In [11]:
```python
# 1.    Load the basic libraries and packages

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from keras.models import Sequential
from keras.layers import LSTM, Dropout, Dense
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

warnings.filterwarnings('ignore')
```

In [12]:
```python
# 2.    Load the dataset

dataset = pd.read_csv("/content/yahoofinance.csv")
print(dataset.head())
```

```
         Date        Open        High         Low       Close   Adj Close  \
0  2019-03-25  191.509995  191.979996  186.600006  188.740005  186.301483
1  2019-03-26  191.660004  192.880005  184.580002  186.789993  184.376678
2  2019-03-27  188.750000  189.759995  186.550003  188.470001  186.034973
3  2019-03-28  188.949997  189.559998  187.529999  188.720001  186.281738
4  2019-03-29  189.830002  190.080002  188.539993  189.949997  187.495865

     Volume
0  43845300
1  49800500
2  29848400
3  20780400
4  23564000
```

In [13]:
```python
# 3.    Analyse the dataset

# Summary statistics
print(dataset.describe())

# Dataset info
print(dataset.info())
```

```
             Open        High         Low       Close   Adj Close  \
count  253.000000  253.000000  253.000000  253.000000  253.000000
mean   236.844229  239.629328  234.602332  237.338498  236.091135
std     43.534331   44.316390   43.064055   43.892846   44.482641
min    175.440002  177.919998  170.270004  173.300003  171.719727
25%    201.410004  203.529999  199.669998  201.750000  200.239075
50%    220.539993  222.490005  217.649994  220.699997  219.518402
75%    267.480011  271.000000  265.390015  268.480011  267.844330
max    324.739990  327.850006  323.350006  327.200012  327.200012

             Volume
count  2.530000e+02
mean   3.164449e+07
std    1.677487e+07
min    1.136200e+07
25%    2.114340e+07
50%    2.655100e+07
75%    3.480580e+07
max    1.067212e+08
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253 entries, 0 to 252
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       253 non-null    object
 1   Open       253 non-null    float64
 2   High       253 non-null    float64
 3   Low        253 non-null    float64
 4   Close      253 non-null    float64
 5   Adj Close  253 non-null    float64
 6   Volume     253 non-null    int64
dtypes: float64(5), int64(1), object(1)
memory usage: 14.0+ KB
None
```

In [14]:

```
# 4.    Apply LSTM Model

# Extract the feature column for modeling
apple_training = dataset.iloc[:, 1:2].values

# Normalize the feature using Min-Max Scaler
scaler = MinMaxScaler(feature_range=(0, 1))
apple_training_scaled = scaler.fit_transform(apple_training)

# Create the feature set and labels
feature_set = []
labels = []

for i in range(60, 253):
    feature_set.append(apple_training_scaled[i-60:i, 0])
    labels.append(apple_training_scaled[i, 0])

# Convert to numpy arrays
feature_set = np.array(feature_set)
labels = np.array(labels)

# Reshape feature set for LSTM input
feature_set = np.reshape(feature_set, (feature_set.shape[0], feature_set.shape[1

# Initialize the model
```

```python
model = Sequential()

# Add LSTM layers with Dropout
model.add(LSTM(units=60, return_sequences=True, input_shape=(feature_set.shape[1
model.add(Dropout(0.20))

model.add(LSTM(units=60, return_sequences=True))
model.add(Dropout(0.20))

model.add(LSTM(units=60, return_sequences=True))
model.add(Dropout(0.20))

model.add(LSTM(units=60))
model.add(Dropout(0.20))

# Add output layer
model.add(Dense(units=1))

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])
```

