TASK 6:

**-- Task 6: Web Scraping & Job Data Analysis**

**-- 1. Live Website (RemoteOK)**

* **Target URL:** https://remoteok.com/remote-data-engineer-jobs
* **Goal:** Scrape live Data Engineer job listings (Title, Company, Location, Link).
* **Notes:** Standard HTML; use a browser User-Agent header to avoid blocking.

**Tools & Technologies Used**

* **Python** – Programming language for automation and analysis.
* **Requests** – To fetch live HTML content from RemoteOK.
* **BeautifulSoup (bs4)** – For parsing and extracting job details from the HTML.
* **Pandas** – To store and manipulate scraped data in a DataFrame.
* **Google Colab** – Cloud-based environment to run and test the Python script.

**-- 2. Python Code Snippet:**

**Fetching Live Page**

import requests

from bs4 import BeautifulSoup

import pandas as pd

url = "https://remoteok.com/remote-data-engineer-jobs"

headers = {

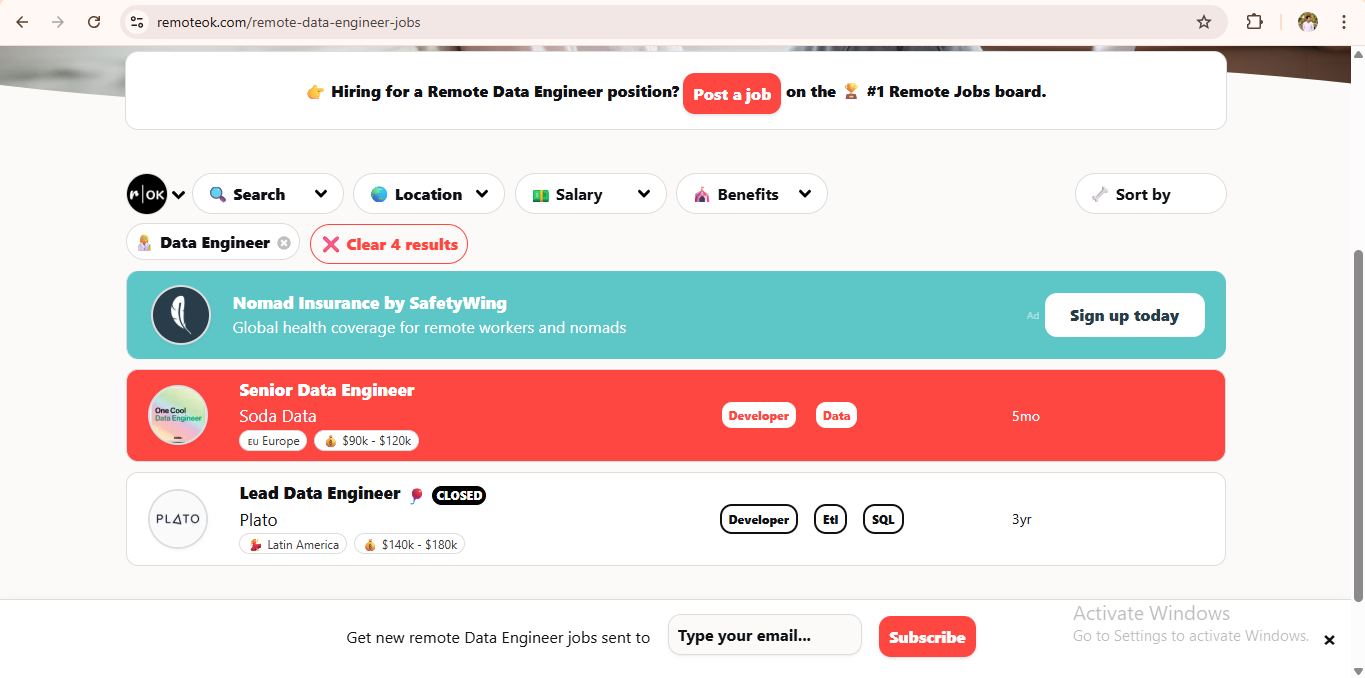
"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 "

"(KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"

}

response = requests.get(url, headers=headers)

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

****

**Parsing Job Listing**

rows = soup.find\_all("tr", class\_="job")

**Extracting Fields**

data = []

for r in rows:

title = r.find("h2", itemprop="title")

company = r.find("h3", itemprop="name")

loc = r.find("div", class\_="location")

link = r.find("a", class\_="preventLink")

if title and company and link:

data.append([

title.get\_text(strip=True),

company.get\_text(strip=True),

loc.get\_text(strip=True) if loc else "Not specified",

"https://remoteok.com" + link["href"]

])

