TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**TRẦN PHƯƠNG NGỌC ANH**

**TRƯƠNG NGUYỄN NGỌC CHÂU**

**PHÁT TRIỂN HỆ THỐNG XỬ LÝ SONG SONG THU THẬP DỮ LIỆU TRÊN NỀN TẢNG PYSPARK VÀ ỨNG DỤNG**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KỸ THUẬT PHẦN MỀM**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**TRẦN PHƯƠNG NGỌC ANH**

**TRƯƠNG NGUYỄN NGỌC CHÂU**

**PHÁT TRIỂN HỆ THỐNG XỬ LÝ SONG SONG THU THẬP DỮ LIỆU TRÊN NỀN TẢNG PYSPARK VÀ ỨNG DỤNG**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN 2**

**KỸ THUẬT PHẦN MỀM**

Người hướng dẫn

**PhGS Lê Anh Cường**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

LỜI CẢM ƠN

Trong quá trình thực hiện dự án công nghệ thông tin 2, không thể không nhắc đến sự quan tâm và hướng dẫn tận tình của giảng viên hướng dẫn – thầy Lê Anh Cường. Chúng em xin gửi lời cảm ơn chân thành đến thầy, người truyền tải những kiến thức và kinh nghiệm dẫn dắt chúng em hoàn thành dự án một cách tốt nhất. Bên cạnh đó, những lời hướng dẫn và dạy bảo tận tình của thầy còn là nguồn tri thức quý báu cho chúng em tích lũy thêm nhiều kiến thức mới trên con đường hoàn thiện chính mình.

Do vốn hiểu biết vẫn còn hạn chế nên trong quá trình thực hiện không thể tránh khỏi có những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp của thầy để bài làm được hoàn thiện hơn.

Một lần nữa chúng em xin chân thành cảm ơn.

*TP.Hồ Chí Minh, ngày 23 tháng 07 năm 2021*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của PhGS Lê Anh Cường;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 23 tháng 07 năm 2021*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Phương Ngọc Anh*

*Trương Nguyễn Ngọc Châu*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Đề tài phát triển hệ thống xử lý song song thu thập dữ liệu (tin tức và văn bản) các cơ quan bộ ngành trên nền tảng PySpark và ứng dụng bao gồm các phần như sau:

1. Thu thập dữ liệu của các cơ quan bộ ngành: crawl các loại tin tức, văn bản pháp quy và văn bản điều hành từ trang web chính thức của các cơ quan bộ ngành.
2. Xây dựng hệ thống xử lý song song trên nền tảng PySpark.
3. Phát triển ứng dụng bằng cách tích hợp tích hợp hệ thống xử lý song song thu thập dữ liệu vào trang web để hiển thị dữ liệu được crawl.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc78403282)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc78403283)

[TÓM TẮT iv](#_Toc78403284)

[MỤC LỤC 1](#_Toc78403285)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc78403286)

[CHƯƠNG 1 – GIỚI THIỆU 6](#_Toc78403287)

[1.1 Giới thiệu đề tài 6](#_Toc78403288)

[1.2 Phạm vi nghiên cứu 6](#_Toc78403289)

[CHƯƠNG 2 – TỔNG QUAN 7](#_Toc78403290)

[2.1 Tổng quan hệ thống 7](#_Toc78403291)

[2.2 Người dùng hệ thống 7](#_Toc78403292)

[2.3 Mô tả chức năng hệ thống 7](#_Toc78403293)

[2.4 Công nghệ sử dụng 8](#_Toc78403294)

[2.4.1 Python 8](#_Toc78403295)

[2.4.1.1 Tính năng chính của Python 9](#_Toc78403296)

[2.4.1.2 Ứng dụng của Python 9](#_Toc78403297)

[2.4.2 Scrapy 10](#_Toc78403298)

[2.4.3 SQL Server 13](#_Toc78403299)

[2.4.3.1 Lịch sử phát triển 14](#_Toc78403300)

[2.4.3.2 Đặc điểm và các ấn bản của SQL Server 14](#_Toc78403301)

[2.4.3.3 Mục đích sử dụng SQL Server 15](#_Toc78403302)

[2.4.3.4 Các câu lệnh và mệnh đề truy vấn trong SQL 15](#_Toc78403303)

[2.4.4 Web - Django 17](#_Toc78403304)

[2.4.4.1 Ưu điểm 18](#_Toc78403305)

[2.4.4.2 Nhược điểm 19](#_Toc78403306)

[2.4.4.3 Mô hình MTV (Model - Template - View) 19](#_Toc78403307)

[2.4.4.4 Cấu trúc thư mục của một project Django 20](#_Toc78403308)

[2.4.4.5 So sánh Django và Flask 22](#_Toc78403309)

[2.4.4.6 Lí do nên chọn Django 23](#_Toc78403310)

[2.4.4.7 Ứng dụng framework Django 24](#_Toc78403311)

[2.4.5 PySpark 25](#_Toc78403312)

[2.4.5.1 Spark 25](#_Toc78403313)

[2.4.5.2 Dataframe 26](#_Toc78403314)

[2.4.5.3 Kỹ thuật xử lý song song 27](#_Toc78403315)

[CHƯƠNG 3 – THỰC NGHIỆM 29](#_Toc78403316)

[3.1 Các sơ đồ đặc tả hệ thống 29](#_Toc78403317)

[3.1.1 Sơ đồ usecase 29](#_Toc78403318)

[3.1.2 Sơ đồ lớp 30](#_Toc78403319)

[3.1.3 Sơ đồ ERD 31](#_Toc78403320)

[3.2 Phân tích dữ liệu: 31](#_Toc78403321)

[3.3 Phân tích CSDL 32](#_Toc78403322)

[3.4 Ứng dụng thực tiễn 39](#_Toc78403323)

[CHƯƠNG 4 – TỔNG KẾT 49](#_Toc78403324)

[4.1 Kết quả thu được 49](#_Toc78403325)

[4.2 Hướng phát triển 49](#_Toc78403326)

**DANH MỤC CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

API Application Programming Interface

ASGI Asynchronous Server Gateway Interface

CI/CD Continuous delivery and/or Continuous deployment

CMS Content Management System

CPU Central Processing Unit

CSDL Cơ sở dữ liệu

DCL Data control language

DDL Data definition language

DML Data manipulation language

DRY Don’t Repeat Yourself

ERD Entity–relationship model

ETL Extract - Transform - Load

GPS Global Positioning System

MTV Model - Template - View

MVC Model View Controller

OOP Object-oriented programming

ORM Object-relational mapping

RAM Random Access Memory

RSS Really Simple Syndication

SSAS SQL Server Analysis Services

SSIS SQL Server Integration Services

SSRS SQL Server Reporting Services

SQL Structured Query Language

URL Uniform Resource Locator

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Minh họa ứng dụng của Python 10](#_Toc78403257)

[Hình 2.2: Hình ảnh thư viện Scrapy trong Python 11](#_Toc78403258)

[Hình 2.3: Hình ảnh Microsoft SQL Server 13](#_Toc78403259)

[Hình 2.4: Hình ảnh Django Python 18](#_Toc78403260)

[Hình 2.5: Hình ảnh mô hình MTV (Model - Template - View) Django 20](#_Toc78403261)

[Hình 2.6: Ví dụ cấu trúc thư mục của một project Django 21](#_Toc78403262)

[Hình 2.7: Khởi chạy dự án Django 22](#_Toc78403263)

[Hình 2.8: Minh họa project sử dụng Django 24](#_Toc78403264)

[Hình 2.9: Ưu điểm của Spark 26](#_Toc78403265)

[Hình 2.10: Hình minh họa đoạn code sử dụng Dataframe 27](#_Toc78403266)

[Hình 3.1: Sơ đồ usecase - Chức năng user 29](#_Toc78403267)

[Hình 3.2: Sơ đồ usecase - Chức năng admin 29](#_Toc78403268)

[Hình 3.3: Sơ đồ lớp 30](#_Toc78403269)

[Hình 3.4: Sơ đồ ERD 31](#_Toc78403270)

[Hình 3.5: Tổng quan CSDL 33](#_Toc78403271)

[Hình 3.6: Trang hiển thị tin tức trên hệ thống 40](#_Toc78403272)

[Hình 3.7: Trang hiển thị văn bản trên hệ thống 41](#_Toc78403273)

[Hình 3.8: Phần cuối trang của hệ thống 42](#_Toc78403274)

[Hình 3.9: Phần đầu trang bài viết chi tiết 42](#_Toc78403275)

[Hình 3.10: Phần cuối trang bài viết chi tiết 43](#_Toc78403276)

[Hình 3.11: Trang văn bản chi tiết 43](#_Toc78403277)

[Hình 3.12: Mục phân loại của hệ thống 44](#_Toc78403278)

[Hình 3.13: Trang hiển thị kết quả tìm kiếm theo phân loại 45](#_Toc78403279)

[Hình 3.14: Trang hiển thị kết quả tìm kiếm theo từ khóa 46](#_Toc78403280)

[Hình 3.15: Đoạn code áp dụng Pyspark để tìm kiếm dữ liệu 48](#_Toc78403281)

**DANH MỤC BẢNG**

[Bảng 2.1: Các câu lệnh cơ bản trong SQL Server 16](#_Toc77675780)

[Bảng 2.2: Các mệnh đề SQL cơ bản 17](#_Toc77675781)

[Bảng 2.3: Mô tả thành phần mô hình MTV 20](#_Toc77675782)

[Bảng 2.4: Bảng so sánh Flask và Django 22](#_Toc77675783)

[Bảng 2.5: Tiêu chí chọn framework 23](#_Toc77675784)

[Bảng 3.1: Khả năng thu thập dữ liệu của trang web 31](#_Toc77675785)

[Bảng 3.2: Bảng ministry\_info 32](#_Toc77675786)

[Bảng 3.3: Bảng article\_category\_type 32](#_Toc77675787)

[Bảng 3.4: Bảng legislation\_category\_type 33](#_Toc77675788)

[Bảng 3.5: Bảng article\_category\_info 33](#_Toc77675789)

[Bảng 3.6: Bảng legislation\_category\_info 34](#_Toc77675790)

[Bảng 3.7: Bảng ministry\_article\_category\_configuration 34](#_Toc77675791)

[Bảng 3.8: Bảng ministry\_article\_detail\_configuration 35](#_Toc77675792)

[Bảng 3.9: Bảng ministry\_legislation\_category\_configuration 35](#_Toc77675793)

[Bảng 3.10: Bảng ministry\_legislation\_detail\_configuration 36](#_Toc77675794)

[Bảng 3.11: Bảng ministry\_legislation\_detail\_configuration 37](#_Toc77675795)

[Bảng 3.12: Bảng legislation\_info 38](#_Toc77675796)

CHƯƠNG 1 – GIỚI THIỆU

1. Giới thiệu đề tài

Trong thời đại mà các thông tin gây nhiễu xuất hiện tràn lan và khó kiểm soát, ta sẽ khó có thể nắm bắt được những tin thật từ những trang tin tức chính thống từ các cơ quan bộ ngành. Để mọi người có thể cập nhật kịp thời các tin tức và các văn bản mới, chúng em cho ra đời hệ thống thu thập dữ liệu từ trang web của các tổ chức, cơ quan thuộc các bộ và các ngành, từ đó đảm bảo việc lưu trữ, xử lý và truyền tải các thông tin được nhanh chóng và an toàn cho đọc giả.

Tuy nhiên, hiệu suất hoạt động của một hệ thống với dữ liệu khổng lồ như vậy cũng là mối quan tâm hàng đầu của các nhà phát triển.

1. Phạm vi nghiên cứu

Ứng dụng được xây dựng bằng ngôn ngữ lập trình Python dựa trên nền tảng Django - một trong những framework phổ biến hỗ trợ xây dựng web đơn giản và tiện lợi. Bên cạnh đó, để phục vụ cho việc thu thập dữ liệu thì Scrapy là một ứng cử viên sáng giá có khả năng hoàn thành nhiệm vụ một cách hiệu quả và dễ dàng. Dữ liệu thu thập được lúc này sẽ được lưu trữ và tương tác với trang web dưới dạng CSDL thông qua SQL Server. Để đáp ứng nhu cầu tìm kiếm thông tin nhanh chóng và chuẩn xác nhất có thể, PySpark chính là một công cụ hỗ trợ đắc lực đó. Là thư viện mã nguồn mở, Pyspark hỗ trợ các lập trình viên xử lý song song đối với hệ thống các dữ liệu quy mô lớn, là một tiện ích mở rộng tuyệt vời của ngôn ngữ lập trình python.

