Streamlit\Mall_Customer_Segmentation_app.py

```python
import streamlit as st
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import pickle

# Load pre-trained KMeans model

filename = 'C:\\Users\Jan Saida\Mall_prediction.pkl'
with open(filename, 'rb') as file:
    kmeans = pickle.load(file)

# Function to load dataset and show initial preview
def load_data():
    dataset = pd.read_csv(r"C:\Users\Jan Saida\OneDrive\Documents\Desktop\Excel sheets\Mall_Customers.csv")
    return dataset

# Title for Streamlit app
st.title("Mall Customer Segmentation Using K-Means Clustering")

# Dataset display
st.header("Dataset Preview")
dataset = load_data()
st.dataframe(dataset.head())

# Allow user to input custom values for clustering
st.sidebar.header("Input Parameters for Clustering")

# Taking input for annual income and spending score
annual_income = st.sidebar.slider("Annual Income (k$)", float(dataset['Annual Income (k$)'].min()), float(dataset['Annual Income (k$)'].max()))
spending_score = st.sidebar.slider("Spending Score (1-100)", 1, 100)

# Display the user's input
st.write(f"User Input - Annual Income: {annual_income}k$ | Spending Score: {spending_score}")

# Predict the cluster based on the user input
user_input = np.array([[annual_income, spending_score]])
cluster_prediction = kmeans.predict(user_input)
```

```python

st.write(f"The customer is predicted to belong to Cluster {cluster_prediction[0] + 1}")

# Visualize the clusters with the user's input
st.header("Cluster Visualization")

# Get the coordinates of the clusters
x = dataset.iloc[:, [3, 4]].values
y_kmeans = kmeans.predict(x)

# Plotting the clusters
plt.figure(figsize=(10, 6))
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(x[y_kmeans == 3, 0], x[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(x[y_kmeans == 4, 0], x[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')

# Plotting the centroids
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')

# Labels and title
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
st.pyplot(plt)

# Running the app
if __name__ == '__main__':
    st.write("This is a simple app that predicts which customer cluster a person falls into based on their annual income and spending score.")

```