Name: Daniel Mehta

IDNo.: n01753264

Lab 4: Time Series Forecasting

Use the provided data EnergyProduction.csv to answer all the questions in this Notebook

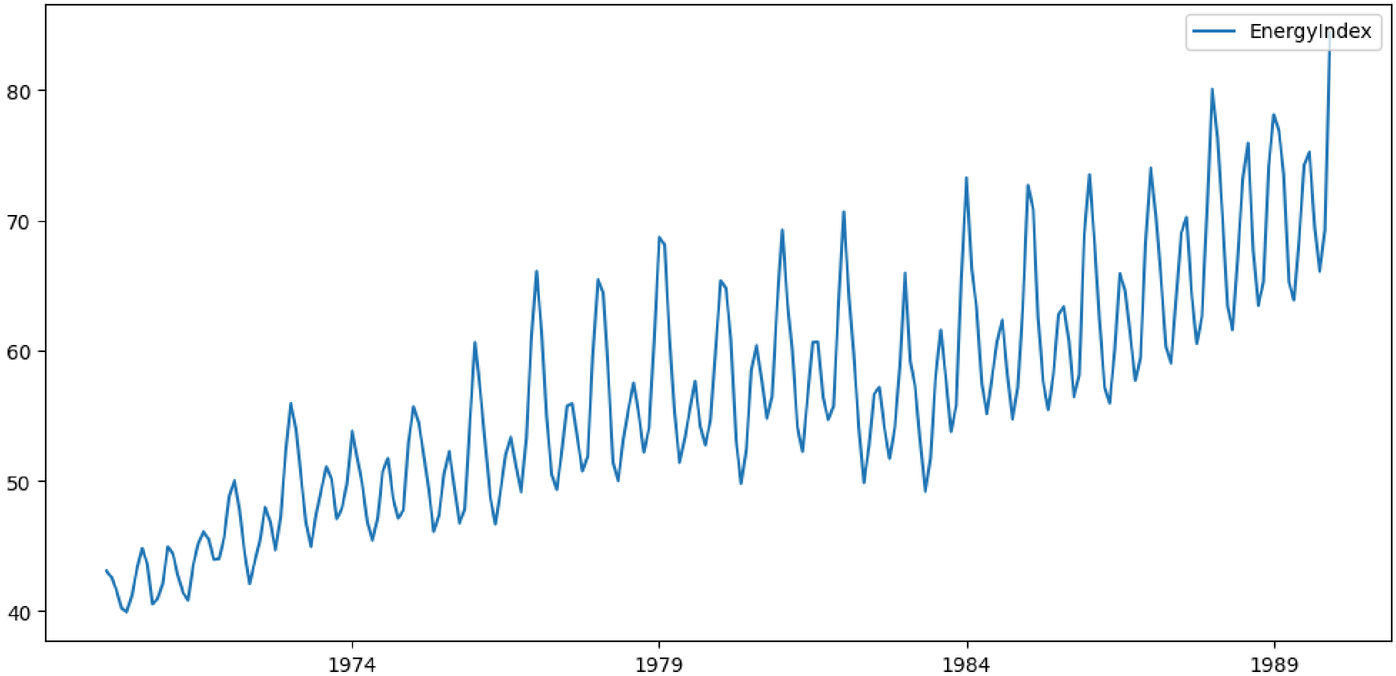
In [15]:
```
#import the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.holtwinters import ExponentialSmoothing
```

In [2]:
```
#import the your data here with date as index and properly formatted data type as below:
df = pd.read_csv("EnergyProduction.csv")
df.head()
```

Out[2]:

	DATE	EnergyIndex
0	1970-01-01	43.0869
1	1970-02-01	42.5577
2	1970-03-01	41.6215
3	1970-04-01	40.1982
4	1970-05-01	39.9321

