

ĐẠI HỌC BÁCH KHOA HÀ NỘI

KHOA TOÁN - TIN

----- O  O -----



THU THẬP DỮ LIỆU TUYỂN DỤNG

ĐỒ ÁN I

Chuyên ngành: TOÁN TIN

Giảng viên hướng dẫn :

ThS. Nguyễn Danh Tú

Sinh viên thực hiện :

Đương Công Thái

Mã số sinh viên :

20216883

HÀ NỘI - 2024

MỤC LỤC

Lời nói đầu	4
I. Khảo sát	5
1. Nhu cầu phân tích.....	5
2. Các website thu thập thông tin.....	6
3. Công cụ thu thập dữ liệu	45
II. Phân tích và thiết kế hệ thống.....	46
1. Kiến trúc hệ thống phân tích dữ liệu.....	46
2. Cơ sở dữ liệu	47
3. Phân tích tin tuyển dụng các nguồn	48
4. Data Pipeline	48
III. Xây dựng chương trình	49
1. Cấu hình	49
2. Xây dựng các phân hệ thu thập dữ liệu	53
3. Cách chạy chương trình đóng gói.....	84
4. Một số hình ảnh truy vấn từ cơ sở dữ liệu sau thu thập	86
IV. Kết luận	90
V. Tài liệu tham khảo	91

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục đích và nội dung đồ án:.....

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

2. Kết quả đạt được:

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

3. Ý thức làm việc của sinh viên:

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

Hà Nội, ngày tháng năm

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

Lời nói đầu

Dữ liệu và phân tích dữ liệu đóng vai trò quan trọng trong nhiều lĩnh vực và ngành công nghiệp, mang lại nhiều lợi ích quan trọng.

Dữ liệu cung cấp cơ sở chắc chắn để ra quyết định thông minh. Doanh nghiệp và tổ chức có thể dựa vào dữ liệu để hiểu rõ hơn về môi trường kinh doanh, đối thủ cạnh tranh, và đáp ứng nhanh chóng với biến động thị trường.

Phân tích dữ liệu là công cụ hỗ trợ quyết định quan trọng. Những thông tin chi tiết từ phân tích giúp người quyết định có cái nhìn tổng thể và chính xác hơn về tình hình.

Phân tích dữ liệu có thể sử dụng để xây dựng mô hình dự đoán, từ việc dự đoán doanh số bán hàng đến phân loại khách hàng. Điều này mang lại lợi ích lớn trong việc kế hoạch và triển khai chiến lược.

Bằng cách phân tích dữ liệu thị trường và phản hồi từ khách hàng, doanh nghiệp có thể đáp ứng nhanh chóng với nhu cầu thị trường.

Nguồn dữ liệu chính là nền tảng quan trọng nhất cho bất kỳ hoạt động phân tích dữ liệu nào. Để cập nhật thông tin về thị trường nhanh chóng, đòi hỏi nguồn dữ liệu phải là nguồn dữ liệu được cập nhật liên tục.

Chính vì thế, trong đồ án này em xin trình bày về một số phương pháp thu thập dữ liệu trên các website nhằm phục vụ cho nhu cầu dữ liệu. Đồ án sẽ tập trung vào việc thu thập các tin tuyển dụng ở các website và có thiên hướng về ngành nghề công nghệ thông tin, ngôn ngữ lập trình được dụng chủ yếu trong đồ án là Python. Do thời gian tìm hiểu còn hạn chế nên sẽ chỉ trình bày về một số phương pháp chủ yếu, ngoài ra còn rất nhiều phương pháp khác. Vì vậy, em rất mong được thày và bạn đọc góp ý để có thể hoàn thiện hơn.

Em xin chân thành cảm ơn!

I. Khảo sát

1. Nhu cầu phân tích

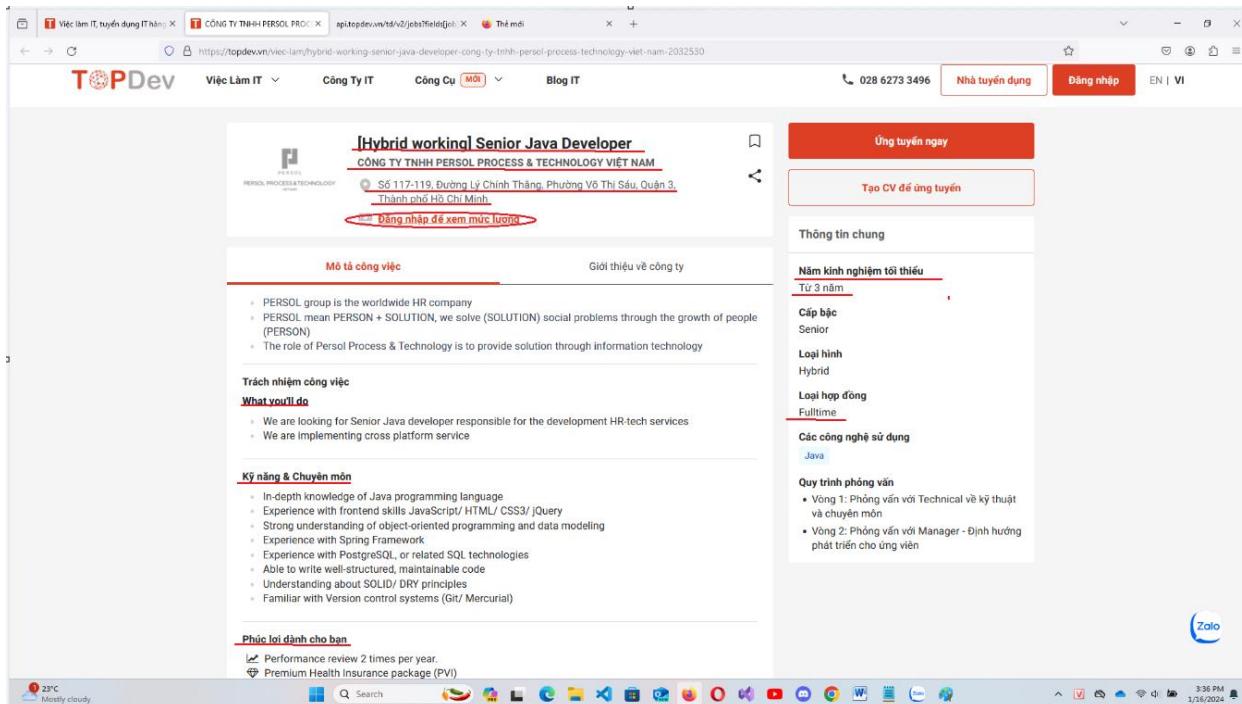
Phân tích dữ liệu thông tin tuyển dụng mang lại nhiều lợi ích cho sinh viên, giúp sinh viên hiểu rõ hơn về thị trường lao động, cũng như nâng cao khả năng thành công trong quá trình tìm kiếm việc làm. Dữ liệu tuyển dụng cung cấp thông tin về các ngành nghề hot, và các kỹ năng được đánh giá cao, từ đó sinh viên có thể tự định hình hướng nghiệp của mình dựa trên thông tin này. Hiểu rõ về yêu cầu của các công ty giúp sinh viên chuẩn bị một cách tốt hơn cho quá trình tìm kiếm việc làm, biết được các kỹ năng cứng, kỹ năng mềm nào được các nhà tuyển dụng ưa chuộng, giúp sinh viên xác định được bản thân phù hợp với ngành nghề nào, ngành nghề nào có tiềm năng phát triển.

Đối với sinh viên khoa Toán – Tin của Đại học Bách khoa Hà Nội thì việc có được cái nhìn tổng quan về thị trường tuyển dụng ngay trong những năm đầu là một bước khởi đầu to lớn để định hình được đường đi trong tương lai. Chính vì thế, dự án phân tích dữ liệu tuyển dụng này sẽ giúp mọi người có được cái nhìn rõ ràng hơn về thị trường việc làm các ngành nghề, đặc biệt là ngành nghề công nghệ thông tin.

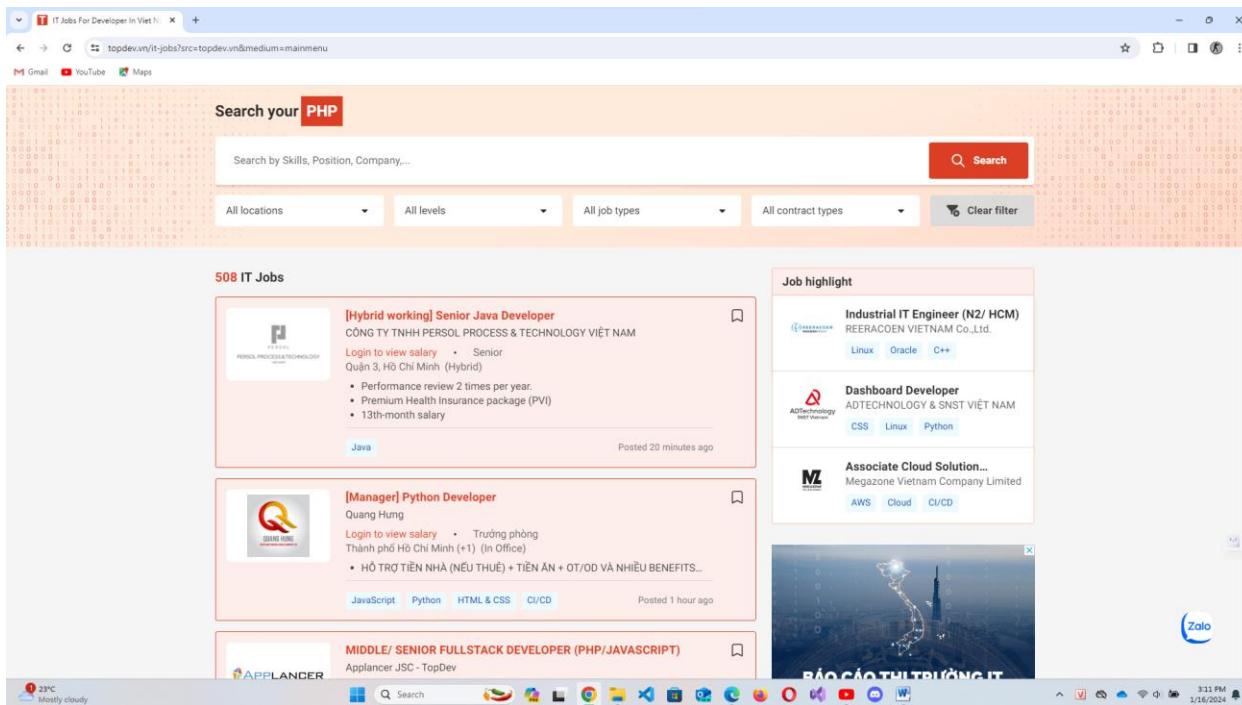
Nhu cầu về phân tích : Mức lương, kinh nghiệm yêu cầu, các kỹ năng yêu cầu, bằng cấp yêu cầu, số lượng tuyển dụng, khu vực tuyển dụng, mô tả công việc thực tế, các quyền lợi được hưởng khi tham gia vào công việc đó...

2. Các website thu thập thông tin

a) TopDev



Hình 1. Khảo sát cấu trúc tin TopDev

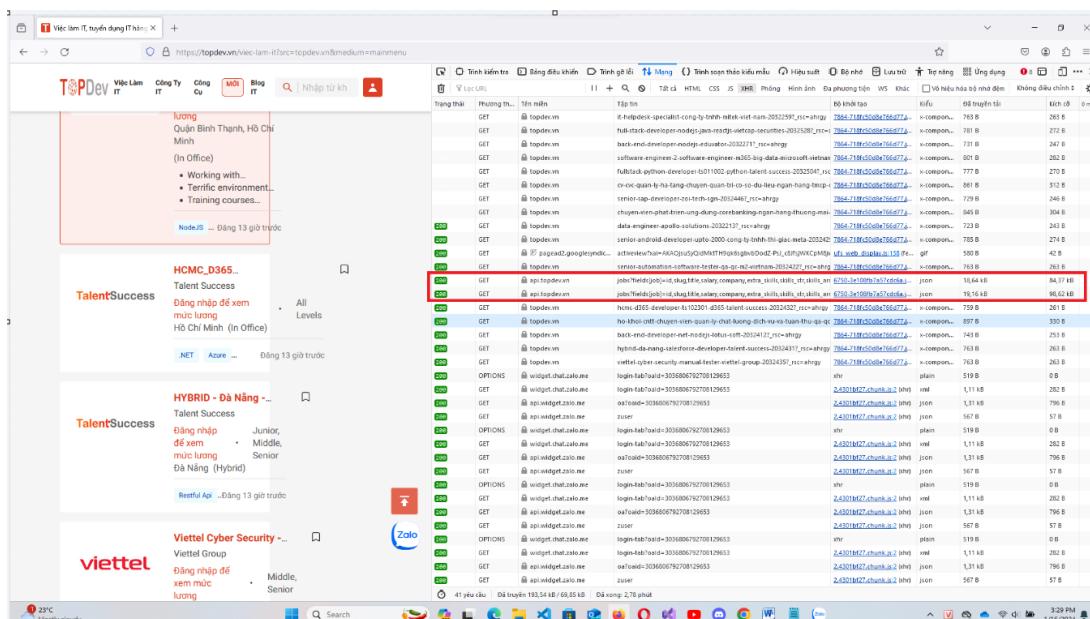


Hình 2. Khảo sát danh sách tin TopDev

Trang web sử dụng một tính năng được gọi là infinite scroll, tức là cuộn vô hạn, là một phương pháp thiết kế giao diện người dùng trong đó nội dung trang web được tải động khi người dùng cuộn xuống trang. Thay vì hiển thị toàn bộ nội dung trang từ đầu, chỉ một lượng nhỏ nội dung ban đầu được tải và hiển thị. Khi người dùng cuộn xuống cuối trang, thêm nội dung sẽ được tải và thêm vào trang hiện tại mà không cần tải lại toàn bộ trang từ đầu.

Trong mô hình infinite scroll, thông thường có thể có việc gửi các yêu cầu (requests) đến máy chủ để lấy thêm dữ liệu khi cần. Điều này thường được thực hiện thông qua API, nơi máy chủ trả về dữ liệu mới trong định dạng JSON hoặc XML. Điều này giúp giảm thời gian tải trang ban đầu và tạo trải nghiệm người dùng liền mạch hơn khi cuộn qua nhiều nội dung.

Vì vậy, ta sẽ tìm kiếm trong các API mà server trả về cho trang web, và kết quả được như sau:



Hình 3. Các API trả về từ Server(TopDev)

```

{
  "id": "1234567890",
  "title": "User Interface / user experience designer",
  "content": "We are looking for a UI/UX designer to design web interfaces as well as mobile interfaces required turning our software into easy-to-use products for our clients. Arbaa is a fast growing trading platform, Customer Portals and Back Office Systems. Arbaa's mission is to make the world a better place by creating better user experiences. Arbaa's culture is focused on designing something that is aesthetically pleasing branding strategies to help provide an easy to use platform and system to sell customers needs. This designer will also ensure that the end-to-end journey of our customers using our platform meets desired outcomes.",
  "benefits": "Excellent compensation package, great benefits, professional development opportunities, competitive salary, stock options, 401(k), health insurance, dental insurance, life insurance, short-term disability, long-term disability, vision insurance, and more.",
  "company": {
    "name": "Arbaa",
    "display_name": "Arbaa",
    "slug": "arbaa",
    "url": "https://www.arbaa.com",
    "description": "Arbaa is a fast growing trading platform, Customer Portals and Back Office Systems. Arbaa's mission is to make the world a better place by creating better user experiences. Arbaa's culture is focused on designing something that is aesthetically pleasing branding strategies to help provide an easy to use platform and system to sell customers needs. This designer will also ensure that the end-to-end journey of our customers using our platform meets desired outcomes."
  },
  "address": {
    "region": "Hà Nội",
    "district": "Hà Đông",
    "ward": "Trung Văn",
    "street": "Khuất Thúy Saigon Pearl, Số 92 Nguyễn Hữu Cánh, Phường 10, Quận Kim Thanh, Thành phố Hà Nội",
    "full_address": "Khuất Thúy Saigon Pearl, Số 92 Nguyễn Hữu Cánh, Phường 10, Quận Kim Thanh, Thành phố Hà Nội"
  },
  "salary": {
    "min": 10000000,
    "max": 20000000,
    "unit": "VNĐ",
    "currency": "VND"
  }
}

```

Hình 4. File json trả về từ Server(TopDev)

Về cơ bản, các thông tin đã đầy đủ cho mục đích, tuy nhiên vẫn còn thiếu một số thông tin như: Hạn nộp CV, Mô tả, Yêu Cầu, Số lượng, Kinh nghiệm. Những thông tin này ta sẽ trích xuất trong mã html trả về của trang web thay vì file json API.

```

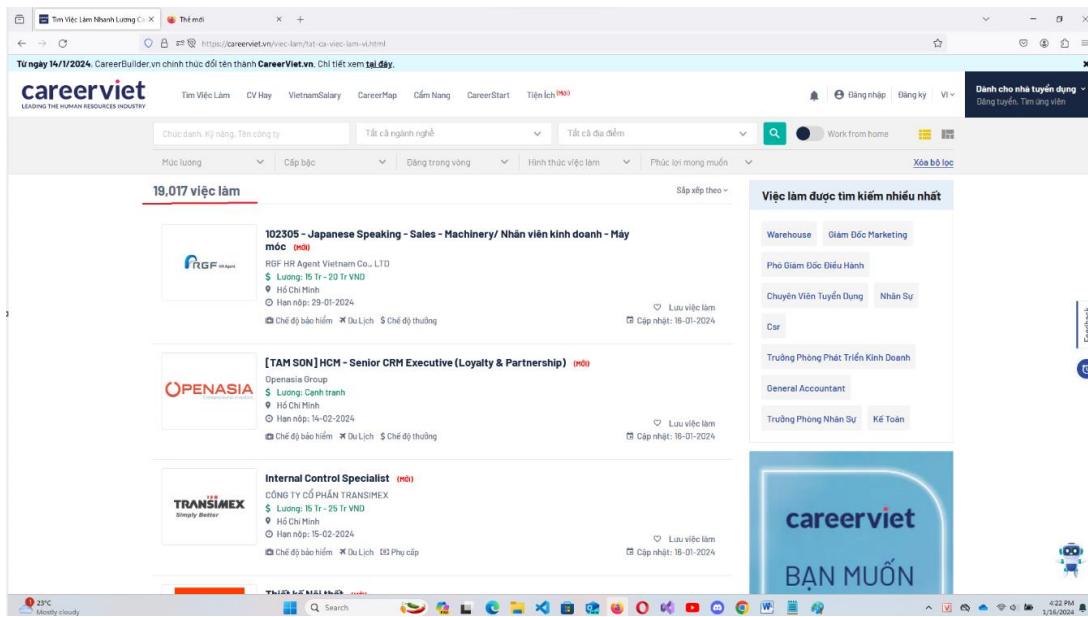
{
  "id": "1234567890",
  "title": "User Interface / user experience designer",
  "content": "We are looking for a UI/UX designer to design web interfaces as well as mobile interfaces required turning our software into easy-to-use products for our clients. Arbaa is a fast growing trading platform, Customer Portals and Back Office Systems. Arbaa's mission is to make the world a better place by creating better user experiences. Arbaa's culture is focused on designing something that is aesthetically pleasing branding strategies to help provide an easy to use platform and system to sell customers needs. This designer will also ensure that the end-to-end journey of our customers using our platform meets desired outcomes.",
  "benefits": "Excellent compensation package, great benefits, professional development opportunities, competitive salary, stock options, 401(k), health insurance, dental insurance, life insurance, short-term disability, long-term disability, vision insurance, and more.",
  "company": {
    "name": "Arbaa",
    "display_name": "Arbaa",
    "slug": "arbaa",
    "url": "https://www.arbaa.com",
    "description": "Arbaa is a fast growing trading platform, Customer Portals and Back Office Systems. Arbaa's mission is to make the world a better place by creating better user experiences. Arbaa's culture is focused on designing something that is aesthetically pleasing branding strategies to help provide an easy to use platform and system to sell customers needs. This designer will also ensure that the end-to-end journey of our customers using our platform meets desired outcomes."
  },
  "address": {
    "region": "Hà Nội",
    "district": "Hà Đông",
    "ward": "Trung Văn",
    "street": "Khuất Thúy Saigon Pearl, Số 92 Nguyễn Hữu Cánh, Phường 10, Quận Kim Thanh, Thành phố Hà Nội",
    "full_address": "Khuất Thúy Saigon Pearl, Số 92 Nguyễn Hữu Cánh, Phường 10, Quận Kim Thanh, Thành phố Hà Nội"
  },
  "salary": {
    "min": 10000000,
    "max": 20000000,
    "unit": "VNĐ",
    "currency": "VND"
  }
}

```

Hình 5. Khảo sát lương trong file json

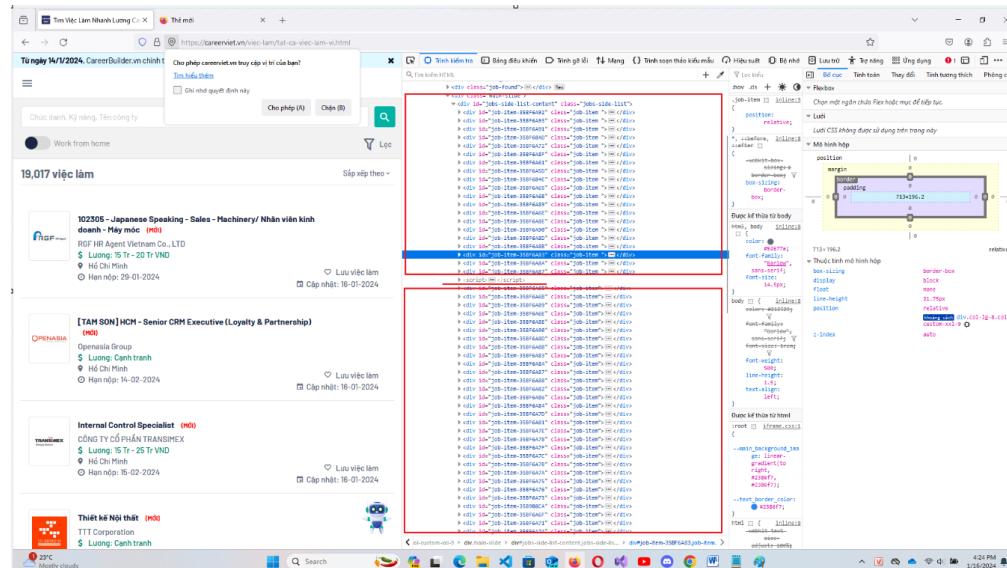
Ngoài ra còn một vấn đề nữa, đó là nếu không đăng nhập thì trang web không hiển thị cho ta xem lương và trong file json API cũng không xem được lương, vì vậy để lấy lương của công việc, ta sẽ tìm trong mã html trả về khi ta gửi yêu cầu tới url công việc.

b) CareerBuilder



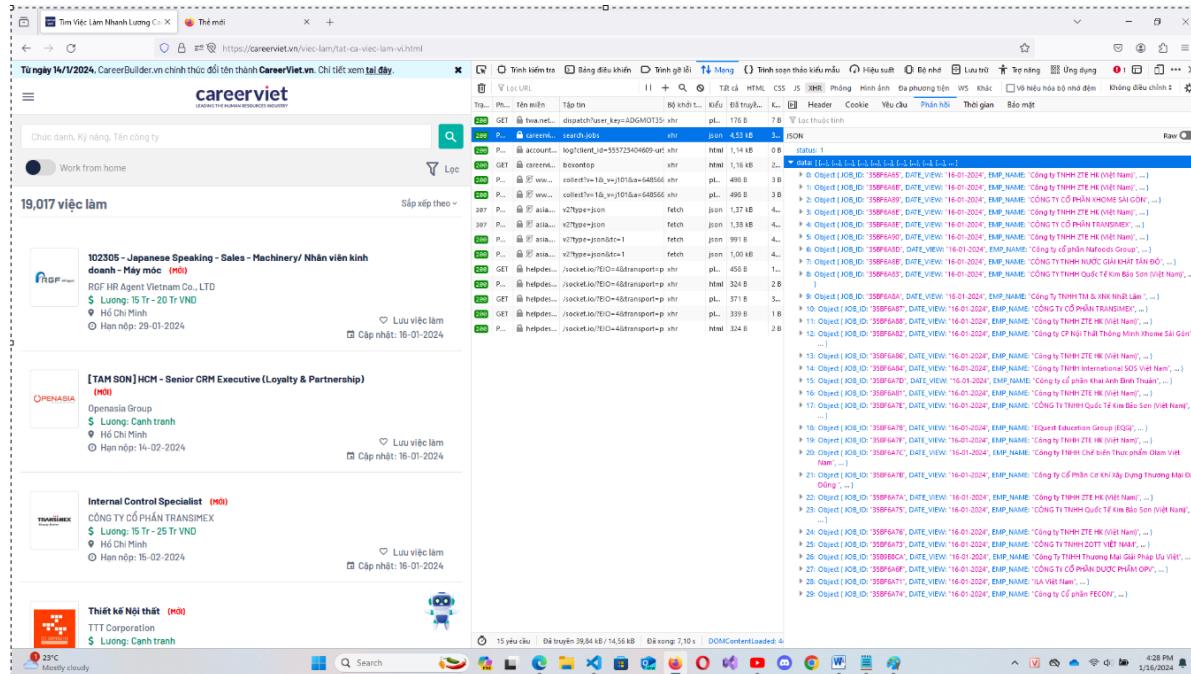
Hình 6. Khảo sát danh sách tin CareerBuilder

Qua khảo sát ta thấy, 1 trang CareerBuilder sẽ trả cho ta 50 công việc, vì vậy ta sẽ lấy số công việc này để tính ra trang cuối cùng cần truy cập tới.

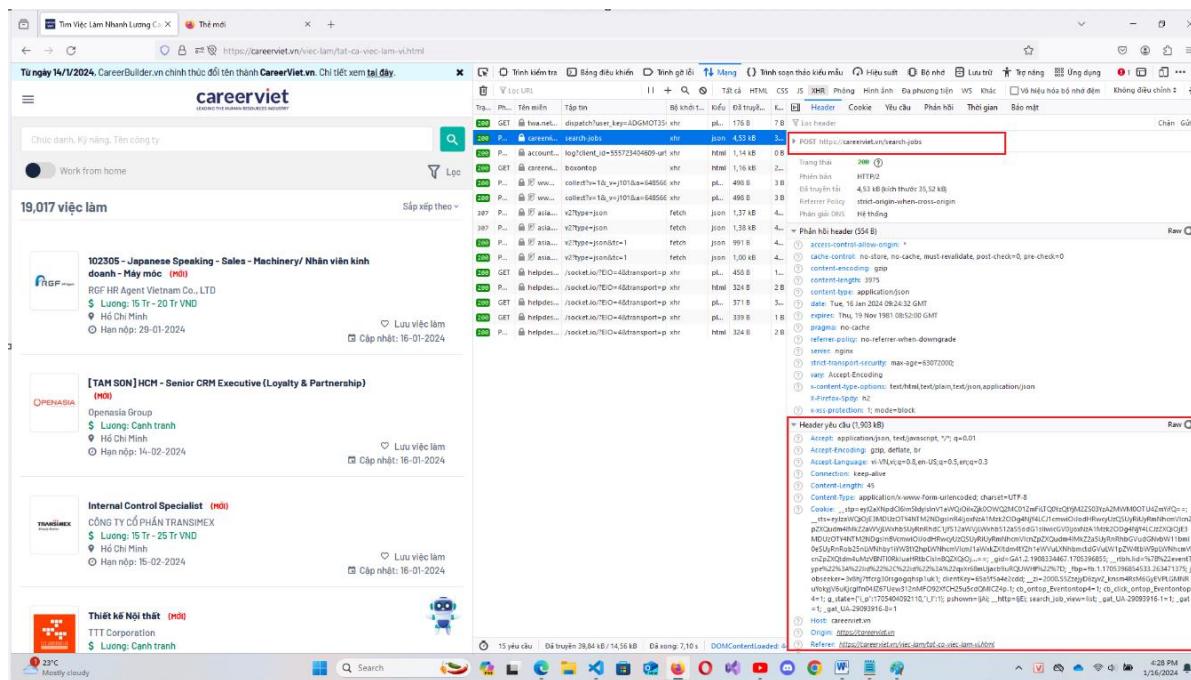


Hình 7. Khảo sát danh sách tin CareerBuilder

Khảo sát cũng cho thấy khi gửi yêu cầu tới trang thì chỉ có 20/50 công việc ở ô màu đỏ phía trên là được trả dưới dạng mã html, còn 30 công việc phía dưới không được trả dưới dạng mã html mà được trả về dưới dạng API.



Hình 8. API trả về từ Server(CareerBuilder)



Hình 9. Khảo sát request header của API(CareerBuilder)

Sau khi khảo sát về cách trả về dữ liệu của trang web, ta khảo sát tới cấu trúc dữ liệu của một tin tuyển dụng :

Careerviet
LEADING THE HUMAN RESOURCES INDUSTRY

Tìm Việc Làm CV Hay VietnamSalary CareerMap Cẩm Nang CareerStart Tiện Ích [Hỗ trợ]

Danh cho nhà tuyển dụng
Đăng tuyển. Tìm ứng viên

[CIS] Senior Content Creator
Canadian International School

Chi tiết Tổng quan công việc

Địa điểm Hồ Chí Minh **Ngày cập nhật** 16/01/2024 **Lương** Cạnh tranh

Ngành nghề Tiếp thị / Marketing , Giáo dục / Bảo tạo , Truyền hình / Báo chí / Biên tập **Kinh nghiệm** 2 - 5 Năm

Hình thức Nhân viên **Cấp bậc** Nhân viên **Hết hạn nộp** 16/02/2024

PHỤC LỢI

- Chế độ bảo hiểm
- Chế độ thường
- Nghỉ phép năm
- Du Lịch
- Chăm sóc sức khỏe
- CLB thể thao
- Phụ cấp
- Bảo tần

MÔ TẢ CÔNG VIỆC

MAIN RESPONSIBILITIES

Content Strategy:

- Develop and execute a comprehensive content strategy aligned with the overall marketing and business goals of CIS.

Lưu việc làm Gửi tôi việc làm tương tự Báo xấu

NỘP ĐƠN ỨNG TUYỂN

CAC CÔNG VIỆC TƯỞNG TỰ

- EQUEST** [CIS] Copy Editor...
Canadian...
\$ Lương: Cạnh Tranh
Hồ Chí Minh
- happyNHN** Content Creator
Công ty cổ phần...
\$ Lương: 12 Tr - 15 Tr VNĐ
Hồ Chí Minh
- SYBSY** Content Creator
SYBSY Ltd.
\$ Lương: 10 Tr - 15 Tr VNĐ
Hà Nội
Hồ Chí Minh
- VJ - Content...** CÔNG TY TNHH MT...
\$ Lương: Cạnh Tranh
Hồ Chí Minh
- NHÂN VIÊN...** CÔNG TY TNHH TM...
\$ Lương: 10 Tr - 12 Tr VNĐ
Hà Nội
Hồ Chí Minh
- Product...** Công ty TNHH Lien...
\$ Lương: 7.5 Tr - 9 Tr VNĐ
Hồ Chí Minh
- Product...** Công ty TNHH Lien...
\$ Lương: 10 Tr - 15 Tr VNĐ
Hồ Chí Minh
- Fullstack...** HOSSOFT COMPANY...
\$ Lương: Cạnh Tranh
Hồ Chí Minh

Hình 10. Khảo sát cấu trúc tin CareerBuilder

careerviet
Leading the Human Resources Industry

Tìm ngày 16/01/2024 | [Tìm kiếm công việc](#)

Cho phép careerviet.vn truy cập vị trí của bạn? **Tin hiệu them**

Ghi nhớ quyết định này

Cho phép (A) Chặn (B)

Địa chỉ Singapore | **Ngày đăng** 16/01/2024 | **Lương** \$ 10 Tr - 15 Tr VNĐ | **Thời gian** 16/01/2024 | **Công việc** Công việc mới | **Đóng góp** 0 | **Đã xem** 0 | **Đã ứng tuyển** 0

ĐĂNG KÝ | **ĐĂNG NHẬP** | **ĐĂNG KÝ** | **VI**

Danh cho nhà tuyển dụng
Đăng tuyển. Tìm ứng viên

[OnOff] Nhân viên
Công ty TNHH TH...
\$ Lương: 10 Tr - 15 Tr VNĐ
Hồ Chí Minh

CRIFER
[OnOff] Nhân viên...
Công ty TNHH TH...
\$ Lương: 7.5 Tr - 9 Tr VNĐ
Hồ Chí Minh

TMI
[OnOff] Nhân viên...
Công ty TNHH Lien...
\$ Lương: 10 Tr - 15 Tr VNĐ
Hồ Chí Minh

Product...
Công ty TNHH Lien...
\$ Lương: 7.5 Tr - 9 Tr VNĐ
Hồ Chí Minh

Product...
Công ty TNHH Lien...
\$ Lương: 10 Tr - 15 Tr VNĐ
Hồ Chí Minh

Fullstack...
HOSSOFT COMPANY...
\$ Lương: Cạnh Tranh
Hồ Chí Minh

XEM TẤT CẢ

Hình 11. Khảo sát cấu trúc tin CareerBuilder

c) CareerLink

The screenshot shows the CareerLink website interface. At the top, there is a search bar with fields for 'Nhập tên vị trí, công ty, từ khóa' and 'Nhập tỉnh, thành phố', a 'Tìm kiếm' button, and navigation links for 'Đăng ký', 'Nhà tuyển dụng', and other filters like 'Ngành nghề', 'Cấp bậc', 'Kinh nghiệm', 'Mức lương', 'Học vấn', 'Loại công việc', and 'Đăng trong'. Below the search bar, it says 'Kết quả tìm kiếm' and '20391 việc làm'. The results list several job posts from different companies:

- Nhân Viên Kinh Doanh Thủ Án Gia Súc (Kv: Tp. HCM-Long An)** at Uni-President Vietnam, Hồ Chí Minh, Long An. Job type: Thực tập | Nhân viên. Published: 2 giờ trước.
- NỮ - NHÂN VIÊN NGHIÊN CỨU** at Uni-President Vietnam, Tiền Giang. Job type: Thực tập | Nhân viên. Published: 2 giờ trước.
- (TUYỂN GẤP) KỸ SƯ CƠ KHÍ CHẾ TẠO MÁY** at Công ty TNHH TM DV Hồng Dương, Hồ Chí Minh. Job type: Kỹ thuật viên / Kỹ sư. Published: 2 giờ trước.
- NHÂN VIÊN BẢO TRÌ TỐT NGHIỆP CHUYÊN NGÀNH ĐIỆN - NHÀ MÁY CASTA LONG KHÁNH** at Công ty CP TMDV ĐÁT MỚI (ALC CORP), Đồng Nai. Job type: Nhân viên. Published: 2 giờ trước.
- CHUYÊN VIÊN KẾ TOÁN** at Công ty TNHH TM DV Hồng Dương. Job type: Nhân viên. Published: 2 giờ trước.

On the right side of the results page, there is a sidebar for 'Tạo CV chất với VietCV.io' and 'Ứng tuyển việc làm với CareerLink.vn', featuring links to download resume templates and apply for jobs.

Hình 12. Khảo sát danh sách tin CareerLink

Sau khi vô hiệu hóa Javascript và tải lại trang, ta kết luận các công việc trên trang được trả về dưới dạng mã html.

The screenshot shows a browser window with the search results from the previous screenshot, but with developer tools open on the right side. The developer tools are used to inspect the HTML structure of the page. The 'Elements' tab is selected, showing the DOM tree for the job listing cards. The 'Styles' tab shows the CSS applied to the elements, and the 'Network' tab shows the network requests made by the page. The 'Console' tab at the bottom shows the message 'Highlights from the Chrome 120 update'.

Hình 13. Khảo sát kiểu trả về của các công việc trang web

Một trang như này sẽ trả về cho ta 50 công việc, vì vậy ý tưởng là sẽ lấy số lượng tổng cộng công việc để tính được trung kết thúc.

