**INSTRUCTION**

1. **File link\_gathering.py**

**Mục tiêu:**

* Truy cập URL đã chỉ định.
* Trích xuất từng đường link cụ thể.
* Bỏ qua các thẻ không có đầy đủ link.
* Ghi dữ liệu đầy đủ vào file csv.

**A screen shot of a computer code

AI-generated content may be incorrect.**

Trong đó:

* *goto* để điều hướng trình duyệt đến **URL** chỉ định.
* *retries* sẽ thử lại tối đa *retries* lần nếu có lỗi *(mạng chậm hoặc lỗi).*
* Cuối cùng sẽ trả về **True** nếu thành công và **False** nếu lỗi.

A black screen with white text

AI-generated content may be incorrect.

* Trả về chuỗi **User – Agent** ngẫu nhiên để tránh bị phát hiện bot. Giúp mỗi lần truy cập sẽ giả làm các trình duyệt các trình duyệt khác nhau.

A screen shot of a computer code

AI-generated content may be incorrect.

Sau khi khởi động **Playwright** ở chế độ đồng bộ:

* Mở trình duyệt **Chromium** ở chế độ hiển thị *(headless=False).*
* Set **User-Agent** ngẫu nhiên cho mỗi phiên duyệt web.
* Tạo một tab trình duyệt mới.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

Mỗi vòng lặp:

1. Chờ *li[itemprop='itemListElement'*] xuất *hiện (tức các khối tin đăng).*
2. Nếu không load được:
   * Đổi chế độ **headless** (*ẩn/hiện trình duyệt*) → Có thể giúp bypass lỗi.
   * Khởi động lại trình duyệt, vào lại trang.
3. Nếu load thành công:
   * Lấy số lượng tin đăng trên trang.
   * Với từng tin đăng:
     + Truy cập phần tử **<a>** chứa link.
     + Bỏ qua nếu link rỗng hoặc là link trang chủ.
     + Ghi link đầy đủ vào file CSV.

Sau đó:

* Tăng biến **current\_page.**
* Tạo **URL** mới cho trang tiếp theo.
* Gọi **safe\_goto** để mở.

Kết quả cuối cùng:

* File: <file\_name>.csv.
* Nội dung: Mỗi dòng là một link tin bất động sản dạng.

1. **File output\_nha-o.py, output\_dat.py, output\_can-ho.py**

**Mục tiêu:**

* Đọc danh sách các đường link bài đăng từ file csv đã lưu ở trên.
* Truy cập từng link, trích xuất dữ liệu chi tiết như: tiêu đề, loại bất động sản, giá tiền, diện tích, địa chỉ,...
* Bỏ qua các link bị lỗi hoặc không đủ dữ liệu.
* Ghi dữ liệu đầy đủ vào file csv.



Đọc danh sách các link trong file input

A screen shot of a computer program

AI-generated content may be incorrect.

Tiến hành truy cập link nếu lỗi sẽ trả về None.

A computer screen shot of text

AI-generated content may be incorrect.

Tiến hành xử lý popup quảng cáo bằng cách click vào chính giữa màn hình để ẩn nội dung popup.

A screen shot of a computer program

AI-generated content may be incorrect.

Xử lý để nhấn “Xem thêm”:

* Tìm nút xem thêm với bất kỳ thẻ *<button>* nào có nội dung là **“Xem thêm”**
* Gọi scrollIntoView() để cuộn trang xuống giúp nút nằm trong tầm nhìn.
* Tránh lỗi nhấn nút nằm ngoài khung nhìn trình duyệt (*viewport).*
* Nhấn nút đó với **timeout** là 10 giây.
* *time.sleep(2):* chờ 2 giây cho nội dung mở rộng được tải xong.
* *Nếu không có nút xem thêm sẽ bỏ qua mọi lỗi.*

**A screen shot of a computer code

AI-generated content may be incorrect.**

Sau khi đã xử lý xong “Xem thêm” sử dụng CSS selector để lấy text của từng trường thông tin như ở trên, ví dụ:

* + ***h1:*** *tiêu đề*
  + ***.a3jfi3v[itemprop="house\_type"]:*** *loại nhà*
  + ***.pyhk1dv****: giá*
  + ***.bwq0cbs.flex-1****: địa chỉ*

*A screen shot of a computer code

AI-generated content may be incorrect.*

Nếu trường nào đó không có thông tin hoặc không tồn tại thì sẽ trả về ‘N/A’.

*A screenshot of a computer screen

AI-generated content may be incorrect.*

Lưu các text đã lấy vào một dictionary đã thiết lập sẵn ứng với các trường thông tin sau khi sử dụng CSS selector

A screen shot of a computer program

AI-generated content may be incorrect.

Tiến hành lưu các thông tin đã thu thập được vào file output. Ngoài ra còn xử lý các dữ liệu có thông tin là ‘N/A’ sau khi đã gắn ở trên.

1. **File GUI.py**

**A computer screen shot of text

AI-generated content may be incorrect.**

* + Map từng loại dữ liệu vào script scrape tương ứng.
  + Script link\_gathering.py là để crawl link.

**Hàm run\_script(...)**

Đây là phần chính để:

* Chạy script **crawl link** (nếu ở chế độ crawl\_and\_scrape).
* Rồi chạy script **scrape nội dung**.

**Trong đó:**

* Dùng re.sub để **chèn thông số (link, max\_pages, file CSV)** vào script crawl/scrape gốc.
* Lưu bản **tạm thời** (temp\_...) của script rồi subprocess.run() để thực thi.
* Xóa script tạm sau khi chạy xong.

**Giao diện người dùng (GUI)**

Sử dụng tkinter:

* Các thành phần nhập liệu: Entry, OptionMenu, Radiobutton, Button.
* Một khung form\_frame chứa các dòng form nhập.
* Hai chế độ: Crawl + scrape hoặc Scrape only (ẩn hiện ô nhập tương ứng).

**Tương tác chính:**

* Button "🚀 Bắt đầu" → gọi hàm start()
* Hàm start() sẽ validate đầu vào, rồi gọi run\_script() trong một luồng riêng (threading.Thread) để tránh **đơ giao diện**.