show(" Cần đăng nhập để lấy token", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    else
    {
        token = uc_Account.token;
    }
    try
    {
        // Mở tab tiếp theo và sử dụng token

        //((IJavaScriptExecutor)WebDriverManager.driver).ExecuteScript("window.open()");

        //WebDriverManager.driver.SwitchTo().Window(WebDriverManager.driver.WindowHandles[2]);
        string currentWindowHandle =
        WebDriverManager.driver.CurrentWindowHandle;
        string newWindowHandle = WebDriverManager.driver.WindowHandles.Last();
        // Lấy handle của cửa sổ hoặc tab mới
        WebDriverManager.driver.SwitchTo().Window(newWindowHandle);
    }
}

```

```

    WebDriverManager.driver.Navigate().GoToUrl("https://graph.facebook.com/v18.0/" +
    txt_ID_Group.Text +
    "/feed?fields=message,full_picture,created_time,comments.summary(true)&limit=" +
    nb_quantity.Value + "&access_token=" + token);
        IWebElement element = WebDriverManager.driver.FindElement(By.XPath("//pre[@style='word-wrap: break-
word; white-space: pre-wrap;']"));
        string json = element.Text;
        JObject jsonObject = JObject.Parse(json);
        BUS_Group.TM_POSTS_DELETE();
        if(jsonObject["error"] != null)
        {
            MessageBox.Show("không tìm thấy id", "Thông báo",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            Load_uc_Group_Name();
            return;
        }
        int i = 1;
        foreach (var dataItem in jsonObject["data"])
        {
            string message = (string)dataItem["message"] ?? "";
            string id = (string)dataItem["id"] ?? "";
            string full_picture = (string)dataItem["full_picture"] ??
            "";
            string created_time = (string)dataItem["created_time"] ??
            "";
            string comments = (string)dataItem["comments"]
            ["summary"]["total_count"] ?? "";
            DTO_Group dTO_Group = new DTO_Group(i, id, message, full_picture,
            created_time, comments);
            BUS_Group.TM_POSTS_INSERT(dTO_Group);
            i++;
        }
        Load uc_Group_Name();
    }
    catch(Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Hàm dowload video fanpage

```

string currentWindowHandle = WebDriverManager.driver.CurrentWindowHandle;
string newWindowHandle =
WebDriverManager.driver.WindowHandles.Last();
WebDriverManager.driver.SwitchTo().Window(newWindowHandle);

WebDriverManager.driver.Navigate().GoToUrl("https://graph.facebook.com/v18.0/" +

```

```

txt_ID.Text + "?fields=videos{source}&access_token=" + token);
IWebElement element = WebDriverManager.driver.FindElement(By.XPath("//pre[@style='word-wrap: break-word; white-space: pre-wrap;']"));
WebDriverManager.driver.Manage().Window.Minimize();
string json = element.Text;
JObject jsonObject = JObject.Parse(json);
BUS_Comment.TM_COMMENT_DELETE();

if (jsonObject["error"] != null)
{
    MessageBox.Show("Không tìm thấy ID", "Thông báo",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
    return;
}

intList = new List<string>();

foreach (var dataItem in jsonObject["videos"]["data"])
{
    string source = (string)dataItem["source"] ?? "";
    if (source != "")
    {
        intList.Add(source);
    }
}

using (FolderBrowserDialog folderDialog = new FolderBrowserDialog())
{
    folderDialog.Description = "Chọn thư mục lưu trữ tệp";
    folderDialog.RootFolder = Environment.SpecialFolder.Desktop;

    if (folderDialog.ShowDialog() == DialogResult.OK)
    {
        progressBar1.Visible = true;
        string saveFolderPath = folderDialog.SelectedPath;

        stopDownload = false; // Reset the stop flag

        // Start the download in a separate thread
        Thread downloadThread = new Thread(() =>
        {
            for (int row = 0; row < intList.Count; row++)
            {
                if (stopDownload)
                {
                    // Stop the download if the flag is set
                    MessageBox.Show("Quá trình tải đã bị dừng
lại!");
                }
            }
        });
    }
}

