HW 2
URL Summarizer for multiple LLMs

Create a LLM Streamlit application that can use several different Language Models (LLMs).

The application will allow users to input:

- A web page URL
- The type of summary.
- An output language
- The LLM model to use

Create a multi page HW App

1. Create a multi page app, with the title of “HW manager”.
2. Add the previously created HW1.py as a page in the main app (and add HW1.py to the “HWs” folder).
3. Add a new file named “HW2.py” to the “HWs” folder, this is a copy of the Lab2.py file, that you created, and will update for your HW.

Update the application to read a URL (rather than a PDF file or txt file)

4. Let the user enter a URL (at the top of the screen, not the sidebar).
5. In the sidebar, keep the menu for the type of summary (similar to Lab2).
6. Display the summary (as you did previously), but now it is for a URL
7. Below is a function to read the content from a URL (and return the text of that PDF file):

```
import requests
from bs4 import BeautifulSoup

def read_url_content(url):
    try:
        response = requests.get(url)
        response.raise_for_status()  # Raise an exception for HTTP errors
        soup = BeautifulSoup(response.content, 'html.parser')
        return soup.get_text()
    except requests.RequestException as e:
        print(f"Error reading {url}: {e}")
        return None
```

Select the language to output.

8. Create a dropdown menu to select the output language (ex. English and French). At Least with three options.
9. Update the prompt to ensure you are outputting the text in the correct language.

Select the LLM to use.

10. On the sidebar, add the option menu to select an option for different LLMs. (in addition to the 'use advanced model' checkbox that existed in Lab2)
11. Get keys for at least two other LLM API's besides OpenAI (ex. claude, cohere, gemini, mistral)
    a. After the user selects an LLM, the application should change to use the selected LLM.
    b. When a key (for the LLM) is entered, make sure the key is valid for that LLM.
    c. Reference: https://medium.com/@lars.chr.wiik/how-to-use-llm-apis-openai-claude-google-50bc7ce2c8de
12. Evaluate 3 LLMs with their 'advanced models' (need to make sure you have the correct key for the correct API)
    → Which LLM is "best" — explain your logic.
13. Do the same evaluation when using their less expensive model.
    a. Which LLM is best
    b. Are the more expensive models much better for this task?

Deploy the app

14. Deploy the application on the web.
15. Use secrets.toml
16. Make sure to update the requirements.txt file as needed

What to submit:

- Submit the link and your short write (which LLMs are better / worse; explain your logic)
- The HW2 python file (.py)