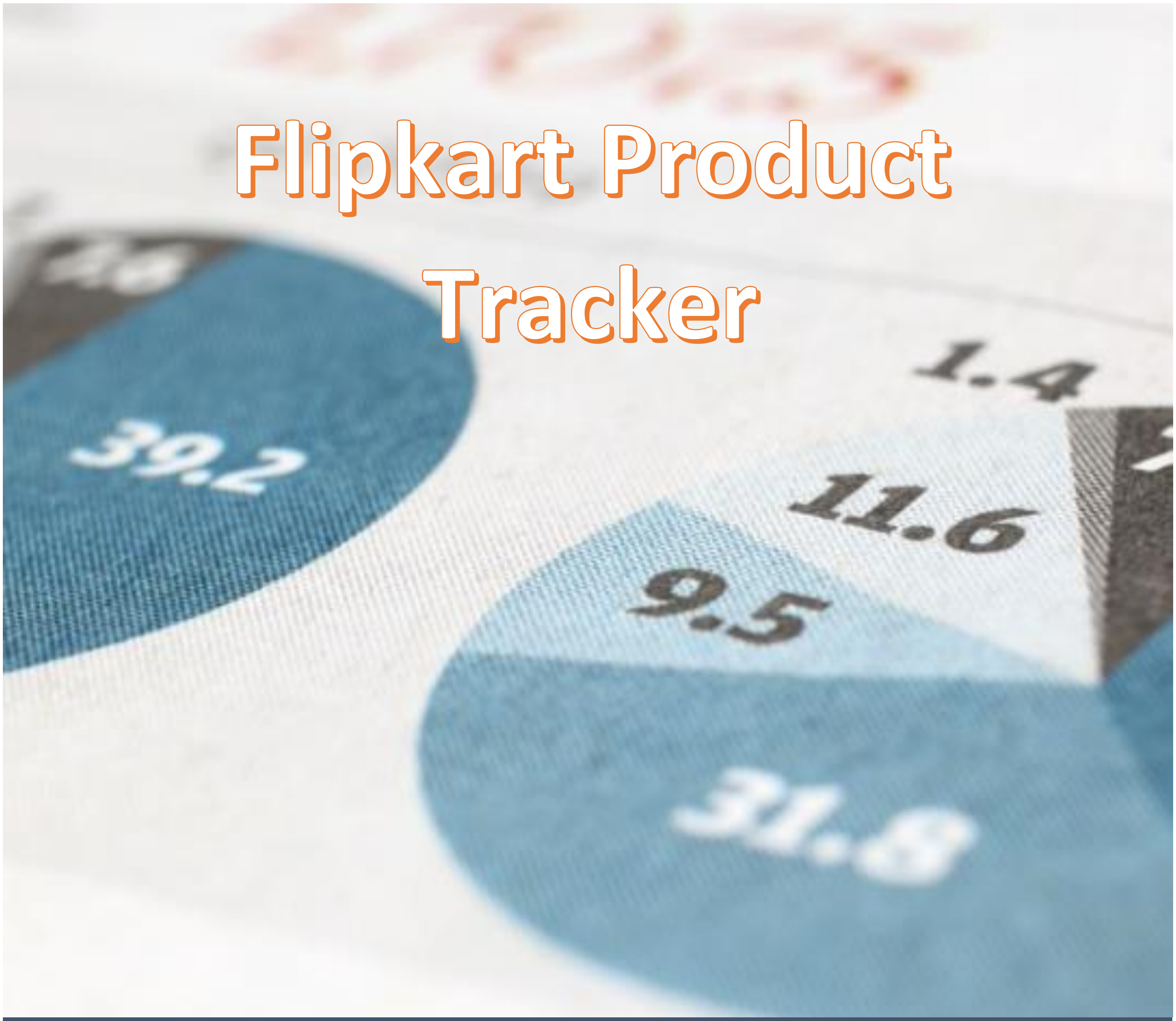


Flipkart Product Tracker

Team Members

Siddharth Choudhary

Himanshu Choudhary

Sneha R Iyer

Himanshu Sharma

Team: Mitsukeru 見つける   7/9/21   Python is Cool, Kids

Contents

1. Introduction

This project will introduce a tool for web scrapping ecommerce websites, built using python libraries and packages. This project is built to determine changes in the attributes of a commodity in the e-commerce website for a user. There are two major functional components of this package. The first is scrapping an ecommerce website for product price, availability, and other attributes. In our case we chose flipkart as the ecommerce platform to scrape data. Then an updates section to check for any change in the attributes and send the user an email to alert them. The entire package is then wrapped up in GUI.

1.1. Problem definition

The product tracker tool should maintain a database of the products along with the attributes that need to be monitored. There should be a way for it to check each product's current page, check for changes in the attributes and if needed, it should update the latest values and alert the user about the changes. There should be a convenient way for the user to add and remove products from the database in the form of a GUI.

1.2. Importance

The period of 2015 to 2020 saw the rise of internet penetration in India from 19% to 50% and with it increased the use of ecommerce websites. A way to monitor a list of products without going to each product's page and checking the changes is very convenient in such a time. People can use it to keep an eye on the prices or availability of their favourite gadgets, shoes or other products they are interested in. If the idea is further expanded to include multiple ecommerce platforms, users can get the best price for their product without browsing these sites and looking for their products.

1.3. Motivation

The motivation behind the project was to use python and it's powerful libraries to create something cool and get new experience. The idea of using python for some web application was attractive. First we thought about using web scraping to make a twitter bot or something similar and after some discussion the idea evolved to it's current state. The usability of our project in the life of students like us was also a big factor in deciding to make this tool.

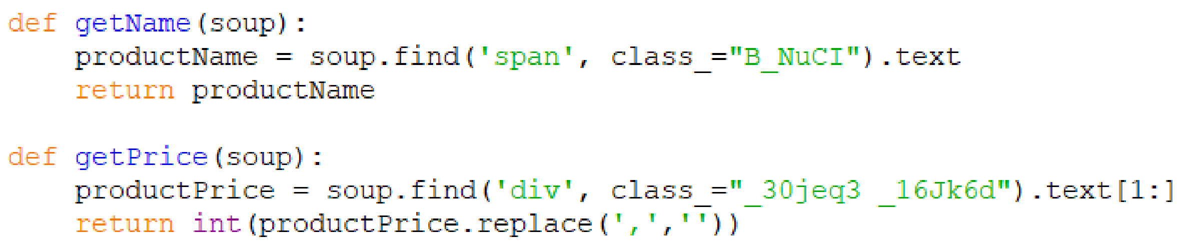
2. Implementation Details

This section describes the libraries and the packages used in developing this project. The primary focus will be on the implementation of web scrapping using beautiful soup and requests, cleaning and sorting data using pandas, email alerts through smtplib, and then GUI packaging using Tkinter. Some of the problems faced while implementing these aspects are also discussed here.

2.1. Web Scrapping using Requests and Beautiful Soup

The first step towards creating a web scrapper involves the basic understanding of HTML code of a particular website and extracting the relevant tags to scrap data from. However, before we start implementing the web scrapper in our system, we first must send a get request to a webserver for a particular page, then use BeautifulSoup to parse the data into a "soup" which we can easily navigate through to get our attributes.

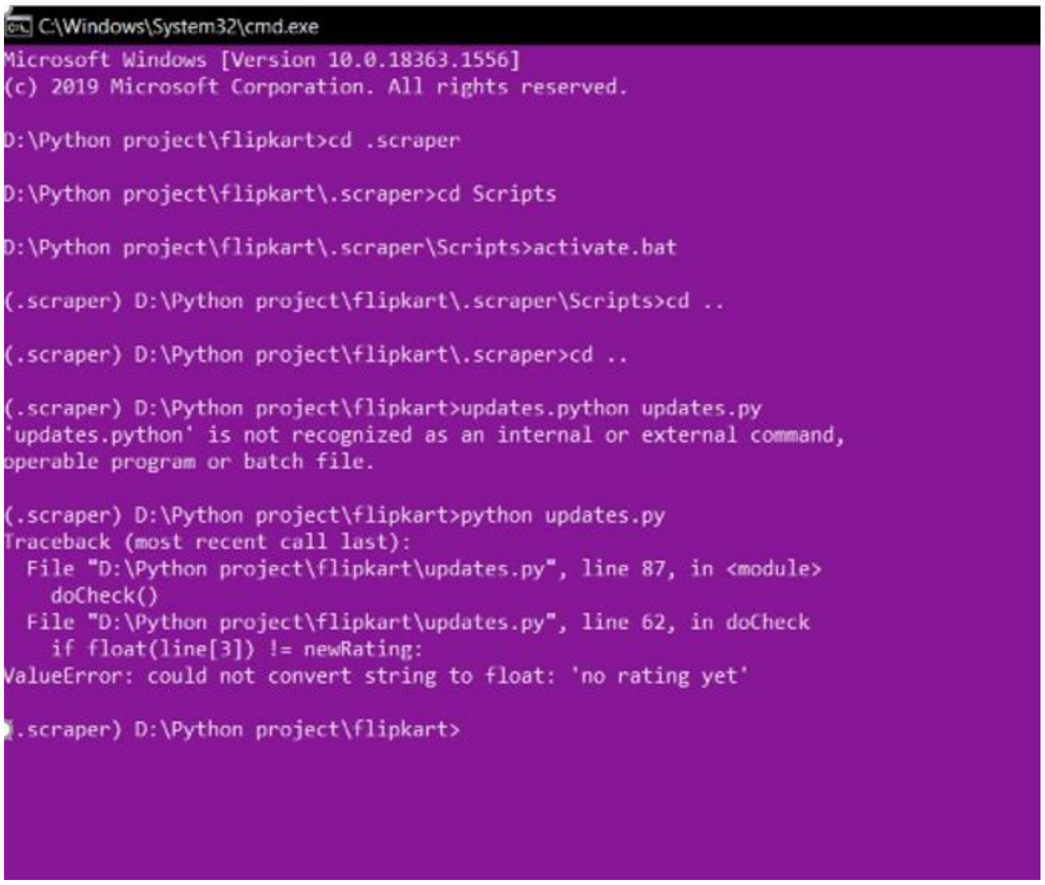
To get attributes like name, price, etc from the soup we inspect the page element and find the common tag and class names where those attributes reside in every product page. For example, *functions.py* called in the *GUI.py* file identifies the unique tags in the HTML code of that webpage and picks details of Name, Price and Availability (Fig. 1).



```
def getName(soup):
    productName = soup.find('span', class_="B_NuCI").text
    return productName

def getPrice(soup):
    productPrice = soup.find('div', class_="_30jeq3 _16Jk6d").text[1:]
    return int(productPrice.replace(',',''))
```