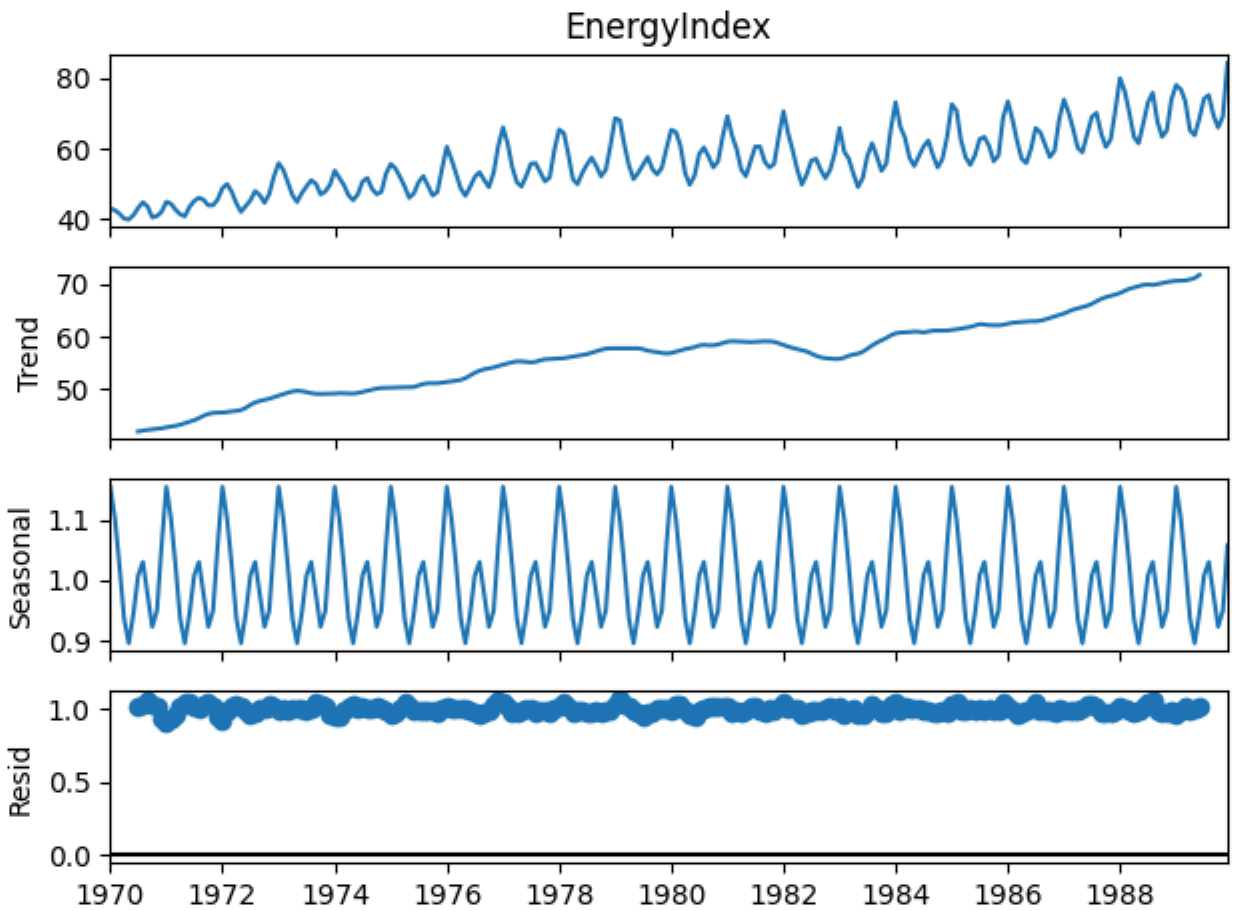
In [5]:
```
# plot the below plot using the dataset
df["DATE"] = pd.to_datetime(df["DATE"])

plt.figure(figsize=(10, 5))
plt.plot(df["DATE"],df["EnergyIndex"], label="EnergyIndex")

tick_years = pd.date_range(start="1974", end="1990", freq="5YS")
plt.xticks(tick_years)
ax = plt.gca()
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))

plt.legend(loc="upper right")
plt.tight_layout()
plt.show()
```



