Task Web Scrapping

Import Library and Web:

```python
import requests
from bs4 import BeautifulSoup
```

```python
soup = BeautifulSoup(response.text, 'html.parser')
```

```python
url = ' https://www.baraasallout.com/test.html'
response = requests.get(url)
print(response.text)
```

Task 1

- **Extract Text Data:**
  - **Extract all headings (<h1>, <h2>).**
  - **Extract all text content inside <p> and <li> tags.**
  - **Savethis data into a Extract_Text_Data.CSV file.**
  - **https://www.pythontutorial.net/python-basics/python-write-csv-file/**

```python
headers_1 = soup.find_all('h1')
headers_1
```

```
[20]
[<h1>Web Scraping Practice</h1>]
```

```python
headers_2 = soup.find_all('h2')
headers_2
```

```
[21]
[<h2>Available Products</h2>,
 <h2>Product Table</h2>,
 <h2>Watch This Video</h2>,
 <h2>Contact Us</h2>,
 <h2>Product Information</h2>,
 <h2>Featured Products</h2>]
```

```python
paragraph = soup.find('p')
paragraph
```

[9] Python

```
<p>Welcome to the web scraping task! Use your skills to extract the required data from this page.</p>
```

```python
all_p = soup.find_all('p')
all_p
```

[10] Python

```
[<p>Welcome to the web scraping task! Use your skills to extract the required data from this page.</p>,
 <p><strong>Sharp Objects</strong></p>,
 <p style="color: green;">£47.82</p>,
 <p style="color: green;">✓ In stock</p>,
 <p><strong>In a Dark, Dark Wood</strong></p>,
 <p style="color: green;">£19.63</p>,
 <p style="color: green;">✓ In stock</p>,
 <p><strong>The Past Never Ends</strong></p>,
 <p style="color: green;">£56.50</p>,
 <p style="color: green;">✓ In stock</p>,
 <p><strong>A Murder in Time</strong></p>,
 <p style="color: green;">£16.64</p>,
 <p style="color: green;"> Out stock</p>,
 <p class="name">Wireless Headphones</p>,
 <p class="price" style="display: none;">$49.99</p>,
```

```python
first_ul = soup.find('ul')

```

[25] Python

```python
all_ul_li = first_ul.find_all('li')
all_ul_li
```

[26] Python

```
[<li class="highlight">Laptop</li>,
 <li>Smartphone</li>,
 <li>Tablet</li>,
 <li>Smartwatch</li>]
```

```python
for li in all_ul_li:
    print(li.get_text())
```

[27] Python

```
Laptop
Smartphone
Tablet
Smartwatch
```

```python

# Prepare the data
data = (
    [('Heading', heading) for heading in headings] +
    [('Paragraph', paragraph) for paragraph in all_p] +
    [('List Item', list_item) for list_item in all_ul_li]
)

# Convert the data into a DataFrame
df = pd.DataFrame(data, columns=['Type', 'Content'])

csv_file = 'Extract_Text_Data.csv'
df.to_csv(csv_file, index=False, encoding='utf-8')

print(f"Data has been saved to {csv_file}.")

```

Task 2

- **Extract Table Data:**
  - **Extract data from the table, including:**
  - **Product Name.**
  - **Price.**
  - **StockStatus.**
  - **Savethis data into a Extract_Table_Data.CSV file.**
  - **https://www.pythontutorial.net/python-basics/python-write-csv-fil**

```
1 tables = soup.find_all('table')
2 tables
```

[32] Python

```
[<table>
<tr>
<th>Product</th>
<th>Price</th>
<th>In Stock</th>
</tr>
<tr>
<td>Laptop</td>
<td>$1000</td>
<td>Yes</td>
</tr>
<tr>
<td>Smartphone</td>
<td>$800</td>
<td>No</td>
</tr>
<tr>
<td>Tablet</td>
<td>$500</td>
<td>Yes</td>
</tr>
</table>]
```

```
1 all_divs = soup.find_all('div')
2 all_divs
```