In [15]:
```python
# 5.    Apply the training over the dataset to minimize the loss

history = model.fit(feature_set, labels, epochs=100, batch_size=32, validation_s
```

```
Epoch 1/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 9s 306ms/step - accuracy: 0.0000e+00 - loss: 0.2080 - va
l_accuracy: 0.0256 - val_loss: 0.1115
Epoch 2/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 2s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0283 - va
l_accuracy: 0.0256 - val_loss: 0.0404
Epoch 3/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0179 - va
l_accuracy: 0.0256 - val_loss: 0.0407
Epoch 4/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 142ms/step - accuracy: 0.0000e+00 - loss: 0.0105 - va
l_accuracy: 0.0256 - val_loss: 0.0950
Epoch 5/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0083 - va
l_accuracy: 0.0256 - val_loss: 0.0503
Epoch 6/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 143ms/step - accuracy: 0.0000e+00 - loss: 0.0063 - va
l_accuracy: 0.0256 - val_loss: 0.1485
Epoch 7/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 209ms/step - accuracy: 0.0000e+00 - loss: 0.0044 - va
l_accuracy: 0.0256 - val_loss: 0.0979
Epoch 8/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 238ms/step - accuracy: 0.0000e+00 - loss: 0.0047 - va
l_accuracy: 0.0256 - val_loss: 0.1641
Epoch 9/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 243ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.1058
Epoch 10/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 171ms/step - accuracy: 0.0000e+00 - loss: 0.0044 - va
l_accuracy: 0.0256 - val_loss: 0.1282
Epoch 11/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0050 - va
l_accuracy: 0.0256 - val_loss: 0.1112
Epoch 12/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 133ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.0834
Epoch 13/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0048 - va
l_accuracy: 0.0256 - val_loss: 0.1075
Epoch 14/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 160ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.1137
Epoch 15/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 147ms/step - accuracy: 0.0000e+00 - loss: 0.0044 - va
l_accuracy: 0.0256 - val_loss: 0.1124
Epoch 16/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.1214
Epoch 17/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0043 - va
l_accuracy: 0.0256 - val_loss: 0.1325
Epoch 18/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0040 - va
l_accuracy: 0.0256 - val_loss: 0.1257
Epoch 19/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 137ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.1298
Epoch 20/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 2s 254ms/step - accuracy: 0.0000e+00 - loss: 0.0044 - va
l_accuracy: 0.0256 - val_loss: 0.1019
```

```
Epoch 21/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 217ms/step - accuracy: 0.0000e+00 - loss: 0.0051 - va
l_accuracy: 0.0256 - val_loss: 0.1289
Epoch 22/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 253ms/step - accuracy: 0.0000e+00 - loss: 0.0038 - va
l_accuracy: 0.0256 - val_loss: 0.1206
Epoch 23/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 145ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.1095
Epoch 24/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 135ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.1175
Epoch 25/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.1151
Epoch 26/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 150ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.1534
Epoch 27/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 143ms/step - accuracy: 0.0000e+00 - loss: 0.0048 - va
l_accuracy: 0.0256 - val_loss: 0.0952
Epoch 28/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 135ms/step - accuracy: 0.0000e+00 - loss: 0.0042 - va
l_accuracy: 0.0256 - val_loss: 0.0996
Epoch 29/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 136ms/step - accuracy: 0.0000e+00 - loss: 0.0053 - va
l_accuracy: 0.0256 - val_loss: 0.1661
Epoch 30/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 140ms/step - accuracy: 0.0000e+00 - loss: 0.0057 - va
l_accuracy: 0.0256 - val_loss: 0.0771
Epoch 31/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.1290
Epoch 32/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 166ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.1154
Epoch 33/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 220ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0862
Epoch 34/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 242ms/step - accuracy: 0.0000e+00 - loss: 0.0048 - va
l_accuracy: 0.0256 - val_loss: 0.1131
Epoch 35/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 170ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.0953
Epoch 36/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0039 - va
l_accuracy: 0.0256 - val_loss: 0.1399
Epoch 37/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0042 - va
l_accuracy: 0.0256 - val_loss: 0.0687
Epoch 38/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 157ms/step - accuracy: 0.0000e+00 - loss: 0.0054 - va
l_accuracy: 0.0256 - val_loss: 0.1571
Epoch 39/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 152ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0882
Epoch 40/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 152ms/step - accuracy: 0.0000e+00 - loss: 0.0039 - va
l_accuracy: 0.0256 - val_loss: 0.1360
```

