iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: V Month of publication: May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51478

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

E-Commerce Product Price Tracker using Dynamic Pricing Algorithm

Divya Khairkar[1], Shreya Dambhare[2], Om Gupta[3], Vedant Thakre[4], Nilay Raut[5], Prof. S. N. Dagadkar[6]
[1, 2, 3, 4, 5, 6]*Department of Computer Engineering, Rashtraasant Tukadoji Maharaj Nagpur University*

Abstract: The increasing popularity of e-commerce has led to a growing demand for tools that help consumers track prices and make informed purchasing decisions. This paper presents an online price tracker that enables users to monitor the prices of products across multiple e-commerce websites in real-time. The tracker utilizes web scraping techniques to extract pricing data from various e-commerce platforms and presents it in a user-friendly interface. Users can set up custom alerts to receive notifications when the price of a product drops below a certain threshold, allowing them to take advantage of discounts and save money. The price tracker also provides historical price data for each product, enabling users to analyze trends and make more informed purchasing decisions. Overall, the online price tracker provides a valuable tool for consumers to make informed decisions when shopping online.

General Terms Introduction, Web scrapping, Problem formulation, Proposed system.
Keywords: Web Scrapping, Dynamic Pricing Algorithm, Price Alerts, Problem Formulation, E-Commerce, Price Comparison

I. INTRODUCTION

In recent years, online shopping has become increasingly popular, with more and more people opting to purchase products and services online. As a result, businesses have had to adapt to this new trend and come up with innovative ways to attract and retain customers. One such way is by implementing dynamic pricing algorithms, which allow businesses to adjust their prices in real-time based on a variety of factors, such as demand, competition, and inventory levels.

An online price tracker that uses dynamic pricing algorithms can be an invaluable tool for both businesses and consumers. For businesses, it can help them stay competitive by allowing them to adjust their prices in response to changes in the market. For consumers, it can help them save money by alerting them to price drops or sales. There are several types of dynamic pricing algorithms, including rule-based, machine learning-based, and hybrid algorithms. Rule-based algorithms are relatively simple and rely on a set of pre-defined rules to determine pricing. Machine learning-based algorithms, on the other hand, use complex algorithms to analyze large amounts of data and make pricing decisions based on that analysis. Hybrid algorithms combine the best of both worlds, using a combination of pre-defined rules and machine learning to determine pricing. Regardless of the type of algorithm used, an online price tracker that uses dynamic pricing can provide valuable insights and help businesses make data-driven pricing decisions. By keeping track of price trends and adjusting prices in real-time, businesses can stay ahead of the competition and maximize their profits. And for consumers, it can help them save money and find the best deals online.

A. Motivation

People who are constantly looking for the best deal on online products to buy are what drive the E-commerce product price tracker. Online shoppers are mostly drawn to prices and discounts. The expansion of internet purchasing is also a result of price. Additionally, this will contribute to the growth of small businesses, online retailers, etc.The development of a system that uses the renowned dynamic pricing algorithm to automatically identify the convenient price of a chosen product is another goal. The system will shorten the user's search time and alert them when prices drop. The main driving factors include need, preference, cost, and many more factors. That helps customers, marketers, and product developers to advertise better products, boost corporate profits, offer quality services, and easily fill orders with suppliers using the standard quality system.

B. Objectives

1) To give clients a platform to locate the best pricing for a product while saving them time.
2) Researching the internet buying habits of consumers, market upward and downward trends, etc.
3) To offer a new algorithm that is comparable to the dynamic pricing algorithm in terms of effectiveness.
4) To create a quality website utilising web scraping and improve alerting skills using emails.

C. Scope

The project involves tracking prices and analyzing them to determine the lowest cost for a product. This price tracker helps users to understand price changes over time. As the number of internet users continues to increase, more people can benefit from using an e-commerce product price tracker. Dynamic pricing algorithms are used by many businesses to improve profitability and manage supply and demand.

