EXPERIMENT - 3

AIM: To develop linear regression model for forecasting time series data.

PROCEDURE AND CODE:

Steps 1: Importing the necessary packages.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

Step 2: Splitting the dataset.

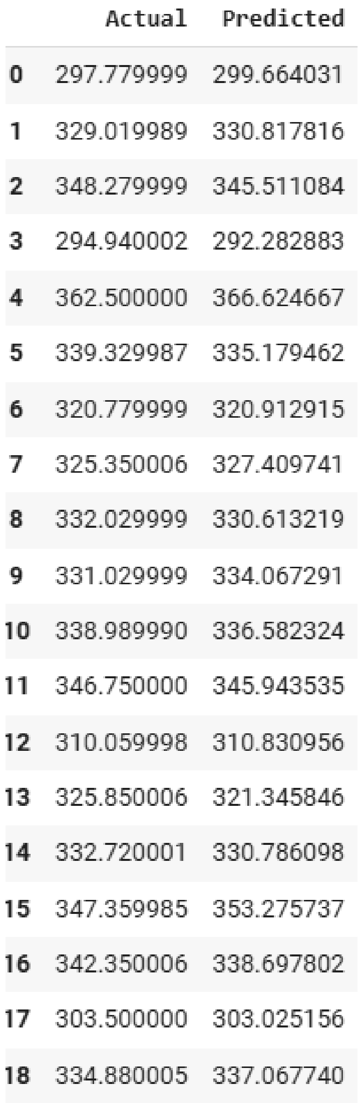
```
X = data[['High','Low','Open','Volume']].values
y = data['Close'].values
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3, random_state=1)
```

Step 3: Fitting the linear regression model.

```
regressor = LinearRegression()
regressor.fit(X_train, y_train)
predicted = regressor.predict(X_test)
```

Step 4: Printing the actual and predicted value for better understanding

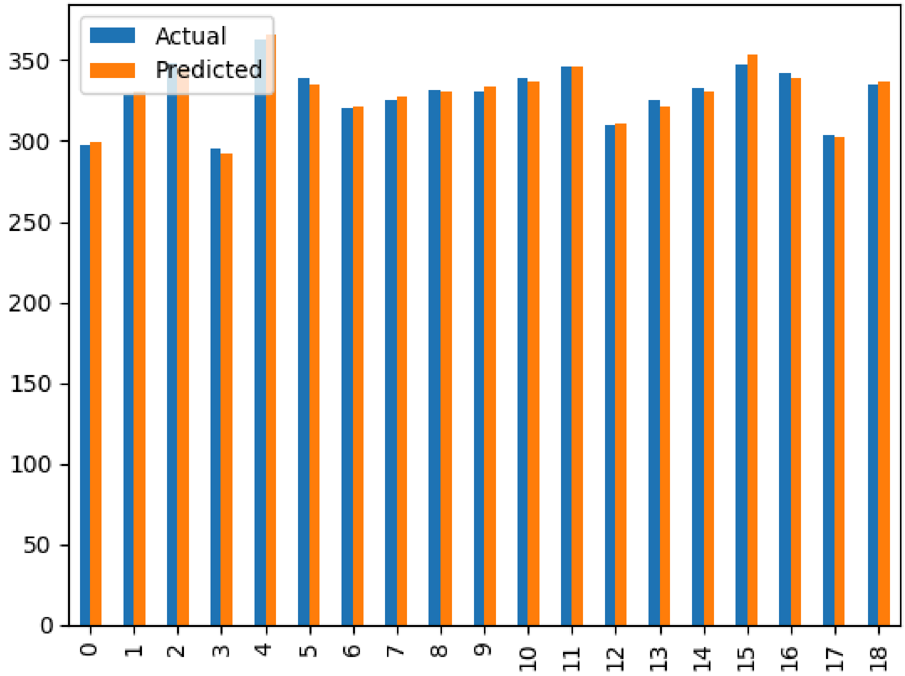
```
data1 = pd.DataFrame({'Actual': y_test.flatten(), 'Predicted' : predicted.flatten()})
data1.head(20)
```



	Actual	Predicted
0	297.779999	299.664031
1	329.019989	330.817816
2	348.279999	345.511084
3	294.940002	292.282883
4	362.500000	366.624667
5	339.329987	335.179462
6	320.779999	320.912915
7	325.350006	327.409741
8	332.029999	330.613219
9	331.029999	334.067291
10	338.989990	336.582324
11	346.750000	345.943535
12	310.059998	310.830956
13	325.850006	321.345846
14	332.720001	330.786098
15	347.359985	353.275737
16	342.350006	338.697802
17	303.500000	303.025156
18	334.880005	337.067740

Steps 5: Visualizing the actual and predicted value using graph plot.

```
graph = data1.head(20)
graph.plot(kind='bar')
```



Result: The program to implement a program for visualizing time series data is successfully implemented.