### **Web Scraping Task**

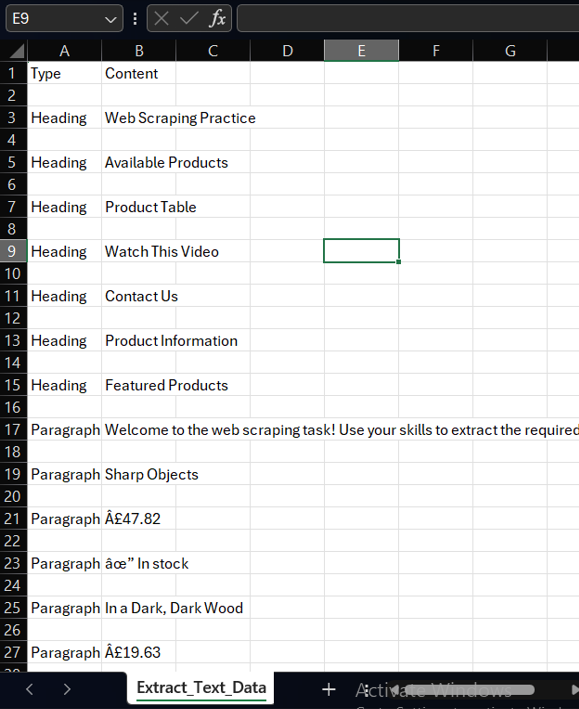
#### **Task Description:**

You are tasked to perform web scraping on a provided HTML page that contains different types of elements. The goal is to extract spexcific data from the page and process it into structured formats such as CSV or JSON.

### [**https://www.baraasallout.com/test.html**](https://www.baraasallout.com/test.html)

### **Steps to Complete the Task**

#### **1. Extract Text Data: ()**

* Extract all headings (<h1>, <h2>).
* Extract all text content inside <p> and <li> tags
* Save this data into a **Extract\_Text\_Data.CSV** file.
* <https://www.pythontutorial.net/python-basics/python-write-csv-file/>

#### **2. Extract Table Data:**

* Extract data from the table, including:
  + Product Name.
  + Price.
  + Stock Status.
* Save this data into a **Extract\_Table\_Data.CSV** file.
* <https://www.pythontutorial.net/python-basics/python-write-csv-file/>

#### **3. Extract Product Information (Cards Section):**

* Extract data from the book cards at the bottom of the page, including:
  + Book Title.
  + Price.
  + Stock Availability.
  + Button text (e.g., "Add to basket").
* Save the data into a  **Product\_Information.**JSON file.
* <https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/>

#### **4. Extract Form Details:**

* Extract all input fields from the form, including:
  + Field name (e.g., username, password).
  + Input type (e.g., text, password, checkbox, etc.).
  + Default values, if any.
  + Save the data into a JSON file.
  + <https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/>

#### **5. Extract Links and Multimedia:**

* Extract the hyperlink (<a> tag) and its href value.
* Extract the video link from the <iframe> tag.
* Save the data into a JSON file.
* <https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/>

#### **6. Scraping Challenge:**

Students must write a script to extract data from the **Featured Products** section with the following requirements:

* Product Name: Located within <span class="name">.
* Hidden Price: Located within <span class="price">, which has style="display: none;".
* Available Colors: Located within <span class="colors">.
* Product ID: The value stored in the data-id attribute.

##### **Example Output:**

[

{'id': '101', 'name': 'Wireless Headphones', 'price': '$49.99', 'colors': 'Black, White, Blue'},

{'id': '102', 'name': 'Smart Speaker', 'price': '$89.99', 'colors': 'Grey, Black'},

{'id': '103', 'name': 'Smart Watch', 'price': '$149.99', 'colors': 'Black, Silver, Gold'}

]

### **Deliverables:**

1. Extracted Data:
   * CSV file containing the table data.
   * JSON file containing data from the book cards and headings.
2. Code Script:
   * Provide the Python script used for scraping (comment your code for clarity).
3. Documentation:
   * Write a simple explanation of your approach, tools used, and any challenges faced.
4. Upload files on your GitHub and submit the repo link.