

Web Scraping Tool - Scraping and Storing Book Data

Table of Contents:

1. Overview
2. Prerequisites
3. Step-by-Step Breakdown of the Script
 - Fetching Data
 - Parsing Data
 - Storing Data in SQLite
4. Error Handling
5. Running the Script
6. Verifying the Data
7. Testing the Script
8. Optimizing the Script for Multiple Pages
9. Conclusion

1. Overview

This web scraping tool extracts book information from the Books to Scrape website. It scrapes:

Book Title

Price

Rating

URL for more details

The scraped data is saved in an SQLite database for later use. The tool is built using Python with the requests and BeautifulSoup libraries for web scraping and parsing.

2. Prerequisites

Before using the script, you need to install the required libraries:

requests: To send HTTP requests and fetch web pages.

beautifulsoup4: To parse the HTML content of the web page

pip install requests beautifulsoup4

3. Step-by-Step Breakdown of the Script

- Fetching Data

The function `fetch_page(url)` is responsible for sending an HTTP request to the given URL and fetching the page content.

- Parsing Data

The function `parse_books(page_content)` takes the page content as input and parses the book data (title, price, rating, and URL) from the HTML using BeautifulSoup.

- Storing Data in SQLite

The `save_data_to_db(data)` function saves the extracted data into a SQLite database named `books.db`.

4. Error Handling

The script contains basic error handling for connectivity issues:

`requests.exceptions.RequestException`: If there's a problem fetching the page (e.g., invalid URL, network issues), the error is caught and printed.

Database Errors: Errors related to SQLite operations (like issues inserting data) will be raised by Python's database interface.

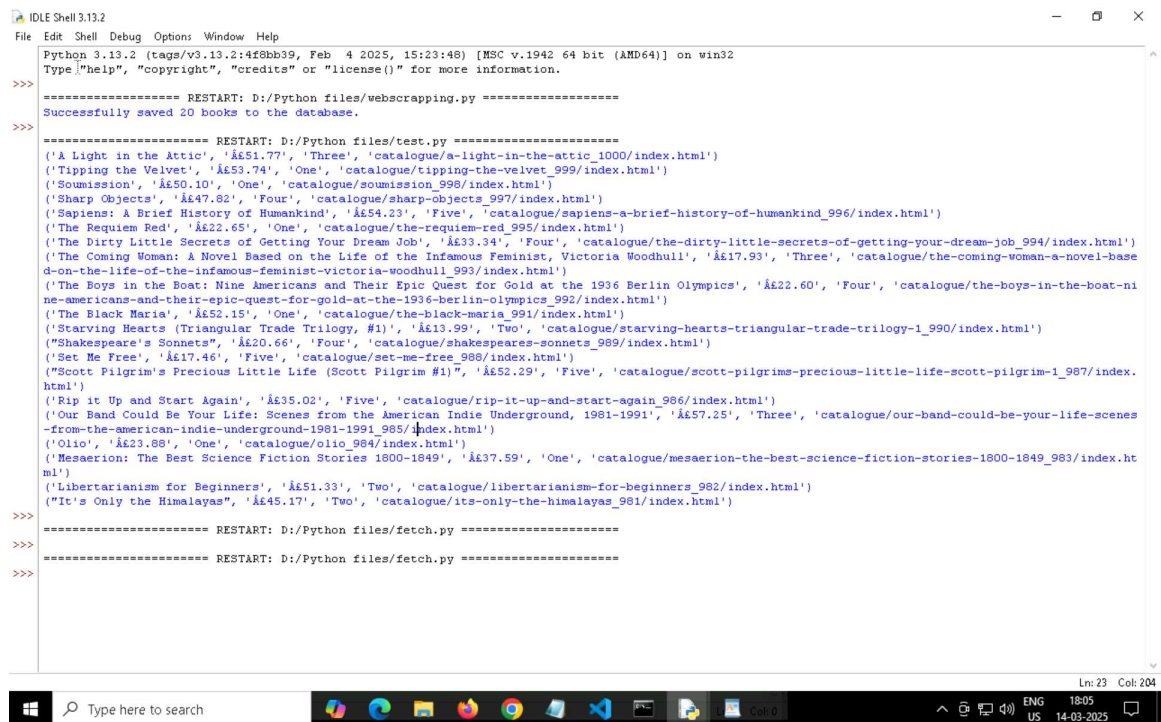
5. Running the Script

Run the Script: To start scraping the website, simply run the script:

```
===== RESTART: D:/Python files/webscrapping.py =====  
Successfully saved 20 books to the database.
```

6. Verifying the Data

After running the script, you can verify the scraped data by querying the SQLite database.



```
IDLE Shell 3.13.2  
File Edit Shell Debug Options Window Help  
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: D:/Python files/webscrapping.py =====  
Successfully saved 20 books to the database.  
>>>  
===== RESTART: D:/Python files/test.py =====  
(('A Light in the Attic', '4.51', 'Three', 'catalogue/a-light-in-the-attic_1000/index.html'))  
(('Tipping the Velvet', '4.53', 'One', 'catalogue/tipping-the-velvet_999/index.html'))  
(('Soumission', '4.50', 'One', 'catalogue/soumission_998/index.html'))  
(('Sharp Objects', '4.47', 'Four', 'catalogue/sharp-objects_997/index.html'))  
(('Sapiens: A Brief History of Humankind', '4.54', 'Five', 'catalogue/sapiens-a-brief-history-of-humankind_996/index.html'))  
(('The Requiem Red', '4.22', 'One', 'catalogue/the-requiem-red_995/index.html'))  
(('The Dirty Little Secrets of Getting Your Dream Job', '4.33', 'Four', 'catalogue/the-dirty-little-secrets-of-getting-your-dream-job_994/index.html'))  
(('The Coming Woman: A Novel Based on the Life of the Infamous Feminist, Victoria Woodhull', '4.17', 'Three', 'catalogue/the-coming-woman-a-novel-based-on-the-life-of-the-infamous-feminist-victoria-woodhull_993/index.html'))  
(('The Boys in the Boat: Nine Americans and Their Epic Quest for Gold at the 1936 Berlin Olympics', '4.22', 'Four', 'catalogue/the-boys-in-the-boat-nine-americans-and-their-epic-quest-for-gold-at-the-1936-berlin-olympics_992/index.html'))  
(('The Black Maria', '4.52', 'One', 'catalogue/the-black-maria_991/index.html'))  
(('Starving Hearts (Triangular Trade Trilogy, #1)', '4.13', 'Two', 'catalogue/starving-hearts-triangular-trade-trilogy-1_990/index.html'))  
(('Shakespeare's Sonnets', '4.20', 'Four', 'catalogue/shakespeares-sonnets_989/index.html'))  
(('Set Me Free', '4.17', 'Five', 'catalogue/set-me-free_988/index.html'))  
(('Scott Pilgrim's Precious Little Life (Scott Pilgrim #1)', '4.52', 'Five', 'catalogue/scott-pilgrims-precious-little-life-scott-pilgrim-1_987/index.html'))  
(('Rip It Up and Start Again', '4.35', 'Five', 'catalogue/rip-it-up-and-start-again_986/index.html'))  
(('Our Band Could Be Your Life: Scenes from the American Indie Underground, 1981-1991', '4.57', 'Three', 'catalogue/our-band-could-be-your-life-scenes-from-the-american-indie-underground-1981-1991_985/index.html'))  
(('Olio', '4.23', 'One', 'catalogue/olio_984/index.html'))  
(('Mesaerion: The Best Science Fiction Stories 1800-1849', '4.37', 'One', 'catalogue/mesaerion-the-best-science-fiction-stories-1800-1849_983/index.html'))  
(('Libertarianism for Beginners', '4.51', 'Two', 'catalogue/libertarianism-for-beginners_982/index.html'))  
(('It's Only the Himalayas', '4.45', 'Two', 'catalogue/its-only-the-himalayas_981/index.html'))  
>>>  
===== RESTART: D:/Python files/fetch.py =====  
>>>  
===== RESTART: D:/Python files/fetch.py =====  
>>>
```

7. Conclusion

This guide covers the process of building a web scraping tool to extract book data from the Books to Scrape website, store it in a SQLite database, and verify the correctness of the data.

Script Functionality: The script fetches, parses, and stores book data (title, price, rating, and URL).

Error Handling: The script includes basic error handling for network and database issues.

Verifying Output: You can verify the stored data by querying the SQLite database and

comparing it to the website's content.