WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Amaze (Amazon Price Tracker)

Joy Nigrel[1], Sudhanshu Nijap[2], Sheldon Patel[3], Swaroop Dhuri[4], Nilambari Narkar[5]

[1,2,3,4] Student, Computer Engineering, Xavier Institute of Engineering, Mahim

[5] Mentor, Computer Engineering, Xavier Institute of Engineering, Mahim

ABSTRACT:

In recent years, web scraping has emerged as an indispensable asset for E-commerce enterprises. With the exponential growth of online retail ventures, the internet has become a treasure trove of data. Data extraction, particularly from dynamic websites, presents a complex challenge due to their constantly evolving nature. Data extraction, especially from dynamic websites, poses a complex challenge due to their constantly evolving nature. However, web scraping offers a vital solution for E-commerce businesses seeking to stay ahead in the competitive landscape. It serves as a powerful tool for gathering competitive intelligence, refining pricing strategies, and acquiring comprehensive product information. By leveraging web scraping techniques to extract data from competitor websites, businesses can unlock valuable insights into various aspects such as pricing dynamics, product attributes, and customer sentiments reflected in reviews.

Essentially, web scraping facilitates the automatic retrieval of large volumes of data from diverse online sources. The extracted data, typically in the form of unstructured HTML elements, undergoes transformation into structured formats suitable for analysis, such as spreadsheets or databases. This process empowers E-commerce enterprises to make informed, data-driven decisions, fostering competitiveness and fueling growth. In essence, web scraping not only enables E-commerce businesses to keep pace with industry trends but also empowers them to chart a course towards sustained success through informed decision-making.

Keywords: Web scraping,, E-commerce, Price Tracking.

Introduction

The rapid growth of e-commerce has significantly transformed consumer behavior, offering greater convenience and access to a wide range of products. Platforms like Amazon have become central to this shift, providing users with the ability to compare, purchase, and receive products with ease. However, the constantly changing nature of product prices and availability poses a challenge for consumers aiming to make informed purchasing decisions.

To address this issue, the project titled “Amaze (Amazon Price Tracker)” presents a web-based solution for tracking product prices and availability on Amazon. The system is developed using Django, a robust Python web framework, and utilizes Supabase as the backend database for real-time data storage and retrieval. To handle background tasks such as scheduled scraping and email notifications, the application integrates Celery with Redis as the message broker.

The primary objective of this project is to assist users in monitoring price fluctuations and stock updates efficiently. By automating data collection and providing timely alerts, the application enables users to make well-informed purchasing decisions. This work contributes to improving transparency in e-commerce and enhancing the overall user experience through the use of modern web technologies.

Literature Review

The domain of web scraping in e-commerce has seen considerable academic and industry interest due to its capability to extract valuable product-related data from websites efficiently. In the context of **Project Amaze**, which focuses on tracking price changes of products on Amazon,understanding the methodologies, tools, challenges, and ethical considerations surrounding web scraping is essential.

2.1 Role of Web Scraping in E-Commerce

Web scraping has become a vital tool in e-commerce for collecting product-related data directly from websites. It allows systems to extract live information such as prices, availability, and ratings without relying on third-party APIs. For a project like **Project Amaze**, which aims to track price changes on Amazon, scraping is the most effective method to gather real-time product data, especially since many e-commerce platforms restrict public API access.

2.2 Ethical and Legal Responsibilities

While web scraping is technically possible, it must be done responsibly. Ethical scraping includes respecting website rules defined in robots.txt, avoiding aggressive data requests that overload servers, and never extracting sensitive or personal user data. Ensuring that scraping activity is transparent and in line with the platform’s terms of use helps avoid legal consequences and builds trust in the tool being developed.2.3 Technical Challenges in E-commerce Scraping

2.4 Technical Challenges in Scraping Amazon

Amazon employs several anti-scraping mechanisms, including bot detection, frequent layout changes, and dynamic JavaScript content. These barriers make scraping more complex and require the use of advanced techniques like headless browsers, rotating proxies, and user-agent spoofing. A successful price tracker must be capable of adapting to such challenges to ensure stable data collection.

2.5 Tools and Libraries Used

Several Python-based tools support efficient and scalable scraping. **BeautifulSoup** is widely used for parsing static HTML, while **Selenium** is preferred for handling JavaScript-rendered content. For large-scale operations, services like **ScraperAPI** can automate tasks, manage proxies, and handle CAPTCHAs. These technologies are crucial for powering a robust backend for Project Amaze.