In [7]:
```
# Assign a frequency of 'MS' to the DatetimeIndex as below
df.set_index("DATE", inplace=True)

df = df.asfreq('MS')
print(df.index)
print(df.index.freq)
```

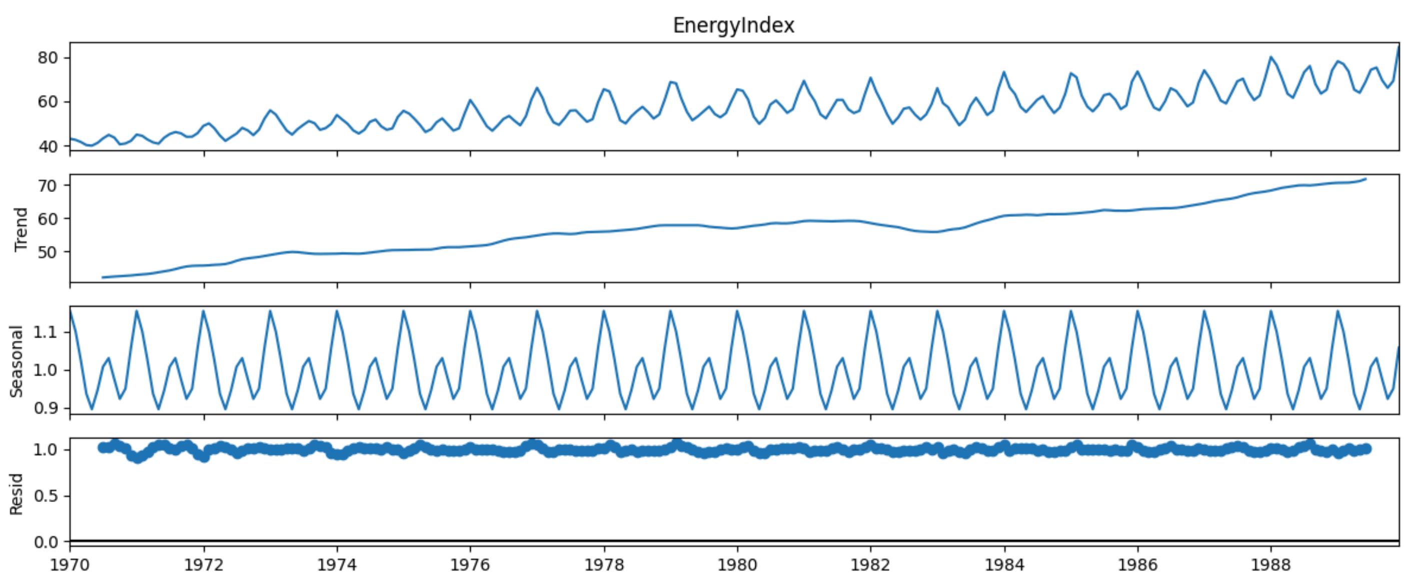
```
DatetimeIndex(['1970-01-01', '1970-02-01', '1970-03-01', '1970-04-01',
               '1970-05-01', '1970-06-01', '1970-07-01', '1970-08-01',
               '1970-09-01', '1970-10-01',
               ...
               '1989-03-01', '1989-04-01', '1989-05-01', '1989-06-01',
               '1989-07-01', '1989-08-01', '1989-09-01', '1989-10-01',
               '1989-11-01', '1989-12-01'],
              dtype='datetime64[ns]', name='DATE', length=240, freq='MS')
<MonthBegin>
```

In [9]:
```
#  Decompose Trend, Cyclic and Error as shown below
result = seasonal_decompose(df["EnergyIndex"], model="multiplicative")
result.plot()
plt.tight_layout()
plt.show()
```



4. Change the size of the figure to be more clear.

In [11]:
```
from pylab import rcParams
rcParams["figure.figsize"]= 12,5
result.plot();
```



