```
!pip install keras-tuner
```

```
Collecting keras-tuner
  Downloading keras_tuner-1.4.7-py3-none-any.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.1/129.1 kB 4.7 MB/s eta 0:00:00
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (from keras-tuner) (2.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from keras-tuner) (24.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from keras-tuner) (2.31.0)
Collecting kt-legacy (from keras-tuner)
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (3.3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->keras-tuner) (2024.2.2
Installing collected packages: kt-legacy, keras-tuner
Successfully installed keras-tuner-1.4.7 kt-legacy-1.0.5
```

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns

from tensorflow import keras
from tensorflow.keras import layers
from kerastuner.tuners import RandomSearch
```

```
<ipython-input-3-7d202ea570ea>:9: DeprecationWarning: `import kerastuner` is deprecated, please use `import keras_tuner`.
  from kerastuner.tuners import RandomSearch
```

```
df= pd.read_csv('/content/S&P dataset.csv')
```

```
df.describe
```

```
pandas.core.generic.NDFrame.describe
def describe(percentiles=None, include=None, exclude=None) -> NDFrameT

Generate descriptive statistics.

Descriptive statistics include those that summarize the central
tendency, dispersion and shape of a
dataset's distribution, excluding ``NaN`` values.
```

```
df.tail()
```

	Date	Open	High	Low	Close	Adj Close	Volum
5212	2020-09-21	3285.570068	3285.570068	3229.100098	3281.060059	3281.060059	482835000
5213	2020-09-22	3295.750000	3320.310059	3270.949951	3315.570068	3315.570068	396330000
5214	2020-09-23	3320.110107	3323.350098	3232.570068	3236.919922	3236.919922	436450000

```
df["Date"] = pd.to_datetime(df["Date"])
```

```
df = df.set_index("Date")
print(df.shape)
print(df.columns)
```

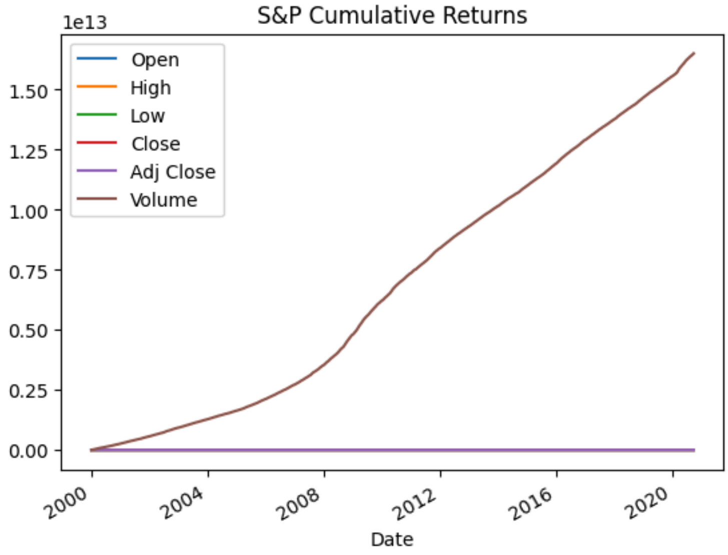
```
(5217, 6)
Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

```
df.head(5)
```

Date	Open	High	Low	Close	Adj Close	Volume
2000-01-03	1469.250000	1478.000000	1438.359985	1455.219971	1455.219971	931800000
2000-01-04	1455.219971	1455.219971	1397.430054	1399.420044	1399.420044	1009000000
2000-01-05	1399.420044	1413.270020	1377.680054	1402.109985	1402.109985	1085500000
2000-01-06	1402.109985	1411.900024	1392.099976	1403.449951	1403.449951	1092300000
2000-01-07	1403.449951	1441.469971	1400.729980	1441.469971	1441.469971	1225200000