CHƯƠNG 2 – TỔNG QUAN

* 1. Tổng quan hệ thống

Ứng dụng thu thập dữ liệu nhằm mục đích thu thập những bài báo và những văn bản mới nhất được đăng tải ở trang web của các bộ ngành. Mục đích để cho người dùng có thể cập nhật tin tức chính thống nhanh và chính xác hơn. Ứng dụng được xây dựng trên nền tảng web với các chức năng:

* Thu thập thông tin từ các trang web của bộ sau mỗi 12 tiếng, hệ thống sẽ thực hiện lưu trữ thông tin các bài báo mới và các văn bản mới sau khi thu thập thành công.
* Phân loại tin tức: người dùng có thể dễ dàng tìm kiếm các thông tin dễ dàng dựa trên phân loại giữa các bộ ngành.
* Tìm kiếm: tìm kiếm văn bản, bài báo theo từ khóa người dùng nhập vào.
* Hiển thị chi tiết bài báo: hiển thị hình ảnh và nội dung bài báo/văn bản thu thập được từ trang web các bộ ngành, đồng thời hiển thị trang web nguồn chính thức của các bài báo.
  1. Người dùng hệ thống

Hệ thống thu thập và lưu trữ dữ liệu các bộ ngành với người dùng được chia thành 2 vai trò:

* Admin (Quản lý hệ thống): Kiểm soát các hoạt động thu thập và hiển thị dữ liệu lên trang web.
* User (Người dùng): Sử dụng trang web để xem và tìm kiếm thông tin, tin tức về các bộ ngành.
  1. Mô tả chức năng hệ thống
* Crawl thông tin: trang web sử dụng thư viện Scrapy của Python để thu thập thông tin của các bài báo và văn bản. Ở các bài báo, bộ thu thập sẽ lấy các thông tin như: hình ảnh, nội dung, tiêu đề, thời gian, tác giả. Ở các văn bản bộ thu thập sẽ lấy các thông tin như: số hiệu văn bản, tên, ngày ký, ngày có hiệu lực, trích yếu, người ký, cơ quan ban hành, tình trạng, thể loại và link download. Các thông tin sẽ được lấy dưới dạng xpath qua các trang web chính thống của bộ.
* Phân loại: các tin tức sau khi thu thập sẽ được đưa lên web, hiển thị theo các loại tin hoặc các loại văn bản. Ngoài ra các bài báo và văn bản được phân loại theo từng bộ - ngành ban hành. Người dùng có thể sử dụng chức năng này để tìm thông tin cần thiết một cách thuận tiện nhất.
* Tìm kiếm: người dùng có thể tìm kiếm thông tin bằng cách nhập chi tiết tiêu đề hoặc các thuộc tính của bài báo hay thông qua các từ khóa nhập vào.
* Hiển thị thông tin chi tiết: trên trang web sẽ hiển thị thông tin nội dung, hình ảnh, ngày đăng và tác giả như trên trang web của chính phủ. Người dùng có thể xem thông tin ngay trên hệ thống mà không cần di chuyển đến trang web chính.
  1. Công nghệ sử dụng
     1. Python

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao, ngữ nghĩa động được giới thiệu vào năm [1991](https://vi.wikipedia.org/wiki/1990). Với ưu điểm dễ đọc, dễ học và dễ nhớ, Python là ngôn ngữ phù hợp cho những người mới bắt đầu tiếp xúc với việc lập trình. Cấu trúc được thiết kế gọn và rõ ràng của Python giúp người viết mã nhanh chóng và đơn giản hơn. Cú pháp dễ hiểu cùng với cách gõ linh động đã đưa Python nhanh chóng trở thành một ngôn ngữ mạnh mẽ và lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực.[27]

Python hỗ trợ nhiều mô hình lập trình bao gồm lập trình có cấu trúc, hướng đối tượng và chức năng. Python được thiết kế để có thể mở rộng thêm với các module. Tính module nhỏ gọn này đã khiến nó trở nên đặc biệt phổ biến như một phương tiện bổ sung các giao diện vào các ứng dụng hiện có. Mục tiêu của Python là để trở thành một ngôn ngữ có core nhỏ với thư viện tiêu chuẩn lớn và trình thông dịch có thể mở rộng dễ dàng [3]

* + - 1. Tính năng chính của Python
* Đơn giản, dễ học: cú pháp đơn giản, dễ đọc dễ viết hơn C++, Java hay C#, giúp lập trình viên có thể dành nhiều thời gian tập trung vào dự án hơn. Đây chính là đặc điểm chính trong ý tưởng thiết kế ngôn ngữ mở rộng của tác giả.
* Mã nguồn mở: Python có một cộng đồng lớn (CodeAcademy, StackOverflow), liên tục được đổi mới qua mỗi lần cập nhật và có thêm nhiều đặc điểm mới.
* Chạy đa nền tảng: Windows, macOS, Linux.
* Khả năng mở rộng nhúng: Đối với các ứng dụng phức tạp, lập trình viên có thể sử dụng các đoạng code viết bằng những ngôn ngữ khác (Java, C#, C++,…) vào dự án của mình.
* Ngôn ngữ thông dịch cấp cao: có khả năng tự động chuyển đổi code sang ngôn ngữ máy mà không cần thông qua một bước trung gian là thông dịch, giảm bớt một bước cho người lập trình viên. Đồng thời cũng giúp giảm thiểu những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,...
* Thư viện tiêu chuẩn lớn: hỗ trợ xử lý những bài toán phổ biến, giảm thiểu thời gian giải quyết vấn đề. [3]
  + - 1. Ứng dụng của Python
* Lập trình ứng dụng web: có thể tạo ra hệ thống các trang web có khả năng mở rộng (scalable) bằng cách sử dụng framework và CMS (hệ thống quản trị nội dung). Một số công nghệ được sử dụng nhiều: Django, Flask, Pyramid, Plone,...
* Khoa học và số liệu ứng dụng: Python có nhiều thư viện được thiết kế để phân tích thống kê và phân tích số liệu (SciPy, NumPy, EarthPy - khoa học trái đất, AstroPy - Thiên văn học,... ). Ngoài ra, Python cũng được ứng dụng nhiều trong lĩnh vực học máy, trí tuệ nhân tạo, xử lý ngôn ngữ tự nhiên,… (TensorFlow, Keras, Pytorch)



Hình 2.: Minh họa ứng dụng của Python

(Nguồn: https://bachkhoa-aptech.edu.vn/kham-pha-ngay-3-ung-dung-noi-bat-cua-python/549.html)

* Tạo phần mềm nguyên mẫu: Python là ngôn ngữ thích hợp để tạo những nguyên mẫu - bản chạy thử.
* Phù hợp để đào tạo lập trình: Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình nhờ tính đơn giản và dễ sử dụng của ngôn ngữ này.
* Cú pháp đơn giản: Là ngôn ngữ lập trình bậc cao, những đoạn code viết bằng Python rất gần với ngôn ngữ tự nhiên nên rất dễ hiểu để sử dụng.
* Không quá khắt khe: Trong số những tính năng dễ sử dụng của Python chính là các kiểu dữ liệu không cần phải khai báo trước, không cần thêm dấu chấm phẩy vào cuối mỗi dòng code, sử dụng dấu hai chấm để định nghĩa các đoạn mã trong hàm, lớp … [3]
  + 1. Scrapy

Scrapy là một thư viện mã nguồn mở hỗ trợ thu thập dữ liệu từ trang web viết bằng ngôn ngữ Python. Ban đầu, Scrapy được thiết kể để tìm kiếm các trang web, dần dần nó còn được sử dụng để trích xuất dữ liệu bằng cách dùng API hoặc một hệ thống thu thập thông tin web. Khi ta phát triển một web cần lấy thông tin từ các bài báo ở trang web khác, thì ta sẽ dùng scrapy để làm điều đó.



Hình 2.: Hình ảnh thư viện Scrapy trong Python

(Nguồn: https://ichi.pro/vi/cach-cao-bang-scrapy-cua-tran-va-danh-bai-captcha-124640164858391)

Các thành phần của Scrapy như sau:

* Scrapy Engine: có nhiệm vụ kiểm soát dữ liệu giữa các thành phần trong hệ thống và đồng thời sẽ kích hoạt một số sự kiện khi có một hành động nào đó xảy ra.
* Scheduler: Giúp sắp xếp thứ tự các URL mà chúng ta cần lấy dữ liệu về.
* Downloader: Thực hiện việc download dữ liệu và truyền dữ liệu ấy cho Scrapy Engine.
* Spiders: Đây là lớp mà chúng ta sẽ định nghĩa để thực hiện công việc tách dữ liệu cần thiết khi download về và có thể cung cấp các url mới để thêm vô phần Scheduler.
* Item pipeline: Sau khi dữ liệu được lấy và qua xử lý thì sẽ được đưa vào đây và item pipeline có nhiệm vụ xử lý và lưu vào CSDL.
* Middlewares: Là các thành phần nằm giữa Scrapy Engine và các thành phần khác. Chúng giúp ta có thể mở rộng khả năng xử lý cho các thành phần.
  + Spider middlewares: Nằm giữa Scrapy Engine và Spiders, chúng xử lý các dữ liệu trước và sau khi đưa vào Spiders.
  + Downloader middlewares: Nằm giữa Scrapy Engine và Downloader, ở đây chúng xử lý các request được đẩy vào Downloader từ Scrapy Engine và đồng thời xử lý các response từ Downloader.
  + Scheduler middlewares: Nằm giữa Scrapy Engine và Scheduler để xử lý các request giữa hai thành phần. [16]

Cách hoạt động của Scrapy như sau:

1. Khi muốn lấy dữ liệu từ một website thì Scrapy Engine sẽ yêu cầu Spiders cung cấp danh sách mà người dùng đã nạp.
2. Sau khi Scrapy Engine có được danh sách URL thì sẽ gửi cho Scheduler để sắp xếp thứ tự các url cần tải dữ liệu về.
3. Sau đó Scrapy Engine nhận được nhanh sách các url theo thứ tự từ Scheduler và đưa danh sách ấy vào Downloader thông qua Downloader Middleware (theo hướng đầu vào).
4. Downloader nhận được các URL và bắt đầu thực hiện tải dữ liệu về, cứ sau mỗi một URL tải xong sẽ tạo ra một response trả về cho Scrapy Engine thông qua Downloader Middleware (theo hướng đầu ra).
5. Sau khi Scrapy Engine nhận được kết quả trả về thì sẽ đẩy kết quả ấy cho Spiders thông qua Spider Middleware (theo hướng đầu vào) để xử lý kết quả trả về.
6. Spiders sẽ xử lý dữ liệu và trả lại kết quả sau khi xử lý đồng thời sẽ cung cấp các URL mới có thể được nạp vào và đẩy lại cho Scrapy Engine thông qua Spider Middleware (theo hướng đầu ra).
7. Scrapy Engine sau khi nhận được kết quả trả về từ Spiders thì sẽ thực hiện hai việc là một là đẩy dữ liệu sau khi được xử lý cho Item Pipeline để xử lý và lưu vào CSDL, hai là đẩy danh sách các url mới cho Scheduler và thực hiện lại bước thứ 3.
   * 1. SQL Server

SQL là ngôn ngữ truy vấn có cấu trúc, hỗ trợ truy xuất và thực hiện thao tác trên CSDL.

SQL Server là hệ quản trị CSDL do Microsoft phát triển. Là một máy chủ CSDL, SQL Server có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng khác. Nó cung cấp các câu lệnh rất đa dạng và hữu ích, có thể được sử dụng cùng nhau khi truy vấn dữ liệu. Các câu lệnh của SQL thực hiện các công việc như thêm, xóa, chỉnh sửa dữ liệu có thể với một hoặc nhiều điều kiện. [3][4][6][14]



Hình 2.: Hình ảnh Microsoft SQL Server

(Nguồn: <https://www.semtek.com.vn/sql-server-2/>)

* + - 1. Lịch sử phát triển

SQL Server ban đầu được sử dụng trong để quản lý CSDL và chạy trên các máy riêng biệt. Song do sự phát triển nhanh chóng của những nhu cần xây dựng những CSDL lớn theo mô hình client-server (toàn bộ CSDL được tập trung trên máy chủ). Tất cả tác thao tác lên dữ liệu đều diễn ra trên máy chủ bằng các lệnh SQL, máy trạm chỉ dùng để cập nhật hoặc lấy thông tin từ máy chủ. Ngày nay, ngôn ngữ SQL còn được kết hợp và sử dụng với các ngôn ngữ bậc cao khác giúp việc tạo các trang web động nhanh chóng hơn. [8][9][25]

* + - 1. Đặc điểm và các ấn bản của SQL Server

Về đặc điểm, SQL cung cấp đa dạng các câu lệnh cho các công việc truy vấn dữ liệu:

* Thêm, sửa, xoá các dòng dữ liệu.
* Tạo, sửa đổi, thêm và xoá các đối tượng trong CSDL.
* Điều khiển việc truy xuất tới CSDL và các đối tượng để đảm bảo tính bảo mật của CSDL.
* Đảm bảo tính nhất quán và các ràng buộc của CSDL.

Đối tượng của SQL là các bảng (các quan hệ dữ liệu 2 chiều). Mỗi bảng thường có nhiều cột (trường) và nhiều dòng (bản ghi). Các cột được đặt tên và được xác định bằng kiểu dữ liệu cụ thể tạo nên cấu trúc bảng.

Các ấn bản của SQL Server bao gồm:

* Enterprise: là phiên bản cao cấp, cung cấp đầy đủ các tính năng của trung tâm dữ liệu
* Standard: cung cấp cơ sở dữ liệu quản lý dữ liệu cơ bản và cơ sở dữ liệu thông minh, phù hợp cho các công ty vừa và nhỏ, cho phép quản lý CSDL hiệu quả với tài nguyên công nghệ thông tin tối thiểu. Phiên bản này bị giới hạn một số tính năng nâng cao (advanced features) so với bản Enterprise.
* Developer: Bao gồm tính năng tương tự như phiên bản Enterprise nhưng được cấp phép để sử dụng như một hệ thống phát triển và thử nghiệm, không như một máy chủ doanh nghiệp. Phiên bản này phù hợp cho các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng.
* Express: SQL Server Express thích hợp để quản trị CSDL đơn giản. Là CSDL miễn phí, lý tưởng để học và xây dựng các ứng dụng dữ liệu nhỏ trên các máy tính cá nhân, Là sự lựa chọn tốt cho các nhà cung cấp phần mềm độc lập, lập trình viên. Đặc điểm là lưu trữ an toàn và triển khai nhanh chóng. Có thể nâng cấp thành các phiên bản khác có nhiều tính năng hơn. [10]
  + - 1. Mục đích sử dụng SQL Server

Thông thường, SQL Server được sử dụng để:

* Tạo CSDL.
* Duy trì CSDL.
* Phân tích dữ liệu bằng SSAS – SQL Server Analysis Services.
* Tạo báo cáo bằng SSRS – SQL Server Reporting Services.
* Thực hiện quá trình ETL (Extract-Transform-Load) bằng SSIS – SQL Server Integration Services. [9]
  + - 1. Các câu lệnh và mệnh đề truy vấn trong SQL
* Các câu lệnh cơ bản trong SQL Server gồm:

|  |  |  |
| --- | --- | --- |
| **Tính chất** | **Câu lệnh** | **|Chức năng** |
| DDL - Data Definition Language  (Ngôn ngữ định nghĩa dữ liệu) | Create | Tạo bảng/view mới |
| Alter | Chỉnh sửa bảng/view |
| Drop | Xóa bảng/view |
| DML - Data Manipulation Language  (Ngôn ngữ thao tác dữ liệu) | Select | Trích xuất dữ liệu |
| Insert | Chèn dữ liệu |
| Update | Cập nhật dữ liệu |
| Delete | Xóa dữ liệu |
| DCL - Data Control Language  (Ngôn ngữ kiểm soát dữ liệu) | Grant | Cấp quyền |
| Revoke | Lấy lại quyền đã cấp |

Bảng 2.: Các câu lệnh cơ bản trong SQL Server

(Nguồn: http://sqladvice.com/tong-quan-ve-sql/#Dac\_diem\_va\_doi\_tuong\_cua\_SQL)

* Các mệnh đề truy vấn cơ bản gồm:

|  |  |  |
| --- | --- | --- |
| **Mệnh đề truy vấn** | **Chức năng** | **Cú pháp** |
| Where | Xác định điều kiện khi thao tác với dữ liệu | Select col1, col2  From table\_name  Where [conditions] |
| Select | Trích xuất dữ liệu từ bảng, trả về các tập hợp kết quả ở dạng bảng | Select col1, col2  From table\_name |
| Order By | Sắp xếp dữ liệu theo thứ tự trên cùng hoặc nhiều cột | Select col1, col2  From table\_name  Where [conditions]  Order By col1, col2,… [asc/desc] |
| Group By | Sắp xếp dữ liệu đồng nhất vào trong các nhóm | Select col1, col2  From table\_name  Where [conditions]  Group By col1, col2  Order By col1, col2,… [asc/desc] |
| Like | Tìm kiếm mẫu xác định trong cột | Select col1, col2  From table\_name  Where [conditions]  Like pattern |

Bảng 2.: Các mệnh đề SQL cơ bản

(Nguồn: <http://sqladvice.com/tong-quan-ve-sql/#Dac_diem_va_doi_tuong_cua_SQL)>

* + 1. Web - Django

Django là một trong số những [framework](https://bizfly.vn/techblog/web-framework-la-gi.html) mã nguồn mở miễn phí, được viết bằng ngôn ngữ Python dựa trên mô hình mô hình MTV (gồm Model-Template-Views). Hiện tại, framework này được phát triển, quản lý bởi Django Software Foundation.

