Amazon Web Scraper Project

Introduction

This project involves web scraping Amazon product data using Python. The goal is to extract product titles and prices, store them in a CSV file, and track changes over time. The script automates data collection and can be extended for further analysis.

Methodology

The project follows these key steps:

1. **Web Scraping Setup:** Use requests and BeautifulSoup to fetch and parse HTML data from Amazon product pages.
2. **Data Extraction:** Identify and extract product titles and prices from the parsed HTML.
3. **Data Cleaning:** Remove unnecessary spaces and characters from the extracted data.
4. **Data Storage:** Save extracted data into a CSV file with timestamps.
5. **Automation:** Implement appending functionality to track price changes over time.

Implementation

1. Importing Required Libraries

The script uses the following Python libraries:

- BeautifulSoup (for web scraping)
- requests (for HTTP requests)
- csv (for data storage)
- datetime (for timestamps)
- pandas (for data visualization)

2. Connecting to Amazon and Fetching Data

The script sends an HTTP request to an Amazon product page using custom headers to mimic a real browser request. The response is parsed using BeautifulSoup to extract:

- **Product Title** (identified using id='productTitle')
- **Product Price** (identified using id='priceblock_ourprice')

3. Cleaning and Formatting Data

Extracted text is stripped of unnecessary characters, ensuring clean and readable output.

4. Saving Data to CSV

A CSV file (AmazonWebScraperDataset.csv) is created to store:

- Product Title
- Price

- Date of extraction

The script appends new data over time to track price fluctuations.

5. Automating Data Collection

The scraper can be scheduled to run at specific intervals using Python's scheduling libraries or a cron job to monitor price changes dynamically.

Results and Future Enhancements

The scraper successfully collects Amazon product details and stores them in a structured format. Future improvements include:

- Automating data collection using schedule or cron
- Expanding the script to scrape multiple products
- Sending email alerts for price drops
- Integrating data visualization to analyze price trends

Conclusion

This project provides a foundation for automating Amazon price tracking. It can be extended for dynamic product monitoring, making it a useful tool for competitive pricing analysis and personal shopping insights.