In [18]:
```python
# 1. Importing the Necessar Modules

import nltk
import spacy
import numpy as np
import networkx as nx
from sklearn.metrics import jaccard_score
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import MultiLabelBinarizer
import warnings

warnings.filterwarnings("ignore")
```

In [3]:
```python
# 2. Downloading the Necessary Libraries and Modules

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab')

from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
```

In [4]:
```python
# 3. Load spaCy model for vectorization

nlp = spacy.load("en_core_web_sm")
```

In [5]:
```python
# 4. Tokenize documents into sentences

def tokenize_sentences(text):
    return sent_tokenize(text)
```

In [6]:
```python
# 5. Preprocess each sentence

def preprocess_sentence(sentence):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(sentence.lower())
    return [word for word in words if word.isalnum() and word not in stop_words]
```

In [7]:
```python
# 6. Extract key phrases using CountVectorizer

def extract_key_phrases(sentences):
    preprocessed_sentences = [' '.join(preprocess_sentence(s)) for s in sentence
    vectorizer = CountVectorizer().fit(preprocessed_sentences)
    key_phrases = vectorizer.get_feature_names_out()
    return key_phrases
```

In [8]:
```python
# 7. Jaccard Similarity Matrix between sentences and key phrases

def build_similarity_matrix(sentences, key_phrases):
    binarizer = MultiLabelBinarizer(classes=key_phrases)
```

```
    sentence_sets = [set(preprocess_sentence(s)) for s in sentences]
    binary_matrix = binarizer.fit_transform(sentence_sets)

    n = len(sentences)
    similarity_matrix = np.zeros((n, n))

    for i in range(n):
        for j in range(n):
            if i != j:
                similarity_matrix[i][j] = jaccard_score(binary_matrix[i], binary

    return similarity_matrix
```

In [9]:
```
# 8. Rank Sentences

def rank_sentences(similarity_matrix):
    graph = nx.from_numpy_array(similarity_matrix)
    scores = nx.pagerank(graph)
    return scores
```

In [10]:
```
# 9. Get summary

def textrank_summarize(text, summary_ratio=0.3):
    sentences = tokenize_sentences(text)
    key_phrases = extract_key_phrases(sentences)
    similarity_matrix = build_similarity_matrix(sentences, key_phrases)
    scores = rank_sentences(similarity_matrix)

    ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(sentences)),

    top_n = int(len(sentences) * summary_ratio)
    summary = ' '.join([sent for _, sent in ranked_sentences[:top_n]])

    return summary
```

In [11]:
```
# 10. Creating a Portfolio.

portfolio = """
I am Aryan Langhanoja
A Student of Semester 6 In Department of Information and Communication Technolog
I am Serving as a  Deputy Convenor of Competitive Programming Club.
I had done many projects in HTML CSS JS PHP Flutter React Node Express MongoDB P
I am trying to imporoving my problem solving skills by practicing DSA.
"""
```

In [19]:
```
# 11. Summary of the 50% of the size of my portfolio

print("Summary of the 50% of the size of my portfolio  :- ")
print(textrank_summarize(portfolio , 0.5))
```

```
Summary of the 50% of the size of my portfolio  :-
I had done many projects in HTML CSS JS PHP Flutter React Node Express MongoDB Po
stgreSQL etc. I am trying to imporoving my problem solving skills by practicing D
SA.
```

In [20]:
```
# 12. Summary of the 25% of the size of my portfolio

print("Summary of the 25% of the size of my portfolio  :- ")
print(textrank_summarize(portfolio , 0.25))
```

Summary of the 25% of the size of my portfolio :-
I had done many projects in HTML CSS JS PHP Flutter React Node Express MongoDB Po
stgreSQL etc.