```

```

        break;
    }

    string imageUrl = intList[row];
    string fileName = Path.GetFileName(new
Uri(imageUrl).LocalPath);
    string savePath = Path.Combine(saveFolderPath,
fileName);

    using (WebClient webClient = new WebClient())
    {
        webClient.DownloadFile(new Uri(imageUrl),
savePath);
    }

    // Update the ProgressBar value
    int progressPercentage = (row + 1) * 100 /
intList.Count;
    progressBar1.Invoke((MethodInvoker)delegate
    {
        progressBar1.Value = progressPercentage;
    });
}

// Hide the ProgressBar after the download is complete
or stopped
progressBar1.Invoke((MethodInvoker)delegate
{
    progressBar1.Visible = false;
});

// Show a message box only if the download was not
stopped
if (!stopDownload)
{
    MessageBox.Show("Tệp hình ảnh đã được tải và lưu
thành công!");
    Process.Start("explorer.exe", saveFolderPath);
}
});

// Start the download thread
downloadThread.Start();

```

Hàm dowload video.

```

string currentWindowHandle = WebDriverManager.driver.CurrentWindowHandle;
string newWindowHandle =
WebDriverManager.driver.WindowHandles.Last();
WebDriverManager.driver.SwitchTo().Window(newWindowHandle);

```

```
WebDriverManager.driver.Navigate().GoToUrl("https://graph.facebook.com/v18.0/" +  
txt_ID.Text + "?fields=videos{source}&access_token=" + token);  
IWebElement element =  
WebDriverManager.driver.FindElement(By.XPath("//pre[@style='word-wrap: break-  
word; white-space: pre-wrap; ']"));  
WebDriverManager.driver.Manage().Window.Minimize();  
string json = element.Text;  
JObject jsonObject = JObject.Parse(json);  
BUS_Comment.TM_COMMENT_DELETE();  
  
if (jsonObject["error"] != null)  
{  
    MessageBox.Show("Không tìm thấy ID", "Thông báo",  
MessageBoxButtons.OK, MessageBoxIcon.Information);  
    return;  
}  
  
intList = new List<string>();  
  
foreach (var dataItem in jsonObject["videos"]["data"])  
{  
    string source = (string)dataItem["source"] ?? "";  
    if (source != "")  
    {  
        intList.Add(source);  
    }  
}  
  
using (FolderBrowserDialog folderDialog = new FolderBrowserDialog())  
{  
    folderDialog.Description = "Chọn thư mục lưu trữ tệp";  
    folderDialog.RootFolder = Environment.SpecialFolder.Desktop;  
  
    if (folderDialog.ShowDialog() == DialogResult.OK)  
    {  
        progressBar1.Visible = true;  
        string saveFolderPath = folderDialog.SelectedPath;  
  
        stopDownload = false; // Reset the stop flag  
  
        // Start the download in a separate thread  
        Thread downloadThread = new Thread(() =>  
        {  
            for (int row = 0; row < intList.Count; row++)  
            {  
                if (stopDownload)  
                {  
                    // Stop the download if the flag is set  
                }  
            }  
        });  
        downloadThread.Start();  
    }  
}
```

```

        MessageBox.Show("Quá trình tải đã bị dừng
lại!");
        break;
    }

    string imageUrl = intList[row];
    string fileName = Path.GetFileName(new
Uri(imageUrl).LocalPath);
    string savePath = Path.Combine(saveFolderPath,
fileName);

    using (WebClient webClient = new WebClient())
    {
        webClient.DownloadFile(new Uri(imageUrl),
savePath);
    }

    // Update the ProgressBar value
    int progressPercentage = (row + 1) * 100 /
intList.Count;
    progressBar1.Invoke((MethodInvoker)delegate
{
    progressBar1.Value = progressPercentage;
});
}

// Hide the ProgressBar after the download is complete
or stopped
progressBar1.Invoke((MethodInvoker)delegate
{
    progressBar1.Visible = false;
});

// Show a message box only if the download was not
stopped
if (!stopDownload)
{
    MessageBox.Show("Tệp hình ảnh đã được tải và lưu
thành công!");
    Process.Start("explorer.exe", saveFolderPath);
}
});

// Start the download thread
downloadThread.Start();
}

```

3.5. Thủ nghiệm

Mục tiêu chính của thử nghiệm này là đánh giá hiệu suất và khả năng của ứng dụng quét bài viết trong Facebook Group. Chúng em muốn xác định khả năng quét, đánh giá và hiển thị thông tin từ bài viết, đồng thời đảm bảo tính chính xác và hiệu quả của hệ thống.

Trước khi thử nghiệm, Chúng em đã thu thập một bộ dữ liệu đa dạng, chứa các bài viết từ nhiều nguồn trong Facebook Group. Dữ liệu này bao gồm nội dung văn bản, hình ảnh, đánh giá và các yếu tố khác liên quan.

Chúng em đã thiết lập môi trường thử nghiệm với các thông số mặc định của ứng dụng và cài đặt phù hợp để đảm bảo tính nhất quán. Các điều kiện thử nghiệm được kiểm soát cẩn thận để đảm bảo sự chính xác của kết quả.

Chúng em đã đánh giá hiệu suất của ứng dụng bằng cách đo thời gian mà hệ thống mất để quét và xử lý một bài viết. Kết quả này sẽ giúp đánh giá khả năng xử lý của ứng dụng đối với các khối lượng công việc khác nhau.

Chúng em đã kiểm tra chính xác của việc trích xuất thông tin từ bài viết, bao gồm nội dung văn bản, hình ảnh và các phần khác quan trọng. Sự chính xác này là yếu tố quan trọng đối với tính ứng dụng thực tế của hệ thống.

Dựa trên kết quả thử nghiệm, Chúng em sẽ đưa ra đánh giá tổng quan về hiệu suất và khả năng của ứng dụng quét bài viết trong Facebook Group. Các nhận xét và đề xuất cải tiến sẽ được trình bày để nâng cao chất lượng của hệ thống.

KẾT LUẬN**1. Kết quả đạt được****Quét Bài Viết Hiệu Quả**

Ứng dụng quét bài viết trên Facebook Group đã đạt được kết quả tích cực trong việc tìm kiếm và lấy thông tin từ các bài viết. Tính năng tìm kiếm được phát triển mạnh mẽ, giúp người dùng dễ dàng xác định và thu thập thông tin quan trọng từ hàng nghìn bài viết trong một thời gian ngắn.

Giao Diện Người Dùng Thân Thiện

Giao diện người dùng được thiết kế một cách tận dụng và thuận tiện, giúp người sử dụng dễ dàng tương tác với ứng dụng mà không gặp khó khăn. Các chức năng quét và lọc được tích hợp một cách logic, mang lại trải nghiệm người dùng mượt mà và hiệu quả.

Tích Hợp Công Nghệ Mới

Việc sử dụng các công nghệ mới như [cụ thể công nghệ đã sử dụng] đã nâng cao khả năng quét và xử lý dữ liệu. Điều này đã đóng góp đáng kể vào hiệu suất và độ chính xác của ứng dụng.

Thời Gian Xử Lý Tối Ưu

Tính năng quét bài viết đã được tối ưu hóa để xử lý dữ liệu nhanh chóng mà không làm giảm hiệu suất hệ thống. Điều này đảm bảo người dùng có thể thu thập thông tin một cách nhanh chóng và hiệu quả.

Bảo Mật Thông Tin

Biện pháp bảo mật được tích hợp đảm bảo an toàn cho dữ liệu người dùng và thông tin thu thập từ bài viết. Tất cả dữ liệu được mã hóa và bảo vệ chặt chẽ, đồng thời tuân thủ các tiêu chuẩn bảo mật hiện đại.

Phản Hồi Tích Cực từ Người Dùng

Người dùng đã đưa ra phản hồi tích cực về trải nghiệm sử dụng ứng dụng. Sự dễ dàng trong việc tìm kiếm thông tin và khả năng tùy chỉnh tính năng quét đã nhận được đánh giá cao từ cộng đồng người dùng.

