In [2]:
```
import pandas as pd
from nltk.stem.porter import PorterStemmer
from sklearn.tree import DecisionTreeClassifier
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pickle
import re
```

In [3]:
```
# Load the dataset
df = pd.read_csv('Drug prescription Dataset.csv', delimiter=",")
```

In [3]:
```
# Handle missing values and duplicates
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
```

In [4]:
```
# Process and preprocess the text data
ps = PorterStemmer()
df['tags'] = (df['drug'] + ' ' + df['disease']).apply(lambda x: ' '.join([ps.stem(word
```

In [5]:
```
# Vectorize using CountVectorizer
cv = CountVectorizer(stop_words='english', max_features=5000)
vectors = cv.fit_transform(df['tags']).toarray()
```

In [6]:
```
# Calculate cosine similarity
similarity = cosine_similarity(vectors)
```

In [7]:
```
# Save the model
with open('ayurvedic_cosine_similarity_model.pkl', 'wb') as model_file:
    pickle.dump((cv, similarity, df), model_file)
```

File

In [8]:
```
import pickle
import re
```

In [9]:
```
# Load the saved model
with open('ayurvedic_cosine_similarity_model.pkl', 'rb') as model_file:
    cv, similarity, df = pickle.load(model_file)
```

In [10]:
```
# User input and recommendation
def recommend(input_text):
    ps = PorterStemmer()
    keywords = [ps.stem(word.lower()) for word in re.findall(r'\b\w+\b', input_text)]

    input_vector = cv.transform([" ".join(keywords)]).toarray()
    input_similarity = cosine_similarity(input_vector, vectors)

    similar_medicines = []
    for i in range(3):
        index = input_similarity.argsort()[0][-i-2]
        similar_medicines.append(df.iloc[index]['drug'])

    # Remove duplicates from the list
    unique_medicines = list(set(similar_medicines))
    return unique_medicines
```

In [27]:

```
# Get user input and provide recommendation
user_input = input("Enter a sentence: ")
similar_medicines = recommend(user_input)
print("Similar Medicines:")
for medicine in similar_medicines:
    print(medicine)
```

```
Enter a sentence: fever
Similar Medicines:
panchkol churna
phadke
ashta choornam
```

In [ ]: