1) 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

Ans:

import requests

from bs4 import BeautifulSoup

def search\_product(product):

# Format the search query

search\_query = product.replace(' ', '+')

# Send a GET request to Amazon.in search page

url = f'https://www.amazon.in/s?k={search\_query}'

response = requests.get(url)

# Parse the HTML content using BeautifulSoup

soup = BeautifulSoup(response.content, 'html.parser')

# Find all the product elements on the page

products = soup.find\_all('div', {'data-component-type': 's-search-result'})

# Iterate over the products and extract relevant information

for product in products:

# Extract the product title

title = product.find('span', {'class': 'a-size-medium'}).text.strip()

# Extract the product price

price = product.find('span', {'class': 'a-price-whole'}).text.strip()

# Extract the product rating

rating = product.find('span', {'class': 'a-icon-alt'}).text.strip()

# Print the product information

print(f'Title: {title}')

print(f'Price: {price}')

print(f'Rating: {rating}')

print('---')

# Take user input for the product to search

product = input('Enter the product to search: ')

# Call the search\_product function with the user input

search\_product(product)

2. In the above question, now scrape the following details of each product listed in first 3 pages 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 products available under that product name. Details to be scraped are: "Brand Name", "Name of the Product", "Price", "Return/Exchange", "Expected Delivery", "Availability" and “Product URL”. In case, if any of the details are missing for any of the product then replace it by “-“

3. Write a python program to access the search bar and search button on images.google.com and scrape 10 images each for keywords ‘fruits’, ‘cars’ and ‘Machine Learning’, ‘Guitar’, ‘Cakes’.

Ans:

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

# Set the path to the web driver executable

driver\_path = 'path\_to\_chromedriver'

# Create a new instance of the Chrome driver

driver = webdriver.Chrome(driver\_path)

# Open images.google.com

driver.get('https://images.google.com')

# Find the search bar element and enter the keywords

search\_bar = driver.find\_element(By.NAME, 'q')

keywords = ['fruits', 'cars', 'Machine Learning', 'Guitar', 'Cakes']

for keyword in keywords:

search\_bar.clear()

search\_bar.send\_keys(keyword)

search\_bar.send\_keys(Keys.RETURN)

# Wait for the search results to load

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.CLASS\_NAME, 'rg\_i')))

# Scrape the image URLs

image\_elements = driver.find\_elements(By.CLASS\_NAME, 'rg\_i')

image\_urls = [element.get\_attribute('src') for element in image\_elements]

# Print the first 10 image URLs

print(f"Top 10 images for '{keyword}':")

for url in image\_urls[:10]:

print(url)

# Close the browser

driver.quit()

4.Write a python program to search for a smartphone(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”, “Smartphone name”, “Colour”, “RAM”, “Storage(ROM)”, “Primary Camera”, “Secondary Camera”, “Display Size”, “Battery Capacity”, “Price”, “Product URL”. Incase if any of the details is missing then replace it by “- “. Save your results in a dataframe and CSV.

Ans:

import requests

from bs4 import BeautifulSoup

import pandas as pd

# Function to scrape smartphone details

def scrape\_smartphones():

url = "https://www.flipkart.com/search?q=smartphone" # Replace "smartphone" with your desired search query

response = requests.get(url)

soup = BeautifulSoup(response.content, 'html.parser')

smartphones = []

results = soup.find\_all('div', {'class': '\_1AtVbE'})

for result in results:

details = {}

# Extracting details from each search result

details['Brand Name'] = result.find('div', {'class': '\_4rR01T'}).text

details['Smartphone Name'] = result.find('a', {'class': 'IRpwTa'}).text

details['Colour'] = result.find('div', {'class': '\_2WkVRV'}).text

details['RAM'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[0].text

details['Storage(ROM)'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[1].text

details['Primary Camera'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[2].text

details['Secondary Camera'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[3].text

details['Display Size'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[4].text

details['Battery Capacity'] = result.find('ul', {'class': '\_1xgFaf'}).find\_all('li')[5].text

details['Price'] = result.find('div', {'class': '\_30jeq3 \_1\_WHN1'}).text

details['Product URL'] = "https://www.flipkart.com" + result.find('a', {'class': 'IRpwTa'})['href']

smartphones.append(details)

return smartphones

# Scrape smartphone details

smartphones = scrape\_smartphones()

# Create dataframe from the scraped details

df = pd.DataFrame(smartphones)

# Replace missing details with "-"

df.fillna("-", inplace=True)

# Save dataframe to CSV

df.to\_csv('smartphones.csv', index=False)

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

Ans:

import requests

from bs4 import BeautifulSoup

def get\_coordinates(city):

# Format the search query

query = f"google maps {city} coordinates"

# Send a GET request to Google search

response = requests.get(f"https://www.google.com/search?q={query}")

# Parse the HTML response using BeautifulSoup

soup = BeautifulSoup(response.text, 'html.parser')

# Find the element containing the coordinates

result = soup.find("div", class\_="BNeawe iBp4i AP7Wnd")

# Extract the coordinates from the element

coordinates = result.text.split(",")

latitude = coordinates[0].strip()

longitude = coordinates[1].strip()

return latitude, longitude

# Example usage

city = "New York"

latitude, longitude = get\_coordinates(city)

print(f"The coordinates of {city} are: Latitude - {latitude}, Longitude - {longitude}")

6. Write a program to scrap all the available details of best gaming laptops from digit.in

Ans: from selenium import webdriver

import time

# Set up the WebDriver

driver = webdriver.Chrome('path\_to\_chromedriver')

# Open the website

driver.get('https://www.digit.in/')

# Search for gaming laptops

search\_bar = driver.find\_element\_by\_id('searchDiv')

search\_bar.send\_keys('gaming laptops')

search\_bar.submit()

# Wait for the search results to load

time.sleep(2)

# Scrape the details

laptop\_elements = driver.find\_elements\_by\_class\_name('searchPage')

laptop\_details = []

for laptop in laptop\_elements:

name = laptop.find\_element\_by\_class\_name('searchProductTitle').text

price = laptop.find\_element\_by\_class\_name('searchPrice').text

specifications = laptop.find\_element\_by\_class\_name('searchSpec').text

laptop\_details.append({

'Name': name,

'Price': price,

'Specifications': specifications

})

# Print the scraped details

for laptop in laptop\_details:

print(laptop)

# Close the WebDriver

driver.quit()

7. 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”.

Ans:

import requests

from bs4 import BeautifulSoup

# Send a GET request to the Forbes website

url = "https://www.forbes.com/billionaires/"

response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content

soup = BeautifulSoup(response.content, "html.parser")

# Find the table containing the billionaire details

table = soup.find("table", class\_="table")

# Find all the rows in the table

rows = table.find\_all("tr")

# Iterate over each row and extract the required details

for row in rows:

# Find all the columns in the row

columns = row.find\_all("td")

# Extract the required details from the columns

rank = columns[0].text.strip()

name = columns[1].text.strip()

net\_worth = columns[2].text.strip()

age = columns[3].text.strip()

citizenship = columns[4].text.strip()

source = columns[5].text.strip()

industry = columns[6].text.strip()

# Print the extracted details

print("Rank:", rank)

print("Name:", name)

print("Net Worth:", net\_worth)

print("Age:", age)

print("Citizenship:", citizenship)

print("Source:", source)

print("Industry:", industry)

print()

8. . Write a program to extract at least 500 Comments, Comment upvote and time when comment was posted from any YouTube Video.

Ans:

from selenium import webdriver

import time

# Set up the WebDriver

driver = webdriver.Chrome('path\_to\_chromedriver') # Replace with the path to your WebDriver executable

# Open the YouTube video

video\_url = 'https://www.youtube.com/watch?v=your\_video\_id' # Replace with the URL of the YouTube video

driver.get(video\_url)

# Scroll to load comments

scroll\_pause\_time = 2 # Adjust the pause time as needed

scrolls = 10 # Adjust the number of scrolls as needed

for \_ in range(scrolls):

driver.execute\_script("window.scrollTo(0, document.documentElement.scrollHeight);")

time.sleep(scroll\_pause\_time)

# Extract comments, upvotes, and time

comments = driver.find\_elements\_by\_xpath('//yt-formatted-string[@id="content-text"]')

upvotes = driver.find\_elements\_by\_xpath('//span[@id="vote-count-middle"]')

times = driver.find\_elements\_by\_xpath('//a[@class="yt-simple-endpoint style-scope yt-formatted-string"]')

# Store the extracted data

extracted\_data = []

for comment, upvote, time in zip(comments, upvotes, times):

extracted\_data.append({

'comment': comment.text,

'upvote': upvote.text,

'time': time.text

})

# Close the WebDriver

driver.quit()

# Print the extracted data

for data in extracted\_data:

print(data)

9. Write a python program to scrape a data for all available Hostels from https://www.hostelworld.com/ in “London” location. You have to scrape hostel name, distance from city centre, ratings, total reviews, overall reviews, privates from price, dorms from price, facilities and property description.

Ans:

import requests

from bs4 import BeautifulSoup

# Send a GET request to the website

url = "https://www.hostelworld.com/hostels/London"

response = requests.get(url)

# Create a BeautifulSoup object to parse the HTML content

soup = BeautifulSoup(response.content, "html.parser")

# Find all the hostel listings on the page

hostel\_listings = soup.find\_all("div", class\_="fabresult")

# Iterate over each hostel listing and extract the required information

for hostel in hostel\_listings:

# Extract hostel name

name = hostel.find("h2", class\_="title").text.strip()

# Extract distance from city centre

distance = hostel.find("span", class\_="distance").text.strip()

# Extract ratings

ratings = hostel.find("div", class\_="score orange").text.strip()

# Extract total reviews

total\_reviews = hostel.find("div", class\_="reviews").text.strip()

# Extract overall reviews

overall\_reviews = hostel.find("div", class\_="rating").text.strip()

# Extract privates from price

privates\_price = hostel.find("div", class\_="price").text.strip()

# Extract dorms from price

dorms\_price = hostel.find("div", class\_="price dorms").text.strip()

# Extract facilities

facilities = [facility.text.strip() for facility in hostel.find\_all("span", class\_="facilities")]

# Extract property description

description = hostel.find("div", class\_="desc").text.strip()

# Print the extracted information

print("Name:", name)

print("Distance from city centre:", distance)

print("Ratings:", ratings)

print("Total reviews:", total\_reviews)

print("Overall reviews:", overall\_reviews)

print("Privates from price:", privates\_price)

print("Dorms from price:", dorms\_price)

print("Facilities:", facilities)

print("Description:", description)

print()