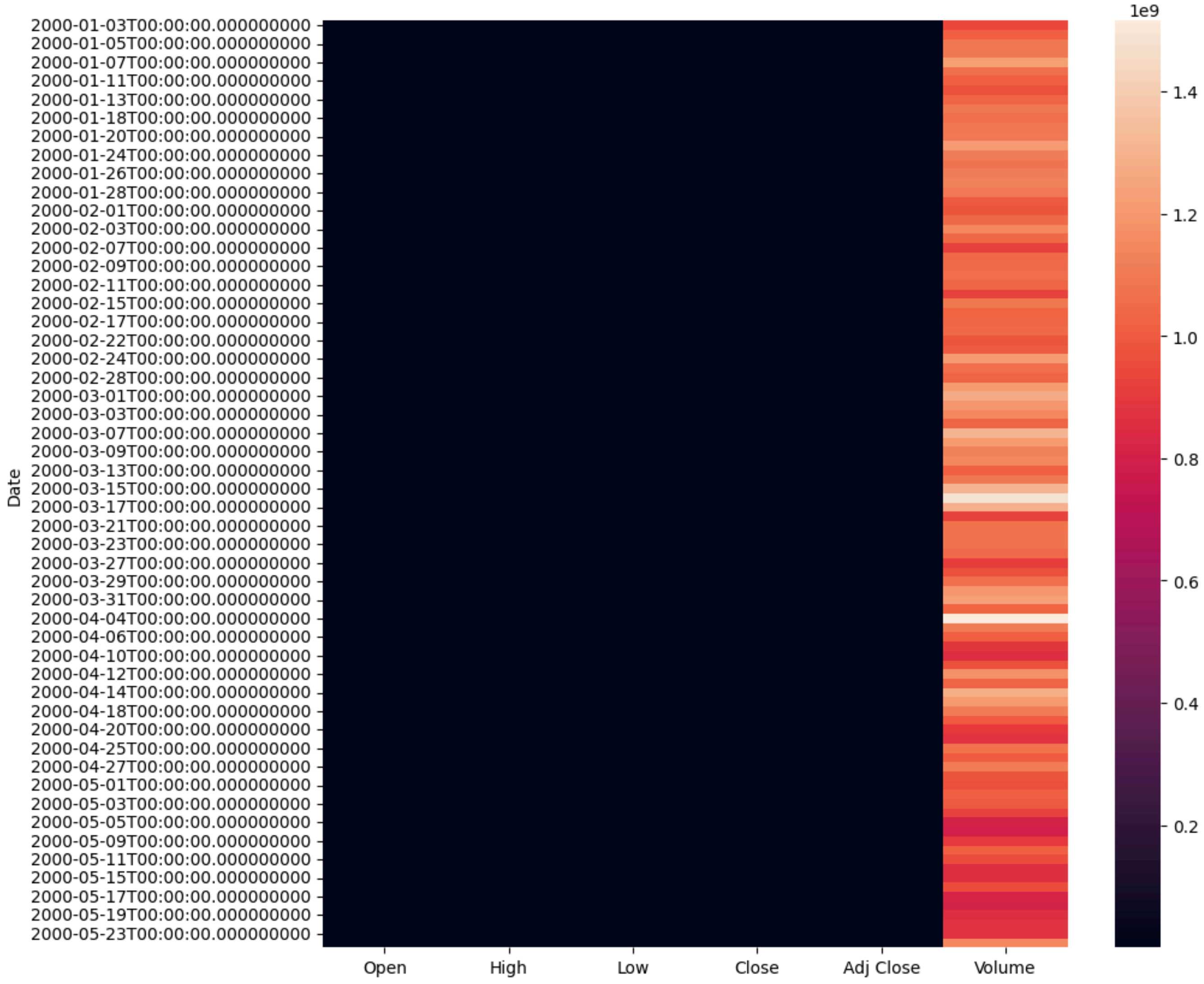
```
plt.figure(figsize=(20,20))
dr = df.cumsum()
dr.plot()
plt.title('S&P Cumulative Returns')
```

```
Text(0.5, 1.0, 'S&P Cumulative Returns')
<Figure size 2000x2000 with 0 Axes>
```



```
plt.figure(figsize=(10,10))
sns.heatmap(df[:100],   robust=False,
                annot=None, fmt='.2g', annot_kws=None, linewidths=0, linecolor='white', cbar=True,
                 square=False, xticklabels='auto', yticklabels='auto', mask=None, ax=None)
```

```
<Axes: ylabel='Date'>
```



```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
df=scaler.fit_transform(df)
```

```
df[0]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-ad11118bc8f3> in <cell line: 1>()
----> 1 df[0]

NameError: name 'df' is not defined
```

```
import numpy as np
def create_dataset(dataset,time_stamp =1):
  X, Y = [], []
  for i in range(len(dataset)-time_stamp-1):
    a= dataset[i:(i+time_stamp),0]
    X.append(a)
    Y.append(df[i+time_stamp,0])
  return np.array(X),np.array(Y)
```

```
train=df[0:3000]
test =df[3500:]
```

```
time_stamp=100
x_train, y_train=create_dataset(train,time_stamp)
x_test, y_test = create_dataset(test, time_stamp)
```

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

```
print(x_test)
print(x_train)
print(x_train.shape)
```

```
[[0.39067256 0.38843719 0.38602857 ... 0.41589207 0.41681048 0.41533757]
 [0.38843719 0.38602857 0.38591418 ... 0.41681048 0.41533757 0.41092928]
 [0.38602857 0.38591418 0.38436851 ... 0.41533757 0.41092928 0.41293243]
 ...
 [0.78033655 0.75891195 0.74016968 ... 0.92449042 0.9281362  0.90324942]
 [0.75891195 0.74016968 0.7588391  ... 0.9281362  0.90324942 0.90677743]
 [0.74016968 0.7588391  0.76378113 ... 0.90324942 0.90677743 0.9152198 ]]
[[0.2737761  0.26891378 0.24957547 ... 0.25218507 0.25002598 0.24071724]
 [0.26891378 0.24957547 0.25050771 ... 0.25002598 0.24071724 0.24944724]
 [0.24957547 0.25050771 0.25097209 ... 0.24071724 0.24944724 0.24337194]
 ...
 [0.23327649 0.2301297  0.22191608 ... 0.17612442 0.16708948 0.1661399 ]
 [0.2301297  0.22191608 0.22012779 ... 0.16708948 0.1661399  0.17788499]
 [0.22191608 0.22012779 0.221268   ... 0.1661399  0.17788499 0.17932668]]
(2899, 100)
```

```
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)
```

```
print(x_test.shape)
print(y_test.shape)
```

```
(1616, 100, 1)
(1616,)
```

```
print(x_train.shape)
print(y_train.shape)
```

```
(2899, 100, 1)
(2899,)
```

```
x_train
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-80784060c951> in <cell line: 1>()
----> 1 x_train

NameError: name 'x_train' is not defined
```

```
model=Sequential()
model.add(LSTM(100,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(100,return_sequences=True))
model.add(LSTM(50,return_sequences=False))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

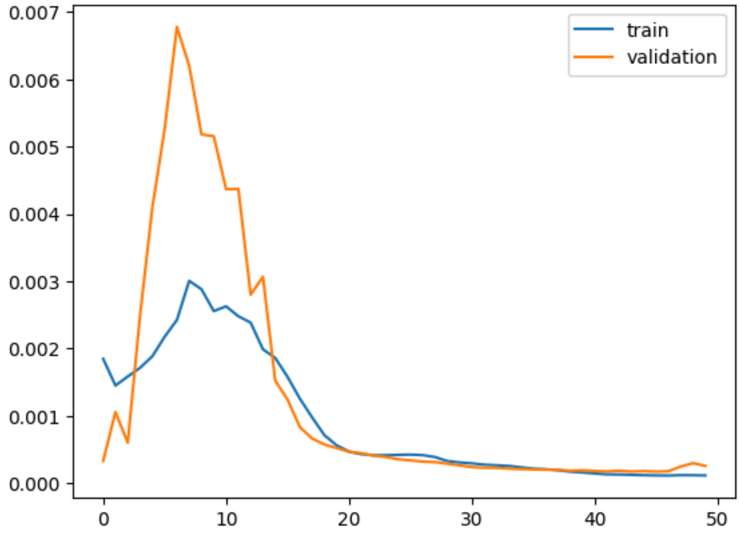
```
history=model.fit(
    x_train,y_train,
    validation_split=0.1,
    shuffle=False,
    epochs=50,batch_size=16,verbose=1)
```

