In [20]:
```
!pip install scikit-learn pandas
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.1.4)
Requirement already satisfied: numpy<2.0,>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.26.4)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [22]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.decomposition import PCA
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import seaborn as sns
```

In [23]:
```
# Step 1: Load the Dataset
data = pd.read_csv('Student_Performance.csv')
```

In [24]:
```
# Explore the dataset
print("First 5 rows of the dataset:")
print(data.head())
print("\nDataset summary:")
print(data.describe(include='all'))
print("\nData types:")
print(data.dtypes)
```

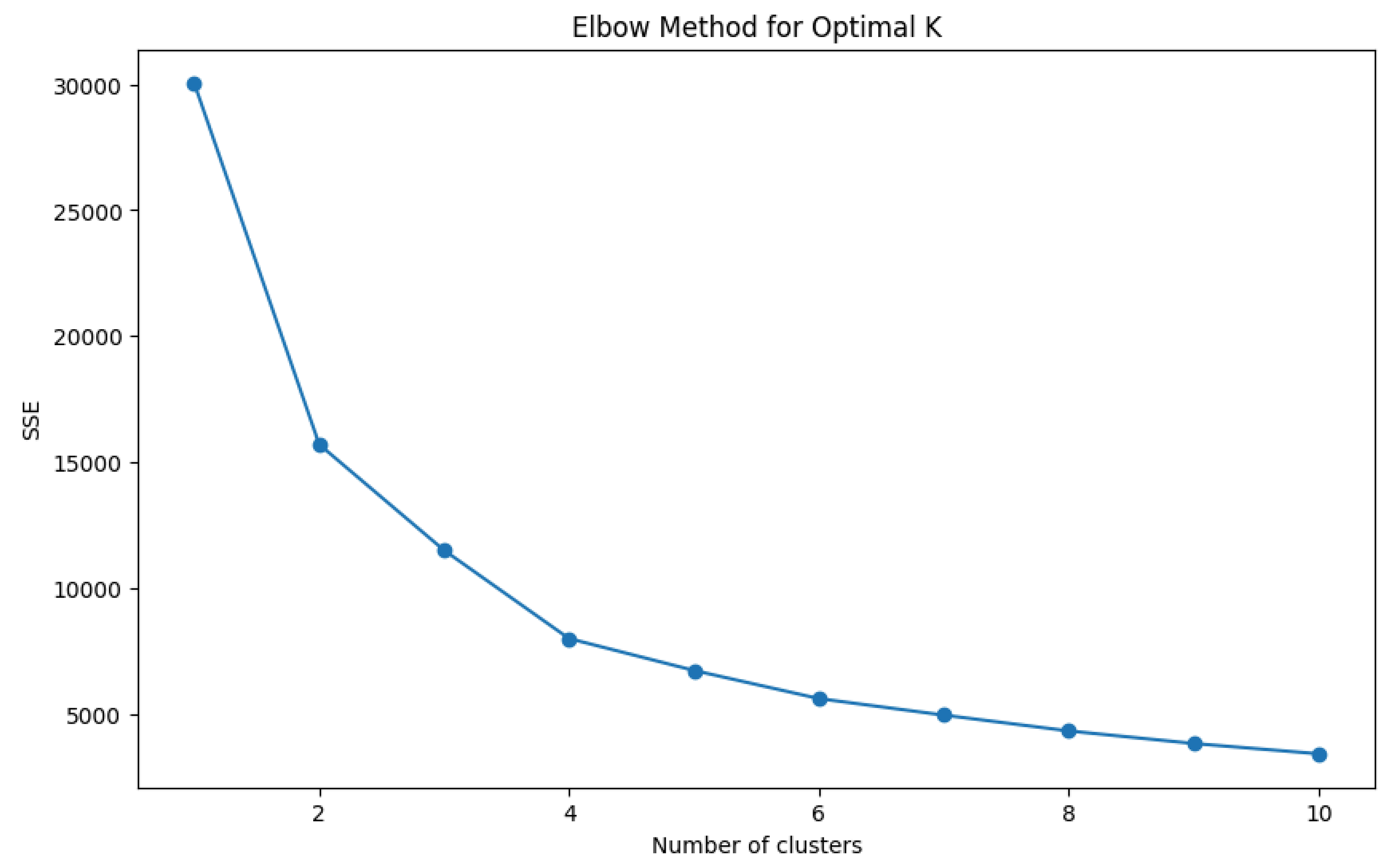
```
First 5 rows of the dataset:
   Hours Studied  Previous Scores Extracurricular Activities  Sleep Hours  \
0              7               99                        Yes            9
1              4               82                         No            4
2              8               51                        Yes            7
3              5               52                        Yes            5
4              7               75                         No            8

   Sample Question Papers Practiced  Performance Index
0                                 1               91.0
1                                 2               65.0
2                                 2               45.0
3                                 2               36.0
4                                 5               66.0

Dataset summary:
        Hours Studied  Previous Scores Extracurricular Activities  \
count    10000.000000     10000.000000                      10000
unique            NaN              NaN                          2
top               NaN              NaN                         No
freq              NaN              NaN                       5052
mean         4.992900        69.445700                        NaN
std          2.589309        17.343152                        NaN
min          1.000000        40.000000                        NaN
25%          3.000000        54.000000                        NaN
50%          5.000000        69.000000                        NaN
75%          7.000000        85.000000                        NaN
max          9.000000        99.000000                        NaN

         Sleep Hours  Sample Question Papers Practiced  Performance Index
count   10000.000000                      10000.000000       10000.000000
unique           NaN                               NaN                NaN
top              NaN                               NaN                NaN
freq             NaN                               NaN                NaN
mean        6.530600                          4.583300          55.224800
std         1.695863                          2.867348          19.212558
min         4.000000                          0.000000          10.000000
25%         5.000000                          2.000000          40.000000
50%         7.000000                          5.000000          55.000000
75%         8.000000                          7.000000          71.000000
max         9.000000                          9.000000         100.000000

Data types:
Hours Studied                         int64
Previous Scores                       int64
Extracurricular Activities           object
Sleep Hours                           int64
Sample Question Papers Practiced      int64
Performance Index                   float64
dtype: object
```

In [25]:
```
# Step 2: Preprocess the Data
# Identify numeric and categorical columns
numeric_features = data.select_dtypes(include=['int64', 'float64']).columns
categorical_features = data.select_dtypes(include=['object', 'bool']).columns

# Create preprocessing pipelines for numeric and categorical data
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))])

# Combine preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)])

# Apply preprocessing
preprocessed_data = preprocessor.fit_transform(data)
```

In [26]:
```
# Step 3: Reduce Dimensionality
# Reduce dimensionality using PCA
pca = PCA(n_components=2)  # Adjust the number of components as needed
reduced_data = pca.fit_transform(preprocessed_data)
```

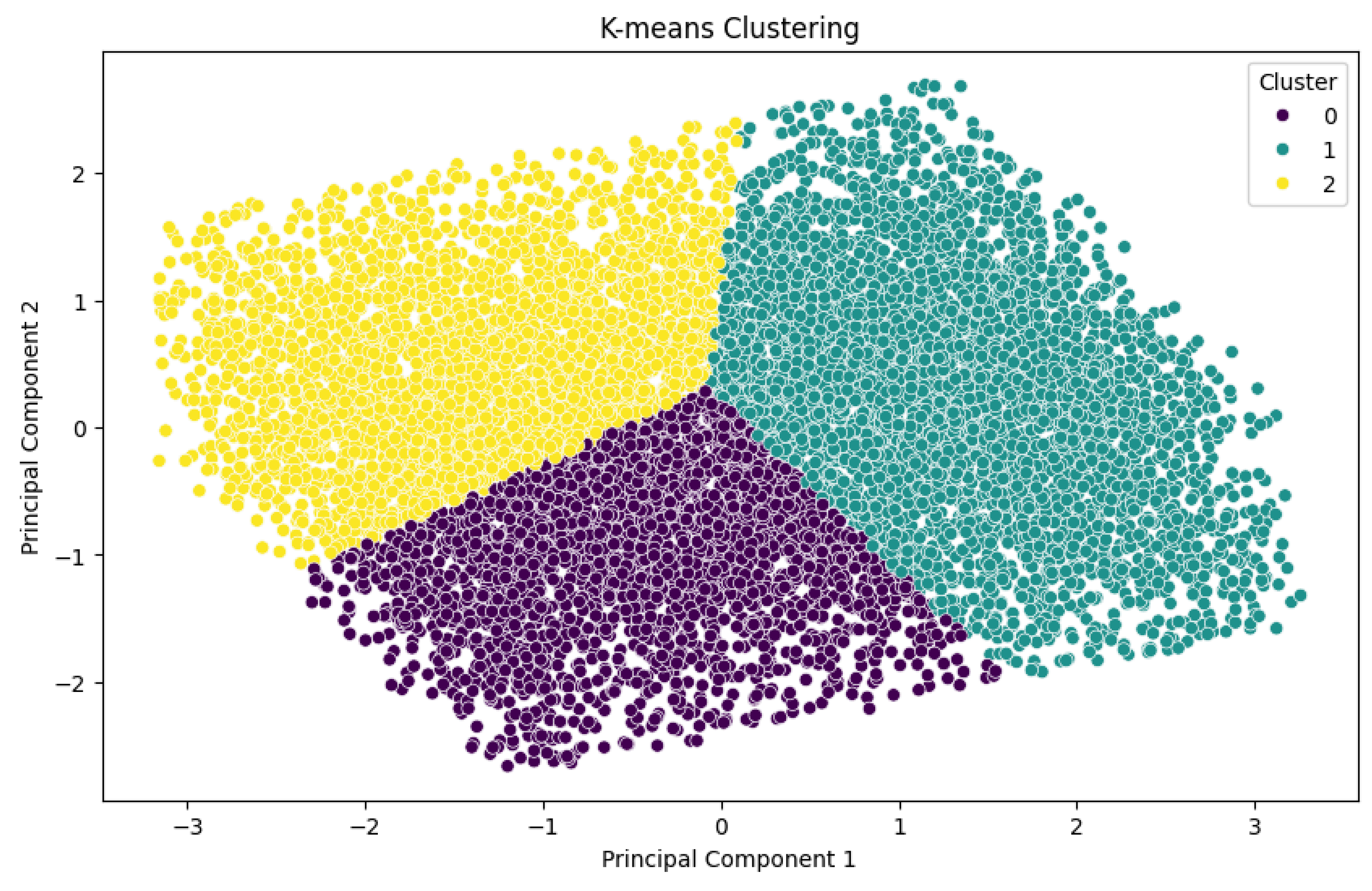
In [27]:
```
# Step 4: Apply K-means Clustering
# Determine the optimal number of clusters using the elbow method
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(reduced_data)
    sse.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), sse, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.show()

# Based on the elbow plot, choose the optimal number of clusters (e.g., 3)
optimal_k = 3

# Apply K-means with the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
clusters = kmeans.fit_predict(reduced_data)

# Add the cluster labels to the original dataset
data['Cluster'] = clusters

# Save the dataset with cluster labels
data.to_csv('clustered_dataset.csv', index=False)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
```




```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: 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
  super()._check_params_vs_input(X, default_n_init=10)
```

In [28]:
```
# Step 5: Visualize the Clusters
plt.figure(figsize=(10, 6))
sns.scatterplot(x=reduced_data[:, 0], y=reduced_data[:, 1], hue=clusters, palette='viridis')
plt.title('K-means Clustering')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(title='Cluster')
plt.show()
```



In [29]:
```
# Step 6: Evaluate the Clustering
# Calculate the silhouette score
silhouette_avg = silhouette_score(reduced_data, clusters)
print(f"Silhouette Score: {silhouette_avg}")

# Print the cluster centers in the reduced dimensionality space
print("Cluster Centers (in PCA-reduced space):")
print(kmeans.cluster_centers_)
```

```
Silhouette Score: 0.3680095774845573
Cluster Centers (in PCA-reduced space):
[[-0.46393051 -1.04208224]
 [ 1.32797097  0.29331202]
 [-1.44287067  0.53262019]]
```

In [29]: