Approach to the Solution:

1. **Data Loading and Extraction:**
    - Used pandas to read the input file ('Input.xlsx').
    - Utilized BeautifulSoup and requests for web scraping to extract article text from each URL.
    - Saved the extracted article text in text files with the URL_ID as the file name.
2. **Text/Data Analysis:**
    - Imported necessary libraries, including NLTK for natural language processing.
    - Read stop words lists and master dictionaries for sentiment analysis.
    - Created functions for text cleaning and calculating various variables related to sentiment, readability, and other factors.
    - Applied text analysis to each article, calculated sentiment scores, readability scores, and other variables.
    - Combined the results into a single DataFrame named **output_data**.
3. **Output Handling:**
    - Dropped unnecessary columns from **output_data** DataFrame.
    - Converted selected columns to uppercase as per the specified format.
    - Loaded the existing Excel file ('Output Data Structure.xlsx').
    - Merged the existing data with the new **output_data** DataFrame based on the URL_ID.
    - Renamed the resulting DataFrame to **merged_df_2**.
    - Exported the final DataFrame to a new Excel file named 'New_Output_File.xlsx'.

How to Run the .py File to Generate Output:

1. Ensure you have the required dependencies installed. You can install them using the following commands:

   ```
   pip install pandas beautifulsoup4 requests openpyxl nltk
   ```

2. Save the provided Python code in a file, let's say **data_analysis_script.py**.

3. Run the script using the following command in your terminal or command prompt:

   ```
   python data_analysis_script.py
   ```

4. The script will execute the entire process, from data loading to analysis and output generation.

Dependencies:

- pandas
- beautifulsoup4
- requests
- openpyxl
- nltk