Với Django, lập trình viên được hỗ trợ thiết kế các website phức tạp dựa trên những CSDL có sẵn. [12]



Hình 2.: Hình ảnh Django Python

(Nguồn: https://bizfly.vn/techblog/django-la-gi.html)

* + - 1. Ưu điểm

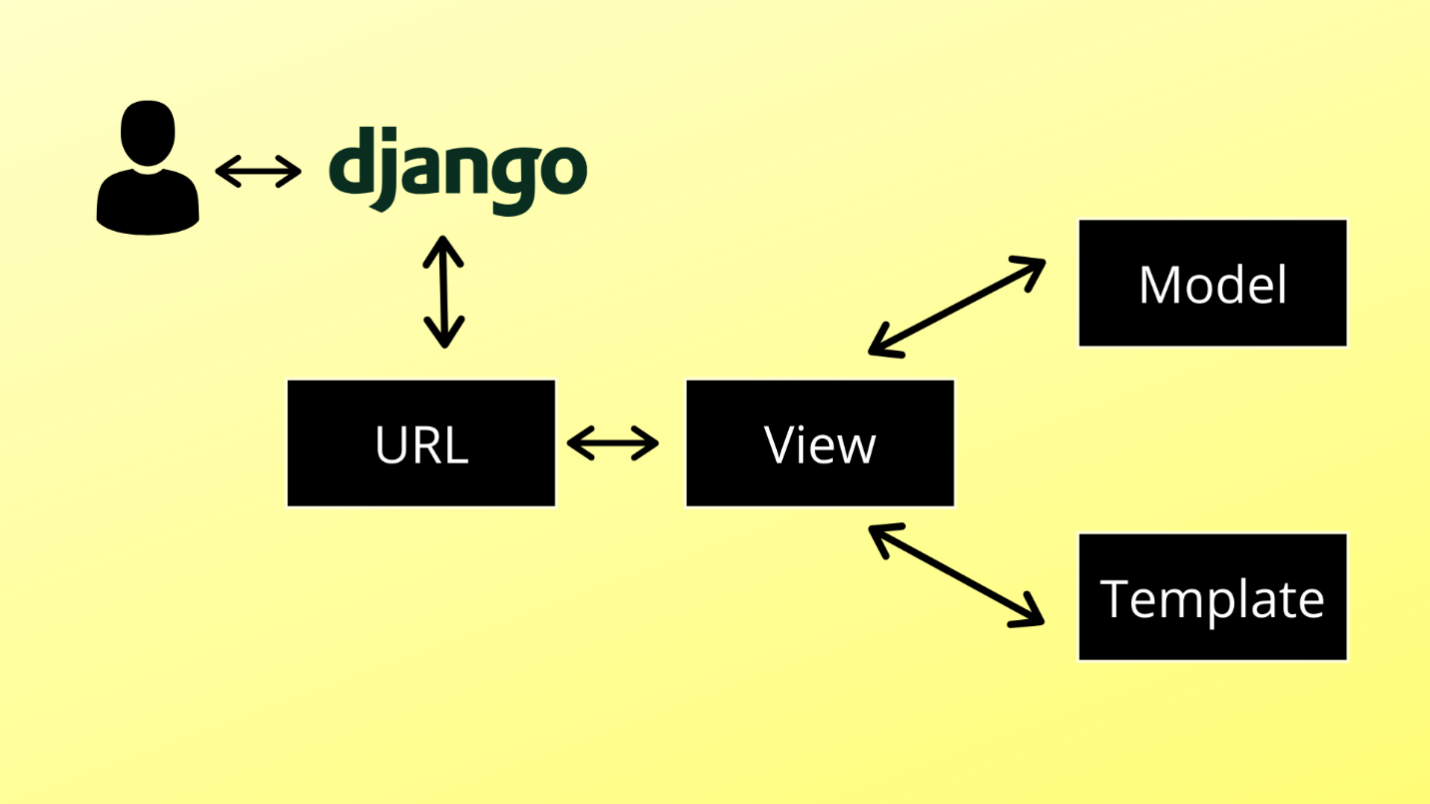
Lợi thế hàng đầu của Django là tốc độ xây dựng web. Ngoài ra, còn nhiều ưu điểm khác khiến nó trở thành framwork được ưa chuộng và sử dụng rộng rãi:

* Bảo mật tốt: Với các vấn đề an toàn mạng, lập trình viên có thể hoàn toàn tránh được các lỗi bảo mật thường gặp (SQL injection, cross-site script, clickjacking,…). Hệ thống xác thực người dùng của Django cung cấp một giải pháp an toàn để quản lý tài khoản và mật khẩu của người dùng.
* Mở rộng: Tính mở rộng có sẵn trong framework hỗ trợ tốt cho các lập trình viên trong việc quản lý lưu lượng người truy cập, thích hợp với các trang có traffic lớn.
* Linh hoạt: Django được viết bằng [Python](https://bizfly.vn/techblog/python-la-gi.html) và mô hình MVC nên rất dễ ứng dụng trong các dự án, hỗ trợ nhiều ngôn ngữ. Có thể dùng để xây dựng các hệ thống, mạng xã hội cho đến nền tảng khoa học máy tính.
* Tích hợp tính năng: ở Django có nhiều tính năng bổ sung để xử lý các tác vụ web thông thường: xác thực người dùng, quản trị nội dung, thiết kế url/view/template,… [12][19][23][24]
  + - 1. Nhược điểm

Bên cạnh những ưu điểm kể trên, Django cũng tồn tại một số nhược điểm sau đây:

* Thiếu quy ước: Django không có quy tắc nào về cấu hình để các nhà phát triển tuân theo, dễ dẫn đến các vấn đề phát sinh về cấu hình.
* Không thích hợp cho dự án nhỏ: Django yêu cầu nhiều mã hóa, làm mất thời gian xử lý của máy chủ và băng thông khi phát triển.
* Các vấn đề về request: Django không thể xử lý nhiều request đồng thời, mỗi request sẽ được xử lý riêng lẻ. [12][23]
  + - 1. Mô hình MTV (Model - Template - View)

Django sử dụng mô hình MTV tương tự như mô hình MVC trong các framework khác. Trong mô hình hoạt động của Django, người dùng gửi một request đến Django, tiếp đó Django hoạt động như một bộ điều khiển kiểm tra các tài nguyên có sẵn trong URL. Nếu các URL đúng, nó sẽ đưa các request từ các URL đến các Views khác nhau. View sẽ tiếp tục tương tác với Model và Template tương ứng. Django sẽ gửi Template đã được tạo để phản hồi tới người dùng. [14][17][18]



Hình 2.: Hình ảnh mô hình MTV (Model - Template - View) Django

(Nguồn: https://iq.opengenus.org/model-view-controller-django/)

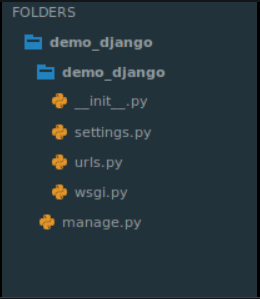
|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| Model | Lớp thiết kế CSDL mà Django ORM cung cấp những phương thức xử lý, nghiệp vụ lên CSDL |
| Template | Là những mẫu được tạo ra để xử lý output thành html/css cho trang web |
| Views | Các function để xử lý các request từ người dùng |

Bảng 2.: Mô tả thành phần mô hình MTV

(Nguồn: [https://vinasupport.com/django-la-gi-tong-quan-ve-django-framework/)](https://iq.opengenus.org/model-view-controller-django/))

* + - 1. Cấu trúc thư mục của một project Django

Cấu trúc thư mục ban đầu có dạng:



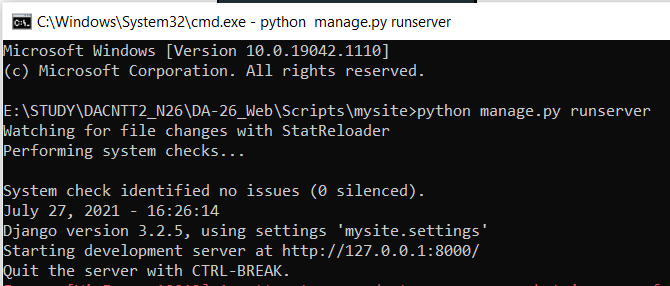
Hình 2.: Ví dụ cấu trúc thư mục của một project Django

(Nguồn: https://phonghopamway.com.vn/django-python-la-gi/)

Trong đó:

* \_\_init\_\_.py: Là tệp tin nguồn được thực thi đầu tiên, có nhiệm vụ thông báo cho trình biên dịch Python về các thư mục khác dưới dạng module.
* settings.py: Tập tin cấu hình dùng để quản lý và cài đặt về các thư viện, đường dẫn được sử dụng trong project.
* urls.py: Bao gồm tất cả các khai báo URL đến các views để xử lý request.
* wsgi.py: Đây là lối vào cho các máy chủ web tương thích WSGI.
* manage.py: Cho phép quản lý, tương tác với Django hoặc khởi tạo server. [14]

Để khởi chạy ứng dụng Django, người dùng cần truy cập đến folder có chứa file manage.py và chạy câu lệnh “python manage.py runserver”. Hệ thống sẽ cung cấp thông tin đường dẫn chạy server. Ví dụ trong hình 2.7, server được triển khai tại http://127.0.0.1:8000/.



Hình 2.: Khởi chạy dự án Django

* + - 1. So sánh Django và Flask

Flask và Django đều là những framework nổi tiếng được sử dụng rộng rãi để xây dựng các trang web bằng ngôn ngữ Python. Flask là một microframework tập trung chủ yếu vào sự tối giản và khả năng kiểm soát chi tiết. Mặc khác, Django cung cấp nhiều tính năng hơn: ORM, giao diện CSDL, cấu trúc thư mục,…

So sánh Flask và Django:

|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Flask** | **Django** |
| Tốc độ phát triển | Sự đơn giản của Flask cũng cho phép các nhà phát triển có kinh nghiệm tạo ra các ứng dụng nhỏ hơn trong thời gian ngắn | Có khả năng phát triển nhanh chóng các ứng dụng web phức tạp với đầy đủ các tính năng |
| Tính linh hoạt và kiểm soát | Flask không hạn chế các nhà phát triển, họ có thể dùng nhiều thư viện và tiện ích bổ sung bên ngoài, làm ứng dụng | Django có nhiều tính năng và module được tích hợp sẵn, ít linh hoạt và kiểm soát nhiều hơn |
| Cộng đồng | Cộng đồng chưa lớn bằng | Có cộng đồng lớn, có nhiều hỗ trợ hơn |
| Sự phát triển | Ra đời trễ hơn nên không có nhiều tiện ích hay tính năng được tích hợp | Là một framework ra đời sớm, có nhiều tiện ích mở rộng, plugin,… |

Bảng 2.: Bảng so sánh Flask và Django

(Nguồn: https://www.netguru.com/blog/flask-vs-django)

Mỗi framework đều có những đặc điểm riêng phù hợp với những dự án khác nhau. Vì vậy, tùy thuộc vào yêu cầu và những tiêu chí nhất định, ta có thể lựa chọn được framework thích hợp với mục tiêu của mình. [23]

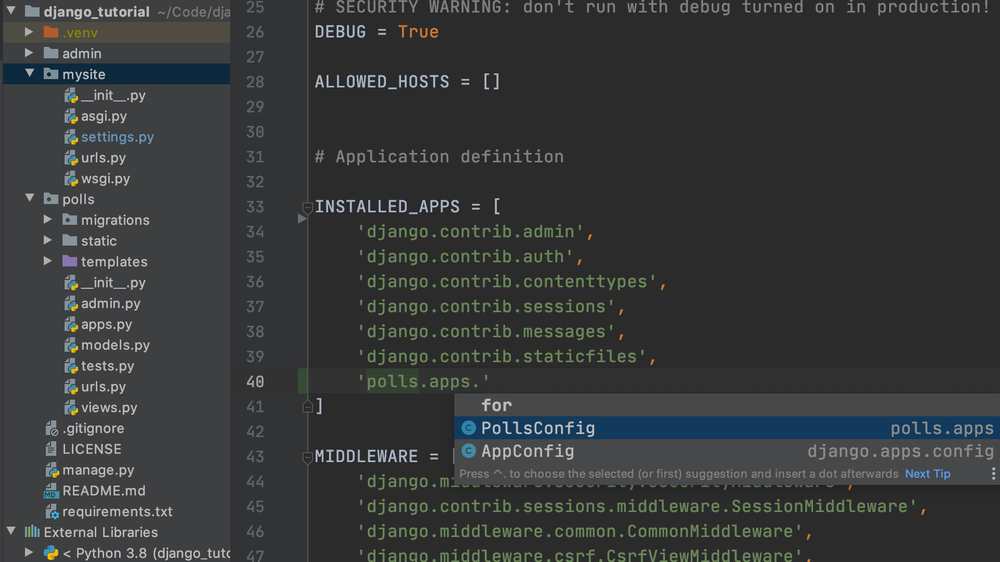
|  |  |  |
| --- | --- | --- |
| **Tiêu chí** | **Flask** | **Django** |
| Quy mô dự án | Phù hợp với ứng dụng nhỏ hơn, ít phức tạp. | Phù hợp với các ứng dụng lớn hơn, phức tạp hơn và tải trọng cao |
| Kết cấu | Phù hợp với nhu cầu về các công cụ và thư viện cụ thể hoặc muốn có một khuôn khổ có khả năng tùy biến cao | Phù hợp nếu không có yêu cầu điều chỉnh chi tiết với các tính năng tích hợp và các giá trị mặc định hợp lý |
| Loại dự án | Ứng dụng đơn giản với nội dung tĩnh | Cửa hàng trực tuyến, mạng xã hội |

Bảng 2.: Tiêu chí chọn framework

(Nguồn: https://www.netguru.com/blog/flask-vs-django)

* + - 1. Lí do nên chọn Django

So với các framework khác, các ưu điểm của Django là những điểm sáng giúp nó trở thành một framework lý tưởng để sử dụng.



