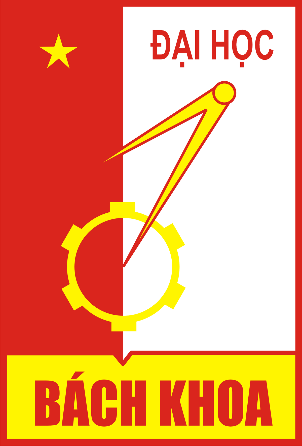
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**KHOA TOÁN ỨNG DỤNG VÀ TIN HỌC**

**----- □  □ -----**

****

**BÁO CÁO ĐỒ ÁN I**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | **ThS. Nguyễn Danh Tú** |
| Sinh viên thực hiện : | **Dương Công Thái** |
| Mã số sinh viên : | **20216883** |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# 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 thong 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 websiste 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 và cách khác. Vì vậy, em rất mong được thầy và bạn đọc góp ý để có thể hoàn thiện hơn.

Em xin chân thành cảm ơn!

1. **Giới thiệu công cụ**

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….

Scrapy gồm một số file sau:

1. Items.py : Định hình khung cấu trúc cho dữ liệu lấy về
2. Middlewares.py : Các phần trung gian dung để truy cập vào trang web
3. Pipelines.py : Thiết kế đường ống chuyển dữ liệu vào nơi lưu trữ.
4. Settings.py : Các thiết lập cài đặt cho robot.
5. Folder spiders: Nơi ta cài đặt spider, tức là robot sẽ thực hiện công việc truy cập và lấy dữ liệu.

Selenium: là một bộ công cụ được sử dụng để automate 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 web.

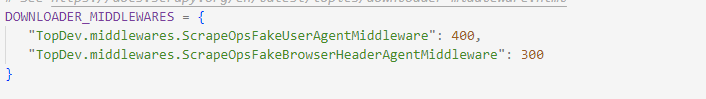
Middleware mặc định được em sử dụng gồm hai class nhằm giả mạo user-agent và giả mạo browser-header như sau:





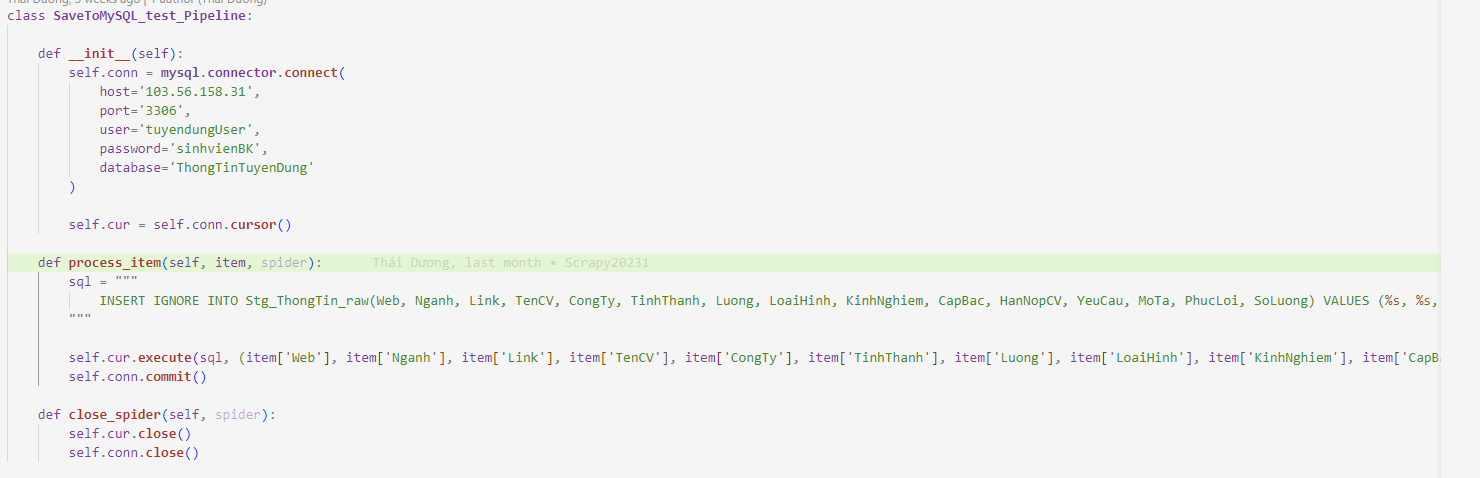
Để sử dụng file middleware này, cài đặt được sử dụng mặc định trong các file setting.py như sau:



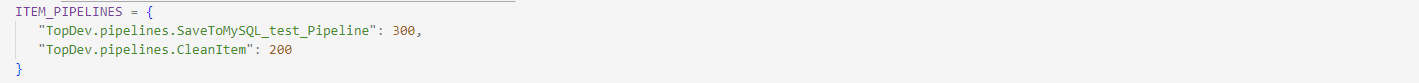


File pipelines.py sẽ định hình cách thức và nơi lưu dữ liệu:









1. **Cấu trúc của bảng tin cơ sở dữ liệu**

Trong cơ sở dữ liệu lưu trữ thông tin, việc đổ dữ liệu sẽ thực hiện trên bảng Stg\_ThongTin\_raw và sau các thao tác tiền xử lý dữ liệu, dữ liệu sẽ được đổ lại sang bảng Stg\_ThongTin.