Fig. 1. Calling *functions* file to pick up attribute data.

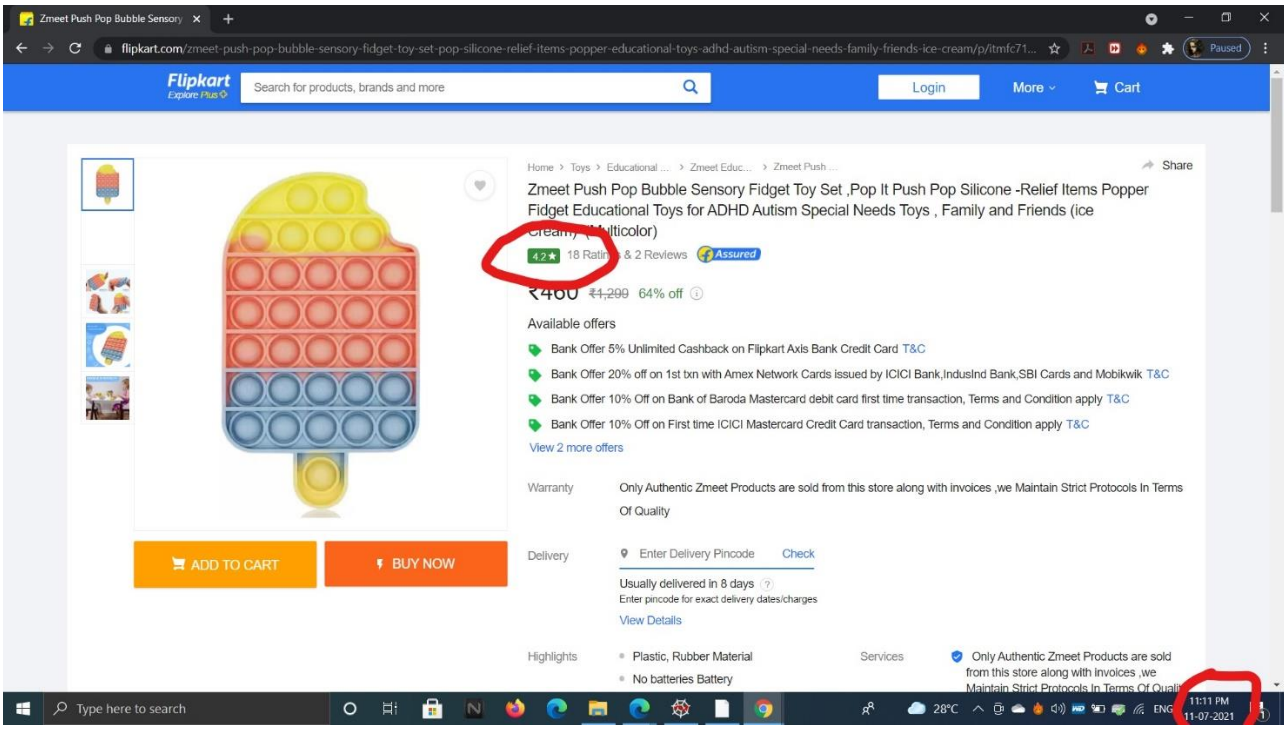
Some of the plausible errors and problems occurs in products which do not have rating (hence our search does not give any output). The following Fig. 2 shows the error encountered for ratings part.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Python project\flipkart>cd .scraper

D:\Python project\flipkart\.scraper>cd Scripts

D:\Python project\flipkart\.scraper\Scripts>activate.bat

(.scraper) D:\Python project\flipkart\.scraper\Scripts>cd ..

(.scraper) D:\Python project\flipkart\.scraper>cd ..

(.scraper) D:\Python project\flipkart>updates.python updates.py
'updates.python' is not recognized as an internal or external command,
operable program or batch file.

(.scraper) D:\Python project\flipkart>python updates.py
Traceback (most recent call last):
  File "D:\Python project\flipkart\updates.py", line 87, in <module>
    doCheck()
  File "D:\Python project\flipkart\updates.py", line 62, in doCheck
    if float(line[3]) != newRating:
ValueError: could not convert string to float: 'no rating yet'

(.scraper) D:\Python project\flipkart>
```

Fig. 2. Error in identifying and converting rating attribute from string to float.

One problem that resulted in us omitting the rating attribute was the inexplicable back and forth change in ratings of certain products within seconds. Disabling use of cache & cookies didn't resolve this issue. Figures 3 and 4 show the difference in rating of the same product at around same time.


Zmeet Push Pop Bubble Sensory

flipkart.com/zmeet-push-pop-bubble-sensory-fidget-toy-set-pop-silicone-relief-items-popper-educational-toys-adhd-autism-special-needs-family-friends-ice-cream/p/itmfc71...

Flipkart Explore Plus

Search for products, brands and more

Login More Cart

Home › Toys › Educational ... › Zmeet Educ... › Zmeet Push ...

Share

Zmeet Push Pop Bubble Sensory Fidget Toy Set ,Pop It Push Pop Silicone -Relief Items Popper Fidget Educational Toys for ADHD Autism Special Needs Toys , Family and Friends (ice Cream) (Multicolor)

4.2★ 18 Ratings & 2 Reviews Assured

₹460 ~~₹1,299~~ 64% off

Available offers

- Bank Offer 5% Unlimited Cashback on Flipkart Axis Bank Credit Card T&C
- Bank Offer 20% off on 1st txn with Amex Network Cards issued by ICICI Bank,IndusInd Bank,SBI Cards and Mobikwik T&C
- Bank Offer 10% Off on Bank of Baroda Mastercard debit card first time transaction, Terms and Condition apply T&C
- Bank Offer 10% Off on First time ICICI Mastercard Credit Card transaction, Terms and Condition apply T&C

View 2 more offers

Warranty Only Authentic Zmeet Products are sold from this store along with invoices ,we Maintain Strict Protocols In Terms Of Quality

Delivery Enter Delivery Pincode Check

Usually delivered in 8 days

Enter pincode for exact delivery dates/charges

View Details

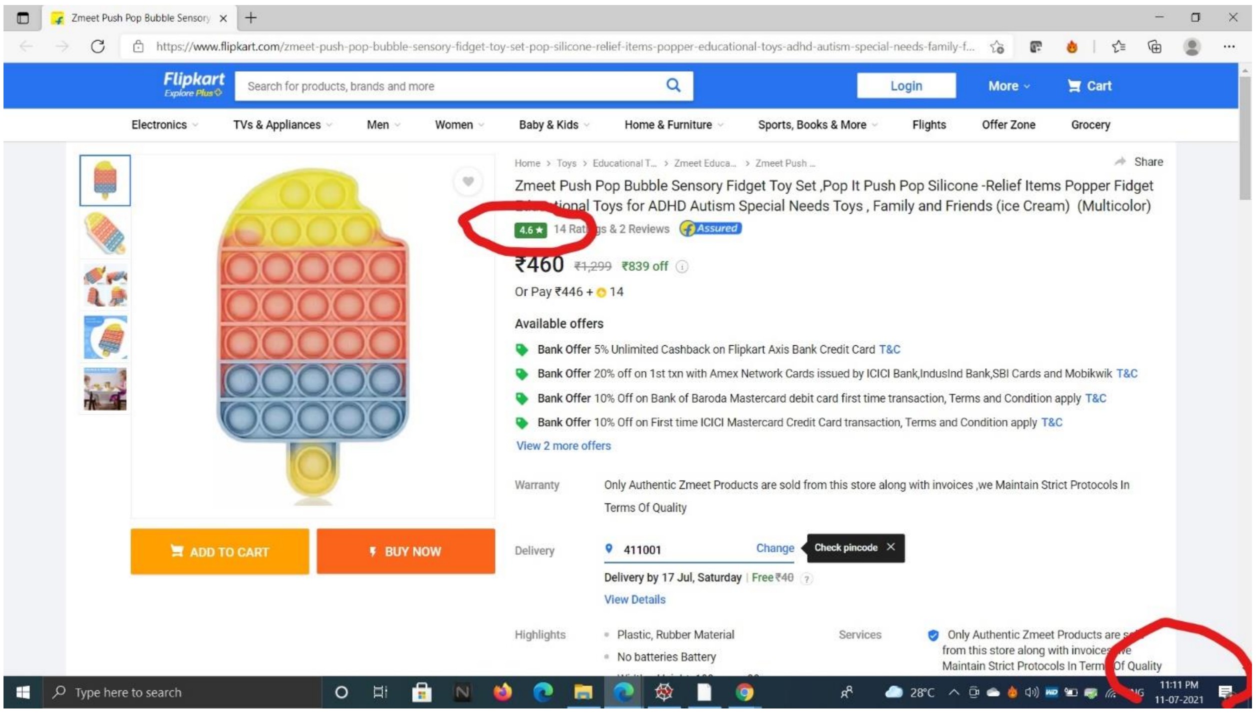
Highlights
- Plastic, Rubber Material
- No batteries Battery

Services Only Authentic Zmeet Products are sold from this store along with invoices ,we Maintain Strict Protocols In Terms Of Quality

ADD TO CART BUY NOW

Type here to search 28°C ENG 11:11 PM 11-07-2021

Fig. 3. Product page displaying rating 4.2


Zmeet Push Pop Bubble Sensory

https://www.flipkart.com/zmeet-push-pop-bubble-sensory-fidget-toy-set-pop-silicone-relief-items-popper-educational-toys-adhd-autism-special-needs-family-f...

Flipkart Explore Plus

Search for products, brands and more

Login More Cart

Electronics TVs & Appliances Men Women Baby & Kids Home & Furniture Sports, Books & More Flights Offer Zone Grocery

Home › Toys › Educational T... › Zmeet Educa... › Zmeet Push ...

Share

Zmeet Push Pop Bubble Sensory Fidget Toy Set ,Pop It Push Pop Silicone -Relief Items Popper Fidget Educational Toys for ADHD Autism Special Needs Toys , Family and Friends (ice Cream) (Multicolor)

4.6★ 14 Ratings & 2 Reviews Assured

₹460 ~~₹1,299~~ ₹839 off

Or Pay ₹446 + 14

Available offers

- **Bank Offer** 5% Unlimited Cashback on Flipkart Axis Bank Credit Card T&C
- **Bank Offer** 20% off on 1st txn with Amex Network Cards issued by ICICI Bank,IndusInd Bank,SBI Cards and Mobikwik T&C
- **Bank Offer** 10% Off on Bank of Baroda Mastercard debit card first time transaction, Terms and Condition apply T&C
- **Bank Offer** 10% Off on First time ICICI Mastercard Credit Card transaction, Terms and Condition apply T&C

View 2 more offers

Warranty Only Authentic Zmeet Products are sold from this store along with invoices ,we Maintain Strict Protocols In Terms Of Quality

Delivery 411001 Change Check pincode

Delivery by 17 Jul, Saturday | Free ₹40

View Details

Highlights
- Plastic, Rubber Material
- No batteries Battery

Services Only Authentic Zmeet Products are sold from this store along with invoices ,we Maintain Strict Protocols In Terms Of Quality

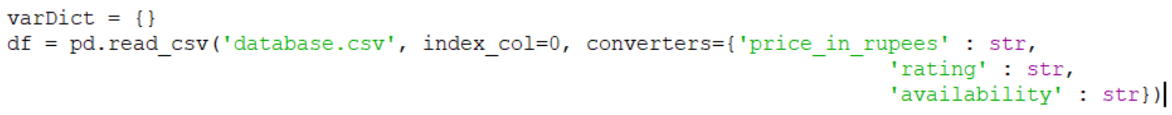
ADD TO CART BUY NOW

Type here to search 28°C 11:11 PM 11-07-2021

Fig. 4. Same product page displaying product rating 4.6

2.2. Pandas and smtplib to handle the database and email alerts respectively

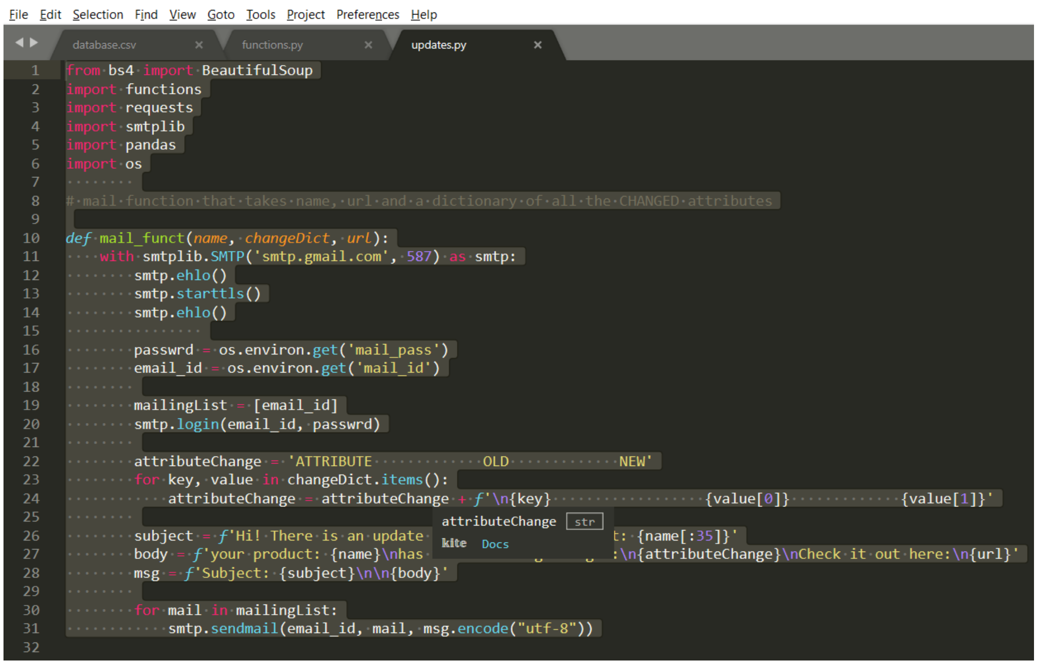
Pandas was mainly introduced for the purpose of handling the data collected through the web scrapping in section 2.1 in a better fashion, allowing databases to get back in shape after deleting entries. Pandas dataframes allowed control of datatype of attributes, the same is shown through the block of code in Fig. 3.



```
varDict = {}
df = pd.read_csv('database.csv', index_col=0, converters={'price_in_rupees' : str,
                                                          'rating' : str,
                                                          'availability' : str})
```

Fig. 5. Using Pandas to read columns relevant datatypes.

Moving on to smtiplib in file *updates.py*. In this file the code looks for difference in the price and availability of all products compared to the most recently fetched data in the doCheck() function. If any changes are detected it creates a dictionary with keys as attribute names and old and new values. Using smtplib an email alert is sent to the user. The email alerts the user about the name of the product and the relevant changes in its attributes. Figure 6 shows the mail function and how it formats the body using the dictionary of changes.


File Edit Selection Find View Goto Tools Project Preferences Help
database.csv
functions.py
updates.py
attributeChange str
kite Docs

```
from bs4 import BeautifulSoup
import functions
import requests
import smtplib
import pandas
import os

#-mail function that takes name, url and a dictionary of all the CHANGED attributes

def mail_funct(name, changeDict, url):
    with smtplib.SMTP('smtp.gmail.com', 587) as smtp:
        smtp.ehlo()
        smtp.starttls()
        smtp.ehlo()

        passwrd = os.environ.get('mail_pass')
        email_id = os.environ.get('mail_id')

        mailingList = [email_id]
        smtp.login(email_id, passwrd)

        attributeChange = 'ATTRIBUTE              OLD              NEW'
        for key, value in changeDict.items():
            attributeChange = attributeChange + f'\n{key}                  {value[0]}             {value[1]}'

        subject = f'Hi! There is an update [illegible]t: {name[:35]}'
        body = f'your product: {name}\nhas [illegible]:\n{attributeChange}\nCheck it out here:\n{url}'
        msg = f'Subject: {subject}\n\n{body}'

        for mail in mailingList:
            smtp.sendmail(email_id, mail, msg.encode("utf-8"))
```

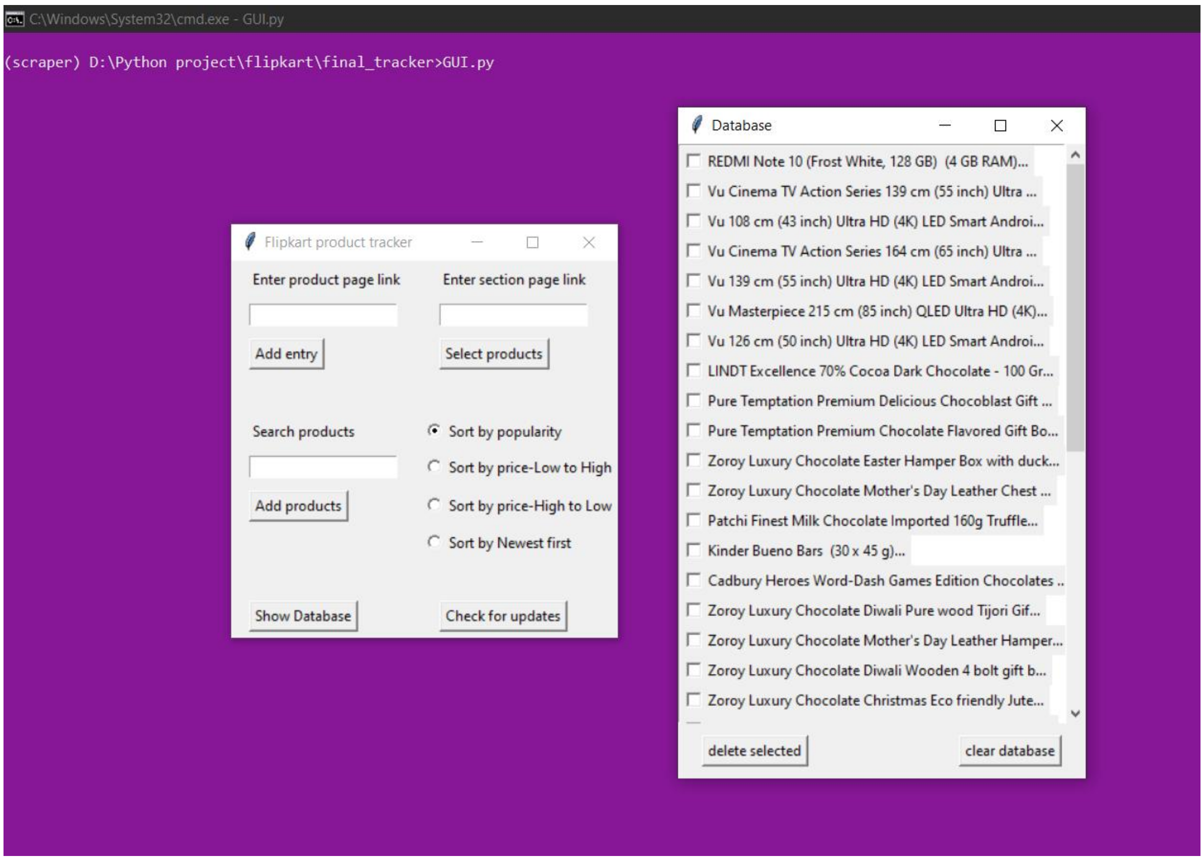
Fig. 6. Setting up and sending updates message via email using smtplib.

Requests, Beautiful Soup, pandas, smtplib helped us prepare the main working skeleton of our project. However, we decided that packaging it in GUI would give us a better finish, in terms of creating a convenient blackbox for the user. The next section discusses the GUI developed using Tkinter.

2.3. Tkinter for GUI

Tkinter helps in creating the interface required to carry out the desired functions.

Figure 6 shows the tracker GUI and the new window created by clicking the 'show database' button, allowing the user to either select the items to be deleted or clear the entire database.



```
C:\Windows\System32\cmd.exe - GUI.py
(scraper) D:\Python project\flipkart\final_tracker>GUI.py
```

Database

- ☐ REDMI Note 10 (Frost White, 128 GB) (4 GB RAM)...
- ☐ Vu Cinema TV Action Series 139 cm (55 inch) Ultra ...
- ☐ Vu 108 cm (43 inch) Ultra HD (4K) LED Smart Androi...
- ☐ Vu Cinema TV Action Series 164 cm (65 inch) Ultra ...
- ☐ Vu 139 cm (55 inch) Ultra HD (4K) LED Smart Androi...
- ☐ Vu Masterpiece 215 cm (85 inch) QLED Ultra HD (4K)...
- ☐ Vu 126 cm (50 inch) Ultra HD (4K) LED Smart Androi...
- ☐ LINDT Excellence 70% Cocoa Dark Chocolate - 100 Gr...
- ☐ Pure Temptation Premium Delicious Chocoblast Gift ...
- ☐ Pure Temptation Premium Chocolate Flavored Gift Bo...
- ☐ Zoroy Luxury Chocolate Easter Hamper Box with duck...
- ☐ Zoroy Luxury Chocolate Mother's Day Leather Chest ...
- ☐ Patchi Finest Milk Chocolate Imported 160g Truffle...
- ☐ Kinder Bueno Bars (30 x 45 g)...
- ☐ Cadbury Heroes Word-Dash Games Edition Chocolates ..
- ☐ Zoroy Luxury Chocolate Diwali Pure wood Tijori Gif...
- ☐ Zoroy Luxury Chocolate Mother's Day Leather Hamper...
- ☐ Zoroy Luxury Chocolate Diwali Wooden 4 bolt gift b...
- ☐ Zoroy Luxury Chocolate Christmas Eco friendly Jute...

Flipkart product tracker

Enter product page link

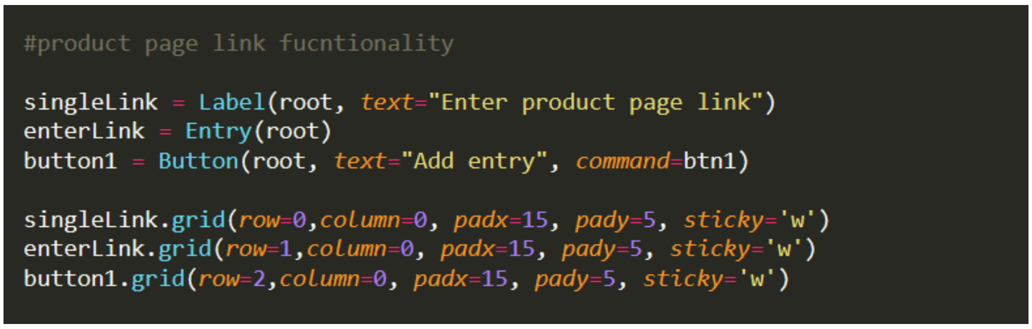
Enter section page link

Search products

[✓ Sort by popularity] Sort by price-Low to High Sort by price-High to Low Sort by Newest first

Fig. 7. GUI with the database displayed.

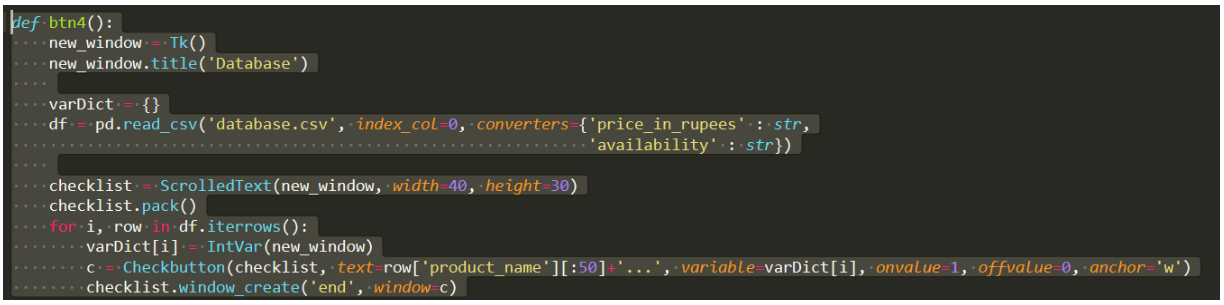
Each little section has a label, entry and button widget (search product has the radio buttons for extra functionality). These are placed using the grid system. Figure 8 displays the code block creating and placing the add single entry functionality on the main screen. Padding has been added wherever necessary to give it better shape.



```python
#product page link fucntionality

singleLink = Label(root, text="Enter product page link")
enterLink = Entry(root)
button1 = Button(root, text="Add entry", command=btn1)

singleLink.grid(row=0,column=0, padx=15, pady=5, sticky='w')
enterLink.grid(row=1,column=0, padx=15, pady=5, sticky='w')
button1.grid(row=2,column=0, padx=15, pady=5, sticky='w')
```

Fig. 8. Creating and placing widgets.

One problem area was to realize the display database functionality to show the list of products in a dynamic length list of checkboxes to make deletion easier. After seeking some help and guidance, a scrolled text widget was implemented along with a dictionary to contain all the checklist variables to know which ones to delete. Figure 9 shows the same.



```python
def btn4():
    new_window = Tk()
    new_window.title('Database')
    
    varDict = {}
    df = pd.read_csv('database.csv', index_col=0, converters={'price_in_rupees' : str,
                                                              'availability' : str})
    
    checklist = ScrolledText(new_window, width=40, height=30)
    checklist.pack()
    for i, row in df.iterrows():
        varDict[i] = IntVar(new_window)
        c = Checkbutton(checklist, text=row['product_name'][:50]+'...', variable=varDict[i], onvalue=1, offvalue=0, anchor='w')
        checklist.window_create('end', window=c)
```

Fig. 9. Dynamic checkbox list of products implemented.

Check for updates button calls doCheck() function from functions.py to go through the entire database and alert the user of possible changes in price and availability. Its functionality can be checked by either waiting for the prices to change overtime or editing the attributes manually in database.csv.

3. Conclusions

- The Flipkart product tracker tool can scrape Fipkart products for their attributes and store it. Adding products can be done by 3 methods- by product page link, section link and keyword search (which also has radio buttons for prioritised searches). Entries can be deleted.

- Initially we started with a basic functionality in terms of individual parts, like separate web scrapping and email alerts. These were then integrated which posed their own problems that needed solving.

- The overall look was finished using Tkinter to create GUI. The GUI was created from a user point of view so that there is ease of use and the user does not have to bother with the code behind it.

- Multiple tests were performed on different systems to check its functionality. We were able to conclude that it works and gives desired results.

- Due to certain time constraints and lack of specific knowledge we were unable to include rating in the list of attributes in the database. This is one of the gaps which we plan to take care at a later stage.

- Many other interesting features can also be introduced in the next version of this package, like support for other e-commerce websites like amazon, shopclues, snapdeal, etc.