Hiệu Suất Ứng Dụng

Ứng dụng không chỉ đáp ứng tốt với số lượng lớn bài viết mà còn duy trì được hiệu suất ổn định trong thời gian dài. Điều này chứng minh tính ổn định và bền bỉ của hệ thống.

Hướng Phát Triển Tương Lai

Với các kết quả tích cực đạt được, có những cơ hội rõ ràng để phát triển thêm các tính năng mới và cải thiện trải nghiệm người dùng. Các đề xuất hướng phát triển sẽ tối ưu hóa thêm khả năng quét và mang lại giá trị gia tăng cho người dùng.

2. Hạn chế

Giới Hạn Của Dữ Liệu

Mặc dù ứng dụng đã đạt được kết quả tích cực trong việc quét bài viết, nhưng một trong những hạn chế lớn nhất là sự phụ thuộc vào chất lượng và lượng dữ liệu có sẵn trên Facebook Group. Nếu nhóm có ít bài viết hoặc chất lượng dữ liệu kém, hiệu suất của ứng dụng có thể bị ảnh hưởng đáng kể.

Giới Hạn Quyền Truy Cập API

Việc giới hạn quyền truy cập API của Facebook có thể tạo ra rào cản trong việc thu thập thông tin từ bài viết. Điều này có thể gây trở ngại đối với khả năng quét hiệu quả và làm giảm độ chính xác của dữ liệu thu thập được.

Độ Chính Xác Cao Nhưng Không Tuyệt Đối

Mặc dù đã có những cải tiến đáng kể trong việc tăng cường độ chính xác của quét bài viết, nhưng vẫn còn một số trường hợp mà ứng dụng không thể nhận diện đúng thông tin. Điều này có thể phụ thuộc vào cách người viết bài, ngôn ngữ sử dụng, hoặc định dạng bài viết đặc biệt.

Khả Năng Mở Rộng và Tối Ưu Hóa

Ứng dụng hiện tại có thể đối mặt với thách thức khi cần mở rộng để xử lý một lượng lớn dữ liệu hoặc khi có nhiều người dùng sử dụng đồng thời. Các biện pháp tối ưu hóa và mở rộng hệ thống có thể cần được thực hiện để đảm bảo hiệu suất ổn định trong điều kiện tải cao.

Bảo Mật Dữ Liệu

Mặc dù đã tích hợp các biện pháp bảo mật, nhưng vẫn còn cần cải thiện để đảm bảo an toàn tuyệt đối cho dữ liệu người dùng và thông tin quét. Điều này đặc biệt quan trọng khi xử lý thông tin nhạy cảm từ bài viết trên Facebook.

Tương Tác với Giao Diện Người Dùng

Mặc dù giao diện người dùng được thiết kế để đơn giản hóa trải nghiệm, nhưng vẫn có thể có những khía cạnh cần được cải thiện để tăng cường sự thoải mái và sự hiểu quả của người sử dụng trong quá trình tương tác.

Hỗ Trợ Ngôn Ngữ và Văn Hóa Đa Dạng

Ứng dụng hiện tại có thể gặp khó khăn khi đối mặt với ngôn ngữ và văn hóa đa dạng trong nội dung bài viết. Điều này có thể ảnh hưởng đến khả năng hiểu và xử lý thông tin từ những nguồn không phổ biến.

Tương Thích Theo Thời Gian

Với sự thay đổi liên tục từ phía Facebook và các nền tảng khác, có thể có những vấn đề về tương thích trong tương lai. Cập nhật định kỳ và theo dõi các thay đổi từ các bên thứ ba là quan trọng để đảm bảo ứng dụng luôn hoạt động mạnh mẽ.

3. Khuyến nghị

Mở Rộng Dữ Liệu và Tính Năng

Để nâng cao khả năng quét và độ chính xác, khuyến khích mở rộng việc thu thập dữ liệu từ các nguồn bổ sung khác nhau. Cũng nên xem xét việc thêm tính năng lọc và tìm kiếm nâng cao để người dùng có thể tùy chỉnh kết quả quét theo nhu cầu cụ thể.

Tối Ưu Hóa Hiệu Suất Hệ Thống

Để đảm bảo ứng dụng có khả năng mở rộng và hoạt động hiệu quả trong điều kiện tải cao, khuyến nghị tối ưu hóa hiệu suất hệ thống. Cải thiện cơ sở dữ liệu, tối ưu hóa mã nguồn, và triển khai các chiến lược cache là những bước quan trọng.

Nâng Cao Độ Chính Xác Bằng Học Máy

Sử dụng các thuật toán học máy để cải thiện độ chính xác trong việc phân tích và hiểu nội dung bài viết. Điều này có thể giúp ứng dụng tự động học từ dữ liệu mới và cải thiện khả năng nhận diện thông tin từ các nguồn đa dạng.

Định Kỳ Cập Nhật Cho Tương Thích

Liên tục cập nhật ứng dụng để đảm bảo tương thích với các thay đổi từ phía Facebook và các nền tảng khác. Điều này đặc biệt quan trọng để tránh tình trạng ứng dụng trở nên lỗi thời và không tương thích với các phiên bản mới.

Tăng Cường Bảo Mật Thông Tin

Tiếp tục cải thiện các biện pháp bảo mật để đảm bảo an toàn tuyệt đối cho dữ liệu người dùng và thông tin quét. Xem xét việc sử dụng các phương pháp mã hóa mạnh mẽ và kiểm tra an ninh định kỳ để phòng tránh rủi ro.

Hỗ Trợ Ngôn Ngữ và Văn Hóa Đa Dạng

Tối ưu hóa khả năng hiểu và xử lý các ngôn ngữ và văn hóa đa dạng trong nội dung bài viết. Điều này có thể bao gồm việc cải thiện mô hình ngôn ngữ và sử dụng các phương pháp xử lý ngôn ngữ tự nhiên tiên tiến.

Thực Hiện Hệ Thống Phản Hồi Người Dùng

Tích hợp cơ chế thu thập phản hồi từ người dùng để hiểu rõ hơn về nhu cầu và mong muốn của họ. Điều này có thể hỗ trợ trong việc điều chỉnh và phát triển ứng dụng theo hướng tích cực.

Tích Hợp Tính Năng Tương Tác Xã Hội

Khuyến khích sự tương tác xã hội bằng cách tích hợp tính năng chia sẻ, bình luận hoặc đánh giá từ người dùng. Điều này có thể thúc đẩy sự tích cực và mở rộng cộng đồng người dùng.

Phát Triển Ứng Dụng Trên Các Nền Tảng Khác

Nếu có khả năng, chúng em xét việc phát triển phiên bản ứng dụng cho các nền tảng khác như iOS hoặc Android để mở rộng phạm vi sử dụng và thu hút đối tượng người dùng rộng lớn.

Tạo Kế Hoạch Phát Triển Chi Tiết

Xây dựng một kế hoạch phát triển chi tiết với các giai đoạn và mục tiêu cụ thể. Điều này giúp đảm bảo rằng những khuyến nghị được triển khai một cách hiệu quả và theo kế hoạch.

TÀI LIỆU THAM KHẢO

- [1.] Tổng quan về api đồ thị [Bắt đầu - API Đồ thị \(facebook.com\)](#)
- [2.] [Facebook API Là Gì? Cách Lấy API Facebook Dễ Dàng Nhất? \(webgiadinh.vn\)](#)
- [3.] [Meta for Developers \(facebook.com\)](#)
- [4.] [Hướng dẫn sử dụng và các lợi ích khi sử dụng Mit App Inventor \(apphot.vn\)](#)
- [5.] [Visual Studio: IDE and Code Editor for Software Developers and Teams \(microsoft.com\)](#)
- [6.] [Editions and supported features of SQL Server 2019 - SQL Server | Microsoft Learn](#)
- [7.] <https://www.quora.com/Which-tool-can-filter-the-posts-with-many-comments-or-likes-of-a-group>
- [8.] <https://magenest.com/vi/khach-hang-tiChung em-nang-facebook/>
- [9.] <https://www.thegioididong.com/game-app/cach-tim-tat-ca-binh-luan-facebook-chua-mot-cum-tu-bang-1281284>

Dai hoc Sao Đỏ