Amazon, the world's largest company, updates their prices every ten minutes to ensure accurate pricing, even during periods of price volatility. Similarly, Uber uses dynamic pricing algorithms to maximize profits and match supply with demand. This approach generates revenue and supports small online stores. Users can create a Chrome plugin to track product pricing according to their specific needs.

D. Background History

A price tracker is a useful tool that allows online wholesalers, manufacturers, and retailers to monitor and compare prices from different sources. It helps to simplify the tedious task of price monitoring and provides valuable data for pricing decisions. Mr. Robert Crandall's price tracker concept is based on the dynamic pricing algorithm, which is a significant advancement in e-commerce and market revenue for meeting constantly changing supply and demand.

The price tracker uses Selenium to crawl websites and a web scraping engine based on Beautiful Soup and Request library. It checks an item's price periodically and sends an email notification to the user if there is a decrease in price. This site scraping project is similar to AntiZon, an Amazon pricing checker that helps users get the best bargain.

II. PROBLEM FORMULATION

A. Problem Definition

The problem that an online price tracker aims to solve is the difficulty of monitoring and comparing prices across multiple online sources. With so many e-commerce websites and online retailers, it can be a time-consuming and tedious task to keep track of price changes for a particular product. Additionally, pricing can be affected by various factors, such as supply and demand, competition, and promotions, making it challenging to make informed pricing or purchasing decisions. The online price tracker addresses these issues by providing a tool for monitoring and comparing prices in real-time, enabling businesses and consumers to make data-driven decisions based on the latest market trends.The problem that an online price tracker aims to solve is the difficulty of monitoring and comparing prices across multiple online sources. With so many e-commerce websites and online retailers, it can be a time-consuming and tedious task to keep track of price changes for a particular product. Additionally, pricing can be affected by various factors, such as supply and demand, competition, and promotions, making it challenging to make informed pricing or purchasing decisions. The online price tracker addresses these issues by providing a tool for monitoring and comparing prices in real-time, enabling businesses and consumers to make data-driven decisions based on the latest market trends.

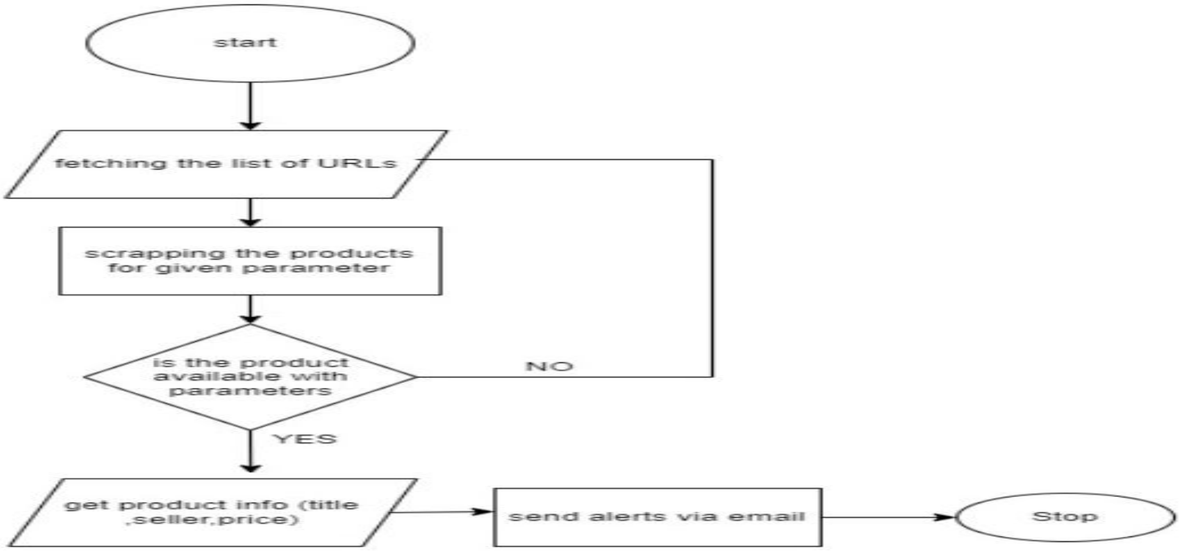

start
fetching the list of URLs
scrapping the products for given parameter
is the product available with parameters
NO
YES
get product info (title ,seller,price)
send alerts via email
Stop

Fig 1: Price Tracker Process System

III. ALGORITHM

A. Dynamic Pricing Algorithm

The dynamic pricing algorithm was first employed by Robert Crandall, the chairman of American Airlines, in the 1980s. This algorithm uses an input and instruction set to provide users with prices under specific conditions. The best dynamic pricing algorithm criteria include maximizing profits, reducing customer turnover, attracting competitors' customers, improving customer experience, and maintaining loyalty.

This algorithm is used by many businesses, including Amazon, Uber, and Airbnb, to optimize prices, generate profitability, increase demand, and boost sales. Five essential stages are required for an effective dynamic pricing configuration: setting business goals, creating a price plan, selecting a pricing strategy, and creating pricing guidelines. The goal of dynamic pricing is to maximize anticipated revenues by adjusting prices dynamically over a limited sales horizon. While realized demand is observed over time, the underlying functional relationship between price and mean demand rate, known as the demand function or demand curve, is unknown.

B. Algorithm

The development team also developed a new similar comparison algorithm to address the shortcomings of the dynamic pricing algorithm, which is currently in use. These are the steps:

1) Install the essential libraries, including price parser, beautiful soup, and pandas.
2) Managing them in CSV or JSON after obtaining the destination URL.
3) Using pandas, reading the CSV file.
4) Lopping over the URLs when scraping the prices.
5) The ability to extract HTML and enter the price from it.
6) A beautiful soup object and locate a price element.
7) To extract the price as a float for comparison with the alert price, use the price-parser library.
8) Push notifications will be sent to the user.

IV. PROPOSED SYSTEM

The proposed system is a real-time search engine that allows users to find the best products across multiple e-commerce websites, saving them time and money by eliminating the need for manual filtering. It is designed to provide customers with an enhanced shopping experience, while also helping e-commerce companies identify pricing errors on their website and offer better service to their customers. The system is automated, user-friendly, and easy to use.

To achieve maximum efficiency, the system employs highly efficient and focused dynamic web crawlers that filter products based on predefined parameters integrated into the system.

The products are then ranked according to the dynamic pricing algorithm, ensuring that only the best products are available to customers. This system is similar to other e-commerce systems, where users can search for products and get desired results within a few seconds.

A. Flow Diagram

The operational flow of the proposed system is outlined in Figure 1. Once the user inputs a product into the search box, the system commences product evaluation. The system checks if the product that is scraped matches the category being searched. Each product has specific attributes like price, discount, brand, and popularity. The system utilizes a dynamic pricing algorithm to rank products based on these characteristics, presenting the user with the product that has the lowest price, the best brand, and the highest popularity.

B. Data Flow Diagram

The proposed system's data flow is illustrated in Figure 2. When the user enters a link, it goes through a scraping process that generates an output suitable for the user. This process involves the flow of data between various modules such as user details management, product details management, website URL management, and tracking report management to ensure efficient site operation.

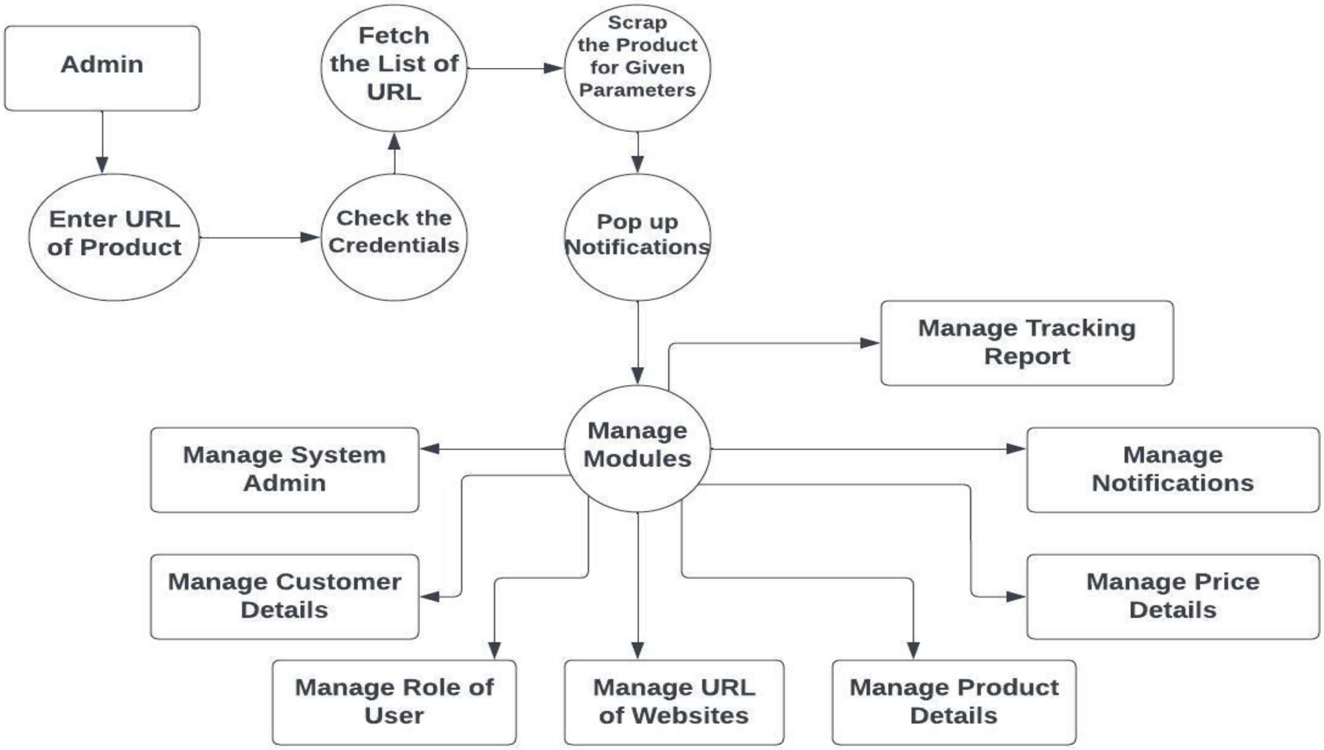


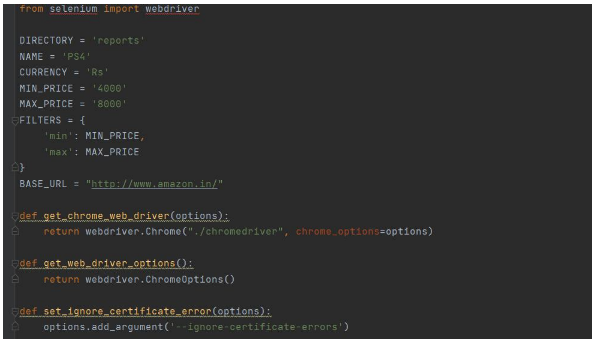
Fig 2 : DFD of Price Tracker

V. METHODOLOGY

1) The first step of price tracking involves scanning a website for product details such as title, price, and stock information. Some price trackers also gather additional information like reviews and photos.
2) In the second stage, web scraping is used to find the best prices. The price tracker keeps track of product information while scraping prices and ranking websites accordingly.
3) The necessary operations, including obtaining HTML, locating the price element, and extracting the price from the beautiful soup object, are performed.
4) The user is notified via email when the price is extracted with the help of the price-parser library. The user can view this data on a dashboard or in a list format.
5) Although dynamically generated websites, which rely on JavaScript, are more challenging to scan than static HTML pages, some price trackers still struggle with this task.