Khảo sát cấu trúc tin:

[CÁN THỢ VIỆC LÀM TIẾNG NHẬT] TUYỂN GẤP - NHÂN VIÊN THỜI VỤ NHẬP LIỆU TIẾNG NHẬT - LÀM VIỆC TẠI CÔNG TY (CA 7H10 - 17H, NGHỈ T7 & CN)

ĐỊA ĐIỂM LÀM VIỆC: Lầu 9, tòa nhà Techcombank, 45-47 đường 30/4, Ninh Kiều, Cần Thơ

LƯƠNG: - Dào tạo dự án 2 ngày: 35.000VNĐ/giờ

Mô tả công việc

* MÔ TẢ CHI TIẾT CÔNG VIỆC:

- Nhập dữ liệu tiếng Nhật vào hệ thống theo hướng dẫn của dự án. Bạn sẽ được đào tạo dự án.
- Thời gian làm việc: 07:10 - 17:00, T2 - T6, nghỉ T7 & CN.
- Dự án sẽ bắt đầu từ tháng 01 -> tháng 03 (Có làm Tết, nghỉ mùng 1.2.3)
- Địa điểm làm việc: Lầu 9, tòa nhà Techcombank, 45-47 đường 30/4, Ninh Kiều, Cần Thơ

* LƯƠNG:

Nhân Viên Kỹ Thuật - Bảo Trì

HỆ THỐNG Y KHOA ÁI NGHĨA

Đồng Nai

\$ Thưởng lương

Thư Ký Bếp Khách Sạn

KHÁCH SẠN NESTA CẦN THƠ

Cần Thơ

\$ Thưởng lương

Kế Toán Trưởng

KHÁCH SẠN NESTA CẦN THƠ

Cần Thơ

\$ Thưởng lương

Giám đốc/Trưởng phòng Khách Hàng Cá Nhân - Cà Ná

Công Ty Cổ Phần Chứng khoán Rồng Việt

Cần Thơ

\$ 15 triệu - 25 triệu

Giám đốc Điều hành Khách Sạn

KHÁCH SẠN NESTA CẦN THƠ

Cần Thơ

\$ Cạnh tranh

Nhân Viên Nhà Hàng Nesta

Hình 14. Khảo sát cấu trúc tin CareerLink

NHÂN VIÊN BẢO TRÌ TỐT NGHIỆP

Nhập tên vị trí, công ty, từ khóa

Nhập tỉnh, thành phố

Mô tả Kỹ năng yêu cầu Chi tiết công việc Liên hệ Võ công ty

Về sinh may móc, tinh tế và khéo léo, am hiểu, nòng nhẹ.

Thực hiện kiểm tra, Bảo dưỡng bảo trì máy móc thiết bị hàng ngày theo bảng danh mục

Hướng dẫn công thường xuyên kiểm tra bảo dưỡng máy thiết bị, công cụ dụng cụ hàng tuần, hàng tháng

Giảm thiểu thời gian ngưng máy, tăng thời hạn máy, giảm chi phí sản xuất

Các nhiệm vụ khác được giao bởi cấp trên

Kinh nghiệm / Kỹ năng chi tiết

Giới tính: Nam

Tốt nghiệp Trung cấp trở lên chuyên ngành điện công nghiệp, điện tự động hoặc ngành khác có liên quan

Không yêu cầu kinh nghiệm

Tinh thần làm việc nhóm; giao tiếp hiệu quả

Mô tả

Loại công việc

Nhân viên toàn thời gian

Cấp bậc

Nhân viên

Học vấn

Trung cấp

Ngành nghề

Kỹ thuật ứng dụng / Cơ khí , Điện / Điện tử , Bảo trì / Sửa chữa

Thông tin liên hệ

Tên liên hệ: Bộ Phận Nhân Sự

Lô B2, Đường C2, KCN Cát Lái, Cụm 2, Phường Thạnh Mỹ Lợi, (Bên cạnh Trạm thu phí cầu Phú Mỹ), Thành Phố Thủ Đức, Hồ Chí Minh, Việt Nam

Quản lý Bộ Phận Cát Vá Bảo Trì

CÔNG TY TNHH CÁNH ĐỒNG VÀNG

Đồng Nai

\$ Thưởng lương

Nhân Viên Văn Hành Máy - Làm Việc Tại

CÁNH ĐỒNG

Đồng Nai

\$ 10 triệu - 15 triệu

NHÂN VIÊN BẢO TRÌ CƠ KHÍ

CÔNG TY TNHH CÁNH ĐỒNG VÀNG

Đồng Nai

\$ 15 triệu - 20 triệu

ofi

NHÀ MÁY GIANG ĐIỀN- CTY TNHH OLAM VIỆT NAM

Đồng Nai

\$ Thưởng lương

Quản lý Bộ Phận Cát Vá Bảo Trì

CÔNG TY TNHH Matsuya R&D (Việt Nam)

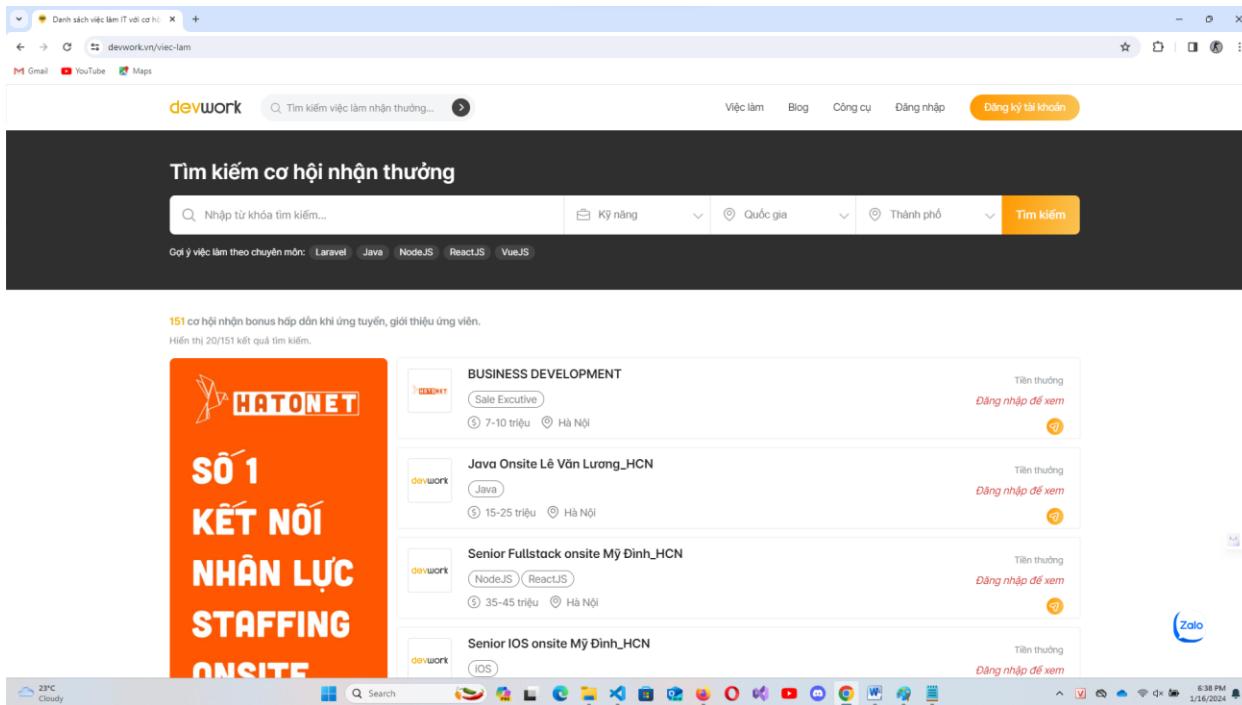
Đồng Nai

\$ Thưởng lương

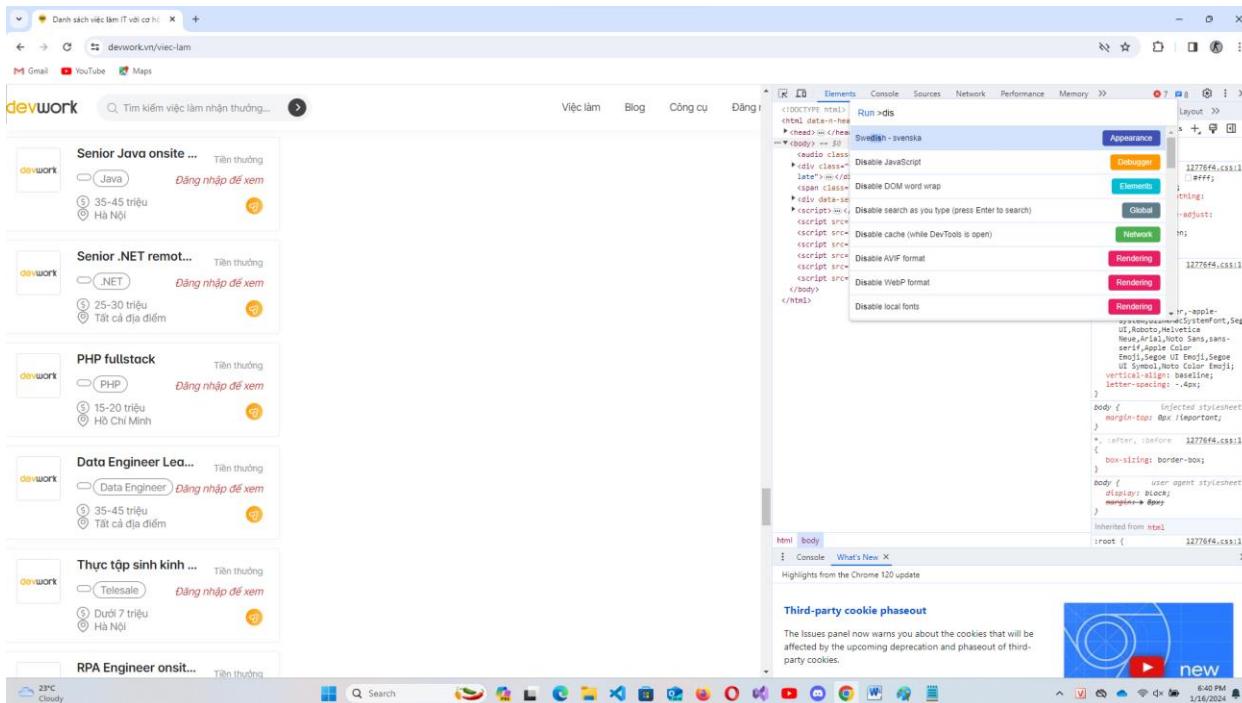
Chào bạn! Minh có thể giúp gì cho bạn?

Hình 15. Khảo sát cấu trúc tin CareerLink

d) DevWork



Hình 16. Khảo sát danh sách tin DevWork



Hình 17. Khảo sát kiểu trả về của các tin DevWork

Khi ta vô hiệu hóa Javascript trên trang web, thì thấy các công việc trên web vẫn được tải ra như bình thường, vậy ta có thể lấy các công việc của trang thông qua mã html.

Khảo sát tin tuyển dụng của trang ta được:

The screenshot shows a job listing for an RPA Engineer onsite Trung Kính LA. The page has a dark header with the DevWork logo and navigation links. The main content includes:

- Job Title:** RPA Engineer onsite Trung Kính LA
- Company:** Công ty Cổ phần Phần mềm Devwork
- Location:** Hà Nội
- Skills:** C#, Java, PHP, Python
- Description:** Mô tả công việc (Job Description) mentioning RPA tools like RPA-UIPath, RPA/Microsoft Power Automate, RPA:Automation Anywhere, LCAP-Microsoft Power platform, LCAP-Salesforce, AI: Line Cloba OCR, AI: Microsoft Axure OpenAI.
- Requirements:** Yêu cầu công việc (Job Requirements) mentioning Python, Java, .NET, C#, VBA, etc., and experience in RPA.
- Benefits:** Tiền thưởng (Bonus) - Đăng nhập để xem (Log in to view), Mức lương (Salary) - 15-25 triệu (15-25 million VND).
- Information:** Thông tin (Information) including Kinh nghiệm (Experience) - 1 năm (1 year), Trình độ (Level) - Không yêu cầu (No requirements), Vị trí (Position) - Junior, Loại công việc (Job Type) - Backend.

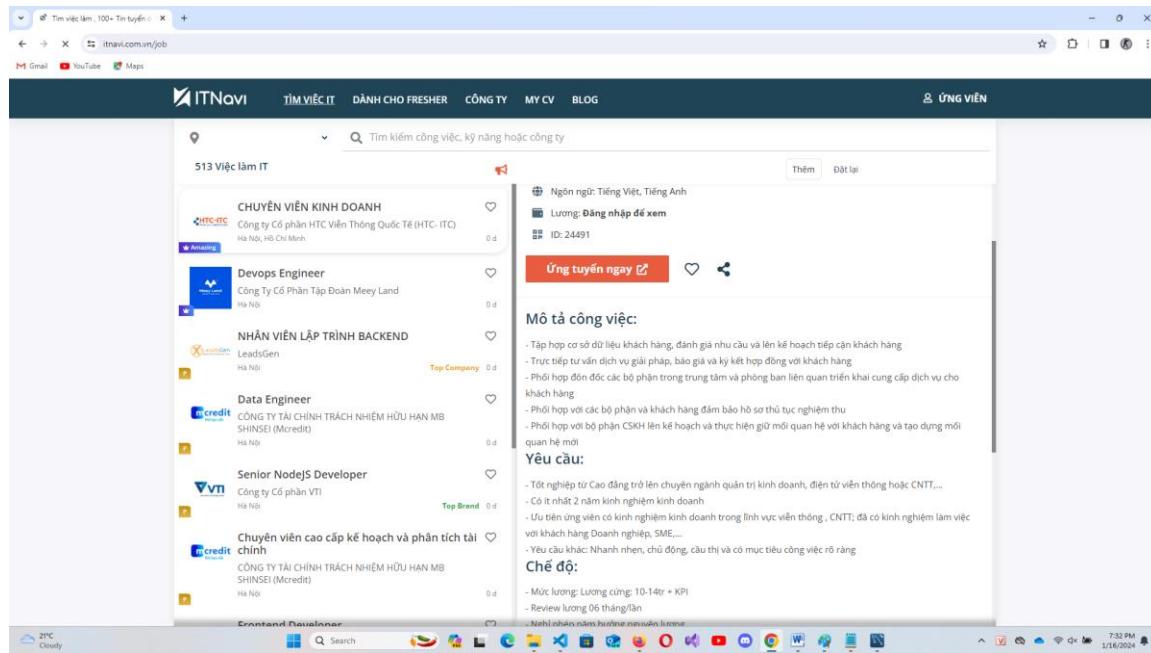
Hình 18. Khảo sát cấu trúc tin DevWork

The screenshot shows a job listing for an iOS developer. The page has a dark header with the DevWork logo and navigation links. The main content includes:

- Job Title:** Kỹ năng (Skills) - iOS
- Description:** Mô tả công việc (Job Description) mentioning building software for iOS, conducting research, and performing other tasks.
- Requirements:** Yêu cầu công việc (Job Requirements) mentioning Swift, Objective-C, English proficiency, payment work, and working days/times.
- Information:** Thông tin (Information) including Kinh nghiệm (Experience) - 5 năm (5 years), Trình độ (Level) - Không yêu cầu (No requirements), Vị trí (Position) - Senior, Loại công việc (Job Type) - Mobile Apps, Hình thức (Type) - Full-time, Hạn nộp hồ sơ (Deadline) - 2024-02-14, Số lượng (Quantity) - 1 người (1 person), Quy trình phỏng vấn (Interview Process) - 1 vòng (1 round).

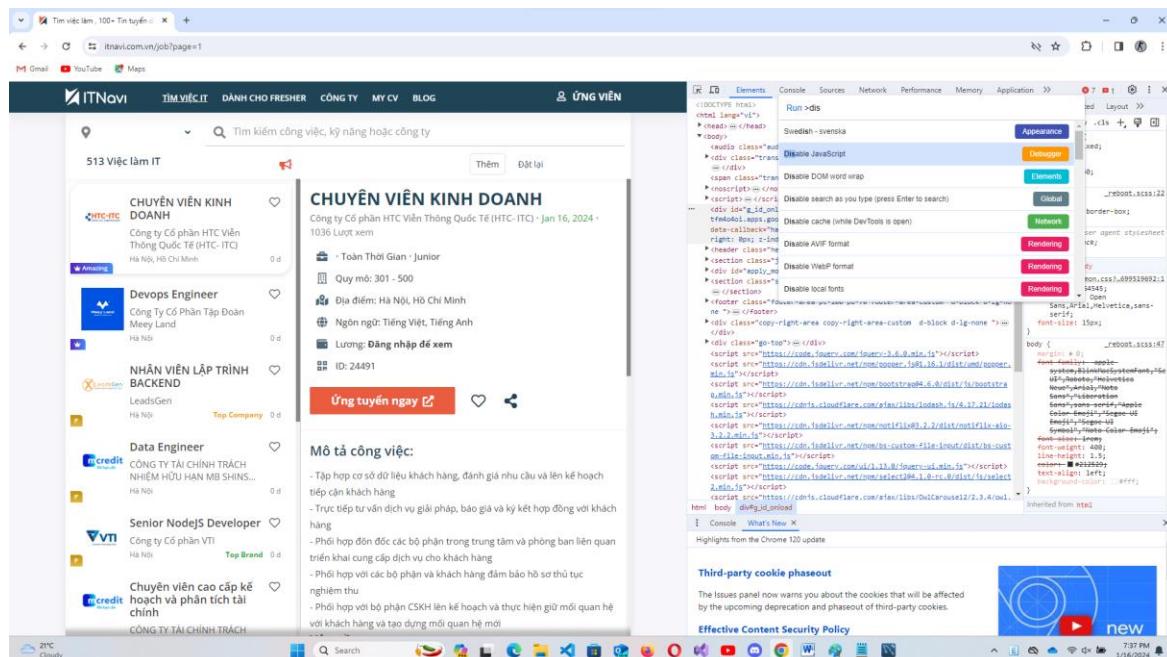
Hình 19. Khảo sát cấu trúc tin DevWork

e) ITNaVi



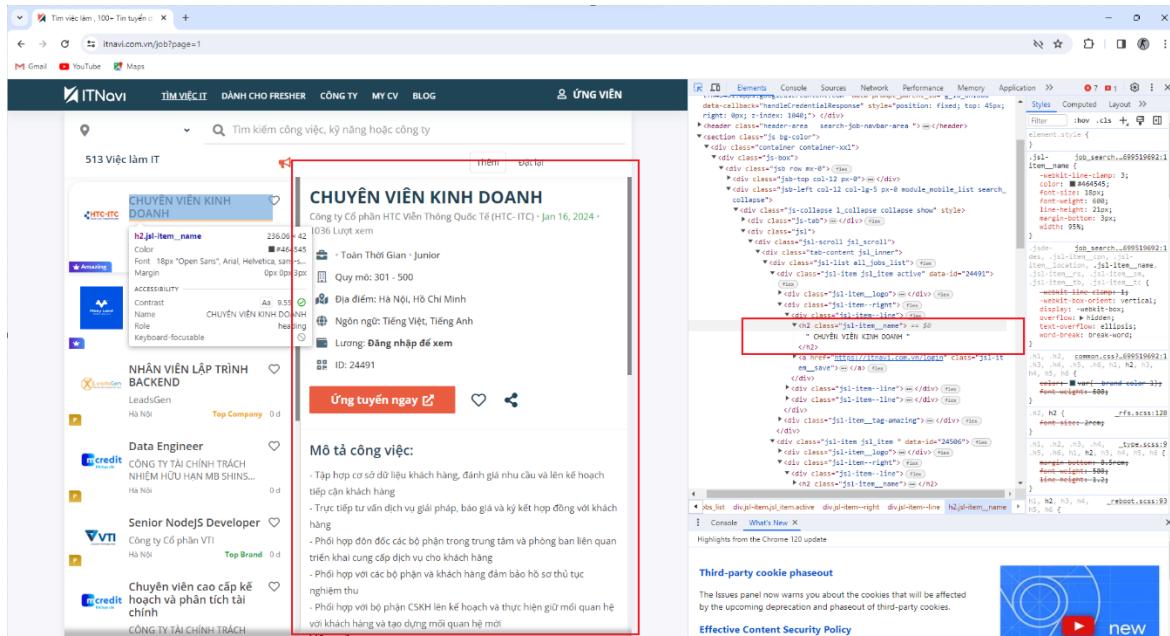
Hình 20. Khảo sát danh sách tin ITNaVi

Vô hiệu hóa Javascript và tải lại trang, ta thấy các công việc của trang web được trả dưới dạng mã html.



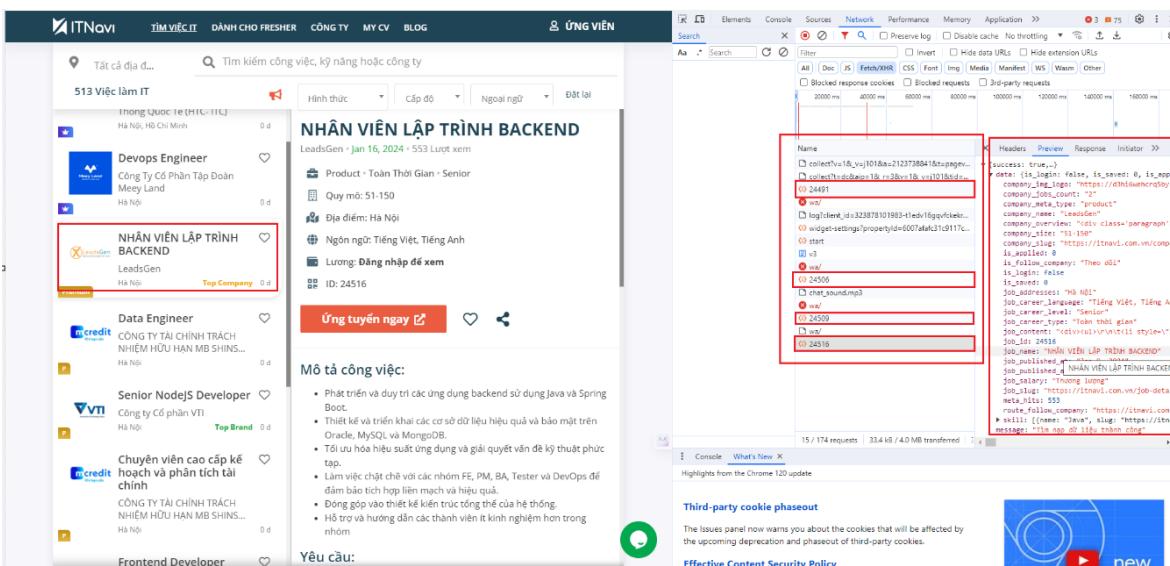
Hình 21. Khảo sát kiểu trả về của các tin DevWork

Tuy nhiên có một vấn đề đó là khi ta nhấp vào một công việc thì trang web sẽ không tải ra một cửa sổ mới mà lại tải luôn thông tin vào phần bên tay phải (phần ô màu đỏ to) và ta cũng không thể tìm được đường dẫn đến công việc đó (phần ô màu đỏ nhỏ) trong mã html.



Hình 22. Khảo sát mã html trả về trang web

Tuy nhiên kiểm tra kết quả trả về từ server khi click vào 1 công việc bất kì, ta nhận được 1 file dạng json với rất nhiều thông tin như hình dưới. Các số 24491, 24506, 24509, 24516 đường như là mã định danh cho công việc trên trang web.



Hình 23. Khảo sát API trả về từ Server(ITNavI)

The screenshot shows the ITNaVi browser extension interface. On the left, a job detail page for 'NHÂN VIÊN LẬP TRÌNH BACKEND' is displayed, listing requirements like Java/Spring Boot experience and a salary range of 51-150k. On the right, the Network tab of the developer tools shows a request to 'https://itnavi.com.vn/api/getJob-by-id/24516'. The Headers section shows the 'Request URL' and 'Request Method' (GET). The Response Headers section includes 'Cache-Control: no-cache, private', 'Content-Type: application/json', 'Date: Tue, 16 Jan 2024 15:44:45 GMT', 'Server: nginx', and 'Set-Cookie: 24516'. The Set-Cookie value is highlighted with a red box. The Response tab shows the JSON response content.

Hình 24. Khảo sát request header API(ITNaVi)

This screenshot shows a detailed view of a JSON response body from a request to 'https://itnavi.com.vn/api/job-detail/nhan-vien-lap-trinh-backend-TBWg4'. The response contains various job details and requirements. A specific cookie value '24516' is highlighted with a red box in the Set-Cookie field of the Headers section. The JSON content is displayed in the Response tab.

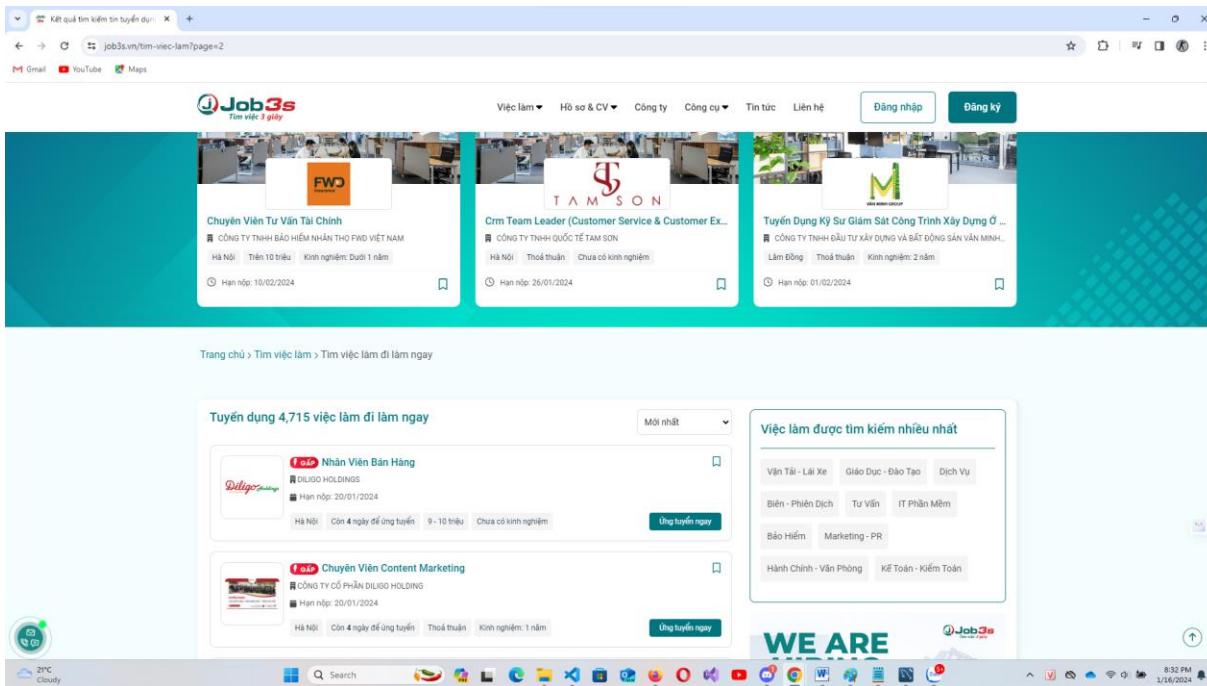
Hình 25. Khảo sát ngày hết hạn tin (ITNaVi)

The screenshot shows a job listing on the ITNaVi website. At the top, there's a navigation bar with links for 'TÌM VIỆC IT', 'DÀNH CHO FRESHER', 'CÔNG TY', 'MY CV', 'BLOG', and 'ỨNG VIÊN' (Job Seeker). The main content area has a section titled 'Tổng quan' (Overview) which includes details like 'Product - Toàn Thời Gian - Senior', 'Quy mô: 51-150', 'Jan 16, 2024', and '3 ứng viên'. Below this is a 'Mô tả công việc' (Job Description) section with a bulleted list of requirements, followed by a 'Yêu cầu:' (Requirements) section with another bulleted list. On the right side, there's a sidebar with sharing options for social media and a 'Chia sẻ công việc này' (Share this job) button.

Hình 26. Khảo sát cấu trúc tin(ITNaVi)

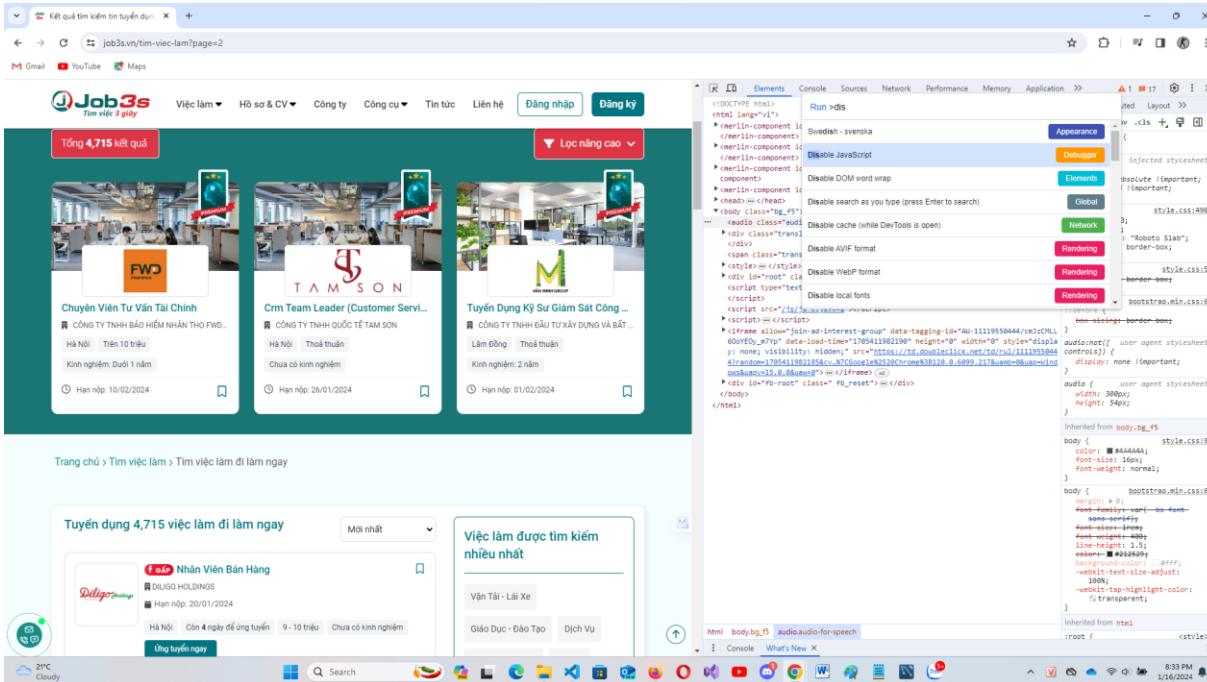
Qua khảo sát thấy mỗi trang web chứa danh sách công việc khi được tải ra sẽ hiển thị 10 phần tử, tương ứng với 10 công việc khác nhau.

f) Job3S

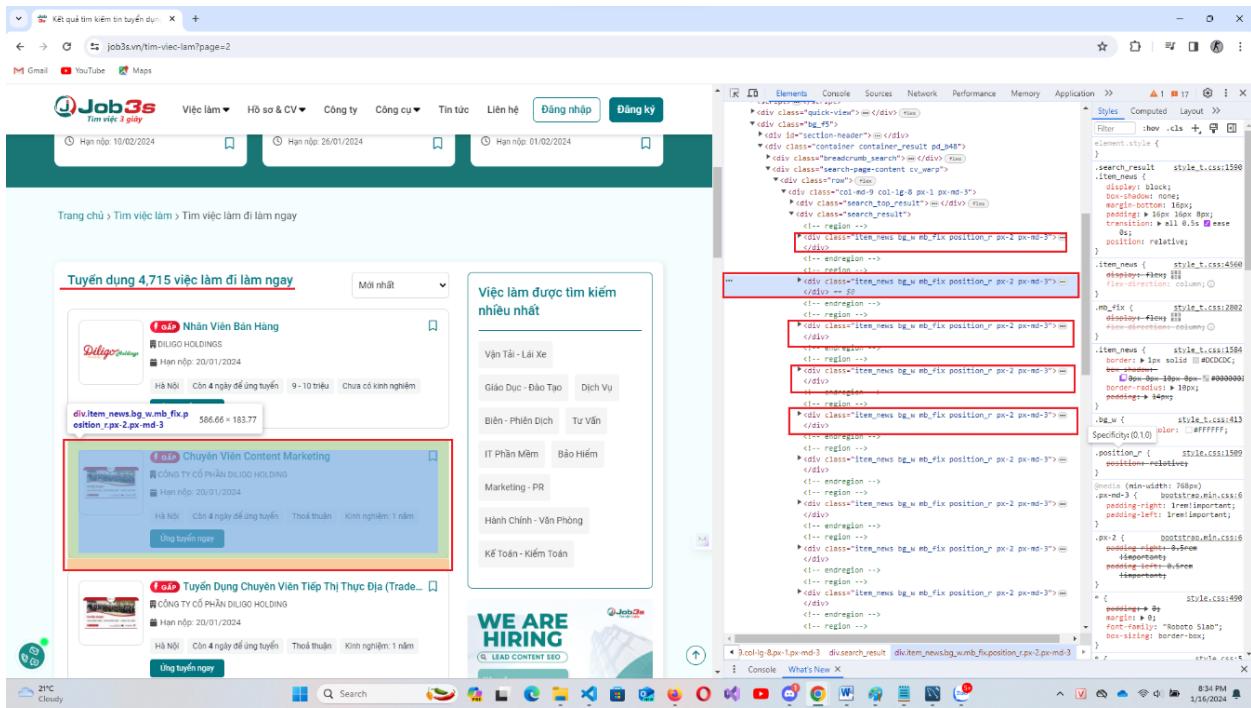


Hình 27. Khảo sát danh sách tin Job3S

Vô hiệu hóa Javascript và tải lại trang web, ta kết luận các công việc trên trang web được trả về dưới dạng mã html.



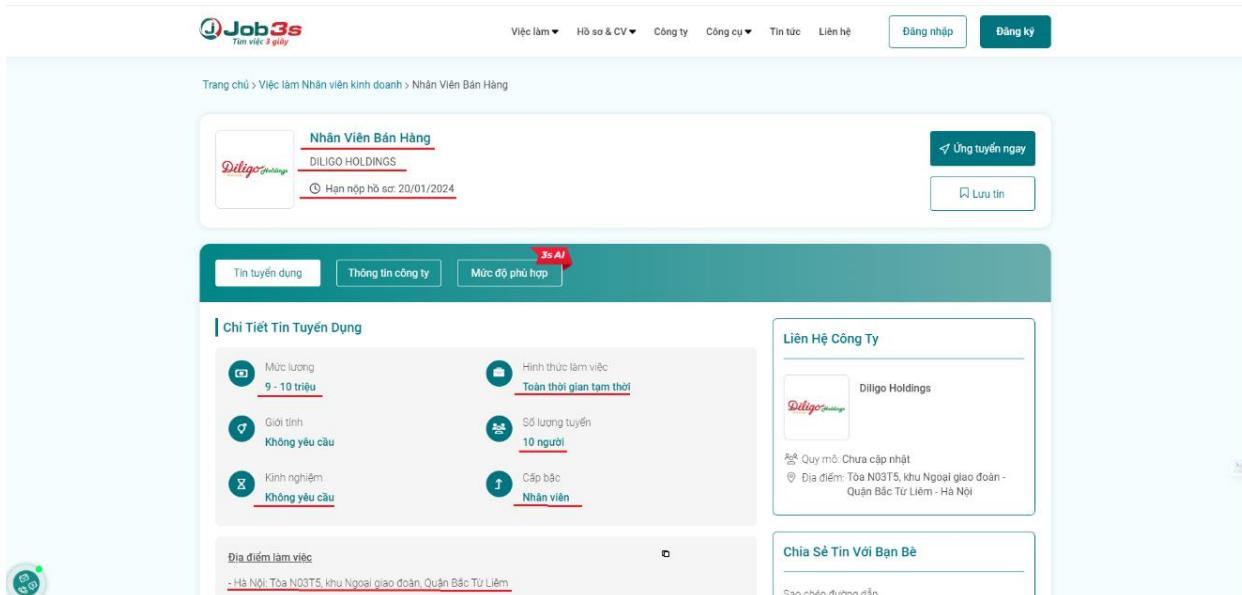
Hình 28. Khảo sát kiểu trả về các tin



Hình 29. Khảo sát danh sách tin Job3s

Khảo sát cũng cho thấy mỗi trang web sẽ trả về cho ta 15 url, vậy từ số lượng công việc, ta sẽ tính được trang cuối cùng cần gửi yêu cầu.

Khảo sát cấu trúc tin của 1 trang web, ta được như sau:



Hình 30. Khảo sát cấu trúc tin Job3s

The screenshot shows a job listing on the Job3s website. The page header includes links for 'Việc làm', 'Hồ sơ & CV', 'Công ty', 'Công cụ', 'Tin tức', 'Liên hệ', 'Đăng nhập' (Login), and 'Đăng ký' (Sign up). The main content area displays a job description for a 'Nhân viên bán hàng' position. The description includes requirements like 'Đảm bảo hàng hoá tại bộ phận phát đầy đủ, mã hàng hoá, loại hàng hoá...', responsibilities such as 'Nhập hàng, lên bảng kê những sản phẩm còn thiếu dựa theo số lượng hàng tồn và tốc độ tiêu thụ hàng hoá, chuyển sang cho cửa hàng trưởng xem xét và bao vệ công ty để đặt hàng. Luôn chủ động trong việc đặt hàng và đảm bảo hàng luôn đầy đủ để bán...', and other details like 'Xuất bản: Luôn để ý tới khu vực trưng bày để giúp khách hàng lựa chọn sản phẩm. Theo dõi tốc độ tiêu thụ của mỗi mã hàng hóa và báo cáo chi tiết số lượng hàng mỗi ngày...'. To the right, there's a sidebar with a QR code for the app, download links for Google Play and App Store, and social media links for Facebook, LinkedIn, and Twitter.

