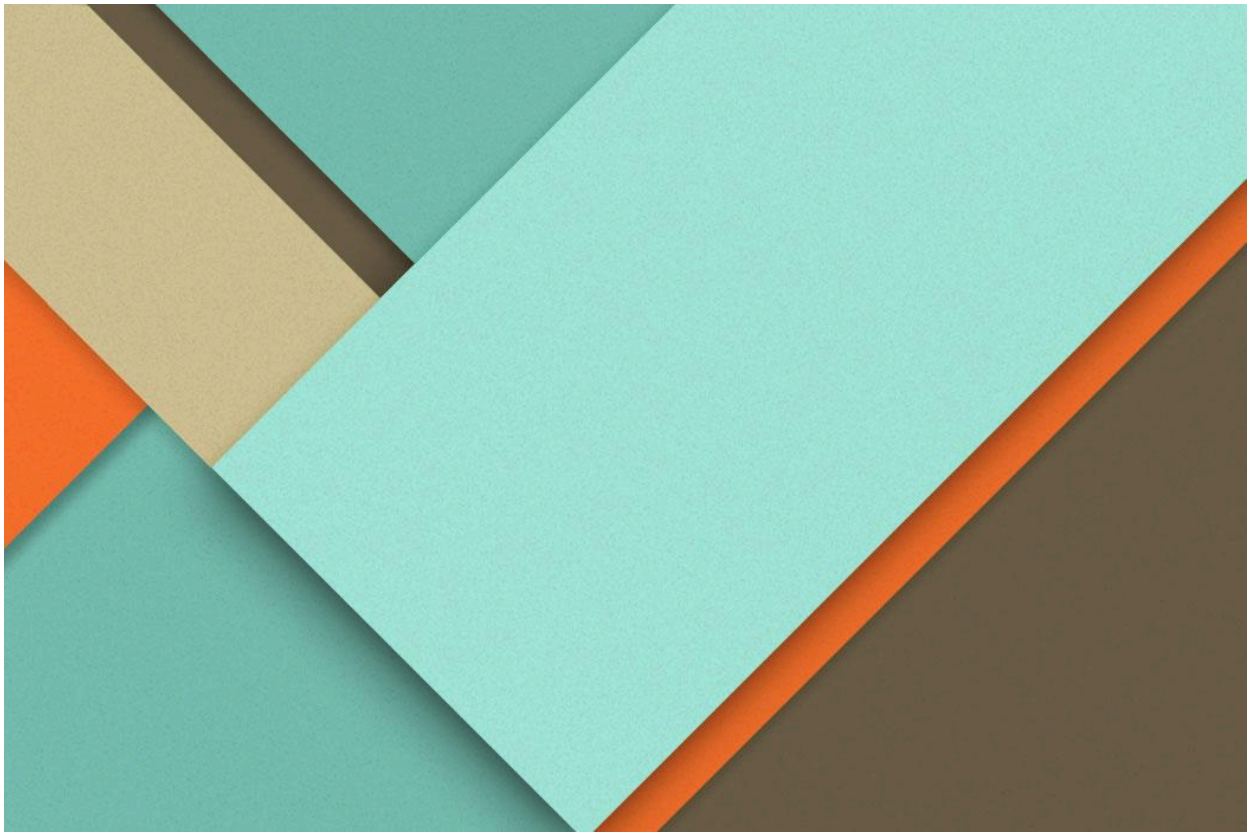

Web Scraping Tool For Data Extraction

—

Shalini Chaudhary
linishachaudhary@gmail.com

Overview

This project is a Python-based web scraping tool designed to extract tooltip information (title attributes) from websites. It efficiently collects embedded tooltip data using BeautifulSoup and requests, stores structured information in JSON, and includes error handling, retry logic , and logging mechanisms for enterprise-grade performance.

Features

1. Extracts tooltip information from elements with `title` attributes.
2. Handles network errors & retries with exponential backoff and ensures a maximum of 5 requests per 10 seconds to prevent getting blocked.
3. Logs scraping activities to scraper.log for debugging.
4. Validate extracted data and remove empty or irrelevant tooltips.
5. Saves extracted data in JSON format (tooltips_info.json)
6. Modular & Scalable with well-structured functions.
7. Supports command-line arguments to specify the target URL dynamically

Approach

To ensure a structured and scalable scraping solution, we follow a modular and iterative development approach:

I. Identify Target Data

Analyzed HTML structure of sample sites (`https://cga.nic.in/`, `https://flipkart.com/`, `https://cbse.gov.in/`, blinkit). And identified tooltip elements based on the `title` attribute.

II. Fetching Web Page Content

Used `requests.get()` to retrieve HTML content.