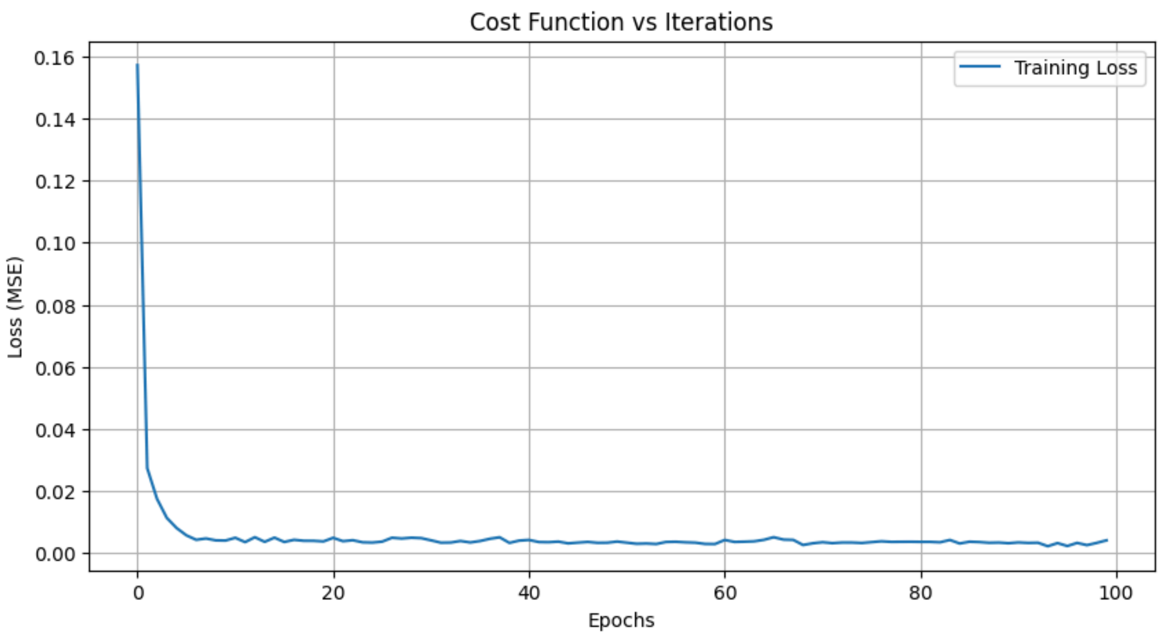
```
Epoch 41/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.1083
Epoch 42/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.1094
Epoch 43/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 2s 284ms/step - accuracy: 0.0000e+00 - loss: 0.0040 - va
l_accuracy: 0.0256 - val_loss: 0.1059
Epoch 44/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 155ms/step - accuracy: 0.0000e+00 - loss: 0.0038 - va
l_accuracy: 0.0256 - val_loss: 0.1056
Epoch 45/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 215ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1033
Epoch 46/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 218ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1380
Epoch 47/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 238ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.0830
Epoch 48/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.1023
Epoch 49/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 171ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1297
Epoch 50/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 148ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.0893
Epoch 51/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 140ms/step - accuracy: 0.0000e+00 - loss: 0.0032 - va
l_accuracy: 0.0256 - val_loss: 0.1102
Epoch 52/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 142ms/step - accuracy: 0.0000e+00 - loss: 0.0028 - va
l_accuracy: 0.0256 - val_loss: 0.0968
Epoch 53/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 135ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0889
Epoch 54/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0029 - va
l_accuracy: 0.0256 - val_loss: 0.1324
Epoch 55/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.0828
Epoch 56/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 137ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.1027
Epoch 57/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 170ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.1060
Epoch 58/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 2s 223ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.0803
Epoch 59/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 240ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.1064
Epoch 60/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.0682
```

```
Epoch 61/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 158ms/step - accuracy: 0.0000e+00 - loss: 0.0040 - va
l_accuracy: 0.0256 - val_loss: 0.1269
Epoch 62/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 180ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.0717
Epoch 63/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 155ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.1005
Epoch 64/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 172ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.1071
Epoch 65/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 149ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.0611
Epoch 66/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 148ms/step - accuracy: 0.0000e+00 - loss: 0.0062 - va
l_accuracy: 0.0256 - val_loss: 0.1267
Epoch 67/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0047 - va
l_accuracy: 0.0256 - val_loss: 0.0668
Epoch 68/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 147ms/step - accuracy: 0.0000e+00 - loss: 0.0043 - va
l_accuracy: 0.0256 - val_loss: 0.1003
Epoch 69/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 145ms/step - accuracy: 0.0000e+00 - loss: 0.0025 - va
l_accuracy: 0.0256 - val_loss: 0.0796
Epoch 70/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 187ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.1086
Epoch 71/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 219ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0835
Epoch 72/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 256ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.1182
Epoch 73/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 179ms/step - accuracy: 0.0000e+00 - loss: 0.0036 - va
l_accuracy: 0.0256 - val_loss: 0.0893
Epoch 74/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 143ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.0843
Epoch 75/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 135ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.1086
Epoch 76/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 156ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.0901
Epoch 77/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.1214
Epoch 78/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.0676
Epoch 79/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0038 - va
l_accuracy: 0.0256 - val_loss: 0.1096
Epoch 80/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 147ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.0891
```

```
Epoch 81/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 146ms/step - accuracy: 0.0000e+00 - loss: 0.0039 - va
l_accuracy: 0.0256 - val_loss: 0.0996
Epoch 82/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.1006
Epoch 83/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 140ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.0709
Epoch 84/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 188ms/step - accuracy: 0.0000e+00 - loss: 0.0047 - va
l_accuracy: 0.0256 - val_loss: 0.1142
Epoch 85/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 220ms/step - accuracy: 0.0000e+00 - loss: 0.0035 - va
l_accuracy: 0.0256 - val_loss: 0.0879
Epoch 86/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 243ms/step - accuracy: 0.0000e+00 - loss: 0.0039 - va
l_accuracy: 0.0256 - val_loss: 0.1092
Epoch 87/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 138ms/step - accuracy: 0.0000e+00 - loss: 0.0037 - va
l_accuracy: 0.0256 - val_loss: 0.0865
Epoch 88/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 154ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.1035
Epoch 89/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.0873
Epoch 90/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 159ms/step - accuracy: 0.0000e+00 - loss: 0.0030 - va
l_accuracy: 0.0256 - val_loss: 0.1020
Epoch 91/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 141ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0784
Epoch 92/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 146ms/step - accuracy: 0.0000e+00 - loss: 0.0034 - va
l_accuracy: 0.0256 - val_loss: 0.1238
Epoch 93/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 153ms/step - accuracy: 0.0000e+00 - loss: 0.0033 - va
l_accuracy: 0.0256 - val_loss: 0.0985
Epoch 94/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 143ms/step - accuracy: 0.0000e+00 - loss: 0.0022 - va
l_accuracy: 0.0256 - val_loss: 0.0951
Epoch 95/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 140ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1170
Epoch 96/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 175ms/step - accuracy: 0.0000e+00 - loss: 0.0026 - va
l_accuracy: 0.0256 - val_loss: 0.0748
Epoch 97/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 2s 245ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1113
Epoch 98/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 244ms/step - accuracy: 0.0000e+00 - loss: 0.0027 - va
l_accuracy: 0.0256 - val_loss: 0.0744
Epoch 99/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 134ms/step - accuracy: 0.0000e+00 - loss: 0.0031 - va
l_accuracy: 0.0256 - val_loss: 0.1246
Epoch 100/100
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.0000e+00 - loss: 0.0041 - va
l_accuracy: 0.0256 - val_loss: 0.0660
```