[29] Python

```
[<div style="display: flex; justify-content: space-around; margin-top: 20px;">
<div style="text-align: center; width: 200px; border: 1px solid #ddd; padding: 10px; border-radius: 5px;">
<img alt="Sharp Objects" src="book1.jpg" style="width: 100%; height: auto; border-radius: 5px;"/>
<p><strong>Sharp Objects</strong></p>
<p style="color: green;">£47.82</p>
<p style="color: green;">✓ In stock</p>
<button style="background-color: blue; color: white; border: none; padding: 10px 15px; border-radius: 5px; cursor: pointer;">Add to basket</button>
</div>
<div style="text-align: center; width: 200px; border: 1px solid #ddd; padding: 10px; border-radius: 5px;">
<img alt="In a Dark, Dark Wood" src="book2.jpg" style="width: 100%; height: auto; border-radius: 5px;"/>
<p><strong>In a Dark, Dark Wood</strong></p>
<p style="color: green;">£19.63</p>
<p style="color: green;">✓ In stock</p>
<button style="background-color: blue; color: white; border: none; padding: 10px 15px; border-radius: 5px; cursor: pointer;">Add to basket</button>
</div>
<div style="text-align: center; width: 200px; border: 1px solid #ddd; padding: 10px; border-radius: 5px;">
<img alt="The Past Never Ends" href="http://books.toscrape.com/media/cache/c0/59/c05972805aa7201171b8fc71a5b00292.jpg" style="width: 100%; height: au
<p><strong>The Past Never Ends</strong></p>
<p style="color: green;">£56.50</p>
<p style="color: green;">✓ In stock</p>
<button style="background-color: blue; color: white; border: none; padding: 10px 15px; border-radius: 5px; cursor: pointer;">Add to basket</button>
</div>
<div style="text-align: center; width: 200px; border: 1px solid #ddd; padding: 10px; border-radius: 5px;">
<img alt="A Murder in Time" src="book4.jpg" style="width: 100%; height: auto; border-radius: 5px;"/>
<p><strong>A Murder in Time</strong></p>
```

```python
# Prepare the data for saving to CSV
data = {
    'Table Data': table_data,
    'BG Yellow Elements': bg_yellow_elements,
    'DIV Content': all_divs
}

# Adjust for different lengths (pad with None)
max_length = max(len(data['Table Data']), len(data['BG Yellow Elements']), len(data['DIV Content']))
data['Table Data'] += [None] * (max_length - len(data['Table Data']))
data['BG Yellow Elements'] += [None] * (max_length - len(data['BG Yellow Elements']))
data['DIV Content'] += [None] * (max_length - len(data['DIV Content']))

# Create a DataFrame
df = pd.DataFrame({
    'Table Data': ['; '.join(row) if row else None for row in data['Table Data']],  # Join table rows
    'BG Yellow Elements': data['BG Yellow Elements'],
    'DIV Content': data['DIV Content']
})

csv_file = 'Extract_Table_Data.csv'
df.to_csv(csv_file, index=False, encoding='utf-8')

print(f"Data has been saved to {csv_file}.")
```

Task 3

- **Extract Product Information (Cards Section):**
  - **Extract data from the book cards at the bottom of the page, including:**
    - **BookTitle.**
    - **Price.**
    - **StockAvailability.**
    - **Button text (e.g., "Add to basket").**
    - **Savethedata into a Product_Information.JSON file.**
    - **https://www.geeksforgeeks.org/how-to-convert-python- dic|**

```python
book_cards = soup.find_all('div', style=lambda value: value and 'text-align: center' in value)

books_data = []
for card in book_cards:
    title = card.find('p').strong.text
    price = card.find('p', style=lambda value: value and 'color: green' in value).text
    stock = card.find_all('p', style=lambda value: value and 'color: green' in value)[-1].text.strip()
    button_text = card.find('button').text.strip()

    books_data.append({
        'BookTitle': title,
        'Price': price,
        'StockAvailability': stock,
        'ButtonText': button_text
    })

for book in books_data:
    print(book)
```

```
{'BookTitle': 'Sharp Objects', 'Price': '£47.82', 'StockAvailability': '✓ In stock', 'ButtonText': 'Add to basket'}
{'BookTitle': 'In a Dark, Dark Wood', 'Price': '£19.63', 'StockAvailability': '✓ In stock', 'ButtonText': 'Add to basket'}
{'BookTitle': 'The Past Never Ends', 'Price': '£56.50', 'StockAvailability': '✓ In stock', 'ButtonText': 'Add to basket'}
{'BookTitle': 'A Murder in Time', 'Price': '£16.64', 'StockAvailability': 'Out stock', 'ButtonText': 'Add to basket'}
```

```python
    # Append extracted information to the products list
    products.append({
        "BookTitle": title,
        "Price": price,
        "StockAvailability": stock_availability,
        "ButtonText": button
    })

json_file = 'Product_Information.json'
with open(json_file, 'w', encoding='utf-8') as file:
    json.dump(products, file, indent=4, ensure_ascii=False)

print(f"Product data has been saved to {json_file}.")
```

Task 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.**
  - **Savethedata into a JSON file.**
  - **https://www.geeksforgeeks.org/how-to-convert-python-dictionary-to-json/**

```
[51]
forms = soup.find_all('form')
forms
```

