Programme	:	**B.Tech** Semester : **Win Sem 21-22**
Course	:	**Web Mining Lab** Code : **CSE3024**
Faculty	:	**Dr.Bhuvaneswari A** Slot : **L7+L8**
Date	:	**28-01-2022** Marks : **10 Points**

Vaibhav Agarwal

19BCE1413

1. Build the inverted index for the following documents:

ID1 : Selenium is a portable framework for testing web applications

ID2 : Beautiful Soup is useful for web scraping

ID3: It is a python package for parsing the pages
ID4: Java programming can be used for web applications
ID5: scraping web and crawling web is useful

```python
import math
import re
import nltk
import string
import ssl

try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    pass
else:
    ssl._create_default_https_context = _create_unverified_https_context
```

[9] Python

```python
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
```

[10] Python

```python
nltk.download("punkt")
```

[11] Python

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/vaibhavagarwal/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

True

```python
docs = [
    "Selenium is a portable framework for testing web applications",
"Beautiful Soup is useful for web scraping",
"It is a python package for parsing the pages",
"Java programming can be used for web applications",
"scraping web and crawling web is useful"]
print(docs)
```

[4] Python

```
['Selenium is a portable framework for testing web applications', 'Beautiful Soup is useful for web scraping', 'It is a python package for parsing the pages', 'Java programming can be used for web applications', 'scraping web and crawling web is useful']
```

```python
def textpreprocess(text):
    s = text.lower()
    s = s.replace('/[^A-Za-z0-9]/g', '')
    s = s.strip()
    words = word_tokenize(s)
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]
    return words
```

[5] Python

```python
nltk.download("punkt")
nltk.download('stopwords')
```

[12] Python

```
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/vaibhavagarwal/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/vaibhavagarwal/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

True

```python
def findOccurance(text, word) :
    text = text.replace('/[^A-Za-z0-9]/g', '')
    text = text.replace(' ', ' ')
    text = text.lower()
    text_words = text.strip().split()
    word_count = 0
    word_positions = []
    for i in range(len(text_words)) :
        if word == text_words[i] :
            word_count += 1
            word_positions.append(i)
    return (word_count, word_positions)
```

[7] Python

```python
inverted_index = {}
for (i, doc) in enumerate(documents) :
    words = textpreprocess(doc)
    for word in words :
        if word not in inverted_index :
            inverted_index[word] = []
        occurance_count, occurance_pos_list = findOccurance(doc, word)
        inverted_index[word].append(((i+1), occurance_count, occurance_pos_list))
```

Python

```python
print("INVERTED INDEX")
for ind in inverted_index.items():
    print(ind)
```

Python

```
INVERTED INDEX
('selenium', [(1, 1, [0])])
('portable', [(1, 1, [3])])
('framework', [(1, 1, [4])])
('testing', [(1, 1, [6])])
('web', [(1, 1, [7]), (2, 1, [5]), (4, 1, [6]), (5, 2, [1, 4]), (5, 2, [1, 4])])
('applications', [(1, 1, [8]), (4, 1, [7])])
('beautiful', [(2, 1, [0])])
('soup', [(2, 1, [1])])
('useful', [(2, 1, [3]), (5, 1, [6])])
('scraping', [(2, 1, [6]), (5, 1, [0])])
('python', [(3, 1, [3])])
('package', [(3, 1, [4])])
('parsing', [(3, 1, [6])])
('pages', [(3, 1, [8])])
('java', [(4, 1, [0])])
('programming', [(4, 1, [1])])
('used', [(4, 1, [4])])
('crawling', [(5, 1, [3])])
```

2. Search following words using the inverted index
    a. Selenium AND web
    b. Soup
    c. Python OR java
    d. Web AND craw

Question 2

Search following words using the inverted index

a. Selenium AND web

```
print("Selenium word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if index[0]=="selenium" :
        for indexes in index[1]:
            print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])
print("web word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if(index[0]=="web"):
        for indexes in index[1]:
            print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])


print("Selenium AND web word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if index[0]=="selenium" or index[0]=="web":
        for indexes in index[1]:
            if(indexes[0]==1):
                print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])
```

[14] Python

```
Selenium word occurs in the following position
Doc no  no.of times  offset number
D 1          1          [0]
web word occurs in the following position
Doc no  no.of times  offset number
D 1          1          [7]
D 2          1          [5]
D 4          1          [6]
D 5          2          [1, 4]
D 5          2          [1, 4]
```

```
Selenium AND web word occurs in the following position
Doc no  no.of times  offset number
D 1          1          [0]
D 1          1          [7]
```

b. Soup

```
print("Soup word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if(index[0]=="soup"):
        for indexes in index[1]:
            print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])
```

[15] Python

```
Soup word occurs in the following position
Doc no  no.of times  offset number
D 2          1          [1]
```

c. Python OR java

```
print("Python OR java word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if index[0]=="python" or index[0]=="java":
        for indexes in index[1]:
            print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])
```

[16] Python

```
Python OR java word occurs in the following position
Doc no  no.of times  offset number
D 3          1          [3]
D 4          1          [0]
```

d. Web AND craw

```
print("Web AND craw word occurs in the following position")
print("Doc no  no.of times  offset number")
for index in inverted_index.items():
    if index[0]=="web" and index[0]=="craw":
        for indexes in index[1]:
            print("D",indexes[0],"        ",indexes[1],"        ",indexes[2])
```

[18] Python

```
Web AND craw word occurs in the following position
Doc no  no.of times  offset number
```

[ ] Python