```
Epoch 1/50
164/164 [==============================] - 11s 21ms/step - loss: 0.0018 - val_loss: 3.3245e-04
Epoch 2/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0015 - val_loss: 0.0011
Epoch 3/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0016 - val_loss: 6.0106e-04
Epoch 4/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0017 - val_loss: 0.0025
Epoch 5/50
164/164 [==============================] - 3s 19ms/step - loss: 0.0019 - val_loss: 0.0041
Epoch 6/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0022 - val_loss: 0.0053
Epoch 7/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0024 - val_loss: 0.0068
Epoch 8/50
```

```
164/164 [==============================] - 2s 14ms/step - loss: 0.0030 - val_loss: 0.0062
Epoch 9/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0029 - val_loss: 0.0052
Epoch 10/50
164/164 [==============================] - 3s 17ms/step - loss: 0.0026 - val_loss: 0.0052
Epoch 11/50
164/164 [==============================] - 2s 15ms/step - loss: 0.0026 - val_loss: 0.0044
Epoch 12/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0025 - val_loss: 0.0044
Epoch 13/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0024 - val_loss: 0.0028
Epoch 14/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0020 - val_loss: 0.0031
Epoch 15/50
164/164 [==============================] - 3s 16ms/step - loss: 0.0019 - val_loss: 0.0015
Epoch 16/50
164/164 [==============================] - 3s 17ms/step - loss: 0.0016 - val_loss: 0.0012
Epoch 17/50
164/164 [==============================] - 2s 14ms/step - loss: 0.0013 - val_loss: 8.3607e-04
Epoch 18/50
164/164 [==============================] - 2s 14ms/step - loss: 9.7914e-04 - val_loss: 6.6515e-04
Epoch 19/50
164/164 [==============================] - 2s 14ms/step - loss: 7.1501e-04 - val_loss: 5.7584e-04
Epoch 20/50
164/164 [==============================] - 2s 14ms/step - loss: 5.6328e-04 - val_loss: 5.2350e-04
Epoch 21/50
164/164 [==============================] - 3s 19ms/step - loss: 4.7055e-04 - val_loss: 4.6866e-04
Epoch 22/50
164/164 [==============================] - 2s 14ms/step - loss: 4.3233e-04 - val_loss: 4.4724e-04
Epoch 23/50
164/164 [==============================] - 2s 14ms/step - loss: 4.1585e-04 - val_loss: 4.1016e-04
Epoch 24/50
164/164 [==============================] - 2s 14ms/step - loss: 4.1638e-04 - val_loss: 3.9031e-04
Epoch 25/50
164/164 [==============================] - 2s 14ms/step - loss: 4.2185e-04 - val_loss: 3.5682e-04
Epoch 26/50
164/164 [==============================] - 3s 18ms/step - loss: 4.2552e-04 - val_loss: 3.3958e-04
Epoch 27/50
164/164 [==============================] - 3s 15ms/step - loss: 4.1776e-04 - val_loss: 3.2206e-04
Epoch 28/50
164/164 [==============================] - 2s 14ms/step - loss: 3.8970e-04 - val_loss: 3.1486e-04
Epoch 29/50
164/164 [==============================] - 2s 14ms/step - loss: 3.3008e-04 - val_loss: 2.9112e-04
```

```
plt.plot(history.history['loss'],label='train')
plt.plot(history.history['val_loss'],label='validation')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7df96385d120>
```



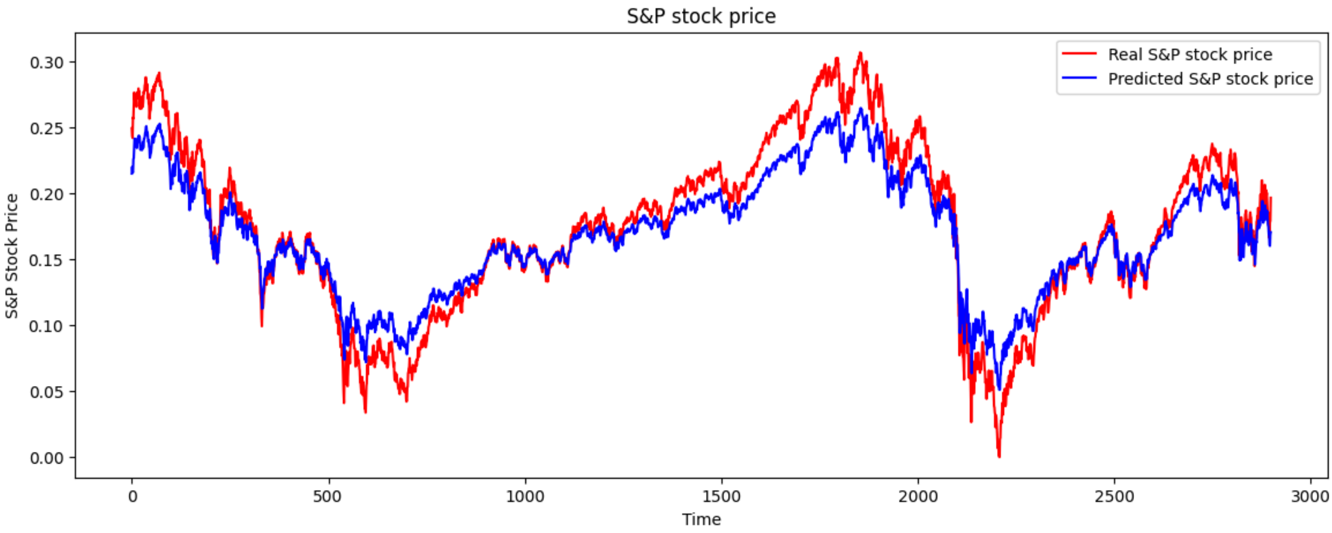
```
train_predict=model.predict(x_train)
test_predict=model.predict(x_test)
```

```
91/91 [==============================] - 3s 9ms/step
51/51 [==============================] - 0s 6ms/step
```

```
import math
from sklearn.metrics import mean_squared_error, precision_score,recall_score,f1_score
math.sqrt(mean_squared_error(y_train,train_predict))
from sklearn.metrics import confusion_matrix
x=confusion_matrix=(x_test, model.predict(x_test))
```

