Beginner Level Task...

Task-2 Stock Market Prediction And Forecasting Using Stacked LSTM :

Dataset: https://raw.githubusercontent.com/mwitiderrick/stockprice/master/NSE-TATAGLOBAL.csv

Name: Madhav Pandey

1. Importing Required Libraries and Packages

In [19]:
```python
import numpy as np
import pandas as pd
import tensorflow as ft
import datetime
import math
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense , LSTM
from sklearn.metrics import mean_squared_error
print(" All required packages included successfully!")
```

```
 All required packages included successfully!
```

2. Importing the Dataset

In [20]:
```python
Dataset_link='https://raw.githubusercontent.com/mwitiderrick/stockprice/master/NSE-TATAGLOBAL.csv'
```

In [21]:
```python
df= pd.read_csv(Dataset_link, parse_dates=True,)
df.reset_index()
df.head(10)
```

Out[21]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-09-28	234.05	235.95	230.20	233.50	233.75	3069914	7162.35
1	2018-09-27	234.55	236.80	231.10	233.80	233.25	5082859	11859.95
2	2018-09-26	240.00	240.00	232.50	235.00	234.25	2240909	5248.60
3	2018-09-25	233.30	236.75	232.00	236.25	236.10	2349368	5503.90
4	2018-09-24	233.55	239.20	230.75	234.00	233.30	3423509	7999.55
5	2018-09-21	235.00	237.00	227.95	233.75	234.60	5395319	12589.59
6	2018-09-19	235.95	237.20	233.45	234.60	234.90	1362058	3202.78
7	2018-09-18	237.90	239.25	233.50	235.50	235.05	2614794	6163.70
8	2018-09-17	233.15	238.00	230.25	236.40	236.60	3170894	7445.41
9	2018-09-14	223.45	236.70	223.30	234.00	233.95	6377909	14784.50

3. Performing EDA

In [22]:
```python
df.sample(10)
```

Out[22]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
538	2016-07-28	138.00	142.25	136.55	140.10	140.15	3754038	5255.56
26	2018-08-20	244.00	247.00	243.00	244.70	245.15	1690225	4141.83
266	2017-09-04	198.00	198.40	190.50	193.30	193.30	4372250	8461.60
1993	2010-09-17	122.80	125.35	121.60	123.15	122.65	2098722	2583.19
1989	2010-09-23	121.60	124.25	119.90	123.60	123.55	1845230	2272.82
1940	2010-12-02	115.90	117.50	114.35	115.55	115.25	887683	1026.67
1633	2012-02-27	119.00	120.20	112.00	113.10	113.05	1524588	1756.34
1445	2012-11-26	164.70	166.60	162.25	163.35	163.40	3080275	5070.26
1845	2011-04-21	101.50	103.60	101.50	103.05	103.05	778698	802.26
1309	2013-06-12	142.75	144.95	135.25	141.55	141.60	4596668	6453.26

In [23]:
```python
df.columns
```

Out[23]:
```
Index(['Date', 'Open', 'High', 'Low', 'Last', 'Close', 'Total Trade Quantity',
       'Turnover (Lacs)'],
      dtype='object')
```