VI. IMPLEMENTED MODULE

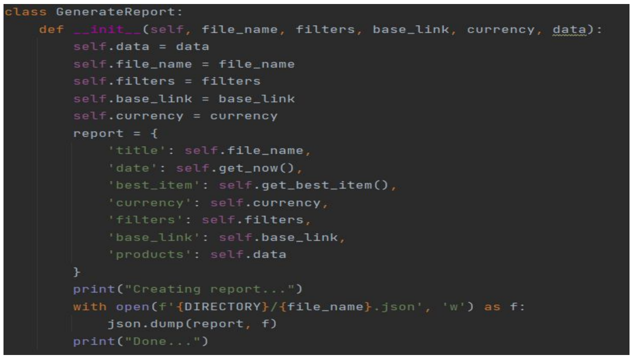
1) Configuration of the file for the products.



```
from selenium import webdriver

DIRECTORY = 'reports'
NAME = 'PS4'
CURRENCY = 'Rs'
MIN_PRICE = '4000'
MAX_PRICE = '8000'
FILTERS = {
    'min': MIN_PRICE,
    'max': MAX_PRICE
}
BASE_URL = "http://www.amazon.in/"

def get_chrome_web_driver(options):
    return webdriver.Chrome("./chromedriver", chrome_options=options)

def get_web_driver_options():
    return webdriver.ChromeOptions()

def set_ignore_certificate_error(options):
    options.add_argument('--ignore-certificate-errors')
```

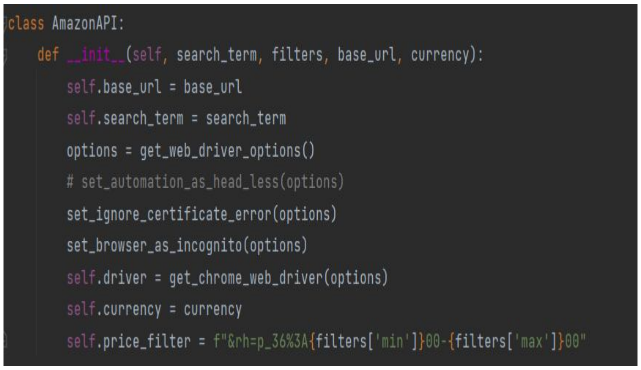
Screenshot no.1

2) Building the scrapper



```
class GenerateReport:
    def __init__(self, file_name, filters, base_link, currency, data):
        self.data = data
        self.file_name = file_name
        self.filters = filters
        self.base_link = base_link
        self.currency = currency
        report = {
            'title': self.file_name,
            'date': self.get_now(),
            'best_item': self.get_best_item(),
            'currency': self.currency,
            'filters': self.filters,
            'base_link': self.base_link,
            'products': self.data
        }
        print("Creating report...")
        with open(f'{DIRECTORY}/{file_name}.json', 'w') as f:
            json.dump(report, f)
        print("Done...")
```

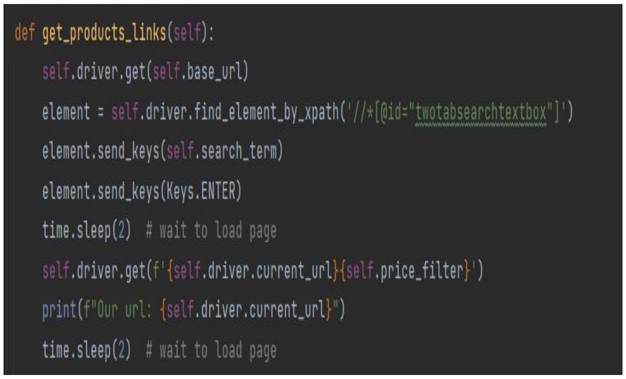
Screenshot no.2



```
class AmazonAPI:
    def __init__(self, search_term, filters, base_url, currency):
        self.base_url = base_url
        self.search_term = search_term
        options = get_web_driver_options()
        # set_automation_as_head_less(options)
        set_ignore_certificate_error(options)
        set_browser_as_incognito(options)
        self.driver = get_chrome_web_driver(options)
        self.currency = currency
        self.price_filter = f"&rh=p_36%3A{filters['min']}00-{filters['max']}00"
```

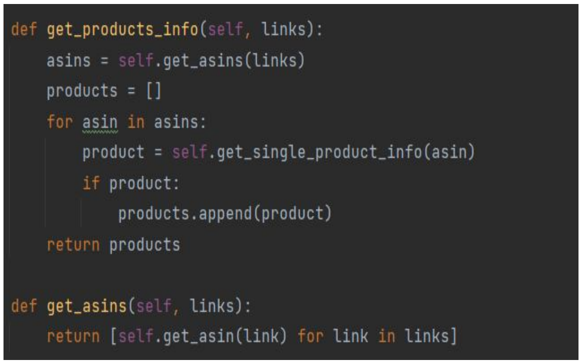
Screenshot no.3 Setting parameters

3) Scraping



```
def get_products_links(self):
    self.driver.get(self.base_url)
    element = self.driver.find_element_by_xpath('//*[@id="twotabsearchtextbox"]')
    element.send_keys(self.search_term)
    element.send_keys(Keys.ENTER)
    time.sleep(2)  # wait to load page
    self.driver.get(f'{self.driver.current_url}{self.price_filter}')
    print(f"Our url: {self.driver.current_url}")
    time.sleep(2)  # wait to load page
```

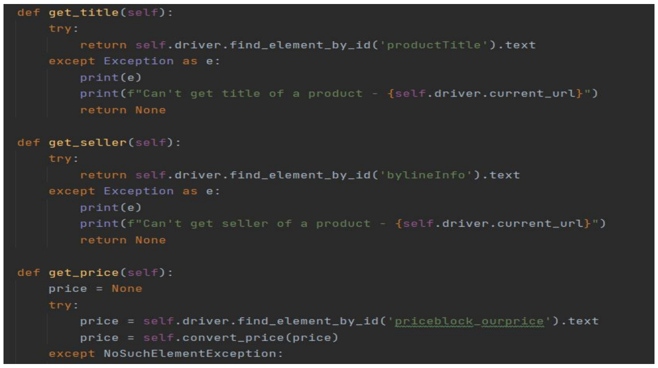
Screenshot no.4. Getting the product links



```
def get_products_info(self, links):
    asins = self.get_asins(links)
    products = []
    for asin in asins:
        product = self.get_single_product_info(asin)
        if product:
            products.append(product)
    return products

def get_asins(self, links):
    return [self.get_asin(link) for link in links]
```

Screenshot no.5. Getting product information and asins

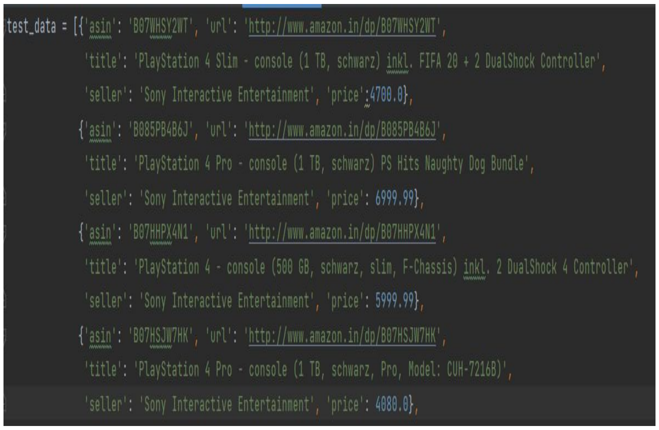
4) Getting the product name, seller, and price



```
def get_title(self):
    try:
        return self.driver.find_element_by_id('productTitle').text
    except Exception as e:
        print(e)
        print(f"Can't get title of a product - {self.driver.current_url}")
        return None

def get_seller(self):
    try:
        return self.driver.find_element_by_id('bylineInfo').text
    except Exception as e:
        print(e)
        print(f"Can't get seller of a product - {self.driver.current_url}")
        return None

def get_price(self):
    price = None
    try:
        price = self.driver.find_element_by_id('priceblock_ourprice').text
        price = self.convert_price(price)
    except NoSuchElementException:
```

Screenshot no.6.