```
51/51 [==============================] - 1s 10ms/step
```

```
plt.figure(figsize=(14,5))
plt.plot(y_train, color = 'red', label = 'Real S&P stock price')
plt.plot(train_predict, color = 'blue', label = 'Predicted S&P stock price')
plt.title('S&P stock price')
plt.xlabel('Time')
plt.ylabel('S&P Stock Price')
plt.legend()
plt.show()
```



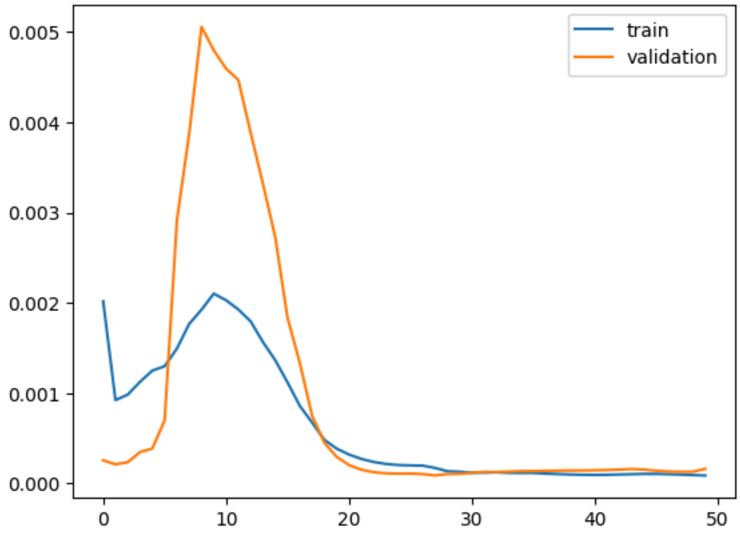
```
model=Sequential()
model.add(tf.keras.layers.GRU(100,return_sequences=True,input_shape=(100,1)))
model.add(tf.keras.layers.GRU(50,return_sequences=False))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

```
history=model.fit(
    x_train,y_train,
    validation_split=0.1,
    shuffle=False,
    epochs=50,batch_size=16,verbose=1)
```

```
Epoch 39/50
164/164 [==============================] - 2s 10ms/step - loss: 9.9563e-05 - val_loss: 1.4387e-04
Epoch 40/50
164/164 [==============================] - 2s 10ms/step - loss: 9.6286e-05 - val_loss: 1.4446e-04
Epoch 41/50
164/164 [==============================] - 2s 10ms/step - loss: 9.4956e-05 - val_loss: 1.4643e-04
Epoch 42/50
164/164 [==============================] - 2s 11ms/step - loss: 9.6219e-05 - val_loss: 1.4951e-04
Epoch 43/50
164/164 [==============================] - 2s 14ms/step - loss: 9.9368e-05 - val_loss: 1.5435e-04
Epoch 44/50
164/164 [==============================] - 2s 10ms/step - loss: 1.0306e-04 - val_loss: 1.6050e-04
Epoch 45/50
164/164 [==============================] - 2s 10ms/step - loss: 1.0623e-04 - val_loss: 1.5430e-04
Epoch 46/50
164/164 [==============================] - 2s 10ms/step - loss: 1.0621e-04 - val_loss: 1.4150e-04
Epoch 47/50
164/164 [==============================] - 2s 10ms/step - loss: 1.0293e-04 - val_loss: 1.3306e-04
Epoch 48/50
164/164 [==============================] - 2s 10ms/step - loss: 9.8424e-05 - val_loss: 1.2921e-04
Epoch 49/50
164/164 [==============================] - 2s 12ms/step - loss: 9.3454e-05 - val_loss: 1.3048e-04
Epoch 50/50
164/164 [==============================] - 2s 14ms/step - loss: 8.8171e-05 - val_loss: 1.6270e-04
```

```
plt.plot(history.history['loss'],label='train')
plt.plot(history.history['val_loss'],label='validation')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7df963594ac0>
```



```
train_predict=model.predict(x_train)
test_predict=model.predict(x_test)
```

```
91/91 [==============================] - 2s 8ms/step
51/51 [==============================] - 0s 6ms/step
```

```
plt.figure(figsize=(14,5))
plt.plot(y_train, color = 'red', label = 'Real S&P stock price')
plt.plot(train_predict, color = 'blue', label = 'Predicted S&P stock price')
plt.title('S&P stock price')
plt.xlabel('Time')
plt.ylabel('S&P Stock Price')
plt.legend()
plt.show()
```