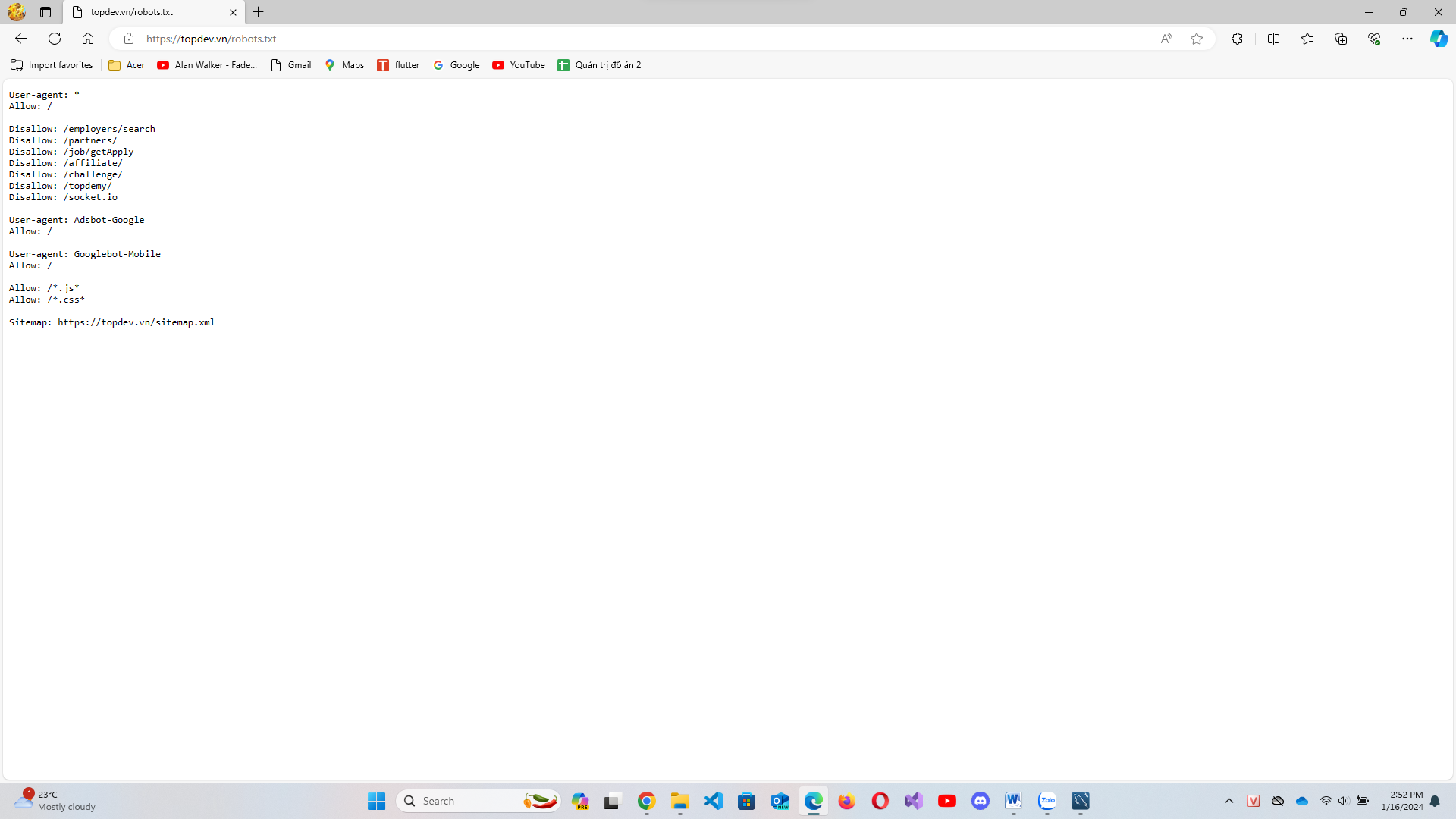
Các cột dữ liệu và mô tả được thống kê trong bảng sau:

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | Mô tả |
| ID | (PK)(AI)INT(11) | Mã số cho một tin bất kì trong CSDL |
| Web | VARCHAR(20) | Trang web thu thập dữ liệu |
| Nganh | VARCHAR(150) | Ngành việc mà tin thu thập thuộc về |
| Link | (UQ)VARCHAR(300) | Đường dẫn đến công việc |
| CongTy | VARCHAR(300) | Tổ chức đăng tin tuyển dung |
| TinhThanh | VARCHAR(150) | Tỉnh thành nơi làm việc |
| LoaiHinh | VARCHAR(70) | Loại hình công việc |
| KinhNghiem | VARCHAR(100) | Kinh nghiệm yêu cầu của nhà tuyển dụng |
| CapBac | VARCHAR(50) | Cấp bậc của công việc tuyển dụng |
| HanNopCV | VARCHAR(30) | Ngày kết thúc nộp hồ sơ ứng tuyển |
| SoLuong | INT(11) | Số lượng muốn tuyển |
| Luong | VARCHAR(100) | Mức lương àm nhà tuyển dụng đưa ra |
| YeuCau | TEXT | Yêu cầu công việc |
| MoTa | TEXT | Mô tả công việc |
| PhucLoi | TEXT | Phúc lợi, quyền lợi khi làm việc |

Khi tiến hành thu thập thông tin ở một trang web nào đó, ta sẽ tiến hành khảo sát thông tin ban đầu của trang web đó để tìm hiểu cấu trúc của trang web, xem trong những trường dữ liệu ta yêu cầu, trang web đáp ứng được các trường dữ liệu nào, các trường dữ liệu còn lại sẽ tùy trường hợp để xử lý.

1. **ROBOTS.TXT**

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 ảnh trên là một ví dụ về tệp tin robots.txt của trang TopDev. User-agent là định danh cho trình duyệt người dung. 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.

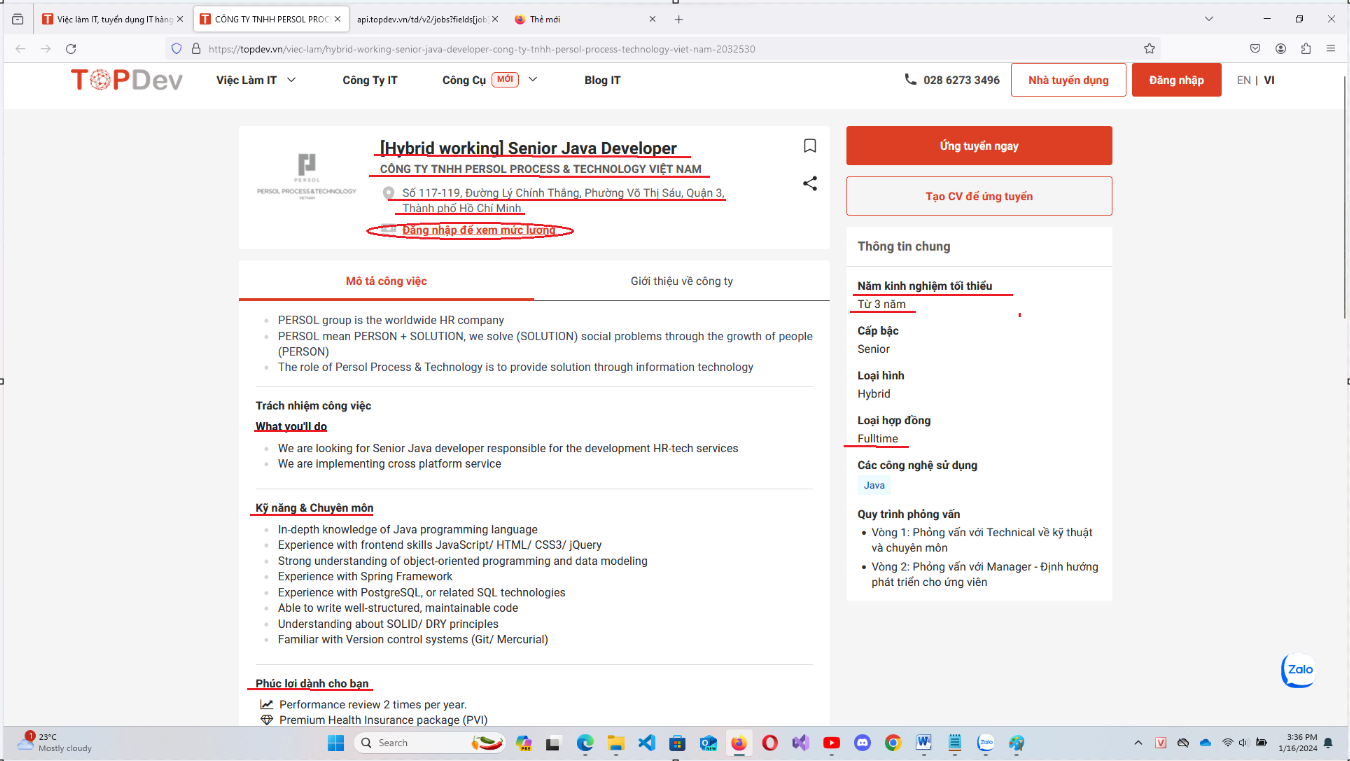
Ta có thể tùy chọn tuân thủ robots.txt hoặc không, tuy nhiên khi không tuân thủ robots.txt, việc thu thập dữ liệu sẽ có khả năng bị phát hiện và có khả năng rất lớn sẽ bị chặn địa chỉ IP.

