Instructions documentation

1. explaining how you approached the solution

* **BeautifulSoup**: This library was chosen for web scraping due to its simplicity and robustness in parsing HTML and XML documents. It allows efficient extraction of data from web pages with minimal code
* **NLTK (Natural Language Toolkit)**: NLTK was used for text preprocessing and analysis. Its extensive suite of tools for tokenization, stemming, and sentiment analysis made it an ideal choice

1. **Data Collection**: Using BeautifulSoup, I extracted text content from the target source. This ensured that all relevant information was parsed and cleaned efficiently.
2. **Text Preprocessing**:
   * **Tokenization**: The extracted content was tokenized into sentences and words using NLTK.
   * **Stemming**: I employed the Porter Stemmer from NLTK to reduce words to their base forms, which helped in identifying patterns and trends in the text.
   * **Stopwords Removal**: Common stopwords (e.g., "and," "the") were removed using NLTK’s pre-defined stopwords list to focus on meaningful words.
3. **Analysis and Visualization**:
   * Performed word frequency analysis to determine the output variables.
4. How to run the .py file to generate output
5. Save the Python script as **main.py.**
6. Open a terminal or command prompt and navigate to the directory containing the file. Ex: **cd sample** (where sample is the current working directory)
7. Run the script using the following command:

python main.py

1. **Include all dependencies required :**

* Python Libraries:

**beautifulsoup4**: For web scraping and data extraction.

**nltk:** For text processing and analysis.

**requests**: For handling HTTP requests to fetch web content.

* **Python** (Version 3.7 or higher)
* Use the following commands to install the necessary libraries:  
    
  **pip install beautifulsoup4 nltk requests**