* Project Story: Daily BBC Top 10 Headlines Scraper

When we started this project, the goal was to build a daily-updating scraper for the top 10 news headlines from BBC News website (<https://www.bbc.com/news>), storing them in a CSV file for analysis or display. The idea was to combine web scraping and automation in Python. AI-assisted coding was leveraged for faster prototyping and further debugging.

* Initial Approach – Scraping HTML

We began by inspecting the BBC homepage and identifying the headline elements (**<h2 data-testid="card-headline">**).

Using **requests** with **BeautifulSoup**, we were able to fetch the page and extract headline text and links.

* Dealing with 403 error and data cleaning

Early on, we faced a 403 Forbidden error when accessing the page. We solved this by including a standard User-Agent header (**Mozilla/5.0**) to mimic a real browser.

Next, we handled duplicate headlines and filtered out non-article elements (**Next page button, Save button etc.**), making sure only real news stories were included.

* Saving data with the proper instructions

Initially, we saved scraped headlines to a CSV using **mode="a"**, which appended every run.

This led to CSV rows growing over time, creating dozens of rows even though we only wanted the top 10 per day.

We tried reading the existing CSV and using **concat** with **drop\_duplicates**, but it still didn’t solve the problem: old data kept accumulating and the total number of rows kept growing to 20-30-40, so ten rows were added at each new iteration of the whole code.

We realized the key was overwriting the CSV every day rather than appending.

The final approach scrapes up to 10 unique headlines, fills any missing spots with placeholders if BBC provides fewer than 10 (which is unlikely to happen, but had to be included nonetheless in case we would want to expand the project’s scope), adds today’s date, converts to a DataFrame, and writes it directly to CSV.

This guarantees exactly 10 rows every day, no duplicates, and a clean file ready for analysis.

* Automation and Future Steps

The code is structured to easily integrate a scheduler (**schedule or os-level cron/Task Scheduler**), so it can run daily automatically with a few additional manipulations if required.

Potential for further development and analysis - counting a particular word mentioned in every headline daily for a month – transferring data to a separate CSV file for frequency analysis to track public attention on specific topics over time.