In [24]:
```python
df.shape
```

Out[24]:
```
(2035, 8)
```

In [25]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2035 entries, 0 to 2034
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Date                  2035 non-null   object 
 1   Open                  2035 non-null   float64
 2   High                  2035 non-null   float64
 3   Low                   2035 non-null   float64
 4   Last                  2035 non-null   float64
 5   Close                 2035 non-null   float64
 6   Total Trade Quantity  2035 non-null   int64  
 7   Turnover (Lacs)       2035 non-null   float64
dtypes: float64(6), int64(1), object(1)
memory usage: 127.3+ KB
```

In [26]:
```python
df.isnull().sum()
```

Out[26]:
```
Date                    0
Open                    0
High                    0
Low                     0
Last                    0
Close                   0
Total Trade Quantity    0
Turnover (Lacs)         0
dtype: int64
```

In [27]:
```python
df.describe()
```

Out[27]:

	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
count	2035.000000	2035.000000	2035.000000	2035.000000	2035.00000	2.035000e+03	2035.000000
mean	149.713735	151.992826	147.293931	149.474251	149.45027	2.335681e+06	3899.980565
std	48.664509	49.413109	47.931958	48.732570	48.71204	2.091778e+06	4570.767877
min	81.100000	82.800000	80.000000	81.000000	80.95000	3.961000e+04	37.040000
25%	120.025000	122.100000	118.300000	120.075000	120.05000	1.146444e+06	1427.460000
50%	141.500000	143.400000	139.600000	141.100000	141.25000	1.783456e+06	2512.030000
75%	157.175000	159.400000	155.150000	156.925000	156.90000	2.813594e+06	4539.015000
max	327.700000	328.750000	321.650000	325.950000	325.75000	2.919102e+07	55755.080000

4. Parametric Visualization

In [28]:
```python
plt.figure(figsize=(10,6))
df['Close'].plot(kind='line',figsize=(16,7),color='b',label="Closing Price")
plt.ylabel("Price")
plt.legend(loc="upper right")