Python

```
[<form>
<label for="username">Username:</label>
<input id="username" name="username" placeholder="Enter your username" type="text"/>
<label for="password">Password:</label>
<input id="password" name="password" placeholder="Enter your password" type="password"/>
<label for="options">Choose an option:</label>
<select id="options" name="options">
<option value="option1">Option 1</option>
<option value="option2">Option 2</option>
<option value="option3">Option 3</option>
</select>
<label>
<input name="terms" type="checkbox"/> I agree to the terms and conditions
          </label>
<input type="submit" value="Submit"/>
</form>]
```

```
[57]
# Extract input fields
input_fields = first_form.find_all('input')
input_fields
```

```
[<input id="username" name="username" placeholder="Enter your username" type="text"/>,
 <input id="password" name="password" placeholder="Enter your password" type="password"/>,
 <input name="terms" type="checkbox"/>,
 <input type="submit" value="Submit"/>]
```

```
[54]
first_form.get_text()
```

```
'\nUsername:\n\nPassword:\n\nChoose an option:\n\nOption 1\nOption 2\nOption 3\n\n\n I agree to the terms and conditions\n          \n\n'
```

+ Code + Markdown

```python
fields_data = []

for field in input_fields:
    field_name = field.get('name', 'N/A')
    input_type = field.get('type', 'text')
    default_value = field.get('value', 'N/A')
    fields_data.append({
        'Field Name': field_name,
        'Input Type': input_type,
        'Default Value': default_value
    })

for field in fields_data:
    print(field)
```

```
{'Field Name': 'username', 'Input Type': 'text', 'Default Value': 'N/A'}
{'Field Name': 'password', 'Input Type': 'password', 'Default Value': 'N/A'}
{'Field Name': 'terms', 'Input Type': 'checkbox', 'Default Value': 'N/A'}
{'Field Name': 'N/A', 'Input Type': 'submit', 'Default Value': 'Submit'}
```

```python

for form in forms:
    inputs = form.find_all('input')
    input_details = []
    for input_tag in inputs:
        field_name = input_tag.get('name', 'N/A')  # Field name
        input_type = input_tag.get('type', 'text')  # Input type
        default_value = input_tag.get('value', None)  # Default value
        input_details.append({
            "FieldName": field_name,
            "InputType": input_type,
            "DefaultValue": default_value
        })
    form_details.append(input_details)

form_json_file = 'Form_Details.json'
with open(form_json_file, 'w', encoding='utf-8') as file:
    json.dump(form_details, file, indent=4, ensure_ascii=False)

```

Task 5

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

```
[73]
1 links = soup.find_all('a')
2 links
Python
[]
```

```
1 first_link.get_text()
2 first_link.attrs # Not Existing attributes
Python
AttributeError                            Traceback (most recent call last)
Cell In[34], <a href='vscode-notebook-cell:?execution_count=34&line=1'>line 1</a>
----> <a href='vscode-notebook-cell:?execution_count=34&line=1'>1</a> first_link.get_text()
      <a href='vscode-notebook-cell:?execution_count=34&line=2'>2</a> first_link.attrs

AttributeError: 'NoneType' object has no attribute 'get_text'

1 first_link.attrs['href'] # Not  Existing attributes
Python
AttributeError                            Traceback (most recent call last)
Cell In[36], <a href='vscode-notebook-cell:?execution_count=36&line=1'>line 1</a>
----> <a href='vscode-notebook-cell:?execution_count=36&line=1'>1</a> first_link.attrs['href']

AttributeError: 'NoneType' object has no attribute 'attrs'
```

>>>>>> No <a> in the site

- Extract the video link from the iframe tag.

```
[79]
1 iframe = soup.find('iframe')
2 iframe
Python
<iframe height="315" src="https://www.youtube.com/watch?v=ujf9RNuBdCU" width="560"></iframe>
```

```
8
9  # Prepare the data
10 multimedia_data = {
11     "Hyperlinks": hyperlinks,
12     "Videos": videos
13 }
14
15 links_json_file = 'Links_and_Multimedia.json'
16 with open(links_json_file, 'w', encoding='utf-8') as file:
17     json.dump(multimedia_data, file, indent=4, ensure_ascii=False)
18
```

Task 6

- **Scraping Challenge**
  - **Product Name: Located within .**
  - **HiddenPrice: Located within , which has style="display: none;".**
  - **Available Colors: Located within .**
  - **Product ID: The value stored in the data-id attribute.**
  - **ExampleOutput:**

  [
  {'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'}
  ]

```python
products = []
for card in product_cards:
    product_id = card.get("data-id")
    name = card.find("p", class_="name").text
    price = card.find("p", class_="price").text.strip()  # Extract hidden price
    colors = card.find("p", class_="colors").text.replace("Available colors: ", "").strip()
    products.append({
        "id": product_id,
        "name": name,
        "price": price,
        "colors": colors
    })

for product in products:
    print(product)
```

```
{'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'}
```

Thanks