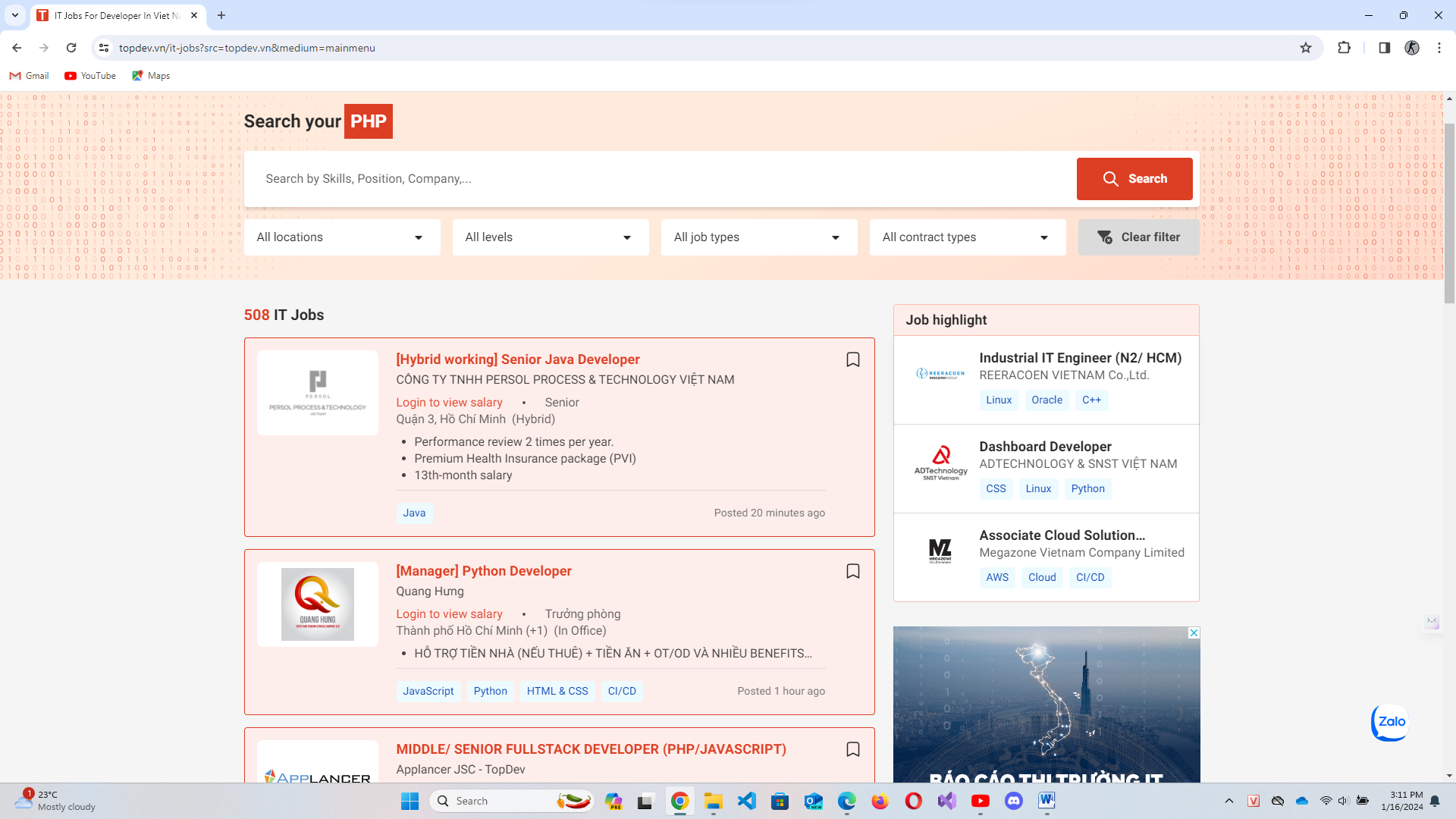
Trong bài báo cáo này, các thiết lặp mặc định khi sử dụng scrapy sẽ được đặt là KHÔNG tuân theo robots.txt.



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.

