In [ ]:
```
#1. Loading and Preprocessing
```

In [1]:
```
# Import necessary libraries
import pandas as pd
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Iris dataset
iris = load_iris()

# Convert to DataFrame for easier manipulation
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
print(iris_df.head())
```

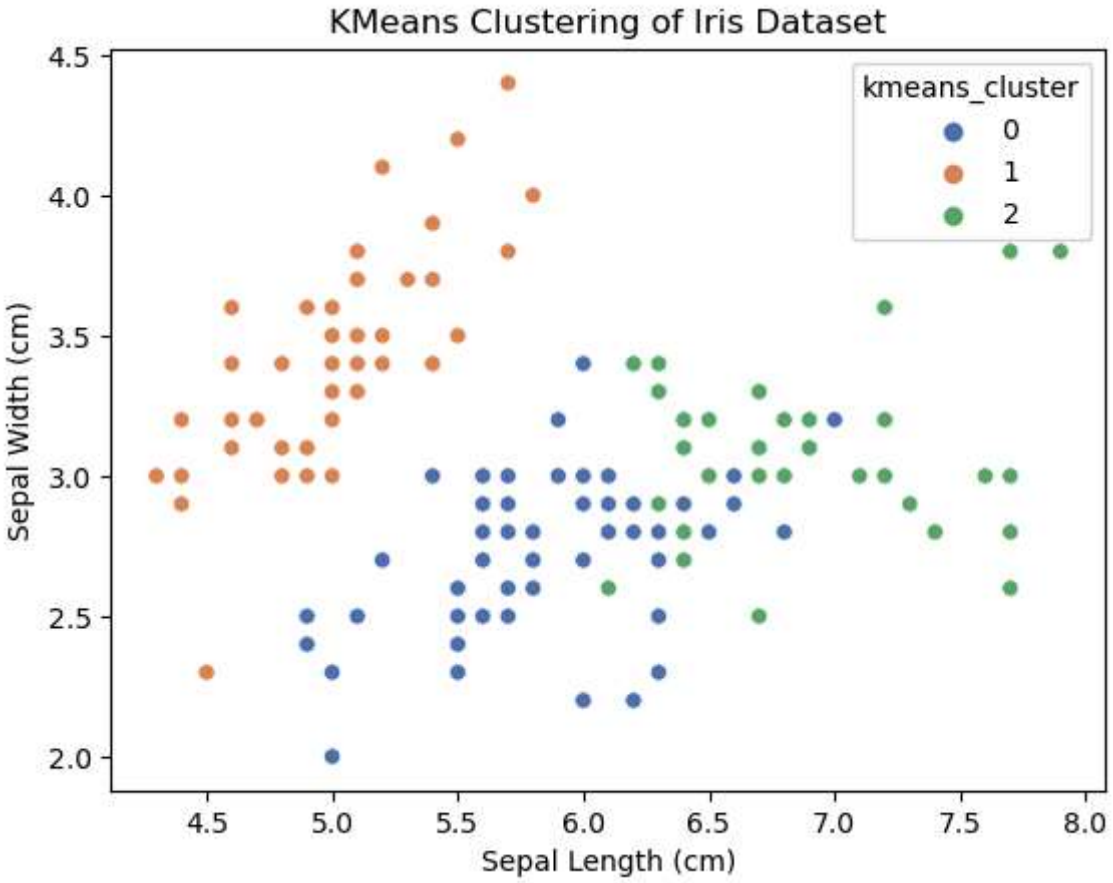
```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1               3.5                1.4               0.2
1                4.9               3.0                1.4               0.2
2                4.7               3.2                1.3               0.2
3                4.6               3.1                1.5               0.2
4                5.0               3.6                1.4               0.2
```

In [ ]:
```
#2.Clustering Algorithm Implementation
```

In [ ]:
```
#A) KMeans Clustering (4 marks)
KMeans is an iterative clustering algorithm that partitions a dataset into K d
```

