* Crawler.py

import requests

import json

import time

import config

def crawl\_products(page=1):

url = f"https://tiki.vn/api/personalish/v1/blocks/listings?limit={config.LIMIT}&page={page}&category={config.CATEGORY\_ID}"

headers = {"User-Agent": "Mozilla/5.0"}

try:

response = requests.get(url, headers=headers, timeout=10)

response.raise\_for\_status()

return response.json().get('data', [])

except Exception as e:

print(f"Trang {page} lỗi: {e}")

return []

def extract\_fields(product):

rating = product.get("rating\_average")

try:

rating = float(rating) if rating is not None else 0.0

except (ValueError, TypeError):

rating = 0.0

return {

"title": product.get("name"),

"price": product.get("price"),

"rating": rating,

"url": f"https://tiki.vn/{product.get('url\_path')}"

}

def save\_crawled\_data():

total\_products = 0

with open("tiki\_book\_data.jsonl", "w", encoding="utf-8") as f:

for page in range(1, config.MAX\_PAGE + 1):

print(f"Đang crawl trang {page}")

products = crawl\_products(page)

if not products:

break

for product in products:

if total\_products >= 1500:

break

item = extract\_fields(product)

f.write(json.dumps(item, ensure\_ascii=False) + "\n")

total\_products += 1

if total\_products >= 1500:

break

time.sleep(1)

print(f"\nĐã lưu {total\_products} sách vào 'tiki\_book\_data.jsonl'.")

if \_\_name\_\_ == "\_\_main\_\_":

save\_crawled\_data()

* Config.py

MYSQL\_CONFIG = {

"host": "localhost",

"user": "crawler",

"password": "123456",

"database": "crawl\_data\_tiki"

}

CATEGORY\_ID = 8322

LIMIT = 20 # 20 sản phẩm / trang

MAX\_PAGE = 50 # 75 trang x 20 = 1500 sản phẩm

* Cleaner.py

import json

import re

INPUT\_FILE = "tiki\_book\_data.jsonl"

OUTPUT\_FILE = "tiki\_book\_data\_clean.jsonl"

def clean\_price(price):

if price is None:

return None

price\_str = re.sub(r"[^\d]", "", str(price))

if not price\_str:

return None

return int(price\_str)

def clean\_rating(rating):

try:

r = float(rating)

if 0 < r <= 5: # loại bỏ rating = 0

return r

else:

return None

except:

return None

def clean\_and\_deduplicate():

seen\_titles = set()

clean\_data = []

with open(INPUT\_FILE, "r", encoding="utf-8") as f:

for line in f:

try:

item = json.loads(line)

title = item.get("title")

price = clean\_price(item.get("price"))

rating = clean\_rating(item.get("rating"))

url = item.get("url")

if not title or price is None or rating is None or not url:

continue

if title.lower() in seen\_titles:

continue

seen\_titles.add(title.lower())

item["price"] = price

item["rating"] = rating

clean\_data.append(item)

except:

continue

with open(OUTPUT\_FILE, "w", encoding="utf-8") as f\_out:

for item in clean\_data:

f\_out.write(json.dumps(item, ensure\_ascii=False) + "\n")

print(f"{len(clean\_data)} sản phẩm còn lại sau khi làm sạch.")

if \_\_name\_\_ == "\_\_main\_\_":

clean\_and\_deduplicate()

* save\_to\_mysql.py

import pymysql

import json

from config import MYSQL\_CONFIG

INPUT\_FILE = "tiki\_book\_data\_clean.jsonl"

def create\_books\_tiki\_table():

conn = pymysql.connect(

host=MYSQL\_CONFIG["host"],

user=MYSQL\_CONFIG["user"],

password=MYSQL\_CONFIG["password"],

database=MYSQL\_CONFIG["database"],

charset='utf8mb4'

)

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS books\_tiki (

id INT AUTO\_INCREMENT PRIMARY KEY,

title TEXT,

price INT,

rating FLOAT,

url TEXT

)

""")

conn.commit()

cursor.close()

conn.close()

print("Đã tạo bảng books\_tiki")

def load\_jsonl(filepath):

items = []

with open(filepath, "r", encoding="utf-8") as f:

for line in f:

try:

items.append(json.loads(line))

except:

pass

return items

def save\_to\_books\_tiki(data):

conn = pymysql.connect(

host=MYSQL\_CONFIG["host"],

user=MYSQL\_CONFIG["user"],

password=MYSQL\_CONFIG["password"],

database=MYSQL\_CONFIG["database"],

charset='utf8mb4'

)

cursor = conn.cursor()

insert\_query = """

INSERT INTO books\_tiki (title, price, rating, url)

VALUES (%s, %s, %s, %s)

"""

for item in data:

cursor.execute(insert\_query, (

item.get("title"),

item.get("price"),

item.get("rating"),

item.get("url")

))

conn.commit()

cursor.close()

conn.close()

print(f"Đã lưu {len(data)} dòng vào bảng books\_tiki")

if \_\_name\_\_ == "\_\_main\_\_":

create\_books\_tiki\_table()

data = load\_jsonl(INPUT\_FILE)

save\_to\_books\_tiki(data)