6. Implement program to apply moving average smoothing for data preparation and time series forecasting.

EX.N0 : 6	**Implement program to apply moving average smoothing for data preparation and time series forecasting.**
DATE : 05/04/2025	

AIM:

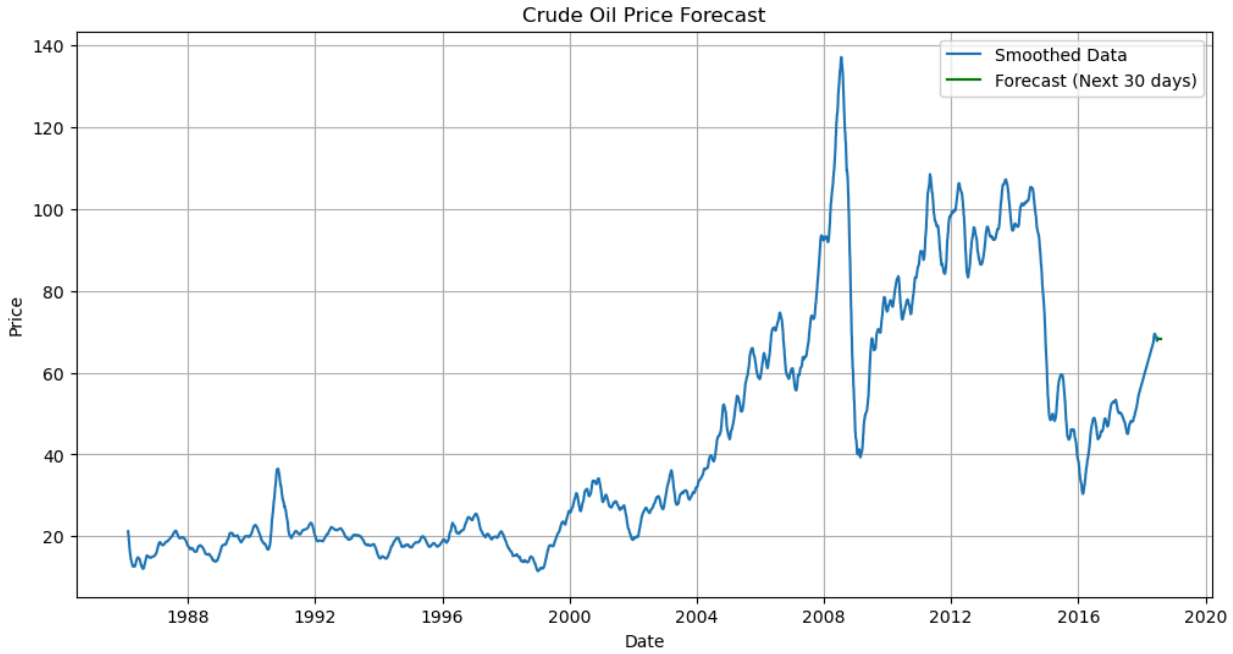
To Implement program to apply moving average smoothing for data preparation and time series forecasting.

PROGRAM:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from statsmodels.tsa.holtwinters import SimpleExpSmoothing

file_path = r"C:\Users\heman\OneDrive\Desktop\abdul\TSA\EX 5\Crude Oil Prices Daily.xlsx"
(1)\FINAL_USO.csv" # Update path
df = pd.read_csv(file_path, parse_dates=["Date"], index_col="Date")

df.columns = df.columns.str.strip()

if "Adj Close" not in df.columns:
 print("Error: 'Adj Close' column not found!")
 print("Available columns:", df.columns)
 exit()

target = "Adj Close"

df["SMA_10"] = df[target].rolling(window=10).mean() # 10-day SMA
df["SMA_20"] = df[target].rolling(window=20).mean() # 20-day SMA

df.dropna(inplace=True) # Remove NaN values from SMA columns

X = df[["SMA_10", "SMA_20"]]
y = df[target]

train_size = int(len(df) * 0.8) # 80% train, 20% test
X_train, X_test = X.iloc[:train_size], X.iloc[train_size:]
y_train, y_test = y.iloc[:train_size], y.iloc[train_size:]

model = SimpleExpSmoothing(y_train).fit(smoothing_level=0.2,
optimized=False) forecast = model.forecast(len(y_test)) # Predict the same length
as test set
```

```
forecast.index = y_test.index

plt.figure(figsize=(12, 5))
plt.plot(y.index, y, label="Actual Price", color="blue", alpha=0.5)
plt.plot(y_test.index, forecast, label="Forecast", color="red", linestyle="dashed")
plt.title("Gold Price Forecast using Moving Average & Exponential Smoothing")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.show()
```

OUTPUT:



RESULT:

Thus, the program for Implement program to apply moving average smoothing for data preparation and time series forecasting is executed successfully.