5) After running the test data collected



```
test_data = [{'asin': 'B07WHSY2WT', 'url': 'http://www.amazon.in/dp/B07WHSY2WT',
              'title': 'PlayStation 4 Slim - console (1 TB, schwarz) inkl. FIFA 20 + 2 DualShock Controller',
              'seller': 'Sony Interactive Entertainment', 'price':4700.0},
             {'asin': 'B085PB4B6J', 'url': 'http://www.amazon.in/dp/B085PB4B6J',
              'title': 'PlayStation 4 Pro - console (1 TB, schwarz) PS Hits Naughty Dog Bundle',
              'seller': 'Sony Interactive Entertainment', 'price': 6999.99},
             {'asin': 'B07HHPX4N1', 'url': 'http://www.amazon.in/dp/B07HHPX4N1',
              'title': 'PlayStation 4 - console (500 GB, schwarz, slim, F-Chassis) inkl. 2 DualShock 4 Controller',
              'seller': 'Sony Interactive Entertainment', 'price': 5999.99},
             {'asin': 'B07HSJW7HK', 'url': 'http://www.amazon.in/dp/B07HSJW7HK',
              'title': 'PlayStation 4 Pro - console (1 TB, schwarz, Pro, Model: CUH-7216B)',
              'seller': 'Sony Interactive Entertainment', 'price': 4080.0},
```

Screenshot no.7.

VII. FUTURE SCOPE

To ensure effective ranking of pages, it is essential to design new algorithms that meet our standards for ranking difficulties and provide relevant and use-specific information quickly and efficiently [3].

Cross-validation can be used with price and marketing strategy charts to determine pricing and marketing strategies for competitors' products. However, this approach may sometimes result in short-notice warnings that could potentially irritate customers, which is a disadvantage that we aim to overcome in the future [5].

If the price of a product decreases after it has sold out, customers may become dissatisfied. Thus, we need to develop a faster and more appropriate algorithm to address this issue in the future [8].

VIII. RESULT

The proposed system is a Real-Time Search Engine where users will find best products among multiple ecommerce websites. It will reduce time and money wasted in manual filtering and will provide customers a better shopping experience. Moreover, it will also help ecommerce companies to identify price errors on their website and help provide better service to customer. It will be automated, user-friendly and easy to use. The system uses highly efficient and focused dynamic web crawlers which will filter out products at first level based on the predefined set of parameters integrated into the system. The scraped products will pass through dynamic pricing algorithm where each product will be ranked accordingly. Filtration process during these stages will be highly rigorous and this will result in availability of best products to the customers.

REFERENCES

[1] Piyush Rawal, Priyansh Gupta, Shubham Gaur and Faraj Chishti “Comparison Website for Online Shopping” International Journal of Research in Engineering, IT and Social Sciences,5 May 2020

[2] Urvesh Rathod, Aruna Pavate, Vabhav Pssatil “Product Ranked Based Search Engine for E-commerce” International Conference for Convergence in Technology,06-08 April 2018.

[3] Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry, “The PageRank Citation Ranking: Bringing Order to the Web”, Tech. Rep., pp.1-17, 1998

[4] Andrea Harch, Holger Kett, Antee Weisbecker “Mining E-commerce data from E-shop Websites”, (IEEE), 03 December 2015.

[5] Shin-Yen Huang “Developing a product pricing and marketing analysis system on the website”, International Conference on Applied System Innovation (ICASI), (IEEE) 24 July 2017.

[6] Rida Yaqoob, Sanaa, Muhmmad Haris, Munam Ali Shah “The Price Scraping Bot Threat on E-commerce Store Using Custom XPATH Technique”, International Conference on Automation and Computing (ICAC) ,(IEEE) 26 November 2021.

[7] Anish Gupta, Priya, “Focused Web Crawlers and its Approaches”, International Conference on Futuristic trend in Computational Analysis and Knowledge Management IEEE 13 July 2015.

[8] Omar and Besbes, Assaf, Zeevi, “Dynamic Pricing Without Knowing the Demand Function: Risk Bounds and Near-Optimal Algorithms”, Institute for Operations Research and the Management Sciences (INFORMS),24 April 2009