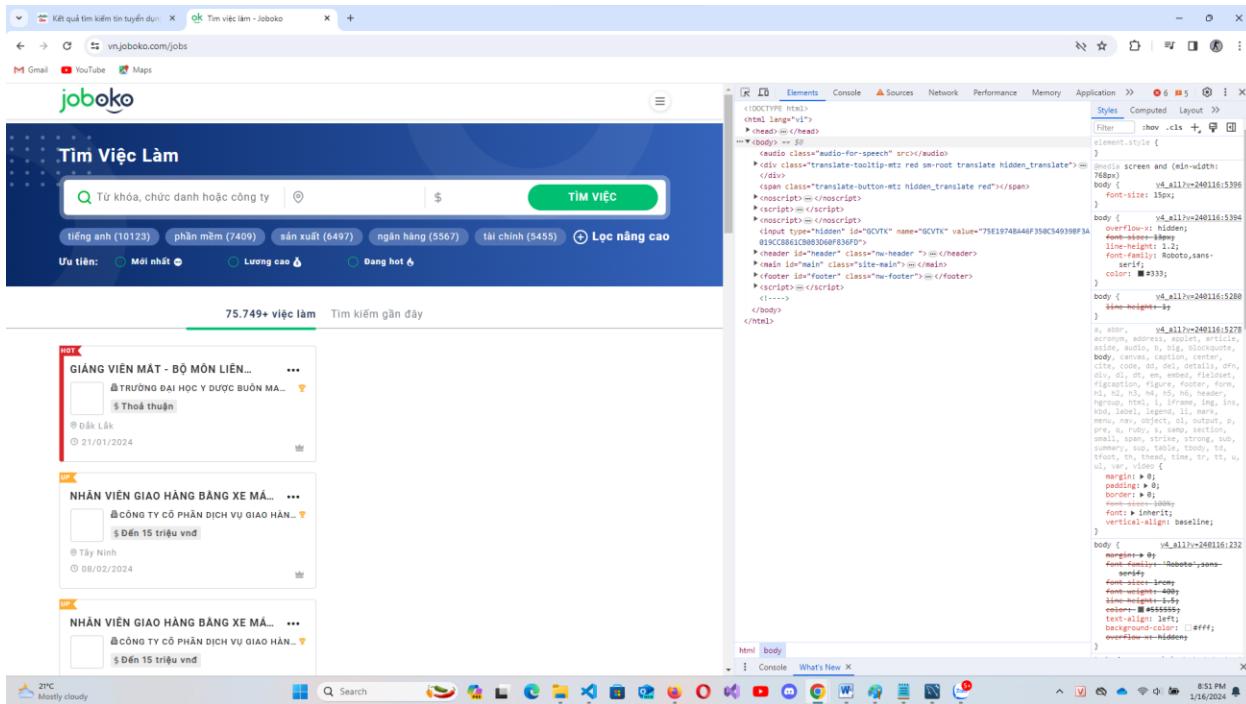
Hình 31. Khảo sát cấu trúc tin Job3s

g) Joboko

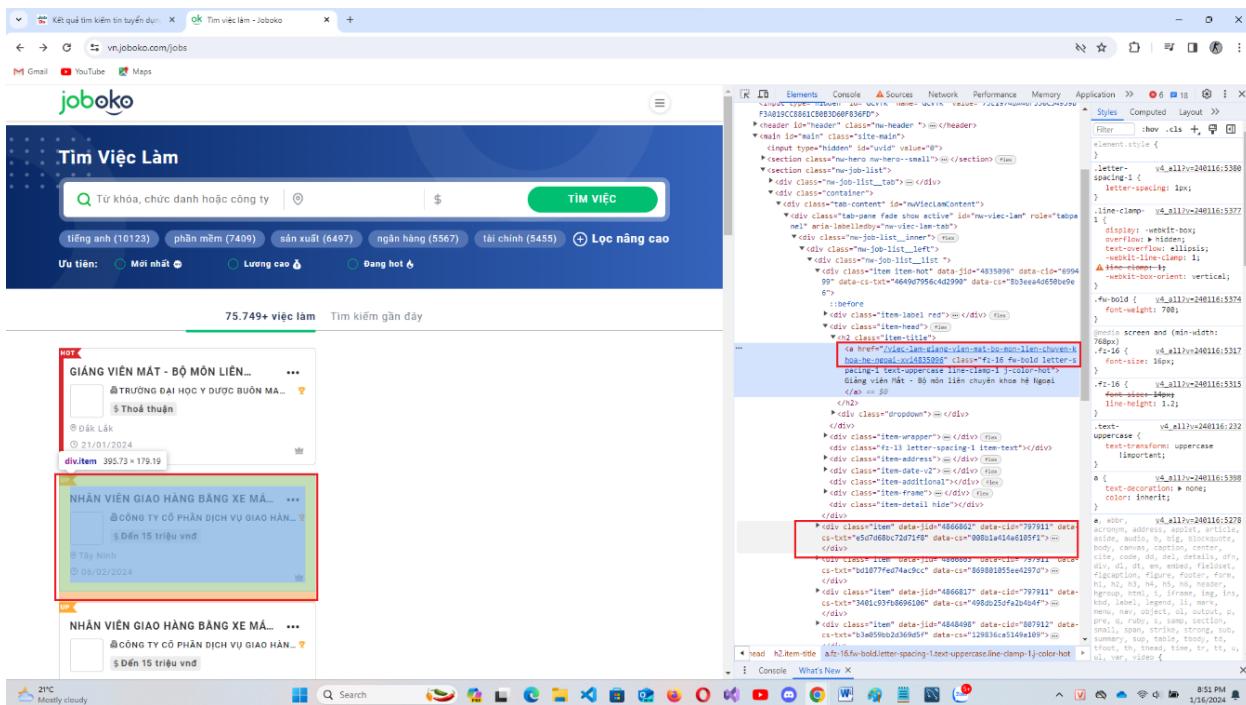
The screenshot shows the Joboko website interface. At the top, there are filters for 'Uu tiên:' (Priority), 'Mới nhất' (Newest), 'Lương cao' (High salary), and 'Đang hot' (Hot). Below this, a search bar shows '75.749+ việc làm' (75,749+ jobs) and 'Tim kiếm gần đây' (Recently searched). The main content area displays several job listings for 'NHÂN VIÊN GIAO HÀNG BẰNG XE MÁY TẠI TÂY NINH'. Each listing includes a thumbnail, title, location ('Tây Ninh'), salary ('Đến 15 triệu vnd'), and a 'Xem chi tiết công việc' (View job details) button. On the right side of the page, the browser's developer tools are open, specifically the 'Elements' tab, showing the HTML structure of one of the job listing cards. The code includes elements like `<div>`, `<audio>`, `<script>`, and `<div>` for the job description and requirements.

Hình 32. Khảo sát danh sách tin Joboko

Vô hiệu hóa Javascript và tải lại trang web, ta kết luận các công việc trên trang web trả về dưới dạng mã html.



Hình 33. Khảo sát kiểu trả về các tin Joboko



Hình 34. Khảo sát danh sách tin Joboko

Qua khảo sát ban đầu, thấy mỗi trang web danh sách việc trả cho ta 10 công việc, mỗi công việc có thể lấy được url thông qua mã html trả về.

Khảo sát cấu trúc tin của 1 công việc, ta được như sau :

The screenshot shows a job listing on the Joboko website. The job title is "GIÁNG VIÊN MÃT - BỘ MÔN LIÊN CHUYÊN KHOA HÈ NGOẠI". The employer is "TRƯỜNG ĐẠI HỌC Y DƯỢC BUÔN MA THUỘT". The location is "B'Lak". The pay rate is "Thỏa thuận". The posting date is "21/01/2024". The responsibilities listed include: a) Giảng dạy (Teach), b) Nghiên cứu khoa học (Research), c) Tham gia phục vụ cộng đồng (Participate in community services), and d) Thực hiện các công việc khác (Perform other tasks). The requirements include: a) Chứng chỉ sư phạm, b) Chứng chỉ Phương pháp dạy - học lâm sàng, and c) Sinh viên tốt nghiệp loại giỏi trở lên của Trường Đại học Y Dược Buôn Ma Thuột.

Hình 35. Khảo sát cấu trúc tin Joboko

This screenshot shows a detailed view of the same job listing from Joboko. It highlights the "YÊU CẦU CÔNG VIỆC" (Requirements) section, which lists: Là công dân Việt Nam, có sức khỏe tốt; Có khả năng làm việc độc lập, làm việc nhóm và chịu áp lực công việc; Có tinh thần trách nhiệm cao, cần thận, trung thực, nhiệt tình, siêng năng, tỉ mỉ; Có khả năng giao tiếp tốt; Tinh cam kết, ổn định và gắn bó lâu dài; Trình độ tin học: có chứng chỉ ứng dụng công nghệ thông tin cơ bản, có kỹ năng soạn thảo, sử dụng power point; Trình độ ngoại ngữ: có trình độ bậc B1 (bậc 3) hoặc IELTS 4.5 trở lên, TOEIC 450 trở lên; and УУУ. The "QUYỀN LỢI" (Benefits) section includes: Ký kết hợp đồng lao động theo quy định của pháp luật; Được tham gia đóng nộp các khoản BHXH, BHYT, BHTN... theo đúng quy định của pháp luật và được hưởng các chế độ phúc lợi khác theo quy định của Nhà trường; Được hưởng các chế độ đãi ngộ bao gồm: nghỉ phép hàng năm, nghỉ lễ, tết, hiếu, hỷ, thường lễ, tết...; Có cơ hội tham quan, học tập, nâng cao kinh nghiệm trong lĩnh vực chuyên môn; and Mức lương: theo quy chế của Nhà trường.

Hình 36. Khảo sát cấu trúc tin Joboko

h) JobsGo

Hình 37. Khảo sát danh sách tin JobsGo

Vô hiệu hóa Javascript, ta thấy các công việc trên trang web được trả về dưới dạng mã html.

Hình 38. Khảo sát kiểu trả về của danh sách tin

Qua khảo sát ta thấy, mỗi trang web danh sách công việc gồm có 50 tin, vì vậy ta sẽ lấy tổng số lượng tin và tính ra tổng số trang cuối cùng cần gửi yêu cầu tới.

Khảo sát cấu trúc tin của 1 công việc, ta được như sau :

The screenshot shows a job listing for 'Tổ Trưởng Bảo Trì (Điện, Tự Động Hóa)' at Công Ty Cổ Phần Sản Xuất Thương Mại Ký Phát. The listing includes:

- Tình trạng:** Hết hạn trong 10 ngày nữa
- Lương:** 11 - 19 triệu VNĐ
- Vị trí/Chức vụ:** Trưởng/Nhóm/Truong Phòng
- Ngày đăng tuyển:** 01/11/2023
- Yêu cầu bằng cấp:** Cao đẳng
- Yêu cầu kinh nghiệm:** 2 - 5 năm
- Địa điểm làm việc:** Số 26, Khu Công Nghiệp Vĩnh Lộc, Xã Vĩnh Lộc A, Huyện Bình Chánh, TP Hồ Chí Minh, Việt Nam
- Nghề nghiệp:** Kỹ Thuật Công Nghiệp, Bảo Trì - Sửa Chữa, Tự Động Hóa, Vận Hành/Sản Xuất, Cơ Điện, Điện/Điện tử
- Mô tả công việc:**
 - Thường xuyên kiểm tra, bảo trì bảo dưỡng các máy móc thiết bị ngành bảo bì mảng nhựa như máy mẻ mảng CPP, máy chia cuộn OPP, máy lăn lạnh, máy stretch film và máy dùn
 - Hỗ trợ trưởng phòng bảo trì lập kế hoạch bảo trì bảo dưỡng
 - Phối hợp với bộ phận sản xuất để thực hiện thẩm định lắp đặt, vận hành, chất lượng máy móc nhà xưởng và đánh giá khả năng hoạt động của thiết bị máy móc
 - Kiểm tra chất chất lượng bề mặt máy móc, các thử túi quy trình để sẵn sàng các đoàn thanh tra
 - Đánh giá chất lượng nhà cung cấp thiết bị nhà thầu hàng năm cùng với các bộ phận liên quan
 - Quản lý và cập nhật thiết bị, linh kiện dự phòng cho các máy móc trong nhà máy
 - Cập nhật các quy trình thẩm định và tái thẩm định
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - Cải tiến hiệu năng máy móc để đạt năng suất cao nhất
 - Nhận thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa nhanh chóng, đảm bảo tiến độ công việc chung cho toàn nhà máy - giám sát chất lượng quá trình bảo trì thiết bị nhằm thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa
 - Hỗ trợ trưởng phòng nghiên cứu cải tiến máy móc phù hợp với nhu cầu sản phẩm và đảm bảo sự an toàn trong vận hành của công nhân
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - Cải tiến hiệu năng máy móc để đạt năng suất cao nhất
 - Nhận thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa nhanh chóng, đảm bảo tiến độ công việc chung cho toàn nhà máy - giám sát chất lượng quá trình bảo trì thiết bị nhằm thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa
 - Hỗ trợ trưởng phòng nghiên cứu cải tiến máy móc phù hợp với nhu cầu sản phẩm và đảm bảo sự an toàn trong vận hành của công nhân
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - ***Chi tiết công việc trao đổi cụ thể trong quá trình phỏng vấn
- Yêu cầu công việc:**
 - Tốt nghiệp 2 năm kinh nghiệm ở vị trí tương đương tại các nhà máy sản xuất bảo bì nhựa
 - Thời gian làm việc theo ca 12 tiếng
 - Ưu tiên có kiến thức về hệ thống ISO 9001, 14001, 5S, TPM, ISO
 - Sức khỏe tốt, trung thực, chăm chỉ và có trách nhiệm trong công việc
 - Tốt nghiệp đại học hoặc cao đẳng chuyên ngành
 - Điện tử, Tự động hóa, Cơ Điện tử và các ngành tương tự
- Quyền lợi được hưởng:**
 - Lương từ 11 - 19 triệu (thỏa thuận theo năng lực)
 - Thu nhập ổn định lâu dài
 - Tham gia đầy đủ BHYT, BHTN, bảo hiểm sức khỏe

Hình 39. Khảo sát cấu trúc tin JobsGo

The screenshot shows a job listing for 'Tổ Trưởng Bảo Trì (Điện, Tự Động Hóa)' at Công Ty Cổ Phần Sản Xuất Thương Mại Ký Phát. The listing includes:

- Mô tả công việc:**
 - Thường xuyên kiểm tra, bảo trì bảo dưỡng các máy móc thiết bị ngành bảo bì mảng nhựa như máy mẻ mảng CPP, máy chia cuộn OPP, máy lăn lạnh, máy stretch film và máy dùn
 - Hỗ trợ trưởng phòng bảo trì lập kế hoạch bảo trì bảo dưỡng
 - Phối hợp với bộ phận sản xuất để thực hiện thẩm định lắp đặt, vận hành, chất lượng máy móc nhà xưởng và đánh giá khả năng hoạt động của thiết bị máy móc
 - Kiểm tra chất chất lượng bề mặt máy móc, các thử túi quy trình để sẵn sàng các đoàn thanh tra
 - Đánh giá chất lượng nhà cung cấp thiết bị nhà thầu hàng năm cùng với các bộ phận liên quan
 - Quản lý và cập nhật thiết bị, linh kiện dự phòng cho các máy móc trong nhà máy
 - Cập nhật các quy trình thẩm định và tái thẩm định
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - Cải tiến hiệu năng máy móc để đạt năng suất cao nhất
 - Nhận thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa nhanh chóng, đảm bảo tiến độ công việc chung cho toàn nhà máy - giám sát chất lượng quá trình bảo trì thiết bị nhằm thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa
 - Hỗ trợ trưởng phòng nghiên cứu cải tiến máy móc phù hợp với nhu cầu sản phẩm và đảm bảo sự an toàn trong vận hành của công nhân
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - Cải tiến hiệu năng máy móc để đạt năng suất cao nhất
 - Nhận thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa nhanh chóng, đảm bảo tiến độ công việc chung cho toàn nhà máy - giám sát chất lượng quá trình bảo trì thiết bị nhằm thông tin hư hỏng thiết bị từ các bộ phận - phản công kỹ thuật viên đảm nhận việc kiểm tra, sửa chữa
 - Hỗ trợ trưởng phòng nghiên cứu cải tiến máy móc phù hợp với nhu cầu sản phẩm và đảm bảo sự an toàn trong vận hành của công nhân
 - Báo cáo tình hình máy móc hàng tuần và hàng tháng thiết bị cho quản lý
 - ***Chi tiết công việc trao đổi cụ thể trong quá trình phỏng vấn
- Yêu cầu công việc:**
 - Tốt nghiệp 2 năm kinh nghiệm ở vị trí tương đương tại các nhà máy sản xuất bảo bì nhựa
 - Thời gian làm việc theo ca 12 tiếng
 - Ưu tiên có kiến thức về hệ thống ISO 9001, 14001, 5S, TPM, ISO
 - Sức khỏe tốt, trung thực, chăm chỉ và có trách nhiệm trong công việc
 - Tốt nghiệp đại học hoặc cao đẳng chuyên ngành
 - Điện tử, Tự động hóa, Cơ Điện tử và các ngành tương tự
- Quyền lợi được hưởng:**
 - Lương từ 11 - 19 triệu (thỏa thuận theo năng lực)
 - Thu nhập ổn định lâu dài
 - Tham gia đầy đủ BHYT, BHTN, bảo hiểm sức khỏe

Hình 40. Khảo sát cấu trúc tin JobsGo

i) StudentJob

The screenshot shows the StudentJob website interface. At the top, there's a navigation bar with links for 'TÌM VIỆC', 'NHÀ TRỌ SINH VIÊN', 'THỰC TẬP', 'CÔNG TY', 'TUYỂN DỤNG NHANH', and 'CẨM NANG NGHỀ NGHIỆP'. On the right side of the header, there are 'Đăng nhập' and 'Đăng ký' buttons, along with a 'NHÀ TUYỂN DỤNG' section. Below the header, a search bar displays the query 'VIỆC LÀM' and shows a result count of 'Có 179222 việc làm được tìm thấy'. The main content area lists several job posts with details like company name, location, salary, and a brief description. To the right, there are two columns: 'Việc làm 2024' and 'Việc làm tuyển gấp', each containing three job listings. At the bottom right, there's a 'Nhà trọ sinh viên' section.

Hình 41. Khảo sát danh sách tin StudentJob

Vô hiệu hóa Javascript và tải lại trang, ta thấy các công việc trên trang web được trả về dưới dạng mã html.

This screenshot shows the same StudentJob website as above, but with JavaScript disabled. The browser's developer tools are open, specifically the 'Elements' tab, which displays the raw HTML code for the page. The rendered content from the screenshot in Figure 41 is now replaced by this raw HTML. The page structure remains the same, with sections for 'VIỆC LÀM', 'Việc làm 2024', 'Việc làm tuyển gấp', and 'Nhà trọ sinh viên', but all the text and images are now represented as their corresponding HTML elements.

Hình 42. Khảo sát kiểu trả về các tin StudentJob

Qua khảo sát cũng thấy mỗi trang tin tuyển dụng có 18 tin, vì vậy dựa vào tổng số lượng tin tuyển dụng, ta có thể tính được trang tuyển dụng cuối cùng cần gửi yêu cầu truy cập.

Khảo sát cấu trúc của 1 tin tuyển dụng, ta được như sau :

TUYỂN DỤNG Đầu bếp, Bồi bàn đi Đức làm việc hợp pháp

CÔNG TY TNHH ĐẦU TƯ GIÁO DỤC QUỐC TẾ ÂU CHÂU

Địa chỉ làm việc: Germany

Hết hạn ngày: 20/02/2024 ID: job367225

MÔ TẢ CÔNG VIỆC

Tuyển dụng Đầu bếp, Bồi bàn đi Đức làm việc hợp pháp

Lương cầm tay:

- Đầu bếp ít kinh nghiệm 1700-2200€ (40-55 triệu VND)
- Đầu bếp kinh nghiệm trên 2 năm: 2200-3000€ (55-80 triệu VND)
- Bao ăn ở, đã trú thuế, bảo hiểm

YÊU CẦU CÔNG VIỆC

- Có bằng ô TC/CD/DH ở Việt nam
- BHXH có hoặc không có đều được.
- Thời gian xuất cảnh: 2-3 tháng.

HÌNH THỨC

Toàn thời gian cố định

ĐỊA CHỈ LÀM VIỆC: Germany

NGÀY ĐĂNG TUYỂN: 13/01/2024

VỊ TRÍ: Nhân viên

Hình 43. Khảo sát cấu trúc tin StudentJob

YÊU CẦU CÔNG VIỆC

- Có bằng ô TC/CD/DH ở Việt nam
- BHXH có hoặc không có đều được.
- Thời gian xuất cảnh: 2-3 tháng.

HÌNH THỨC

Toàn thời gian cố định

MỨC LƯƠNG

Thỏa thuận

ĐỊA CHỈ LÀM VIỆC: Germany

NGÀY ĐĂNG TUYỂN: 13/01/2024

VỊ TRÍ: Nhân viên

NGÀNH NGHỀ: Khách sạn / Nhà hàng / Du lịch

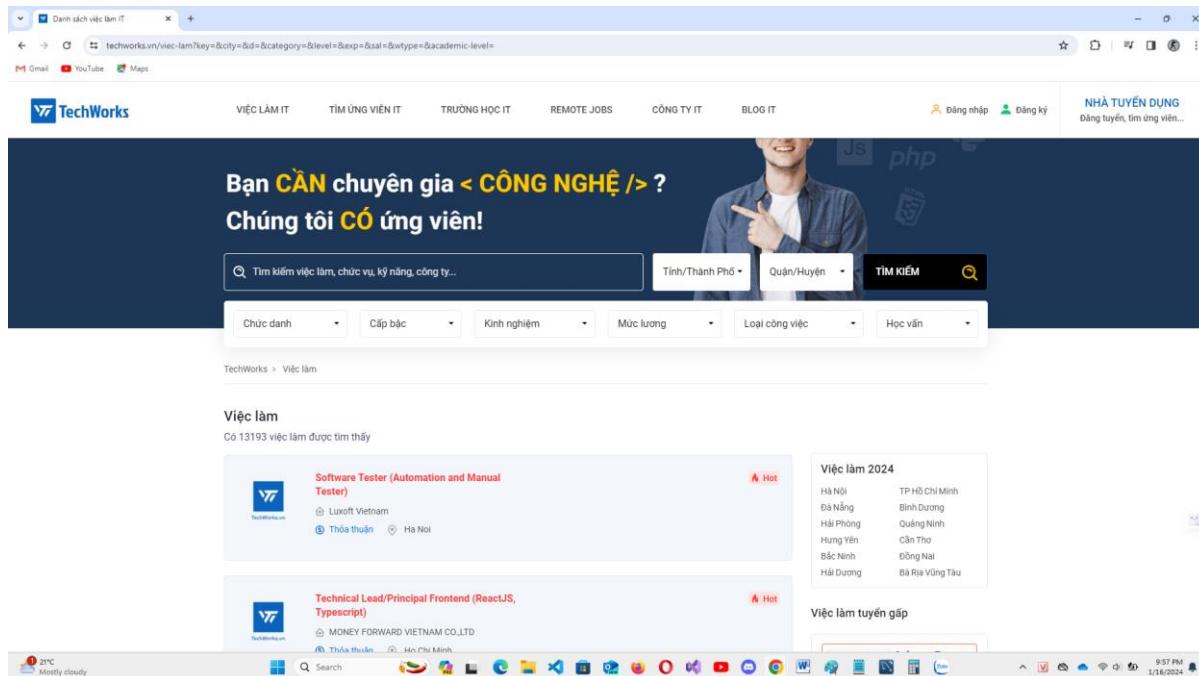
LOẠI CÔNG VIỆC: Toàn thời gian cố định

NHÀ TUYỂN DỤNG: Công ty TNHH Đầu tư Giáo dục quốc tế Âu Châu

LIÊN HỆ: Đào Kim Hoàng
Email: duhocaucachau.hanoi@gmail.com
ĐT: 0982799066

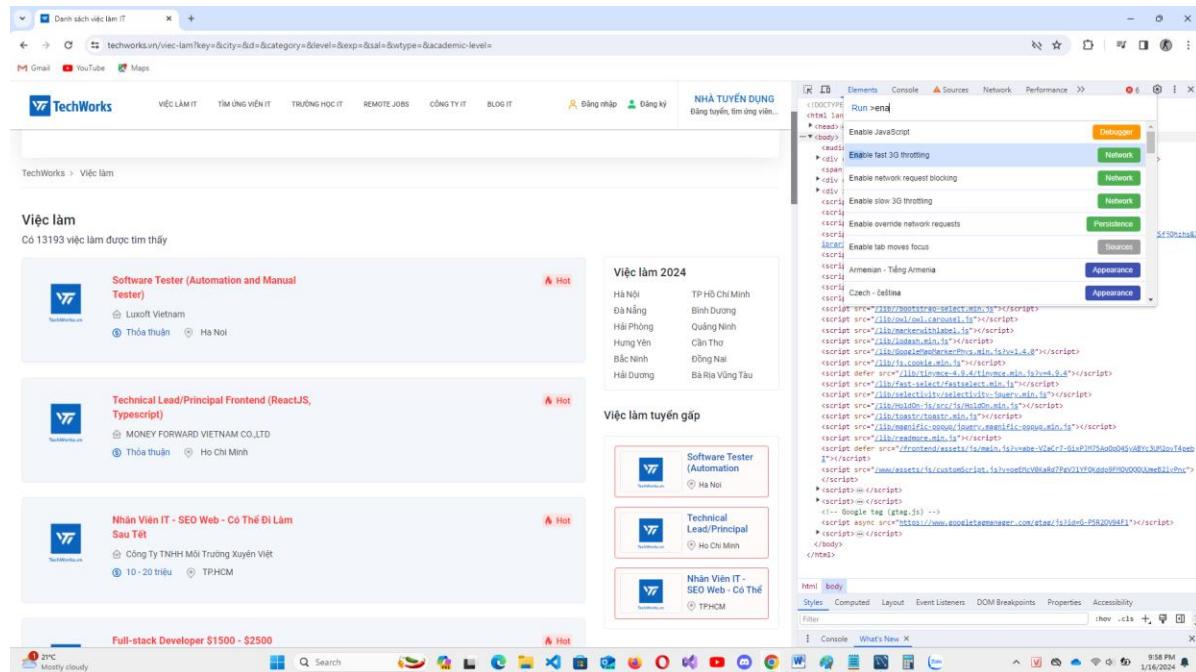
Hình 44. Khảo sát cấu trúc tin StudentJob

j) TechWorks



Hình 45. Khảo sát danh sách tin TechWorks

Vô hiệu hóa Javascript và tải lại trang web, ta thấy các công việc trên trang web được trả về dưới dạng mã html.



Hình 46. Khảo sát kiểu trả về các tin TechWorks

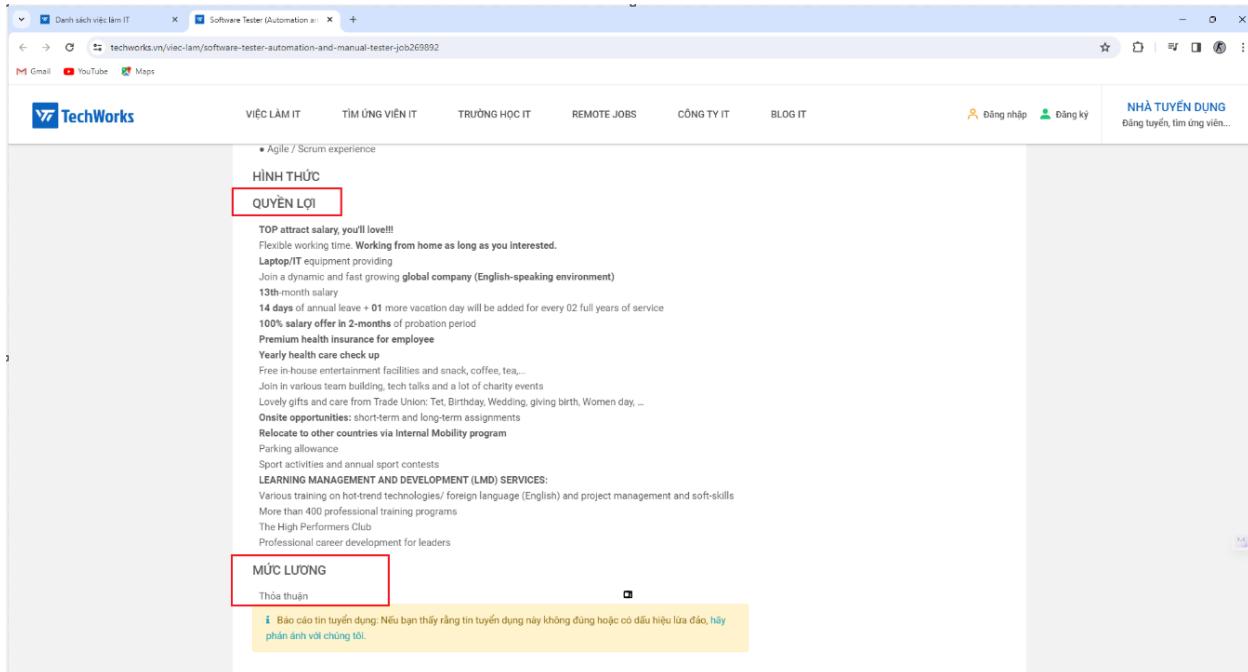
Khảo sát cấu trúc của một tin tuyển dụng, ta được kết quả sau :

The screenshot shows a job listing on TechWorks. The job title is "Software Tester (Automation and Manual Tester)" at LUXOFF VIETNAM in Hanoi. The listing includes a map of Hanoi showing the job location. Key responsibilities listed include Test Planning, Test Case Design, Test Execution, Automation Testing, Regression Testing, Performance Testing, Security Testing, Usability Testing, and Test Documentation. The application deadline is 13/04/2024. The job type is Software Engineer.

Hình 47. Khảo sát cấu trúc tin TechWorks

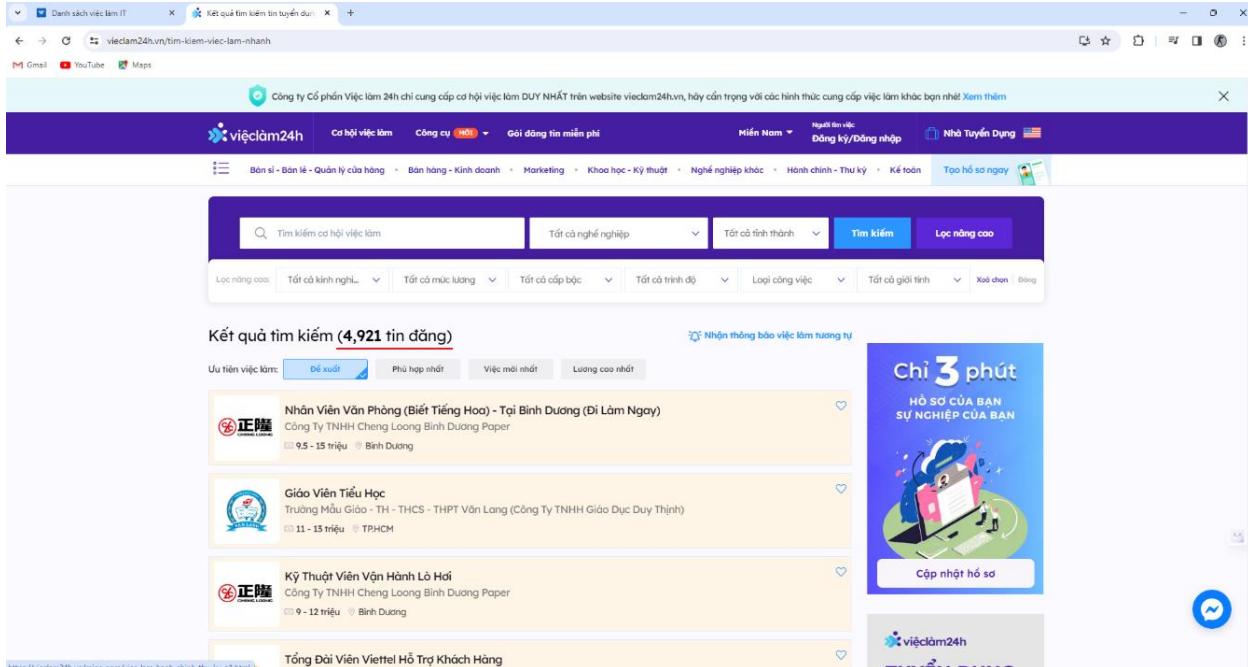
The screenshot shows a job listing on TechWorks. The job title is "Software Tester (Automation and Manual Tester)" at LUXOFF VIETNAM in Hanoi. The listing includes a map of Hanoi showing the job location. Key responsibilities listed include Test Planning, Test Case Design, Test Execution, Automation Testing, Regression Testing, Performance Testing, Security Testing, Usability Testing, and Test Documentation. The application deadline is 13/01/2024. The job type is Software Engineer. On the right side of the listing, several fields are highlighted with red boxes: Địa chỉ làm việc (Address), Ngày đăng tuyển (Posting date), Vị trí (Location), Ngành nghề (Industry), Loại công việc (Job type), and Nhà tuyển dụng (Recruiter).

Hình 48. Khảo sát cấu trúc tin TechWorks



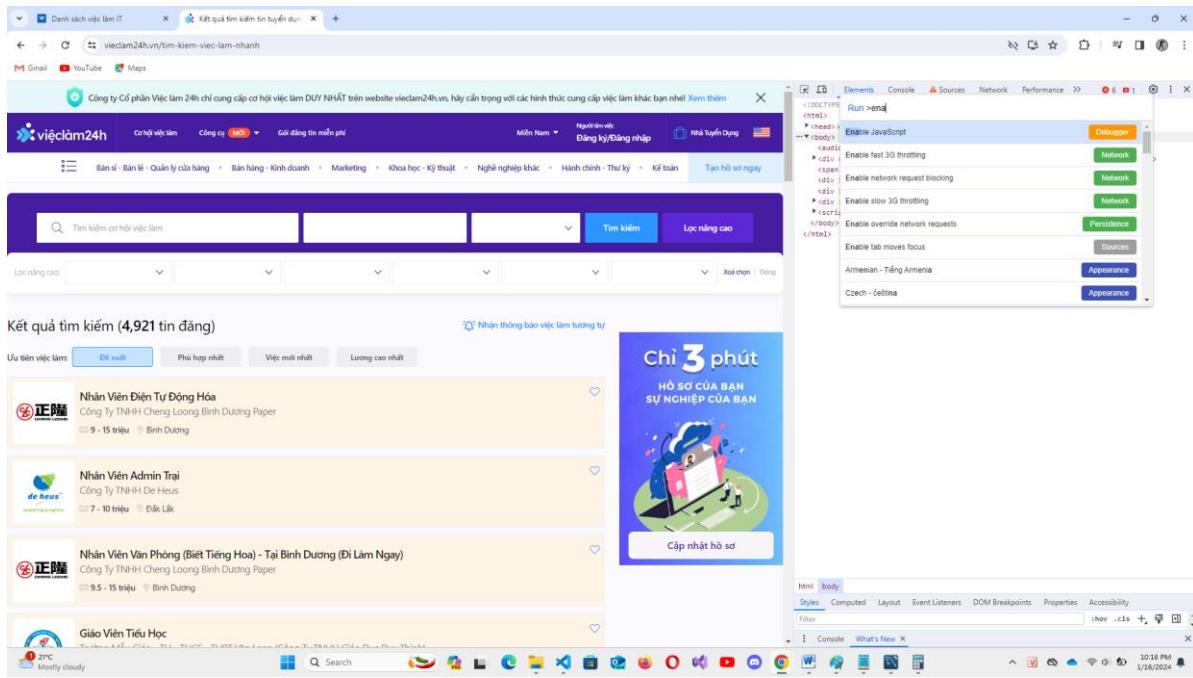
Hình 49. Khảo sát cấu trúc tin TechWorks

k) ViecLam24h



Hình 50. Khảo sát danh sách tin ViecLam24h

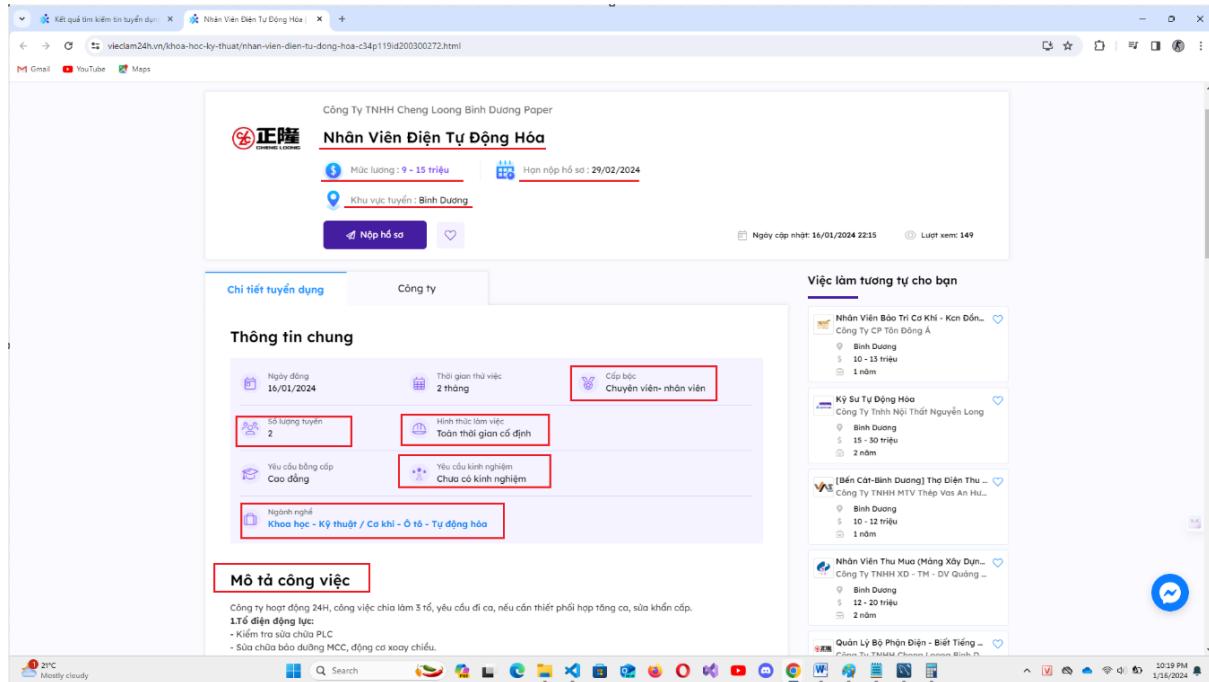
Vô hiệu hóa Javascript và tải lại trang web, ta thấy các công việc đều được trả về dưới dạng mã html.



Hình 51. Khảo sát kiểu trả về các tin ViecLam24h

Qua khảo sát ta thấy một trang tin sẽ gồm có 30 tin tuyển dụng. Như vậy, từ số lượng tin, ta sẽ tính được trang tin tuyển dụng cuối cùng mà có thể truy cập.

Khảo sát cấu trúc của một tin tuyển dụng, ta được như sau :



Hình 52. Khảo sát cấu trúc tin ViecLam24h

The screenshot shows a search result page for 'Nhân Viên Điện Tự Động Hóa' (Electrical Engineer) on ViecLam24h. The results are filtered by location (Bình Dương), salary range (9-10 million VND), and experience (1 year). The first listing is for 'Thợ Điện Cơ Khi' at Công Ty TNHH Pháp Quốc, with a salary of 9-10 million VND and 1 year of experience. Other listings include 'Nhân Viên Kỹ Thuật Bảo Trì' at Công Ty Cổ Phần Thép Nam Kim, 'Nhân Viên Kỹ Thuật Cơ Khí / Điện...' at Công Ty Trách Nhiệm Hữu Hạn Tư V..., and 'Nhân Viên Kỹ Thuật (Đầu Nối Điện,...)' at Công Ty Cổ Phần Công Nghệ Chế Tg... The page also features sections for job requirements and benefits.

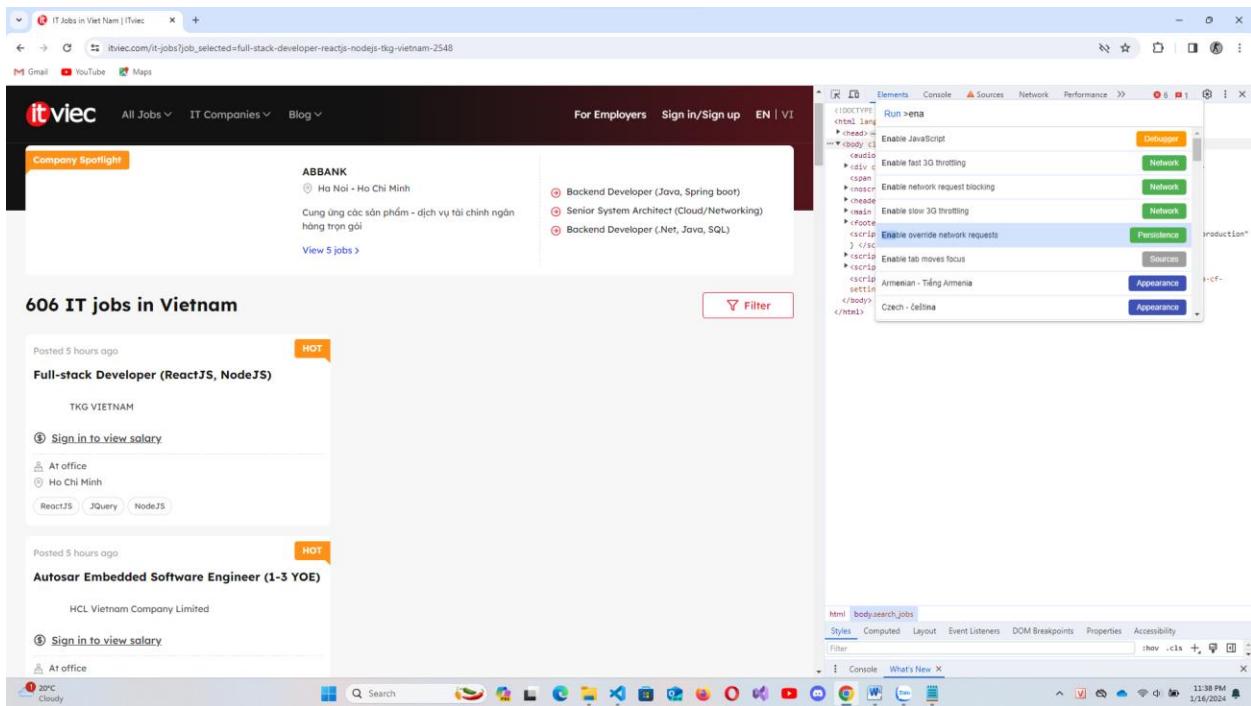
Hình 53. Khảo sát cấu trúc tin ViecLam24h

I) ITViec

The screenshot shows the ITViec website interface. At the top, there are navigation links for 'All Jobs', 'IT Companies', and 'Blog'. The main search bar allows users to enter keywords, skills, job titles, or company names. Below the search bar, a 'Company Spotlight' section features an image of the Samsung Electronics HCMC CE Complex. A search result for 'Samsung Electronics HCMC CE Complex' shows three job openings: 'Embedded Software Engineer', 'C++ Developer', and 'Embedded Project Management'. The main content area displays '606 IT jobs in Vietnam'. Two specific job listings are highlighted: 'Full-stack Developer (ReactJS, NodeJS)' posted 5 hours ago by TKG VIETNAM, and 'Autosar Embedded Software Engineer (1-3 YOE)' posted 5 hours ago by HCL Vietnam Company Limited. Both listings include 'HOT' tags and 'Sign in to view salary' buttons. The bottom of the screen shows a taskbar with various application icons and system status indicators.

Hình 54. Khảo sát danh sách tin ITViec

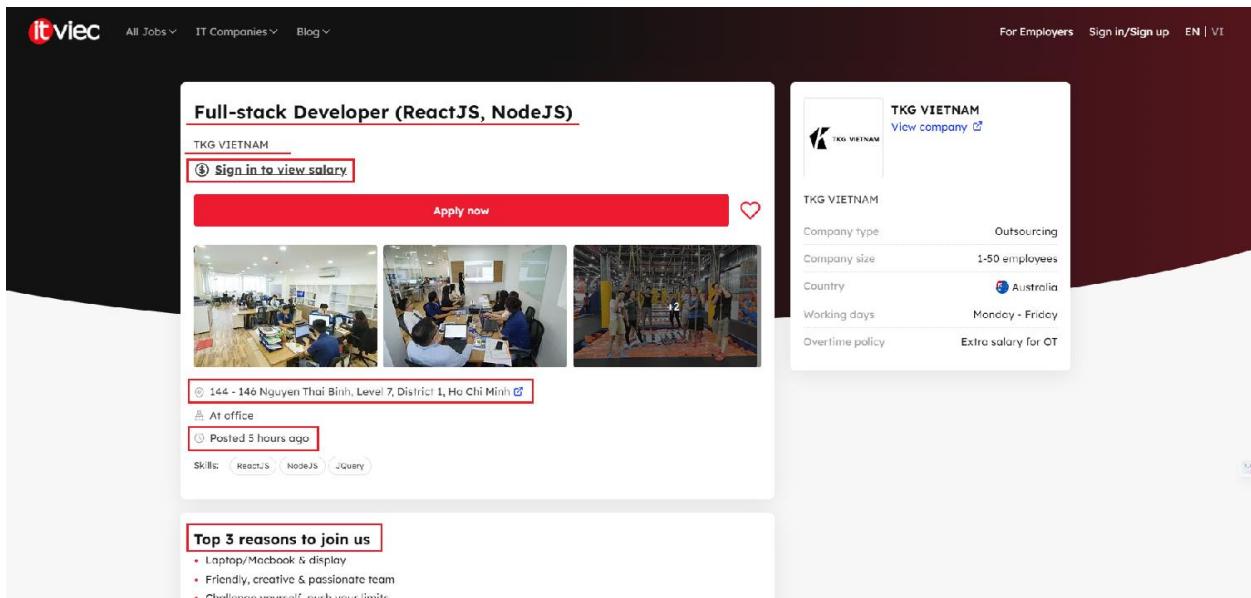
Vô hiệu hóa Javascript và tải lại trang, ta thấy các công việc trên trang được trả về dưới dạng mã html.



Hình 55. Khảo sát kiểu trả về của các tin ITviec

Một trang danh sách tin tuyển dụng như này sẽ có 20 tin, vì vậy ta có thể dùng số lượng tin để tính ra được trang cuối cùng cần duyệt qua.

Khảo sát một tin tuyển dụng, ta được kết quả như sau :



Hình 56. Khảo sát cấu trúc tin ITviec

The screenshot shows a job listing on the ITViec website. The job title is "Full-stack Developer (ReactJS, NodeJS)". The company is TKG VIETNAM. The listing includes a "Sign in to view salary" button, an "Apply now" button, and a brief description: "Challenge yourself, push your limits". A "Job description" section details the role involves learning and building an IT career, working with international customers, and maintaining world-class solutions. A "Your skills and experience" section lists required skills: 3+ years of experience, ReactJS, Redux, jQuery, EJS, ES6, NodeJS, and MySQL, RabbitMQ, Redis. To the right, there's a sidebar for TKG VIETNAM showing company details: Outsourcing type, 1-50 employees, Australia location, Monday - Friday working days, and Extra salary for OT overtime policy.

Hình 57. Khảo sát cấu trúc tin ITViec

This screenshot shows a similar job listing for a Full-stack Developer (ReactJS, NodeJS) at TKG VIETNAM. It includes a "Sign in to view salary" button and an "Apply now" button. The "Your skills and experience" section has been updated to include CSS Pre-processors (SASS, LESS, Stylus), Unit testing frameworks and tools, Soft skills (Conversational English, Ability to work independently and within a team environment, Attention to detail and strong problem-solving skills), Bonus skills (React Native, Technical SEO, Google analytics tag manager, Experience with Mapbox and Shopify APIs), and Why you'll love working here (Equipment: MacBook + display, Travel: optional opportunity to travel and relocate overseas, Team events: annual company trips, regular team lunch and dinner, Culture: fun, dynamic, collaborative and creative working environment with a focus on work-life balance).

Hình 58. Khảo sát cấu trúc tin ITViec

m) 123Job (Chỉ lấy IT)

The screenshot shows the 123Job website interface. The search bar at the top is set to 'IT phần mềm'. The main content area displays a list of 13182 job postings. On the left, there's a sidebar with a 'Tiết kiệm 5% ITSM service desk' offer and a link to 'lp.agileops.vn'. On the right, there's a sidebar with a Google sign-in message and a 'Tin nhắn' (Message) button.

Hình 59. Khảo sát danh sách tin 123Job

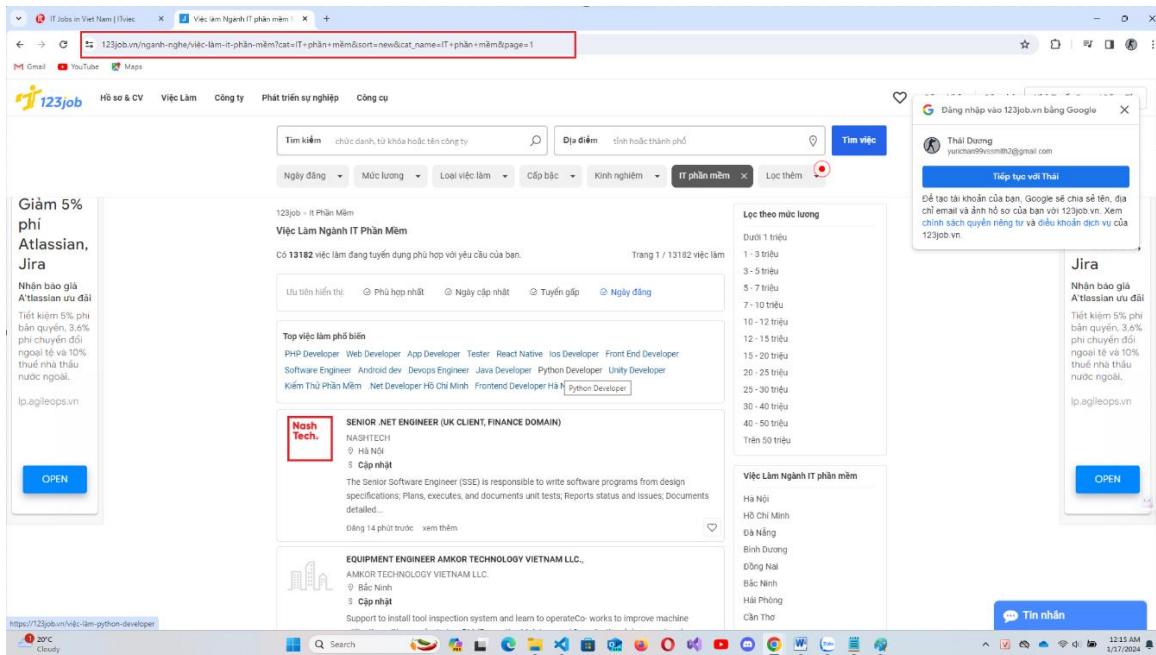
Vô hiệu hóa Javascript và tải lại trang, ta nhận thấy các công việc đều được trả về dưới dạng mã html.

The screenshot shows the same 123Job website after disabling JavaScript. The page is now displayed as plain HTML code, showing the underlying structure of the job listings. The job details, such as company names, locations, and descriptions, are present but lack the visual styling and interactivity provided by JavaScript.

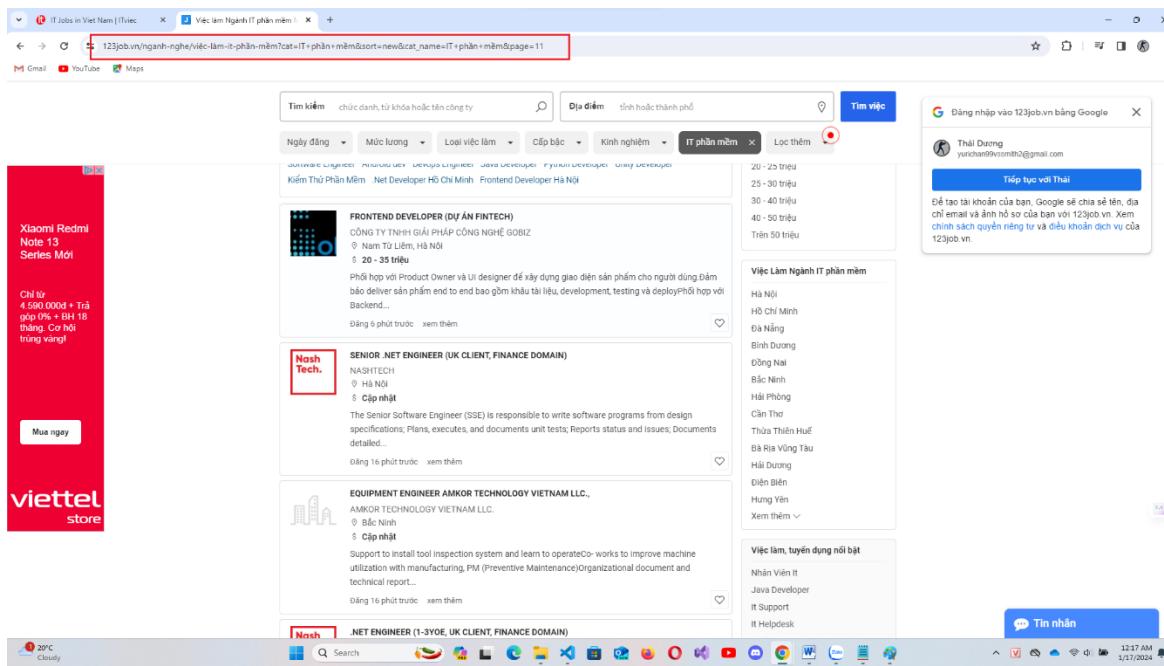
Hình 60. Khảo sát kiểu trả về các tin 123Job

Ở trang web này có một ván đè nhỏ, đó là tuy hiển thị đang có 13182 việc làm, nhưng thực tế, mỗi trang web chỉ có 30 việc và chỉ có 10 trang đầu là khả dụng, khi ấn vào trang thứ 11, trang web sẽ tự động quay về trang 1.

Dưới đây là hình ảnh so sánh hai trang 1 và 11.



Hình 61. Trang 1



Hình 62. Trang 11

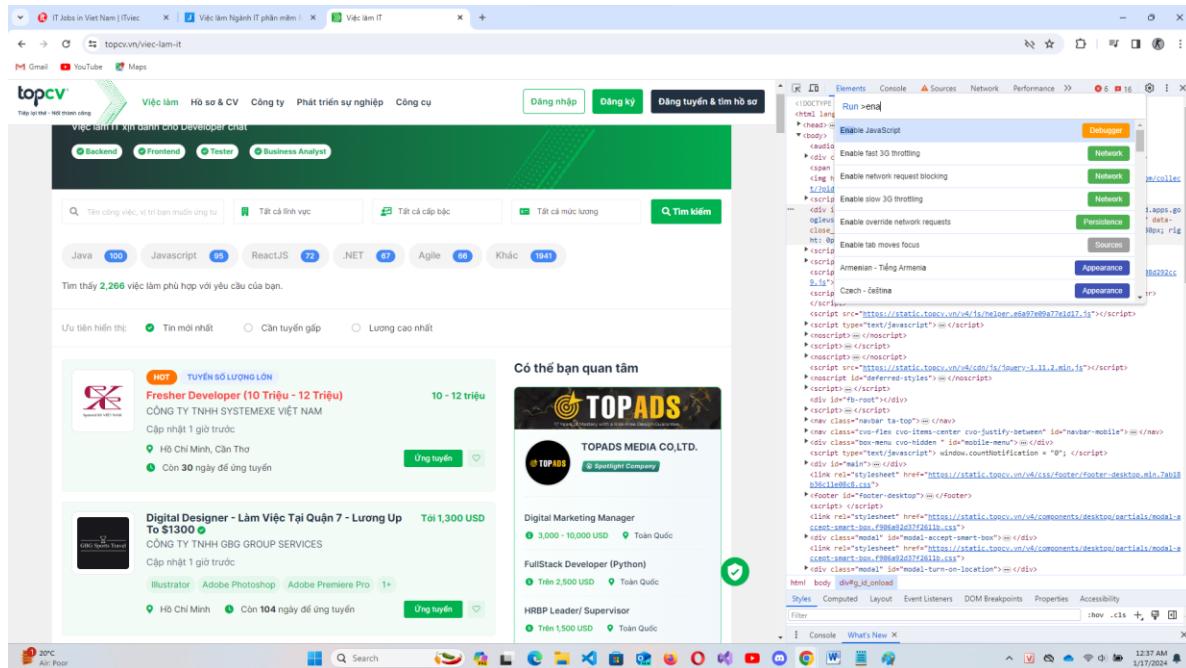
Khảo sát một tin tuyển dụng, ta thu được kết quả sau :

Hình 63. Khảo sát cấu trúc tin 123Job

n) TopCV (Chỉ lấy IT)

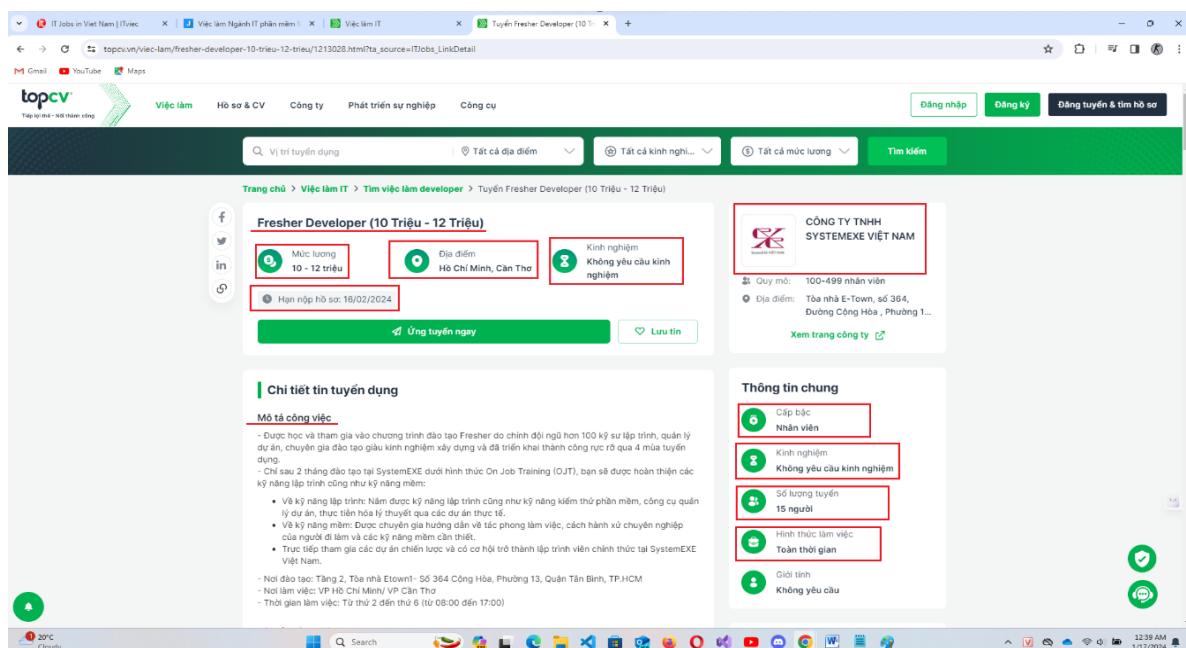
Hình 64. Khảo sát danh sách tin TopCV

Vô hiệu hóa Javascript và tải lại trang web, ta thấy các công việc cần thu thập đều được trả về dưới dạng mã html. Khảo sát cũng thấy mỗi trang danh sách tin tuyển dụng có 50 tin, vì vậy ta có thể tính được trang cuối cùng dựa vào tổng số lượng tin.

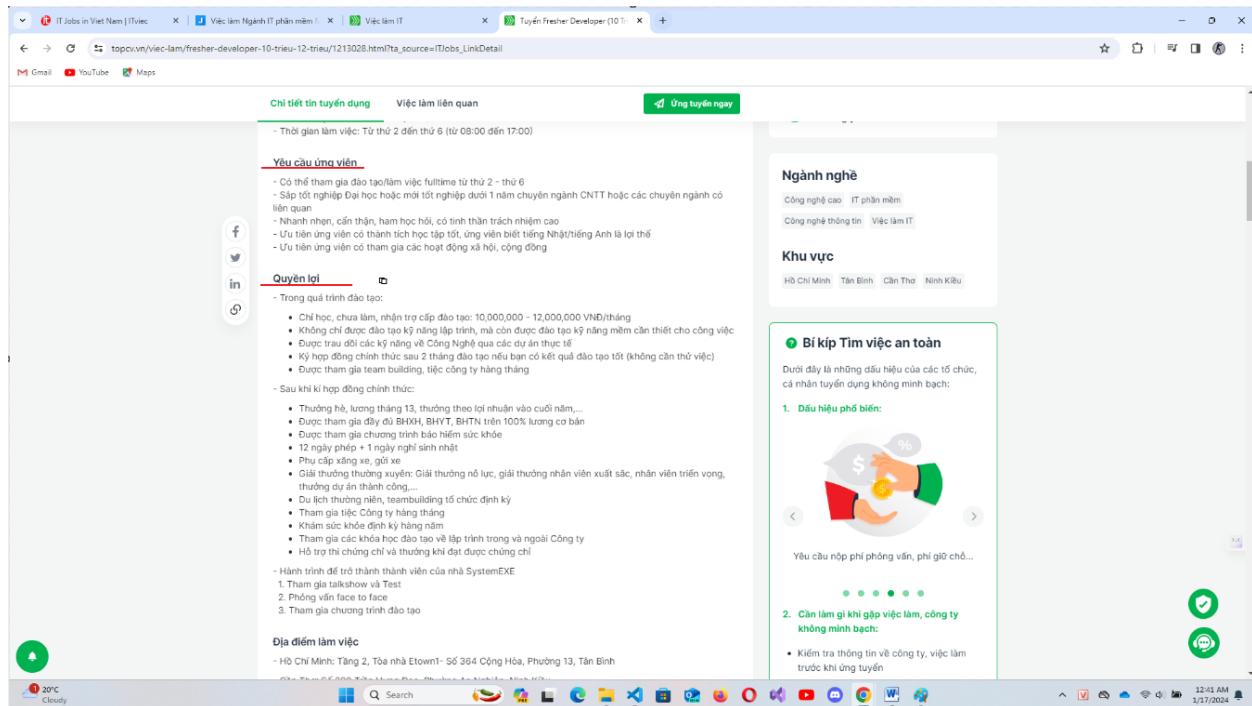


Hình 65. Khảo sát kiểu trả về của các tin

Khảo sát cấu trúc của 1 tin tuyển dụng, ta được kết quả sau :

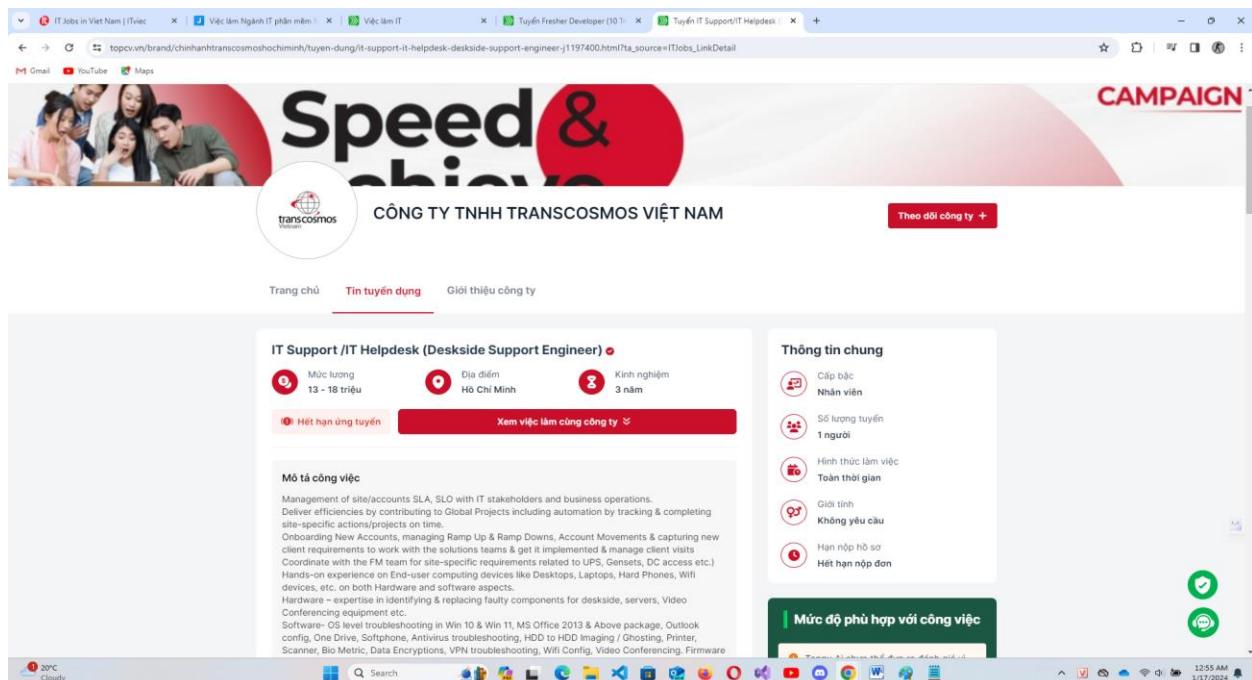


Hình 66. Khảo sát cấu trúc tin

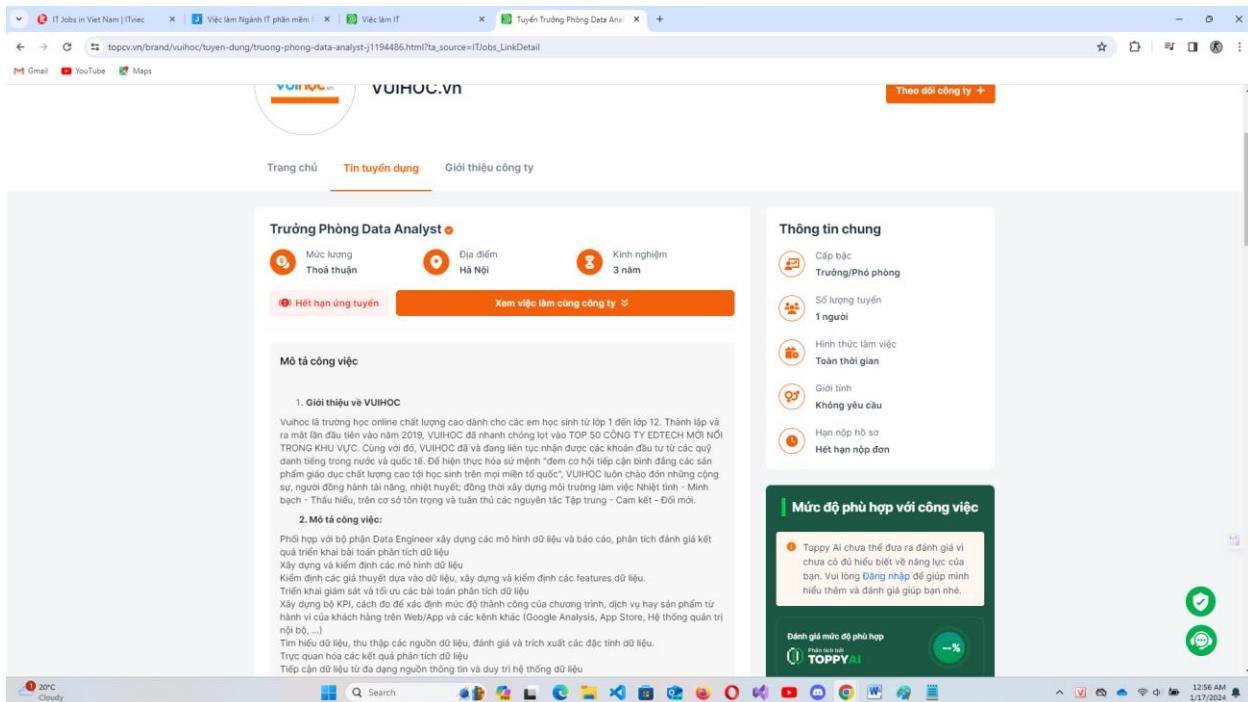


Hình 67. Khảo sát cấu trúc tin TopCV

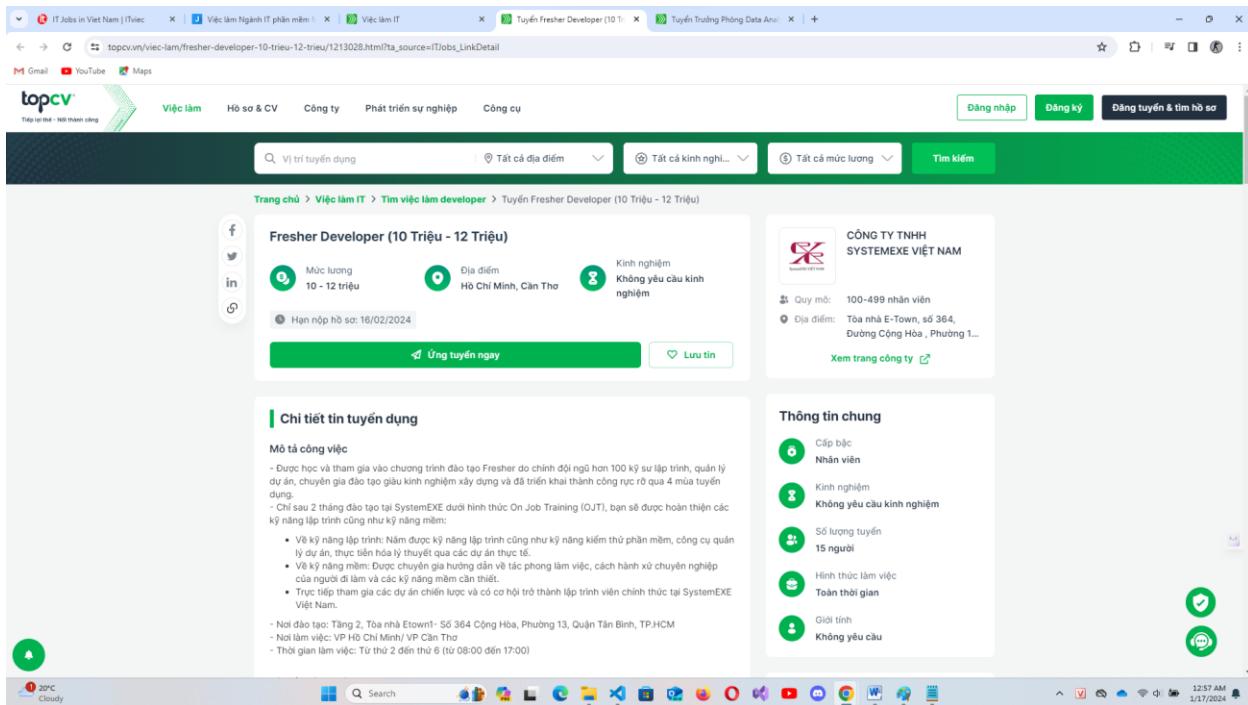
Có một số tin tuyển dụng có cấu trúc khác với đại đa số các cấu trúc mặc định ở trên TopCV:



Hình 68. Một số tin có cấu trúc khác mặc định



Hình 69. Một số tin có cấu trúc khác mặc định

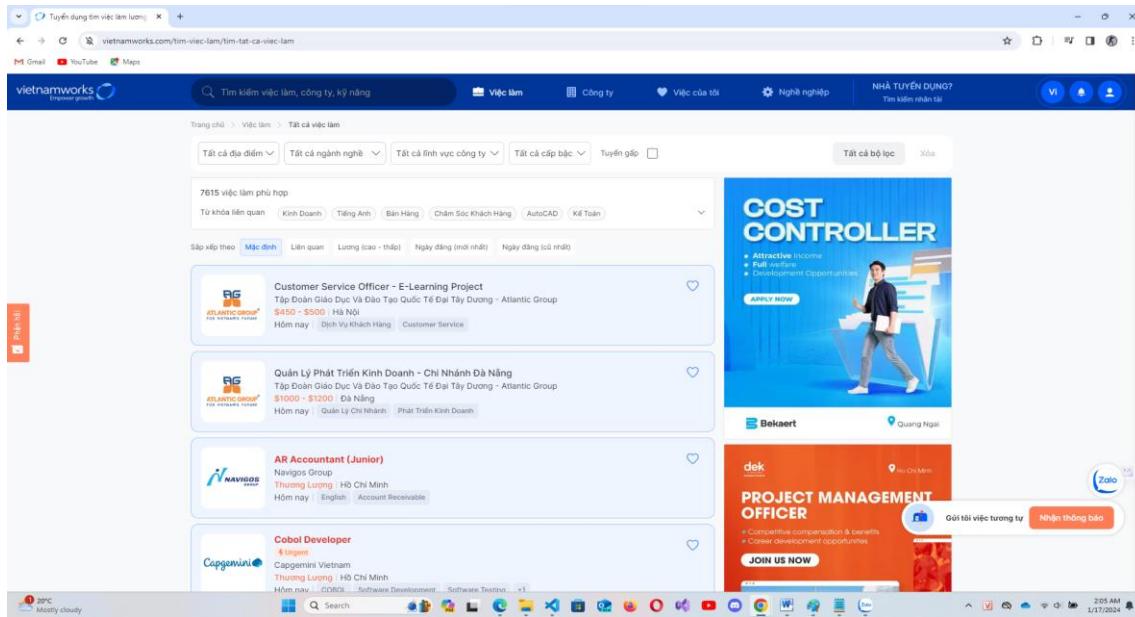


Hình 70. Cấu trúc tin mặc định

Ba hình này là của ba tin tuyển dụng khác nhau, trong đó hình thứ 3 với tông màu xanh lá là cấu trúc mặc định của TopCV, và ba cấu trúc tin này có 3 loại mã html trả về khác nhau. Do số lượng tin có cấu trúc khác mặc định là rất ít và url của các tin loại này đều có dạng

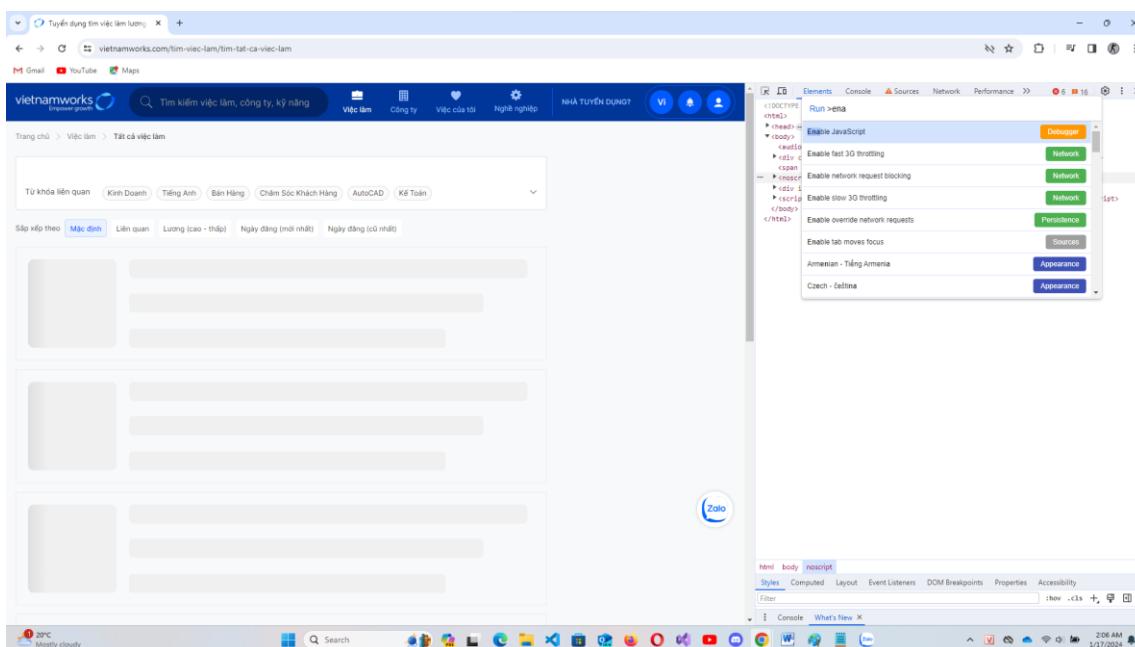
<https://www.topcv.vn/brand/.....> Nên ta sẽ mặc định bỏ qua các url dạng này và chỉ truy cập các url có cấu trúc mặc định của TopCV.

o) VietNamWork



Hình 71. Khảo sát danh sách tin VietNamWork

Vô hiệu hóa Javascript và tải lại trang web, ta thấy các công việc của trang web không tải ra, vì vậy các công việc trên trang này sẽ không được trả về dưới dạng mã html.



Hình 72. Khảo sát kiểu trả về các tin VietNamWork

Tìm trong các API trả về từ server, ta nhận được kết quả sau :

```

{
    "meta": {
        "code": 200,
        "message": "Success"
    },
    "data": [
        {
            "id": 17204780,
            "title": "Customer Service Officer - E-Learning Project",
            "url": "https://vietnamworks.com/jobs/17204780",
            "company": "ATLANTIC GROUP VIETNAM CO., LTD",
            "location": "Hà Nội",
            "description": "We are looking for a Customer Service Officer to join our E-Learning Project team. The ideal candidate will have excellent communication skills, ability to work independently, and a passion for customer service. Experience in customer service or sales is preferred but not required. If you are interested in this role, please apply now!",
            "requirements": "Bachelor's degree in English or related field; Excellent communication skills; Strong problem-solving abilities; Ability to work independently and as part of a team; Good time management skills; Experience in customer service or sales is preferred but not required.",
            "benefits": "Competitive salary and performance-based bonuses; Comprehensive benefits package including health insurance, retirement savings plan, and paid time off; Opportunities for professional development and career growth; A supportive and inclusive work environment.", ...
        }
    ]
}

```

Hình 73. Khảo sát API trả về từ Server (VietNamWork)

Tất cả 50 công việc trong trang đều được trả về dưới dạng file json.

```

{
    "meta": {
        "code": 200,
        "message": "Success"
    },
    "data": [
        {
            "id": 17204780,
            "title": "Customer Service Officer - E-Learning Project",
            "url": "https://vietnamworks.com/jobs/17204780",
            "company": "ATLANTIC GROUP VIETNAM CO., LTD",
            "location": "Hà Nội",
            "description": "We are looking for a Customer Service Officer to join our E-Learning Project team. The ideal candidate will have excellent communication skills, ability to work independently, and a passion for customer service. Experience in customer service or sales is preferred but not required. If you are interested in this role, please apply now!",
            "requirements": "Bachelor's degree in English or related field; Excellent communication skills; Strong problem-solving abilities; Ability to work independently and as part of a team; Good time management skills; Experience in customer service or sales is preferred but not required.",
            "benefits": "Competitive salary and performance-based bonuses; Comprehensive benefits package including health insurance, retirement savings plan, and paid time off; Opportunities for professional development and career growth; A supportive and inclusive work environment.", ...
        }
    ]
}

```

Hình 74. Khảo sát file json trả về từ Server (VietNamWork)

Ngoài ra trong file json trả về còn mang thông tin về số lượng công việc hiện có và số lượng công việc trả về cho 1 trang.

Khảo sát phương thức mà trang web gửi yêu cầu cho server ta được như sau :

Hình 75. Khảo sát request header API

Hình 76. Khảo sát request body API

3. Công cụ thu thập dữ liệu

Scrapy: là một framework mạnh mẽ cho việc cào dữ liệu từ web, scrapy cung cấp rất nhiều công cụ khác nhau để thực hiện gửi yêu cầu, trích rút mã html từ trang web, xây dựng đường ống dữ liệu pipeline hay để vượt hang rào bảo mật của trang web....

Một dự án scrapy gồm một số file sau:

- a) Items.py : Định hình khung cấu trúc cho dữ liệu lấy về.
- b) Middlewares.py : Các phần trung gian dùng để hỗ trợ việc truy cập vào trang web.
- c) Pipelines.py : Thiết kế đường ống chuyển dữ liệu vào nơi lưu trữ.
- d) Settings.py : Các thiết lập cài đặt cho robot.
- e) Folder spiders: Nơi ta cài đặt spider, tức là robot sẽ thực hiện công việc truy cập web và lấy dữ liệu.

Selenium: là một bộ công cụ được sử dụng để tự động các thao tác với trình duyệt, hay dễ hiểu hơn là nó giúp giả lập lại các tương tác trên trình duyệt như một người dùng thực sự.

BeautifulSoup: là một thư viện Python được sử dụng để phân tích cú pháp HTML và XML, giúp cho việc trích xuất thông tin từ các trang web trở nên dễ dàng hơn. Thư viện này cung cấp các công cụ để điều hướng, tìm kiếm và trích xuất dữ liệu từ cấu trúc HTML hoặc XML.

Requests: Thư viện requests là một thư viện HTTP đơn giản dành cho Python., nó sử dụng để gửi yêu cầu HTTP qua các dịch vụ web API.

Scrapy hoặc các phương thức gửi yêu cầu để lấy API sẽ được sử dụng ưu tiên sử dụng với các trang web có lượng tin yêu cầu lớn, nếu không thể ta sẽ chuyển qua cách khác là sử dụng công cụ tự động Selenium.

Việc tự động hóa giúp cho trang web sẽ có nhận thức rằng người đang thực hiện các thao tác là một con người thực sự, chứ không phải một robot tự động, nhờ đó mà hạn chế khả năng bị phát hiện.

Mô tả quá trình khi sử dụng scrapy: scrapy sẽ tìm phương thức start_requests ngay khi chương trình bắt đầu được thực hiện, nó kiểm tra các setting có trong file setting.py và cấu hình chương trình phù hợp, sau đó sẽ tiến hành gửi yêu cầu tới trang đích.

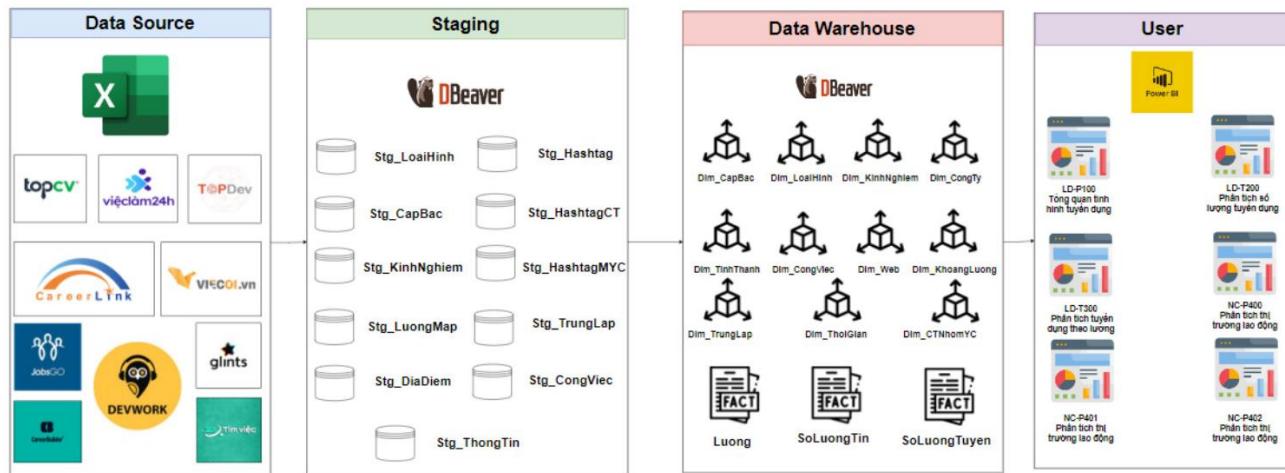
II. Phân tích và thiết kế hệ thống

Tổng quan về hệ thống phân tích dữ liệu của dự án, phần này trình bày về kiến trúc của hệ thống phân tích dữ liệu tuyển dụng, cấu trúc của vùng đệm và cơ sở dữ liệu vùng đệm, luồng dữ liệu trong hệ thống.

1. Kiến trúc hệ thống phân tích dữ liệu

Kiến trúc hệ thống phân tích dữ liệu gồm vùng chính:

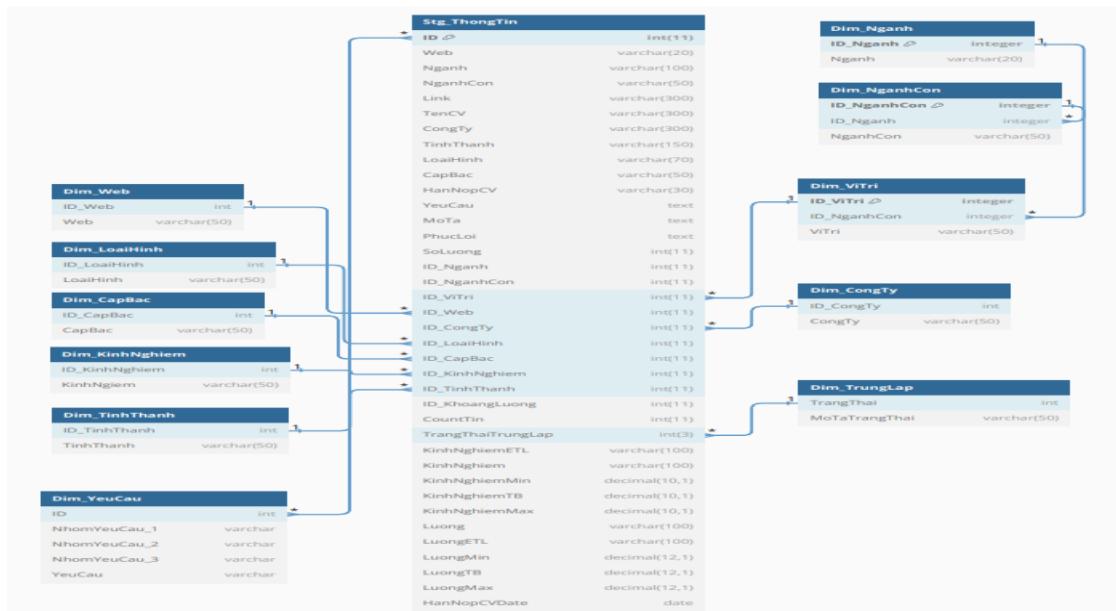
- Data Source: Các nguồn thu thập dữ liệu.
- Staging: Vùng đệm nơi lưu trữ dữ liệu thô sau khi thu thập, nơi xử lý dữ liệu.
- Data Warehouse: Kho dữ liệu lưu trữ dữ liệu lâu dài theo thời gian sau khi được xử lý.
- User: Các hoạt động khai thác dữ liệu của người dùng.



Hình 77. Kiến trúc tổng quát hệ thống phân tích dữ liệu

2. Cơ sở dữ liệu

Cấu trúc của cơ sở dữ liệu lưu trữ dữ liệu sau khi thu thập:



Hình 78. Cấu trúc cơ sở dữ liệu vùng đệm

Việc đó dữ liệu sẽ tiến hành trên bảng **Stg_ThongTin_raw** có cùng cấu trúc với **Stg_ThongTin**. Dữ liệu sau khi làm sạch một lần nữa sẽ được đó từ **Stg_ThongTin_raw** vào **Stg_ThongTin**. Mô tả một số cột trong **Stg_ThongTin_raw** mà ta sẽ đó dữ liệu sau khi lấy vào như sau:

Stg_ThongTin_raw	
ID	INT(11)
Web	VARCHAR(20)
Nganh	VARCHAR(150)
Luong	VARCHAR(100)
Link	VARCHAR(300)
TenCV	VARCHAR(300)
CongTy	VARCHAR(300)
TinhThanh	VARCHAR(150)
LoaiHinh	VARCHAR(70)
KinhNghiem	VARCHAR(100)
CapBac	VARCHAR(50)
HanNopCV	VARCHAR(30)
YeuCau	TEXT
MoTa	TEXT
PhucLoi	TEXT
SoLuong	INT(11)

Bảng 1. Một số thuộc tính sử dụng

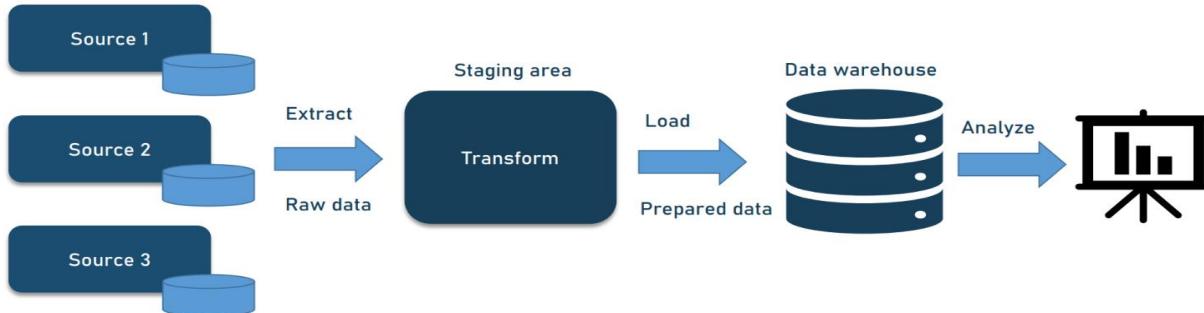
3. Phân tích tin tuyển dụng các nguồn

Ô được tô màu xanh nghĩa là đã có dữ liệu khi khảo sát, ô tô màu đỏ nghĩa là khi khảo sát chưa thấy có hoặc chưa lấy được dữ liệu.

	Web	Nganh	Link	Ten CV	Cong Ty	Tinh Thanh	Loai Hinh	Kinh Nghiem	Cap Bac	HanNop CV	Yeu Cau	Mo Ta	Phuc Loi	So Luong	Luong
TopDev															
Career Builder															
CareerLink															
DevWork															
ITNaVi															
Job3S															
Joboko															
JobsGo															
StudentJob															
TechWorks															
Viec Lam24h															
ITViec															
123Job															
TopCV															
VietNam Work															

Bảng 2. Tổng hợp khảo sát cấu trúc tin

4. Data Pipeline



Hình 79. Luồng dữ liệu trong hệ thống

Dữ liệu thô được trích xuất từ các nguồn dữ liệu khác nhau, được quy về một loại định dạng duy nhất cho tất cả các loại dữ liệu khác nhau và sau đó được tải vào một hệ thống đích đến chung là DataWarehouse hoặc DataMart. Các dữ liệu này sẽ được sử dụng phục vụ cho các công việc BI, xây dựng các mô hình dự đoán dựa trên dữ liệu...

Báo cáo này sẽ tập trung vào việc thu thập dữ liệu từ bước đầu, tức là lấy dữ liệu từ các nguồn khác nhau, chuyển về một định dạng chung và tải vào đích đến Kho dữ liệu.

III. Xây dựng chương trình

1. Cấu hình

Đối với scrapy, middleware mặc định được em sử dụng gồm hai class nhằm giả mạo user-agent và giả mạo browser-header như sau:

```
from urllib.parse import urlencode
from random import randint
import requests

Thái Dương, last month | 1 author (Thái Dương)
class ScrapeOpsFakeUserAgentMiddleware:

    @classmethod
    def from_crawler(cls, crawler):
        return cls(crawler.settings)

    def __init__(self, settings):
        self.scrapeops_api_key = settings.get('SCRAPEOPS_API_KEY')
        self.scrapeops_endpoint = settings.get('SCRAPEOPS_FAKE_USER_AGENT_ENDPOINT', 'http://headers.scrapeops.io/v1/user-agents?')
        self.scrapeops_fake_user_agents_active = settings.get('SCRAPEOPS_FAKE_USER_AGENT_ENABLED', False)
        self.scrapeops_num_results = settings.get('SCRAPEOPS_NUM_RESULTS')
        self.headers_list = []
        self._get_user_agents_list()
        self._scrapeops_fake_user_agents_enabled()

    def _get_user_agents_list(self):
        payload = {'api_key': self.scrapeops_api_key}
        if self.scrapeops_num_results is not None:
            payload['num_results'] = self.scrapeops_num_results
        response = requests.get(self.scrapeops_endpoint, params=urlencode(payload))
        json_response = response.json()
        self.user_agents_list = json_response.get('result', [])

    def _get_random_user_agent(self):
        random_index = randint(0, len(self.user_agents_list) - 1)
        return self.user_agents_list[random_index]

    def _scrapeops_fake_user_agents_enabled(self):
        if self.scrapeops_api_key is None or self.scrapeops_api_key == '' or self.scrapeops_fake_user_agents_active == False:
            self.scrapeops_fake_user_agents_active = False
        else:
            self.scrapeops_fake_user_agents_active = True

    def process_request(self, request, spider):
        random_user_agent = self._get_random_user_agent()
        request.headers['User-Agent'] = random_user_agent
        print(random_user_agent)
```

Hình 80. Giả mạo user-agent

```
Thái Dương, last month | 1 author (Thái Dương)
class ScrapeOpsFakeBrowserHeaderAgentMiddleware:

    @classmethod
    def from_crawler(cls, crawler):
        return cls(crawler.settings)

    def __init__(self, settings):
        self.scrapeops_api_key = settings.get('SCRAPEOPS_API_KEY')
        self.scrapeops_endpoint = settings.get('SCRAPEOPS_FAKE_BROWSER_HEADER_ENDPOINT', 'http://headers.scrapeops.io/v1/browser-headers?')
        self.scrapeops_fake_browser_headers_active = settings.get('SCRAPEOPS_FAKE_BROWSER_HEADER_ENABLED', False)
        self.scrapeops_num_results = settings.get('SCRAPEOPS_NUM_RESULTS')
        self.headers_list = []
        self._get_headers_list()
        self._scrapeops_fake_browser_headers_enabled()

    def _get_headers_list(self):
        payload = {'api_key': self.scrapeops_api_key}
        if self.scrapeops_num_results is not None:
            payload['num_results'] = self.scrapeops_num_results
        response = requests.get(self.scrapeops_endpoint, params=urlencode(payload))
        json_response = response.json()
        self.headers_list = json_response.get('result', [])

    def _get_random_browser_header(self):
        random_index = randint(0, len(self.headers_list) - 1)
        return self.headers_list[random_index]

    def _scrapeops_fake_browser_headers_enabled(self):
        if self.scrapeops_api_key is None or self.scrapeops_api_key == '' or self.scrapeops_fake_browser_headers_active == False:
            self.scrapeops_fake_browser_headers_active = False
        else:
            self.scrapeops_fake_browser_headers_active = True

    def process_request(self, request, spider):
        random_browser_header = self._get_random_browser_header()
        request.headers['accept-language'] = random_browser_header['accept-language']
        request.headers['sec-fetch-user'] = random_browser_header['sec-fetch-user']
        request.headers['sec-fetch-mod'] = random_browser_header['sec-fetch-mod']
        request.headers['sec-fetch-site'] = random_browser_header['sec-fetch-site']
        request.headers['sec-ch-ua-platform'] = random_browser_header['sec-ch-ua-platform']
        request.headers['sec-ch-ua-mobile'] = random_browser_header['sec-ch-ua-mobile']
        request.headers['sec-ch-ua'] = random_browser_header['sec-ch-ua']
        request.headers['accept'] = random_browser_header['accept']
        request.headers['user-agent'] = random_browser_header['user-agent']
        request.headers['upgrade-insecure-requests'] = random_browser_header.get('upgrade-insecure-requests')
```

Hình 81. Giả mạo header-browser

Để sử dụng hai class middleware này, cài đặt được sử dụng mặc định trong các file setting.py như sau:

```
SCRAPEOPS_API_KEY = 'cca4ced0-490d-41a0-b258-46f2ad7e74b3'

SCRAPEOPS_FAKE_USER_AGENT_ENDPOINT = 'https://headers.scrapeops.io/v1/user-agents'
SCRAPEOPS_FAKE_USER_AGENT_ENABLED = True
SCRAPEOPS_NUM_RESULTS = 96

SCRAPEOPS_FAKE_BROWSER_HEADER_ENDPOINT = 'https://headers.scrapeops.io/v1/browser-headers'
SCRAPEOPS_FAKE_BROWSER_HEADER_ENABLED = True
```

```
DOWNLOADER_MIDDLEWARES = {
    "TopDev.middlewares.ScrapeOpsFakeUserAgentMiddleware": 400,
    "TopDev.middlewares.ScrapeOpsFakeBrowserHeaderAgentMiddleware": 300
}
```

File pipelines.py sẽ giúp ta định nghĩa cách thức và nơi lưu dữ liệu:

```
import mysql.connector
from itemadapter import ItemAdapter

class SaveToMySQL_test_Pipeline:
    def __init__(self):
        self.conn = mysql.connector.connect(
            host='192.168.1.1',
            port='3306',
            user='tuyendungUser',
            password='sinhvienBK',
            database='ThongTinTuyenDung'
        )
        self.cur = self.conn.cursor()

    def process_item(self, item, spider):
        Thai Dương, last month • Scrapy20231
        sql = """
        INSERT IGNORE INTO Stg_ThongTin_raw(Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, SoLuong) VALUES (%s, %s,
        %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        """
        self.cur.execute(sql, (item['Web'], item['Nganh'], item['Link'], item['TenCV'], item['CongTy'], item['TinhThanh'], item['Luong'], item['LoaiHinh'], item['KinhNghiem'], item['CapBac'],
        item['HanNopCV'], item['YeuCau'], item['MoTa'], item['PhucLoi'], item['SoLuong']))
        self.conn.commit()

    def close_spider(self, spider):
        self.cur.close()
        self.conn.close()
```

```
Thái Dương, 3 weeks ago | author (Thái Dương)
class DatabaseConnector:
    def __init__(self, host, user, port, password, database):
        self.host = host
        self.user = user
        self.port = port
        self.password = password
        self.database = database

    def connect(self):
        return mysql.connector.connect(
            host=self.host,
            port = self.port,
            user=self.user,
            password=self.password,
            database=self.database
        )

    def get_links_from_database(self):
        connection = self.connect()
        cursor = connection.cursor()

        query = "SELECT Link FROM Stg_ThongTin_raw WHERE Web =\\'TopDev\\'"
        cursor.execute(query)

        links = [row[0] for row in cursor.fetchall()]

        cursor.close()
        connection.close()

        return links
```

```
ITEM_PIPELINES = {
    "TopDev.pipelines.SaveToMySQL_test_Pipeline": 300,
    "TopDev.pipelines.CleanItem": 200
}
```

File items.py sẽ giúp ta định hình dữ liệu khi lấy về:

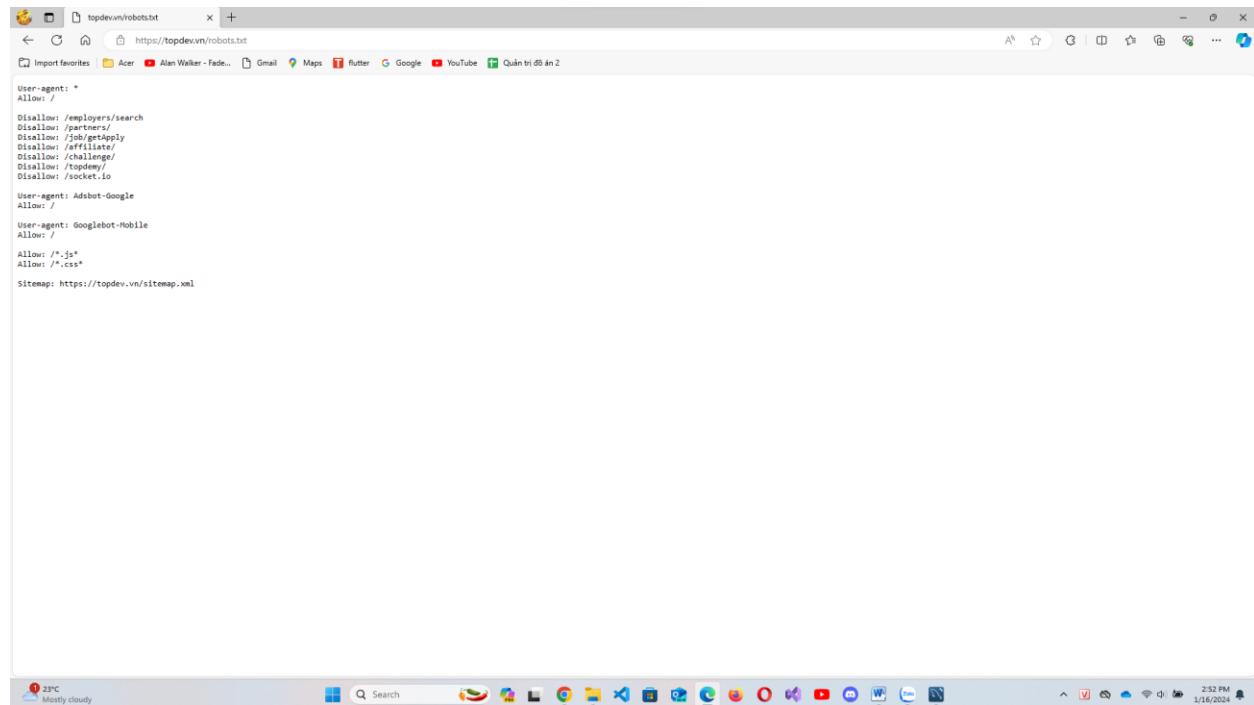
```
Thái Dương, last month | 1 author (Thái Dương)
# Define here the models for your scraped items
#
# See documentation in:
# https://docs.scrapy.org/en/latest/topics/items.html
#
import scrapy

class IT_Item(scrapy.Item):
    ID = scrapy.Field() #
    Web = scrapy.Field() #
    Nganh = scrapy.Field() #
    Link = scrapy.Field() #
    TenCV = scrapy.Field() #
    CongTy = scrapy.Field() #
    TinhThanh = scrapy.Field() #
    Luong = scrapy.Field() #
    LoaiHinh = scrapy.Field() #
    KinhNghiem = scrapy.Field() #
    CapBac = scrapy.Field() #
    YeuCau = scrapy.Field() #
    MoTa = scrapy.Field() #
    PhuLoi = scrapy.Field() #
    HanNopCV = scrapy.Field() #
    SoLuong = scrapy.Field()

Thái Dương, last month • Scrapy20231
```

Hình 82. File items.py chung cho tất cả các project sử dụng

Robots.txt là một tệp tin văn bản nằm trong thư mục gốc của trang web và cung cấp hướng dẫn cho các công cụ tìm kiếm, ở đây là các robot tự động, về các trang mà nó có thể truy cập và thu thập thông tin.



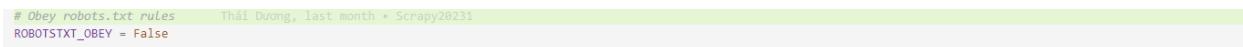
Hình 83. File robots.txt của TopDev

Hình ảnh trên là một ví dụ về tệp tin robots.txt của trang TopDev. User-agent là định danh cho trình duyệt người dùng. Allow là những trang web mà các robot có thể tiến hành truy cập và

thu thập dữ liệu, disallow là các trang web mà các robot không được phép truy cập và thao tác trên đó.

Ta có thể tùy chọn tuân thủ robots.txt hoặc không, tuy nhiên khi không tuân thủ robots.txt, việc thu thập dữ liệu sẽ có khả năng bị phát hiện khi truy cập vào các trang cấm và có khả năng rất lớn sẽ bị chặn địa chỉ IP.

Trong bài báo cáo này, thiết lập mặc định khi sử dụng scrapy sẽ được đặt là KHÔNG tuân theo robots.txt.



```
# Obey robots.txt rules
ROBOTSTXT_OBEY = False
```

Hình 84. Thiết lập mặc định không tuân theo robots.txt

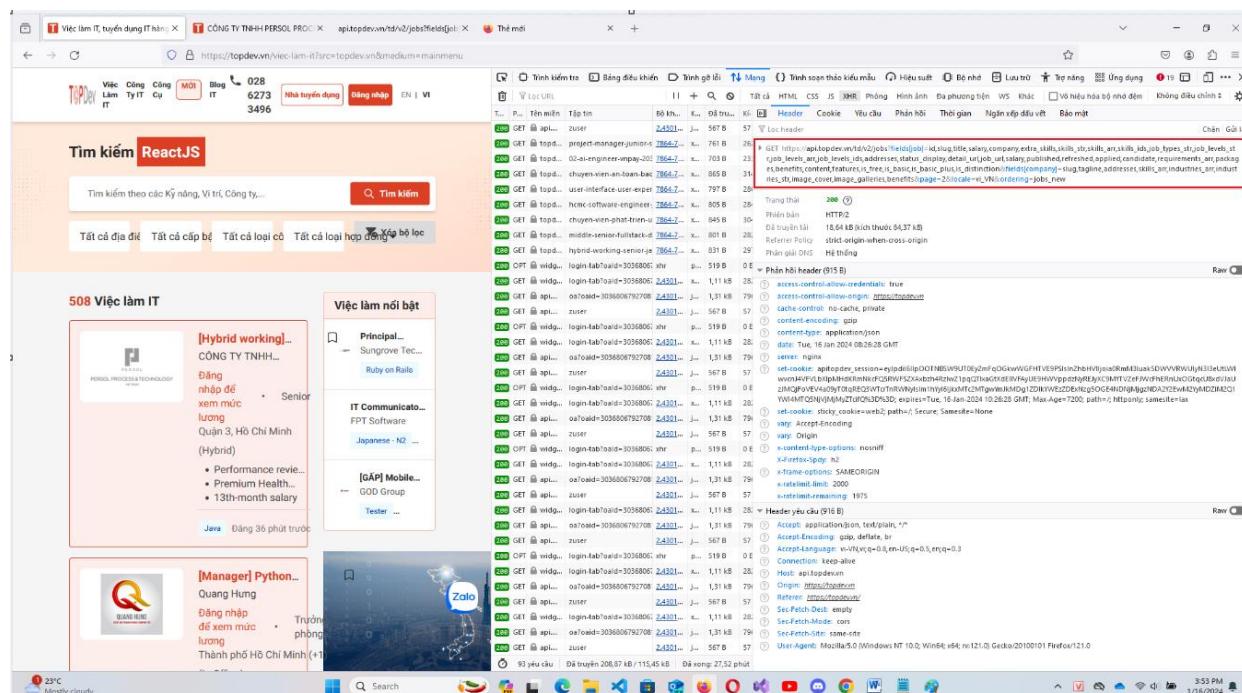
Khi cào một trang web nào đó, việc tự động luôn được khuyến nghị hơn là sử dụng các middleware, tức là các phần trung gian để hỗ trợ cho việc truy cập vào trang web dễ dàng hơn, vì việc sử dụng chúng có thể khiến cho việc bị phát hiện và bị chặn IP xảy ra dễ dàng hơn, và selenium là lựa chọn tốt cho trường hợp này. Tuy nhiên việc sử dụng selenium chỉ nên được thực hiện ở các trang có số lượng tin ít vì thời gian lâu hơn nhiều so với scrapy.

2. Xây dựng các phân hệ thu thập dữ liệu

a) TopDev

Với các trường dữ liệu là HanNopCV và SoLuong chưa có, ta sẽ để mặc định là HanNopCV là ngày lấy dữ liệu và SoLuong là 1.

Mỗi lần cuộn xuống, trang web sẽ tải thêm cho ta 10 phần tử tương ứng với 10 công việc khác nhau mới. Thuật toán là duyệt tất cả các url của TopDev có trong CSDL và sẽ không gửi yêu cầu tới các url đã có trong CSDL.



Hình 85. Khảo sát request header API (TopDev)

```

Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class ScrapeSpider(scrapy.Spider):
    name = "scrape"
    allowed_domains = ["topdev.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        for page_number in range(1, 100):
            url = f'https://api.topdev.vn/td/v2/jobs?fields=[job]=id,slug,title,salary,company,extra_skills,skills_str,skills_arr,skills_ids,job_types_str,job_levels_str,job_levels_arr,jo
yield scrapy.Request(url, method = 'GET', callback = self.parse)

```

```

def parse(self, response):
    data = response.json()

    if len(data['data']) != 0:
        for cv_count in range(len(data['data'])):
            ID = "IT_TD_" + str(data['data'][cv_count]['id']) #Xử lý ID
            Web = "TopDev" #Tên trang web
            Nganh = "IT" #Tên ngành
            Link = data['data'][cv_count]['detail_url'] #Link công việc
            TenCV = data['data'][cv_count]['title'] #Tên công việc
            CongTy = data['data'][cv_count]['company']['display_name'] #Tên công ty
            TinhThanh = data['data'][cv_count]['addresses'][0]['address_region_list'] #Địa điểm
            if data['data'][cv_count]['salary']['value'] == "": #Lương
                Luong = "Thương lượng"
            else:
                Luong = "Pending"
            LoaiHinh = data['data'][cv_count]['job_types_str'] #Loại hình
            KinhNghiem = "Pending" #Kinh nghiệm chưa có
            CapBac = data['data'][cv_count]['job_levels_str'] #Cấp bậc
            HanNopCV = date.today() #Hạn nộp CV
            YeuCau = ""
            for requirement in data['data'][cv_count]['requirements_arr']: #Yêu cầu
                if type(requirement['value']) == list:
                    for requirement_TG in requirement['value']:
                        requirement_TG = decode_special_string(requirement_TG)
                        YeuCau = YeuCau + requirement_TG + "\n"
                else:
                    requirement_TG = decode_special_string(requirement['value'])
                    YeuCau = YeuCau + requirement_TG + "\n"
            MoTa = "Pending" #Mô tả chưa có
            PhucLoi = ""
            if len(data['data'][cv_count]['company']['benefits']) == 0:
                pl = data['data'][cv_count]['benefits']
            else:
                pl = data['data'][cv_count]['company']['benefits'] #Phúc lợi
            for benefit in pl: #Phúc lợi
                if type(benefit['value']) == list:
                    for benefit_TG in benefit['value']:
                        benefit_TG = decode_special_string(benefit_TG)
                        PhucLoi = PhucLoi + benefit_TG + "\n"
                else:
                    benefit_TG = decode_special_string(benefit['value'])
                    PhucLoi = PhucLoi + benefit_TG + "\n"

            for benefit in pl:
                if type(benefit['value']) == list:
                    for benefit_TG in benefit['value']:
                        benefit_TG = decode_special_string(benefit_TG)
                        PhucLoi = PhucLoi + benefit_TG + "\n"
                else:
                    benefit_TG = decode_special_string(benefit['value'])
                    PhucLoi = PhucLoi + benefit_TG + "\n"

            SoLuong = "1"
            item = IT_Item()
            item['ID'] = ID
            item['Web'] = Web
            item['Nganh'] = Nganh
            item['Link'] = Link
            item['TenCV'] = TenCV
            item['CongTy'] = CongTy
            item['TinhThanh'] = TinhThanh
            item['Luong'] = Luong
            item['LoaiHinh'] = LoaiHinh
            item['KinhNghiem'] = KinhNghiem
            item['CapBac'] = CapBac
            item['YeuCau'] = YeuCau
            item['MoTa'] = MoTa
            item['PhucLoi'] = PhucLoi
            item['HanNopCV'] = HanNopCV
            item['SoLuong'] = SoLuong
            if Link in self.remove_url_list:
                print("Trùng lặp: ", Link)
                continue
            else:
                yield scrapy.Request(Link, callback = self.parse_2, meta = {"my_item": item})
        else:
    
```

```

def parse_2(self, response):
    item = response.meta['my_item']
    script = response.css('div script')[1].extract()
    script = script.split("{}")
    *****
    check = []
    for i in range(len(script)):
        u = 0
        if "minValue" in script[i]:
            u += 1
        if "maxValue" in script[i]:
            u += 1
        if "value" in script[i]:
            u += 1
        check.append(u)
    check_max = check.index(max(check))
    script_after = script[check_max].split("))")
    for j in range(len(script_after)):
        if "value" in script_after[j]:
            script_after_2 = script_after[j].split(",")
            break
    for k in range(len(script_after_2)):
        if "value" in script_after_2[k]:
            if item['Luong'] == "Pending":
                Luong = script_after_2[k].split(":")[1].replace("\'", "")
            item['Luong'] = Luong
    #####
    #Luong

    for i in range(len(script)):
        if "monthsOfExperience" in script[i]:
            script_after = script[i].split("))")
            KinhNghiem = script_after[0].split(":")[-1].replace("\'", "").strip() + " tháng"
            break
        else:
            KinhNghiem = "Không yêu cầu"
    item['KinhNghiem'] = KinhNghiem
    #####
    #Kinh Nghiêm

    for i in range(len(script)):
        if "Your role & responsibilities" in script[i]:
            script_after = script[i].split("Your role & responsibilities")[1]
            break
    if "Your skills & qualifications" in script_after:
        script_after_2 = script_after.split("Your skills & qualifications")[0]
    soup = BeautifulSoup(script_after_2, 'html.parser')
    Mota = soup.get_text(separator='\n', strip=True)
    item['MoTa'] = decode_special_string(Mota)
    #####
    #Mô tả

    yield item

```

b) CareerBuilder

Chương trình sẽ cố gắng thu thập tất cả các url, sau đó sẽ kiểm tra xem url đó có trong CSDL hay chưa và gửi yêu cầu tới url đó nếu nó chưa có trong CSDL.

Do SoLuong chưa có nên sẽ đặt mặc định là 1.

```

import scrapy
import math
import numpy as np
import json
from urllib.parse import urlencode
from Career.pipelines import DatabaseConnector
from Career.items import CBItem
from datetime import date
Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class CareerSpider(scrapy.Spider):
    name = "career"
    allowed_domains = ["careerbuilder.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port=3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://careerbuilder.vn/viec-lam/tat-ca-viec-lam-vi.html", callback=self.parse)

    def parse(self, response):
        job_count_text = response.css('div.job-found-amout h1::text').get()
        number_cv = ''.join(filter(str.isdigit, job_count_text))
        cv_count = int(number_cv)
        print("Số lượng công việc lấy được: ", cv_count)
        if cv_count % 50 == 0:
            max_page = int(cv_count / 50)
        else:
            max_page = math.floor(cv_count / 50) + 1
        # max_page = 10
        for page_number in range(1, max_page+1):
            page_url = f"https://careerbuilder.vn/viec-lam/tat-ca-viec-lam-trang-{page_number}-vi.html"
            yield scrapy.Request(page_url, callback=self.job_url_parse)

```

```

def job_url_parse(self, response):
    url_list_1 = response.css('.job_link::attr(href)').extract()
    url_list_1 = np.unique(url_list_1)
    url_list_1 = list(url_list_1) #Lấy mảng url đầu tiên và loại bỏ trùng lặp
    #####
    url = response.url
    page_number_str = ''.join(c for c in url.split('-')[-2] if c.isdigit())
    page_number = int(page_number_str)
    #####
    if page_number >=1 and page_number <= 9:
        data_one = 'a:1:{s:4:"PAGE";s:1:' + str(page_number) + "';}"
    elif page_number >=10 and page_number <99:
        data_one = 'a:1:{s:4:"PAGE";s:2:' + str(page_number) + "';}"
    elif page_number >=100 and page_number <999:
        data_one = 'a:1:{s:4:"PAGE";s:3:' + str(page_number) + "';}"
    data_two = 'a:0:{}'
    #####
    # Mã hóa dữ liệu
    encoded_data_one = urlencode({'dataOne': data_one})
    encoded_data_two = urlencode({'dataTwo': data_two})
    # Kết hợp dữ liệu
    payload = f'{encoded_data_one}&{encoded_data_two}'

    # Định nghĩa header gửi yêu cầu
    header = {
        "Accept": "application/json, text/javascript, */*; q=0.01",
        "Accept-Encoding": "gzip, deflate, br",
        "Accept-Language": "en-US,en;q=0.9,vi;q=0.8",
        "Content-Type": "application/x-www-form-urlencoded; charset=UTF-8",
        "Origin": "https://careerbuilder.vn",
        "Referer": url,
        "X-Requested-With": "XMLHttpRequest",
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36 Edg/119.0.0.0"
    }

    # Lấy dữ liệu và lấy kết quả trả về dạng json
    yield scrapy.Request("https://careerbuilder.vn/search-jobs", method = 'POST', body=json.dumps(payload), headers = header, callback = self.json_parse, meta={'url_list_1': url_list_1})

```

```

def json_parse(self, response):
    json_data = response.json()
    page_number = response.meta.get('page_number')
    url_list_1 = response.meta.get('url_list_1', [])
    url_list_2 = []
    for i in range(len(json_data['data'])):
        url_list_2.append(json_data['data'][i]['LINK_JOB']) #Lấy được mảng url_list_2
    url_list = url_list_1 + url_list_2
    print("Số url của trang: ", page_number, "là: ", len(url_list))
    for url_job in url_list:
        if url_job in self.remove_url_list:
            print("Trùng lặp: ", url_job)
            continue
        else:
            yield scrapy.Request(url_job, callback = self.job_parse)

```

```

def job_parse(self, response):
    ID = "CB_" + response.url.split(".")[-2]
    Web = "CareerBuilder"
    Link = response.url
    Nganh =""
    Luong =""
    LoaiHinh =""
    CapBac =""
    HanNopCV = date.today() #Trường kinh nghiệm đã được xử lý phía dưới
    #####
    try:
        col_1 = response.css('div[class="detail-box has-background"]')[0] #Loại 1
        for i in range(len(col_1.css('ul li'))):
            try:
                if 'Ngành nghề' in col_1.css('ul li')[i].css('strong::text').extract():
                    Nganh = col_1.css('ul li')[i].css('p a::text').getall()
                if 'Hình thức' in col_1.css('ul li')[i].css('strong::text').extract():
                    LoaiHinh = col_1.css('ul li')[i].css('p::text').get()
            except:
                continue
        TenCV = response.css('div.job-desc h1[class="title"]::text').get()
        CongTy = response.css('div.job-desc a[class = "employer-job-company-name"]::text').get()
        TinhThanh = response.css('div.map p a::text').get()
        col_2 = response.css('div.detail-box.has-background')[1]
        for i in range(len(col_2.css('ul li'))):
            try:
                if 'Lương' in col_2.css('ul li')[i].css('strong::text').extract():
                    Luong = col_2.css('ul li')[i].css('p::text').get()
                if 'Kinh nghiệm' in col_2.css('ul li')[i].css('strong::text').extract():
                    KinhNghiem = col_2.css('ul li')[i].css('p::text').get()
                if 'Cấp bậc' in col_2.css('ul li')[i].css('strong::text').extract():
                    CapBac = col_2.css('ul li')[i].css('p::text').get()
                if 'Hết hạn nộp' in col_2.css('ul li')[i].css('strong::text').extract():
                    HanNopCV = col_2.css('ul li')[i].css('p::text').get()
            except:
                continue
        PhucLoi = response.css('ul.welfare-list li::text').getall()
    
```

Ngoài ra, vì số lượng tin là khá lớn, nên ta tăng số lượng yêu cầu được gửi cùng một lúc tới trang web là 50 yêu cầu. Tuy nhiên, việc tăng số lượng yêu cầu lên cao có thể khiến server quá tải và có thể khiến việc bị chặn dễ dàng xảy ra hơn.

```
# Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 50
# Configure a delay for requests for the same website (default: 0)
# See also autothrottle settings and the PROXY_LIST setting
# CONCURRENT_REQUESTS_PER_DOMAIN = 16
# CONCURRENT_REQUESTS_PER_IP = 16
# Set the proxy list
# PROXY_LIST = 'proxies.txt'
```

c) CareerLink

Ý tưởng xây dựng chương trình là lấy về tất cả url của các công việc, sau đó sẽ kiểm tra xem url đó có tồn tại trong CSDL hay chưa, nếu chưa thì gửi yêu cầu tới url đó để lấy dữ liệu.

Do Số Lượng chưa có nên được đặt mặc định là 1.

```
import scrapy
from CareerLink.items import CareerlinkItem
from datetime import date, timedelta
from CareerLink.pipelines import DatabaseConnector
Thái Dương, 4 weeks ago | 1 author (Thái Dương)
class CareerlinkSpider(scrapy.Spider):
    name = "careerlink"
    allowed_domains = ["www.careerlink.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request('https://www.careerlink.vn/vieclam/list', callback = self.count_job)

    def count_job(self, response):
        job_count = int(response.css('.jobs-count-number::text').get().replace('\n', ""))
        if job_count % 50 == 0:
            max_page = job_count / 50
        else:
            max_page = job_count // 50 + 1
        for page_number in range(1, max_page+1):
            # for page_number in range(1, 10):
            page_url = "https://www.careerlink.vn/vieclam/list?page=" + str(page_number)
            yield scrapy.Request(url=page_url, callback=self.job_url_parse)

    def job_url_parse(self, response):
        job_url_list = response.css('li.list-group-item a.job-link::attr(href)').extract()
        for job_url in job_url_list:
            if "https://www.careerlink.vn" in job_url:
                job_next_url = job_url
            else:
                job_next_url = "https://www.careerlink.vn" + job_url

            if job_next_url in self.remove_url_list:
                print("Trùng lặp: ", job_next_url)
                continue
            else:
                yield scrapy.Request(url=job_next_url, callback=self.job_parse)
```

```

def job_parse(self, response):
    ID = "CareerLink_" + (response.url).split("/")[-1].split("?")[0]
    Web = "CareerLink"
    for i in range(len(response.css('div[class="col-6 pl-1 pr-3 pl-md-2"]').css('div[class="job-summary-item d-block"]'))):
        if response.css('div[class="col-6 pl-1 pr-3 pl-md-2"]')[i].css('div[class="my-0 summary-label"]')[1].css('::text').get() == "Ngành nghề":
            Nganh_TG = response.css('div[class="col-6 pl-1 pr-3 pl-md-2"]')[i].css('div[class="job-summary-item d-block"]')[1].css('div')[2].css('*:not(:empty)::text').getall()
            Nganh = ''
            for Nganh_TG in Nganh_TG:
                if Nganh_TG != '\n':
                    Nganh += Nganh_TG
            if 'CNTT' in Nganh:
                Nganh = 'IT'
            Link = response.url
            TenCV = response.css('h1[class="job-title mb-0"]::text').get()
            CongTy = response.css('p[class="org-name mb-2"] span::text').get()
            TinhThanh = ''
            TinhThansh_TG = response.css('div[class="d-flex align-items-start mb-2"] *:not(:empty)::text').getall()
            for TinhThansh_TG in TinhThansh_TG:
                if TinhThansh_TG != '\n':
                    TinhThanh += TinhThansh_TG
            Luong = response.css('div[class="d-flex align-items-center mb-2"]')[0].css('span::text').get()
            KinhNghiem = response.css('div[class="d-flex align-items-center mb-2"]')[1].css('span::text').get()
            deadline = response.css('div[class="d-flex align-items-center mb-2"]')[2].css('b::text').get().split("\n")[1]
            try:
                HanlopCV = date.today() + timedelta(days = int(deadline))
            except:
                HanlopCV = date.today()
            for i in range(len(response.css('div[class="col-6 pr-1 pl-3 pr-md-2"] div[class="job-summary-item d-block"]'))):
                if response.css('div[class="col-6 pr-1 pl-3 pr-md-2"] div[class="job-summary-item d-block"]')[i].css('div[class="my-0 summary-label"]::text').get() == "Cấp bậc":
                    CapBac = response.css('div[class="col-6 pr-1 pl-3 pr-md-2"] div[class="job-summary-item d-block"]')[i].css('div')[2].css('::text').get()
                if response.css('div[class="col-6 pr-1 pl-3 pr-md-2"] div[class="job-summary-item d-block"]')[i].css('div[class="my-0 summary-label"]::text').get() == "Loại công việc":
                    LoaiHinh = response.css('div[class="col-6 pr-1 pl-3 pr-md-2"] div[class="job-summary-item d-block"]')[i].css('div')[2].css('::text').get()
            SoLuong = 1
            MoTa = ''
            MoTas_TG = response.css('div[id="section-job-description"] *:not(:empty)::text').getall()
            for MoTa_TG in MoTas_TG:
                if MoTa_TG != '\n':
                    MoTa += MoTa_TG
            YeuCau = ''
            YeuCaus_TG = response.css('div[id="section-job-skills"] *:not(:empty)::text').getall()
            for YeuCau_TG in YeuCaus_TG:
                if YeuCau_TG != '\n':
                    YeuCau += YeuCau_TG
            PhucLoi = ''
            PhucLois_TG = response.css('div[id="section-job-benefits"] *:not(:empty)::text').getall()
            for PhucLoi_TG in PhucLois_TG:
                PhucLoi += PhucLoi_TG
        MoTa = ''
        MoTas_TG = response.css('div[id="section-job-description"] *:not(:empty)::text').getall()
        for MoTa_TG in MoTas_TG:
            if MoTa_TG != '\n':
                MoTa += MoTa_TG
        YeuCau = ''
        YeuCaus_TG = response.css('div[id="section-job-skills"] *:not(:empty)::text').getall()
        for YeuCau_TG in YeuCaus_TG:
            if YeuCau_TG != '\n':
                YeuCau += YeuCau_TG
        PhucLoi = ''
        PhucLois_TG = response.css('div[id="section-job-benefits"] *:not(:empty)::text').getall()
        for PhucLoi_TG in PhucLois_TG:
            PhucLoi += PhucLoi_TG
    SoLuong = 1
    ThoiDuong = 4 weeks ago * final
    MoTa = ''
    MoTas_TG = response.css('div[id="section-job-description"] *:not(:empty)::text').getall()
    for MoTa_TG in MoTas_TG:
        if MoTa_TG != '\n':
            MoTa += MoTa_TG
    item = CareerLinkItem()
    item['ID'] = ID
    item['Web'] = Web
    item['Nganh'] = Nganh
    item['Link'] = Link
    item['TenCV'] = TenCV
    item['CongTy'] = CongTy
    item['TinhThanh'] = TinhThanh
    item['Luong'] = Luong
    item['LoaiHinh'] = LoaiHinh
    item['KinhNghiem'] = KinhNghiem
    item['CapBac'] = CapBac
    item['YeuCau'] = YeuCau
    item['MoTa'] = MoTa
    item['PhucLoi'] = PhucLoi
    item['HanlopCV'] = HanlopCV
    item['SoLuong'] = SoLuong
    yield item

```

```

PhucLoi = ''
PhucLois_TG = response.css('div[id="section-job-benefits"] *:not(:empty)::text').getall()
for PhucLoi_TG in PhucLois_TG:
    if PhucLoi_TG != '\n':
        PhucLoi += PhucLoi_TG
item = CareerLinkItem()
item['ID'] = ID
item['Web'] = Web
item['Nganh'] = Nganh
item['Link'] = Link
item['TenCV'] = TenCV
item['CongTy'] = CongTy
item['TinhThanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhucLoi'] = PhucLoi
item['HanlopCV'] = HanlopCV
item['SoLuong'] = SoLuong
yield item

```

Tuy nhiên, khi thực hiện chương trình lại nhận về lỗi mã 429. Do đó ta sẽ giới hạn khả năng gửi yêu cầu của chương trình tới trang web với các tùy chọn sau trong setting.py :

```

DOWNLOAD_DELAY = 1
AUTOTHROTTLE_ENABLED = True
AUTOTHROTTLE_START_DELAY = 6
AUTOTHROTTLE_TARGET_CONCURRENCY = 1.0
CONCURRENT_REQUESTS = 6
CONCURRENT_REQUESTS_PER_DOMAIN = 6
RETRY_TIMES = 5

```

Cài đặt này sẽ giới hạn số lượng yêu cầu cùng một lúc mà chương trình gửi đi là 6 và số lượng yêu cầu cùng lúc chương trình gửi tới 1 tên miền là 6, thời gian chờ đợi giữa các lần gửi yêu cầu là 1 giây, và nếu 1 yêu cầu thất bại thì số lần thử lại(retry_times) là 5 lần.

d) DevWork

```

import scrapy
from DevWork.items import DevWorkItem
import random
from time import sleep
from DevWork.pipelines import DatabaseConnector
from datetime import date
Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class DeworkSpider(scrapy.Spider):
    name = "devwork"
    allowed_domains = ["devwork.vn"]
    start_urls = ["https://devwork.vn/viec-lam?page=1"]
    db_connector = DatabaseConnector(host="103.56.158.31", port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
    remove_url_list_local = db_connector.get_links_from_database()
    remove_url_list = remove_url_list_local
    print("Số lượng url trong CSDL: ", len(remove_url_list))

    def parse(self, response):
        job_url_list = []
        job_list = response.css('div[class="listing"]')
        for job_num in range(len(job_list)):
            job_url_list.append(job_list[job_num].css('a::attr(href)').get())
        for job_url in job_url_list:
            if "https://devwork.vn" in job_url:
                job_next_url = job_url
            else:
                job_next_url = "https://devwork.vn" + job_url
            if job_next_url in self.remove_url_list:
                print("Trùng lặp: ", job_next_url)
                continue
            else:
                yield scrapy.Request(job_next_url, callback=self.parse_job)
        #Trang tiếp theo
        next_page_url = response.css('li[class="pagination-item"] a.page-next::attr(href)').get()
        if next_page_url is not None:
            yield response.follow(next_page_url, callback = self.parse)

    def parse_job(self, response):
        ID = "IT_DL_" + str((response.url).split('/')[-2])
        Web = "DevWork"
        Link = response.url
        NganH = "IT"
        TenCV = response.css('div[class="header-details"] h1[class="mb-3"]::text').get().replace("\n", "").strip()
        CongTy = response.css('div[class="header-details"] h5[class="mb-10 fw-400"] a::text').get().replace("\n", "").strip()
        TinhThanh = response.css('div[class="header-details"] p::text').get().replace("\n", "").strip()
        #*****
        col = response.css('div[class="job-overview mt-20"] li')
        Luong = col[0].css('div span::text').get()
        LoaiHinh = col[5].css('div span::text').get()          Thái Dương, last month • Scrapy2023
        KinhNghiem = col[1].css('div span::text').get()
        CapBac = col[3].css('div span::text').get()
        try:
            HanNopCV = col[6].css('div span::text').get()
            if HanNopCV == "":
                HanNopCV = date.today()
        except:
            HanNopCV = date.today()
        SoLuong = col[7].css('div span::text').get()
        #*****
        text = response.css('div[class="block-desc"]')
        MoTa = ''
        MoTas_TG = text[0].css('::text').extract()
        for MoTa_TG in MoTas_TG:
            MoTa += MoTa_TG
        YeuCau = ''
        YeuCaus_TG = text[1].css('::text').extract()
        for YeuCau_TG in YeuCaus_TG:
            YeuCau += YeuCau_TG
        PhuLoi = ''
        PhuLois_TG = text[3].css('::text').extract()
        for PhuLoi_TG in PhuLois_TG:
            PhuLoi += PhuLoi_TG
        #*****

```

```

item = DevWorkItem()
item['ID'] = ID
item['Web'] = Web
item['Nganh'] = Nganh
item['Link'] = Link
item['TenCV'] = TenCV
item['CongTy'] = CongTy
item['TinhThanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhucLoi'] = PhucLoi
item['HanNopCV'] = HanNopCV
item['SoLuong'] = SoLuong
yield item

```

Xây dựng chương trình với ý tưởng rằng ta sẽ lấy về các url của DevWork đã có trong CSDL và gửi yêu cầu tới các url chưa có trong CSDL để lấy thông tin.

e) ITNaVi

Ý tưởng thuật toán là gửi yêu cầu tới 1 trang web, sau đó trích ra tất cả các id của 10 công việc trong trang đó, từ đó xác định được url ajax để lấy dữ liệu công việc, và gửi yêu cầu tới url để lấy về file json chứa dữ liệu công việc. Trong file json trả về sẽ không có yêu cầu, mô tả, phúc lợi, số lượng và hạn nộp cv. Nếu kiểm tra thấy url công việc chưa từng có trong CSDL, ta sẽ gửi yêu cầu tới url đó để lấy nốt các thông tin còn lại, nếu không thì ta sẽ chuyển sang url khác và lặp lại quá trình.

Vì không có kinh nghiệm nên ta sẽ để mặc định giá trị KinhNghiem là “Không có”.

```

from typing import Iterable
import scrapy
from scrapy.http import Request
import math
from bs4 import BeautifulSoup
import json
import requests
from ITNaVi.items import ITNaVi
from ITNaVi.pipelines import DatabaseConnector
import re
import html

def decode_special_string(input_str):
    # Tìm các chuỗi Unicode và giải mã
    unicode_matches = re.finditer(r'\\u([0-9a-fA-F]{4})', input_str)
    decoded_str = input_str
    for match in unicode_matches:
        unicode_str = match.group(0)
        unicode_char = chr(int(unicode_str[2:], 16))
        decoded_str = decoded_str.replace(unicode_str, unicode_char)

    # Giải mã các ký tự HTML
    decoded_str = html.unescape(decoded_str)

    return decoded_str

Thái Dương, 4 weeks ago | author (Thái Dương)
class ItnaviSpider(scrapy.Spider):
    name = "itnavi"
    allowed_domains = ["itnavi.com.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://itnavi.com.vn/job?", callback = self.parse)


```

```

def parse(self, response):
    text = response.css('.js-tab__show h1::text').get()
    number_cv = ''.join(filter(str.isdigit, text))
    cv_count = int(number_cv)
    if cv_count % 10 == 0:
        max_page = int(cv_count / 10)
    else:
        max_page = math.floor(cv_count/10) + 1
    #*****
    for page_number in range(1, max_page+1):
        page_url = "https://itnavi.com.vn/job?page=" + str(page_number)
        yield scrapy.Request(page_url, callback = self.id_parse)

def id_parse(self, response):
    jobs_id_list = response.css('.jsl-item::attr(data-id)').extract()
    for job_id in jobs_id_list:
        job_url_by_id = "https://itnavi.com.vn/ajax/get-job-by-id/" + job_id
        yield scrapy.Request(job_url_by_id, callback = self.it_parse)    Thái Dương, last month • Scrapy20231

def it_parse(self, response):
    data_json = response.json()
    #*****
    item = ITNaVi()
    ID = "IT_NV_" + str(data_json["data"]["job_id"])
    Web = "ITNaVi"
    Nganh = "IT"
    Link = data_json["data"]["job_slug"]
    TenCV = data_json["data"]["job_name"]
    Congty = data_json["data"]["company_name"]
    TinhThanh = data_json["data"]["job_addresses"]
    Luong = data_json["data"]["job_salary"]
    LoaiHinh = data_json["data"]["job_career_type"]
    KinhNghiem = "Không có"
    CapBac = data_json["data"]["job_career_level"]
    item['ID'] = ID
    item['Web'] = Web
    item['Nganh'] = Nganh
    item['Link'] = Link
    item['TenCV'] = TenCV
    item['Congty'] = Congty
    item['Tinhnhanh'] = TinhThanh
    item['Luong'] = Luong
    item['Loaihinh'] = LoaiHinh
    item['KinhNghiem'] = KinhNghiem
    item['CapBac'] = CapBac
    if Link in self.remove_url_list:
        print("Trùng lặp: ", Link)
        return
    else:
        yield scrapy.Request(Link, method = 'GET', callback = self.it_parse_2, meta = {'item': item, 'data_json': data_json})
    #*****


def it_parse_2(self, response):
    HanNopCV = response.text.split("\\"validThrough": \"")[-1].split("\\")
    item = response.meta.get('item')
    data_json = response.meta.get('data_json')
    #*****
    soup = BeautifulSoup(response.text, "html.parser")
    icon_i = soup.find('i', class_="fas fa-users")
    if icon_i:
        sibling_p = icon_i.find_next_sibling("p")
        if sibling_p:
            Soluong = sibling_p.get_text(strip=True)
        else:
            Soluong = "Không có"
    else:
        Soluong = "Không có"
    #*****
    root = data_json["data"]["job_content"]
    soup = BeautifulSoup(root, 'html.parser')
    cleaned_root = soup.get_text(separators='').lower()
    #*****
    split_1 = ["n yêu cầu:\n", "\n yêu cầu công việc\n"]
    split_2 = ["\n ché độ:\n", "\n các phúc lợi dành cho bạn\n", "\n tại sao bạn sẽ yêu thích làm việc tại đây\n"]
    check = False
    split_check_2 = ""
    for split_string_1 in split_1:
        if split_string_1 in cleaned_root:
            Mota = cleaned_root.split(split_string_1)[0]
            cleaned_root_TG = decode_special_string(cleaned_root.split(split_string_1)[1])
    for split_string_2 in split_2:
        if split_string_2 in cleaned_root_TG:
            split_check_2 += split_string_2
            check = True
    if check == True:
        YeuCau = decode_special_string(cleaned_root_TG.split(split_check_2)[0])
        PhucLoi = decode_special_string(cleaned_root_TG.split(split_check_2)[1])
    else:
        YeuCau = cleaned_root_TG
        PhucLoi = ""
    #*****
    item['HanNopCV'] = HanNopCV
    item['Soluong'] = Soluong
    item['YeuCau'] = YeuCau
    item['Mota'] = Mota
    item['PhucLoi'] = PhucLoi
    yield item

```

f) Job3S

Ý tưởng là lấy ra tất cả 15 url trong 1 trang danh sách việc, sau đó sẽ kiểm tra xem url đó đã có trong CSDL hay chưa, nếu chưa, ta sẽ gửi yêu cầu tới url đó để thực hiện việc lấy dữ liệu.

```

import scrapy
import re
from job3s.items import IT_Item
from job3s.pipelines import DatabaseConnector
Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class Job3swebSpider(scrapy.Spider):
    name = "job3sweb"
    allowed_domains = ["job3s.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://job3s.vn/tim-viec-lam?page=1", callback = self.parse)

    def parse(self, response):
        job_count = response.css('.count_title::text').get()
        count = re.search(r'(\b\d+\b)', job_count).group()
        if int(count)%15 == 0:
            max_page = int(count) / 15
        else:
            max_page = int(count) // 15 + 1
        for page_number in range(1, max_page+1):
            yield scrapy.Request("https://job3s.vn/tim-viec-lam?page={page_number}", callback = self.it_parse)

    def it_parse(self, response):
        job_list_url = response.css('[class="content_news_title"] a::attr(href)').extract()
        for job_url in job_list_url:
            if "https://job3s.vn" in job_url:
                next = job_url
            else:
                next = "https://job3s.vn" + job_url
            if next in self.remove_url_list:
                print("Trùng lặp: ", next)
                continue
            else:
                yield scrapy.Request(next, callback = self.it_parse_2)

    def it_parse_2(self, response):
        Web = 'Job3s'
        Nganh = response.css('[class="breadcrumb_new d_flex"] a')[1].css('::text').get().replace("Việc làm", "")          Thái Dương, 3 weeks ago + finalv7
        Link = response.url
        TenCV = response.css('[class="cl_primary pd_b12"]::text').get().replace("\n", "").strip()
        CongTy = response.css('[class="font_s20 line_h23 font_w400 cl_55"]::text').get()
        try:
            TinhThanh = response.css('[class="my-3"] p::text').get().replace("\n", "").strip().split(":")[0].replace("-", "").strip()
        except:
            TinhThanh = "Toàn quốc"
        for i in range(len(response.css('[class="d_flex align_s box-item"]')):
            text = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_55 block"]::text').get().replace("\n", "").strip()
            if "Mô" in text and "lương" in text:
                try:
                    Luong = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_primary block mt_8"]::text').get().replace("\n", "").strip()
                except:
                    Luong = "Thỏa thuận"
            if 'Hình' in text and 'thức làm việc' in text:
                try:
                    LoaiHinh = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_primary block mt_8"]::text').get().replace("\n", "").strip()
                except:
                    LoaiHinh = "Toàn thời gian"
            if 'Kinh' in text and 'nghiem' in text:
                try:
                    KinhNghiem = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_primary block mt_8"]::text').get().replace("\n", "").strip()
                except:
                    KinhNghiem = "Không có"
            if 'Cấp' in text and 'bậc' in text:
                try:
                    CapBac = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_primary block mt_8"]::text').get().replace("\n", "").strip()
                except:
                    CapBac = "Không có"
            if 'Số' in text and 'lượng tuyển' in text:
                try:
                    SoLuong = response.css('[class="d_flex align_s box-item"]')[i].css('[class="font_s16 line_h19 font_w400 cl_primary block mt_8"]::text').get().replace("\n", "").strip()
                except:
                    SoLuong = "1"

```

```

YeuCau = ""
YeuCau_List = response.css('[_class="item_box item-box-content"]')[1].css('::text').getall()
for i in range(len(YeuCau_List)):
    YeuCau += YeuCau_List[i]

MoTa = ""
MoTa_List = response.css('[_class="item_box item-box-content"]')[0].css('::text').getall()
for i in range(len(MoTa_List)):
    MoTa += MoTa_List[i]

PhucLoi = ""
PhucLoi_List = response.css('[_class="item_box item-box-content"]')[2].css('::text').getall()
for i in range(len(PhucLoi_List)):
    PhucLoi += PhucLoi_List[i]

HanNopCV = response.css('[_class="box-header-job__time"] .hight-light::text').get()

item = IT_Item()
item['Web'] = Web
item['Nganh'] = Nganh
item['Link'] = Link
item['TenCV'] = TenCV
item['CongTy'] = CongTy
item['TinhThanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhucLoi'] = PhucLoi
item['HanNopCV'] = HanNopCV
item['SoLuong'] = SoLuong

yield item

```

g) Joboko

Do SoLuong không có nên ta để mặc định SoLuong là 1.

Hai giá trị KinhNghiem và CapBac không có nên được để giá trị mặc định của cả hai là “Không có”.

Ý tưởng của thuật toán là bắt đầu từ trang <https://vn.joboko.com/viec-lam-theo-nganh-nghiep> và duyệt qua tất cả các ngành nghề, trong 1 ngành nghề thì duyệt khoảng 150 trang, như đã nói lúc trước, mỗi trang gồm 10 tin tuyển dụng. Việc duyệt như vậy sẽ giúp cho ta có được một số lượng tin đủ nhiều và đạt tiêu chuẩn cho việc phân tích dữ liệu.



Hình 86. Số lượng ngành nghề trên Joboko

Với mỗi url công việc lấy được trong một trang, ta kiểm tra xem url đó đã có trong CSDL hay chưa, nếu chưa ta sẽ tiến hành gửi yêu cầu tới url đó và thực hiện lấy dữ liệu.

```

import scrapy
from joboko.items import IT_Item
from joboko.pipelines import DatabaseConnector
Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class JobokowebSpider(scrapy.Spider):
    name = "jobokoweb"
    allowed_domains = ["vn.joboko.com"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user="tuyendungUser", password="sinhvienBK", database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://vn.joboko.com/viec-lam-theo-nganh-nghie", callback = self.parse)

    def parse(self, response):
        list_branch_url = response.css('div[class="item"] ul li a::attr(href)').extract()
        list_branch_name = response.css('div[class="item"] ul li a span::text').extract()
        for i in range(len(list_branch_url)):
            if 'https://vn.joboko.com' in list_branch_url[i]:
                branch_url = list_branch_url[i]
            else:
                branch_url = 'https://vn.joboko.com' + list_branch_url[i]
            branch_name = list_branch_name[i]

            for page_number in range(1, 151):
                branch_page = f'{branch_url}?p={page_number}'
                yield scrapy.Request(branch_page, callback = self.branch_parse, meta = {'branch_name': branch_name})

    def branch_parse(self, response):
        job_url_list = response.css('.item-title a::attr(href)').extract()
        branch_name = response.meta.get("branch_name")
        for job_url in job_url_list:
            if 'https://vn.joboko.com' in job_url:
                next = job_url
            else:
                next = 'https://vn.joboko.com' + job_url

            if next in self.remove_url_list:
                print("Trùng lặp: ", next)
                continue
            else:
                yield scrapy.Request(next, callback = self.it_parse, meta = {'branch_name': branch_name})

    def it_parse(self, response):
        Web = 'Joboko'
        Nganh = response.meta.get("branch_name")
        Link = response.url
        TenCV = response.css('[class="nw-company-hero__info"] h2 a::text').get()
        CongTy = response.css('[class="nw-company-hero__info"] a.nw-company-hero__text::text').get()
        TinhThanh = response.css('[class="nw-company-hero__address"] a::text').get()
        Luong = response.css('[class="col-12"] span::text').get()
        KinhNghiem = "Không có"
        CapBac = "Không có"
        for i in range(len(response.css('[class="col-12 col-md-6"]'))):
            if 'Loại hình' in response.css('[class="col-12 col-md-6"]')[i].css('.item-content::text').get():
                LoaiHinh = response.css('[class="col-12 col-md-6"]')[i].css('span::text').get()
            if 'Kinh nghiệm' in response.css('[class="col-12 col-md-6"]')[i].css('.item-content::text').get():
                KinhNghiem = response.css('[class="col-12 col-md-6"]')[i].css('span::text').get()
            if 'Chức vụ' in response.css('[class="col-12 col-md-6"]')[i].css('.item-content::text').get():
                CapBac = response.css('[class="col-12 col-md-6"]')[i].css('span::text').get()

        YeuCau = ""
        YeuCau_List = response.css('[class="text-justify"]')[1].css('*::not(:empty)::text').getall()
        for i in range(len(YeuCau_List)):
            YeuCau += YeuCau_List[i]

        MoTa = ""
        MoTa_List = response.css('[class="text-justify"]')[0].css('*::not(:empty)::text').getall()
        for i in range(len(MoTa_List)):
            MoTa += MoTa_List[i]

        PhucLoi =""
        PhucLoi_List = response.css('[class="text-justify"]')[0].css('*::not(:empty)::text').getall()
        for i in range(len(PhucLoi_List)):
            PhucLoi += PhucLoi_List[i]

        HanNopCV = response.css('[class="item-date"]::attr(data-value)').get().split("T")[0]
        SoLuong= "1"
        Thái Dương, 4 weeks ago * final

```

```

item = IT_Item()
item['Web'] = Web
item['Nganh'] = Nganh
item['Link'] = Link
item['TenCV'] = TenCV
item['CongTy'] = CongTy
item['TinhHanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhucLoi'] = PhucLoi
item['HanNopCV'] = HanNopCV
item['SoLuong'] = SoLuong

yield item

```

Như đã đề cập, với mỗi trang 10 tin, mỗi ngành nghề truy cập vào 150 trang, và có khoảng 95 ngành nghề, nên số tin sẽ khá lớn, vì vậy trong setting.py sẽ cài đặt số requests cùng lúc được gửi đi như sau :

```
# Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 50
```

Việc tăng số request tại 1 thời điểm không được khuyến khích vì nó sẽ làm tăng áp lực lên server nhận yêu cầu.

h) Jobsgo

Do số lượng tuyển dụng không có đề cập trong tin, nên ta để giá trị mặc định là 1.

Ý tưởng của thuật toán là lấy ra tất cả 50 url tin tuyển dụng của 1 trang, sau đó với mỗi url ta kiểm tra trong CSDL xem url đó đã có trong CSDL chưa, nếu chưa có trong CSDL, ta tiến hành gửi yêu cầu tới url đó để thực hiện lấy dữ liệu. Thực hiện vòng lặp như vậy cho tới khi duyệt qua tất cả các trang tin.

```

import scrapy
import re
from Jobsgo.pipelines import DatabaseConnector
from datetime import date, timedelta
from Jobsgo.items import IT_Item

Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class JobsgoSpider(scrapy.Spider):
    name = "jobsgo"
    allowed_domains = ["jobsgo.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://jobsgo.vn/viec-lam.html", callback = self.job_count_parse)

    def job_count_parse(self, response):
        so_luong_viec_lam_text = re.search(r'\b(\d+)\b', response.css('.mrg-bot-15 h1::text').get()).group()
        if int(so_luong_viec_lam_text) % 50 == 0:
            max_page = int(so_luong_viec_lam_text)/ 50
        else:
            max_page = int(so_luong_viec_lam_text) // 50 + 1

        for page_number in range(1, max_page+1):
            yield scrapy.Request(f"https://jobsgo.vn/viec-lam.html?page={page_number}", callback = self.job_url_parse)

    def job_url_parse(self, response):
        job_url_list = response.css('.item-click h3 a[target="_blank"]::attr(href)').extract()
        for job_url in job_url_list:
            if job_url in self.remove_url_list:
                print("Trùng lặp: ", job_url)
                continue
            else:
                yield scrapy.Request(job_url, callback = self.job_parse)

```

```

def job_parse(self, response):
    Web = 'Jobsgo'
    for i in range(len(response.css('div[class="content-group"]'))):
        if 'Ngành nghề' in response.css('div[class="content-group"]')[i].css('::text').extract():
            Nganh = response.css('div[class="content-group"]')[i].css('div a::text').get()
        if 'Yêu cầu công việc' in response.css('div[class="content-group"]')[i].css('::text').extract():
            YeuCau_List = response.css('div[class="content-group"]')[i].css('::text').extract()
        if 'Mô tả công việc' in response.css('div[class="content-group"]')[i].css('::text').extract():
            MoTa_List = response.css('div[class="content-group"]')[i].css('::text').extract()
        if 'Quyền lợi được hưởng' in response.css('div[class="content-group"]')[i].css('::text').extract():
            PhucLoi_List = response.css('div[class="content-group"]')[i].css('::text').extract()
    Link = response.url
    TenCV = response.css('div.media-body-2 h1::text').get()
    CongTy = response.css('div[class="panel-body"] div[class="media-body"] h2 a::text').get()
    try:
        TinhThanh = response.css('div[class="data giaphv"] p::text').extract()[0].replace("\n", "").strip().split(",")[-1].split("-")[-1].split("_")[-1].strip()
    except:
        TinhThanh = response.css('div[class="data giaphv"]::text').extract()[0].replace("\n", "").strip().split(",")[-1].split("-")[-1].split("_")[-1].strip()
    Luong = response.css('.salary::text').get()  # Thai Dương, 3 weeks ago • finaliy
    for i in range(len(response.css('div[class="col-sm-4 col-xs-6"]'))):
        if 'Tình chất công việc' in response.css('div[class="col-sm-4 col-xs-6"]')[i].css('::text').extract():
            try:
                LoaiHinh = response.css('div[class="col-sm-4 col-xs-6"]')[i].css('p')[1].css('::text').get().strip()
            except:
                LoaiHinh = response.css('div[class="col-sm-4 col-xs-6"]')[i].css('p')[1].css('::text').get().strip()
            if 'Yêu cầu kinh nghiệm' in response.css('div[class="col-sm-4 col-xs-6"]')[i].css('::text').extract():
                KinhNghiem = response.css('div[class="col-sm-4 col-xs-6"]')[i].css('p')[1].css('::text').extract()[0].strip()
            if 'Vị trí/đức vụ' in response.css('div[class="col-sm-4 col-xs-6"]')[i].css('::text').extract():
                CapBac = response.css('div[class="col-sm-4 col-xs-6"]')[i].css('p')[1].css('::text').extract()[0].strip()
    YeuCau =""
    for i in range(len(YeuCau_List)):
        YeuCau += YeuCau_List[i]
    MoTa = ""
    for i in range(len(MoTa_List)):
        MoTa += MoTa_List[i]
    PhucLoi = ""
    for i in range(len(PhucLoi_List)):
        PhucLoi += PhucLoi_List[i]
    SoLuong = '1'
    try:
        deadline = response.css('[class="deadline text-bold text-orange"]::text').get().strip()
        HanNopCV = date.today() + timedelta(days = int(deadline))
    except:
        HanNopCV = date.today()
    item = IT_Item()
    item['Web'] = Web
    item['Nganh'] = Nganh
    item['Link'] = Link
    item['TenCV'] = TenCV
    item['CongTy'] = CongTy
    item['TinhThanh'] = TinhThanh
    item['Luong'] = Luong
    item['LoaiHinh'] = LoaiHinh
    item['KinhNghiem'] = KinhNghiem
    item['CapBac'] = CapBac
    item['YeuCau'] = YeuCau
    item['MoTa'] = MoTa
    item['PhucLoi'] = PhucLoi
    item['HanNopCV'] = HanNopCV
    item['SoLuong'] = SoLuong
    yield item

```

Với trang này, vì số lượng tin khá nhiều nên ta tăng số request tại một thời điểm thành 40.

```
# Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 40  # Thai Dương, 3 weeks ago • finaliy
```

i) StudentJob

Với hai trường không có dữ liệu, ta để mặc định SoLuong là 1 và KinhNghiem là “Không có”.

Ý tưởng thuật toán là duyệt qua tất cả các trang, với mỗi trang ta lấy ra tất cả các url của công việc có trong trang đó, nếu url chưa có trong CSDL, ta sẽ gửi yêu cầu tới url đó và lấy dữ liệu, ngược lại sẽ bỏ qua url.

```

hai luong, 3 weeks ago | autor (hai luong)
from typing import Iterable
import scrapy
from scrapy.http import Request
import re
from StudentJob.items import IT_Item
from datetime import date
from StudentJob.pipelines import DatabaseConnector
import numpy as np      #Dùng unique để loại bỏ trùng lặp trong list url
Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class StudentSpider(scrapy.Spider):
    name = "student"
    allowed_domains = ["studentjob.vn"]

    def start_requests(self):
        # db_connector = DatabaseConnector(host='127.0.0.1', port = 3306, user='root', password='Camtruykich123', database='tuyendung_2')
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        yield scrapy.Request("https://studentjob.vn/viec-lam", callback = self.parse)

    def parse(self, response):
        job_count = response.css('.count-job span::text').get()

        so = re.search(r'\b\d+\b', job_count).group()
        if int(so) % 18 == 0:
            max_page = int(so) / 18
        else:
            max_page = int(so) // 18 + 1

        for page_number in range(1, int(max_page)+1):
            yield scrapy.Request(f"https://studentjob.vn/viec-lam?p={page_number}", callback = self.it_parse)

```

```

def it_parse(self, response):
    job_list_urls = response.css('.job-tittle.job-tittle2 a[target="_blank"]').css('::attr(href)').extract()
    for job_url in job_list_urls:
        if 'https://studentjob.vn' in job_url:
            next_url = job_url
        else:
            next_url = 'https://studentjob.vn' + job_url

        if next_url in self.remove_url_list:
            print("Trùng lặp: ", next_url)
            continue
        else:
            yield scrapy.Request(next_url, callback = self.it_parse_2)

```

```

def it_parse_2(self, response):
    Web = 'StudentJob'
    Link = response.url
    TenCV = response.css('.job-title::text').get().replace("\r\n", "").strip()
    CongTy = response.css('.company-name::text').get().replace("\r\n", "").strip()
    TinhThanh = response.css('.company-address::text').get().replace("\r\n", "").strip().split(",")[-1].split("-")[-1].split("_")[-1].strip()
    Luong = response.css('.salary p::text').get()
    for i in range(len(response.css('.summary-content'))):
        if 'Loại công việc' in response.css('.summary-content')[i].css('.content-label::text').get():
            LoaiHinh = response.css('.summary-content')[i].css('.content::text').get()
        if 'Ngành Nghề' in response.css('.summary-content')[i].css('.content-label::text').get():
            Nganh = response.css('.summary-content')[i].css('.content a::text').get()
        if 'Vị trí' in response.css('.summary-content')[i].css('.content-label::text').get():
            try:
                CapBac = response.css('.summary-content')[i].css('.content::text').get()
            except:
                CapBac = "Không có"
            KinhNghiem = "Không có"
        MoTa = ""
        MoTa_List = response.css('.job-description *::text').getall()
        for i in range(len(MoTa_List)):
            MoTa += MoTa_List[i]
        YeuCau = ""
        YeuCau_List = response.css('.job-experience *::text').getall()
        for i in range(len(YeuCau_List)):
            YeuCau += YeuCau_List[i]
        PhucLoi = ""
        PhucLoi_List = response.css('.job-benefits *::text').getall()
        for i in range(len(PhucLoi_List)):
            PhucLoi += PhucLoi_List[i]
        HanNopCV = response.css('div[class="d-flex expiry"] div')[0].css('::text').get().split(":")[-1].strip()
        SoLuong = "1" | Thái Dương, 4 weeks ago + Scrapy20231

    if HanNopCV == "":
        HanNopCV = date.today()
    item = IT_Item()
    item['Web'] = Web
    item['Nganh'] = Nganh
    item['Link'] = Link
    item['TenCV'] = TenCV
    item['CongTy'] = CongTy
    item['TinhThanh'] = TinhThanh
    item['Luong'] = Luong
    item['LoaiHinh'] = LoaiHinh
    item['KinhNghiem'] = KinhNghiem
    item['CapBac'] = CapBac
    item['YeuCau'] = YeuCau
    item['MoTa'] = MoTa
    item['PhucLoi'] = PhucLoi
    item['HanNopCV'] = HanNopCV
    item['SoLuong'] = SoLuong

    yield item

```

Vì số lượng tin là khá lớn, nên số lượng request tại 1 thời điểm sẽ được điều chỉnh để tăng tốc độ thu thập dữ liệu.

```
# Configure maximum concurrent requests performed by Scrapy (default: 16) | Thái Dương, last month + Scrapy20231
CONCURRENT_REQUESTS = 80
```

j) Techwork

Ý tưởng của thuật toán là duyệt qua các trang chứa danh sách tin tuyển dụng, sau đó trích xuất ra danh sách url công việc, kiểm tra xem 1 url đã tồn tại trong CSDL hay chưa, nếu chưa thì tiến hành gửi yêu cầu tới url đó để lấy dữ liệu, ngược lại bỏ qua url đó.

SoLuong không có nên được đặt mặc định là 1.

KinhNghiem và CapBac không có nên đặt mặc định là “Không có”.

```

from typing import Iterable
import scrapy
from scrapy.http import Request
from techwork.items import IT_Item
from techwork.pipelines import DatabaseConnector

Thái Dương, 4 weeks ago | 1 author (Thái Dương)
class TechworkSpider(scrapy.Spider):
    name = "techworkweb"
    allowed_domains = ["techworks.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        for page_number in range(1, 700):
            yield scrapy.Request("https://techworks.vn/viec-lam?p={page_number}", callback = self.parse)

    def parse(self, response):
        job_url_list = response.css('.job-tittle.job-tittle2 a[target="_blank"]::attr(href)').extract()
        for job_url in job_url_list:
            if 'https://techworks.vn' in job_url:
                next = job_url
            else:
                next = 'https://techworks.vn' + job_url

            if next in self.remove_url_list:
                print("Trùng lặp: ", next)
                continue
            else:
                yield scrapy.Request(next, callback = self.it_parse)

    def it_parse(self, response):
        Web = 'TechWorks'
        Nganh = 'IT'
        Link = response.url
        TenCV = response.css('.job-title::text').get().replace("\r\n", "").strip()
        CongTy = response.css('a[class="company-name"]::text').get().replace("\r\n", "").strip()
        try:
            TinhThanh = response.css('div[class="company-location"] a::text').get().split(",")[-1].replace("\r\n", "").strip()
        except:
            TinhThanh = response.css('.company-address::text').get().replace("\r\n", "").split(",")[-1].strip()
        except:
            TinhThanh = "Toàn quốc"
        Luong = response.css('[class="salary"] span::text').get()
        for i in range(len(response.css('[class="summary-content"]'))):
            if 'Loại công việc' in response.css('[class="summary-content"]')[i].css('.content-label::text').get():
                LoaiHinh = response.css('[class="summary-content"]')[i].css('.content::text').get()
        KinhNghiem = "Không có"
        CapBac = "Không có"

        YeuCau = ""
        YeuCau_List = response.css('.job-experience *::text').getall()
        for i in range(len(YeuCau_List)):
            YeuCau += YeuCau_List[i]

        MoTa = ""
        MoTa_List = response.css('.job-description *::text').getall()
        for i in range(len(MoTa_List)):
            MoTa += MoTa_List[i]

        PhuLoi = ""
        PhuLoi_List = response.css('.job-benefits *::text').getall()
        for i in range(len(PhuLoi_List)):
            PhuLoi += PhuLoi_List[i]

        HanNopCV = response.css('[class="expiry"]::text').get().split()[-1]
        SoLuong ="1"
        Thái Dương, 4 weeks ago * final

```

```

item = IT_Item()
item['Web'] = Web
item['Nganh'] = Nganh
item['Link'] = Link
item['TenCV'] = TenCV
item['CongTy'] = CongTy
item['TinhThanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhuLoi'] = PhuLoi
item['HanNopCV'] = HanNopCV
item['SoLuong'] = SoLuong

yield item

```

k) ViecLam24h

Ý tưởng của thuật toán là duyệt qua tất cả các trang tin tuyển dụng, với mỗi trang tin tuyển dụng ta trích ra danh sách các url của các công việc, với mỗi url công việc, ta kiểm tra xem url đó đã tồn tại trong CSDL chưa, nếu chưa thì thực hiện gửi yêu cầu tới url đó để lấy dữ liệu.

```

import scrapy
import math
from ViecLam24.items import ViecLam24Item
from ViecLam24.pipelines import DatabaseConnector

Thái Dương, 3 weeks ago | 1 author (Thái Dương)
class ViecLam24Spider(scrapy.Spider):
    name = "vieclam24"
    allowed_domains = ["vieclam24h.vn"]

    def start_requests(self):
        db_connector = DatabaseConnector(host='103.56.158.31', port = 3306, user='tuyendungUser', password='sinhvienBK', database='ThongTinTuyenDung')
        remove_url_list_local = db_connector.get_links_from_database()
        self.remove_url_list = remove_url_list_local
        print("Số lượng url trong CSDL: ", len(self.remove_url_list))
        url_get_job = "https://vieclam24h.vn/tim-kiem-viec-lam-nhanh?page=1"
        yield scrapy.Request(url_get_job, callback = self.parse)

    def parse(self, response):
        num_job = response.css("div[class='flex items-center'] span[class='font-semibold']::text").get()
        num_job = num_job.replace(",", "")
        num_job = int(num_job)
        if num_job % 30 == 0:
            num_page = num_job/30
        else:
            num_page = math.floor(num_job/30) + 1
        print(num_page)
        for page_number in range(1, int(num_page) + 1):
            # for page_number in range(1, 200):
            url_page = f"https://vieclam24h.vn/tim-kiem-viec-lam-nhanh?page={page_number}"
            yield scrapy.Request(url_page, callback = self.get_job_list)

    def get_job_list(self, response):
        job_list_url = response.css('div.relative a[class="relative lg:h-[115px] w-full flex rounded-sm border lg:mb-3 mb-2 lg:hover:shadow-md !bg-[#FFF5E7] border-se-bl"])
        for url_job in job_list_url:
            if "https://vieclam24h.vn" in url_job:
                url_job = url_job
            else:
                url_job = "https://vieclam24h.vn" + url_job

            if url_job in self.remove_url_list:
                print("Trùng lặp: ", url_job)
                continue
            else:
                yield scrapy.Request(url_job, callback = self.job_parse)

    def job_parse(self, response):
        ID = "VL24_+" + (response.url).split("-")[-1].replace(".html", "")
        Web = "Vieclam24"
        Link = response.url
        Nganhs_TG = response.css('a.jsx-d84db6a84feb175e::text').extract()
        Nganh = ''
        for Nganh_TG in Nganhs_TG:
            Nganh += Nganh_TG + ";"
        TenCV = response.css('h1.leading-snug::text').get()
        CongTy = response.css('div[class="md:ml-7 w-full"] a h3.mb-4::text').get()
        TinhThanh = response.css('div[class="md:ml-7 w-full"] div.flex.items-start a span::text').get()
        Luong = response.css('div[class="md:ml-7 w-full"] div.mt-5 div.ml-3')[0].css('p')[1].css('::text').get()
        HanLopCV = response.css('div[class="md:ml-7 w-full"] div.mt-5 div.ml-3')[1].css('p')[1].css('::text').get()
        #*****#
        col_1 = response.css('div[class="jsx-d84db6a84feb175e md:flex md:border-b border-[#DD06FE] mb-4"]')
        for i in range(len(col_1[0].css('div.ml-3'))):
            if col_1[0].css('div.ml-3')[i].css('p')[0].css('::text').get() == "Cấp bậc":
                CapBac = col_1[0].css('div.ml-3')[i].css('p')[1].css('::text').get()
            if col_1[1].css('div.ml-3')[i].css('p')[0].css('::text').get() == "Số lượng tuyển":
                SoLuong = col_1[1].css('div.ml-3')[i].css('p')[1].css('::text').get()
            if col_1[1].css('div.ml-3')[i].css('p')[0].css('::text').get() == "Hình thức làm việc":
                LoaiHinh = col_1[1].css('div.ml-3')[i].css('p')[1].css('::text').get()
        for i in range(len(col_1[2].css('div.ml-3'))):
            if col_1[2].css('div.ml-3')[i].css('p')[0].css('::text').get() == "Yêu cầu kinh nghiệm":
                KinhNghiem = col_1[2].css('div.ml-3')[i].css('p')[1].css('::text').get()
        MoTa = ''
        MoTas_TG = response.css('div[class="jsx-d84db6a84feb175e"]')[0].css('*:not(:empty)::text').getall()
        for MoTa_TG in MoTas_TG:
            MoTa += MoTa_TG
        YeuCau = ''
        YeuCaus_TG = response.css('div[class="jsx-d84db6a84feb175e mb-4 md:mb-8"] *:not(:empty)::text').getall()
        for YeuCau_TG in YeuCaus_TG:
            YeuCau += YeuCau_TG
        PhucLoi = ''
        PhucLois_TG = response.css('div[class="jsx-d84db6a84feb175e"]')[1].css('*:not(:empty)::text').getall()
        for PhucLoi_TG in PhucLois_TG:
            PhucLoi += PhucLoi_TG

```

```

item = ViecLam24Item()
item['ID'] = ID
item['Web'] = Web
item['Link'] = Link
item['Nganh'] = Nganh
item['TenCV'] = TenCV
    Thái Dương, last month • Scrapy20231
item['CongTy'] = CongTy
item['TinhThanh'] = TinhThanh
item['Luong'] = Luong
item['LoaiHinh'] = LoaiHinh
item['KinhNghiem'] = KinhNghiem
item['CapBac'] = CapBac
item['YeuCau'] = YeuCau
item['MoTa'] = MoTa
item['PhucLoi'] = PhucLoi
item['HanNopCV'] = HanNopCV
item['SoLuong'] = SoLuong

yield item

```

Tuy nhiên do trang web có quy định về số lượng yêu cầu được gửi tới tại cùng thời điểm, nên chương trình cần bổ sung thêm cài đặt như sau, nếu không sẽ rất dễ nhận lại mã lỗi 429 :

```

DOWNLOAD_DELAY = 1.5
AUTOTHROTTLE_ENABLED = True
AUTOTHROTTLE_START_DELAY = 6
AUTOTHROTTLE_TARGET_CONCURRENCY = 1.0
CONCURRENT_REQUESTS = 4
CONCURRENT_REQUESTS_PER_DOMAIN = 4

RETRY_TIMES = 5
    Thời gian chờ giữa các Lần retry
# Thời gian chờ giữa các Lần retry
RETRY_DELAY = 60 # Giây

```

Các cài đặt được thiết lập trong file setting.py, quy định mỗi thời điểm chỉ tồn tại 4 request và thời gian giữa các lần gửi request là 1.5s, tuy thời gian lấy dữ liệu sẽ lâu khi thực hiện lần đầu, tuy nhiên sẽ tránh được mã lỗi 429.

I) ITViec

Việc sử dụng scrapy đối với trang web này là khó khăn khi không sử dụng thêm dịch vụ rotating proxies mặc dù đã cài hai middleware giả mạo user-agent và middleware giả mạo browser-header. Việc nhận được mã lỗi 403 vẫn xảy ra thường xuyên khi thực hiện một yêu cầu tới bất kì trang web nào có tên miền itviec.com, tuy nhiên, các dịch vụ rotating proxies do bên thứ 3 cung cấp thường tính phí hoặc có giới hạn số lượng dùng thử. Vì vậy trong việc thu thập dữ liệu của ITViec, ta ưu tiên sử dụng selenium để tự động hóa và khiến cho trang web làm tưởng rằng “người dùng thực sự” đang sử dụng trình duyệt. Ngoài ra, ta sử dụng selenium khi mà số lượng tin cần lấy là không quá nhiều, trong trường hợp này là rất phù hợp.

Cũng cần lưu ý rằng khi sử dụng công cụ tự động hóa trình duyệt, cần có thời gian giãn cách giữa các lần thực hiện truy cập vào trang web, ở trong bài báo cáo được đặt là một giá trị nguyên ngẫu nhiên trong khoảng [1; 3]. Điều này nhằm tránh bị phát hiện là công cụ tự động và tránh dính Cloudflare.

Ý tưởng của thuật toán là duyệt qua toàn bộ các trang danh sách tin tuyển dụng và lấy toàn bộ các url của các công việc, lưu lại trong 1 danh sách, sau đó kiểm tra lần lượt từng url, nếu url đó không có trong CSDL thì giữ lại, ngược lại, xóa bỏ url đó ra khỏi danh sách. Sau cùng ta thu được một danh sách các url không có trong CSDL. Lần lượt gửi yêu cầu tới các url đó và lấy dữ liệu.

KinhNghiem, CapBac, Luong không có nên sẽ được để giá trị mặc định là “Không có”.

SoLuong không có nên sẽ được để giá trị mặc định là 1.

HanNopCV không có nên sẽ được để giá trị mặc định là ngày lấy dữ liệu.

```
from bs4 import BeautifulSoup
import requests
# Parsing and creating xml data
from lxml import etree as et

# Store data as a csv file written out
from csv import writer

# In general to use with timing our function calls to Indeed
import time

# Assist with creating incremental timing for our scraping to seem more human
from time import sleep

# Dataframe stuff
import pandas as pd

# Random integer for more realistic timing for clicks, buttons and searches during scraping
from random import randint

# Multi Threading
import threading

# Threading:
from concurrent.futures import ThreadPoolExecutor, wait
import math
import mysql.connector
from datetime import date
```

```
import selenium
# Check version I am running
selenium.__version__ = '4.15.2'

# from selenium import webdriver
# Starting/Stopping Driver: can specify ports or location but not remote access
# from selenium.webdriver.chrome.service import ChromeService
# Manages Binaries needed for WebDriver without installing anything directly
# from webdriver_manager.chrome import ChromeDriverManager
```

```
# Allows searches similar to beautiful soup: find_all
from selenium.webdriver.common.by import By

# Try to establish wait times for the page to Load
from selenium.webdriver.support.ui import WebDriverWait

# Wait for specific condition based on defined task: web elements, boolean are examples
from selenium.webdriver.support import expected_conditions as EC

# Used for keyboard movements, up/down, Left/right, delete, etc
from selenium.webdriver.common.keys import Keys

# Locate elements on page and throw error if they do not exist
from selenium.common.exceptions import NoSuchElementException
```

```

response = requests.get(
    url='https://headers.scrapeops.io/v1/browser-headers',
    params={
        'api_key': 'cc44ced0-490d-41a0-b258-46f2ad7e74b3',
        'num_results': '100'
    }
)
header_browser_list = response.json()
print(header_browser_list['result'][0])

{'upgrade-insecure-requests': '1', 'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36 Edg/89.0.774.76', 'accept': 'text/html'}
```

```

response = requests.get(
    url='https://headers.scrapeops.io/v1/user-agents',
    params={
        'api_key': 'cc44ced0-490d-41a0-b258-46f2ad7e74b3',
        'num_results': '100'
    }
)
user_agent_list = response.json()
print(user_agent_list['result'][0])

Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36
```

```

random_index_user_agent = randint(0, len(user_agent_list)-1)
random_index_header_browser = randint(0, len(header_browser_list)-1)
user_agent_random = user_agent_list['result'][random_index_user_agent]
header_browser_random = header_browser_list['result'][random_index_header_browser]
```

```

# Allows you to customize: incognito mode, maximize window size, headless browser, disable certain features, etc
option = webdriver.ChromeOptions()

# Going undercover:
option.add_argument("--incognito")

# # Consider this if the application works and you know how it works for speed ups and rendering!
option.add_argument('--headless=chrome') #Sử dụng trình duyệt không có giao diện người dùng
user_agent = user_agent_random
option.add_argument(f"user-agent={user_agent}")

# Thêm header vào Options
headers = header_browser_random

for key, value in headers.items():
    option.add_argument(f"--header={key}:{value}")
```

```

driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=option)

driver.get("https://itviec.com/it-jobs")

job_count = driver.find_element(By.CLASS_NAME,'headline-total-jobs').text
number_of_jobs = job_count.split()[0]
if (int(number_of_jobs) % 20 == 0):
    max_page = int(number_of_jobs) / 20
else:
    max_page = math.floor(int(number_of_jobs) / 20) + 1
print(max_page)
driver.quit()
job_uris = [] ##Lưu trữ tất cả url Lấy được từ web
```

```

page_url ="https://itviec.com/it-jobs?page={}"
for page_number in range(1, int(max_page + 1)):
    # for page number in range(1, 3):
    random_index_user_agent = randint(0, len(user_agent_list)-1)
    random_index_header_browser = randint(0, len(header_browser_list)-1)
    user_agent_random = user_agent_list['result'][random_index_user_agent]
    header_browser_random = header_browser_list['result'][random_index_header_browser]
    user_agent = user_agent_random

    option.add_argument(f"user-agent={user_agent}")

    # Thêm header vào Options
    headers = header_browser_random

    for key, value in headers.items():
        option.add_argument(f"--header={key}:{value}")

    driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=option)
    driver.get(page_url.format(page_number))
    sleep(randint(3, 5))
    jobs = driver.find_elements(By.CLASS_NAME,"job-card")
    for job in jobs:
        job_url = job.find_element(By.CSS_SELECTOR,'a').get_attribute("href")
        job_uris.append(job_url)
    # sleep(randint(6, 10))
    driver.quit()
```

32

```
conn = mysql.connector.connect(
    host='103.56.158.31',
    port= 3306,
    user= "tuyendungUser",
    password="sinhvienBK",
    database= 'ThongTinTuyenDung'
)
cursor = conn.cursor()

sql = 'INSERT IGNORE INTO Stg_ThongTin_raw(Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, SoLuong) VALUES (%s, %s, %s)'

sql_link = 'SELECT Link FROM Stg_ThongTin_raw where Web =\'ITViec\''

cursor.execute(sql_link)
result = cursor.fetchall()
remove_url_list = [row[0] for row in result]
print("Số url cần xóa là: ", len(job_urls))
for job_url in remove_url_list:
    if job_url in job_urls:
        job_urls.remove(job_url)
print("Đã lấy link thành công")
print("Số lượng link mới sau xử lý lấy được: ", len(job_urls))

url cần xóa: 625
lấy link thành công
Số lượng link mới sau xử lý lấy được: 29

if len(job_urls) > 0:
    for i in range(len(job_urls)):
        driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=options)
        driver.get(job_urls[i])
        print("Đang xử lý...", job_urls[i])
        sleep(randint(1, 3))
        Web = 'ITViec'
        Nganh = 'IT'
        Link = job_urls[i]
        TenCV = driver.find_element(By.CLASS_NAME, 'ipt-md-6').text
        CongTy = driver.find_element(By.CLASS_NAME, 'employer-name').text
        TinhThanh = driver.find_elements(By.CSS_SELECTOR, '[class="normal-text text-rich-grey"]')[0].text
        Luong = 'Không có'
        LoaiHinh = driver.find_element(By.CSS_SELECTOR, '[class="normal-text text-rich-grey ms-1"]').text
        KinhNghiem = 'Không có'
        CapBac = 'Không có'
        CapBac = 'Không có'
        HanNopCV = date.today()
        try:
            YeuCau = driver.find_elements(By.CLASS_NAME, 'imy-5')[1].text
        except:
            YeuCau = ""
        try:
            MoTa = driver.find_elements(By.CLASS_NAME, 'imy-5')[0].text
        except:
            MoTa = ""
        try:
            PhucLoi = driver.find_elements(By.CLASS_NAME, 'imy-5')[2].text
        except:
            PhucLoi = ""
        SoLuong = '1'
        cursor.execute(sql, (Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, SoLuong))
        conn.commit()
        driver.quit()
    cursor.close()

conn.close()
print("Đã thêm tin thành công")
else:
    print("Không có tin mới để thêm")

xử lý.... https://itviec.com/it-jobs/senior-php-engineer-cw-positive-thinking-company-b-o-t-4206?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-embedded-software-engineer-bosch-global-software-technologies-company-limited-0140?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/automotive-embedded-software-architect-bosch-global-software-technologies-company-limited-5923?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-project-manager-embedded-software-bosch-global-software-technologies-company-limited-5544?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/engineering-manager-nab-innovation-centre-vietnam-5146?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-data-analyst-one-mount-group-4534?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/vcs-backend-developer-java-golang-python-vlettel-group-4146?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/manual-test-specialist-qa-pc-english-netcompany-1011?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-product-owner-product-manage-got-it-0435?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/product-software-business-analyst-agile-bitech-3717?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/solution-architect-nakka-digital-vietnam-co-ltd-3306?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-cloud-engineer-aws-azure-nab-innovation-centre-vietnam-1429?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/devops-engineer-aws-imip-technology-and-solutions-consultancy-5431?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-ui-ux-designer-spiraledge-3733?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/senior-automation-qc-engineer-java-selenium-restapp-house-of-norway-5446?lab_feature=preview_id_page
.
xử lý.... https://itviec.com/it-jobs/c-gt-developer-for-ide-development-revolution-engineering-vietnam-3424?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/drupal-developer-athena-1302?lab_feature=preview_id_page
xử lý.... https://itviec.com/it-jobs/front-end-developer-reactjs-html5-javascript-ideologic-4017?lab_feature=preview_id_page

thêm tin thành công
Output was truncated. View as a scrollable element or open in a text editor. Adjust cell output settings..
```

m) Job123 (Chỉ lấy IT)

Ý tưởng thuật toán là duyệt qua các trang chứa danh sách tin tuyển dụng, sau đó lấy ra tất cả các url tin tuyển dụng và lưu vào 1 danh sách. Với mỗi url đã có trong CSDL, ta loại url đó ra khỏi danh sách. Với mỗi url có trong danh sách lúc sau, ta tiến hành tự động trình duyệt và lấy về dữ liệu qua trích xuất mã html trả về.

```

from bs4 import BeautifulSoup
import requests
# Parsing and creating xml data
from lxml import etree as et

# Store data as a csv file written out
from csv import writer

# In general to use with timing our function calls to Indeed
import time

# Assist with creating incremental timing for our scraping to seem more human
from time import sleep

# Dataframe stuff
import pandas as pd

# Random integer for more realistic timing for clicks, buttons and searches during scraping
from random import randint

# Multi Threading
import threading

# Threading:
from concurrent.futures import ThreadPoolExecutor, wait
import math
import mysql.connector

import selenium
# Check version I am running
selenium.__version__
'4.15.2'

```

```

from selenium import webdriver

# Starting/Stopping Driver: can specify ports or location but not remote access
from selenium.webdriver.chrome.service import Service as ChromeService

# Manages Binaries needed for WebDriver without installing anything directly
from webdriver_manager.chrome import ChromeDriverManager

# Allows search similar to beautiful soup: find_all
from selenium.webdriver.common.by import By

# Try to establish wait times for the page to load
from selenium.webdriver.support.ui import WebDriverWait

# Wait for specific condition based on defined task: web elements, boolean are examples
from selenium.webdriver.support import expected_conditions as EC

# Used for keyboard movements, up/down, left/right, delete, etc
from selenium.webdriver.common.keys import Keys

# Locate elements on page and throw error if they do not exist
from selenium.common.exceptions import NoSuchElementException

```

```
response = requests.get(  
    url='https://headers.scrapeops.io/v1/browser-headers',  
    params={  
        'api_key': 'ccaa4ced0-490d-41a0-b258-46f2ad7e7db3',  
        'num_results': '100'  
    })  
header_browser_list = response.json()  
print(header_browser_list['result'][0])  
  
{'upgrade-insecure-requests': '1', 'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36', 'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8'}  
  
response = requests.get(  
    url='https://headers.scrapeops.io/v1/user-agents',  
    params={  
        'api_key': 'ccaa4ced0-490d-41a0-b258-46f2ad7e7db3',  
        'num_results': '100'  
    })  
user_agent_list = response.json()  
print(user_agent_list['result'][0])  
  
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.2 Safari/605.1.15  
  
random_index_user_agent = randint(0, len(user_agent_list)-1)  
random_index_header_browser = randint(0, len(header_browser_list)-1)  
user_agent_random = user_agent_list['result'][random_index_user_agent]  
header_browser_random = header_browser_list['result'][random_index_header_browser]
```

```
# Allows you to customize: incognito mode, maximize window size, headless browser, disable certain features, etc
option = webdriver.ChromeOptions()

# Going undercover:
option.add_argument("--incognito")

# # Consider this if the application works and you know how it works for speed ups and rendering!
option.add_argument(['-headless=chrome'])
user_agent = user_agent_random
option.add_argument(f"user-agent={user_agent}")

#Thêm header vào Options
headers = header_browser_random

for key, value in headers.items():
    option.add_argument(f"--header-{key}:{value}")

#
```

```
conn = mysql.connector.connect(  
    host='103.56.158.31',  
    port = 3306,  
    user= 'tuyendungUser',  
    password='sinhvienBK',  
    database= 'ThongTinTuyenDung'  
)  
cursor = conn.cursor()  
9] Python  
  


```
sql = 'INSERT IGNORE INTO Stg_ThongTin_raw(Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, SoLuong) VALUES (%s, %s, %s)'
10] Python


```
sql_link = 'SELECT Link FROM Stg_ThongTin_raw where Web =\'123jobs\''  
11] Python  
  


```
cursor.execute(sql_link)
result = cursor.fetchall()
remove_url_list = [row[0] for row in result]
12] Python
```


```


```


```

```

def extract(url):
    driver.get(url)
    Web = '123Jobs'
    Nganh = 'IT'
    Link = url
    TenCV = driver.find_element(By.CSS_SELECTOR,'h1.job-title strong').text
    try:
        CongTy = driver.find_element(By.CSS_SELECTOR,['class="col-md-9 content-group box-apply-top js-item-job"] p').text
    except:
        CongTy = ""
    TinhThanh = driver.find_elements(By.CSS_SELECTOR,['class="item text-black"]')[0].text
    Luong = driver.find_element(By.CSS_SELECTOR,['class = "item salary"]').text
    LoaiHinh = driver.find_elements(By.CSS_SELECTOR,['class="item text-black"]')[1].text
    KinhNghiem = driver.find_elements(By.CSS_SELECTOR,['class="item text-black"]')[2].text
    CapBac = driver.find_elements(By.CSS_SELECTOR,['class="item time-expiry-date"]')[1].text
    HanNopCV = driver.find_elements(By.CSS_SELECTOR,['class="item time-expiry-date"]')[0].text
    try:
        YeuCau = driver.find_element(By.CSS_SELECTOR, '[class="collslap"]').find_elements(By.CSS_SELECTOR,['class="content-group"]')[1].text
    except:
        YeuCau = ""
    try:
        MoTa = driver.find_element(By.CSS_SELECTOR, '[class="collslap"]').find_elements(By.CSS_SELECTOR,['class="content-group"]')[0].text
    except:
        MoTa = ""
    try:
        PhucLoi = driver.find_element(By.CSS_SELECTOR, '[class="collslap"]').find_elements(By.CSS_SELECTOR,['class="content-group"]')[2].text
    except:
        PhucLoi = ""
    SoLuong = driver.find_elements(By.CSS_SELECTOR,['class="item text-black'])[3].text
    cursor.execute(sql, (Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, SoLuong))
    conn.commit()

```

Python

```

job_urls = []
def it(url_first):
    driver.get(url_first)
    job_page_url_list = driver.find_elements(By.CSS_SELECTOR, '[class="job_list-item-title"]')
    for url in job_page_url_list:
        link = url.find_element(By.CSS_SELECTOR, 'a').get_attribute('href')
        job_urls.append(link)
    next = driver.find_element(By.CSS_SELECTOR, '[rel="Next"]').get_attribute("href")
    if "page=11" in next:
        next = None
    if next is not None:
        it(next)

```

Python

```

driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=options)
it("https://123job.vn/nganh-nghe/vi%1BBB%87c-1%CC3%A0m-it-ph%E1%BA%A7n-m%E1%BB%81m?cat=It+ph%E1%BA%A7n+m%E1%BB%81m&sort=new&cat_name=IT+ph%E1%BA%A7n+m%E1%BB%81m&page=1")
driver.quit()
print("Số url cần vẽ: ", len(job_urls))
for job_url in remove_url_list:
    if job_url in job_urls:
        job_urls.remove(job_url)
print("Số lượng url mới đã lấy được: ", len(job_urls), " url")
driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=options)
if len(job_urls) > 0:
    for i in range(len(job_urls)):
        print("Đang xử lý: ", job_urls[i])
        extract(job_urls[i])
        print("Complete")
else:
    print("Không có tin mới")
driver.quit()
cursor.close()
conn.close()

```

Python

```

Số url cần vẽ:  270
Số lượng url mới đã lấy được: 187 url
Đang xử lý: https://123job.vn/viec-lam/it-manager-BpP4MkVr0z
Complete
Đang xử lý: https://123job.vn/viec-lam/ngi-project-leader-v91BwXyOrw
Complete
Đang xử lý: https://123job.vn/viec-lam/mobile-android-developer-java-kotlin-o9jyJx8zDR
Complete
Đang xử lý: https://123job.vn/viec-lam/senior-cloud-engineer-azureaws-0q7zyXvq8
Complete

```

n) TopCV (Chỉ lấy IT)

Việc sử dụng scrapy ở trang web này là không tối ưu. Lý do là mặc dù đã sử dụng cả 2 middleware giả mạo user-agent và giả mạo browser-header. Tuy nhiên tỉ lệ nhận được mã 403 trả về rất cao, và số lượng tin lấy được rất ít, ngoài ra các dịch vụ rotating proxies do bên thứ 3 cung cấp đều tính phí hoặc giới hạn lượng dùng thử.

Mặt khác, trang web trả về các công việc đều ở dưới dạng html, vì vậy đây là điều kiện thuận lợi để sử dụng selenium, mặc dù thời gian chạy lần đầu sẽ hơi lâu, nhưng các lần tiếp theo sẽ nhanh hơn do tần suất cập nhật dữ liệu của TopCV cũng không phải là quá nhiều trong 1 tuần.

Ý tưởng của thuật toán là duyệt qua toàn bộ các trang danh sách tin, trích xuất ra danh sách tất cả url của các tin tuyển dụng và lưu vào 1 list. Sau đó lại kiểm tra xem với mỗi url trong đó, đã tồn tại trong CSDL hay chưa, nếu đã tồn tại, tiến hành loại bỏ nó ra khỏi list.

Danh sách sau cùng thu được gồm toàn các url mới không có trong CSDL, ta tự động trình duyệt và trích xuất dữ liệu từ mã html trả về khi truy cập vào các url này.

```

from bs4 import BeautifulSoup
import requests
# Parsing and creating xml data
from lxml import etree as et

# Store data as a csv file written out
from csv import writer

# In general to use with timing our function calls to Indeed
import time

# Assist with creating incremental timing for our scraping to seem more human
from time import sleep

# Dataframe stuff
import pandas as pd

# Random integer for more realistic timing for clicks, buttons and searches during scraping
from random import randint

# Multi Threading
import threading

# Threading:
from concurrent.futures import ThreadPoolExecutor, wait
import math
import mysql.connector
from datetime import date
import json
1
import selenium
# Check version I am running
selenium.__version__
1
'4.15.2'
1

```

```

from selenium import webdriver
# Starting/Stopping Driver: can specify ports or location but not remote access
from selenium.webdriver.chrome.service import Service as ChromeService
# Manages Binaries needed for WebDriver without installing anything directly
from webdriver_manager.chrome import ChromeDriverManager

```

Python

```

# Allows searches similar to beautiful soup: find_all
from selenium.webdriver.common.by import By

# Try to establish wait times for the page to load
from selenium.webdriver.support.ui import WebDriverWait

# Wait for specific condition based on defined task: web elements, boolean are examples
from selenium.webdriver.support import expected_conditions as EC

# Used for keyboard movements, up/down, Left/right, delete, etc
from selenium.webdriver.common.keys import Keys

# Locate elements on page and throw error if they do not exist
from selenium.common.exceptions import NoSuchElementException

```

Python

```

response = requests.get(
    url='https://headers.scrapeops.io/v1/browser-headers',
    params={
        'api_key': 'cca4ced0-490d-41a0-b258-46f2ad7e74b3',
        'num_results': '100'
    }
)
header_browser_list = response.json()
print(header_browser_list['result'][0])

```

[5] Python

```

... {'upgrade-insecure-requests': '1', 'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.67 Safari/537.36 Edg/87.0.664.52', 'accept': 'text/html', 'accept-encoding': 'gzip, deflate, br', 'accept-language': 'en-US,en;q=0.9', 'cache-control': 'max-age=0', 'sec-fetch-mode': 'navigate', 'sec-fetch-user': '?1', 'upgrade-insecure-requests': '1', 'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36 Edg/90.0.818.56'}

```

```

response = requests.get(
    url='https://headers.scrapeops.io/v1/user-agents',
    params={
        'api_key': 'ccaa4ced0-490d-41a0-b258-46f2ad7e74b3',
        'num_results': '100'
    }
)
user_agent_list = response.json()
print(user_agent_list['result'][0])

```

[6] Python

```

... Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36 Edg/90.0.818.56

```

```

random_index_user_agent = randint(0, len(user_agent_list)-1)
random_index_header_browser = randint(0, len(header_browser_list)-1)
user_agent_random = user_agent_list['result'][random_index_user_agent]
header_browser_random = header_browser_list['result'][random_index_header_browser]

```

[7] Python

```

# Allows you to customize: incognito mode, maximize window size, headless browser, disable certain features, etc
option = webdriver.ChromeOptions()

# Going undercover:
option.add_argument("--incognito")

# # Consider this if the application works and you know how it works for speed ups and rendering!
option.add_argument('--headless=chrome')
user_agent = user_agent_random
option.add_argument(f"user-agent={user_agent}")

# Thêm header vào Options
headers = header_browser_random

for key, value in headers.items():
    option.add_argument(f"--header={key}:{value}")

```

[8] Python

```

driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),options=option)
driver.implicitly_wait(10) # Set the implicit wait time to 10 seconds
driver.set_page_load_timeout(30)
driver.get("https://www.topcv.vn/viec-lam-it")

job_count = driver.find_element(By.CSS_SELECTOR,'[class="job-header"] h2 b').text.replace(",","")
if (int(job_count) % 50 == 0):
    max_page = int(job_count) / 50
else:
    max_page = math.floor(int(job_count) / 50) + 1
print(max_page)
driver.quit()

```

[9] Python

46

```

job_urls = []
driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), options=options)
for page_number in range(1, int(max_page+1)):
    driver.get(f"https://www.topcv.vn/viec-lam-it?page={page_number}")
    job_url_list = driver.find_elements(By.CSS_SELECTOR, '.job-item-2 h3[class="title"] a')
    for uu in job_url_list:
        link = uu.get_attribute("href")
        job_uris.append(link)
driver.quit()

```

8] Python


```

conn = mysql.connector.connect(
    host='103.56.158.31',
    port = 3306,
    user= 'tuyendungUser',
    password='sinhvienBK',
    database= 'ThongTinTuyenDung'
)
cursor = conn.cursor()

sql = 'INSERT IGNORE INTO Stg_ThongTin_raw(Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, Soluong) VALUES (%s, %s, %s)'

cursor.execute(sql_link)
result = cursor.fetchall()
remove_url_list = [row[0] for row in result]

```

1] Python


```

sql_link = "SELECT Link FROM Stg_ThongTin_raw where Web =\`TopCV\`"
cursor.execute(sql_link)
result = cursor.fetchall()
remove_url_list = [row[0] for row in result]

```

2] Python


```

cursor.execute(sql_link)
result = cursor.fetchall()
remove_url_list = [row[0] for row in result]

print("Số lượng url cần xóa: ", len(job_uris))
for job_url in remove_url_list:
    if job_url in job_uris:
        job_uris.remove(job_url)
print("Số lượng url mới lấy được sau xử lý: ", len(job_uris))

Số lượng url cần xóa:  2266
Số lượng url mới lấy được sau xử lý: 159

```

3] Python


```

driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()), options=options)
if len(job_uris) > 0:
    for job_url in job_uris:
        if "https://www.topcv.vn/brand/" in job_url:
            continue
        link = job_url
        print("Đang xử lý....", link)
        driver.get(link)
        Web = 'TopCV'
        Nganh = 'IT'
        Link = link
        TenCV = driver.find_element(By.CSS_SELECTOR, '.job-detail__info--title').text
        CongTy = driver.find_element(By.CSS_SELECTOR, '[class="company-name-label"] a').text
        Soluong = ""
        YeuCau = ""
        MoTa = ""
        PhucLoi = ""
        for i in range(len(driver.find_elements(By.CSS_SELECTOR, '[class="job-detail__info--section"]'))):
            if "Mức lương" in driver.find_elements(By.CSS_SELECTOR, '[class="job-detail__info--section"]')[i].find_element(By.CSS_SELECTOR, '[class="job-detail__info--section-content-title"]').text:
                Luong = driver.find_elements(By.CSS_SELECTOR, '[class="job-detail__info--section"]')[i].find_element(By.CSS_SELECTOR, '[class="job-detail__info--section-content-value"]').text
            if "Địa điểm" in driver.find_elements(By.CSS_SELECTOR, '[class="job-detail__info--section"]')[i].find_element(By.CSS_SELECTOR, '[class="job-detail__info--section-content-title"]').text:
                TinhThanh = driver.find_elements(By.CSS_SELECTOR, '[class="job-detail__info--section"]')[i].find_element(By.CSS_SELECTOR, '[class="job-detail__info--section-content-value"]').text
            try:
                HanNopCV = driver.find_element(By.CSS_SELECTOR, '[class="job-detail__info--deadline"]').text.split(":")[-1].strip()
            except:
                HanNopCV = date.today()
            soup = BeautifulSoup(driver.page_source, 'html.parser')
            List = soup.find_all('div', class_='job-description__item')
            YeuCau = List[1].text
            MoTa = List[0].text
            PhucLoi = List[2].text
            for i in range(len(soup.find_all('div', class_='box-general-group'))):
                if "Số lượng tuyển" in soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-title').text:
                    Soluong = soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-value').text.split()[0]
                if "Hình thức làm việc" in soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-title').text:
                    LoaiHinh = soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-value').text
                try:
                    if "Kinh nghiệm" in soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-title').text:
                        KinhNghiem = soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-value').text
                    if "Cap bắc" in soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-title').text:
                        CapBac = soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-value').text
                except:
                    KinhNghiem = "Không có"
                if "Cap bắc" in soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-title').text:
                    CapBac = soup.find_all('div', class_='box-general-group')[i].find('div', class_='box-general-group-info-value').text
cursor.execute(sql, (Web, Nganh, Link, TenCV, CongTy, TinhThanh, Luong, LoaiHinh, KinhNghiem, CapBac, HanNopCV, YeuCau, MoTa, PhucLoi, Soluong))
conn.commit()

```

4] Python


```

else:
    print("Không có tin mới để thêm.")
    driver.quit()
    cursor.close()
    conn.close()

```

5] Python

```
Dang xu li.... https://www.toncv.vn/viec-lam/ky-su-cau-noi-tieng-han-bien-phien-dich-tieng-han-nganh-it-korean-brse/1132735.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/fx-artist-junior/1168046.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/lap-trinh-vien-software-developers-muc-luong-den-35k/1154432.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/tester-mobile-app/1016516.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/chanh-ky-thuat-may-tinh/1206111.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/nhan-vien-3d-artist-game-mobile/257687.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/nhan-vien-3d-artist-game-mobile/257687.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/nhan-vien-kiem-thu-phan-mem-mang-chung-khoan/1206536.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/php-laravel-from-mid/1206435.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/seo-website/1205174.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/senior-seo-specialist/1206182.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/business-analyist/1206318.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/senior-php-developer/1184668.html?ta_source=ITJobs_LinkDetail
...
Dang xu li.... https://www.toncv.vn/viec-lam/full-stack-developer-nodejs-reactjs/1043437.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/ux-ui-designer/1200116.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/nhan-vien-thiet-ke/1190590.html?ta_source=ITJobs_LinkDetail
Dang xu li.... https://www.toncv.vn/viec-lam/nhan-vien-seo-marketing/1190601.html?ta_source=ITJobs_LinkDetail
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings.
```

o) VietNamWork

Ý tưởng thuật toán là ta sẽ tiến hành dùng scrapy để gửi 1 request với các tham số tương tự cho server để lấy file json mà server trả về. Rồi sau đó từ trong file json trả về, ta sẽ tính trang danh sách việc cuối cùng cần lấy dữ liệu và thực hiện một vòng lặp for qua tất cả các trang để lấy dữ liệu. Với mỗi lần lặp ta sẽ lấy được về số lượng tin là 50 tin trong 1 file json.

LoaiHinh và KinhNghiem sẽ mặc định là “Không có”, SoLuong mặc định là 1.

```
Thái Dương, 4 weeks ago | 1 author (Thái Dương)
import scrapy
import json
from VN_Work.items import VNWorkItem

Thái Dương, 4 weeks ago | 1 author (Thái Dương)
class VietnamworkSpider(scrapy.Spider):
    name = "vietnamwork"
    url = "https://ms.vietnamworks.com/job-search/v1.0/search"
    headers = {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate, br",
        "Accept-Language": "en-US,en;q=0.9,vi;q=0.8",
        "Connection": "keep-alive",
        "Content-Type": "application/json",
        "Host": "ms.vietnamworks.com",
        "Origin": "https://www.vietnamworks.com",
        "Referer": "https://www.vietnamworks.com/",
        "X-Source": "Page-Container"
    }

    payload = {
        "userId": 0,
        "query": "",
        "filter": [],
        "ranges": [],
        "order": [],
        "hitsPerPage": 50,
        "retrieveFields": [
            "address",
            "benefits",
            "jobTitle",
            "salaryMax",
            "isSalaryVisible",
            "jobLevelId",
            "isShowLogo",
            "salaryMin",
            "companyLogo",
            "userId",
            "jobLevel",
            "jobLevelId",
            "jobId",
            "companyId",
            "approvedOn",
            "isAnonymous",
            "alias",
            "expiredOn",
            "industries",
            "workingLocations",
            "services",
            "companyName",
            "salary",
            "onlineOn",
            "simpleServices",
            "visibilityDisplay",
            "isShowLogoInSearch",
            "priorityOrder",
            "skills",
            "profilePublishedSiteMask",
            "jobDescription",
            "jobRequirement",
            "prettySalary",
            "requiredCoverLetter",
            "languageSelectedVI",
            "languageSelected",
            "languageSelectedId"
        ]
    }
```

```

def start_requests(self):
    payload = self.payload
    payload['page'] = 2
    yield scrapy.Request(
        self.url,
        method='POST',
        body=json.dumps(payload),
        headers= self.headers,
        callback=self.page_count
    )
def page_count(self, response):
    job_count = int(response.json()['meta']['nbHits'])
    if job_count % 50 == 0:
        max_page = job_count / 50
    else:
        max_page = job_count // 50 + 1
    print(max_page)
    for page_number in range(0, max_page):
        payload = self.payload
        payload['page'] = page_number

        yield scrapy.Request(
            self.url,
            method='POST',
            body=json.dumps(payload),
            headers= self.headers,
            callback=self.parse
        )
def parse(self, response):
    json_souce = response.json()
    for i in range(len(json_souce["data"])):
        ID = "VNW_" + str(json_souce["data"][i]["jobId"])
        Web = "VietnamWork"
        Nganh = json_souce["data"][i]["jobFunctionsV3"]["jobFunctionV3NameVI"]
        Link = "https://www.vietnamworks.com/" + json_souce["data"][i]["alias"] + "-" + str(json_souce["data"][i]["jobId"]) + "-jv"
        TenCV = json_souce["data"][i]["jobTitle"]
        CongTy = json_souce["data"][i]["companyName"]
        TinhThanh = json_souce["data"][i]["workingLocations"][0]["cityNameVI"]
        Luong = json_souce["data"][i]["prettySalary"]
        LoaiHinh = "Không có"
        KinhNghiem = "Không có"
        CapBac= json_souce["data"][i]["jobLevelVI"]
        HanKopCV = json_souce["data"][i]["expiredOn"]
        YeuCau = json_souce["data"][i]["jobRequirement"]
        MoTa = json_souce["data"][i]["jobDescription"]
        Phucloi = ""
        for j in range(len(json_souce["data"][i]["benefits"])):
            Phucloi += json_souce["data"][i]["benefits"][j]["benefitValue"]
        SoLuong = "1" | Thai Duong, last month * Scrapy2023
        item = VnWorkItem()
        item["ID"] = ID
        item["Web"] = Web
        item["Nganh"] = Nganh
        item["Link"] = Link
        item["TenCV"] = TenCV
        item["CongTy"] = CongTy
        item["TinhThanh"] = TinhThanh
        item["Luong"] = Luong
        item["LoaiHinh"] = LoaiHinh
        item["KinhNghiem"] = KinhNghiem
        item["CapBac"] = CapBac
        item["HanKopCV"] = HanKopCV
        item["YeuCau"] = YeuCau
        item["MoTa"] = MoTa
        item["Phucloi"] = Phucloi
        item["SoLuong"] = SoLuong
        yield item

```

3. Cách chạy chương trình đóng gói

Mở thư mục chứa code bằng Visual Studio Code, tiến hành cài thư viện pyautogui bằng terminal như sau :

```

import pyautogui as pag
import os
import time

current = os.getcwd().replace("\\", "/") #lấy vị trí hiện tại
print(current)

pag.hotkey('shift', 'ctrl', '')
time.sleep(1) #Thời gian trả về chờ terminal đã mở trước khi nhập
pag.typewrite("pip install bs4\n"
            "pip install requests\n"
            "pip install selenium\n"
            "pip install pandas\n"
            "pip install mysql-connector-python\n"
            "pip install numpy\n"
            "pip install selenium\n"
            "pip install webdriver_manager")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER COMMENTS GITLENS

PS E:\Scrapy_DA2> pip install pyautogui

Hình 87. Cài đặt pyautogui

Bước tiếp theo mở file setting_before.ipynb, chọn “Select kernel”, và chọn phiên bản Python tương ứng của máy, nếu máy chưa cài Jupyter notebook, chương trình sẽ đưa ra gợi ý, hãy cài đặt đầy đủ Jupyter notebook. Sau đó TẮT PHẦN MỀM GÕ TIẾNG VIỆT và chạy file setting_before.ipynb bằng cách nhấn vào “Run all” như hình dưới :

```

import pyautogui as pag
import os
import time

current = os.getcwd().replace("\\", "/") #lấy vị trí hiện tại
print(current)

pag.hotkey('shift', 'ctrl', '')
time.sleep(1) #Thời gian trả về chờ terminal đã mở trước khi nhập
pag.typewrite("pip install bs4\n"
            "pip install requests\n"
            "pip install selenium\n"
            "pip install pandas\n"
            "pip install mysql-connector-python\n"
            "pip install numpy\n"
            "pip install selenium\n"
            "pip install webdriver_manager")
pag.press('enter', interval=0.1) #cài đặt các thư viện cần thiết ngoài da

pag.hotkey('shift', 'ctrl', '')
time.sleep(1) #Thời gian trả về chờ terminal đã mở trước khi nhập
pag.typewrite("python -m venv venv\n"
            "f'(current)/venv/scripts/activate.ps1\n")
pag.typewrite("pip install scrapy\n"
            "pip install numpy\n"
            "pip install requests\n"
            "pip install mysql-connector-python\n"
            "pip install selenium\n"
            "pip install typing\n"
            "pip install bs4")
pag.press('enter', interval=0.1) #Kích hoạt môi trường do venv

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER COMMENTS GITLENS

Hình 88. Chạy setting_before.ipynb

Chương trình sẽ cài đặt tất cả môi trường và thư viện cần thiết.

Để chạy chương trình hãy mở file process.ipynb và nhấn vào “Run all” như hình dưới , lưu ý khi nhấn “Run all” không thao tác bất cứ thứ gì KẾ CẢ DI CHUỘT.

```

import pyautogui as pag
import os
import time

current = os.getcwd().replace("\\", "/") #Lấy vị trí hiện tại
print(current)

e:/Scrapy_DA2

x, y = pag.position()
x_save, y_save = x, y
print(x_save, y_save)

#Career Builder
pag.hotkey('shift', 'ctrl', '')
time.sleep(1) # Thêm độ trễ để đảm bảo terminal đã mở trước khi nhập
pag.typewrite(current+"/venv/Scripts/Activate.ps1")
pag.press('enter', interval=0.1) #Kích hoạt môi trường do venv
pag.press('enter', interval=0.1)
pag.press('enter', interval=0.1)
pag.typewrite('scrapy crawl career')
pag.press('enter', interval=0.1)

pag.hotkey('shift', 'ctrl', '')
time.sleep(1) # Thêm độ trễ để đảm bảo terminal đã mở trước khi nhập
pag.typewrite(current+"/venv/Scripts/Activate.ps1")
pag.press('enter', interval=0.1) #Kích hoạt môi trường do venv
pag.typewrite('cd Careerlink/Careerlink')
pag.press('enter', interval=0.1)

```

Hình 89. Chạy file process.ipynb

Đợi đến khi Terminal hiện ra bảng sau là có thể sử dụng máy tính và phần mềm gõ tiếng việt bình thường, tuy nhiên không tắt Visual Studio Code :

```

PS E:\Scrapy_DA2> & e:/Scrapy_DA2/venv/Scripts/Activate.ps1
○ (venv) PS E:\Scrapy_DA2> Da chay xong chuong trinh, ban co the thao tac binh thuong, vui long khong tat VSCode.

```

Hình 90. Màn hình terminal kết thúc

4. Một số hình ảnh truy vấn từ cơ sở dữ liệu sau thu thập

Truy vấn dữ liệu chung:

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Table: Stg_ThongTin_raw

Columns:

- ID int(11) A1
- Web varchar(20)
- Nganh varchar(20)
- NganHoc varchar(50)
- Link varchar(300)
- TenCty varchar(300)
- CongTy varchar(300)
- TrinhThanh varchar(70)
- LaoLienKham varchar(70)
- KinhNghiem varchar(100)
- YeuCau varchar(30)
- MoTa text

Object Info Session

Query Completed

10°C Cloudy

Result Grid

Action Output

Time Action

1 21:54:06 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw LIMIT 0, 5000
5000 rows returned

2 21:54:20 SELECT COUNT(*) FROM ThongTinTuyenDung.Stg_ThongTin_raw LIMIT 0, 5000
1 rows returned

3 22:01:59 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw LIMIT 0, 5000
5000 rows returned

Duration / Fetch 0.016 sec / 3.265 sec
170.437 sec / 0.000 sec
0.063 sec / 2.078 sec

10:03 PM 1/23/2024

Hình 91. Lấy ra dữ liệu từ Stg_ThongTin_raw

Truy vấn dữ liệu ITViec:

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Table: Stg_ThongTin_raw

Columns:

- ID int(11) A1
- Web varchar(20)
- Nganh varchar(20)
- NganHoc varchar(50)
- Link varchar(300)
- TenCty varchar(300)
- CongTy varchar(300)
- TrinhThanh varchar(70)
- LaoLienKham varchar(70)
- KinhNghiem varchar(100)
- YeuCau varchar(30)
- MoTa text

Object Info Session

Query Completed

10°C Cloudy

Result Grid

Action Output

Time Action

1 21:54:23 SELECT COUNT(*) FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITViec' 5000 rows returned

2 22:01:59 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITViec' LIMIT 0, 5000 5000 rows returned

3 22:03:43 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITViec' LIMIT 0, 5000 450 rows returned

4 22:06:23 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITViec' LIMIT 0, 5000 799 rows returned

5 22:08:09 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITViec' LIMIT 0, 1000 799 rows returned

6 22:10:27 SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw LIMIT 0, 1000 1000 rows returned

Duration / Fetch 170.437 sec / 0.000 sec
0.063 sec / 2.078 sec
0.031 sec / 95.266 sec
0.063 sec / 95.500 sec
0.031 sec / 137.750 sec
0.375 sec / 0.531 sec

10:13 PM 1/23/2024

Hình 92. Dữ liệu thu thập từ ITViec

Truy vấn dữ liệu CareerBuilder:

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
1 • SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'careerBuilder'
```

The results grid displays data from the Stg_ThongTin table, filtered by Web = 'careerBuilder'. The columns include ID, Web, Nganh, NganhCon, Link, TenCV, CongTy, TinhThanh, and others. The results show various job postings from CareerBuilder, such as R&D Scientist, ACCOUNT MANAGER, Project Leader, etc., across different companies like CÔNG TY CỔ PHẦN GIẢI PHÁP GENE - GENE SO... and CÔNG TY TNHH THƯƠNG MẠI VÀ DỊCH VỤ TRAILER (HUYỀN THOM).

Hình 93. Dữ liệu thu thập từ CareerBuilder

Truy vấn dữ liệu ITNaVi:

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
1 • SELECT * FROM ThongTinTuyenDung.Stg_ThongTin_raw WHERE Web = 'ITNaVi'
```

The results grid displays data from the Stg_ThongTin table, filtered by Web = 'ITNaVi'. The columns include ID, Web, Nganh, NganhCon, Link, TenCV, CongTy, TinhThanh, and others. The results show various job postings from ITNaVi, such as PHP Dev, Software Tester, Principal Senior PHP Developer, etc., across different companies like CÔNG TY TNHH K+S VIỆT NAM and Yamaha Motor Vietnam.

Hình 94. Dữ liệu thu thập từ ITNaVi

Truy vấn dữ liệu DevWork:

Table: Stg_ThongTin

Columns:

- ID**: int(11) A1
- Web**: varchar(20)
- Nganh**: varchar(50)
- NganhCan**: varchar(50)
- Link**: varchar(200)
- TenCV**: varchar(50)
- CongTy**: varchar(50)
- TinhThanh**: varchar(50)
- LoaiLinh**: varchar(70)
- CapBac**: varchar(50)
- HanhNganCV**: varchar(50)
- YeuCau**: text
- Mota**: text
- ...**

Action Output

Output

Action Output

Output

Object Info Session Query Completed 10°C Cloudy 10:19 PM 1/23/2024

Hình 95. Dữ liệu thu thập từ DevWork

Truy vấn dữ liệu ViecLam24h:

Table: Stg_ThongTin

Columns:

- ID**: int(11) A1
- Web**: varchar(20)
- Nganh**: varchar(50)
- NganhCan**: varchar(50)
- Link**: varchar(200)
- TenCV**: varchar(50)
- CongTy**: varchar(50)
- TinhThanh**: varchar(50)
- LoaiLinh**: varchar(70)
- CapBac**: varchar(50)
- HanhNganCV**: varchar(50)
- YeuCau**: text
- Mota**: text
- ...**

Action Output

Output

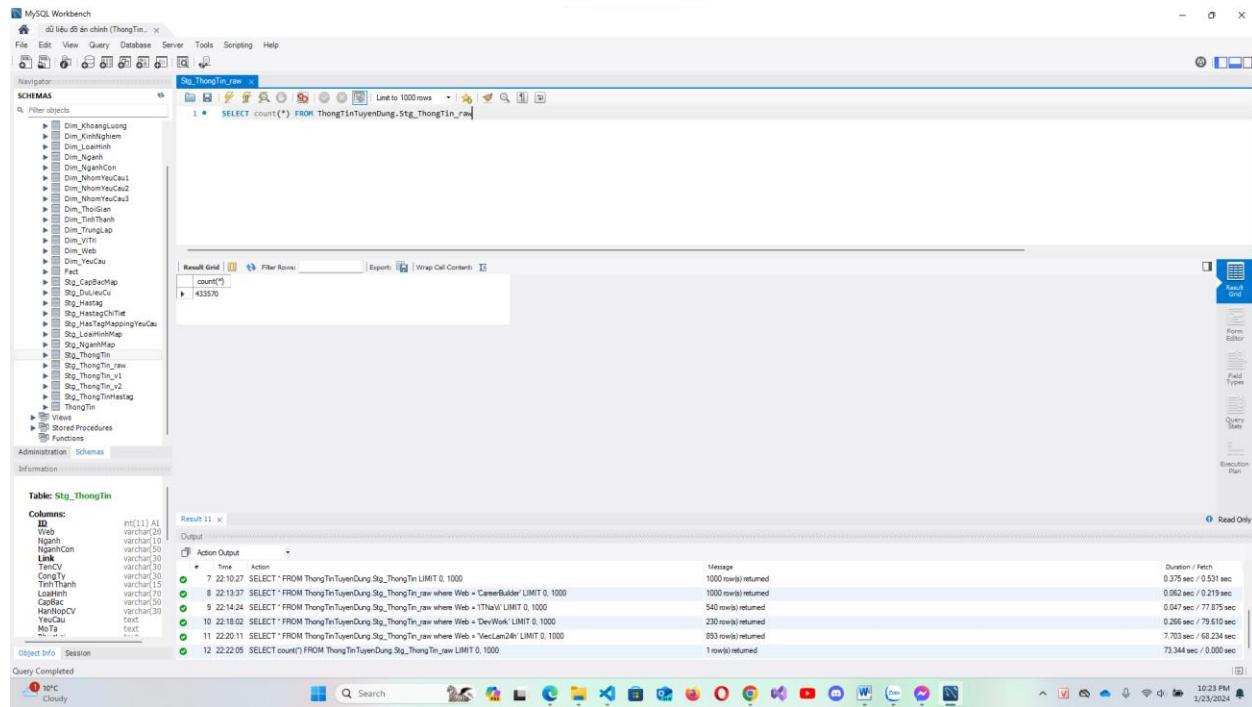
Action Output

Output

Object Info Session Query Completed 10°C Cloudy 10:21 PM 1/23/2024

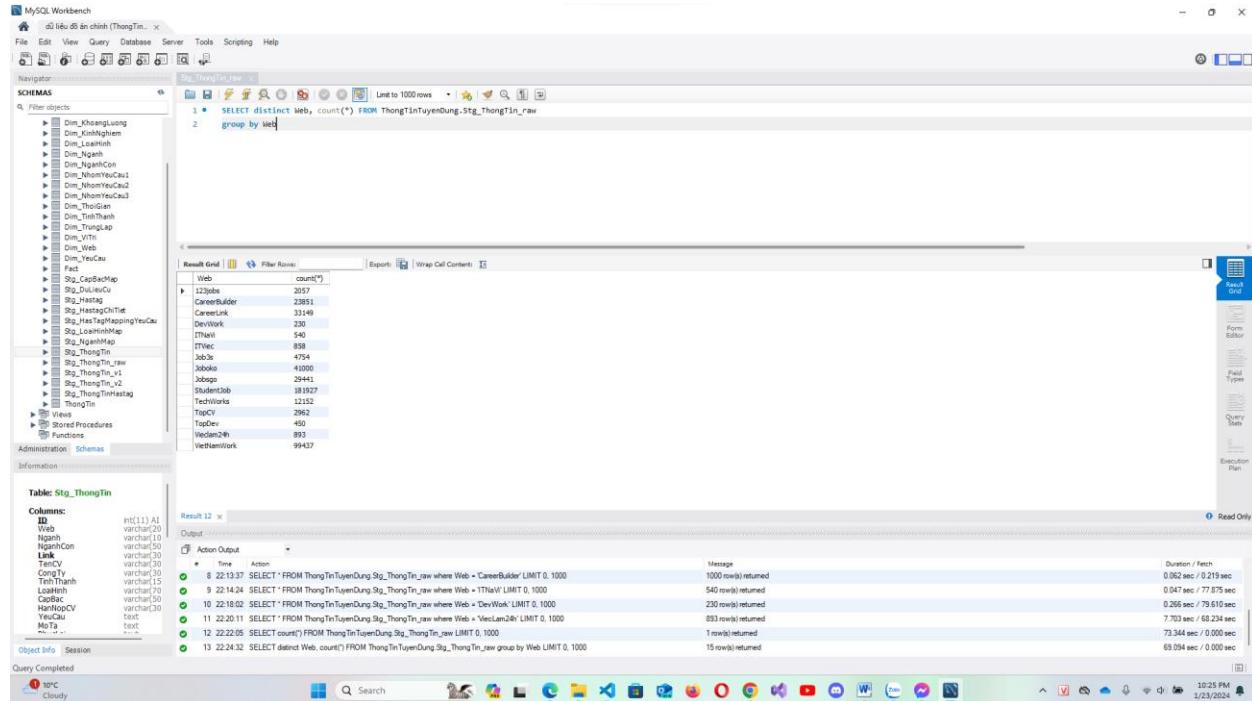
Hình 96. Dữ liệu thu thập từ ViecLam24h

Kiểm tra số lượng tin thu thập:



Hình 97. Tổng số lượng tin thu thập

Kiểm tra số lượng web thu thập và số lượng tin thu thập mỗi web:



Hình 98. Kiểm tra số lượng tin thu thập của từng web

IV. Kết luận

Trong đồ án này, em đã thu thập được các tin tuyển dụng từ 15 trang web khác nhau với các cấu trúc web khác nhau.

Thực hiện đồ án này, em đã có cơ hội tiếp xúc với rất nhiều công nghệ mới, nắm được các kỹ năng và nghiệp vụ cần thiết trong một dự án phân tích dữ liệu, quy trình của một dự án phân tích dữ liệu, đặc biệt là trong khâu chuẩn bị dữ liệu, hiểu được tình trạng thị trường tuyển dụng hiện nay và đã được thực hành với một dự án thực tế.

Với mỗi trang web khác nhau, người thiết kế có thể có nhiều cách thiết kế khác nhau, vì vậy trước khi lấy dữ liệu từ một trang web nào đó cần nghiên cứu trang web đó và thiết kế chương trình phù hợp với trang web.

Do cấu trúc trang web có khả năng sẽ thay đổi trong tương lai do đội ngũ IT của trang web đó làm việc liên tục và cập nhật liên tục các thay đổi, vì thế cũng cần liên tục thay đổi code để phù hợp với cấu trúc trang web mới khi có sự thay đổi.

Bài viết sẽ còn nhiều thiếu sót do thời gian tìm hiểu và thực hiện còn hạn chế, còn rất nhiều cách để có thể lấy dữ liệu từ một trang web và tăng tốc độ lấy dữ liệu chẳng hạn như chạy đa luồng các trình tự động hay các request tới server....

Các phương thức bảo mật web sẽ ngày càng được nâng cao để tránh việc truy cập trái phép vào dữ liệu và các phương thức bảo mật web cũng sẽ ngày càng được cải tiến để bắt kịp với tốc độ phát triển của hàng rào bảo mật web.

Toàn bộ source code trong bài viết có thể tìm thấy tại đây :

https://github.com/Silen187/Scrapy_20231.git

V. Tài liệu tham khảo

1. *The Scrapy Playbook, The FreeCodeCamp Python Scrapy Beginners Course,*
<https://thepythonscrapyclaybook.com/freecodecamp-beginner-course/>
2. MrFuguDataScience, *Selenium Webdriver issue July 2023, 2023,*
<https://github.com/MrFuguDataScience/Webscraping.git>