In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the

#import os
#print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.
```

In [2]:
```
dataset_train = pd.read_csv("trainset.csv")
```

In [3]:
```
dataset_train
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2013-01-02	357.385559	361.151062	355.959839	359.288177	359.288177	5115500
1	2013-01-03	360.122742	363.600128	358.031342	359.496826	359.496826	4666500
2	2013-01-04	362.313507	368.339294	361.488861	366.600616	366.600616	5562800
3	2013-01-07	365.348755	367.301056	362.929504	365.001007	365.001007	3332900
4	2013-01-08	365.393463	365.771027	359.874359	364.280701	364.280701	3373900
...	...	...	...	...	...	...	...
1254	2017-12-22	1061.109985	1064.199951	1059.439941	1060.119995	1060.119995	755100
1255	2017-12-26	1058.069946	1060.119995	1050.199951	1056.739990	1056.739990	760600
1256	2017-12-27	1057.390015	1058.369995	1048.050049	1049.369995	1049.369995	1271900
1257	2017-12-28	1051.599976	1054.750000	1044.770020	1048.140015	1048.140015	837100
1258	2017-12-29	1046.719971	1049.699951	1044.900024	1046.400024	1046.400024	887500

1259 rows × 7 columns

In [4]:
```
trainset = dataset_train.iloc[:,1:2].values
```

In [5]:
```
trainset
```

Out[5]:
```
array([[ 357.385559],
       [ 360.122742],
       [ 362.313507],
       ...,
       [1057.390015],
       [1051.599976],
       [1046.719971]])
```

In [6]:
```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0,1))
training_scaled = sc.fit_transform(trainset)
```

In [7]:
```
training_scaled
```

Out[7]:
```
array([[0.01011148],
       [0.01388614],
       [0.01690727],
       ...,
       [0.97543954],
       [0.9674549 ],
       [0.96072522]])
```

In [8]:
```
x_train = []
y_train = []
```

In [9]:
```
for i in range(60,1259):
    x_train.append(training_scaled[i-60:i, 0])
    y_train.append(training_scaled[i,0])
x_train,y_train = np.array(x_train),np.array(y_train)
```

In [10]:
```
x_train.shape
```

Out[10]:
```
(1199, 60)
```

In [11]:
```
x_train = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))
```

In [12]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
```

In [13]:
```
regressor = Sequential()
regressor.add(LSTM(units = 50,return_sequences = True,input_shape = (x_train.shape
```

In [14]:
```
regressor.add(Dropout(0.2))
```

In [15]:
```
regressor.add(LSTM(units = 50,return_sequences = True))
regressor.add(Dropout(0.2))
```

In [16]:
```
regressor.add(LSTM(units = 50,return_sequences = True))
regressor.add(Dropout(0.2))
```

In [17]:
```
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
```

In [18]:
```
regressor.add(Dense(units = 1))
```

In [19]:
```
regressor.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

In [20]:
```
regressor.fit(x_train,y_train,epochs = 100, batch_size = 32)
```

```
Epoch 1/100
38/38 [==============================] - 13s 114ms/step - loss: 0.0249
Epoch 2/100
38/38 [==============================] - 4s 117ms/step - loss: 0.0045
Epoch 3/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0043
Epoch 4/100
38/38 [==============================] - 4s 112ms/step - loss: 0.0040
Epoch 5/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0038
Epoch 6/100
38/38 [==============================] - 4s 112ms/step - loss: 0.0036
Epoch 7/100
38/38 [==============================] - 4s 114ms/step - loss: 0.0039
Epoch 8/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0034
Epoch 9/100
38/38 [==============================] - 4s 114ms/step - loss: 0.0032
Epoch 10/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0033
Epoch 11/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0031
Epoch 12/100
38/38 [==============================] - 4s 113ms/step - loss: 0.0028
Epoch 13/100
38/38 [==============================] - 4s 115ms/step - loss: 0.0028
Epoch 14/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0027
Epoch 15/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0024
Epoch 16/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0025
Epoch 17/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0030
Epoch 18/100
38/38 [==============================] - 5s 127ms/step - loss: 0.0025
Epoch 19/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0028
Epoch 20/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0023
Epoch 21/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0030
Epoch 22/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0023
Epoch 23/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0022
Epoch 24/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0026
Epoch 25/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0023
Epoch 26/100
38/38 [==============================] - 5s 119ms/step - loss: 0.0022
Epoch 27/100
38/38 [==============================] - 5s 119ms/step - loss: 0.0023
Epoch 28/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0026
Epoch 29/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0020
Epoch 30/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0022
Epoch 31/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0020
Epoch 32/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0020
```

```
Epoch 33/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0025
Epoch 34/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0021
Epoch 35/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0019
Epoch 36/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0020
Epoch 37/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0021
Epoch 38/100
38/38 [==============================] - 5s 123ms/step - loss: 0.0020
Epoch 39/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0017
Epoch 40/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0020
Epoch 41/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0018
Epoch 42/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0017
Epoch 43/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0017
Epoch 44/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0017
Epoch 45/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0017
Epoch 46/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0015
Epoch 47/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0017
Epoch 48/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0016
Epoch 49/100
38/38 [==============================] - 5s 123ms/step - loss: 0.0018
Epoch 50/100
38/38 [==============================] - 5s 125ms/step - loss: 0.0017
Epoch 51/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0017
Epoch 52/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0015
Epoch 53/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0017
Epoch 54/100
38/38 [==============================] - 5s 125ms/step - loss: 0.0016
Epoch 55/100
38/38 [==============================] - 5s 127ms/step - loss: 0.0015
Epoch 56/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0016
Epoch 57/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0014
Epoch 58/100
38/38 [==============================] - 5s 138ms/step - loss: 0.0014
Epoch 59/100
38/38 [==============================] - 5s 140ms/step - loss: 0.0013
Epoch 60/100
38/38 [==============================] - 5s 139ms/step - loss: 0.0015
Epoch 61/100
38/38 [==============================] - 8s 199ms/step - loss: 0.0015
Epoch 62/100
38/38 [==============================] - 7s 175ms/step - loss: 0.0014
Epoch 63/100
38/38 [==============================] - 6s 149ms/step - loss: 0.0014
Epoch 64/100
38/38 [==============================] - 7s 180ms/step - loss: 0.0014
```

```
Epoch 65/100
38/38 [==============================] - 7s 183ms/step - loss: 0.0014
Epoch 66/100
38/38 [==============================] - 7s 175ms/step - loss: 0.0013
Epoch 67/100
38/38 [==============================] - 6s 152ms/step - loss: 0.0014
Epoch 68/100
38/38 [==============================] - 5s 131ms/step - loss: 0.0014
Epoch 69/100
38/38 [==============================] - 5s 125ms/step - loss: 0.0013
Epoch 70/100
38/38 [==============================] - 4s 117ms/step - loss: 0.0014
Epoch 71/100
38/38 [==============================] - 5s 125ms/step - loss: 0.0012
Epoch 72/100
38/38 [==============================] - 5s 124ms/step - loss: 0.0012
Epoch 73/100
38/38 [==============================] - 5s 137ms/step - loss: 0.0013
Epoch 74/100
38/38 [==============================] - 5s 124ms/step - loss: 0.0012
Epoch 75/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0014
Epoch 76/100
38/38 [==============================] - 5s 126ms/step - loss: 0.0012
Epoch 77/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0012
Epoch 78/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0012
Epoch 79/100
38/38 [==============================] - 5s 128ms/step - loss: 0.0011
Epoch 80/100
38/38 [==============================] - 5s 123ms/step - loss: 0.0012
Epoch 81/100
38/38 [==============================] - 5s 120ms/step - loss: 0.0010
Epoch 82/100
38/38 [==============================] - 6s 158ms/step - loss: 0.0012
Epoch 83/100
38/38 [==============================] - 5s 124ms/step - loss: 0.0013
Epoch 84/100
38/38 [==============================] - 5s 134ms/step - loss: 0.0012
Epoch 85/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0011
Epoch 86/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0012
Epoch 87/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0011
Epoch 88/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0010
Epoch 89/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0011
Epoch 90/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0011
Epoch 91/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0011
Epoch 92/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0011
Epoch 93/100
38/38 [==============================] - 5s 124ms/step - loss: 0.0011
Epoch 94/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0011
Epoch 95/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0010
Epoch 96/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0010
```

```
Epoch 97/100
38/38 [==============================] - 5s 121ms/step - loss: 0.0010
Epoch 98/100
38/38 [==============================] - 5s 122ms/step - loss: 0.0011
Epoch 99/100
38/38 [==============================] - 5s 119ms/step - loss: 9.5672e-04
Epoch 100/100
38/38 [==============================] - 5s 119ms/step - loss: 9.4282e-04
```

Out[20]:
```
<keras.callbacks.History at 0x1a828e3ae20>
```

In [21]:
```
dataset_test =pd.read_csv("testset.csv")
```

In [22]:
```
real_stock_price = dataset_test.iloc[:,1:2].values
```

In [23]:
```
dataset_total = pd.concat((dataset_train['Open'],dataset_test['Open']),axis = 0)
dataset_total
```

Out[23]:
```
0       357.385559
1       360.122742
2       362.313507
3       365.348755
4       365.393463
          ...
120    1143.599976
121    1128.000000
122    1121.339966
123    1102.089966
124    1120.000000
Name: Open, Length: 1384, dtype: float64
```

In [24]:
```
inputs = dataset_total[len(dataset_total) - len(dataset_test)-60:].values
inputs
```

Out[24]:
```
array([ 955.48999 ,  966.700012,  980.      ,  980.      ,  973.719971,
        987.450012,  992.      ,  992.099976,  990.289978,  991.77002 ,
        986.      ,  989.440002,  989.52002 ,  970.      ,  968.369995,
        980.      , 1009.190002, 1014.      , 1015.219971, 1017.210022,
       1021.76001 , 1022.109985, 1028.98999 , 1027.27002 , 1030.52002 ,
       1033.98999 , 1026.459961, 1023.419983, 1022.590027, 1019.210022,
       1022.52002 , 1034.01001 , 1020.26001 , 1023.309998, 1035.      ,
       1035.869995, 1040.      , 1055.089966, 1042.680054, 1022.369995,
       1015.799988, 1012.659973,  995.940002, 1001.5     , 1020.429993,
       1037.48999 , 1035.5     , 1039.630005, 1046.119995, 1045.      ,
       1054.609985, 1066.079956, 1075.199951, 1071.780029, 1064.949951,
       1061.109985, 1058.069946, 1057.390015, 1051.599976, 1046.719971,
       1048.339966, 1064.310059, 1088.      , 1094.      , 1102.22998 ,
       1109.400024, 1097.099976, 1106.300049, 1102.410034, 1132.51001 ,
       1126.219971, 1131.410034, 1131.829956, 1137.48999 , 1159.849976,
       1177.329956, 1172.530029, 1175.079956, 1176.47998 , 1167.829956,
       1170.569946, 1162.609985, 1122.      , 1090.599976, 1027.180054,
       1081.540039, 1055.410034, 1017.25    , 1048.      , 1045.      ,
       1048.949951, 1079.069946, 1088.410034, 1090.569946, 1106.469971,
       1116.189941, 1112.640015, 1127.800049, 1141.23999 , 1123.030029,
       1107.869995, 1053.079956, 1075.140015, 1099.219971, 1089.189941,
       1115.319946, 1136.      , 1163.849976, 1170.      , 1145.209961,
       1149.959961, 1154.140015, 1120.01001 , 1099.      , 1092.73999 ,
       1081.880005, 1047.030029, 1046.      , 1063.      ,  998.      ,
       1011.630005, 1022.820007, 1013.909973,  993.409973, 1041.329956,
       1020.      , 1016.799988, 1026.439941, 1027.98999 , 1025.040039,
       1040.880005, 1037.      , 1051.369995, 1077.430054, 1069.400024,
       1082.      , 1077.859985, 1052.      , 1025.52002 , 1029.51001 ,
       1046.      , 1030.01001 , 1013.659973, 1028.099976, 1019.      ,
       1016.900024, 1049.22998 , 1058.540039, 1058.099976, 1086.030029,
       1093.599976, 1100.      , 1090.      , 1077.310059, 1079.890015,
       1061.859985, 1074.060059, 1083.560059, 1065.130005, 1079.      ,
       1079.02002 , 1064.890015, 1063.030029, 1067.560059, 1099.349976,
       1122.329956, 1140.98999 , 1142.170044, 1131.319946, 1118.180054,
       1118.599976, 1131.069946, 1141.119995, 1143.849976, 1148.859985,
       1143.650024, 1158.5     , 1175.310059, 1174.849976, 1159.140015,
       1143.599976, 1128.      , 1121.339966, 1102.089966, 1120.      ])
```

In [25]:
```
inputs = inputs.reshape(-1,1)
```

In [26]:
```
inputs
```

Out[26]:
```
array([[ 955.48999 ],
       [ 966.700012],
       [ 980.      ],
       [ 980.      ],
       [ 973.719971],
       [ 987.450012],
       [ 992.      ],
       [ 992.099976],
       [ 990.289978],
       [ 991.77002 ],
       [ 986.      ],
       [ 989.440002],
       [ 989.52002 ],
       [ 970.      ],
       [ 968.369995],
       [ 980.      ],
       [1009.190002],
       [1014.      ],
       [1015.219971],
       [1017.210022],
       [1021.76001 ],
       [1022.109985],
       [1028.98999 ],
       [1027.27002 ],
       [1030.52002 ],
       [1033.98999 ],
       [1026.459961],
       [1023.419983],
       [1022.590027],
       [1019.210022],
       [1022.52002 ],
       [1034.01001 ],
       [1020.26001 ],
       [1023.309998],
       [1035.      ],
       [1035.869995],
       [1040.      ],
       [1055.089966],
       [1042.680054],
       [1022.369995],
       [1015.799988],
       [1012.659973],
       [ 995.940002],
       [1001.5     ],
       [1020.429993],
       [1037.48999 ],
       [1035.5     ],
       [1039.630005],
       [1046.119995],
       [1045.      ],
       [1054.609985],
       [1066.079956],
       [1075.199951],
       [1071.780029],
       [1064.949951],
       [1061.109985],
       [1058.069946],
       [1057.390015],
       [1051.599976],
       [1046.719971],
       [1048.339966],
       [1064.310059],
       [1088.      ],
       [1094.      ],
```

```
[1102.22998 ],
[1109.400024],
[1097.099976],
[1106.300049],
[1102.410034],
[1132.51001 ],
[1126.219971],
[1131.410034],
[1131.829956],
[1137.48999 ],
[1159.849976],
[1177.329956],
[1172.530029],
[1175.079956],
[1176.47998 ],
[1167.829956],
[1170.569946],
[1162.609985],
[1122.      ],
[1090.599976],
[1027.180054],
[1081.540039],
[1055.410034],
[1017.25    ],
[1048.      ],
[1045.      ],
[1048.949951],
[1079.069946],
[1088.410034],
[1090.569946],
[1106.469971],
[1116.189941],
[1112.640015],
[1127.800049],
[1141.23999 ],
[1123.030029],
[1107.869995],
[1053.079956],
[1075.140015],
[1099.219971],
[1089.189941],
[1115.319946],
[1136.      ],
[1163.849976],
[1170.      ],
[1145.209961],
[1149.959961],
[1154.140015],
[1120.01001 ],
[1099.      ],
[1092.73999 ],
[1081.880005],
[1047.030029],
[1046.      ],
[1063.      ],
[ 998.      ],
[1011.630005],
[1022.820007],
[1013.909973],
[ 993.409973],
[1041.329956],
[1020.      ],
[1016.799988],
[1026.439941],
```

```
       [1027.98999 ],
       [1025.040039],
       [1040.880005],
       [1037.      ],
       [1051.369995],
       [1077.430054],
       [1069.400024],
       [1082.      ],
       [1077.859985],
       [1052.      ],
       [1025.52002 ],
       [1029.51001 ],
       [1046.      ],
       [1030.01001 ],
       [1013.659973],
       [1028.099976],
       [1019.      ],
       [1016.900024],
       [1049.22998 ],
       [1058.540039],
       [1058.099976],
       [1086.030029],
       [1093.599976],
       [1100.      ],
       [1090.      ],
       [1077.310059],
       [1079.890015],
       [1061.859985],
       [1074.060059],
       [1083.560059],
       [1065.130005],
       [1079.      ],
       [1079.02002 ],
       [1064.890015],
       [1063.030029],
       [1067.560059],
       [1099.349976],
       [1122.329956],
       [1140.98999 ],
       [1142.170044],
       [1131.319946],
       [1118.180054],
       [1118.599976],
       [1131.069946],
       [1141.119995],
       [1143.849976],
       [1148.859985],
       [1143.650024],
       [1158.5     ],
       [1175.310059],
       [1174.849976],
       [1159.140015],
       [1143.599976],
       [1128.      ],
       [1121.339966],
       [1102.089966],
       [1120.      ]])
```

In [27]:
```
inputs = sc.transform(inputs)
inputs.shape
```

Out[27]:
```
(185, 1)
```

In [28]:
```
x_test = []
for i in range(60,185):
    x_test.append(inputs[i-60:i,0])
```

In [29]:
```
x_test = np.array(x_test)
x_test.shape
```

Out[29]:
```
(125, 60)
```

In [30]:
```
x_test = np.reshape(x_test, (x_test.shape[0],x_test.shape[1],1))
x_test.shape
```

Out[30]:
```
(125, 60, 1)
```

In [31]:
```
predicted_price = regressor.predict(x_test)
```

```
4/4 [==============================] - 2s 40ms/step
```

In [32]:
```
predicted_price = sc.inverse_transform(predicted_price)
predicted_price
```

Out[32]:
```
array([[1062.5486],
       [1061.2917],
       [1063.7489],
       [1072.1361],
       [1082.6061],
       [1092.1124],
       [1099.5701],
       [1101.9546],
       [1102.7546],
       [1102.7412],
       [1108.0972],
       [1114.7013],
       [1120.7512],
       [1125.1423],
       [1128.8447],
       [1135.6974],
       [1146.101 ],
       [1154.7587],
       [1160.4332],
       [1163.7854],
       [1164.2369],
       [1164.1534],
       [1163.2083],
       [1155.5398],
       [1141.1187],
       [1116.9062],
       [1102.0822],
       [1094.4454],
       [1084.9731],
       [1079.9023],
       [1078.4199],
       [1078.8372],
       [1084.223 ],
       [1092.1284],
       [1098.5038],
       [1104.227 ],
       [1109.7949],
       [1113.1987],
       [1117.159 ],
       [1123.335 ],
       [1126.1998],
       [1123.7347],
       [1110.2096],
       [1097.9507],
       [1095.7228],
       [1098.7268],
       [1107.0674],
       [1119.3202],
       [1134.6855],
       [1148.1073],
       [1151.9373],
       [1150.5538],
       [1148.7622],
       [1142.3306],
       [1131.6748],
       [1121.0702],
       [1112.2205],
       [1100.9161],
       [1090.2499],
       [1085.7399],
       [1075.0256],
       [1063.3169],
       [1056.2959],
       [1052.2762],
```

```
       [1046.1387],
       [1046.9768],
       [1048.6714],
       [1047.9353],
       [1046.8129],
       [1046.1108],
       [1045.0955],
       [1046.5532],
       [1048.5759],
       [1052.4198],
       [1061.0278],
       [1068.8362],
       [1075.5725],
       [1079.496 ],
       [1076.4847],
       [1065.9207],
       [1055.0175],
       [1050.8988],
       [1049.7704],
       [1046.7648],
       [1045.1338],
       [1043.59  ],
       [1041.6158],
       [1045.2109],
       [1053.221 ],
       [1060.7264],
       [1070.0176],
       [1079.5287],
       [1087.5846],
       [1090.997 ],
       [1088.9382],
       [1085.5153],
       [1080.201 ],
       [1077.6609],
       [1079.8658],
       [1081.0935],
       [1083.1359],
       [1085.5034],
       [1084.9102],
       [1082.1841],
       [1080.0504],
       [1084.7563],
       [1096.608 ],
       [1111.9441],
       [1124.4152],
       [1129.6138],
       [1127.59  ],
       [1123.2555],
       [1122.2422],
       [1126.0238],
       [1132.0613],
       [1138.3765],
       [1142.2733],
       [1146.583 ],
       [1153.5225],
       [1160.1927],
       [1162.0658],
       [1158.3788],
       [1150.7166],
       [1142.5703],
       [1133.7119]], dtype=float32)
```

In [33]:
```
plt.plot(real_stock_price,color = 'red', label = 'Real Price')
plt.plot(predicted_price, color = 'blue', label = 'Predicted Price')
```

```
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
plt.show()
```