Hình 2.: Minh họa project sử dụng Django

(Nguồn: https://bizfly.vn/techblog/django-la-gi.html)

* Cú pháp đơn giản.
* Máy chủ riêng.
* Sử dụng nguyên tắc DRY - không trùng lặp.
* Hỗ trợ ORM, tương thích khả năng hoạt động với các CSDL thường gặp như Oracle và SQL.
* Sở hữu nhiều tính năng, giải pháp độc đáo và hữu ích (GPS, chatbot, ứng dụng CSDL, nguồn cung cấp dữ liệu RSS,…) giúp phát triển, [quản trị website](https://bizfly.vn/techblog/quan-tri-website-la-gi.html), chăm sóc khách hàng,…
* Linh hoạt cao: Xây dựng được nhiều dạng website, ứng dụng phù hợp với nhiều mục đích khác nhau.
* Bảo mật tốt. [12]
  + - 1. Ứng dụng framework Django

Django được ứng dụng phổ biến, đặc biệt các nền tảng lớn như Instagram, Spotify, YouTube, Dropbox, trình duyệt Mozilla,… cũng sử dụng framework này. Những doanh nghiệp lớn trên thế giới đều đã và đang sử dụng Django cho website của mình.

Django giữ nhiều vai trò khác nhau như: hỗ trợ quản lý và xử lý dữ liệu, lưu trữ thông tin, hỗ trợ triển khai các thành phần mới,… Với các nền tảng lớn, có lượng người truy cập cao thì đây chính là một framework lý tưởng mang lại hiệu suất hoạt động vô cùng mạnh mẽ. [12]

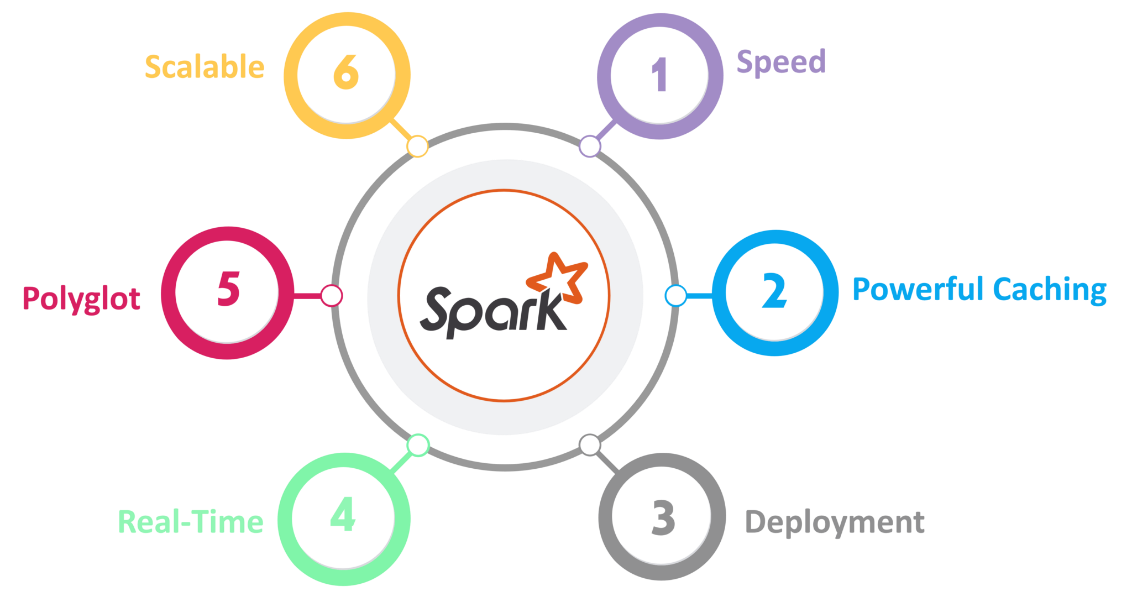
* + 1. PySpark

Trước khi nói đến Pyspark, ta cần tìm hiểu Spark là gì.

1. Spark

Spark, hay tên đầy đủ là Apache Spark, là một open source cluster computing framework, cho phép xây dựng các mô hình dự đoán nhanh chóng với việc tính toán được thực hiện trên một nhóm các máy tính có thể tính toán cùng lúc trên toàn bộ tập dữ liệu mà không cần phải trích xuất mẫu tính toán thử nghiệm. Tốc độ xử lý của Spark còn nhờ vào việc tính toán được thực hiện ở bộ nhớ trong hoặc trên RAM.

Với các tính năng nổi bật như xử lý nhanh, linh hoạt và thân thiện với người dùng, Apache Spark nhanh chóng trở thành nền tảng hàng đầu giúp phân phối và xử lý dữ liệu lớn trong thế giới kỹ thuật số đang phát triển mạnh đòi hỏi những sự tính toán khổng lồ. Cùng khả năng đảm nhiệm ứng dụng lập lịch, phân phối và giám sát, Spark cho phép xử lý phân tán các dữ liệu thành các phần nhỏ có thể quản lý được cho các công việc cần thiết, sau đó kết hợp chúng trở lại. [24]



Hình 2.: Ưu điểm của Spark

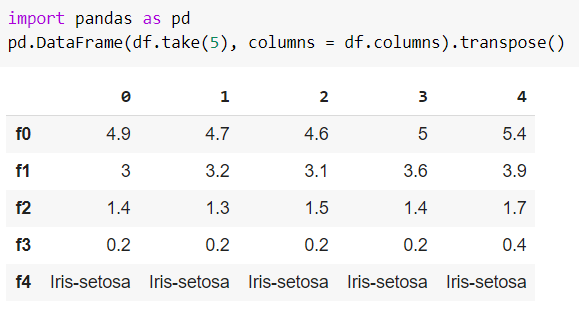
(Nguồn: https://www.edureka.co/blog/spark-architecture/)

Spark có thể được triển khai bằng nhiều ngôn ngữ lập trình như: Java, Scala, Python và R, đồng thời hỗ trợ SQL. Hiện nay, nó được ứng dụng bởi các ngân hàng, công ty viễn thông, công ty trò chơi và những gã công nghệ lớn như Apple, Facebook, Twitter, IBM và Microsoft, …

Như vậy, Spark là công cụ để thực hiện xử lý dữ liệu lớn ở cả quy mô lớn và tốc độ cao, trong khi PySpark là thư viện của Python để sử dụng Spark.

1. Dataframe

Khi nói đến Spark, không thể không kể đến Spark Dataframe. Spark Dataframe được thiết kế để xử lý một tập hợp lớn dữ liệu có cấu trúc cũng như bán cấu trúc (nhiều định dạng dữ liệu có thể đồng nhất hoặc không đồng nhất), được thể hiện qua các cột cung cấp các hoạt động để lọc, nhóm hoặc tính toán tổng hợp và có thể được sử dụng với Spark SQL. Về mặt khái niệm, nó tương đương với bảng trong cơ sở dữ liệu quan hệ, một trang tính Excel có tiêu đề cột hoặc một khung dữ liệu trong R / Python, nhưng với các tối ưu hóa phong phú hơn. [13] Các bảng Dataframe có thể được tạo ra từ nhiều cách như trích xuất từ SQL Database, CSV file, Excel file, …



Hình 2.: Hình minh họa đoạn code sử dụng Dataframe

Sự ra đời của DataFrame mở ra nhiều tiện ích cho người dùng Spark khiến nó nhanh chóng được ưa chuộng với nhiều ưu điểm như:

* Gồm các chức năng như select, where - chọn các hàng, cột và ô theo tên hoặc theo số, lọc ra các hàng, nhóm các dữ liệu, join với các DataFrames khác, tương tự như SQL.
* Sử dụng các phương thức như filter, select để trích xuất dữ liệu theo cột, hàng.
* Thêm cột dễ dàng với UDF (User Defined Function).
* Được hỗ trợ Pivot hữu ích cho việc lập bảng biểu, báo cáo. [13]

Từ đó ta thấy rằng những lợi ích mà DataFrames mang lại chính là tính đơn giản, dễ sử dụng, truy vấn tốt.

1. Kỹ thuật xử lý song song

Việc xử lý song song dữ liệu có gì khác so với cách xử lý thông thường (chạy vòng lặp,…), đặc biệt trong thời điểm các thông tin dần trở nên phức tạp và có số lượng khổng lồ như hiện nay.

Từ lâu, tính toán song song đã được xây dựng và áp dụng rất nhiều trong lĩnh vực tính toán hiệu năng. Nó là một hình thức tính toán trong đó nhiều phép tính được thực hiện đồng thời, hoạt động trên nguyên tắc chia các phần lớn thành nhiều phần nhỏ hơn, sau đó được giải quyết tương tranh. [15]

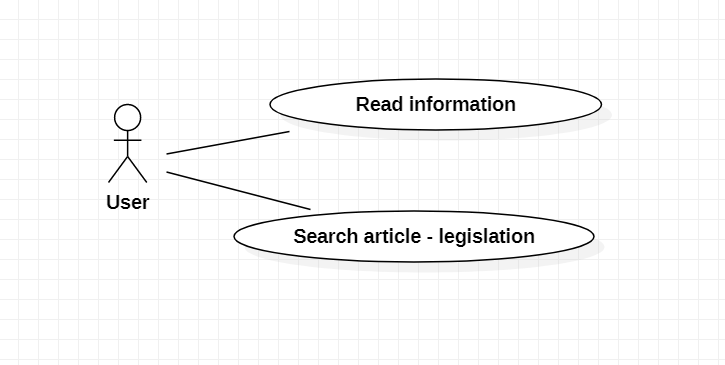
Trong khi các công việc tìm kiếm được thực thi riêng biệt và tuần tự thì việc xử lý song song, đồng thời sẽ giúp tiết kiệm, giảm thiểu thời gian xử lý tổng thể. Các câu lệnh lúc này sẽ được thực hiện cùng lúc trên nhiều bộ xử lý, đưa các tiến trình trở nên linh hoạt và truy xuất nhanh chóng hơn.

Chính vì vậy, kỹ thuật xử lý song song của Pyspark ra đời, là một phần không thể thiếu trong việc tìm kiếm giữa một khối dữ liệu và thông tin cực lớn ở bối cảnh cách mạng công nghiệp 4.0 hiện nay.

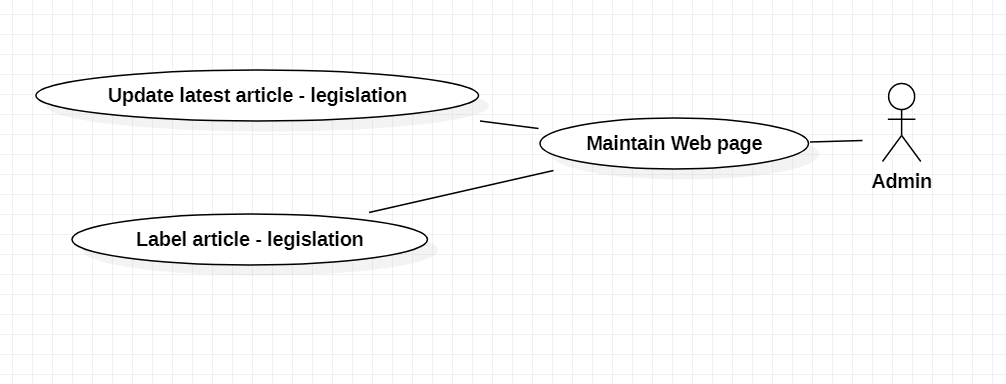
Ứng dụng kỹ thuật này khiến Spark trở thành một công cụ hữu ích cho người dùng qua các nhu cầu xử lý dữ liệu lớn nói chung.

CHƯƠNG 3 – THỰC NGHIỆM

* 1. Các sơ đồ đặc tả hệ thống
     1. Sơ đồ usecase

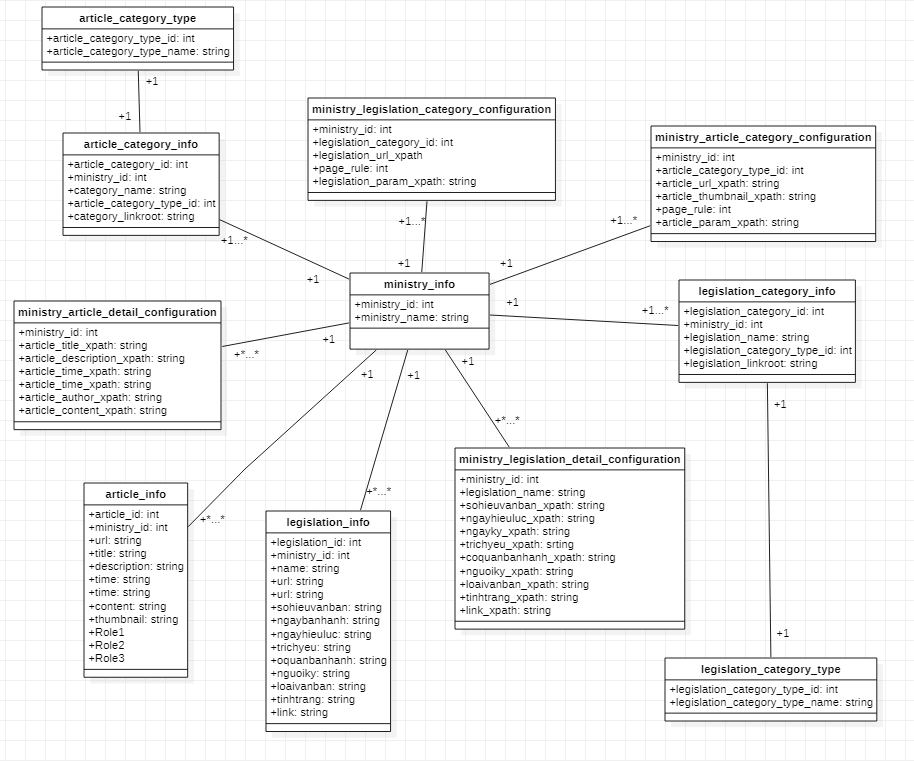


Hình 3.: Sơ đồ usecase - Chức năng user



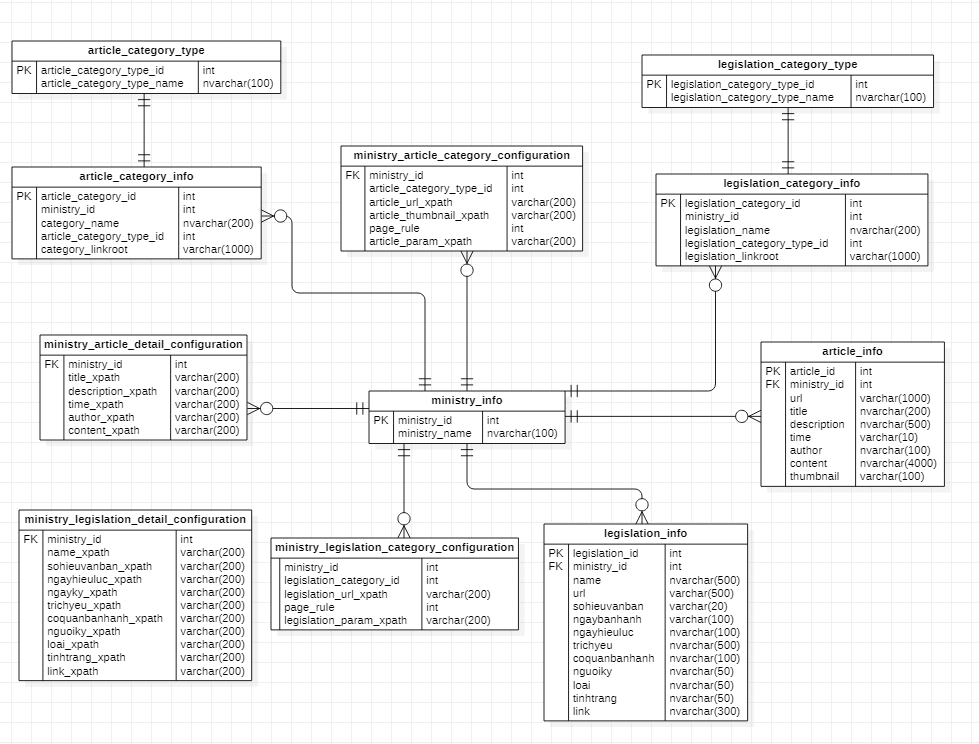
Hình 3.: Sơ đồ usecase - Chức năng admin

* + 1. Sơ đồ lớp



Hình 3.: Sơ đồ lớp

* + 1. Sơ đồ ERD



Hình 3.: Sơ đồ ERD

* 1. Phân tích dữ liệu:

Trong số tất cả các cơ quan thuộc bộ máy Chính phủ Việt Nam, sau khi tìm hiểu và thu thập dữ liệu, chúng em nhận thấy các dữ liệu thu thập được nằm ở các bộ/ ngành sau:

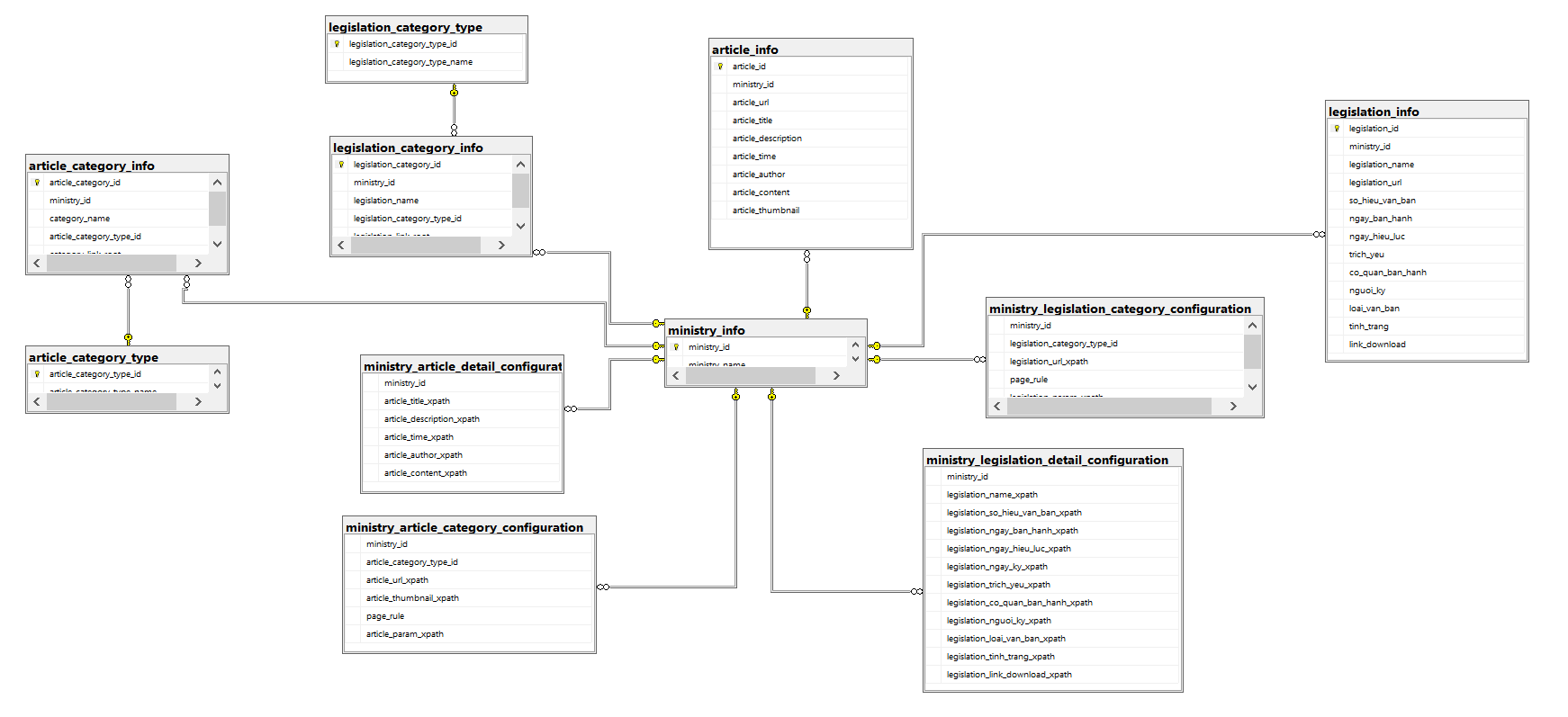
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Bộ/Ngành | Bài báo | Văn bản |
| 1 | [Bộ Công an](https://www.moha.gov.vn/danh-muc/bo-cong-an-9982.html) | x | x |
| 2 | [Bộ Giáo dục và Đào tạo](https://www.moha.gov.vn/danh-muc/bo-giao-duc-va-dao-tao-9997.html) | x | x |
| 3 | [Bộ Giao thông vận tải](https://www.moha.gov.vn/danh-muc/bo-giao-thong-van-tai-9990.html) | x | x |
| 4 | [Bộ Khoa học và Công nghệ](https://www.moha.gov.vn/danh-muc/bo-khoa-hoc-va-cong-nghe-9994.html) | x | x |
| 5 | [Bộ Lao động - Thương Binh và Xã hội](https://www.moha.gov.vn/danh-muc/bo-lao-dong-thuong-binh-va-xa-hoi-9995.html) | x | x |
| 6 | [Bộ Nội vụ](https://www.moha.gov.vn/danh-muc/bo-noi-vu-9984.html) | x | x |
| 7 | [Bộ Quốc phòng](https://www.moha.gov.vn/danh-muc/bo-quoc-phong-9981.html) | x | x |
| 8 | [Bộ Thông tin và Truyền thông](https://www.moha.gov.vn/danh-muc/bo-thong-tin-va-truyen-thong-9993.html) | x | x |
| 9 | [Bộ Văn hóa - Thể thao và Du lịch](https://www.moha.gov.vn/danh-muc/bo-van-hoa-the-thao-va-du-lich-9996.html) | x | x |
| 10 | [Bộ Xây dựng](https://www.moha.gov.vn/danh-muc/bo-xay-dung-9991.html) | x | x |
| 11 | Chính phủ | x |  |
| 12 | Ủy ban Dân tộc | x | x |
| 13 | Bảo hiểm Xã hội Việt Nam | x | x |
| 14 | Viện Hàn lâm Khoa học Xã hội Việt Nam | x |  |
| 15 | Ủy ban quản lý vốn nhà nước tại doanh nghiệp | x | x |

Bảng 3.: Khả năng thu thập dữ liệu của trang web

Một số trang web của các bộ/ngành thuộc Chính phủ có sự tương đồng lẫn nhau, đồng thời được bảo mật thông tin chặt chẽ do đó việc thu thập dữ liệu còn gặp nhiều hạn chế và khó khăn. Bên cạnh đó, việc thu thập dữ liệu không thành công còn vì nguyên nhân của lỗi kết nối mạng.

* 1. Phân tích CSDL

Tổng quan CSDL:



Hình 3.: Tổng quan CSDL

Bảng lưu thông tin của các bộ ngành:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | ministry\_id | int | Id của mỗi bộ |
|  | ministry\_name | nvarchar(100) | Tên của các bộ ngành |

Bảng 3.: Bảng ministry\_info

Bảng lưu thông tin các loại tin tức:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | article\_category\_type\_id | int | Id các loại tin tức |
|  | article\_category\_type\_name | nvarchar(100) | Tên của các mục tin tức được phân loại |

Bảng 3.: Bảng article\_category\_type

Bảng lưu thông tin các loại văn bản:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | legislation\_category\_type\_id | int | Id các loại văn bản |
|  | legislation\_category\_type\_name | nvarchar(100) | Tên các loại văn bản |

Bảng 3.: Bảng legislation\_category\_type

Bảng lưu thông tin chi tiết các loại tin tức:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | article\_category\_id | int | Id các loại tin tức của từng bộ - ngành |
| FK | ministry\_id | int | Id các bộ ngành |
|  | category\_name | Nvarchar(200) | Tên các loại tin tức của từng bộ - ngành |
|  | article\_category\_type\_id | Int | Id các loại tin tức |
|  | category\_link\_root | Varchar(1000) | Link truy cập đến danh sách hiển thị của từng loại tin tức |

Bảng 3.: Bảng article\_category\_info

Bảng lưu thông tin chi tiết các loại tin tức:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | legislation\_category\_id | int | Id các loại văn bản của từng bộ - ngành |
| FK | ministry\_id | int | Id các bộ ngành |
|  | legislation\_name | Nvarchar(200) | Tên các loại văn bản theo từng bộ - ngành |
|  | legislation\_category\_type\_id | Int | Id các loại văn bản |
|  | legislation\_link\_root | Varchar(1000) | Link truy cập đến danh sách hiển thị của từng loại vản bản |

Bảng 3.: Bảng legislation\_category\_info

Bảng lưu thông tin cấu hình các loại tin tức:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| FK | ministry\_id | int | Id các bộ ngành |
|  | article\_category\_type\_id | Int | Id các loại tin tức |
|  | article\_url\_xpath | Varchar(200) | Xpath để lấy đường link dẫn đến chi tiết bài báo |
|  | article\_thumbnail\_xpath | Varchar(200) | Xpath để lấy ảnh thumbnail |
|  | page\_rule | int | Quy tắc khoảng cách giữa các trang |
|  | article\_param\_xpath | Varchar(200) | Xpath để lấy đường link chứa param trong một phân loại bài báo |

Bảng 3.: Bảng ministry\_article\_category\_configuration

Bảng lưu thông tin cấu hình các chi tiết của bài báo:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| FK | ministry\_id | int | Id các bộ ngành |
|  | article\_tittle\_xpath | Varchar(200) | Xpath để lấy tiêu đề bài báo |
|  | article\_description\_xpath | Varchar(200) | Xpath để lấy mô tả bài báo |
|  | article\_time\_xpath | Varchar(200) | Xpath để lấy thời gian đăng tải bài báo |
|  | article\_author\_xpath | Varchar(200) | Xpath để lấy tên tác giả viết bài |
|  | article\_content\_xpath | Varchar(200) | Xpath để lấy nội dung của bài báo |

Bảng 3.: Bảng ministry\_article\_detail\_configuration

Bảng lưu thông tin cấu hình của văn bản:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| FK | ministry\_id | int | Id các bộ ngành |
|  | legislation\_category\_id | int | Id các loại văn bản của từng bộ - ngành |
|  | legislation\_url\_xpath | Varchar(200) | Xpath dẫn đến văn bản chi tiết |
|  | page\_rule | int | Quy tắc khoảng cách giữa các param |
|  | legislation\_param\_xpath | Varchar(200) | Xpath dẫn đến đường dẫn chứa param của trang web |

Bảng 3.: Bảng ministry\_legislation\_category\_configuration

Bảng lưu thông tin cấu hình các chi tiết của một văn bản:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| FK | ministry\_id | int | Id các bộ ngành |
|  | legislation\_name\_xpath | Varchar(200) | Xpath để lấy tên văn bản |
|  | legislation\_so\_hieu\_van\_ban\_xpath | Varchar(200) | Xpath để lấy số hiệu văn bản |
|  | legislation\_ngay\_hieu\_luc\_xpath | Varchar(200) | Xpath để lấy ngày hiệu lực |
|  | legislation\_ngay\_ky\_xpath | Varchar(200) | Xpath để lấy ngày ký |
|  | legislation\_trich\_yeu\_xpath | Varchar(200) | Xpath để lấy trích yếu văn bản |
|  | legislation\_co\_quan\_ban\_hanh\_xpath | Varchar(200) | Xpath để lấy cơ quan ban hành văn bản |
|  | legislation\_nguoi\_ky\_xpath | Varchar(200) | Xpath để lấy người ký văn bản |
|  | legislation\_loai\_van\_ban\_xpath | Varchar(200) | Xpath để lấy tên loại văn bản |
|  | legislation\_tinh\_trang\_xpath | Varchar(200) | Xpath để lấy tình trạng hiệu lực văn bản |
|  | legislation\_link\_download\_xpath | Varchar(200) | Xpath để lấy link download văn bản |

Bảng 3.: Bảng ministry\_legislation\_detail\_configuration

Bảng để lưu trữ thông tin tin tức sau khi crawl:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | article\_id | int | Id của bài báo |
| FK | ministry\_id | int | Id của bộ - ngành |
|  | article\_url | Varchar(1000) | Url của bài báo |
|  | article\_title | Nvarchar(200) | Tiêu đề bài báo |
|  | article\_description | Nvarchar(500) | Mô tải bài báo |
|  | article\_time | Varchar(10) | Thời gian đăng bài báo |
|  | article\_author | Nvarchar(100) | Tác giả bài báo |
|  | article\_content | Nvarchar(max) | Nội dung bài báo |
|  | article\_thumbnail | Varchar(100) | Ảnh thumbnail bài báo |

Bảng 3.: Bảng ministry\_legislation\_detail\_configuration

Bảng để lưu trữ thông tin văn bản sau khi crawl:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| PK | legislation\_id | int | Id của văn bản |
| FK | ministry\_id | int | Id của bộ - ngành |
|  | legislation\_name | Nvarchar(500) | Tên văn bản |
|  | legislation\_url | Varchar(500) | Link dẫn đến chi tiết văn bản |
|  | so\_hieu\_van\_ban | Varchar(20) | Số hiệu văn bản |
|  | ngay\_ban\_hanh | Varchar(100) | Ngày ban hành văn bản |
|  | ngay\_hieu\_luc | Nvarchar(100) | Ngày văn bản có hiệu lực |
|  | trich\_yeu | Nvarchar(500) | Trích yếu của văn bản |
|  | co\_quan\_ban\_hanh | Nvarchar(100) | Cơ quan ban hành văn bản |
|  | nguoi\_ky | Nvarchar(50) | Người ký văn bản |
|  | loai\_van\_ban | Nvarchar(50) | Thể loại văn bản |
|  | tinh\_trang | Nvarchar(50) | Tình trạng hiệu lực của văn bản |
|  | link\_download | Nvarchar(300) | Link tải văn bản |

Bảng 3.: Bảng legislation\_info

1. Ứng dụng thực tiễn

Hệ thống thu thập thông tin từ các trang bài viết của các bộ/ngành Việt Name bằng công cụ Scrapy và lưu trữ dưới dạng các bảng dữ liệu SQL theo các trình tự:

Bước 1: Tạo và thêm thông tin cho các bảng dữ liệu cần thiết phục vụ cho quá trình thu thập như: đường dẫn gốc của bài báo được phân theo trang, xpath của đường dẫn đến bài viết chi tiết, param của trang web đó.

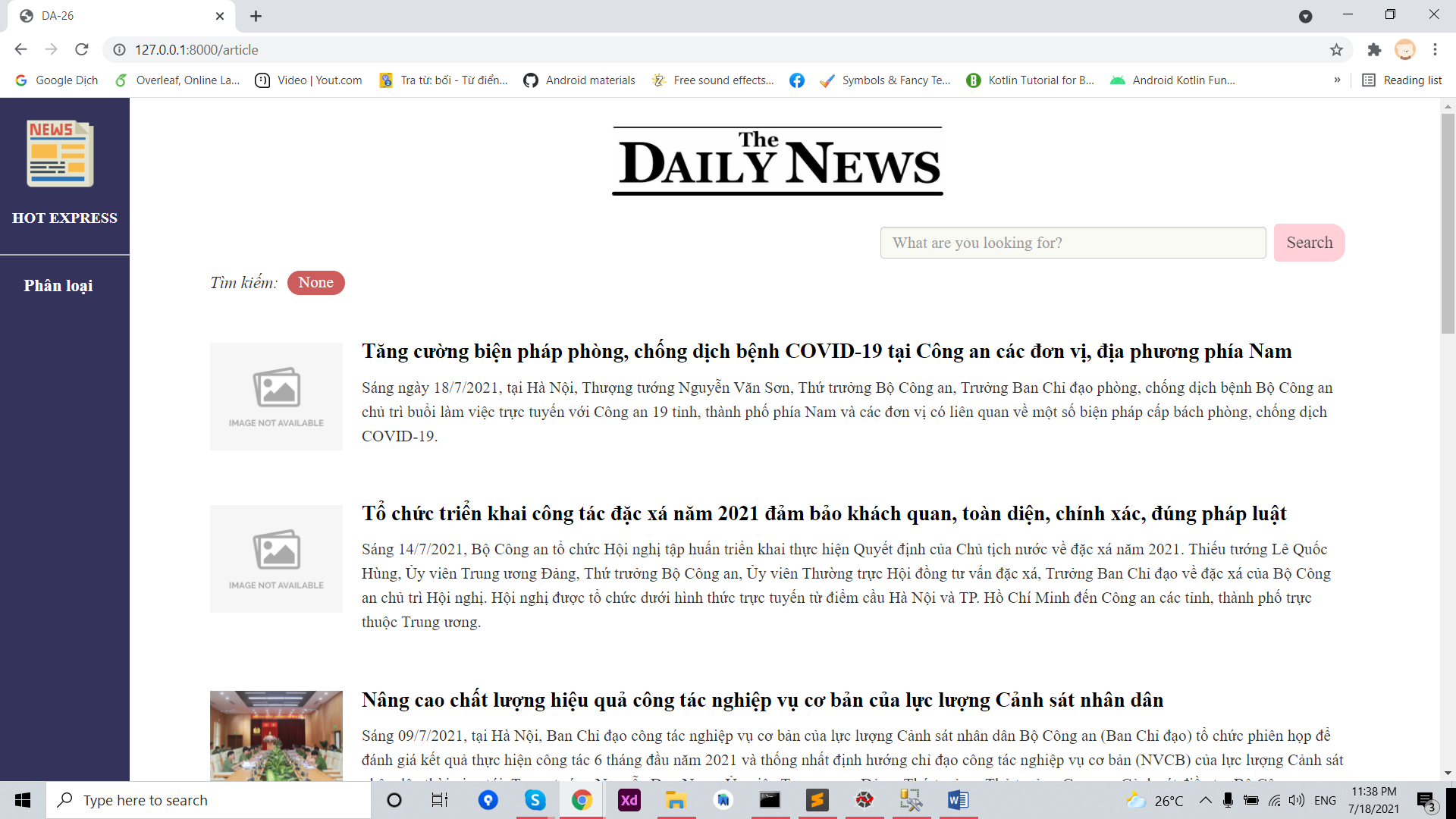
Bước 2: Tạo dự án Django mới, thêm phần giao diện được viết bằng ngôn ngữ đánh dấu siêu văn bản - Hypertext Markup Language (HTML) cho web và file python cần thiết để xử lý phía backend.

Bước 3: Trong file python đã tạo, từ các thông tin có được ở bước 1, ta dùng Scrapy để khởi tạo lớp thực hiện công việc crawl dữ liệu. Bên trong lớp đó ta tạo ra các hàm kết nối với SQL để xử lý dữ liệu được thu thập thông qua các xpath, sau khi hoàn tất quá trình crawl, ta tiến hành lưu trữ các thông tin đã lấy được vào CSDL SQL và dùng nó để tiếp tục các bước hiển thị dữ liệu lên giao diện. Lúc này, Django có nhiệm vụ thực hiện các kết nối giữa backend và giao diện dựa trên mô hình MTV như đã nói ở trên. Ngoài ra, hệ thống còn áp dụng công cụ lập lịch tự động với mỗi 6 tiếng sẽ crawl về các thông tin mới nhất từ trang nguồn.

Bước 4: Để triển khai trang web của dự án Django, ta vào thư mục chứa file manage.py và chạy câu lệnh “'python manage.py runserver” trên cửa sổ Command Prompt.

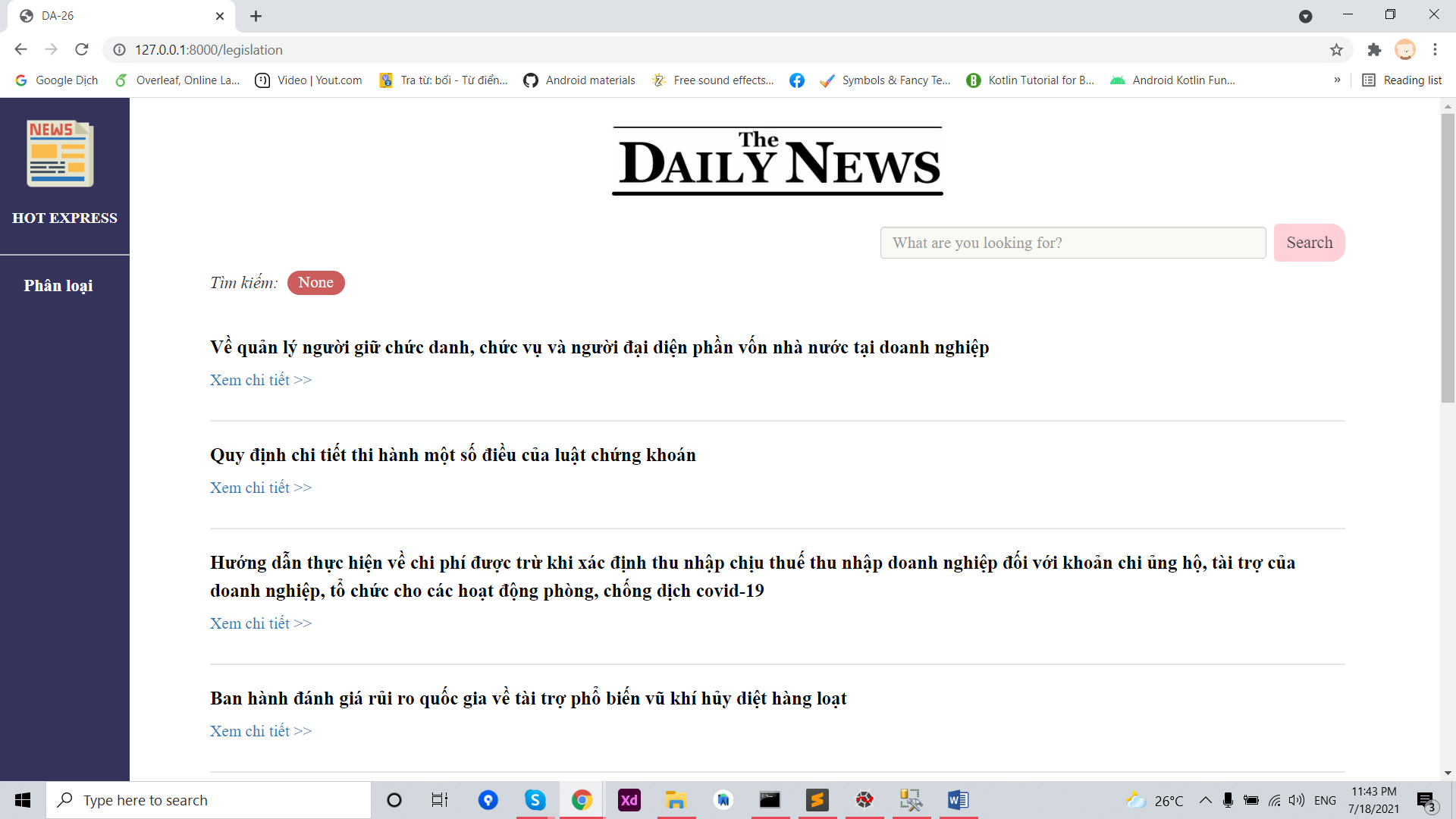
Trang web Hot Express được vận hành với những chức năng cơ bản như tìm kiếm theo từ khóa được nhập, tìm kiếm phân loại theo các bộ/ngành nhằm đem đến cho độc giả những tin tức mới một cách nhanh chóng và chuẩn xác nhất.

Ở trang hiển thị các bài báo, các thông tin mới nhất sẽ được cập nhật lên đầu trang sau mỗi 6 tiếng.



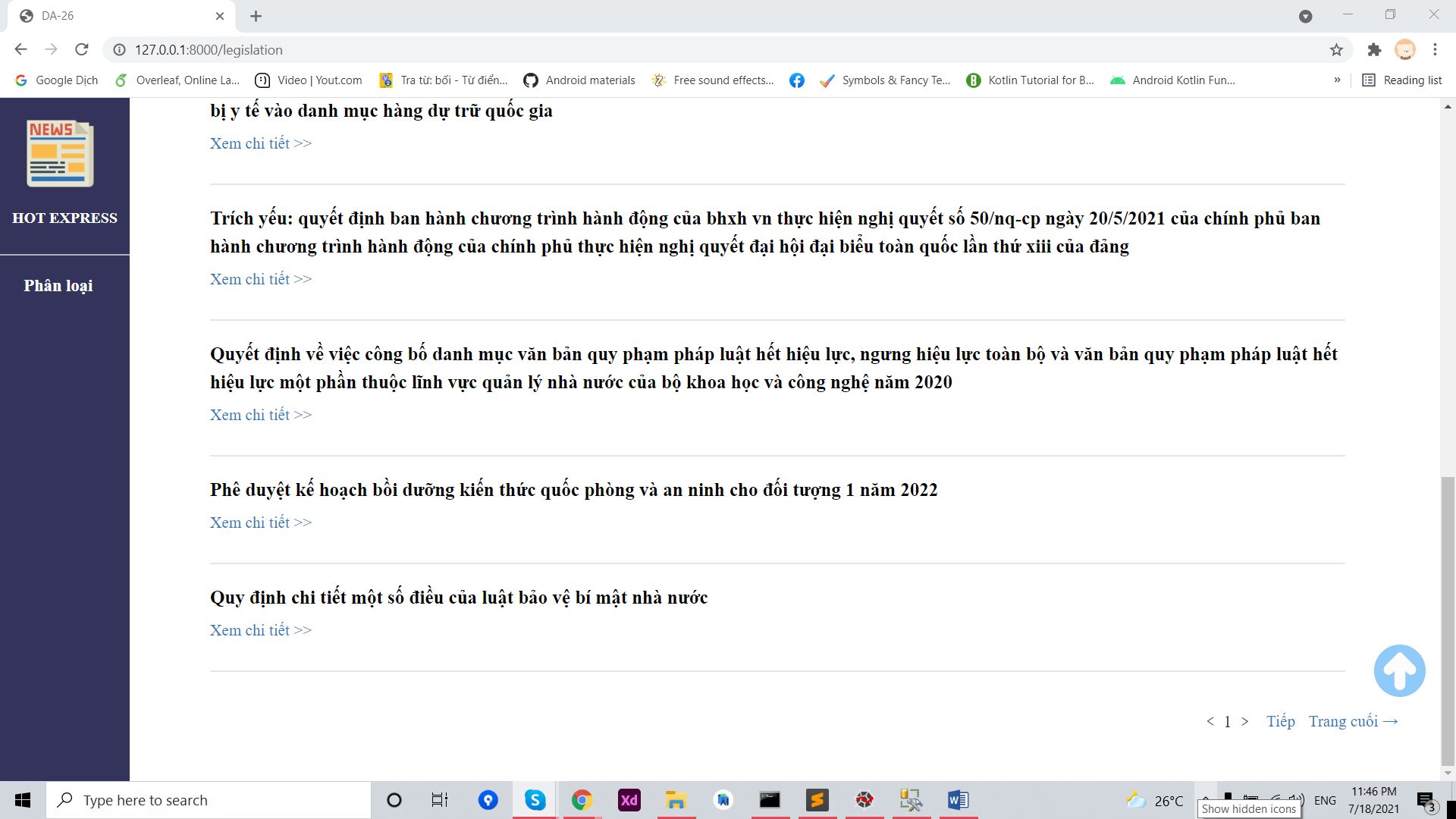
Hình 3.: Trang hiển thị tin tức trên hệ thống

Tương tự, ta có trang văn bản hiển thị các văn bản theo thứ tự được cập nhật gần nhất.



