from tkinter import \*

from bs4 import BeautifulSoup

import requests

from difflib import get\_close\_matches

import webbrowser

from collections import defaultdict

import random

root = Tk()

root.geometry("320x150")

class Price\_compare:

def \_\_init\_\_(self, master):

self.var = StringVar()

self.var\_ebay = StringVar()

self.var\_flipkart = StringVar()

self.var\_amzn = StringVar()

label = Label(master, text='Enter the product')

label.grid(row=0, column=0, padx=(30, 10), pady=30)

entry = Entry(master, textvariable=self.var)

entry.grid(row=0, column=1)

button\_find = Button(master, text='Find', bd=4, command=self.find)

button\_find.grid(row=1, column=1, sticky=W, pady=8)

def find(self):

self.product = self.var.get()

self.product\_arr = self.product.split()

self.n = 1

self.key = ""

self.title\_flip\_var = StringVar()

self.title\_amzn\_var = StringVar()

self.variable\_amzn = StringVar()

self.variable\_flip = StringVar()

for word in self.product\_arr:

if self.n == 1:

self.key = self.key + str(word)

self.n += 1

else:

self.key = self.key + '+' + str(word)

self.window = Toplevel(root)

self.window.title('Price Comparison Engine')

label\_title\_flip = Label(self.window, text='Flipkart Title:')

label\_title\_flip.grid(row=0, column=0, sticky=W)

label\_flipkart = Label(self.window, text='Flipkart price (Rs):')

label\_flipkart.grid(row=1, column=0, sticky=W)

entry\_flipkart = Entry(self.window, textvariable=self.var\_flipkart)

entry\_flipkart.grid(row=1, column=1, sticky=W)

label\_title\_amzn = Label(self.window, text='Amazon Title:')

label\_title\_amzn.grid(row=3, column=0, sticky=W)

label\_amzn = Label(self.window, text='Amazon price (Rs):')

label\_amzn.grid(row=4, column=0, sticky=W)

entry\_amzn = Entry(self.window, textvariable=self.var\_amzn)

entry\_amzn.grid(row=4, column=1, sticky=W)

self.price\_flipkart(self.key)

self.price\_amzn(self.key)

try:

self.variable\_amzn.set(self.matches\_amzn[0])

except:

self.variable\_amzn.set('Product not available')

try:

self.variable\_flip.set(self.matches\_flip[0])

except:

self.variable\_flip.set('Product not available')

option\_amzn = OptionMenu(self.window, self.variable\_amzn, \*self.matches\_amzn)

option\_amzn.grid(row=3, column=1, sticky=W)

lab\_amz = Label(self.window, text='Not this? Try out suggestions by clicking on the title')

lab\_amz.grid(row=3, column=2, padx=4)

option\_flip = OptionMenu(self.window, self.variable\_flip, \*self.matches\_flip)

option\_flip.grid(row=0, column=1, sticky=W)

lab\_flip = Label(self.window, text='Not this? Try out suggestions by clicking on the title')

lab\_flip.grid(row=0, column=2, padx=4)

button\_search = Button(self.window, text='Search', command=self.search, bd=4)

button\_search.grid(row=2, column=2, sticky=E, padx=10, pady=4)

button\_amzn\_visit = Button(self.window, text='Visit Site', command=self.visit\_amzn, bd=4)

button\_amzn\_visit.grid(row=4, column=2, sticky=W)

button\_flip\_visit = Button(self.window, text='Visit Site', command=self.visit\_flip, bd=4)

button\_flip\_visit.grid(row=1, column=2, sticky=W)

def price\_flipkart(self, key):

url\_flip = 'https://www.flipkart.com/search?q=' + str(

key) + '&marketplace=FLIPKART&otracker=start&as-show=on&as=off'

map = defaultdict(list)

self.headers = {

'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_10\_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36'}

source\_code = requests.get(url\_flip, headers=self.headers)

soup = BeautifulSoup(source\_code.text, "html.parser")

self.opt\_title\_flip = StringVar()

home = 'https://www.flipkart.com'

for block in soup.find\_all('div', {'class': '\_2kHMtA'}):

title, price, link = None, 'Currently Unavailable', None

for heading in block.find\_all('div', {'class': '\_4rR01T'}):

title = heading.text

for p in block.find\_all('div', {'class': '\_30jeq3 \_1\_WHN1'}):

price = p.text[1:]

for l in block.find\_all('a', {'class': '\_1fQZEK'}):

link = home + l.get('href')

map[title] = [price, link]

user\_input = self.var.get().title()

self.matches\_flip = get\_close\_matches(user\_input, map.keys(), 20, 0.1)

self.looktable\_flip = {}

for title in self.matches\_flip:

self.looktable\_flip[title] = map[title]

try:

self.opt\_title\_flip.set(self.matches\_flip[0])

self.var\_flipkart.set(self.looktable\_flip[self.matches\_flip[0]][0] + '.00')

self.link\_flip = self.looktable\_flip[self.matches\_flip[0]][1]

except IndexError:

self.opt\_title\_flip.set('Product not found')

def price\_amzn(self, key):

url\_amzn = 'https://www.amazon.in/s/ref=nb\_sb\_noss\_2?url=search-alias%3Daps&field-keywords=' + str(key)

# Faking the visit from a browser

headers = {

'authority': 'www.amazon.com',

'pragma': 'no-cache',

'cache-control': 'no-cache',

'dnt': '1',

'upgrade-insecure-requests': '1',

'user-agent': 'Mozilla/5.0 (X11; CrOS x86\_64 8172.45.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.64 Safari/537.36',

'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9',

'sec-fetch-site': 'none',

'sec-fetch-mode': 'navigate',

'sec-fetch-dest': 'document',

'accept-language': 'en-GB,en-US;q=0.9,en;q=0.8',

}

map = defaultdict(list)

home = 'https://www.amazon.in'

proxies\_list = ["128.199.109.241:8080", "113.53.230.195:3128", "125.141.200.53:80", "125.141.200.14:80",

"128.199.200.112:138", "149.56.123.99:3128", "128.199.200.112:80", "125.141.200.39:80",

"134.213.29.202:4444"]

proxies = {'https': random.choice(proxies\_list)}

source\_code = requests.get(url\_amzn, headers=headers)

plain\_text = source\_code.text

self.opt\_title = StringVar()

self.soup = BeautifulSoup(plain\_text, "html.parser")

# print(self.soup)

# print(self.soup.find\_all('div', {'class': 'sg-col-inner'}))

for html in self.soup.find\_all('div', {'class': 'sg-col-inner'}):

title, link, price = None, None, None

for heading in html.find\_all('span', {'class': 'a-size-medium a-color-base a-text-normal'}):

title = heading.text

for p in html.find\_all('span', {'class': 'a-price-whole'}):

price = p.text

for l in html.find\_all('a', {

'class': 'a-link-normal s-underline-text s-underline-link-text s-link-style a-text-normal'}):

link = home + l.get('href')

if title and link:

map[title] = [price, link]

user\_input = self.var.get().title()

self.matches\_amzn = get\_close\_matches(user\_input, list(map.keys()), 20, 0.01)

self.looktable = {}

for title in self.matches\_amzn:

self.looktable[title] = map[title]

self.opt\_title.set(self.matches\_amzn[0])

self.var\_amzn.set(self.looktable[self.matches\_amzn[0]][0] + '.00')

self.product\_link = self.looktable[self.matches\_amzn[0]][1]

def search(self):

amzn\_get = self.variable\_amzn.get()

self.opt\_title.set(amzn\_get)

product = self.opt\_title.get()

price, self.product\_link = self.looktable[product][0], self.looktable[product][1]

self.var\_amzn.set(price + '.00')

flip\_get = self.variable\_flip.get()

flip\_price, self.link\_flip = self.looktable\_flip[flip\_get][0], self.looktable\_flip[flip\_get][1]

self.var\_flipkart.set(flip\_price + '.00')

def visit\_amzn(self):

webbrowser.open(self.product\_link)

def visit\_flip(self):

webbrowser.open(self.link\_flip)

if \_\_name\_\_ == "\_\_main\_\_":

c = Price\_compare(root)

root.title('Price Comparison website')

root.mainloop()