PAGE No.
DATE / /

My Name Vaishali Ramdas Salve
Batch No : Ds2311
Sir :- Shwetank Mishra
Date :- 9-2-2023

Assignment 3 Web scraping

Q1 write a Python program which searches all the product under a particular product from www.amazon.in. The product to be searched will be taken as input from user. For e.g. if User input is 'guitar'. Then search for guitars.

⇒ Certainly ! Here's Python program using the Beautiful Soup library to scrape Amazon.in and search for products based on user input

Python

```
import requests.
from bs4 import BeautifulSoup
def search_amazon(Product):
    url = f"https://www.amazon.in/s?k={product}"
headers = {
    "User-Agent": Mozilla/5.0 (Windows NT 10.0
Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/108.0.5359.110 Safari/537.3
}
try
response = requests.get(url, headers=headers)
response.raise_for_status()
```

```
soup = BeautifulSoup(response.text, 'html.parser')
products = soup.find_all('div', {'class': 's-result-item'})

if len(products) > 0:
    print(f"Showing results for {product}:\n")
    for item in products:
        print(f"Showing results for {product}:\n")
        for item in products:
            product_name = item.find('span', {'class': 'a-text-normal'}).text.strip()
            product_price = item.find('span', {'class': 'a-price'})
            print(f"Product: {product_name}\nPrice: {product_price}\n")
    else:
        print("No products found.")
except requests.exceptions.HTTPError as e:
    print(f"HTTP error occurred: {e}")
except Exception as e:
    print(f"An error occurred: {e}")
if __name__ == "__main__":
    search_query = input("Enter the product you want to search for on Amazon.in: ")
    search_amazon(search_query)
```

This program will prompt the user to input a product, then it will scrape Amazon.in for that product, and print out the names and prices of the products found. Make sure to install the request and BeautifulSoup4 libraries if you don't have it already.

PAGE No.
DATE / /

A) Copy Code

```
pip install requests
pip install BeautifulSoup 4
```

This code can be further extended to include more detailed information or additional functionality as needed.

Q2 in the above question, now scrape the following details of each product listed in first 3 page of your search.
results and save it in a data frame and csv. in case if any product has less than 3 pages in search results then. Scrape all the product available under that product name. Details to be Scraped are: "Brand name" "name of the product" "price" "Return/Exchange" "Expected Delievery" "Availability" and "Product URL", in case if any of the details are missing for any of the products then replace it by" write a en

⇒ To accomplish this task, we'll need to extend the previous program to scrape the required details for each product listed on the on the first three pages of the search results on Amazon.in We'll use the pandas library to organise the data frame and the save it to to a csv file. csv file. Here' the updated code.

PAGE No.
DATE / /

Python

Copy code

```
import requests
from bs4 import beautifulsoup
import pandas as pd
def scrape_product_details(url):
    headers = {
        "User-Agent": Mailla/5.0 (Windows NT 10.0; Win64; 64) AppleWebKit/537.36 KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"}
    products_details = []
    try
    response = requests.get(url, headers=headers)
    response.raise_for_status()
    soup = BeautifulSoup(response.text, 'html.parser')
    brand_name = soup.find('a', {'id': 'bylineInfo'}).text.strip()
    products_name = soup.find('span', {'id': 'productTitle'}).text.strip()
    price = soup.find('span', {'id': 'priceblock_ourprice'}).text.strip()
    return_exchange = soup.find('a', {'id': 'productSupport and Return Policy_return policy_anchor_text'}).text.strip() if soup.find else.
    expected_delivery = soup.find('div', {'id': 'ddmmessage'}).text.strip() if soup.find('div', {'id': 'ddmdelivery message'}).text.strip().
    availability = soup.find('div', {'id': 'availability'}).text.strip() if soup.find('div', {'id': 'availability'}) else ""
```

```
product_details.append({
    'brand name'          : brand_name,
    'name of the product' : product_name,
    'price'               : price,
    'Return/Exchange'     : return_exchange,
    'Expected Delivery'   : expected_delivery,
    'Availability'        : availability,
    'Product URL'         : url,
})
except requests.exceptions.HTTPError as e:
    print(f"HTTP error occurred: {e}")
except Exception as e:
    print(f"An error occurred: {e}")
return product_details
def search_amazon(product):
    base_url = https://www.amazon.in/s?k=
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"}
    product_details = []
    try:
        for page_num in range(1,4):
            url = f"{base_url}{product}&page={page_num}"
            response = requests.get(url, headers=headers)
            response.raise_for_status()
            soup = BeautifulSoup(response.text, 'html.parser')
            products = soup.find_all('div', {'class': 's-result-item'})
```

```
for item in products:
product_link = item.find('a', class_='a-link-normal')
if product_link:
product_url = "https://www.amazon.in" + product_link['href']
product_details = scrape_product_details(product_url)
product_details = call_extend(product_details)
except requests.exceptions.HTTPError as e:
print(f"HTTP error occurred: {e}")
return product_details_all
if __name__ == "__main__":
search_query = input("Enter the product you want to search for on Amazon.in: ")
product_details = search_amazon(search_query)
if product_details:
df = pd.DataFrame(product_details)
df.to_csv(f"{search_query}_details.csv", index=False)
print("Product details saved to CSV file.")
else:
print("No products found.")
```

This code will search for the given product on amazon.in scrape the details for each product listed on the first three pages of the search Result (or all available page if less than three); and save the details to a CSV file named "{search_query}_details.csv". If any detail is missing for a product, it will be replaced by "2".

Q3 write a python program to access the search bar and search Button on image.google.com and Scrape 10 images each for keywords Fruits, cars and machine learning, Guitar, cakes' write a

⇒ To accomplish this task we can use the Selenium library in python which allows us to automate interactions with a web browser. First make sure you have the selenium library installed pip install selenium) and you also need to download a compatible web driver for the browser you want to automate (e.g Chrome Driver for Google Chrome) Here's the python programs

Python

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import os
import requests.
# Function to save image
def save_image(image_url, directory):
    try:
        img_data = requests.get(image_url).content
        with open(os.path.join(directory, os.path.basename
            (image_url)), 'wb') as handler:
            print("Image saved: {os.path.basename
                (image_url)}")
    except Exception as e:
        print(f"Error saving image: {e}")
```

```
# Function to Scrape Image
def scrape_image(keyword, num_images):
driver = webdriver.Chrome(executable_path="path
                          -to-chromedriver")
driver.get("https://www.google.com/imghp")
search_box = driver.find_element_By_xpath("
  //input[@title='Search']")
search_box.send_keys(keyword+Keys.RETURN)
# wait for results to load
time.sleep(2)
# scroll down to load more image
last_height = driver.execute_script("return docum
  ent.body.scrollHeight")
while True:
  driver.find_elements_by_xpath("//img
    [@class='rg_i Q4LuWd']")
  driver.execute_script("window.scrollTo(0,
    document.body.scrollHeight")
  time.sleep(2)
  new_height = driver.execute_script("return docum
    ent.body.scrollHeight")
  if new_height == last_height
    break
  last_height = new_height
# Get image URLs
image_elements = driver.find_elements_By_xpath
  ("//img[@class='rg_i Q4LuWd']")
image_urls = [elem.get_attribute("src") for
  in img_elements if elem.get_attribute("src")]
# create directory to save image
os.makedirs(keyword, exist_ok=True)
```

```
# save image
for i, url in enumerate(image_urls, [:num_images]):
save_image(url, keyword)
if i == num_images - 1:
break.
driver.quit()
if __name__ == "__main__":
keywords = ['fruits', 'cars', 'machine learning',
'Guitar', 'Cakes', ]
num_images = 10
for keyword in keyword:
scrape_images(keyword, num_images)
```

This script will open Google images in a Chrome browser, search for the specified keywords and scrape the URLs of the images. Then, it will save the first 10 images for each keyword in separate directories named after the keywords. Make sure to replace "path-to-chromedriver" with the actual path and your chrome driver executable.

PAGE No.
DATE / /

Q4 Write a Python program to search for a smart phone (e.g. oneplus, nord, pixel 4A, etc.) on www.flipkart.com and scrape following details for all the search results displayed on 1st page. Details to be scraped: "Brand Name", "Smart phone name", "Colour", "RAM", "Storage (ROM)", "Primary Camera", "Price"

⇒ Python

```
import requests
from bs4 import BeautifulSoup
def scrape_smartphones(product):
    url = f"https://www.flipkart.com/search?q={product}&otracker=search&marketplace=FLIPKART&as-show=on&as=off"
    headers = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.3029.110 Safari/537.3"
    }
    try:
        response = requests.get(url, headers=headers)
        response.raise_for_status()
        soup = BeautifulSoup(response.text, 'html.parser')
        products = soup.find_all('div', {'class': 'IRpwTa'})
        for product in products:
            brand_name = product.find('div', {'class': 'col col-7-12'}).text.strip()
            smartphone_name = product.find('a', {'class': 'IRpwTa'}).text.strip()
            specs = product.find_all('li', {'class': 'rgWa7D'})
            color = specs[0].text.strip()
```

Copy code

```
ram = spec[2].text.strip()
primary_camera = specs[3].text.strip()
print("Brand name:", brand_name)
print("Smart phone name:")
print("Color:", color)
print("RAM:", Ram)
print("Storage(Rom):", rom)
print("primary camera:", primary_camera)
print("\n")
except requests.exceptions.HTTPError as e:
    print("HTTP error occured: {e}")
except Exception as e:
    print("An error occured: {e}")
if __name__ == "__main__":
search_query = input("Enter the Smart phone you want to search for on Flipkart:")
scrape_smart_phone(search_query)
```

This program takes user input for the smart phone to search on Flipkart, scrapes the details for each product listed on the first page of the search results, and prints out the details for each smart phone found. You can run this program, input the smartphone name (e.g. "oneplus nord", "pixel 4a"), and it will display the brand name, smart phone name, color, RAM, Rom, and primary camera for each product listed on the first page of the search results on Flipkart.

PAGE No.
DATE / /

Q5 Write a program to scrap geospatial coordinates (latitude, longitude) of a city searched on google maps.

⇒ To scrape geospatial coordinates (latitude and longitude) of a city searched on google Maps, we can use Google Maps Geocoding API. This API allows us to retrieve geolocation data based on search queries.

Here's Python program to accomplish this using the 'request' library to make API requests:

Python — Copy code

```
import requests
def get_coordinates(city_name):
    api_key = "YOUR_GOOGLE_MAPS_API_KEY"
    url = f"https://maps.googleapis.com/api/geocode/json?address={city_name}&key={api_key}"
    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json()
        if data['status'] == 'OK':
            location = data['results'][0]['geometry']['location']
            latitude = location['lat']
            longitude = location['lng']
            print(f"Geospatial coordinates for {city_name}:")
            print(f"Latitude: {latitude}")
            print(f"Longitude: {longitude}")
        else:
            print("Error: Unable to retrieve coordinates.")
```

except requests.exceptions.HTTPError as e:
print(f"HTTP error occurred: {e}")
except Exception as e:
print(f"An error occurred: {e}")

if __name__ == "__main__":
city = input("Enter the city name to get its coordinates: ")
get_coordinates(city)

Before running the program, make sure to replace 'YOUR_GOOGLE_MAPS_API_KEY' with your actual Google Maps API key. You can obtain an API key by following the instructions here: https://developers.google.com/maps/gmp-get-started.

Once you have your API key, you can run the program, input the city name you want to search for on Google Maps, and it will print out the geospatial coordinates (latitude and longitude) of that city.

PAGE No.
DATE / /

Q6 write a program to scrap all the available details of Best gaming laptops from digit.in.

⇒ To scrape all the available details of the gaming laptop from digit.in we can inspect the website to understand its structure and then use python along with the BeautifulSoup library to extract the necessary information. Here's python program to accomplish this task

```
Python                                    Copy code
import requests
from bs4 import BeautifulSoup
def scrape_gaming_laptops():
url = "https://www.digit.in/top-products/best-gaming-laptops-40.html"
headers = {
"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3
}
try:
response = requests.get(url, headers=headers)
response.raise_for_status()
soup = BeautifulSoup(response.text, 'html.parser')
laptops = soup.find_all('div', {'class': 'top-number-heading-starting-4 sticky-footer'})

for laptop in laptops:
laptop_name = laptop.find('h3').text.strip()
specs = laptop.find_all('p')
```

```
processor = specs[0].text.strip()
ram = specs[1].text.strip()
os = specs[2].text.strip()
display = specs[3].text.strip()
storage = specs[4].text.strip()
price = specs[5].text.strip()

print("Laptop name:", laptop_name)
print("Processor:", processor)
print("Ram:", ram)
print("Operating system:", os)
print("Display:", display)
print("Storage:", storage)
print("Price:", price)
print("\n")

except requests.exceptions.HTTPError as e:
    print("HTTP error occured:", e)
except Exception as e:
    print("An error occured:", e)
if __name__ == "__main__":
    scrape_gaming_laptops()
```

This program will scrape the details of the best gaming laptop from digit.in and print out the laptop name, processor, RAM, Operating system, display, storage & price for each laptop. You can further process this info as per your requirement such as saving to a file or storing it in a database.

PAGE No.
DATE / /

Q7 write a python program to scrape the details for all billionaires from www.forbes.com details to be scrapped.
"Rank", "Name", "Net worth", "Age" "Citizenship", "Source", "Industry"...

⇒ Sure, here's a python program that the beautiful soup library to scrape the details of billionaires from Forbes website.

python

```
import requests
from bs4 import BeautifulSoup
def scrape_billionaires():
url = "https://www.forbes.com/billionaires/"
response = requests.get(url)
soup = BeautifulSoup(response.content, "html.parser")
billionaires_data = []
for row in soup.find_all("div", class_="table-row"):
rank = row.find("div", class_="rank").text.strip()
name = row.find("div", class_="person-name").text.strip()
net_worth = row.find("div", class_="net-worth").text.strip()
age = row.find("div", class_="age").text.strip()
citizenship = row.find("div", class_="country-of-citizenship")
source = row.find("div", class_="source").text.strip()
industry = row.find("div", class_="category").text.strip()
billionaire = {
    "Rank": rank,
    "Name": name,
    "Net worth": net_worth,
    "Age": age,
```

Copy code

"Citizenship" : citizenship,
"Source" : source,
"Industry" : industry

3.

```
billionaires - data-appeed.(Billionire)
return billionaires - data
if -- name -- == "-- main --":
  Billionires = Soupe - Billionaires ()
  for billionairs in billionaires:
      print (billionaire)
```

This code will extract the required details (Rank, name, net worth, Age, citizenship, source, and industry) for each billionaire listed on Forbes website and print them out. Make sure to have the Beautiful Soup library installed ("pip install Beautiful Soup 4") before running this code.

PAGE No.
DATE / /

Q 8 write a program to extract at least 500 comment upvote and time when comment was posted from any you Tube video.

=> to extract comments, comment upvotes, and the time when each comment was posted from a yobe video, you can use the you Tube Data API. First you need to need to set up a project in the Google Cloud Console and enable the you Tube Data API for that project. Then, you can use the google-api-python-client library to interact with the API. Below is a python program that demonstrates how to do this.

Python Copy code

```
from googleapiclient.discovery import build
from datetime import datetime
# set up the you Tube Data API Service
api-key = "YOUR-API-KEY" # Replace "YOUR-API-KEY" with your actual API key
youtube = build("youtube", "v3", developerKey=api_key)
def get-video-Comments video-id-max-results=500):
comments = []
next-page-token = none
while while len(comments) < max-results:
response = youtube-commentThreads().list(
part="snippet",
videoId = video-id.
```

PAGE No.
DATE / /

maxResults = min (max_results - len (comments), 100)

page_token = next_page_token
).execute()
for item in response["items"]:
comment = item["snippet"]["topLevelComment"]["snippet"]
comment_text = comment["textDisplay"]
comment_upvotes = comment.get("likeCount", 0)
comment_time = comment["publishedAt"]
comment_time = datetime.strptime(comment_time, '%Y-%m-%dT%H:%M:%S')

comments.append({
"text" : comment_text,
"upvotes" : comment_upvotes,
"time" : comment_time
})
if "nextPageToken" in response:
next_page_token = response["nextPageToken"]
else:
break

return comments
if __name__ == '__main__':
video_id = "YOUR_VIDEO_ID"
# Replace "YOUR_VIDEO_ID" with the ID of the YouTube video

```
comments = get_video_comments(video_id)

for comment in comments:
    print("Comment", comment["text"])
    print("upvotes", comment["upvotes"])
    print("times", comment["time"])
    print()
```

make sure to replace "YOUR_API_KEY" with your actual YouTube Data API key and "VIDEO_ID" with the ID of the YouTube video you want to extract comments from. This code will print out the comments, the number of upvotes each comment has received, and the time when each comment was posted.

69 Write a python program to Scrape details for all available Hostels from https://www.hostelworld.com/in "London" location. You have to scrape hostel name, distance from city centre, rating, total reviews, overall reviews, private from price, dorms from price, facilities and property description.

PAGE No.
DATE / /

=> To scrape data for all available hostels in London from Hostelworld, you can use the BeautifulSoup library in Python. Here's a Python program to achieve this.

Python

```
import requests
from bs4 import BeautifulSoup

def scrape_hostels_in_london():
    url = "https://www.hostelworld.com/findabed.php/ChosenCity.London/ChosenCountry.England"
    response = requests.get(url)
    soup = BeautifulSoup(response.content, "html.parser")

    hostels = []
    hostel_list = soup.find_all("div", class_="property")

    for hostel in hostel_list:
        name = hostel.find("h2", class_="title").text.strip()
        distance = hostel.find("span", class_="description").text.strip()
        rating = hostel.find("div", class_="score orange").text.strip()
        total_reviews = hostel.find("div", class_="keyword").text.strip()
        overall_reviews = hostel.find("div", class_="keyword").text.strip()
        private_price = hostel.find("div", class_="priceTitle").text.strip()
        dorm_price = hostel.find("div", class_="priceTitle").text.strip()
```

Copy code

PAGE No.
DATE / /

```
text.strip()
facilities = [item.text.strip() for item in hostel.find_all("li", class_="Facility-badge")]
description = hostel.find("div", class_="keyword").find_next_sibling("div").text.strip()
hostel_data = {
"name" : name,
"distance from city centre" : distance,
"Rating" : Rating,
"total_reviews" : total_reviews,
"overall reviews" : overall_reviews,
"privates from price" : privates_price,
"dorms from price" : dorms_price,
"Facilities" : facilities,
"description" : description
}
hostels.append(hostel_data)
return hostels
if __name__ == "__main__":
london_hostels = scrape_hostels_in_london()
for hostel in london_hostels:
    print(hostel)
    print()
```

This code will scrape data for all available hostels in London from Hostelworld, including hostel name, distance from the city centre, rating, total reviews, overall reviews, prices for private rooms and dorms, facilities, and property description. Make sure to have the BeautifulSoup library installed ("pip install BeautifulSoup4") before running this code