plt.title("Change in closing price over the years")
plt.grid()
```

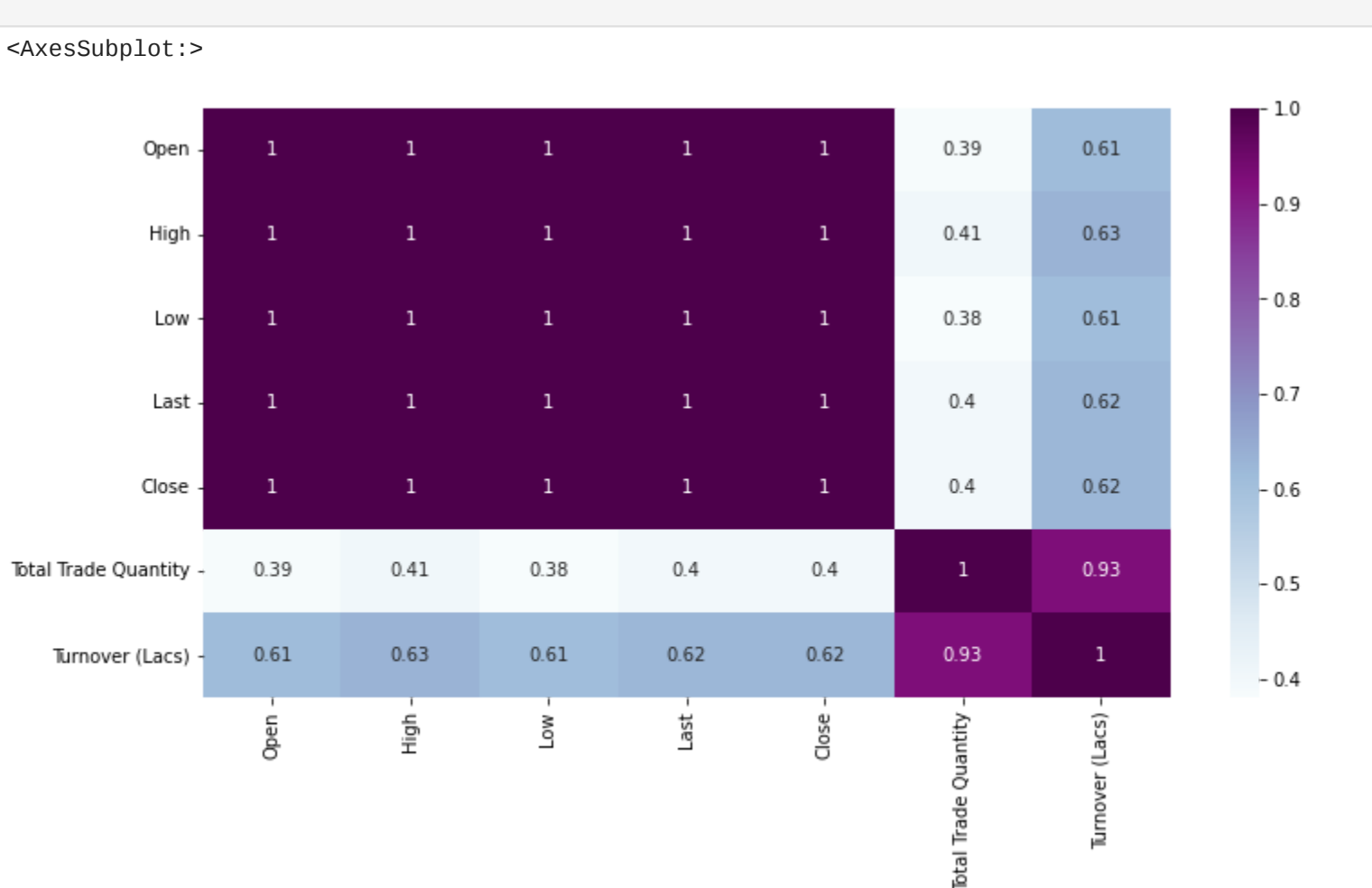
In [29]:
```python
plt.figure(figsize=(10,6))
df['Open'].plot(kind='line',figsize=(16,7),color='g',label="Opening Price")
plt.ylabel("Price")
plt.legend(loc="upper left")
plt.title("Change in opening price over the years")
plt.grid()
```

In [30]:
```python
df1=df.reset_index()['Close']
df1
```

Out[30]:
```
0       233.75
1       233.25
2       234.25
3       236.10
4       233.30
         ...  
2030    118.65
2031    117.60
2032    120.65
2033    120.90
2034    121.55
Name: Close, Length: 2035, dtype: float64
```

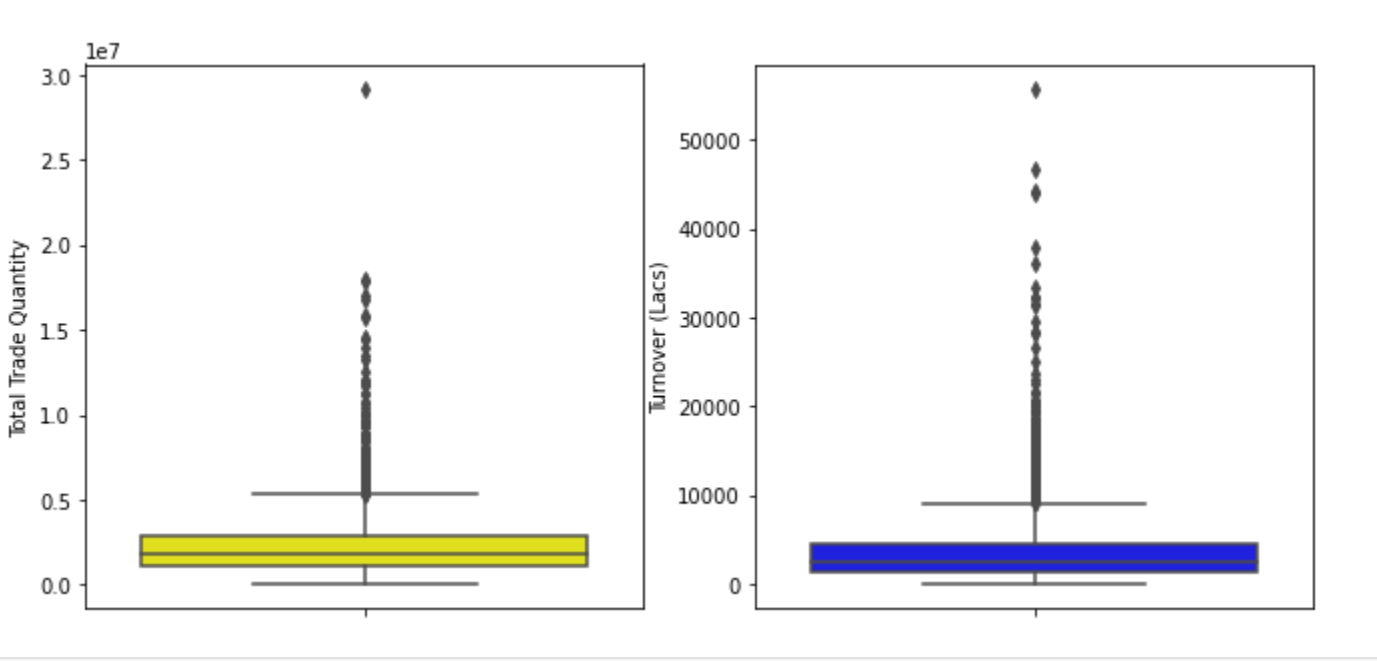
In [31]:
```python
plt.figure(figsize=(12,6))
sns.heatmap(df.corr(),annot=True,cmap='BuPu')
```

Out[31]:
```
<AxesSubplot:>
```



In [32]:
```python
plt.figure(figsize=(11,5))
plt.subplot(1,2,1)
sns.boxplot(data=df,y='Total Trade Quantity',color='yellow')
plt.subplot(1,2,2)
sns.boxplot(data=df,y='Turnover (Lacs)',color='blue')
```

Out[32]:
```
<AxesSubplot:ylabel='Turnover (Lacs)'>
```



In [33]:
```python
training_set= df[['Open']]
training_set=pd.DataFrame(training_set)
training_set
```

Out[33]:

	Open
0	234.05
1	234.55
2	240.00
3	233.30
4	233.55
...	...
2030	117.60
2031	120.10
2032	121.80
2033	120.30
2034	122.10

2035 rows × 1 columns

5.Splitting and Transforming the Dataset

In [34]:
```python
scaler=MinMaxScaler(feature_range=(0,1))
training_set_scaler=scaler.fit_transform(np.array(df1).reshape(-1,1))
```

In [35]:
```python
train_size1= int(len(training_set_scaler)*0.65)
test_size1=int(len(training_set_scaler))-train_size1
train_data1,test_data1=training_set_scaler[0:train_size1,:],training_set_scaler[train_size1:len(df),:1]
```

In [36]:
```python
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 [37]:
```python
time_step=100
x_train, y_train=create_dataset(train_data1, time_step)
x_test, y_test= create_dataset(test_data1, time_step)
```

In [38]:
```python
print(x_train.shape,y_train.shape)
```

```
(1221, 100) (1221,)
```

In [39]:
```python
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)
```

6.Building the Model

In [40]:
```python
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(100,1)))
model.add(LSTM(50, return_sequences=True, input_shape=(100,1)))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics='acc')
```

In [41]:
```python
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 lstm_3 (LSTM)               (None, 100, 50)           10400     

 lstm_4 (LSTM)               (None, 100, 50)           20200     

 lstm_5 (LSTM)               (None, 50)                20200     

 dense_1 (Dense)             (None, 1)                 51        

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

In [42]:
```python
model.fit(x_train, y_train, validation_data = (x_test, y_test), epochs = 75, batch_size = 64, verbose = 1)
```

```
Epoch 1/75
20/20 [==============================] - 10s 202ms/step - loss: 0.0260 - acc: 8.1900e-04 - val_loss: 0.0063 - val_acc: 0.0016
Epoch 2/75
20/20 [==============================] - 3s 138ms/step - loss: 0.0033 - acc: 8.1900e-04 - val_loss: 0.0014 - val_acc: 0.0016
Epoch 3/75
20/20 [==============================] - 3s 145ms/step - loss: 0.0018 - acc: 8.1900e-04 - val_loss: 0.0012 - val_acc: 0.0016
Epoch 4/75
20/20 [==============================] - 3s 147ms/step - loss: 0.0015 - acc: 8.1900e-04 - val_loss: 0.0013 - val_acc: 0.0016
Epoch 5/75
20/20 [==============================] - 3s 143ms/step - loss: 0.0016 - acc: 8.1900e-04 - val_loss: 0.0011 - val_acc: 0.0016
Epoch 6/75
20/20 [==============================] - 3s 138ms/step - loss: 0.0016 - acc: 8.1900e-04 - val_loss: 0.0012 - val_acc: 0.0016
Epoch 7/75
20/20 [==============================] - 3s 152ms/step - loss: 0.0013 - acc: 8.1900e-04 - val_loss: 8.9739e-04 - val_acc: 0.0016
Epoch 8/75
20/20 [==============================] - 3s 154ms/step - loss: 0.0013 - acc: 8.1900e-04 - val_loss: 8.2220e-04 - val_acc: 0.0016
Epoch 9/75
20/20 [==============================] - 3s 156ms/step - loss: 0.0014 - acc: 8.1900e-04 - val_loss: 8.8522e-04 - val_acc: 0.0016
Epoch 10/75
20/20 [==============================] - 3s 142ms/step - loss: 0.0012 - acc: 8.1900e-04 - val_loss: 8.7460e-04 - val_acc: 0.0016
Epoch 11/75
20/20 [==============================] - 3s 163ms/step - loss: 0.0012 - acc: 8.1900e-04 - val_loss: 8.2590e-04 - val_acc: 0.0016
Epoch 12/75
20/20 [==============================] - 3s 153ms/step - loss: 0.0011 - acc: 8.1900e-04 - val_loss: 8.1912e-04 - val_acc: 0.0016
Epoch 13/75
20/20 [==============================] - 3s 156ms/step - loss: 9.9576e-04 - acc: 8.1900e-04 - val_loss: 0.0011 - val_acc: 0.0016
Epoch 14/75
20/20 [==============================] - 3s 144ms/step - loss: 0.0011 - acc: 8.1900e-04 - val_loss: 8.1020e-04 - val_acc: 0.0016
Epoch 15/75
20/20 [==============================] - 3s 147ms/step - loss: 8.8078e-04 - acc: 8.1900e-04 - val_loss: 7.5177e-04 - val_acc: 0.0016
Epoch 16/75
20/20 [==============================] - 3s 150ms/step - loss: 8.5999e-04 - acc: 8.1900e-04 - val_loss: 8.9449e-04 - val_acc: 0.0016
Epoch 17/75
20/20 [==============================] - 3s 147ms/step - loss: 9.5665e-04 - acc: 8.1900e-04 - val_loss: 7.4880e-04 - val_acc: 0.0016
Epoch 18/75
20/20 [==============================] - 3s 149ms/step - loss: 8.5345e-04 - acc: 8.1900e-04 - val_loss: 6.1651e-04 - val_acc: 0.0016
Epoch 19/75
20/20 [==============================] - 3s 144ms/step - loss: 7.8839e-04 - acc: 8.1900e-04 - val_loss: 7.1381e-04 - val_acc: 0.0016
Epoch 20/75
20/20 [==============================] - 3s 157ms/step - loss: 8.2328e-04 - acc: 8.1900e-04 - val_loss: 6.1479e-04 - val_acc: 0.0016
Epoch 21/75
20/20 [==============================] - 3s 168ms/step - loss: 7.2672e-04 - acc: 8.1900e-04 - val_loss: 6.7991e-04 - val_acc: 0.0016
Epoch 22/75
20/20 [==============================] - 3s 161ms/step - loss: 7.3926e-04 - acc: 8.1900e-04 - val_loss: 7.9189e-04 - val_acc: 0.0016
Epoch 23/75
20/20 [==============================] - 3s 164ms/step - loss: 7.4080e-04 - acc: 8.1900e-04 - val_loss: 6.7126e-04 - val_acc: 0.0016
Epoch 24/75
20/20 [==============================] - 3s 156ms/step - loss: 7.2767e-04 - acc: 8.1900e-04 - val_loss: 7.0438e-04 - val_acc: 0.0016
Epoch 25/75
20/20 [==============================] - 3s 159ms/step - loss: 6.6924e-04 - acc: 8.1900e-04 - val_loss: 7.3172e-04 - val_acc: 0.0016
Epoch 26/75
20/20 [==============================] - 3s 154ms/step - loss: 7.7655e-04 - acc: 8.1900e-04 - val_loss: 6.5055e-04 - val_acc: 0.0016
Epoch 27/75
20/20 [==============================] - 3s 152ms/step - loss: 8.8357e-04 - acc: 8.1900e-04 - val_loss: 5.8496e-04 - val_acc: 0.0016
Epoch 28/75
20/20 [==============================] - 3s 150ms/step - loss: 6.6036e-04 - acc: 8.1900e-04 - val_loss: 6.5111e-04 - val_acc: 0.0016
Epoch 29/75
20/20 [==============================] - 3s 160ms/step - loss: 6.1912e-04 - acc: 8.1900e-04 - val_loss: 6.9881e-04 - val_acc: 0.0016
Epoch 30/75
20/20 [==============================] - 3s 157ms/step - loss: 6.1519e-04 - acc: 8.1900e-04 - val_loss: 6.7060e-04 - val_acc: 0.0016
Epoch 31/75
20/20 [==============================] - 3s 154ms/step - loss: 5.8753e-04 - acc: 8.1900e-04 - val_loss: 6.2238e-04 - val_acc: 0.0016
Epoch 32/75
20/20 [==============================] - 3s 149ms/step - loss: 5.8993e-04 - acc: 8.1900e-04 - val_loss: 8.8589e-04 - val_acc: 0.0016
Epoch 33/75
20/20 [==============================] - 3s 148ms/step - loss: 6.2569e-04 - acc: 8.1900e-04 - val_loss: 6.3386e-04 - val_acc: 0.0016
Epoch 34/75
20/20 [==============================] - 3s 148ms/step - loss: 5.6785e-04 - acc: 8.1900e-04 - val_loss: 6.2256e-04 - val_acc: 0.0016
Epoch 35/75
20/20 [==============================] - 3s 171ms/step - loss: 5.6692e-04 - acc: 8.1900e-04 - val_loss: 7.7165e-04 - val_acc: 0.0016
Epoch 36/75
20/20 [==============================] - 3s 162ms/step - loss: 5.1824e-04 - acc: 8.1900e-04 - val_loss: 8.3364e-04 - val_acc: 0.0016
Epoch 37/75
20/20 [==============================] - 3s 149ms/step - loss: 5.6026e-04 - acc: 8.1900e-04 - val_loss: 5.3534e-04 - val_acc: 0.0016
Epoch 38/75
20/20 [==============================] - 3s 152ms/step - loss: 5.5043e-04 - acc: 8.1900e-04 - val_loss: 5.9442e-04 - val_acc: 0.0016
Epoch 39/75
20/20 [==============================] - 3s 149ms/step - loss: 5.1911e-04 - acc: 8.1900e-04 - val_loss: 4.6276e-04 - val_acc: 0.0016
Epoch 40/75
20/20 [==============================] - 3s 147ms/step - loss: 5.5727e-04 - acc: 8.1900e-04 - val_loss: 5.2977e-04 - val_acc: 0.0016
Epoch 41/75
20/20 [==============================] - 4s 188ms/step - loss: 5.0910e-04 - acc: 8.1900e-04 - val_loss: 5.3449e-04 - val_acc: 0.0016
Epoch 42/75
20/20 [==============================] - 3s 172ms/step - loss: 5.1160e-04 - acc: 8.1900e-04 - val_loss: 6.9097e-04 - val_acc: 0.0016
Epoch 43/75
20/20 [==============================] - 3s 146ms/step - loss: 4.9448e-04 - acc: 8.1900e-04 - val_loss: 5.2911e-04 - val_acc: 0.0016
Epoch 44/75
20/20 [==============================] - 3s 156ms/step - loss: 4.9212e-04 - acc: 8.1900e-04 - val_loss: 5.6185e-04 - val_acc: 0.0016
Epoch 45/75
20/20 [==============================] - 3s 148ms/step - loss: 5.0257e-04 - acc: 8.1900e-04 - val_loss: 4.3077e-04 - val_acc: 0.0016
Epoch 46/75
20/20 [==============================] - 3s 165ms/step - loss: 5.2935e-04 - acc: 8.1900e-04 - val_loss: 5.7604e-04 - val_acc: 0.0016
Epoch 47/75
20/20 [==============================] - 3s 157ms/step - loss: 4.2433e-04 - acc: 8.1900e-04 - val_loss: 4.8099e-04 - val_acc: 0.0016
Epoch 48/75
20/20 [==============================] - 3s 156ms/step - loss: 4.4640e-04 - acc: 8.1900e-04 - val_loss: 5.1549e-04 - val_acc: 0.0016
Epoch 49/75
20/20 [==============================] - 3s 146ms/step - loss: 4.3293e-04 - acc: 8.1900e-04 - val_loss: 5.3878e-04 - val_acc: 0.0016
Epoch 50/75
20/20 [==============================] - 3s 149ms/step - loss: 4.0582e-04 - acc: 8.1900e-04 - val_loss: 4.0754e-04 - val_acc: 0.0016
Epoch 51/75
20/20 [==============================] - 3s 144ms/step - loss: 4.5254e-04 - acc: 8.1900e-04 - val_loss: 5.3495e-04 - val_acc: 0.0016
Epoch 52/75
20/20 [==============================] - 3s 153ms/step - loss: 3.9192e-04 - acc: 8.1900e-04 - val_loss: 6.1553e-04 - val_acc: 0.0016
Epoch 53/75
20/20 [==============================] - 3s 146ms/step - loss: 4.7294e-04 - acc: 8.1900e-04 - val_loss: 4.0906e-04 - val_acc: 0.0016
Epoch 54/75
20/20 [==============================] - 3s 145ms/step - loss: 4.0605e-04 - acc: 8.1900e-04 - val_loss: 3.9033e-04 - val_acc: 0.0016
Epoch 55/75
20/20 [==============================] - 3s 148ms/step - loss: 4.1835e-04 - acc: 8.1900e-04 - val_loss: 2.9979e-04 - val_acc: 0.0016
Epoch 56/75
20/20 [==============================] - 3s 143ms/step - loss: 3.5533e-04 - acc: 8.1900e-04 - val_loss: 3.4099e-04 - val_acc: 0.0016
Epoch 57/75