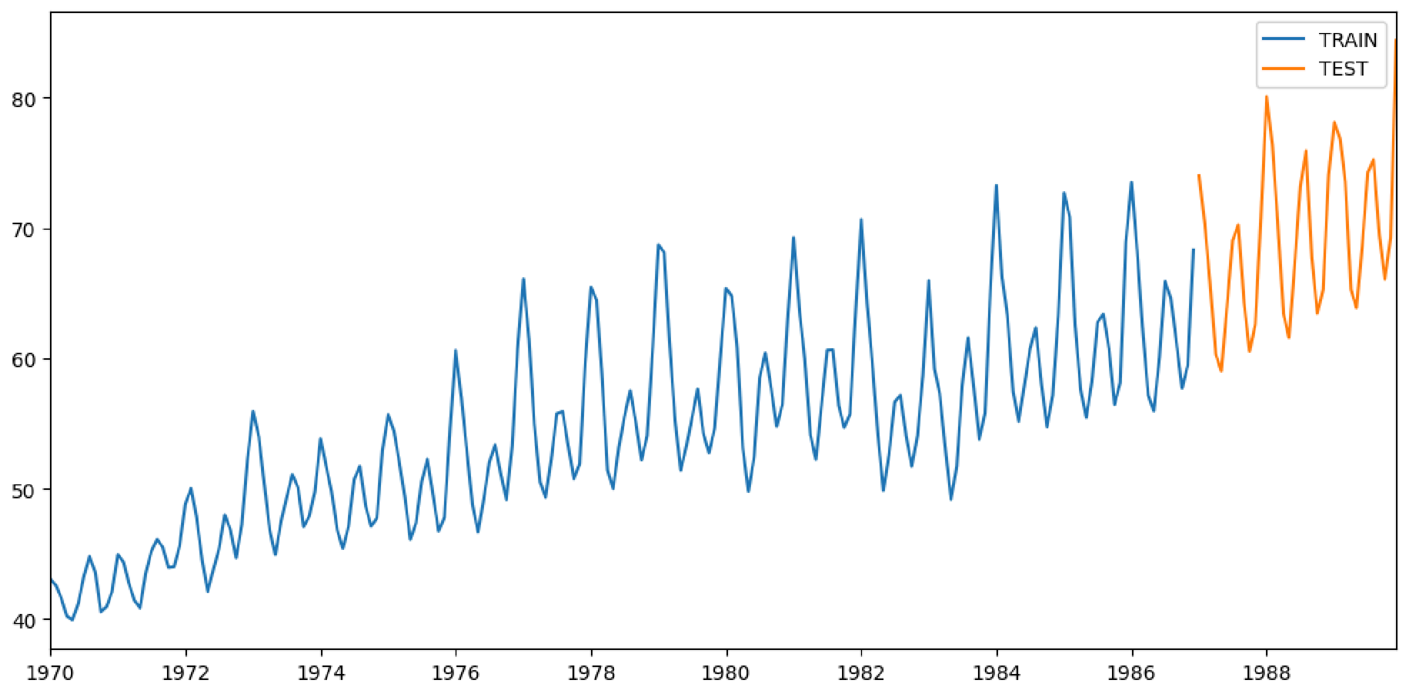
5. Apply Forcasting on Energy Index

In [13]:
```
# Apply Forcasting on Energy Index using training data and testing data

train = df.iloc[:-36]   #All exept the last 36 months
test = df.iloc[-36:]    # Last 36 months
```