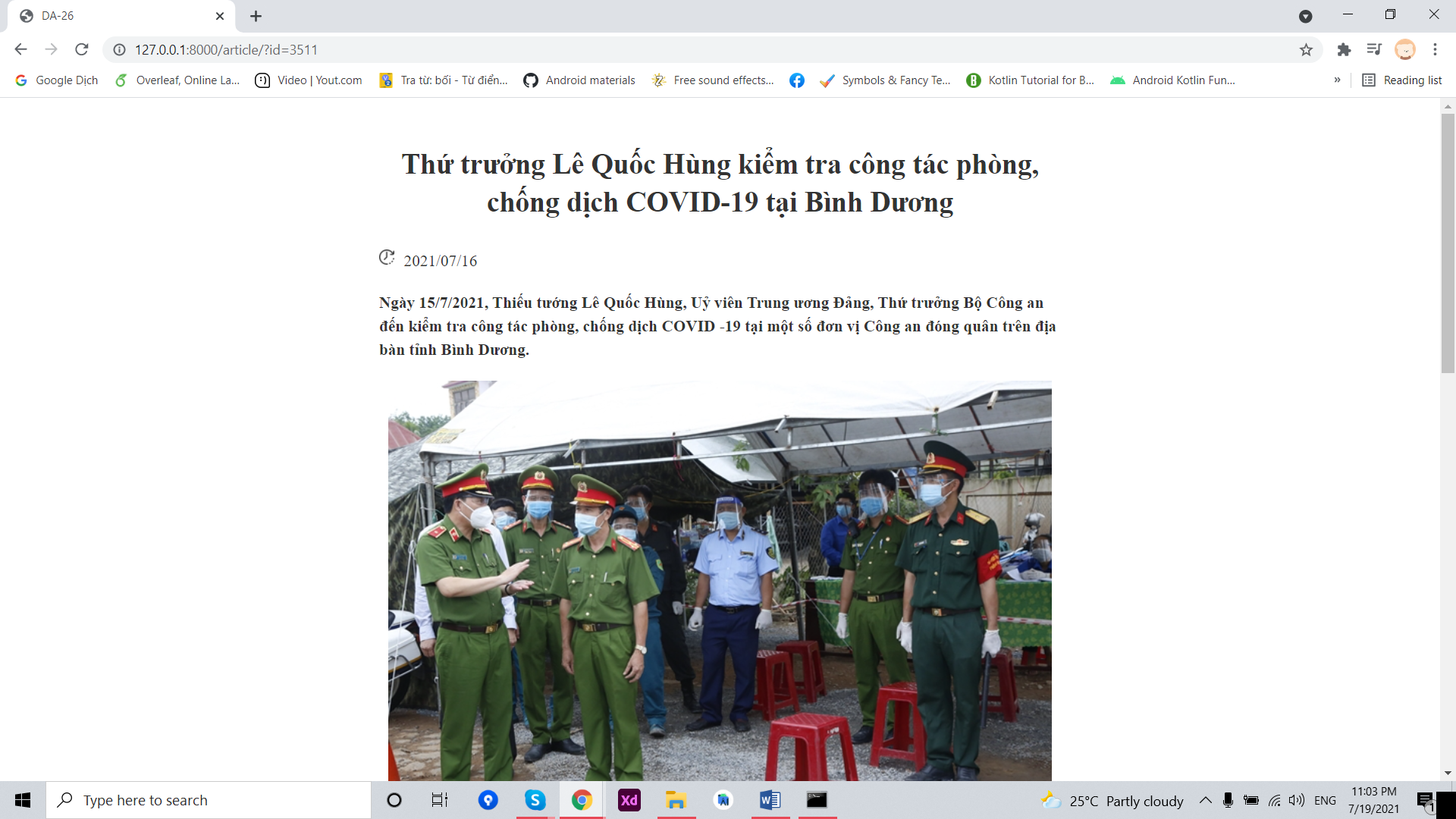
Hình 3.: Trang hiển thị văn bản trên hệ thống

Mỗi trang web hiển thị tối đa 10 bài báo. Trong trường hợp số lượng bài báo nhiều hơn sẽ được phân thành các trang tiếp theo, do đó, phía cuối các bài viết sẽ có các dấu điều hướng trang để chỉ định người dùng xem các thông tin tiếp theo, trở về trang báo trước đó, trở về trang đầu tiên hay hướng đến trang cuối cùng. Ngoài ra, nhằm tạo nên sự tiện lợi khi trải nghiệm đọc tin tức trên hệ thống, người dùng còn có thể dễ dàng trở về đầu trang bằng cách cuộn lên khi bấm vào nút mũi tên ở cuối trang tin tức đó.



Hình 3.: Phần cuối trang của hệ thống

Khi thực hiện xem một bài viết, hệ thống sẽ điều hướng đến trang web chứa thông tin chi tiết của bài viết đó qua id tương ứng. Một bài báo sẽ hiển thị tiêu đề, thời gian đăng tải, hình ảnh minh họa (nếu có), nội dung, tên tác giả và trang web nguồn.



Hình 3.: Phần đầu trang bài viết chi tiết



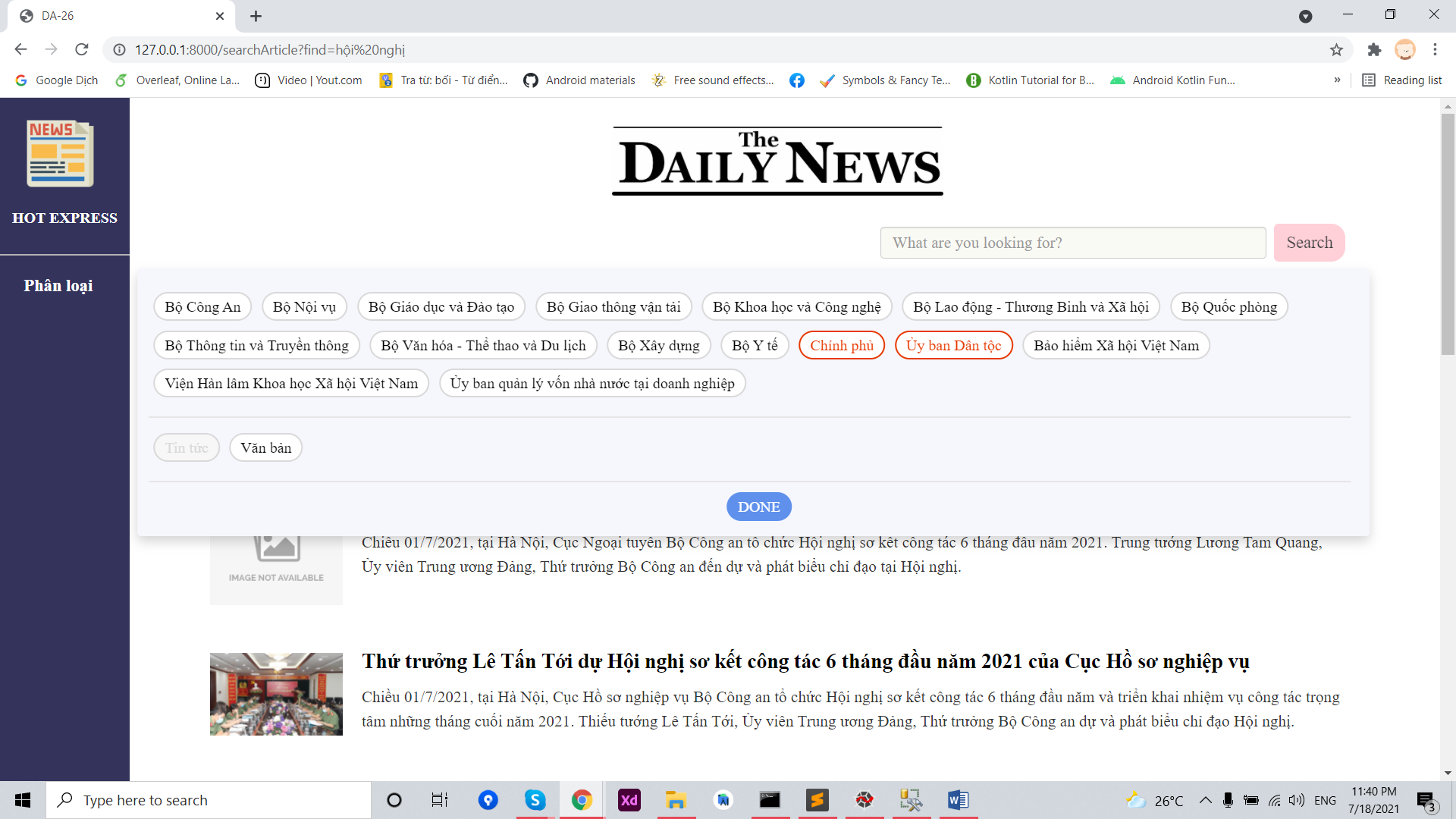
Hình 3.: Phần cuối trang bài viết chi tiết

Ở phần chi tiết các văn bản pháp luật (nghị định, nghị quyết, quyết định,..) sẽ bao gồm số hiệu văn bản, ngày ban hành, ngày hiệu lực, trích yếu, cơ quan ban hành, người ký, loại văn bản, tình trạng và link download bản gốc của văn bản đó.



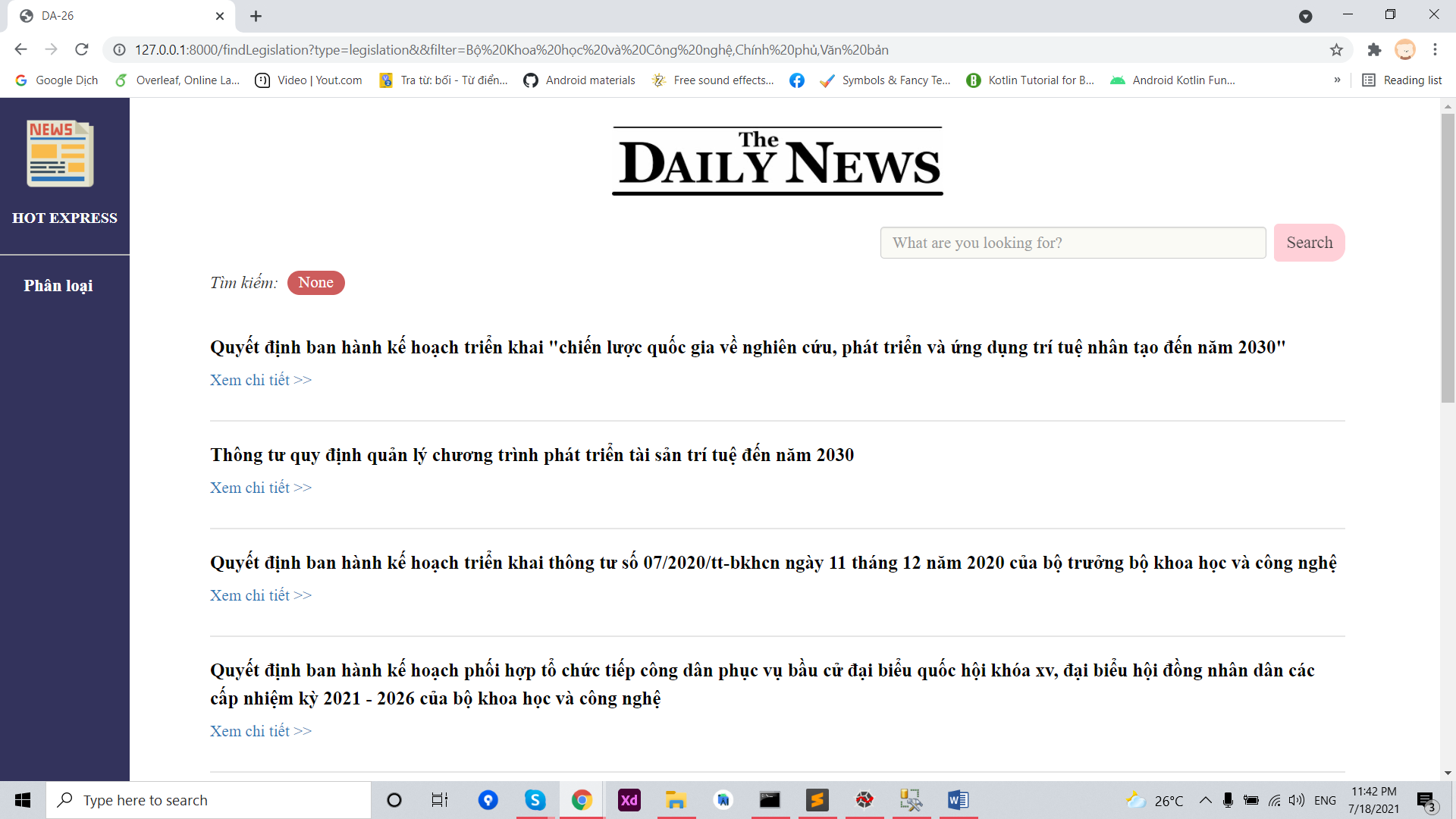
Hình 3.: Trang văn bản chi tiết

Bên trái trang web cho thấy thanh menu dọc chứa chức năng phân loại. Tại đây, các đọc giả được đơn giản hóa công việc tìm kiếm theo các bộ ngành đã được gợi ý. Nếu muốn tìm kiếm phân loại theo dạng bài viết hiện tại, người dùng chỉ cần chọn các đề mục mong muốn và bấm nút Done. Mặt khác để chọn phân loại cho dạng bài viết khác, như ở hình bên dưới là văn bản, người dùng cần chọn thêm phân loại Văn bản và bấm Done.



