Indeed Job Scraper – Project Report

Done By Krishna Gowsalya M

Introduction

The Indeed Job Scraper is a Python-based automation project that collects job postings directly from Indeed India. It helps job seekers, researchers, and analysts to efficiently gather job market data for various roles and locations. The tool captures essential details such as job title, company name, location, job summary, posting date, and job link, and exports them into a CSV file for further analysis.

Prerequisites

- Programming Language - Python
- Require python libraries-Selenium , Pandas , webdriver-manager,OpenPyXL

Technologies Used

- Web Scraping - To extract job posting data from Indeed.
- Browser Automation - Using Selenium to control Chrome browser.
- Data Logging - Storing results in CSV format.
- Headless Mode - To run the scraper silently in the background.
- User-Agent Spoofing - To avoid bot detection and mimic real browsers.

Install Required Libraries

pip install selenium pandas webdriver-manager openpyxl

Python Libraries Used

- webdriver_manager: Automatically downloads and configures ChromeDriver.
- pandas: Structures the scraped job data into tabular format and exports it as CSV.
- time: Adds delays to ensure the webpage loads properly before scraping.
- selenium: Automates browser actions to extract job posting details.

Coding

Step by step Instructions

Step 1: Import Required Modules

```
import re, time, random
from datetime import datetime, timedelta
import pandas as pd
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from webdriver_manager.chrome import ChromeDriverManager
```

Step 2: Define Utility Functions

```
def clean_text(s):
    """Remove extra whitespace and newlines"""
    if not s:
        return ""
    return re.sub(r"\s+", " ", s).strip()

def parse_date_posted(text):
    """Convert relative dates like '2 days ago' or 'Just posted' to YYYY-MM-DD"""
    today = datetime.today().date()
    t = text.lower()
    if "today" in t or "just posted" in t:
        return str(today)
    match = re.search(r"(\d+)\+?\s+day", t)
    if match:
        return str(today - timedelta(days=int(match.group(1))))
    return str(today)
```

Step 3:Configure and Launch Chrome WebDriver

```
options = webdriver.ChromeOptions()
options.add_argument("--disable-blink-features=AutomationControlled")
options.add_argument("user-agent=Mozilla/5.0 (Windows NT 10.0;
Win64; x64) "
  "AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0
Safari/537.36")
options.add_argument("--headless=new")
driver = webdriver.Chrome(options=options)
```

Step 4: Define Scraping Function

```
def scrape_indeed(job_title="Python Developer", location="Chennai",
pages=1):
   base_url = f"https://in.indeed.com/jobs?q={job_title.replace(' ',
'+')}&l={location.replace(' ', '+')}"
   all_jobs = []
```

Step 5: Open Indeed Job Listings Page

```
for page in range(pages):
url = f"{base_url}&start={page*10}"
      print(f"\n Fetching page {page+1}: {url}")
      driver.get(url)
      time.sleep(random.uniform(3, 6))  # human-like delay
```

Step 6: Scroll Page to Load Jobs

```
scroll_height = driver.execute_script("return document.body.scrollHeight")
      for i in range(0, scroll_height, 300):
         driver.execute_script(f"window.scrollTo(0, {i});")
         time.sleep(random.uniform(0.3, 0.8))
```

Step 7: Wait for Job Cards to Appear

```
try:
         WebDriverWait(driver, 15).until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR,
"div.job_seen_beacon"))
         )
```

```
except:
 print(" No jobs found or page blocked")
 continue

job_cards = driver.find_elements(By.CSS_SELECTOR, "div.job_seen_beacon")
       print(f" Found {len(job_cards)} jobs on this page")
```

Step 8: Extract Job Details from Each Card

```
for card in job_cards:
          # --- Title ---
          title = clean_text(card.find_element(By.CSS_SELECTOR,
"h2.jobTitle").text) if card.find_elements(By.CSS_SELECTOR, "h2.jobTitle")
else ""

          # --- Company ---
          company = clean_text(card.find_element(By.CSS_SELECTOR,
"span.companyName").text) if card.find_elements(By.CSS_SELECTOR,
"span.companyName") else ""

          # --- Location ---
          location_ = ""
          if card.find_elements(By.CSS_SELECTOR, "div.companyLocation"):
             location_ = clean_text(card.find_element(By.CSS_SELECTOR,
"div.companyLocation").text)
          elif card.find_elements(By.CSS_SELECTOR, "span.location"):
             location_ = clean_text(card.find_element(By.CSS_SELECTOR,
"span.location").text)
```

```
 elif card.find_elements(By.CSS_SELECTOR, "div.company_location"):
 location_ = clean_text(card.find_element(By.CSS_SELECTOR,
"div.company_location").text)

 # --- Salary ---
 salary = ""
 if card.find_elements(By.CSS_SELECTOR, "div.salary-snippet"):
 salary = clean_text(card.find_element(By.CSS_SELECTOR, "div.salary-
snippet").text)
 elif card.find_elements(By.CSS_SELECTOR, "span.salary-snippet-container"):
 salary = clean_text(card.find_element(By.CSS_SELECTOR, "span.salary-snippet-
container").text)
 elif card.find_elements(By.CSS_SELECTOR, "div.metadata.salary-snippet-
container"):
 salary = clean_text(card.find_element(By.CSS_SELECTOR, "div.metadata.salary-
snippet-container").text)

 # --- Date Posted ---
 date_posted =
parse_date_posted(clean_text(card.find_element(By.CSS_SELECTOR,
"span.date").text)) if card.find_elements(By.CSS_SELECTOR, "span.date") else ""

 # --- Summary ---
 summary = clean_text(card.find_element(By.CSS_SELECTOR, "div.job-
snippet").text) if card.find_elements(By.CSS_SELECTOR, "div.job-snippet") else ""

 # --- Link ---
 link = card.find_element(By.CSS_SELECTOR, "h2.jobTitle a").get_attribute("href")
if card.find_elements(By.CSS_SELECTOR, "h2.jobTitle a") else ""
```

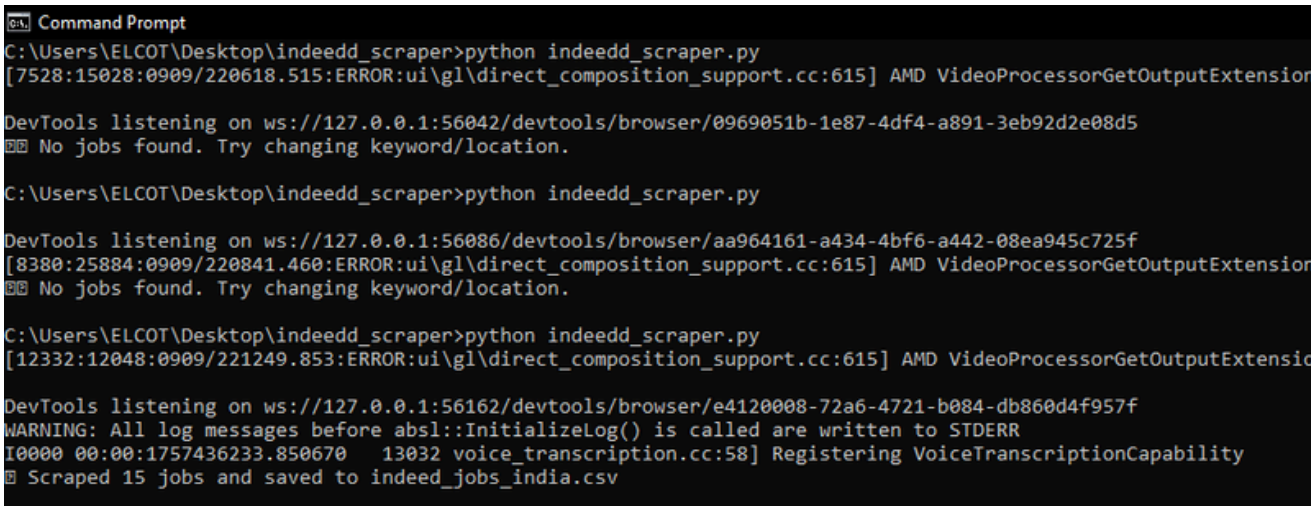
Step 9: Append Data into List

```
all_jobs.append({
            "Title": title,
            "Company": company,
            "Location": location_,
            "Salary": salary,
            "Date Posted": date_posted,
            "Summary": summary,
            "Link": link
        })

   return all_jobs
```

Step 10: Save Results into Excel File

```
jobs = scrape_indeed(pages=1)
wb = Workbook()
ws = wb.active
ws.title = "Indeed Jobs"

headers = ["Title", "Company", "Location", "Salary", "Date Posted",
"Summary", "Link"]
ws.append(headers)

for job in jobs:
   ws.append([job[h] for h in headers])

wb.save("indeedscraper_jobs.xlsx")
print(f"\n Done! Saved {len(jobs)} jobs to scraper_jobs.xlsx")

driver.quit()
```

Output



```
Command Prompt
C:\Users\ELCOT\Desktop\indeedd_scraper>python indeedd_scraper.py
[7528:15028:0909/220618.515:ERROR:ui\gl\direct_composition_support.cc:615] AMD VideoProcessorGetOutputExtensior

DevTools listening on ws://127.0.0.1:56042/devtools/browser/0969051b-1e87-4df4-a891-3eb92d2e08d5
⚠ No jobs found. Try changing keyword/location.

C:\Users\ELCOT\Desktop\indeedd_scraper>python indeedd_scraper.py

DevTools listening on ws://127.0.0.1:56086/devtools/browser/aa964161-a434-4bf6-a442-08ea945c725f
[8380:25884:0909/220841.460:ERROR:ui\gl\direct_composition_support.cc:615] AMD VideoProcessorGetOutputExtensior
⚠ No jobs found. Try changing keyword/location.

C:\Users\ELCOT\Desktop\indeedd_scraper>python indeedd_scraper.py
[12332:12048:0909/221249.853:ERROR:ui\gl\direct_composition_support.cc:615] AMD VideoProcessorGetOutputExtensio

DevTools listening on ws://127.0.0.1:56162/devtools/browser/e4120008-72a6-4721-b084-db860d4f957f
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1757436233.850670   13032 voice_transcription.cc:58] Registering VoiceTranscriptionCapability
✅ Scraped 15 jobs and saved to indeed_jobs_india.csv
```

	Title	Company	Location	Date Posted	Job Link
2	Python Deve	Capgemini Eng	Bengaluru, Karnataka		https://in.indeed.com/q-software-developer-l-bengaluru%2C-karnataka-jobs.html?utm_sou
3	Senior Java I	Propel Technol	Chennai, Tamil Nadu		https://in.indeed.com/q-software-developer-l-chennai%2C-tamil-nadu-jobs.html?utm_sour
4	Python Deve	Bahwan Cybert	Chennai, Tamil Nadu		https://in.indeed.com/q-software-engineer-l-chennai%2C-tamil-nadu-jobs.html?utm_sourc
5	Python Deve	E2logy Softwar	Ahmedabad, Gujarat		https://in.indeed.com/q-software-developer-l-ahmedabad%2C-gujarat-jobs.html?utm_sour
6	Python Deve	FinByz Tech	Ahmedabad, Gujarat		https://in.indeed.com/q-software-developer-l-ahmedabad%2C-gujarat-jobs.html?utm_sour
7	Python deve	Kanhasoft Pvt.L	Ahmedabad, Gujarat		https://in.indeed.com/q-software-developer-l-ahmedabad%2C-gujarat-jobs.html?utm_sour
8	Python Deve	BairesDev	Remote in Ahmedabad, G		https://in.indeed.com/q-software-developer-l-ahmedabad%2C-gujarat-jobs.html?utm_sour
9	Python Deve	ROAMIFY TECH	Abhyankar Nagar, Nagpur,		https://in.indeed.com/q-software-developer-l-maharashtra-jobs.html?utm_source=chatgpt
10	Python Deve	JPMorganChas	Mumbai, Maharashtra		https://in.indeed.com/q-software-developer-l-mumbai%2C-maharashtra-jobs.html?utm_so
11	Python Deve	Monalisa Grou	Remote in Thane, Mahara		https://in.indeed.com/q-software-developer-l-thane%2C-maharashtra-jobs.html?utm_sourc
12	Python Deve	Accenture	Ahmedabad, Gujarat		https://in.indeed.com/viewjob?jk=3d3d3d3d3d3d3d3d
13	Python Deve	SailPoint	Pune, Maharashtra		https://in.indeed.com/viewjob?jk=2c2c2c2c2c2c2c2c
14	Python/Djar	Siemens	Thane, Maharashtra		https://in.indeed.com/q-software-developer-l-thane%2C-maharashtra-jobs.html?utm_sourc
15	FULL STACK	Optum	Chennai, Tamil Nadu		https://in.indeed.com/viewjob?jk=4f5e5e5e5e5e5e5e
16	Python Back	ClanX	Hybrid work in Mumbai, M		https://in.indeed.com/q-software-developer-l-mumbai%2C-maharashtra-jobs.html?utm_so

Troubleshooting Table

Issue	Cause	Solution
No jobs scraped	Page structure changed	Update the CSS selectors/XPaths
CSV not saving	File open/locked	Close the file before re-running script
Empty results	Anti-bot protection	Use time.sleep() and custom user-agent
Chrome mismatch	Version incompatibility	Update webdriver_manager or Chrome

Scope of the project

- Scrapes jobs only from Indeed India.
- Covers multiple pages (based on num_pages parameter).
- Saves results into a CSV file for further use (Excel, Power BI, etc.).
- Can be extended to include filters (salary, job type, remote jobs).

Methodology

1. Launch Chrome in headless mode.
2. Navigate to Indeed job listings.
3. Use Selenium to extract job data (title, company, location, etc.).
4. Append extracted data into a Pandas DataFrame.
5. Export results to a CSV file.
6. Quit the browser instance.

Conclusion

The Indeed Job Scraper automates the collection of job postings, reducing manual effort and providing structured job market data. It can be expanded to include more filters, advanced analytics, or real-time alerts. With Selenium and Pandas, the project is lightweight yet powerful, making it a useful tool for students, researchers, and job seekers.