20/20 [==============================] - 3s 149ms/step - loss: 3.4681e-04 - acc: 8.1900e-04 - val_loss: 4.5556e-04 - val_acc: 0.0016
Epoch 58/75
20/20 [==============================] - 3s 144ms/step - loss: 3.7284e-04 - acc: 8.1900e-04 - val_loss: 3.7616e-04 - val_acc: 0.0016
Epoch 59/75
20/20 [==============================] - 3s 144ms/step - loss: 4.2066e-04 - acc: 8.1900e-04 - val_loss: 3.4693e-04 - val_acc: 0.0016
Epoch 60/75
20/20 [==============================] - 3s 146ms/step - loss: 3.5496e-04 - acc: 8.1900e-04 - val_loss: 3.2456e-04 - val_acc: 0.0016
Epoch 61/75
20/20 [==============================] - 3s 141ms/step - loss: 3.2932e-04 - acc: 8.1900e-04 - val_loss: 3.1912e-04 - val_acc: 0.0016
Epoch 62/75
20/20 [==============================] - 3s 145ms/step - loss: 3.3815e-04 - acc: 8.1900e-04 - val_loss: 3.3586e-04 - val_acc: 0.0016
Epoch 63/75
20/20 [==============================] - 3s 149ms/step - loss: 3.1228e-04 - acc: 8.1900e-04 - val_loss: 4.2619e-04 - val_acc: 0.0016
Epoch 64/75
20/20 [==============================] - 3s 153ms/step - loss: 3.4316e-04 - acc: 8.1900e-04 - val_loss: 3.1115e-04 - val_acc: 0.0016
Epoch 65/75
20/20 [==============================] - 3s 147ms/step - loss: 3.0306e-04 - acc: 8.1900e-04 - val_loss: 2.6619e-04 - val_acc: 0.0016
Epoch 66/75
20/20 [==============================] - 3s 143ms/step - loss: 3.2810e-04 - acc: 8.1900e-04 - val_loss: 2.8419e-04 - val_acc: 0.0016
Epoch 67/75
20/20 [==============================] - 3s 146ms/step - loss: 3.6841e-04 - acc: 8.1900e-04 - val_loss: 3.5210e-04 - val_acc: 0.0016
Epoch 68/75
20/20 [==============================] - 3s 146ms/step - loss: 3.1213e-04 - acc: 8.1900e-04 - val_loss: 3.9871e-04 - val_acc: 0.0016
Epoch 69/75
20/20 [==============================] - 3s 150ms/step - loss: 3.0978e-04 - acc: 8.1900e-04 - val_loss: 3.3989e-04 - val_acc: 0.0016
Epoch 70/75
20/20 [==============================] - 3s 147ms/step - loss: 2.8350e-04 - acc: 8.1900e-04 - val_loss: 3.0507e-04 - val_acc: 0.0016
Epoch 71/75
20/20 [==============================] - 3s 142ms/step - loss: 2.7105e-04 - acc: 8.1900e-04 - val_loss: 2.5275e-04 - val_acc: 0.0016
Epoch 72/75
20/20 [==============================] - 3s 141ms/step - loss: 2.9717e-04 - acc: 8.1900e-04 - val_loss: 2.3409e-04 - val_acc: 0.0016
Epoch 73/75
20/20 [==============================] - 3s 145ms/step - loss: 2.6361e-04 - acc: 8.1900e-04 - val_loss: 4.4546e-04 - val_acc: 0.0016
Epoch 74/75
20/20 [==============================] - 3s 145ms/step - loss: 4.4456e-04 - acc: 8.1900e-04 - val_loss: 2.8315e-04 - val_acc: 0.0016
Epoch 75/75
20/20 [==============================] - 3s 146ms/step - loss: 2.7079e-04 - acc: 8.1900e-04 - val_loss: 2.7821e-04 - val_acc: 0.0016
```

Out[42]:
```
<keras.callbacks.History at 0x22b126a5be0>
```

7.Transforming Back to login

In [43]:
```python
train_predict1=model.predict(x_train)
test_predict1=model.predict(x_test)
train_predict1=scaler.inverse_transform(train_predict1)
test_predict1=scaler.inverse_transform(test_predict1)
```

```
39/39 [==============================] - 4s 26ms/step
20/20 [==============================] - 1s 25ms/step
```

In [44]:
```python
math.sqrt(mean_squared_error(y_train,train_predict1))
```

Out[44]:
```
159.86411890426697
```

In [45]:
```python
math.sqrt(mean_squared_error(y_test,test_predict1))
```

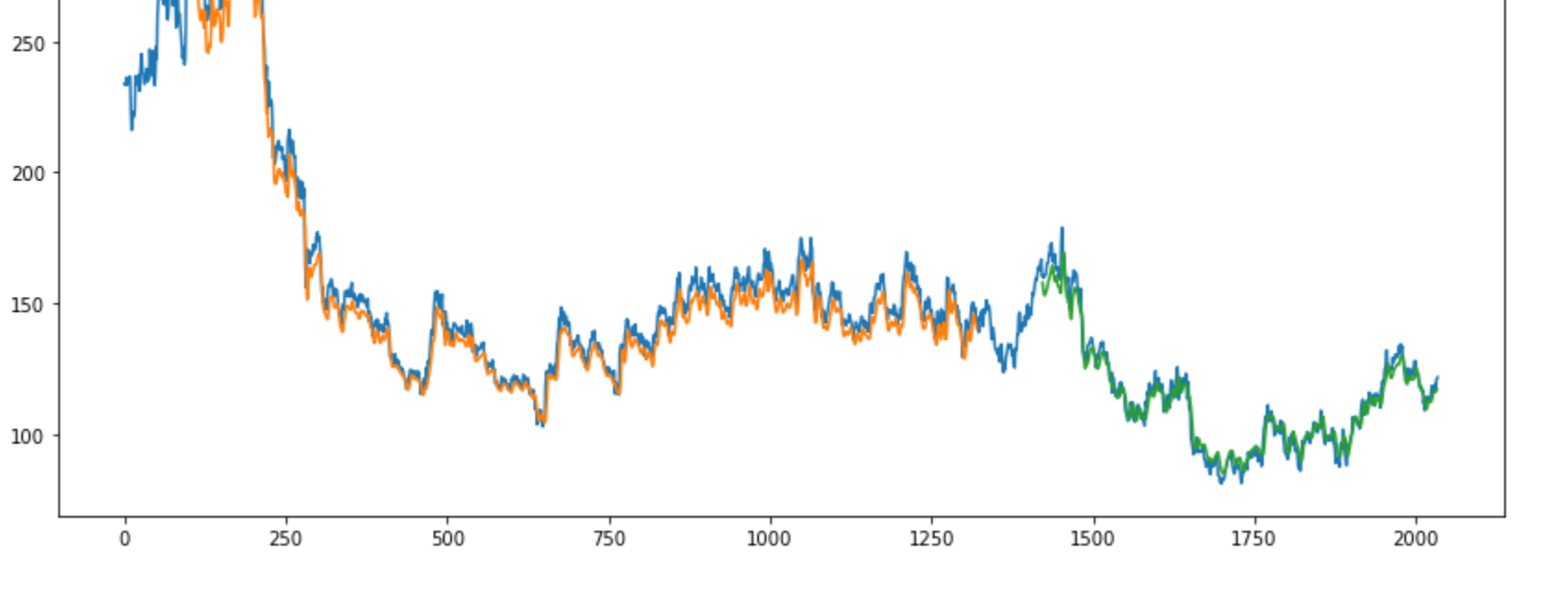
Out[45]:
```
113.66234954895712
```

8.Plotting

In [46]:
```python
# shift train predictions for plotting
look_back=100
trainPredictPlot = np.empty_like(training_set_scaler)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict1)+look_back, :] = train_predict1

# shift test predictions for plotting
testPredictPlot = np.empty_like(training_set_scaler)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(train_predict1)+(look_back*2)+1:len(df1)-1, :] = test_predict1

# plot baseline and predictions
plt.figure(figsize=(14,7))
plt.plot(scaler.inverse_transform(training_set_scaler))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```



THANK YOU!!