1. **TopDev**
2. **Khảo sát trang web**

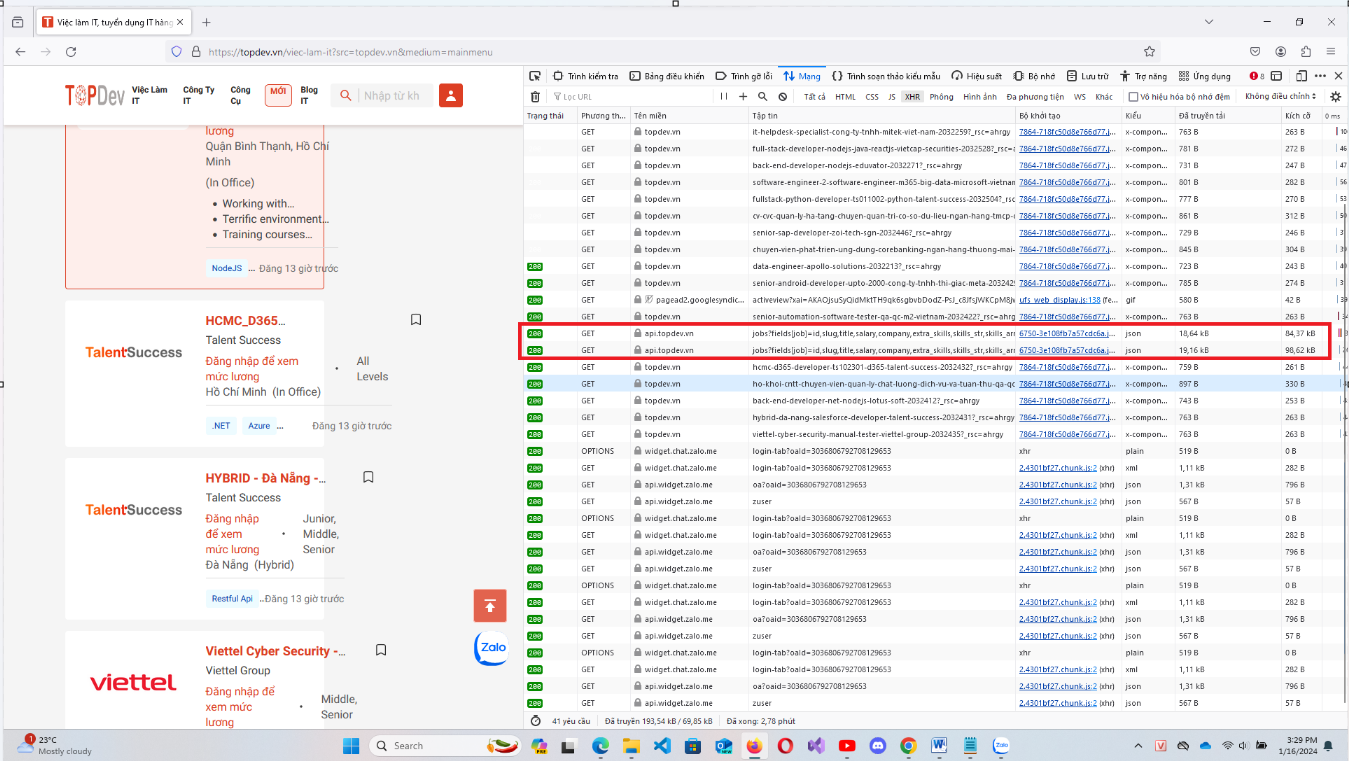


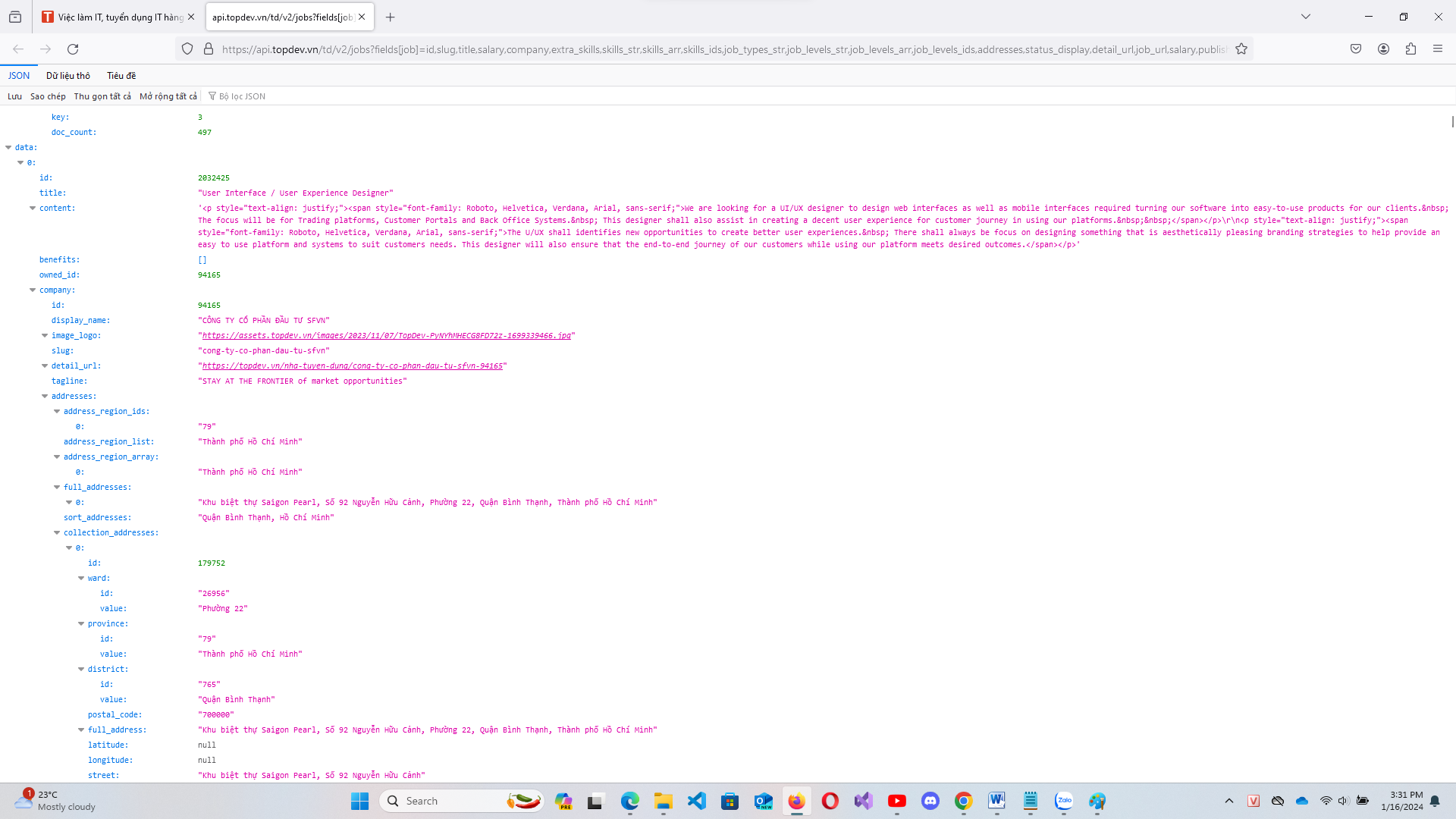
****

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.

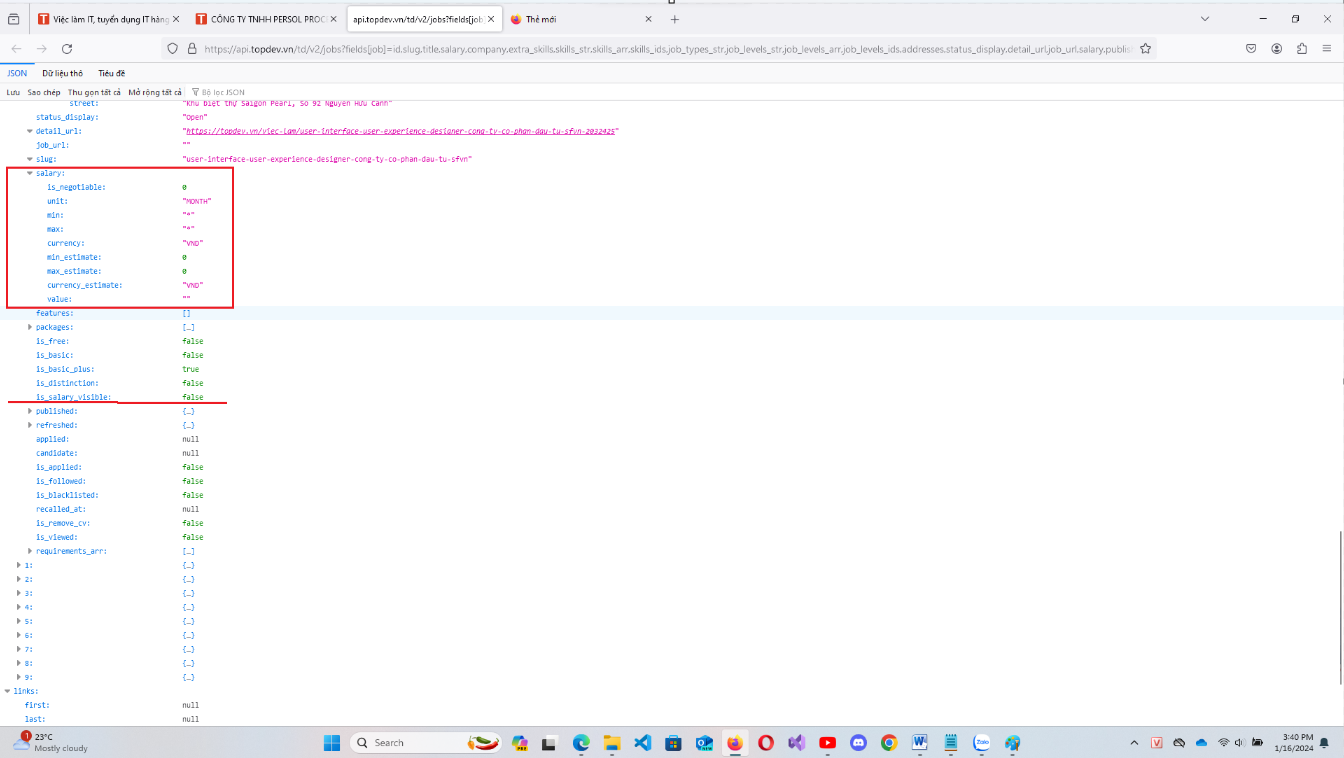
Trong mô hình infinite scroll, thông thường có thể có việc gửi các yêu cầu (requests) đến máy chủ để lấy thêm dữ liệu khi cần. Điều này thường được thực hiện thông qua API, nơi máy chủ trả về dữ liệu mới trong định dạng JSON hoặc XML. Điều này giúp giảm thời gian tải trang ban đầu và tạo trải nghiệm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