Implemented user-agent to avoid bot detection.Incorporated retry logic (exponential backoff) for failed requests and keep track of request times.If too many requests are sent too quickly, pause before making the next one.

III. Parsing HTML & Extracting Tooltips

Utilized BeautifulSoup to parse the HTML content.

Identified elements with tooltips (`title` attributes).

Extracted and cleaned text content from the elements.

IV. Data Validation & Quality Control

Ensured tooltips contain meaningful text (skipped empty values).

Converted extracted data into a structured JSON format.

V. Logging & Error Handling

Implemented detailed logging (`scraper.log`) to track errors, retries, and success messages.

VI. Storing & Exporting Data

Successfully stored extracted tooltip data in `tooltips_info.json`.

VII. Measure Performance

Record total execution time to track efficiency.

Output

JSON file provides structured **tooltip data** with:

- **`tag`** → The HTML tag where the tooltip is found.
- **`tooltip`** → The extracted `title` attribute.
- **`information`** → The visible text from the element.

Future Improvements

1.Multi-threading for Parallel Scraping–Improve speed by scraping multiple pages concurrently.

2.Support for JavaScript-rendered Tooltips–Use Selenium to handle dynamic content.

3.Database Integration–Store extracted tooltips in SQLite.

How to run the code

Step1: Open the terminal and write

```
(venv) linisha@MacBook-Air web_scraping % python3 -m venv venv
(venv) linisha@MacBook-Air web_scraping % source venv/bin/activate
(venv) linisha@MacBook-Air web_scraping % pip install -r req.txt
```

This will create virtual environment and install all necessary requirements to run the code

Step2: Now write , `python3 app.py --url "any website url"`

```
(venv) linisha@MacBook-Air web_scraping % python3 app.py --url "https://cga.nic.in/"
```

Step3: After step2 , tooltips_info.json will be created and check scraper.log ,will have all the logs recorded

Results

Output files:

Tooltips_info.json:

,


```
{   "tag": "a",

     "tooltip": "Automation of Annual Accounts on PFMS - Appropriation Accounts OM No 3051 Dated 30th January 2025",
     "information": "Automation of Annual Accounts on PFMS - Appropriation Accounts OM No 3051 Dated 30th January 2025"

}
```

Scraper.log

1. url : https://cga.nic.in/

2025-02-03 19:05:18,939 - INFO - Starting tooltip scraping for: https://cga.nic.in/

2025-02-03 19:05:20,035 - INFO - Successfully fetched https://cga.nic.in/ with UserAgent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36

2025-02-03 19:05:20,219 - INFO - Data successfully saved in tooltips_info.json

2025-02-03 19:05:20,220 - INFO - Scraping completed in 1.28 seconds!!

2. url: www.cbse.gov.in

2025-02-03 19:13:58,661 - INFO - Starting tooltip scraping for: https://www.cbse.gov.in/
2025-02-03 19:13:58,790 - INFO - Successfully fetched https://www.cbse.gov.in/ with UserAgent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
2025-02-03 19:13:58,802 - INFO - Data successfully saved in tooltips_info.json
2025-02-03 19:13:58,802 - INFO - Scraping completed in 0.14 seconds!!

3. url: https://blinkit.com

2025-02-03 19:17:05,145 - INFO - Starting tooltip scraping for: https://blinkit.com/
2025-02-03 19:17:05,364 - ERROR - error in fetching Url https://blinkit.com/: 403 Client Error: Forbidden for url: https://blinkit.com/
2025-02-03 19:17:05,364 - INFO - retrying in 3.430568765787178 seconds…
2025-02-03 19:17:09,047 - ERROR - error in fetching Url https://blinkit.com/: 403 Client Error: Forbidden for url: https://blinkit.com/
2025-02-03 19:17:09,047 - INFO - retrying in 5.361631644711482 seconds…
2025-02-03 19:17:14,414 - ERROR - max retries reached. Can not fetch the webpage!!
2025-02-03 19:17:14,415 - ERROR - No HTML data received. Now exiting…

4. url: https://flipkart.com

2025-02-03 19:20:14,679 - INFO - Starting tooltip scraping for: https://www.flipkart.com/

2025-02-03 19:20:22,990 - INFO - Successfully fetched https://www.flipkart.com/ with UserAgent: Mozilla/5.0 (iPad; CPU OS 14_0 like Mac OS X) AppleWebKit/537.36 (KHTML, like Gecko) Version/14.0 Mobile/15E148 Safari/537.36

2025-02-03 19:20:23,076 - INFO - Data successfully saved in tooltips_info.json

2025-02-03 19:20:23,078 - INFO - Scraping completed in 8.40 seconds!!