Data Extraction Script Documentation

1. Objective

This script extracts article titles and main text from URLs listed in `Input.xlsx`. The extracted text is saved as `.txt` files with `URL_ID` as the filename. The script ensures that only relevant article content is captured, excluding ads, side links, navigation elements, and redundant titles.

2. Approach

2.1. Setting Up Selenium WebDriver

- Uses **Selenium** with **headless Chrome** for automated webpage interaction.
- WebDriver is set up with **ChromeDriverManager**, ensuring automatic driver installation.
- Key browser options:
  - `--headless`: Runs Chrome in the background.
  - `--disable-gpu`, `--no-sandbox`, `--disable-dev-shm-usage`: Optimize performance.

2.2. Extracting Article Text

- Loads the webpage and waits (`time.sleep(3)`) for content to load.
- Extracts **article title** from `<h1>`.
- Identifies the **main article container** using `.td-post-content`.
- Removes unwanted elements such as:
  - Advertisements (`td-a-rec`)
  - Side links (`td_block_wrap`)
  - Social sharing buttons (`td-post-sharing`)
  - Navigation elements (`td-post-next-prev`)
- Extracts structured content:
  - **Headings (h1, h2, h3)** → Converted to bold `**Title**`
  - **Paragraphs (p)**
  - **Lists (ul, ol, li)** → Formatted as `- List Item`
  - **Hyperlinks (a)** → Captured as `[Link: URL]`
- Captures **Solution Architecture Links** if present after the corresponding heading.

2.3. Saving Extracted Content

- Creates a directory `extracted_articles/` if not present.

- Saves each article as `{URL_ID}.txt` with proper spacing and formatting.

2.4. Iterating Over URLs

- Reads `Input.xlsx` containing `URL_ID` and `URL`.
- Processes each URL sequentially:
  - Extracts article text using Selenium.
  - Saves it to a `.txt` file.
  - Logs progress to the console.

3. Running the Script

Ensure you have Python installed and run the following command to install required packages:

```
pip install selenium pandas openpyxl webdriver-manager
```

Run the following command in the terminal:

```
python data_extraction.py
```

Ensure `Input.xlsx` is in the same directory.

4. Dependencies & Notes

Package	**Purpose**
`selenium`	Automates web browsing to extract text
`pandas`	Reads `Input.xlsx` for URL processing
`openpyxl`	Supports reading Excel files
`webdriver-manager`	Manages ChromeDriver automatically