1. Install Required Libraries: You'll need to install the following Python libraries:

pandas: To read data from Excel. beautifulsoup4: For web scraping. requests: To send HTTP requests. openpyxl: For reading the Excel file.

```
pip install pandas beautifulsoup4 requests openpyxl

```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.12.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.32.3)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.10/dist-packages (3.1.5)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4) (2.6)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests) (2024.8.30)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.10/dist-packages (from openpyxl) (1.1.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
## 2. Read Input File (input.xlsx):
# Use the pandas library to read the URLs from the Excel file.
```

```
import pandas as pd

# Load the Excel file
df = pd.read_excel("/content/Input.xlsx")

# Replace 'URL_ID' and 'URL' with the actual names of the columns in your Excel file
# Based on your global variable `df`, the DataFrame has 2 columns, and we assume the columns have names 'URL_ID' and 'URL'
urls = df[['URL_ID', 'URL']].values
```

3. Web Scraping Using BeautifulSoup: For each URL, you'll extract the title and text from the page, ignoring any headers, footers, or other irrelevant content. Here's an example of how to do this using BeautifulSoup:

```
import requests
from bs4 import BeautifulSoup

def extract_article_text(url):
    try:
        response = requests.get(url)
        if response.status_code == 200:
            soup = BeautifulSoup(response.content, 'html.parser')

            # Extract the title
            title = soup.title.get_text() if soup.title else "No Title"

```

```
            # Extract the main content, modify this based on the site's HTML structure
            article_body = soup.find('article')  # This may vary, adjust based on the actual site
            if article_body:
                text = article_body.get_text(separator=' ', strip=True)
            else:
                text = "No content found"

            return title, text
        else:
            return "Failed to retrieve", "No content"
    except Exception as e:
        return "Error occurred", str(e)

```

4. Save Extracted Text to a File: After extracting the text, save it to a text file named after the URL_ID.

```
def save_article_text(url_id, title, text):
    filename = f"{url_id}.txt"
    with open(filename, 'w', encoding='utf-8') as f:
        f.write(f"Title: {title}\n\n")
        f.write(text)

```

5. Iterate Through All URLs and Extract Data:

```
for url_id, url in urls:
    title, text = extract_article_text(url)
    save_article_text(url_id, title, text)
    print(f"Article {url_id} processed and saved.")

```

```
Article Netclan20241017 processed and saved.
Article Netclan20241018 processed and saved.
Article Netclan20241019 processed and saved.
Article Netclan20241020 processed and saved.
Article Netclan20241021 processed and saved.
Article Netclan20241022 processed and saved.
Article Netclan20241023 processed and saved.
Article Netclan20241024 processed and saved.
Article Netclan20241025 processed and saved.
Article Netclan20241026 processed and saved.
Article Netclan20241027 processed and saved.
Article Netclan20241028 processed and saved.
Article Netclan20241029 processed and saved.
Article Netclan20241030 processed and saved.
Article Netclan20241031 processed and saved.
Article Netclan20241032 processed and saved.
Article Netclan20241033 processed and saved.
Article Netclan20241034 processed and saved.
Article Netclan20241035 processed and saved.
Article Netclan20241036 processed and saved.
Article Netclan20241037 processed and saved.
```

```
Article Netclan20241038 processed and saved.
Article Netclan20241039 processed and saved.
Article Netclan20241040 processed and saved.
Article Netclan20241041 processed and saved.
Article Netclan20241042 processed and saved.
Article Netclan20241043 processed and saved.
Article Netclan20241044 processed and saved.
Article Netclan20241045 processed and saved.
Article Netclan20241046 processed and saved.
Article Netclan20241047 processed and saved.
Article Netclan20241048 processed and saved.
Article Netclan20241049 processed and saved.
Article Netclan20241050 processed and saved.
Article Netclan20241051 processed and saved.
Article Netclan20241052 processed and saved.
Article Netclan20241053 processed and saved.
Article Netclan20241054 processed and saved.
Article Netclan20241055 processed and saved.
Article Netclan20241056 processed and saved.
Article Netclan20241057 processed and saved.
Article Netclan20241058 processed and saved.
Article Netclan20241059 processed and saved.
Article Netclan20241060 processed and saved.
Article Netclan20241061 processed and saved.
Article Netclan20241062 processed and saved.
Article Netclan20241063 processed and saved.
Article Netclan20241064 processed and saved.
Article Netclan20241065 processed and saved.
Article Netclan20241066 processed and saved.
Article Netclan20241067 processed and saved.
Article Netclan20241068 processed and saved.
Article Netclan20241069 processed and saved.
Article Netclan20241070 processed and saved.
Article Netclan20241071 processed and saved.
Article Netclan20241072 processed and saved.
Article Netclan20241073 processed and saved.
```

```
!pip install textblob
from textblob import TextBlob
import pandas as pd

# Define necessary functions
def positive_score(text):
    """Calculates the positive score of a text using TextBlob."""
    analysis = TextBlob(text)
    return analysis.sentiment.polarity  # Assuming polarity represents positive sentiment

def negative_score(text):
    """Calculates the negative score of a text using TextBlob."""
    analysis = TextBlob(text)
    return analysis.sentiment.subjectivity - analysis.sentiment.polarity  # Example calculation

def polarity_score(text):
    # Add your implementation
    pass

def subjectivity_score(text):
```

```
    # Add your implementation
    pass

def avg_sentence_length(text):
    # Add your implementation
    pass

def percentage_complex_words(text):
    # Add your implementation
    pass

def fog_index(text):
    # Add your implementation
    pass

def avg_words_per_sentence(text):
    # Add your implementation
    pass

def complex_word_count(text):
    # Add your implementation
    pass

def word_count(text):
    # Add your implementation
    pass

def syllables_per_word(text):
    # Add your implementation
    pass

def personal_pronouns_count(text):
    # Add your implementation
    pass

def avg_word_length(text):
    # Add your implementation
    pass

output_data = []

for url_id in df['URL_ID']:
    with open(f'{url_id}.txt', 'r', encoding='utf-8') as f:
        text = f.read()

    analysis = {
        'URL_ID': url_id,
        'POSITIVE SCORE': positive_score(text),
        'NEGATIVE SCORE': negative_score(text),
        'POLARITY SCORE': polarity_score(text),
        'SUBJECTIVITY SCORE': subjectivity_score(text),
        'AVG SENTENCE LENGTH': avg_sentence_length(text),
        'PERCENTAGE OF COMPLEX WORDS': percentage_complex_words(text),
        'FOG INDEX': fog_index(text),
        'AVG NUMBER OF WORDS PER SENTENCE': avg_words_per_sentence(text),
```

```
        'COMPLEX WORD COUNT': complex_word_count(text),
        'WORD COUNT': word_count(text),
        'SYLLABLE PER WORD': syllables_per_word(text),
        'PERSONAL PRONOUNS': personal_pronouns_count(text),
        'AVG WORD LENGTH': avg_word_length(text)
    }

    output_data.append(analysis)

# Save to Excel
output_df = pd.DataFrame(output_data)
output_df.to_excel("Output Data Structure.xlsx", index=False)
```

```
Requirement already satisfied: textblob in /usr/local/lib/python3.10/dist-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in /usr/local/lib/python3.10/dist-packages (from textblob) (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (2024.9.11)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk>=3.1->textblob) (4.66.5)
```