**Implement crawling (delays, respect rules) using time.sleep() :**

import requests

import time

from bs4 import BeautifulSoup

URL\_list = [

"https://www.aljazeera.com/news/",

"https://www.aljazeera.com/economy/",

"https://www.aljazeera.com/sports/",

"https://www.aljazeera.com/videos/",

"https://www.aljazeera.com/tag/human-rights/"

#'https://finance.yahoo.com/quote/7545.T/?p=7545.T',

#'https://finance.yahoo.com/quote/7447.T/?p=7447.T'

]

page\_titles = []

section\_names = []

j = 0

while j < len(URL\_list):

try:

time.sleep(2)

headers = {

'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'

}

r = requests.get(URL\_list[j], headers=headers)

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

title = soup.title.text.strip()

page\_titles.append(title)

section\_name = URL\_list[j].split("/")[-2]

section\_names.append(section\_name)

print(j, "-->", title)

print('Done ')

j += 1

except Exception as e:

page\_titles.append("N/A")

section\_names.append(URL\_list[j].split("/")[-2])

print(j, "--> failed ")

print("Error:", e)

j += 1

A Python script was implemented to demonstrate crawling of Al Jazeera's public content sections.  
The script visits five main content URLs, extracts each page's <title> tag using BeautifulSoup, and introduces a delay of 2 seconds between requests to avoid overwhelming the server.  
All requests use a realistic User-Agent string.

Introducing a 2-second delay (`time.sleep(2)`) between each HTTP request ensures that the crawler behaves responsibly, simulating human-like browsing behavior.  
This prevents triggering anti-bot systems and reduces the load on the website's servers.  
Delays like this are recommended for ethical and legal web scraping, particularly when no specific crawl-delay is provided in robots.txt.

All visited sections returned titles in the format: "[Section Name] | Today's latest from Al Jazeera".  
This is because Al Jazeera uses a consistent HTML <title> template across all major content sections.  
While the structure is uniform, the section name at the beginning helps distinguish the topic of each page.  
For example:  
- News → "News | Today's latest from Al Jazeera"  
- Economy → "Economy | Today's latest from Al Jazeera"  
- Sport → "Sport | Today's latest from Al Jazeera"

Output:

A screenshot of a computer

AI-generated content may be incorrect.

import time

import requests

from urllib.robotparser import RobotFileParser

url = "https://www.aljazeera.com/search/article"

#url = "https://www.aljazeera.com/news/"

user\_agent = "MyBot/1.0"

robots\_url = "https://www.aljazeera.com/robots.txt"

rp = RobotFileParser()

rp.set\_url(robots\_url)

rp.read()

if rp.can\_fetch(user\_agent, url):

delay = rp.crawl\_delay(user\_agent)

if delay is None:

delay = 2

print(f"Crawling {url} with delay {delay}s")

response = requests.get(url, headers={"User-Agent": user\_agent})

if response.status\_code == 200:

print("Page fetched successfully")

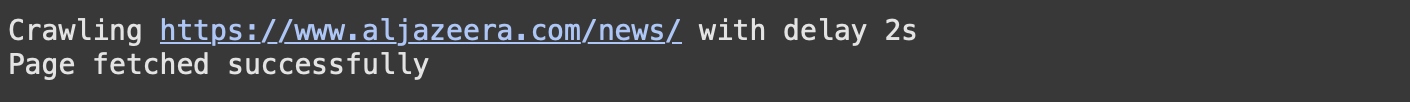
else:

print(f"Failed to fetch page: {response.status\_code}")

time.sleep(delay)

else:

print("Fetching disallowed by robots.txt")



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.