Q). Write a microservice in python that takes a URL of a static website as an input and identifies all the unique words and how many times they occur on a web page. The output is a list of words in JSON format along with the frequency of occurrence (number of times the word is repeated).

Ans:-

Certainly! Here's an updated version of the Python
microservice using Flask that takes a URL of a static website as input and returns the unique words along with their frequencies in JSON format:

python Code:-

```
import requests
from flask import Flask, request, jsonify
from bs4 import BeautifulSoup
from collections import defaultdict

app = Flask(__name__)

def get_words_from_url(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    text = soup.get_text(separator=' ')
    words = text.split()
    return words

@app.route('/word_frequency', methods=['POST'])
def word_frequency():
    url = request.json['url']
    words = get_words_from_url(url)
    word_counts = defaultdict(int)
    for word in words:
        word_counts[word] += 1
    unique_words = [{'word': word, 'frequency': count} for word, count in word_counts.items()]
    return jsonify(unique_words)

if __name__ == '__main__':
    app.run()
```

This code is similar to the previous example, but instead of using the `Counter` class from the `collections` module, we use a `defaultdict` to count the frequency of each word. This allows us to simplify the code by avoiding the need to import the `Counter` class.

To run this microservice, make sure you have Flask, requests, and BeautifulSoup installed. You can install them using pip:

```
pip install flask requests beautifulsoup4
```

Save the code in a file called `app.py` and run it using the following command:

```
python app.py
```

The microservice will start running on http://localhost:5000/word_frequency. You can send a POST request to this URL with the JSON payload containing the URL of the static website you want to analyze. The response will contain the unique words and their frequencies in JSON format.

Note: This example assumes that the static website is publicly accessible and does not require any authentication or authorization.

Output:-

Sure! Here's an example of the output you can expect from the microservice:

Input JSON:

```
{
  "url": "https://example.com"
}
```

Output JSON:

```
[
  {"word": "Lorem", "frequency": 3},
  {"word": "ipsum", "frequency": 2},
  {"word": "dolor", "frequency": 2},
  {"word": "sit", "frequency": 1},
  {"word": "amet", "frequency": 1},
  {"word": "consectetur", "frequency": 1},
  {"word": "adipiscing", "frequency": 1},
  {"word": "elit", "frequency": 1}
]
```

In this example, we assume that the webpage at `https://example.com` contains the following text:

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor amet, adipiscing elit.

The microservice will identify the unique words and their frequencies. The word "Lorem" occurs 3 times, "ipsum" and "dolor" occur 2 times each, and the rest of the words occur once. The output is returned as a JSON array, where each object contains a "word" field and a "frequency" field.