In [16]:
```
# 6.    Observe the cost function vs iterations learning curve

# Plot cost function (loss) vs iterations (epochs)
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.title('Cost Function vs Iterations')
plt.xlabel('Epochs')
plt.ylabel('Loss (MSE)')
plt.legend()
plt.grid(True)
plt.show()
```




Result

In [17]:
```
# a.    Model Summary

model.summary()
```

**Model: "sequential_1"**

| Layer (type) | Output Shape |
|---|---|
| lstm_4 (LSTM) | (None, 60, 60) |
| dropout_4 (Dropout) | (None, 60, 60) |
| lstm_5 (LSTM) | (None, 60, 60) |
| dropout_5 (Dropout) | (None, 60, 60) |
| lstm_6 (LSTM) | (None, 60, 60) |
| dropout_6 (Dropout) | (None, 60, 60) |
| lstm_7 (LSTM) | (None, 60) |
| dropout_7 (Dropout) | (None, 60) |
| dense_1 (Dense) | (None, 1) |

**Total params:** 306,185 (1.17 MB)

**Trainable params:** 102,061 (398.68 KB)

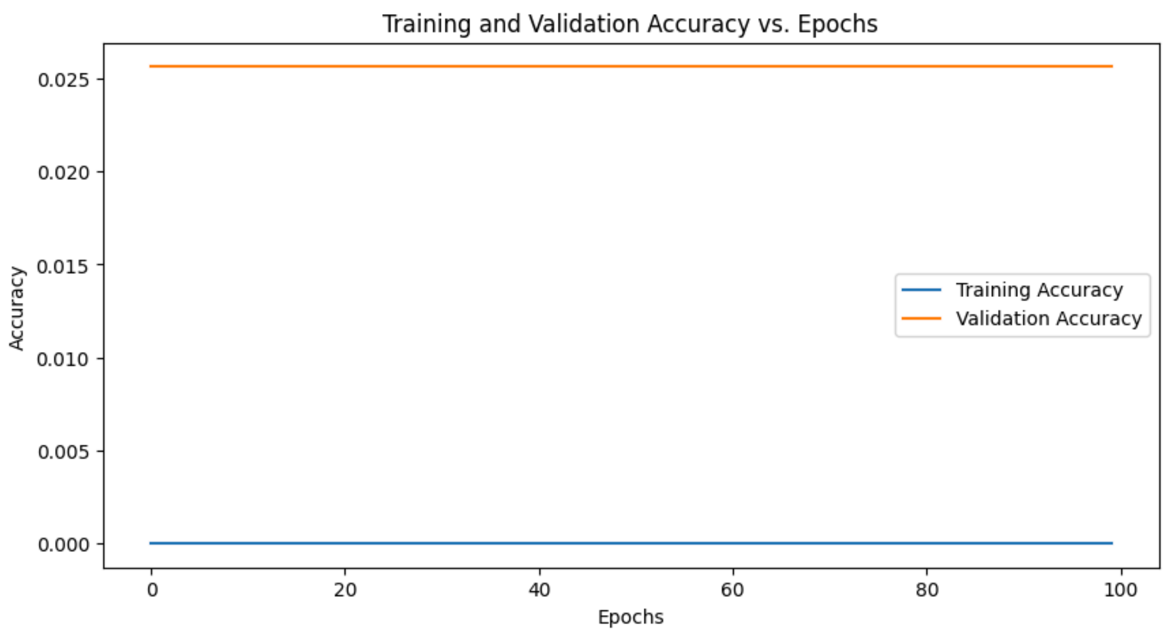
**Non-trainable params:** 0 (0.00 B)

**Optimizer params:** 204,124 (797.36 KB)

In [18]:

```
# b.    Training and Validation accuracy v/s epochs

# Plot training and validation accuracy
plt.figure(figsize=(10, 5))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy vs. Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```



In [19]:
```python
# c.    Training and Validation loss v/s epochs

# Plot training and validation loss
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss vs. Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```


In [20]:
```python
# d. Visualize the Predicted and originalStock Price

# Prepare testing data
testing_data = dataset.iloc[:, 1:2].values
test_feature = testing_data.reshape(-1, 1)
test_feature = scaler.transform(test_feature)

testing_features = []

for i in range(60, 253):
    testing_features.append(test_feature[i-60:i, 0])

testing_features = np.array(testing_features)
testing_features = np.reshape(testing_features, (testing_features.shape[0], test

# Predict the stock prices
predictions = model.predict(testing_features)
predictions = scaler.inverse_transform(predictions)

# Visualize the results
plt.figure(figsize=(12, 6))
plt.plot(testing_data, color='blue', label='Real Apple Stock Price')
plt.plot(predictions, color='red', label='Predicted Apple Stock Price')
plt.title('Real vs. Predicted Apple Stock Price')
plt.xlabel('Time')
plt.ylabel('Stock Price')
```

```
plt.legend()
plt.show()
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 2s 287ms/step
```