**Lab Exercise 3.0- Using BeautifulSoup for Web Scraping for RPA (Robotic Process Automation)**

**Objective**

* Learn to use BeautifulSoup to retrieve and display the HTML structure of a webpage.

**Prerequisites**

Install the required libraries using pip:

pip install requests beautifulsoup4

**Exercise Code**

import requests

from bs4 import BeautifulSoup

# Step 1: Define the target URL

url = "https://example.com" # Replace with the URL you want to scrape

# Step 2: Send an HTTP GET request to the URL

response = requests.get(url)

# Step 3: Check if the request was successful

if response.status\_code == 200:

print("Successfully fetched the webpage!")

# Step 4: Parse the HTML content using BeautifulSoup

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

# Step 5: Print the raw HTML content (formatted for readability)

print("\n--- Webpage HTML Content ---\n")

print(soup.prettify()) # Prettify makes the HTML easier to read

else:

print(f"Failed to fetch the webpage. Status code: {response.status\_code}")

**Steps to Execute**

1. **Copy and Paste the Code**:
   * Open your Python IDE or text editor.
   * Paste the code into a new Python file.
2. **Run the Script**:
   * Replace https://example.com with the URL of a webpage you want to scrape.
   * Run the script in your IDE or terminal.
3. **Output**:
   * If the request is successful, the script will print the HTML structure of the webpage.

**Expected Output**

For the default URL https://example.com, the output will look like:

Successfully fetched the webpage!

--- Webpage HTML Content ---

<!DOCTYPE html>

<html>

<head>

<title>

Example Domain

</title>

...

</head>

<body>

<h1>

Example Domain

</h1>

<p>

This domain is for use in illustrative examples in documents.

</p>

...

</body>

</html>

**Lab Task Variations**

1. **Extract Specific Elements**:
   * Modify the script to extract and print only specific elements, such as titles or headings:

print("\n--- Page Title ---")

print(soup.title.text)

1. **Save HTML to a File**:
   * Save the HTML content to a local file:

with open("webpage.html", "w", encoding="utf-8") as file:

file.write(soup.prettify())

print("HTML content saved to 'webpage.html'")

1. **Test with Other URLs**:
   * Try scraping different websites and observe the HTML structure.