In [2]:

```
# wordnetlematizer library is the responsible for doing the lemmatization functio
import nltk
from nltk.stem import WordNetLemmatizer
```

In [3]:

```
# for removing repetitative word
from nltk.corpus import stopwords
```

In [4]:

```
ti="""Certainly! AI, or Artificial Intelligence, refers to the development and imp
ti
```

Out[4]:

```
'Certainly! AI, or Artificial Intelligence, refers to the development an
d implementation of computer systems that can perform tasks that typical
ly require human intelligence. AI aims to simulate human cognitive abili
ties such as learning, problem-solving, perception, and language underst
anding'
```

In [5]:

```
sentences = nltk.sent_tokenize(ti)
sentences

```

Out[5]:

```
['Certainly!',
 'AI, or Artificial Intelligence, refers to the development and implemen
tation of computer systems that can perform tasks that typically require
human intelligence.',
 'AI aims to simulate human cognitive abilities such as learning, proble
m-solving, perception, and language understanding']
```

In [7]:

```
lemmatizer = WordNetLemmatizer()
```

In [8]:

```
# now lemitizing for finding proper words


# Lemmatization
for i in range(len(sentences)):
    words = nltk.word_tokenize(sentences[i])
    words = [lemmatizer.lemmatize(word) for word in words if word not in set(stopw
    sentences[i] = ' '.join(words)

```

In [9]:

```
sentences
```

Out[9]:

```
['Certainly !',
 'AI , Artificial Intelligence , refers development implementation compu
ter system perform task typically require human intelligence .',
 'AI aim simulate human cognitive ability learning , problem-solving , p
erception , language understanding']
```

In [12]:

```

from nltk.tokenize import word_tokenize#Part of speaach
sent = 'Along with my personal funds, I have secured sponsorship from  who has gra
sent_tokens = word_tokenize(sent)
sent_tokens
```

Out[12]:

```
['Along',
 'with',
 'my',
 'personal',
 'funds',
 ',',
 'I',
 'have',
 'secured',
 'sponsorship',
 'from',
 'who',
 'has',
 'graciously',
 'agreed',
 'to',
 'support',
 'my',
 'visit',
 '.']
```

In [13]:

```
for token in sent_tokens:
    print(nltk.pos_tag([token]))#recognize part of speach
```

```
[('Along', 'IN')]
[('with', 'IN')]
[('my', 'PRP$')]
[('personal', 'JJ')]
[('funds', 'NNS')]
[(',', ',')]
[('I', 'PRP')]
[('have', 'VB')]
[('secured', 'VBN')]
[('sponsorship', 'NN')]
[('from', 'IN')]
[('who', 'WP')]
[('has', 'VBZ')]
[('graciously', 'RB')]
[('agreed', 'VBD')]
[('to', 'TO')]
[('support', 'NN')]
[('my', 'PRP$')]
[('visit', 'NN')]
[('.', '.')]
```

In [14]:

```
#Chunck recognition, NER
#person organization location  recognition
from nltk import ne_chunk
```

In [20]:

```
NE_sent = "AI aim simulate human cognitive ability learning school"
```

In [21]:

```
NE_tokens = word_tokenize(NE_sent)

#after tokenize need to add the pos tags
NE_tokens
```

Out[21]:

```
['AI',
 'aim',
 'simulate',
 'human',
 'cognitive',
 'ability',
 'learning',
 'school']
```

In [22]:

```
NE_tags = nltk.pos_tag(NE_tokens)
NE_tags
```

Out[22]:

```
[('AI', 'NNP'),
 ('aim', 'NN'),
 ('simulate', 'NN'),
 ('human', 'JJ'),
 ('cognitive', 'JJ'),
 ('ability', 'NN'),
 ('learning', 'VBG'),
 ('school', 'NN')]
```

In [23]:

```
NE_NER = ne_chunk(NE_tags)
print(NE_NER)
```

```
(S
  AI/NNP
  aim/NN
  simulate/NN
  human/JJ
  cognitive/JJ
  ability/NN
  learning/VBG
  school/NN)
```

In [25]:

```
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.2-cp310-cp310-macosx_10_9_x86_64.whl (160 k
B)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 160.5/160.5 kB 798.5 kB/s eta
0:00:00a 0:00:01
Requirement already satisfied: pillow in ./anaconda3/lib/python3.10/site
-packages (from wordcloud) (9.4.0)
Requirement already satisfied: matplotlib in ./anaconda3/lib/python3.10/
site-packages (from wordcloud) (3.7.0)
Requirement already satisfied: numpy>=1.6.1 in ./anaconda3/lib/python3.1
0/site-packages (from wordcloud) (1.23.5)
Requirement already satisfied: fonttools>=4.22.0 in ./anaconda3/lib/pyth
on3.10/site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in ./anaconda3/lib/pyth
on3.10/site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/lib/python
3.10/site-packages (from matplotlib->wordcloud) (22.0)
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/lib/p
ython3.10/site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: cycler>=0.10 in ./anaconda3/lib/python3.1
0/site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in ./anaconda3/lib/pytho
n3.10/site-packages (from matplotlib->wordcloud) (1.0.5)
Requirement already satisfied: pyparsing>=2.3.1 in ./anaconda3/lib/pytho
n3.10/site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.10/si
te-packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.2
```

In [26]:

```
#NLG, Natural language generation, Text visualization
# Libraries
from wordcloud import WordCloud
import matplotlib.pyplot as plt

```

In [27]:

```
text=("I believe the documents provided adequately demonstrate my ability to cove
text
```

Out[27]:

```
'I believe the documents provided adequately demonstrate my ability to c
over all expenses and my sincere intention to comply with the immigratio
n regulations of [Destination Country]. However, if there are any additi
onal documents or information required to support my application, please
let me know at your earliest convenience. I am more than willing to prov
ide any further evidence to strengthen my case.'
```

In [28]:

```
# Create the wordcloud object
wordcloud = WordCloud(width=480, height=480, margin=0).generate(text)
```

In [29]:

```
# Display the generated image:
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
plt.show()
```

In [ ]:

```

```