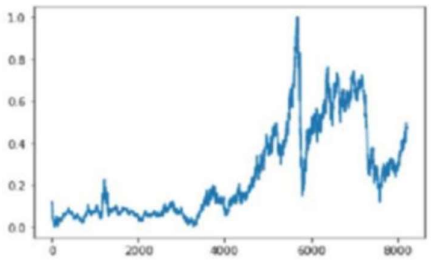
MODEL BUILDING- SAVE THE MODEL

Team ID	PNT2022TMID47669
Project Name	Crude Oil Price Prediction

```
In [3]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt

In [4]: data=pd.read_excel("/content/Crude Oil Prices Daily.xlsx")

In [5]: data.isnull().any()

Out[5]: Date            False
        Closing Value    True
        dtype: bool

In [6]: data.isnull().sum()

Out[6]: Date            0
        Closing Value   7
        dtype: int64

In [7]: data.dropna(axis=0,inplace=True)

In [8]: data.isnull().sum()

Out[8]: Date            0
        Closing Value   0
        dtype: int64

In [9]: data_oil=data.reset_index()['Closing Value']
        data_oil

Out[9]: 0       25.56
        1       26.00
        2       26.53
        3       25.85
        ...
        8211    73.89
        8212    74.19
        8213    73.05
        8214    73.78
        8215    73.93
        Name: Closing Value, Length: 8216, dtype: float64

In [10]: from sklearn.preprocessing import MinMaxScaler
         scaler=MinMaxScaler(feature_range=(0,1))
         data_oil=scaler.fit_transform(np.array(data_oil).reshape(-1,1))

In [11]: data_oil

Out[11]: array([[0.11115781],
                [0.11661464],
                [0.12053802],
                ...,
                [0.46497853],
                [0.47038353],
                [0.47149415]])

In [12]: plt.plot(data_oil)
```

Out[12]:



In [13]:
```
training_size=int(len(data_oil)*0.65)
test_size=len(data_oil)-training_size
train_data,test_data=data_oil[0:training_size,:],data_oil[training_size:len(data_oil),:1]
```

In [14]:
```
training_size,test_size
```

Out[14]: (5340, 2879)

Out[15]: (5340, 1)

In [16]:
```
def create_dataset(dataset,time_step=1):
  dataX,dataY=[],[]
  for i in range(len(dataset)-time_step-1):
    a=dataset[i:(i+time_step),0]
    dataX.append(a)
    dataY.append(dataset[i+time_step,0])
  return np.array(dataX),np.array(dataY)
```

In [17]:
```
time_step=10
x_train,y_train=create_dataset(train_data,time_step)
x_test,y_test=create_dataset(test_data,time_step)
```

In [18]:
```
print(x_train.shape),print(y_train.shape)
```

```
(5329, 10)
(5329,)
```

Out[18]: (None, None)

In [19]:
```
print(x_test.shape),print(y_test.shape)
```

```
(2865, 10)
(2865,)
```

Out[19]: (None, None)

In [20]:
```
x_train
```

Out[20]:
```
array([[0.11333703, 0.11661484, 0.12053902, ..., 0.10880805, 0.1069886 ,
        0.11054846],
       [0.11661484, 0.12053902, 0.11558422, ..., 0.1069886 , 0.11054846,
        0.10163852],
       [0.12053902, 0.11558422, 0.1156528 , ..., 0.11054846, 0.10163852,
        0.09906788],
       ...,
       [0.36731823, 0.35176058, 0.36080261, ..., 0.36391234, 0.37042796,
        0.37042796],
       [0.35176058, 0.36080261, 0.35154657, ..., 0.37042796, 0.37042796,
        0.37879461],
       [0.36080261, 0.35154657, 0.35295624, ..., 0.37042796, 0.37879461,
        0.37910482]])
```

In [21]:
```
x_train=x_train.reshape(x_train.shape[0],x_train.shape[1],1)
x_test=x_test.reshape(x_test.shape[0],x_test.shape[1],1)
```

In [22]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

In [23]:
```
model=Sequential()
```

In [24]:
```
model.add(LSTM(50,return_sequences=True,input_shape=(10,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
```

In [25]:
```
model.add(Dense(1))
```

:!" ' ' ' nudel•P1tIa._Lt'a1it, y_t ‹•a1 'l. va1*de u1uft_J«can(n_Le>t.y_Le¥L7, oDu«t»e¥.tz< L<h_»1<w•b•, vai•tzu >a•a)

```
84/84 [==============================] - 3s 32ms/step - loss: 1.2320e-04 - val_loss: 7.7520e-04
```

```
### Calculate RMSE performance metrics
```

```
In [30]: from tensorflow.keras.models import load_model
```

```
In [31]: model.save("crude oil.hs")
```

```
In [ ]:
```