In summary, the literature on web scraping in e-commerce underscores its importance as a valuable tool for gathering product data, monitoring pricing dynamics, and empowering consumers with information for making informed purchasing decisions. While web scraping presents ethical and technical challenges, advances in technology and the development of innovative solutions offer promising opportunities for leveraging web scraping in e-commerce to enhance competitiveness and promote consumer welfare.

Objective

The goal of Project Amaze is to build a smart, efficient, and user-friendly Amazon price tracking system that monitors product prices in real-time and notifies users of significant changes. The specific objectives of the project are:

1. **Real-Time Amazon Price Scraping**
   - Implement a reliable web scraping mechanism using tools like BeautifulSoup, Requests, or ScraperAPI to fetch current prices of products listed on Amazon.
   - Handle dynamic content and bypass basic anti-scraping measures to ensure uninterrupted data collection.

2. **Automated Price Monitoring and Alerts**
   - Schedule regular scraping tasks using tools like cron jobs or Celery Beat to monitor price fluctuations.
   - Compare current prices with previously stored data and detect drops or changes.

3. **User Notification System**
   - Integrate an email notification system using smtp email service to alert users instantly when their tracked product drops in price or becomes available.
   - Customize notification content based on product name, new price, and user-defined thresholds.

4. **Frontend User Interface**
   - Design a clean, responsive web interface using HTML, CSS, Bootstrap, and optionally JavaScript to allow users to:
     I. Add product URLs for tracking
     II. View current prices and product details
     III. Manage tracked items

5. **Data Storage and Management**
   - Using Supabase (PostgressSq) lfor storing product details, historical price data, and user tracking preferences.
   - Define efficient schemas to support quick retrieval, validation, and updates of data.

6. **User Experience Optimization**
   - Ensure fast loading times, responsive design, and intuitive navigation.
   - It also has 8 bestsellers and today deals
   - Displays tite, price, reviews, product info of the product.

7. **Scalability and Error Handling**
   - Make the application scalable to handle multiple products and users simultaneously.
   - Implement proper exception handling and logging to maintain system stability and identify scraping failures.

Research Methodology

The research methodology for *Project Amaze* follows a structured and iterative development approach to build a real-time Amazon price tracking system. The methodology encompasses requirement gathering, technology selection, modular design, implementation, and testing phases to ensure reliability and user satisfaction.

4.1 Requirement Analysis

- Identified the need for:

- Real-time product price tracking from Amazon
- Automated alerts when product prices drop or become available
- A simple and intuitive interface for user interaction
- Considered constraints such as:
- No public Amazon API for direct price access
- Amazon's anti-bot mechanisms and dynamic page structures

4.2 Technology Selection

Based on the identified requirements and technical challenges, the following technologies were chosen:

- *Python*: Backend language for implementing scraping logic and automation
- *BeautifulSoup* and *Requests*: For HTML parsing and extracting product data
- *ScraperAPI*: To handle dynamic content, rotate IPs, and bypass scraping restrictions
- *Supabase*: Used as the backend-as-a-service (BaaS) platform for storing user data, product information, and tracking records
- *SMTP (Simple Mail Transfer Protocol)*: Configured to send automated email notifications to users
- *HTML, CSS, Bootstrap*: For building a responsive and user-friendly frontend interface

4.3 System Design

The system was architected in a modular fashion to ensure flexibility, scalability, and ease of maintenance:

- **Scraper Module**: Includes the scrape_price_only() function to fetch product title, price, and availability from Amazon pages
- **Scheduler**: Uses cron jobs to regularly trigger the update_all_prices function, ensuring real-time updates
- **Supabase Backend**: Stores user data, product URLs, current prices, and price history
- **Notification System**: Sends email alerts using SMTP when a product's price drops or restocks
- **Frontend Interface**: Allows users to add products, view tracked items, and receive alerts in a clear, responsive UI

4.4 Implementation

- Developed and validated core modules:
  I. Implemented scrape_price_only() for data extraction
  II. Built update_all_prices to handle scheduled scraping and update logic
  III. Created product cards and input forms for the frontend using Bootstrap.
- Integrated Supabase for authentication, real-time data storage, and query management
- Configured SMTP for delivering dynamic, event-based email notifications

4.5 Testing and Validation

- **Unit Testing**: Validated functionality of individual components like scraping, email notifications, and database operations
- **Integration Testing**: Ensured seamless interaction between the scraper, Supabase backend, and SMTP notification system
- **Accuracy Testing**: Manually cross-verified scraped prices with live Amazon pages
- **Performance Evaluation**: Assessed application performance during periodic scraping across multiple product URLs
- **User Feedback**: Collected input from test users to refine the UI and notification features

Result and Workflow

The development and implementation of the "Price Checker Using Amazon WebScraper" application involved a systematic process encompassing both frontend and backend components. This section provides a step-by-step overview of the procedure followed in the development and operation of the application.

1. Frontend Operation:

a) Product Search:

- Search Amazon products using URL or ASIN for accurate and efficient tracking.
- User can see bestseller and today deals section

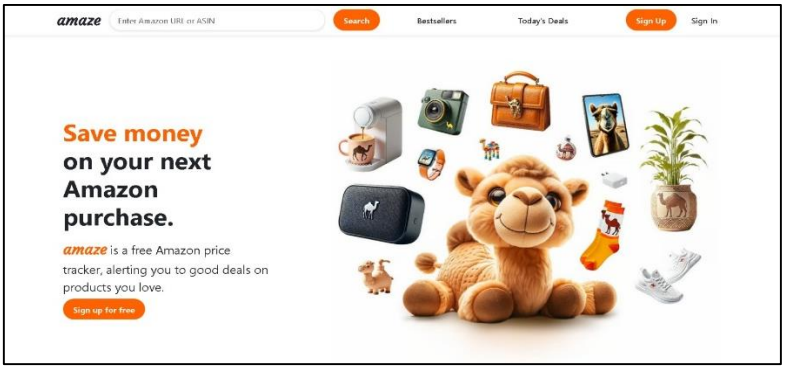

Fig 1.1 Home Page

b) Product Detail Page Display:

- The application communicates with Bright Data's webunlocker service to scrape product information from the provided URL, including product image URL, name, pricing, description, snd other relevant details.

Fig 1.2 Home Page

b) Product Retrieval:

- After retrieving the data, the application displays a product detail page containing the product image, name, current price, reviews, stock status, product info of the product.
- User can set price alerts manually or with smart shortcut buttons:

(–3%, –5%, –7%, –10%, or Any Drop).

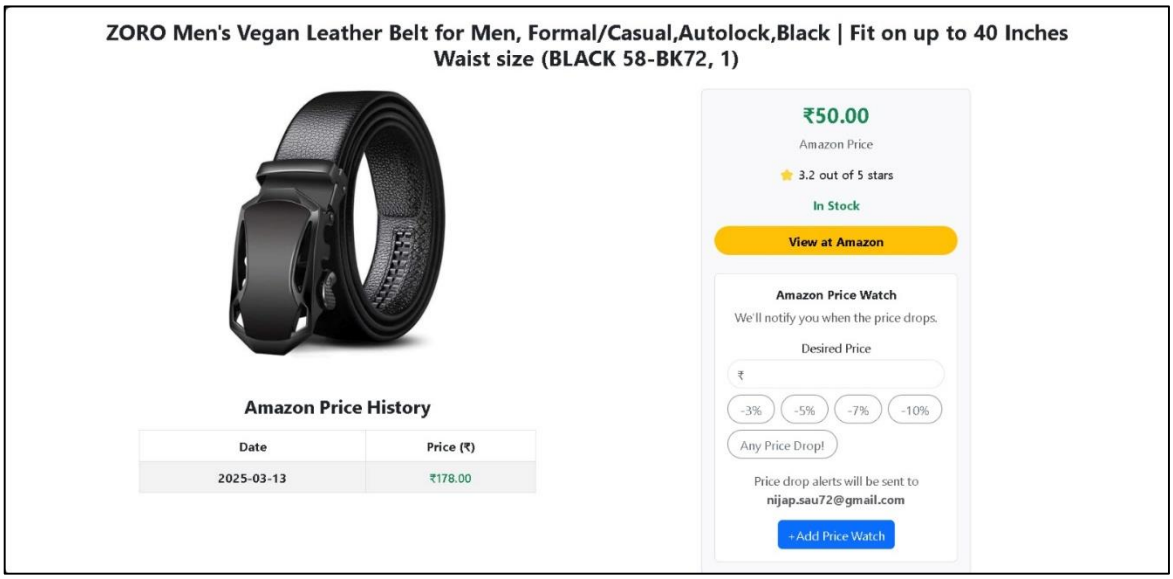

Fig 1.2 Product Detail Page

c) User Dashboard:

- A Personalized dashboard displaying all tracked products, including:

I. Current Price

II. Target Price

- "View on Amazon" button to view product on amazon

- "Remove from Price Watch" button to view product on amazon

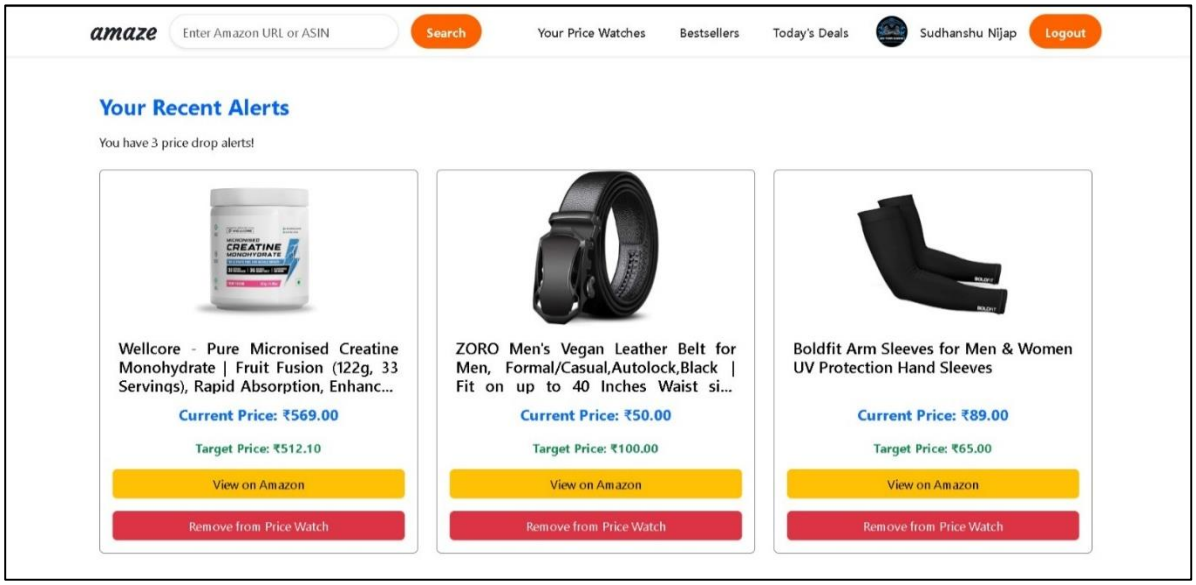

Fig 1.3 Product Track Page

2. Backend Operation:

a) Data Scraping and Processing:

- Upon user initiation, the application sends the product URL to ScraperAPI, which handles the scraping of required product details from the Amazon product page.
- The scraped data, including product information such as title, price, availability, and other relevant details, is then processed and structured into a format suitable for storage in the **Supabase** database.

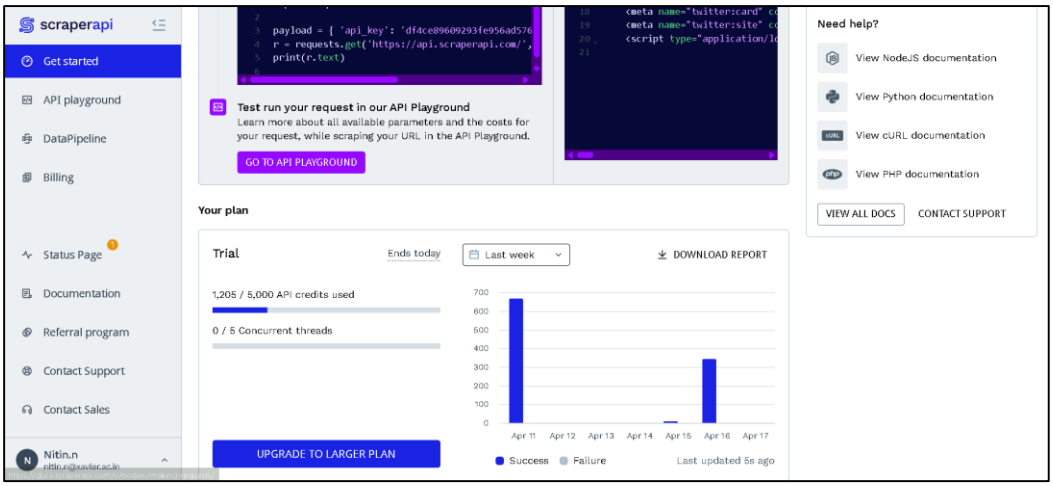

Fig 2.1 Scraper API Console

3. Data Storage and Retrieval:

- The structured data is stored in Supabase, utilizing its PostgreSQL database for efficient and real-time data management.
- The application interacts with the Supabase database using Django's ORM (Object-Relational Mapping), which simplifies the process of defining the product schema and performing database operations.
- When users request product details, the application retrieves the relevant data from Supabase and dynamically generates the product detail page for display, presenting up-to-date pricing and availability information.

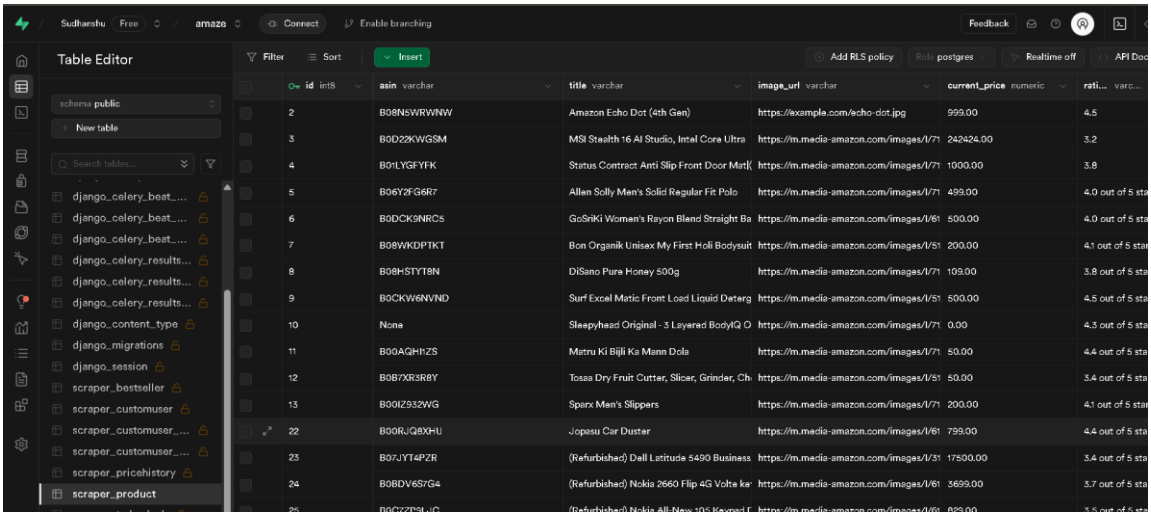

Fig 2.2 Supabase Database Console

b) Email Notification System:

- A cron job implemented using smtp mailer triggers email notifications to users when the product current price becomes lower to the target price.
- Users who have opted-in for price tracking receive timely email alerts with updates on price fluctuations.