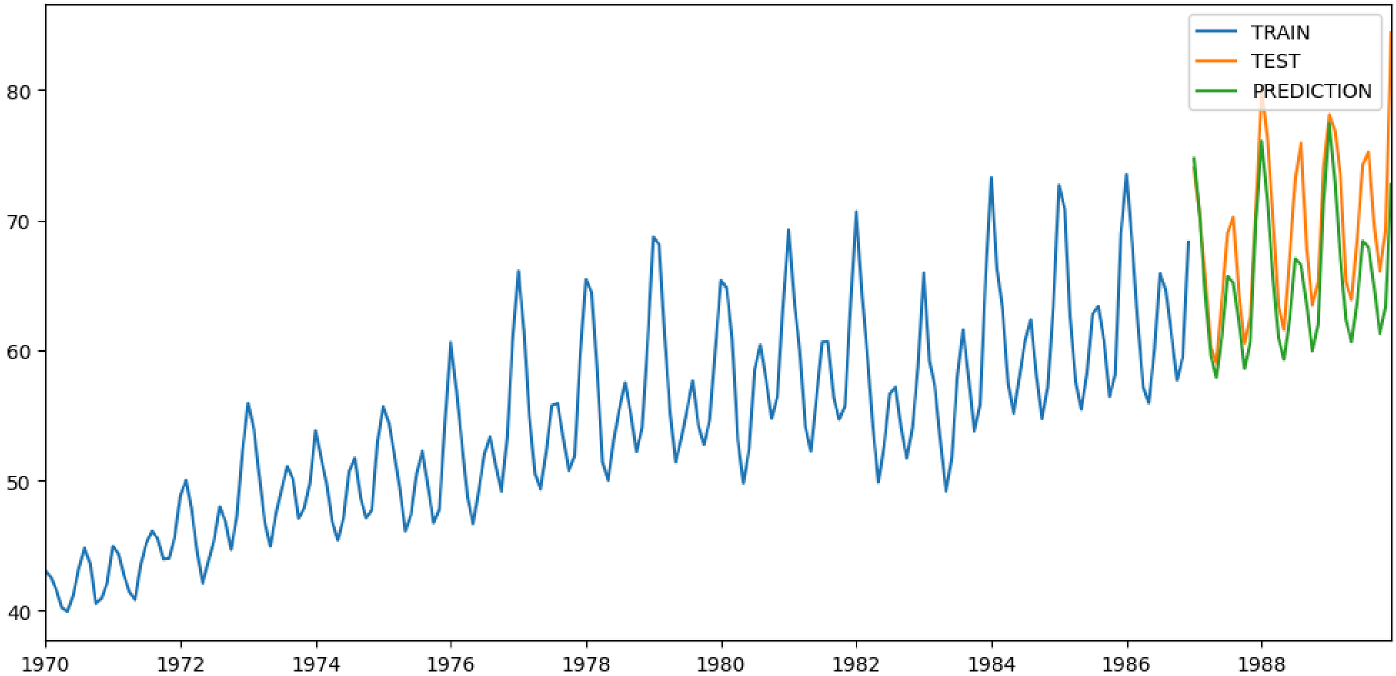
In [17]:
```
# fit the training model using exponentialSmoothing within a period of 12 months
hw_model = ExponentialSmoothing(
    train["EnergyIndex"],
    trend="add",
    seasonal="add",
    seasonal_periods=12
).fit()
```

In [19]:
```
# fit the testing data to 36 months period and rename it to "HW Forecast"
forecast = hw_model.forecast(steps=36)
forecast = forecast.rename("HW Forecast")
```

In [39]:
```
# produce the below plot as shown
start_date = train.index.min()
end_date = test.index.max()

plt.figure(figsize=(10, 5))
plt.plot(train.index, train["EnergyIndex"], label="TRAIN")
plt.plot(test.index, test["EnergyIndex"], label="TEST")

plt.legend(loc="upper right")
plt.xlim([start_date, end_date])
plt.tight_layout()
plt.show()
```