Hình 3.: Mục phân loại của hệ thống

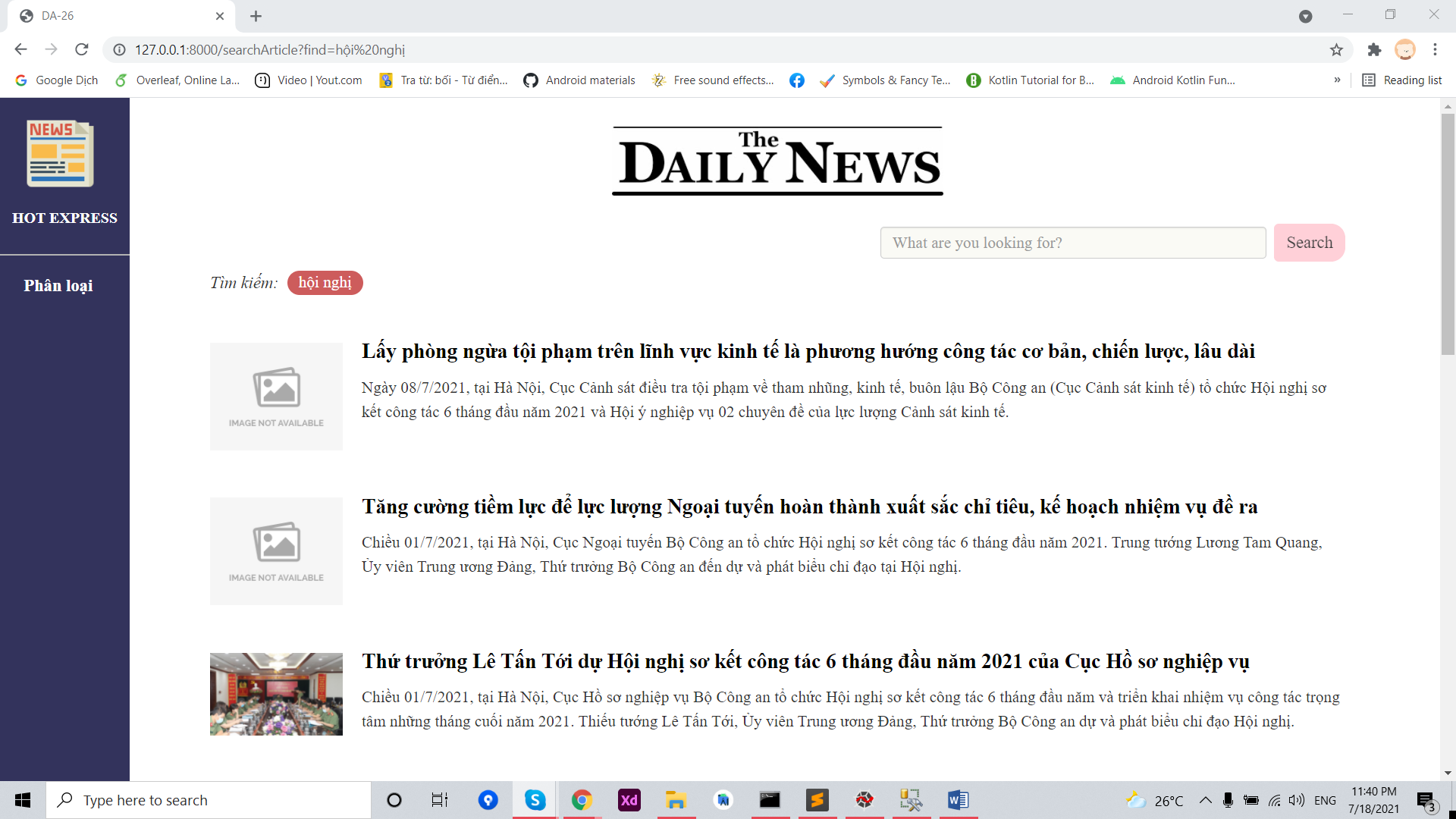
Kết quả trên thanh URL hiển thị tương ứng các chức năng và phân loại người dùng đã lựa chọn. Hình bên dưới (hình 3.13) đang hiển thị kết quả tìm kiếm cho phân loại Bộ Khoa học và Công nghệ, Chính phủ cho phần tin văn bản.



Hình 3.: Trang hiển thị kết quả tìm kiếm theo phân loại

Đối với chức năng tìm kiếm, người dùng sẽ được thuận tiện hơn trong việc ghi nhớ từ khóa đã yêu cầu nhờ vào việc tự động cập nhật từ khóa đó ở phần lưu trữ tìm kiếm phía trên các bài báo, dưới khung Search.

Tìm kiếm theo từ khóa ở phân loại tin tức sẽ trả về các tin tức có nội dung chứa từ khóa đó, còn với văn bản thì sẽ tìm kiếm từ khóa theo tên văn bản.



Hình 3.: Trang hiển thị kết quả tìm kiếm theo từ khóa

Hệ thống lưu trữ và tìm kiếm đóng vai trò như một cỗ máy với trí tuệ nhân tạo thay thể con người thu thập và quản lý lượng thông tin khổng lồ mà bộ não con người không thể kiểm soát hoàn toàn. Phần tìm kiếm ở đây ứng dụng bởi kỹ thuật xử lý song song trên nền tảng Pyspark. Như đã nói, Pyspark là một công cụ hỗ trợ tính toán nhanh, đặc biệt đối với các khung dữ liệu lớn.

Cùng với những lợi ích đó, Spark là sự lựa chọn tuyệt vời cho các mục đích tìm kiếm dữ liệu lớn, là chìa khóa cho các câu hỏi về việc xử lý khối các công việc mà con người không thể làm được.

Trong cơ chế hoạt động của Spark bao gồm một thành phần gọi là RDD. RDD (Resilient Distributed Datasets - Bộ dữ liệu phân phối khả năng phục hồi) là một cấu trúc dữ liệu cơ bản của Spark. Nó là một tập các đối tượng phân tán không thay đổi, chịu được lỗi, có khả năng hoạt động song song dùng để hỗ trợ tính toán xử lý trong bộ nhớ. Mỗi bản ghi trong RDD được chia thành các phân vùng logic, có thể được tính toán trên các node khác nhau của cụm máy chủ.

Nói cách khác, RDD là một tập hợp các đối tượng tương tự như danh sách trong Python, nhưng RDD được tính toán trên một số quy trình nằm rải rác trên nhiều phân vùng logic và xử lý chỉ trong một quy trình. Việc chia nhỏ các công việc trên các luồng riêng biệt giúp đem lại hiệu năng cao hơn, giảm thiểu các chi phí tổn thất không đáng có.

Do đó, so sánh với cách làm thông thường, khi áp dụng Pyspark vào bước xử lý, ta nhận thấy dùng Pyspark sẽ giúp xử lý dữ liệu nhanh hơn bởi vì tính năng của Pyspark, cũng như Spark, được phát triển nhằm hoạt động nhanh hơn 100 lần so với các khung xử lý dữ liệu quy mô lớn truyền thống cũng như tăng tính trải nghiệm đối với người dùng.

Khi thực hiện chức năng tìm kiếm theo từ khóa, dữ liệu thu thập được truy vấn qua câu lệnh SQL và trả về dưới dạng mảng. Để xử lý các mảng dữ liệu này với Pyspark, ta cần chuyển đổi chúng thành dạng Dataframe thông qua thư viện hỗ trợ Pandas. Mặc dù Pyspark cũng sử dụng cấu trúc Dataframe, tuy nhiên, thư viện Pandas đem lại hiệu quả trong các thao tác dữ liệu và nhiều chức năng phong phú hơn cả. Thực tế, cả Pandas và Spark đều có ưu điểm, nhược điểm riêng và có thể bổ sung cho nhau nhưng không hoàn toàn thay thế đối phương. Trong khi Pandas xử lý đơn giản, linh hoạt các dữ liệu, thích hợp với những thử nghiệm, khám phá với dữ liệu nhỏ thì Spark ổn định và mạnh mẽ hơn với lượng dữ liệu lớn đa dạng nguồn. Vì lẽ đó, các nhà phát triển cho rằng hai công cụ này có thể kết hợp cùng nhau để tạo nên hiệu suất tốt nhất cho quá trình xử lý dữ liệu.



Hình 3.: Đoạn code áp dụng Pyspark để tìm kiếm dữ liệu

Sau khi hoàn thành công việc chuyển đổi dữ liệu sang dạng dữ liệu mà Spark có thể quản lý, ta tiến hành thực hiện tìm kiếm thông tin theo yêu cầu. Cũng giống như SQL, Dataframe có cho mình chức năng tìm kiếm theo điều kiện với cú pháp chẳng khác là mấy so với hàm Where trong SQL, chính là Filter, điều này giúp lập trình viên giảm bớt được nhiều khó khăn khi làm quen giữa hai CSDL. Kết quả trả về thông qua hàm Collect là tập dữ liệu mới được hình thành sau tìm kiếm, phần tiếp theo là các bước chuyển đổi thích hợp để truyền tải được thông tin cần thiết đến người dùng.

Như vậy, hệ thống sử dụng kỹ thuật song song trên nền tảng Pyspark trong công cuộc quản lý, truy xuất dữ liệu trở nên thuận tiện và nhanh chóng hơn rất nhiều, tối ưu hóa năng suất thực thi nhiều dòng lệnh cũng như khả năng thực hiện liên tiếp một hàng đợi những công việc gây ra những hạn chế về tính toán và bộ nhớ của chương trình.

CHƯƠNG 4 – TỔNG KẾT

* 1. Kết quả thu được

Ứng dụng đã đáp ứng được những yêu cầu cơ bản về thu thập và tổng hợp thông tin thành một trang tin tức hoàn chỉnh.

Các thông tin tin tức và văn bản từ các website chính thống của các bộ ngành được thu thập thông qua bộ crawl thông tin sử dụng thư viện Scrapy. Thông tin đã lấy được lưu vào CSDL trên SQL Server. Sau mỗi 6 tiếng, hệ thống sẽ tự động crawl lại thông tin tính từ thời điểm thu thập thông tin gần nhất.

Dữ liệu sau khi thu thập được đăng tải lên trang web và chú thích nguồn của bài viết. Trang chủ hiển thị các thông tin mới được thu thập. Ngoài ra nếu cần tìm kiếm hoặc xem thông tin, người dùng có thể nhập từ khóa vào ô tìm kiếm hoặc dùng chức năng phân loại để xem thông tin của các bộ/ngành cụ thể một cách nhanh chóng và chuẩn xác nhất.

Trang chi tiết tin tức và văn bản ở trang web thông tin tổng hợp hiển thị đúng thông tin lấy được từ trang chủ các trang web chính thống của các bộ/ngành, bao gồm: hình ảnh, nội dung, tiêu đề, tác giả, ngày đăng,…

* 1. Hướng phát triển

Hệ thống có thể được phát triển thành ứng dụng điện thoại. Các ứng dụng trên điện thoại di động ngày càng phổ biến cùng với sự phổ biến của điện thoại thông minh. Để hệ thống tiếp cận đến nhiều người dùng hơn, cung cấp các thông tin chính xác và kịp thời hơn, phát triển ứng dụng di động là một hướng đi rất tiềm năng.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Aptech Saigon, *Ứng dụng của Python phổ biến nhất trong thực tế* <https://aptechsaigon.edu.vn/lap-trinh-python-loi-ich-va-ung-dung-thuc-te>
2. Bachkhoa-Aptech (2018), *Khám phá ngay 3 ứng dụng nổi bật của Python* <https://bachkhoa-aptech.edu.vn/kham-pha-ngay-3-ung-dung-noi-bat-cua-python/549.html>
3. Dương Nguyễn (2020), *Python là gì? Tại sao nên chọn Python?* <https://quantrimang.com/python-la-gi-tai-sao-nen-chon-python-140518>
4. Ichi Pro, *Ví dụ về việc sử dụng Apache Spark với PySpark bằng Python*

https://ichi.pro/vi/vi-du-ve-viec-su-dung-apache-spark-voi-pyspark-bang-python-267611095265298

1. Lê Quý Quyết (2017)*, Giới thiệu/hướng dẫn về Crawler với Scrapy Framework*

[https://viblo.asia/p/gioi-thieuhuong-dan-ve-crawler-voi-scrapy-framework- ByEZkWoEZQ0](https://viblo.asia/p/gioi-thieuhuong-dan-ve-crawler-voi-scrapy-framework-ByEZkWoEZQ0)

1. Nguyễn Khánh (2020), *Tìm hiểu về SQL*

<https://viblo.asia/p/tim-hieu-ve-sql-YWOZrapEKQ0>

1. Nguyễn Trang (2017), *Tổng quan về SQL*

<https://quantrimang.com/tong-quan-ve-sql-142305>

1. PACISOFT Help Master (2019), *Microsoft SQL Server là gì? SQL dùng làm gì? Có mấy phiên bản?*

<https://help.pacisoft.com/knowledgebase/sql-server-la-gi-sql-server-dung-lam-gi-co-may-phien-ban/>

1. Semtek (2019), *Tất tần tật các kiến thức về SQL server cho người mới tìm hiểu*

<https://www.semtek.com.vn/sql-server-2/>

1. SQL Advice Community (2021), *SQL là gì? Tổng quan về SQL*

<http://sqladvice.com/tong-quan-ve-sql/#Dac_diem_va_doi_tuong_cua_SQL>

1. Stanford, *Python cho lập trình web, lý do nên dùng Django* https://stanford.com.vn/kien-thuc-lap-trinh/tin-chi-tiet/cagId/27/id/17370/python-cho-lap-trinh-web-ly-do-nen-dung-django
2. Thủy Nguyễn (2021), *Django là gì và lý do vì sao nên sử dụng trong thiết kế web*

<https://bizfly.vn/techblog/django-la-gi.html>

1. Vietnam Lab Center Co., *Ltd, Xử lý dữ liệu với Spark Dataframe*

<https://blog.vietnamlab.vn/xu-ly-du-lieu-voi-spark-dataframe/>

1. VinaSupport (2020), *Django là gì? Tổng quan về Django Framework* <https://vinasupport.com/django-la-gi-tong-quan-ve-django-framework/>
2. Wikipedia (2021), *Tính toán song song*

https://vi.wikipedia.org/wiki/T%C3%ADnh\_to%C3%A1n\_song\_song

**Tiếng Anh**

1. Insophia (2015)*, Architecture overview* <https://doc.scrapy.org/en/0.10.3/topics/architecture.html>
2. Jakub Protasiewicz (2018), *Flask vs. Django - Which Python Framework you should choose?*

<https://www.netguru.com/blog/flask-vs-django>

1. JavaTpoint, *Django MTV*

<https://www.javatpoint.com/django-mvt>

1. Julie Korsun (2021), *Why do we use Django framework and what is Django used for*

<https://djangostars.com/blog/why-we-use-django-framework/>

1. Microsoft Docs (2021), *Editions and supported features of SQL Server 2019 (5.x)*

<https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-version-15?view=sql-server-ver15>

1. Pulkit Sharma, *Model View Controller (MVC) and link with Django (MTV)* https://iq.opengenus.org/model-view-controller-django/
2. Rackspace, *Django*

<https://www.djangoproject.com/start/overview/>

1. Shardul Bhatt (2020), *Pros and Cons of Django framework for App Development*

<https://dzone.com/articles/pros-and-cons-of-django-framework-for-app-developm>

1. Wikipedia (2021), *Django (web framework)* <https://en.wikipedia.org/wiki/Django_(web_framework)>
2. Wikipedia (2021), *Microsoft SQL Server*

https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server

1. Wikipedia (2021), *Scrapy*

<https://en.wikipedia.org/wiki/Scrapy>

1. Wikipedia (2021), *Python (programming language)* <https://en.wikipedia.org/wiki/Python_(programming_language)#Design_philosophy_and_features>