In [9]:

```python
import nltk
from nltk.tokenize import word_tokenize
text = "The quick brown fox jumps over the lazy dog."
# Tokenize the text
tokens = word_tokenize(text)
print(tokens)
```

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog', '.']
```

In [10]:

```python
import nltk
from nltk.tag import pos_tag

# Perform POS tagging
pos_tags = pos_tag(tokens)

print(pos_tags)
```

```
[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'),
('lazy', 'JJ'), ('dog', 'NN'), ('.', '.')]
```

In [11]:

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

# Get the list of English stop words
stop_words = set(stopwords.words('english'))

# Remove stop words
filtered_tokens = [token for token in tokens if token.lower() not in stop_words]

print(filtered_tokens)
```

```
['quick', 'brown', 'fox', 'jumps', 'lazy', 'dog', '.']
```

In [12]:

```python
import nltk
from nltk.stem import SnowballStemmer
# Initialize the Snowball stemmer with English language
stemmer = SnowballStemmer('english')

# Stem the tokens
stemmed_tokens = [stemmer.stem(token) for token in filtered_tokens]

print(stemmed_tokens)
```

```
['quick', 'brown', 'fox', 'jump', 'lazi', 'dog', '.']
```

In [13]:

```python
import nltk
from nltk.stem import WordNetLemmatizer

# Initialize the WordNet lemmatizer
lemmatizer = WordNetLemmatizer()

# Lemmatize the tokens
lemmatized_tokens = [lemmatizer.lemmatize(token) for token in filtered_tokens]

print(lemmatized_tokens)
```

```
['quick', 'brown', 'fox', 'jump', 'lazy', 'dog', '.']
```

In [14]:

```python
from sklearn.feature_extraction.text import TfidfVectorizer

# Create a list of documents (in this case, we have only one document)
documents = ["The quick brown fox jumped over the lazy dog. The dog slept and the fox ran awa
```

```
# Initialize the TF-IDF vectorizer with English stop words and unigram (single word) tokens
vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(1,1))

# Fit and transform the documents to TF-IDF matrix
tfidf_matrix = vectorizer.fit_transform(documents)

# Get the feature names (i.e., the unique tokens in the documents)
feature_names = vectorizer.get_feature_names_out()

# Print the TF-IDF matrix as a pandas DataFrame
import pandas as pd
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=feature_names)
print(tfidf_df)
```

```
   away  brown  dog  fox  jumped  lazy  quick   ran  slept  woods
0  0.25   0.25  0.5  0.5    0.25  0.25   0.25  0.25   0.25   0.25
```

In [15]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Define some example documents
documents = ["The quick brown fox jumped over the lazy dog.", "The dog slept and the fox ran a

# Initialize the TF-IDF vectorizer with English stop words and unigram (single word) tokens
vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(1,1))

# Fit the vectorizer to the documents and transform them to TF-IDF matrix
tfidf_matrix = vectorizer.fit_transform(documents)

# Get the feature names (i.e., the unique tokens in the vectorizer)
feature_names = vectorizer.get_feature_names_out(documents)

# Print the feature names as a pandas DataFrame
feature_names_df = pd.DataFrame({'feature_name': feature_names})
print(feature_names_df)
```

```
  feature_name
0         away
1        brown
2          dog
3          fox
4       jumped
5         lazy
6        quick
7          ran
8        slept
9        woods
```

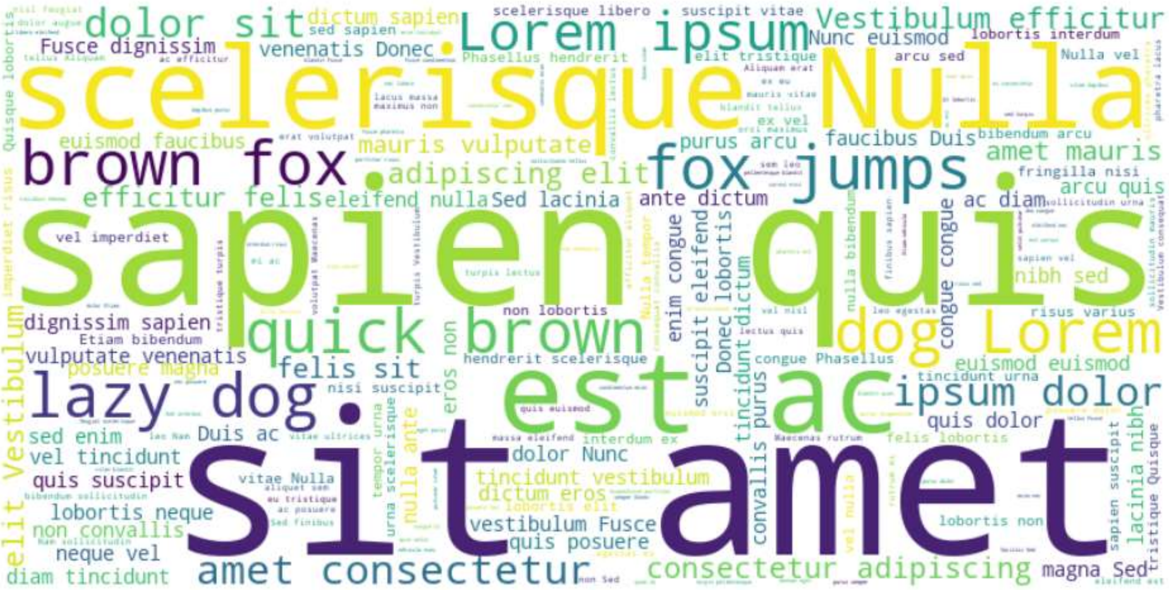
In [26]:

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Combine the tokens into a single string
text = "The quick brown fox jumps over the lazy dog. Lorem ipsum dolor sit amet, consectetur ac

# Generate the word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white', stopwords=stop_wo

# Display the word cloud using matplotlib
plt.figure(figsize=(12, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```


sit amet
sapien quis
scelerisque Nulla
est ac
brown fox
fox jumps
lazy dog
quick brown
dog Lorem
Lorem ipsum
ipsum dolor
dolor sit
amet consectetur
consectetur adipiscing
adipiscing elit
efficitur felis
mauris vulputate
euismod faucibus
Vestibulum efficitur
dictum sapien
Fusce dignissim
venenatis Donec
Nunc euismod
lobortis interdum
scelerisque libero
suscipit vitae
Phasellus hendrerit
elit tristique
arcu sed
Nulla vel
purus arcu
faucibus Duis
amet mauris
bibendum arcu
fringilla nisi
arcu quis
eleifend nulla
Sed lacinia
ante dictum
ac diam
vel imperdiet
nibh sed
dignissim sapien
Etiam bibendum
vulputate venenatis
posuere magna
felis sit
nisi suscipit
non lobortis
hendrerit scelerisque
risus varius
congue Phasellus
tincidunt urna
euismod euismod
quis dolor
sed enim
Duis ac
vel tincidunt
quis suscipit
vitae Nulla
lobortis neque
non convallis
neque vel
diam tincidunt
dolor Nunc
interdum ex
tincidunt vestibulum
dictum eros
lobortis elit
vestibulum Fusce
quis posuere
felis lobortis
lobortis non
magna Sed
elit Vestibulum
enim congue
suscipit eleifend
Donec lobortis
tincidunt dictum
convallis purus
congue congue
lacinia nibh
tristique Quisque
sapien suscipit
nulla ante
eros non
urna scelerisque
tempor urna
vel nulla
imperdiet risus
Quisque lobortis
nulla bibendum