**Amazon Web Scraber**

**Using Selenium and Python:-**

from selenium import webdriver

from selenium.webdriver.edge.service import Service

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

from selenium.common.exceptions import TimeoutException, NoSuchElementException

from dotenv import load\_dotenv

import time

import json

import os

# Load environment variables from .env file

load\_dotenv()

# Setup WebDriver for Microsoft Edge

edge\_driver\_path = r"C:\Users\ASUS\edgedriver\_win64\msedgedriver.exe"  # Update with the path to your msedgedriver.exe

service = Service(edge\_driver\_path)

driver = webdriver.Edge(service=service)

wait = WebDriverWait(driver, 10)

# Amazon login function

def login\_amazon(username, password):

    driver.get("https://www.amazon.in/ap/signin?openid.pape.max\_auth\_age=900&openid.return\_to=https%3A%2F%2Fwww.amazon.in%2Fgp%2Fyourstore%2Fhome%3Fpath%3D%252Fgp%252Fyourstore%252Fhome%26signIn%3D1%26useRedirectOnSuccess%3D1%26action%3Dsign-out%26ref\_%3Dnav\_AccountFlyout\_signout&openid.assoc\_handle=inflex&openid.mode=checkid\_setup&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0")

    try:

        email\_input = wait.until(EC.presence\_of\_element\_located((By.ID, "ap\_email")))

        email\_input.send\_keys(username)

        driver.find\_element(By.ID, "continue").click()

        password\_input = wait.until(EC.presence\_of\_element\_located((By.ID, "ap\_password")))

        password\_input.send\_keys(password)

        driver.find\_element(By.ID, "signInSubmit").click()

    except TimeoutException:

        print("Login failed. Check your credentials or captcha.")

# Scrape category data

def scrape\_category(url, category\_name, output\_dir, max\_products=1500):

    driver.get(url)

    products = []

    product\_count = 0

    while product\_count < max\_products:

        try:

            items = wait.until(EC.presence\_of\_all\_elements\_located((By.CSS\_SELECTOR, ".zg-grid-general-faceout")))

            for item in items:

                try:

                    product\_name = item.find\_element(By.CSS\_SELECTOR, ".p13n-sc-truncate").text

                    product\_price = item.find\_element(By.CSS\_SELECTOR, ".p13n-sc-price").text

                    discount = item.find\_element(By.CSS\_SELECTOR, ".p13n-discount").text if "discount" in item.text.lower() else "N/A"

                    rating = item.find\_element(By.CSS\_SELECTOR, ".a-icon-alt").text

                    sold\_by = item.find\_element(By.CSS\_SELECTOR, ".p13n-seller-details").text if "sold by" in item.text.lower() else "N/A"

                    ship\_from = item.find\_element(By.CSS\_SELECTOR, ".p13n-shipping").text if "ship from" in item.text.lower() else "N/A"

                    # Ensure discount is greater than 50%

                    if "50%" in discount or "more" in discount:

                        product = {

                            "Category Name": category\_name,

                            "Product Name": product\_name,

                            "Price": product\_price,

                            "Discount": discount,

                            "Rating": rating,

                            "Ship From": ship\_from,

                            "Sold By": sold\_by

                        }

                        products.append(product)

                        product\_count += 1

                        if product\_count >= max\_products:

                            break

                except NoSuchElementException:

                    continue

            # Click next page

            next\_button = driver.find\_element(By.CSS\_SELECTOR, ".zg-pagination-next")

            if next\_button.is\_enabled():

                next\_button.click()

                time.sleep(3)

            else:

                break

        except TimeoutException:

            print(f"Failed to load products for category {category\_name}.")

            break

    # Save products to JSON

    with open(os.path.join(output\_dir, f"{category\_name}\_products.json"), "w") as f:

        json.dump(products, f, indent=4)

    print(f"Scraped {len(products)} products from category {category\_name}.")

# Main script

def main():

    # Load credentials from .env file

    AMAZON\_EMAIL = os.getenv("AMAZON\_EMAIL")

    AMAZON\_PASSWORD = os.getenv("AMAZON\_PASSWORD")

    if not AMAZON\_EMAIL or not AMAZON\_PASSWORD:

        print("Error: AMAZON\_EMAIL and AMAZON\_PASSWORD must be set in the .env file.")

        return

    # Create output directory

    output\_dir = "amazon\_best\_sellers"

    if not os.path.exists(output\_dir):

        os.makedirs(output\_dir)

    # Login to Amazon

    login\_amazon(AMAZON\_EMAIL, AMAZON\_PASSWORD)

    # URLs for categories

    category\_urls = {

        "Kitchen": "https://www.amazon.in/gp/bestsellers/kitchen/ref=zg\_bs\_nav\_kitchen\_0",

        "Shoes": "https://www.amazon.in/gp/bestsellers/shoes/ref=zg\_bs\_nav\_shoes\_0",

        "Computers": "https://www.amazon.in/gp/bestsellers/computers/ref=zg\_bs\_nav\_computers\_0",

        "Electronics": "https://www.amazon.in/gp/bestsellers/electronics/ref=zg\_bs\_nav\_electronics\_0"

        # Add more categories as needed

    }

    for category\_name, url in category\_urls.items():

        scrape\_category(url, category\_name, output\_dir)

    # Close the browser

    driver.quit()

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

    main()

* Used .env files for putting login details.
* Scrapped Data from Amazon Website via library.