In [ ]:

```
#Roll No- 3310 Srushti Bhoite
#Problem Statement:Write a program for the Information Retrieval System using appropri
#(such as NLTK, Open NLP, …)
#a. Text tokenization
#b. Count word frequency
#c. Remove stop words
#d. POS tagging
```

In [2]:

```
import nltk
#nltk.download('punkt')
from nltk.corpus import stopwords
#nltk.download('stopwords')
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist
from nltk.tag import pos_tag
#nltk.download('averaged_perceptron_tagger')

text = "This is a sample sentense. It contain multiple words and some of these repeat.

words = word_tokenize(text)
print("tokenized words:")
print (words)

words = [word.lower() for word in words]

fdist = FreqDist(words)
print("Word Frequency:")
for word, freq in fdist.items():
    print(f"{word}: {freq}")

stop_words = set(stopwords.words('english'))
filtered_words = [word for word in words if word.casefold() not in stop_words]
print("Filtered Words")
print(filtered_words)

pos_tags = pos_tag(words)
print("POS Tags:")
print(pos_tags)
```

```
tokenized words:
['This', 'is', 'a', 'sample', 'sentense', '.', 'It', 'contain', 'multiple', 'words',
'and', 'some', 'of', 'these', 'repeat', '.', 'We', 'will', 'analyze', 'this', 'text',
'using', 'NLP', 'text']
Word Frequency:
this: 2
is: 1
a: 1
sample: 1
sentense: 1
.: 2
it: 1
contain: 1
multiple: 1
words: 1
and: 1
some: 1
of: 1
these: 1
repeat: 1
we: 1
will: 1
analyze: 1
text: 2
using: 1
nlp: 1
Filtered Words
['sample', 'sentense', '.', 'contain', 'multiple', 'words', 'repeat', '.', 'analyze',
'text', 'using', 'nlp', 'text']
POS Tags:
[('this', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('sample', 'JJ'), ('sentense', 'NN'),
('.', '.'), ('it', 'PRP'), ('contain', 'VBZ'), ('multiple', 'JJ'), ('words', 'NNS'),
('and', 'CC'), ('some', 'DT'), ('of', 'IN'), ('these', 'DT'), ('repeat', 'NN'), ('.',
'.'), ('we', 'PRP'), ('will', 'MD'), ('analyze', 'VB'), ('this', 'DT'), ('text', 'N
N'), ('using', 'VBG'), ('nlp', 'JJ'), ('text', 'NN')]
```

```
In [ ]:
```