```
#Import libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

!pip install matplotlib
%matplotlib inline

dataset_train=pd.read_csv(r"C:\Users\Dell\Desktop\Internships\WiDs\Google_test_data.csv")
dataset_train
dataset_train.head()

training_set= dataset_train.iloc[:,1:2].values

print(training_set)
print(training_set.shape)

from sklearn.preprocessing import MinMaxScaler

scaler=MinMaxScaler(feature_range =(0,1))
scaled_training_set = scaler.fit_transform(training_set)

scaled_training_set

X_train = []
y_train = []
for i in range(60,252):
    X_train.append(scaled_training_set[i-60:i,0])
    y_train.append(scaled_training_set[i,0])
X_train = np.array(X_train)
y_train = np.array(y_train)

print(X_train.shape)
print(y_train.shape)

X_train = np.reshape(X_train,(X_train.shape[0], X_train.shape[1], 1))

X_train.shape

!pip install jupyterlab
!pip install keras
get_ipython().system('pip install keras')
from keras.models import Sequential
```

```
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Dropout

!pip install regressor



regressor = Sequential()

regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1],1) ))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

regressor.add(Dense(units=1))

!pip install regressor
regressor.compile(optimizer = 'adam',  loss = 'mean_squared_error')
regressor.fit(X_train, y_train, epochs=100,batch_size=32)

dataset_test = pd.read_csv(r"C:\Users\Dell\Desktop\Internships\WiDs\Google_test_data.csv")
actual_stock_price = dataset_test.iloc[:,1:2].values

dataset_total = pd.concat((dataset_train['Open'],dataset_test['Open']), axis=0)
inputs = dataset_total[len(dataset_total)-len(dataset_test)-60:].values

inputs = inputs.reshape(-1,1)
inputs = scaler.transform(inputs)

X_test = []
for i in range (60,80):
    X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
```

```
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = scaler.inverse_transform(predicted_stock_price)

plt.plot(actual_stock_price, color = 'red', label= 'Actual Google Stock Price')
plt.plot(predicted_stock_price,color='blue', label='Predicted  Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
```