****

****

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 cảu trang web thay vì file json API.

****

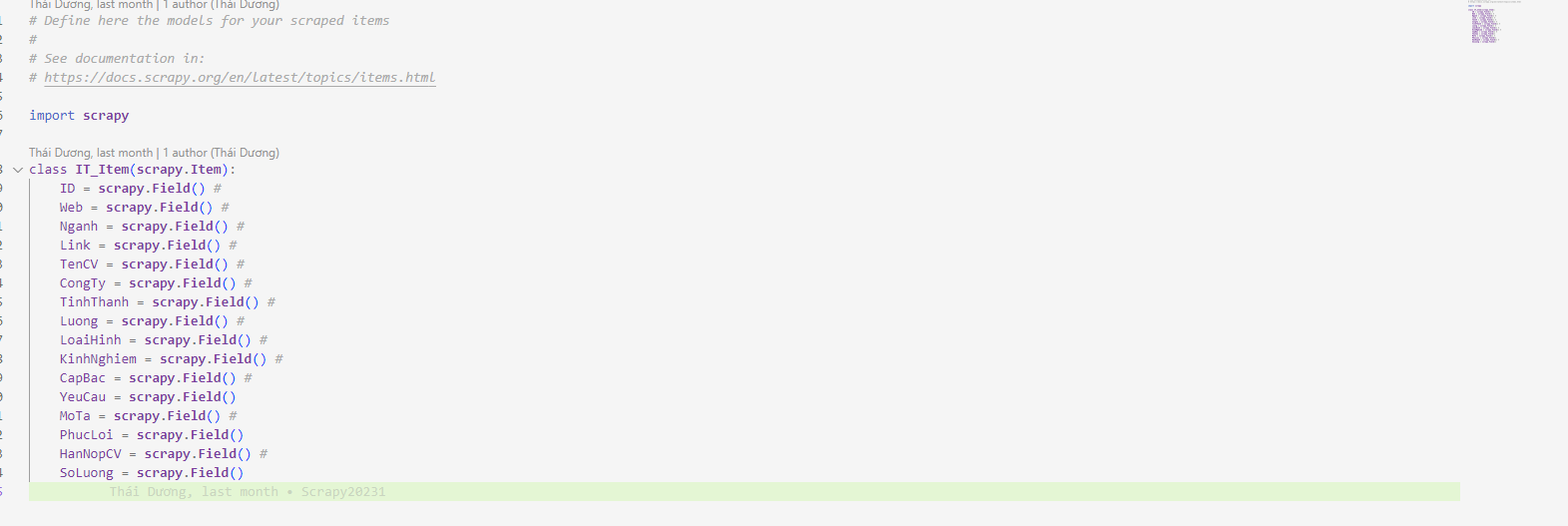
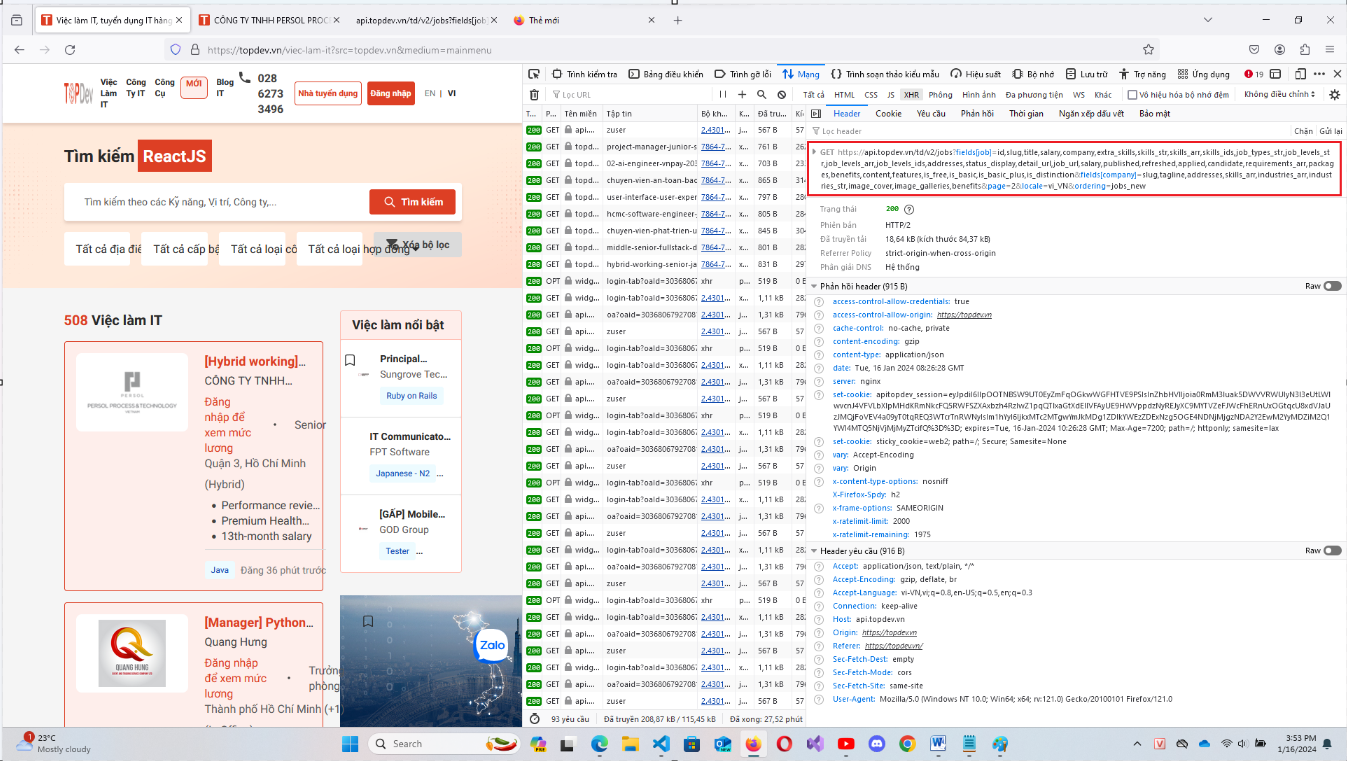
Ngoài ra còn một vấn đề nữa, đó là không đăng nhập thì trang web không hiển thị cho ta xem lương và trong file json API cũng không xem được lương, vì vậy để lấy lương của công việc, ta sẽ tìm trong mã html trả về khi ta gửi yêu cầu tới đường dẫn công việc.

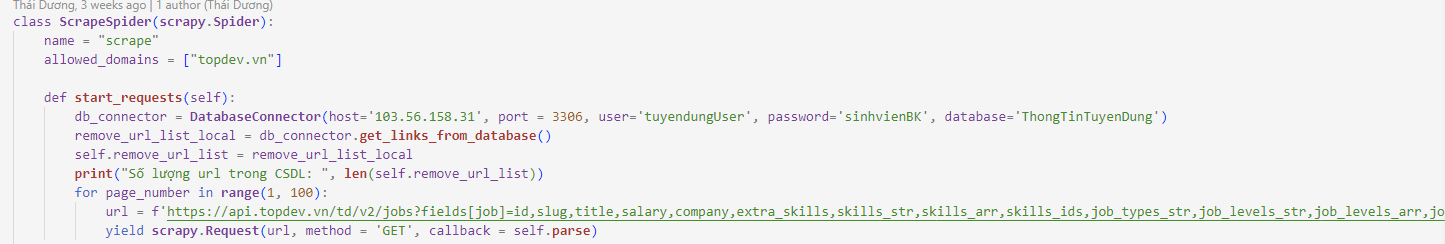
1. **Xây dựng chương trình**

Mỗi lần cuộn xuống, trang web sẽ load them cho ta 10 item 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, đều này sẽ làm tăng tốc độ thực hiện của chương trình và tránh bị chặn do truy cập quá nhiều.

Items.py:



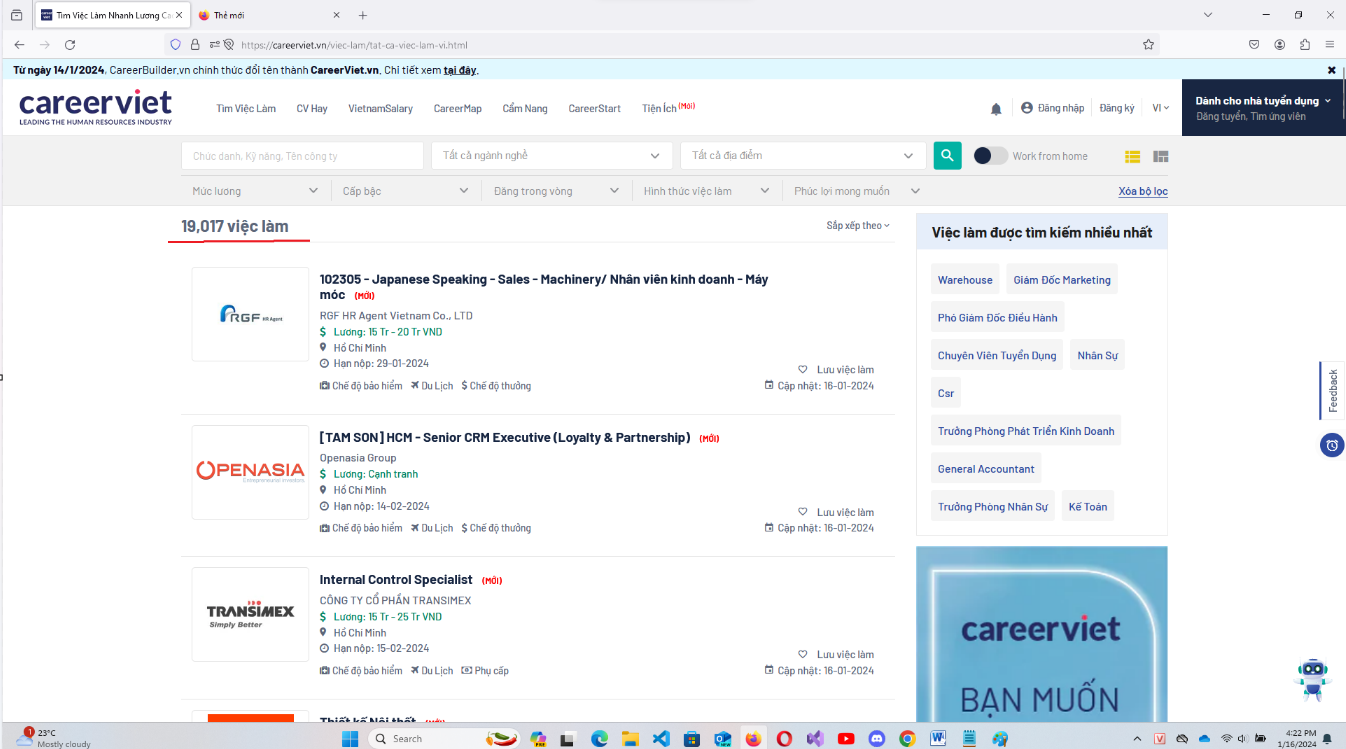




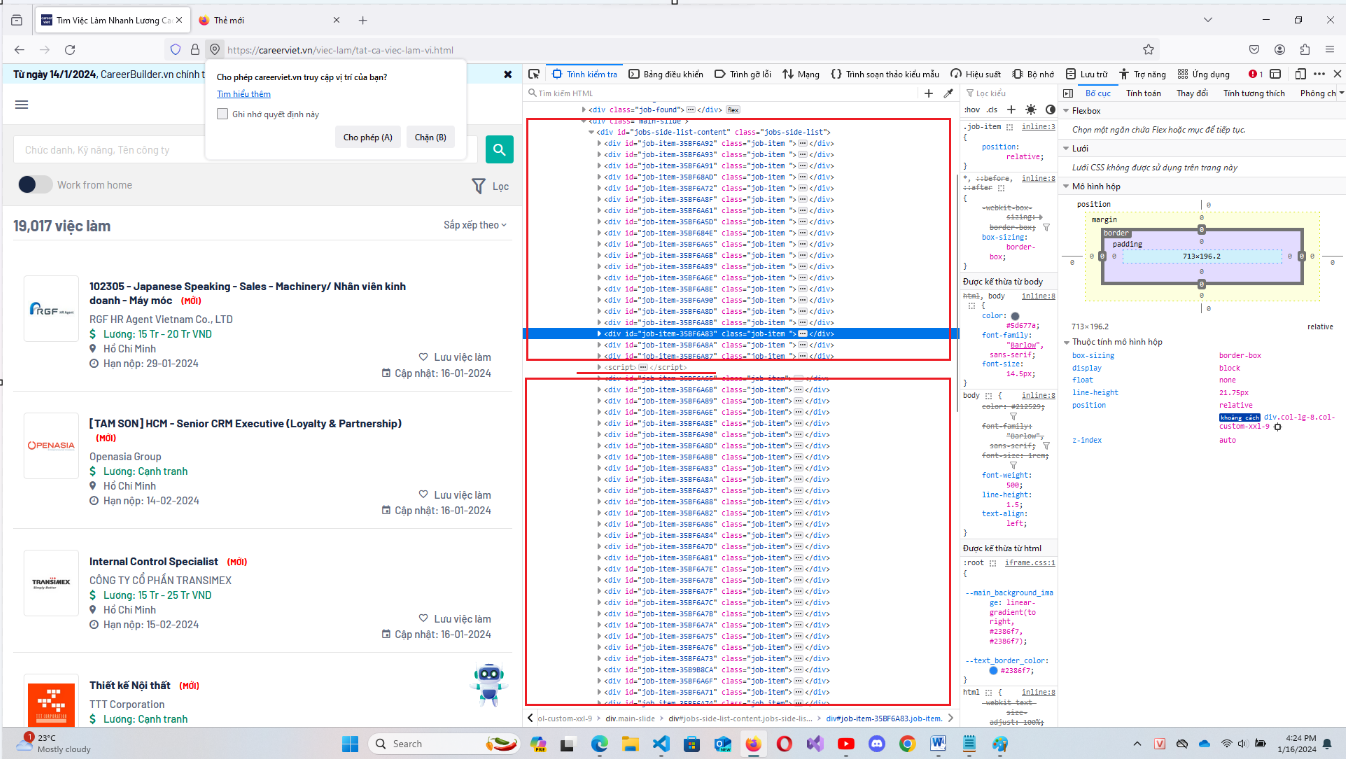




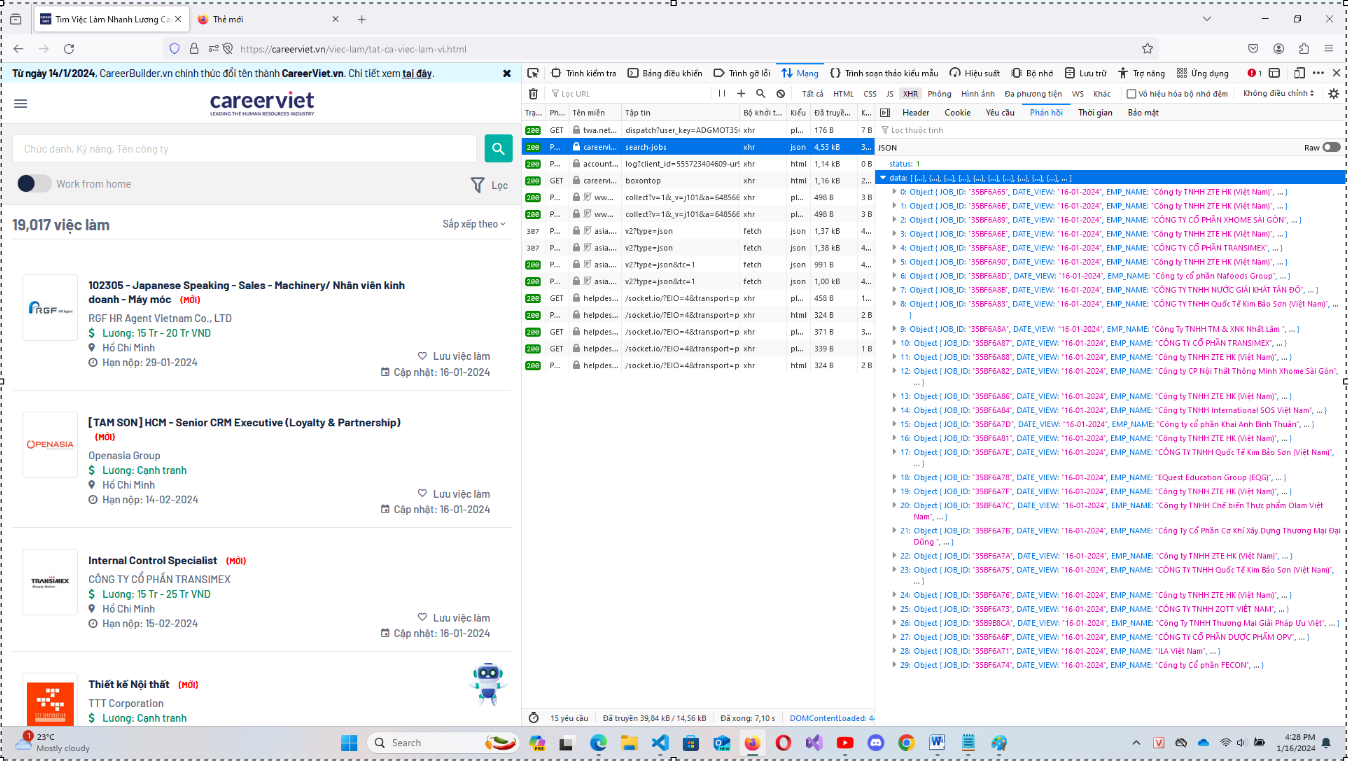
1. **CareerBuilder**
2. **Khảo sát trang web**

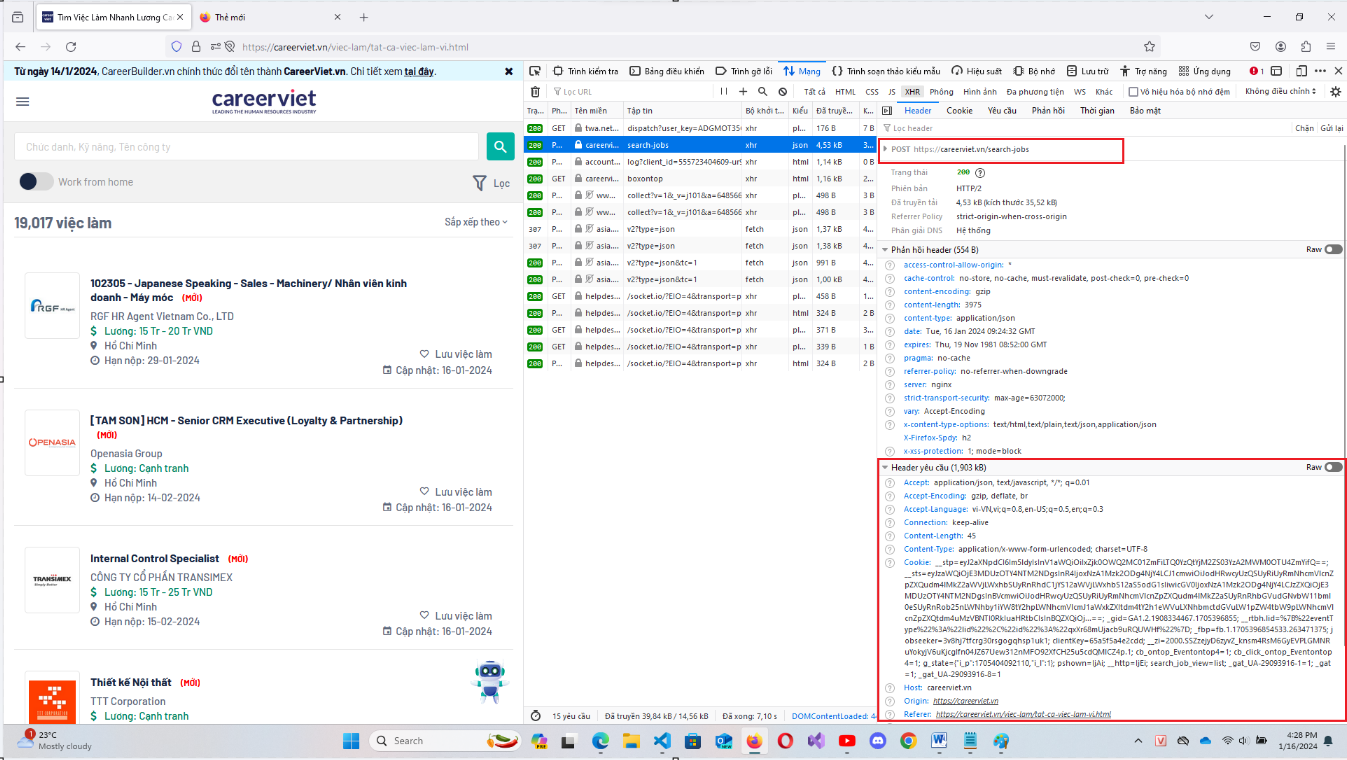


Qua khảo sát em thấy, 1 trang CareerBuilder sẽ trả cho ta 50 công việc, vì vậy ta sẽ lấy số công việc này để tính rat rang cuối cùng cần truy cập tới.

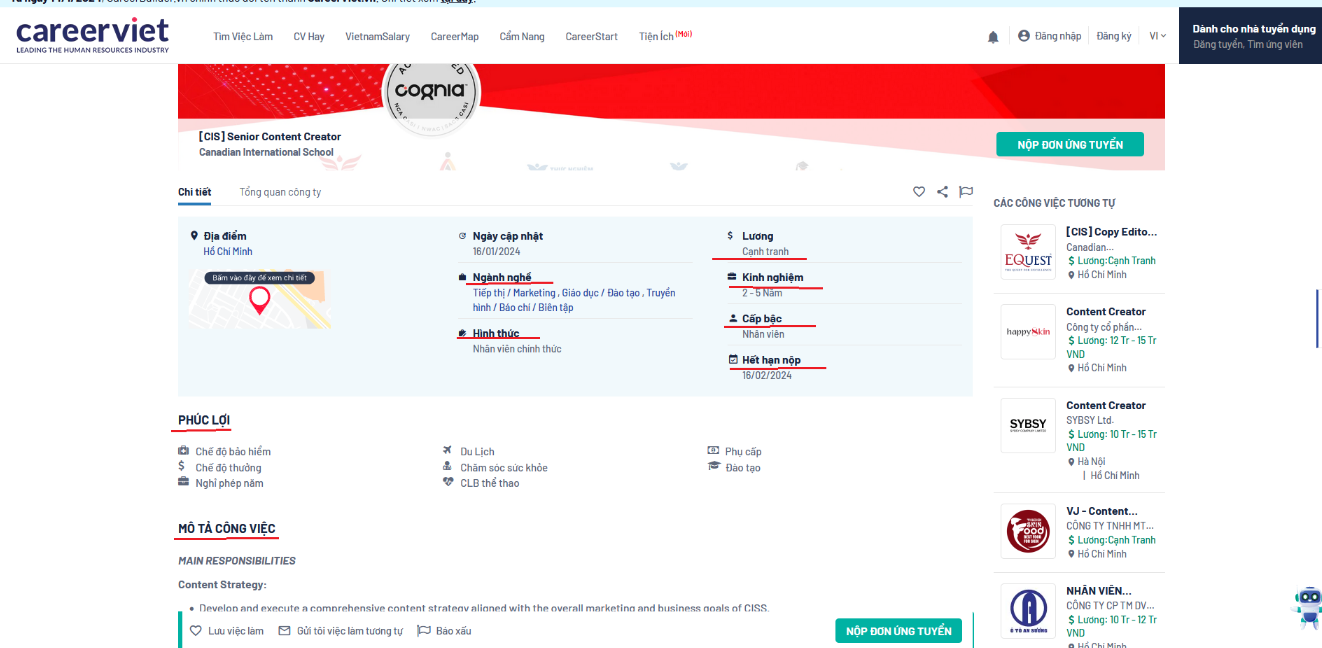


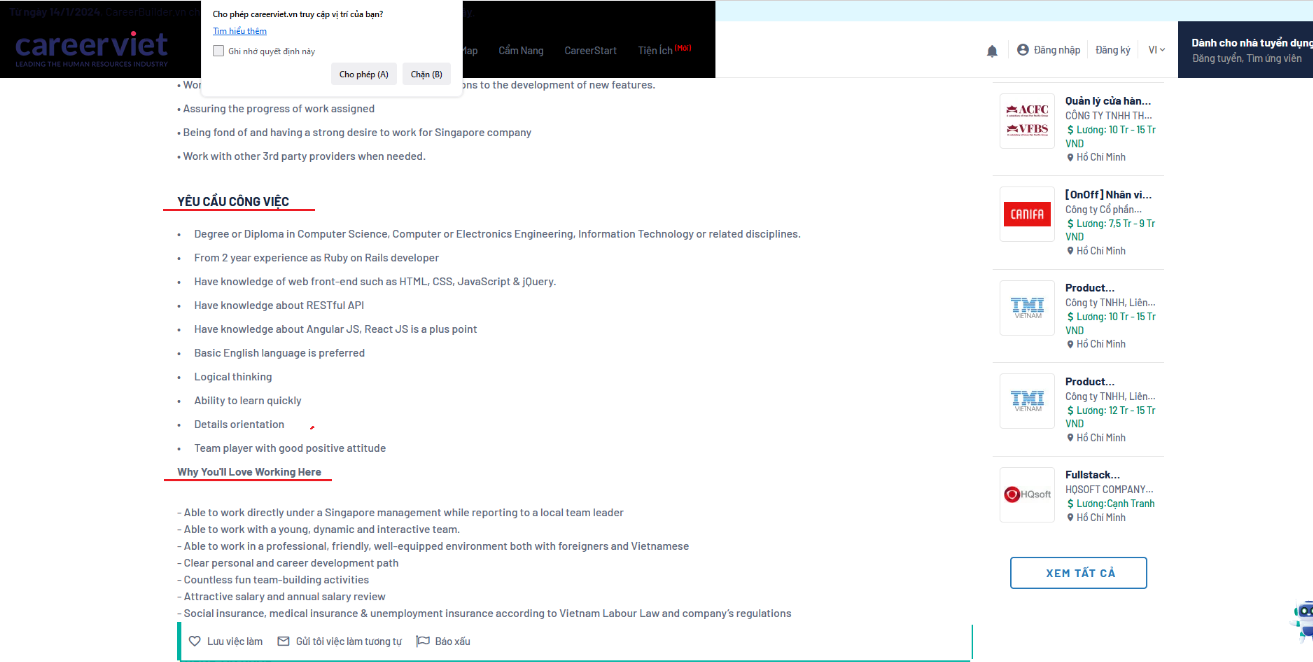
Khảo sát cũng cho thấy khi gửi yêu cầu tới trang thì chỉ có 20/50 công việc ở ô màu đỏ phía trên là được trả dưới dạng mã html, còn 30 công việc phía dưới không được trả dưới dạng mã html mà được trả về dưới dạng API.





Sauk hi khảo sát về cách trả về dữ liệu của trang web, ta khảo sát tới cấu trúc dữ liệu của trang web:





1. **Xây dựng chương trình**

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.