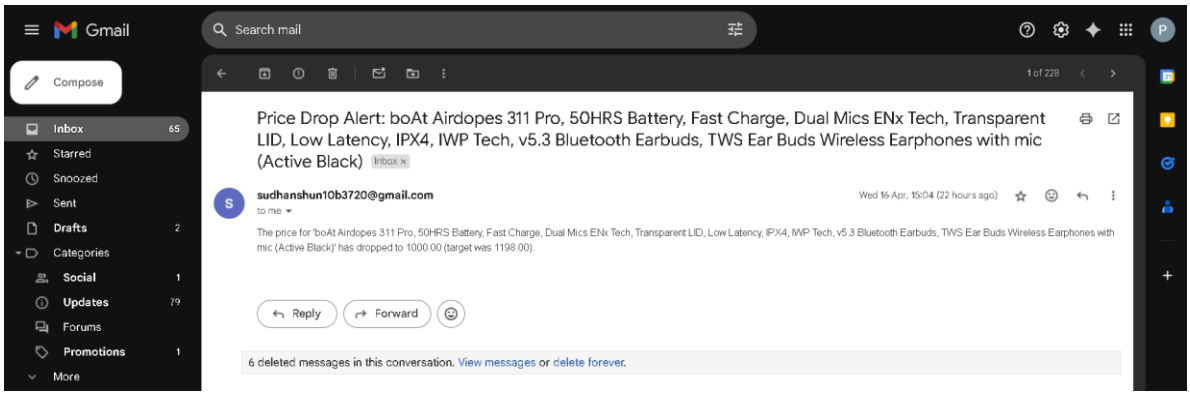

Fig 2.3 Email Notification

By following this comprehensive procedure, the **"Amaze (Amazon Price Tracker)"** application has been successfully developed offering users a valuable tool for monitoring product prices, availability, and market trends in real-time. The systematic approach ensures the reliability, scalability, and effectiveness of the application, ultimately contributing to a more transparent and consumer-friendly e-commerce marketplace.

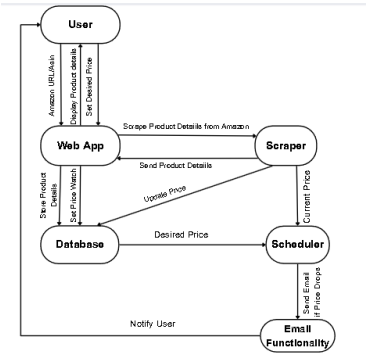

Fig 2.4 Working Diagram of Amaze

Conclusion

The development of the **"Amaze (Amazon Price Tracker)"** application marks a significant milestone in e-commerce technology. By harnessing web scraping technologies, we've empowered consumers with real-time pricing information and businesses with valuable market insights. This research underscores the importance of transparency, competition, and consumer empowerment in the e-commerce landscape.

Amaze offers an efficient solution for tracking Amazon product prices through web scraping. By automating price monitoring and providing real-time alerts, it helps users make informed purchasing decisions. Features like price history tracking and email notifications enhance the shopping experience, allowing users to take advantage of price drops.

REFERENCES:

1. "Price Checker Using Amazon WebScrapper" Project Repository By Sanskar Gouchandra. https://github.com/sanskargouchandra/pricechecker
2. Importance of Web Scraping in E-Commerce Business, By Sandeep Shreekumar, Satyan Mundke, Dr. Murlidhar Dhanawade - NCRD's Technical Review : e-Journal ISSN: 2455-166X Volume 7, Issue 1 (Jan-Dec 2022) https://www.ncrdsims.edu.in/ckfinder/userfiles/files/04(2).pdf
3. Web scraping in e-commerce- web Scraping Techniques and Applications:A Literature Review By Chaimaa Lotfi1, Swetha Srinivasan1, Myriam Ertz1[0000- 0001-9959-2779] and Imen Latrous1 1 LaboNFC, University of Quebec at Chicoutimi, 555 Boulevard de l'Université, Saguenay (QC), Canada https://www.researchgate.net/publication/367719780_Web_Scraping_Techniques_and_Applications_A_Literature_Review
4. Web Scraper Revealing Trends of Target Products and New Insights in Online Shopping Websites (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 6, 2018 By Habib Ullah(1), Zahid Ullah(2), Shahid Maqsood(3), and Abdul Hafeez https://www.researchgate.net/profile/Dr-Zahid- Ullah/publication/326181856_Web_Scraper_Revealing_ Trends_of_Target_ Products_and_New_Insights_in_Online_Shopping_Websites/links/5b 511275a6fdcc8dae2f86c1/Web-Scraper-Revealing-Trends-of-Target-Products-and-New-Insights-in-Online-Shopping-Websites.pdf
5. How to Scrape Amazon: 2024 Guide Learn how to scrape Amazon using python as well as how to use Bright Data's Scraping Browser. By Jakkie Koekemoer https://brightdata.com/blog/how-tos/how-to-scrape-amazon
6. Challenges in web scraping for e-commerce: Triggering an Email Alert Based on price comparison by web scraping Using Python by Zhang et al(2019) By Nan Zhang, Shomir Wilson, Prasenjit Mitra https://aclanthology.org/2022.lrec-1.371/