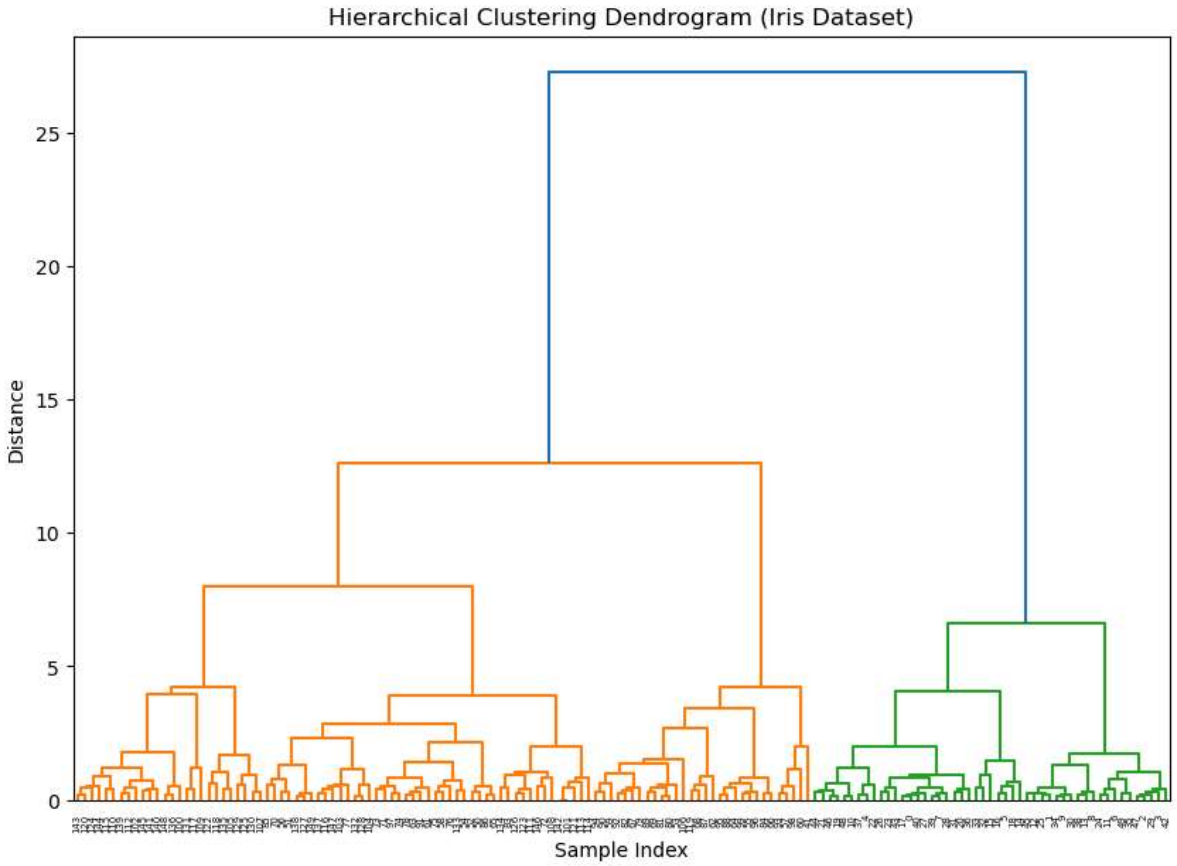
In [2]:
```
from sklearn.cluster import KMeans

# Initialize the KMeans algorithm
kmeans = KMeans(n_clusters=3, random_state=42)

# Fit the model and predict the clusters
iris_df['kmeans_cluster'] = kmeans.fit_predict(iris_df)

# Visualizing the clusters
sns.scatterplot(x=iris_df.iloc[:, 0], y=iris_df.iloc[:, 1], hue=iris_df['kmean
plt.title("KMeans Clustering of Iris Dataset")
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Sepal Width (cm)")
plt.show()
```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: Fu
tureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



In [ ]:
```
#B) Hierarchical Clustering (4 mar
Hierarchical clustering is a method of cluster analysis that seeks to build a

Agglomerative (bottom-up) clustering starts with each point as its own cluster
Divisive (top-down) clustering starts with all data points in one cluster and
```

In [3]:
```
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.preprocessing import StandardScaler

# Standardize the dataset
scaler = StandardScaler()
iris_scaled = scaler.fit_transform(iris_df.drop('kmeans_cluster', axis=1))  #

# Apply hierarchical clustering
linked = linkage(iris_scaled, method='ward')

# Plot the dendrogram
plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_co
plt.title("Hierarchical Clustering Dendrogram (Iris Dataset)")
plt.xlabel("Sample Index")
plt.ylabel("Distance")
plt.show()
```



In [ ]: