```
import requests
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import pandas as pd
from datetime import datetime



# ----------------------------
# BBC News (BeautifulSoup)
# ----------------------------
def get_bbc_news():
    url = "https://www.bbc.com/news"
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    headlines = []
    for h in soup.find_all("h2"):
        text = h.get_text(strip=True)
        if text:
            headlines.append({"Headline": text, "Source": "BBC"})
    return headlines



# ----------------------------
# Google News (Selenium)
# ----------------------------
def get_google_news():
    options = webdriver.ChromeOptions()
```

```
    options.add_argument("--headless")
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)

    url = "https://news.google.com/"
    driver.get(url)

    headlines = []
    elements = driver.find_elements(By.TAG_NAME, "h3")
    for elem in elements:
        text = elem.text.strip()
        if text:
            headlines.append({"Headline": text, "Source": "Google News"})

    driver.quit()
    return headlines



# ----------------------------
# Apply Keyword Filter
# ----------------------------
def filter_headlines(headlines, keywords):
    if not keywords:  # no filter, return all
        return headlines
    keywords = [k.lower() for k in keywords]
    return [h for h in headlines if any(k in h["Headline"].lower() for k in keywords)]



# ----------------------------
# Save to CSV
# ----------------------------
def save_to_csv(headlines, keywords=None, backup=False):
```

```
    df = pd.DataFrame(headlines)
    keyword_tag = "_".join(keywords) if keywords else "all"
    if backup:
        keyword_tag += "_backup"

    filename = f"news_{keyword_tag}_{datetime.now().strftime('%Y%m%d_%H%M%S')}.csv"
    df.to_csv(filename, index=False)
    print(f"✅ Saved {len(headlines)} headlines to {filename}")


# ----------------------------
# Main
# ----------------------------
if __name__ == "__main__":
    # Collect headlines
    bbc_headlines = get_bbc_news()
    google_headlines = get_google_news()
    all_news = bbc_headlines + google_headlines

    # Define keywords
    keywords = ["AI", "Sports", "Economy"]   # change as needed

    # Filter headlines
    filtered_news = filter_headlines(all_news, keywords)

    if filtered_news:
        # Save only filtered results
        save_to_csv(filtered_news, keywords)
    else:
        # No matches found → save all headlines as backup
        print("⚠ No headlines matched your keywords. Saving all headlines instead...")
```

```
save_to_csv(all_news, keywords, backup=True)
```