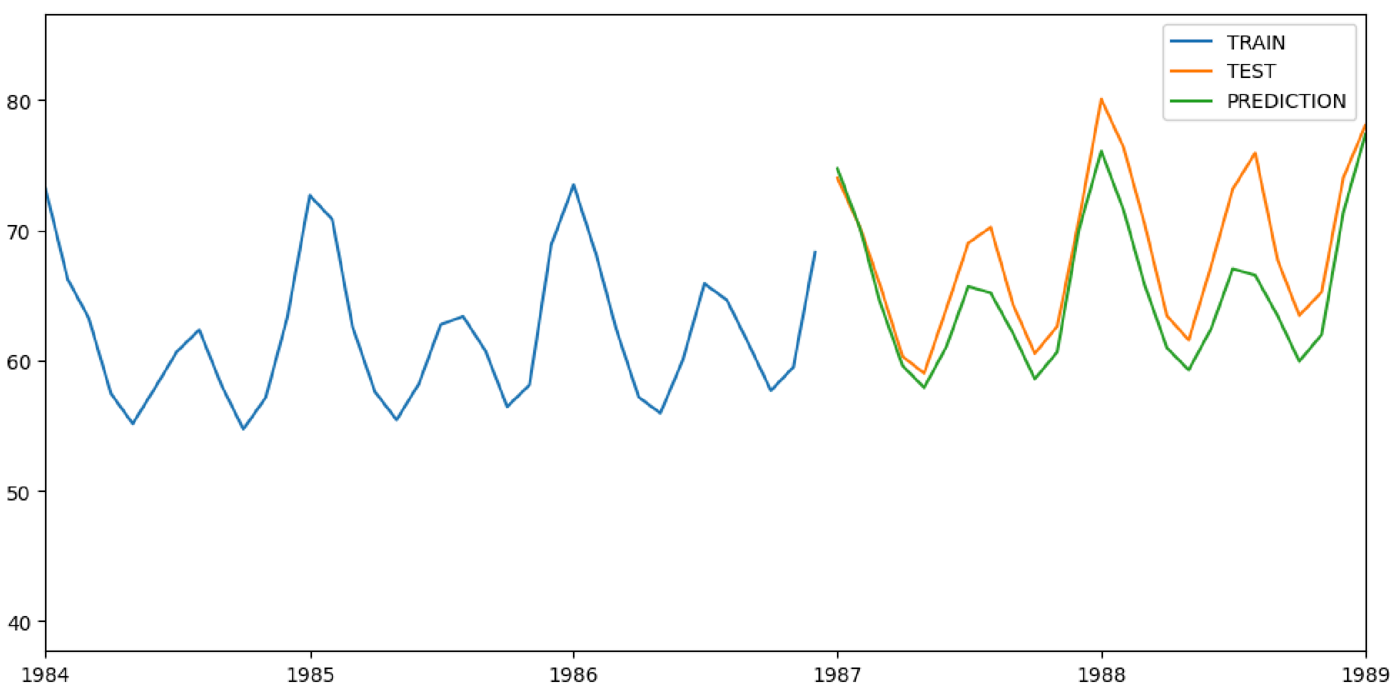


In [51]:
```
# produce the below plot as shown


plt.figure(figsize=(10, 5))
plt.plot(train.index, train["EnergyIndex"], label="TRAIN")
plt.plot(test.index, test["EnergyIndex"], label="TEST")
plt.plot(forecast.index, forecast, label="PREDICTION")

plt.xticks(rotation=0)
plt.legend(loc="upper right")
plt.xlim([start_date, end_date])
plt.tight_layout()
```



In [57]:

```
# produce the below plot as shown with specific period of between 1984-01-01 and 1989-01

plt.figure(figsize=(10, 5))
plt.plot(train.index, train["EnergyIndex"], label="TRAIN")
plt.plot(test.index, test["EnergyIndex"], label="TEST")
plt.plot(forecast.index, forecast, label="PREDICTION")

plt.xlim(pd.to_datetime("1984-01-01"), pd.to_datetime("1989-01-01"))
plt.legend()
plt.tight_layout()
plt.show()
```



Give your conclusion here

- The forecast captures the overall upward trend and repeating seasonal patterns in the Energy Index
- The prediction aligns closely with test data for most of the 36-month period
- Overall, the model performs well for short- to medium-term energy forecasting

In [ ]: