Scraping Data from Amazon Using Python

INDEX

Scraping Product Prices from Amazon Using Python

1. Introduction

Web scraping is a technique used to automatically extract information from websites, which is especially useful in a data-driven world where businesses rely on real-time pricing and availability information for decision-making. This project focuses on scraping product information from the Amazon website using Python and presenting the results through an interactive web interface.

The system uses the requests and BeautifulSoup libraries to collect details such as product name, price, rating, and availability, and a Streamlit-based frontend to make the tool simple and user-friendly for non-technical users.

2. Objective

The main objectives of this project are:

- To understand and implement the basics of web scraping using Python.
- To extract product details from Amazon product pages and search result pages.
- To store the scraped data in a structured CSV format for further analysis.
- To follow ethical scraping practices by limiting requests and respecting website policies.
- To provide a simple Streamlit frontend so users can enter URLs, trigger scraping, view results, and download data.

3. Tools and Technologies Used

- Programming Language: Python 3.x
- Libraries:
    - requests – send HTTP requests and fetch HTML pages.
    - BeautifulSoup – parse and navigate HTML content.
    - csv – write extracted data into CSV files.
    - time – introduce delays between requests to avoid overloading the server.
    - streamlit – build the interactive frontend web application.

- Development Environment: Visual Studio Code.

These technologies together provide a complete pipeline from data extraction to user-facing visualization and download.

4. Methodology

The project workflow is divided into several steps:

1. Sending HTTP Requests
    - The requests library is used to send GET requests to Amazon product or search result URLs.
    - Appropriate headers are added to simulate a real browser and reduce the chance of being blocked.
2. Parsing HTML Content
    - The received HTML pages are parsed using BeautifulSoup.
    - Specific tags and CSS classes are targeted to extract product title, price, rating, and availability text.
3. Multi-Product Scraping from Search Pages
    - For search result pages, all product links are identified and extracted.
    - The script iterates through these links, visiting each product page to collect detailed information.
4. Pagination Handling
    - The search URL's page parameter is modified to scrape multiple pages (for example, 1–2 pages).
    - Users can choose the number of pages to scrape from the Streamlit UI, which controls the loop.
5. Data Storage in CSV
    - Scraped data is stored in a list of dictionaries and then written to a CSV file using the csv module.
    - Columns typically include: Product Name, Price, Rating, and Availability.
6. Frontend Integration with Streamlit
    - A Streamlit application allows the user to:
        - Enter the Amazon search URL.
        - Select the number of pages to scrape.
        - Click a "Scrape Products" button to start the process.
        - View all scraped products in a structured layout.
        - Download the results as a CSV file with a "Download CSV" button.

The screenshots you provided visually confirm the flow: entering URL and page count, seeing a success message like “Scraped 10 products”, viewing each product card, and downloading the CSV.

5. Handling Anti-Scraping Measures

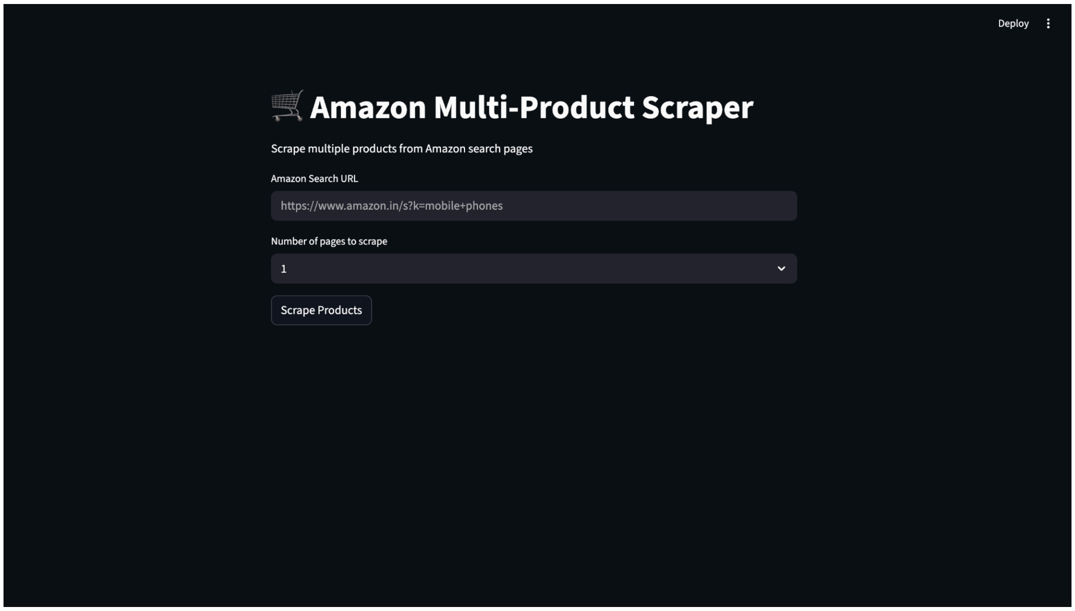
To remain ethical and reduce the risk of getting blocked, several precautions were implemented:

- Custom Request Headers:
  - A User-Agent and other headers were added to resemble genuine browser traffic.
- Delays Between Requests:
  - time.sleep() is used between page and product requests to avoid sending too many requests in a short time.
- Limited Scope:
  - Scraping is intentionally limited to a small number of pages and products per run.
- Respecting Website Policies:
  - Amazon’s terms of use and robots.txt guidelines are taken into account, and the project is used strictly for learning and personal analysis, not for large-scale commercial scraping.

These practices help demonstrate responsible scraping behavior.

6. Output

The project produces two main outputs:



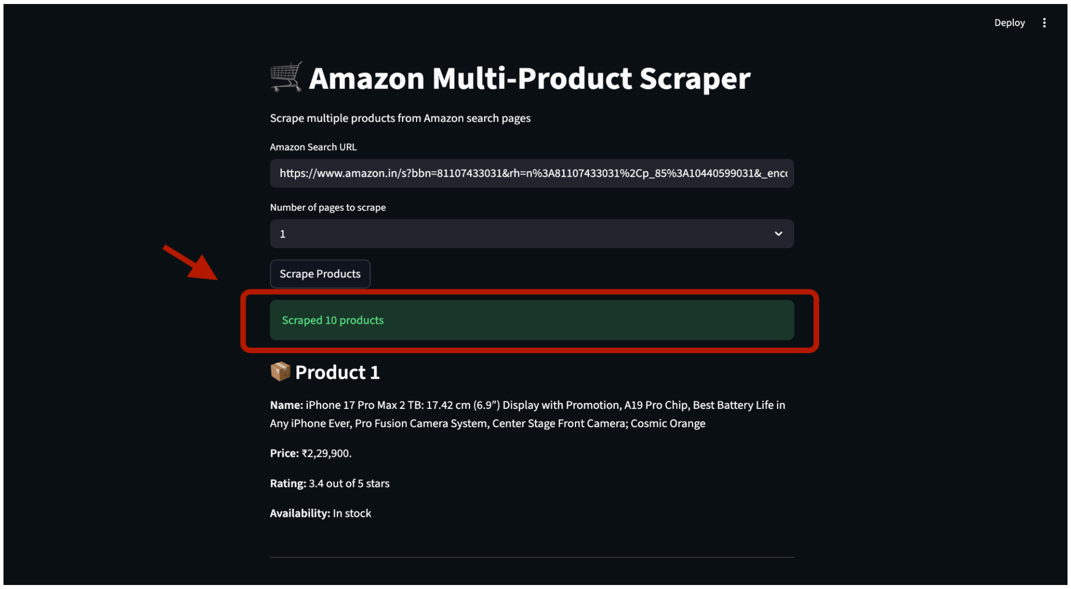
🛒 Amazon Multi-Product Scraper

Scrape multiple products from Amazon search pages

Amazon Search URL: https://www.amazon.in/s?k=mobile+phones

Number of pages to scrape: 1

- On the Streamlit Interface:
  - A success message showing how many products were scraped.
  - A list of product cards displaying:
    - Product name
    - Price
    - Rating
    - Availability status



🛒 Amazon Multi-Product Scraper

Scrape multiple products from Amazon search pages

Amazon Search URL: https://www.amazon.in/s?bbn=81107433031&rh=n%3A81107433031%2Cp_85%3A10440599031&_enc

Number of pages to scrape: 1

Scraped 10 products

📦 Product 1

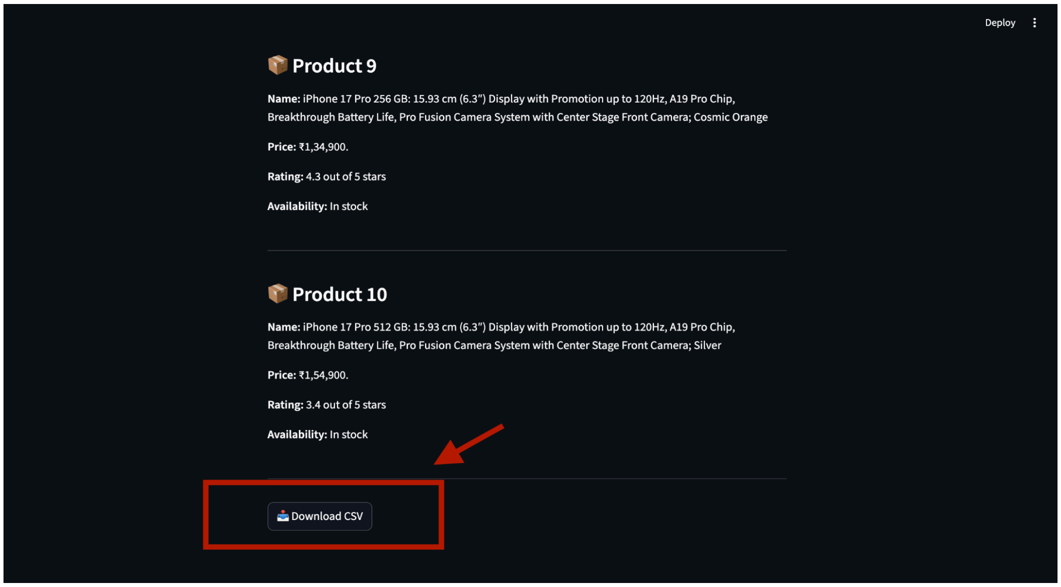
Name: iPhone 17 Pro Max 2 TB: 17.42 cm (6.9") Display with Promotion, A19 Pro Chip, Best Battery Life in Any iPhone Ever, Pro Fusion Camera System, Center Stage Front Camera; Cosmic Orange

Price: ₹2,29,900.

Rating: 3.4 out of 5 stars

Availability: In stock

- CSV File:
    - A downloadable CSV containing all scraped products with columns such as:
        - Product Name
        - Price
        - Rating
        - Availability



📦 Product 9

Name: iPhone 17 Pro 256 GB: 15.93 cm (6.3") Display with Promotion up to 120Hz, A19 Pro Chip, Breakthrough Battery Life, Pro Fusion Camera System with Center Stage Front Camera; Cosmic Orange

Price: ₹1,34,900.

Rating: 4.3 out of 5 stars

Availability: In stock

📦 Product 10

Name: iPhone 17 Pro 512 GB: 15.93 cm (6.3") Display with Promotion up to 120Hz, A19 Pro Chip, Breakthrough Battery Life, Pro Fusion Camera System with Center Stage Front Camera; Silver

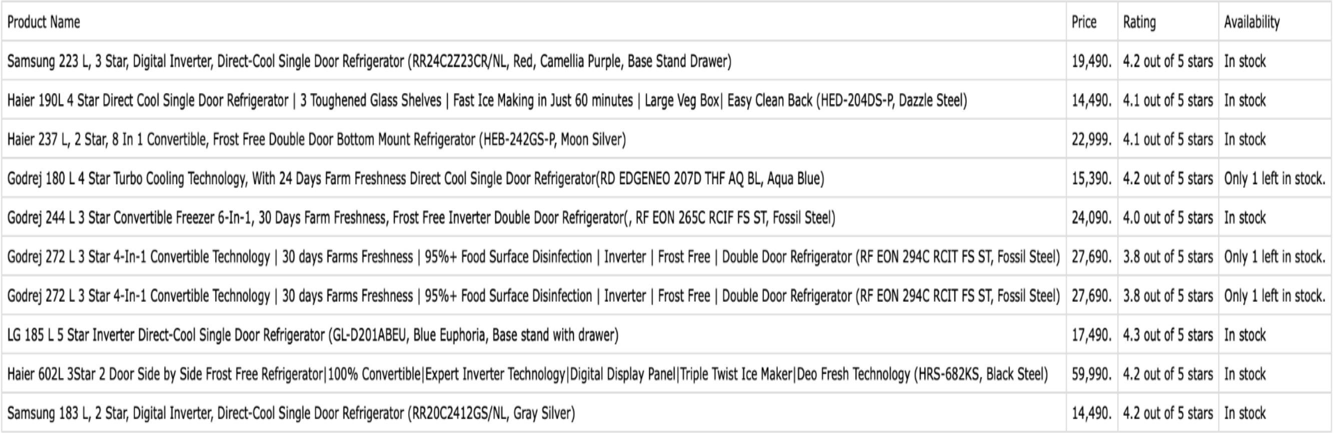
Price: ₹1,54,900.

Rating: 3.4 out of 5 stars

Availability: In stock

This structured dataset can be used for comparison, trend analysis, or importing into Excel, Power BI, or other tools

- CSV FILE -



Product Name	Price	Rating	Availability
Samsung 223 L, 3 Star, Digital Inverter, Direct-Cool Single Door Refrigerator (RR24C2Z23CR/NL, Red, Camellia Purple, Base Stand Drawer)	19,490.	4.2 out of 5 stars	In stock
Haier 190L 4 Star Direct Cool Single Door Refrigerator | 3 Toughened Glass Shelves | Fast Ice Making in Just 60 minutes | Large Veg Box| Easy Clean Back (HED-204DS-P, Dazzle Steel)	14,490.	4.1 out of 5 stars	In stock
Haier 237 L, 2 Star, 8 In 1 Convertible, Frost Free Double Door Bottom Mount Refrigerator (HEB-242GS-P, Moon Silver)	22,999.	4.1 out of 5 stars	In stock
Godrej 180 L 4 Star Turbo Cooling Technology, With 24 Days Farm Freshness Direct Cool Single Door Refrigerator(RD EDGENEO 207D THF AQ BL, Aqua Blue)	15,390.	4.2 out of 5 stars	Only 1 left in stock.
Godrej 244 L 3 Star Convertible Freezer 6-In-1, 30 Days Farm Freshness, Frost Free Inverter Double Door Refrigerator(, RF EON 265C RCIF FS ST, Fossil Steel)	24,090.	4.0 out of 5 stars	In stock
Godrej 272 L 3 Star 4-In-1 Convertible Technology | 30 days Farms Freshness | 95%+ Food Surface Disinfection | Inverter | Frost Free | Double Door Refrigerator (RF EON 294C RCIT FS ST, Fossil Steel)	27,690.	3.8 out of 5 stars	Only 1 left in stock.
Godrej 272 L 3 Star 4-In-1 Convertible Technology | 30 days Farms Freshness | 95%+ Food Surface Disinfection | Inverter | Frost Free | Double Door Refrigerator (RF EON 294C RCIT FS ST, Fossil Steel)	27,690.	3.8 out of 5 stars	Only 1 left in stock.
LG 185 L 5 Star Inverter Direct-Cool Single Door Refrigerator (GL-D201ABEU, Blue Euphoria, Base stand with drawer)	17,490.	4.3 out of 5 stars	In stock
Haier 602L 3Star 2 Door Side by Side Frost Free Refrigerator|100% Convertible|Expert Inverter Technology|Digital Display Panel|Triple Twist Ice Maker|Deo Fresh Technology (HRS-682KS, Black Steel)	59,990.	4.2 out of 5 stars	In stock
Samsung 183 L, 2 Star, Digital Inverter, Direct-Cool Single Door Refrigerator (RR20C2412GS/NL, Gray Silver)	14,490.	4.2 out of 5 stars	In stock

7. Challenges Faced

During development, the following challenges were encountered:

- Dynamic Content and Missing Fields
    - Some product details appeared as “Not Available” because Amazon loads certain elements dynamically or uses different layouts for different products.
- Anti-Scraping Mechanisms
    - Amazon employs various protections that may block or throttle frequent requests, requiring careful rate limiting.
- Inconsistent HTML Structure
    - Product cards and detail pages are not always uniform, which led to parsing errors when tags or class names changed.

These issues were mitigated using robust error handling (try–except blocks), conditional checks before accessing elements, and controlled request frequency.

8. Conclusion

This project successfully demonstrates how Python can be used for web scraping with requests and BeautifulSoup to collect product data from Amazon. It automates the extraction of key fields such as product name, price, rating, and availability, stores them in a structured CSV format, and exposes the functionality through an intuitive Streamlit frontend.

By incorporating ethical scraping practices, handling basic anti-scraping constraints, and offering CSV export capability, the project provides a solid foundation for more advanced data collection, price monitoring, and e-commerce analytics applications in the future.