Source Code

```
#!/usr/bin/env python

# coding: utf-8


# In[1]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime


# In[2]:
import yfinance as yf
import datetime as dt
from datetime import date
START = "2019-01-01"
TODAY = date.today().strftime("%Y-%m-%d")


# Define a function to load the dataset
def load_data(ticker):data = yf.download(ticker, START, TODAY)
data.reset_index(inplace=True)
                data.columns = [col[0] for col in data.columns.values]
                data.index=data.Date
return data
```

```
# In[3]:
data=load_data('AAPL') data

# In[4]:
tech_list = ['AAPL', 'UBER','TSLA', 'MSFT']
end = datetime.now()
start = datetime(end.year - 5, end.month, end.day)
for stock in tech_list: globals()[stock] = yf.download(stock, start,
end)
company_list = [AAPL, UBER,TSLA, MSFT] company_name = ["APPLE",
"UBER","TESLA", "MICROSOFT"]
for company, com_name in zip(company_list, company_name):
company["company_name"] = com_name
df = pd.concat(company_list, axis=0) df.tail(10)

# In[5]:
from alpha_vantage.timeseries import TimeSeries import pandas as pd
# api_key='6OHUH4FO2W5KVR9Y'
api_key='AZT6ECSIS1ORUDMO'
def load_data(ticker):
ts = TimeSeries(key=api_key, output_format="pandas") START =
"2019-01-01"
# Get daily stock data
data, meta_data = ts.get_daily(symbol=ticker, outputsize="full")
data = data.rename(columns={"1. open": "Open","2. high": "High","3.
```

```
low": "Low","4. close": "Close","5. volume": "Volume"})
data = data[::-1]
date_data = data.loc[START:'2025-02-11'] return date_data

# In[6]:
tech_list = ['AAPL', 'UBER','TSLA', 'MSFT']
for stock in tech_list: globals()[stock] = load_data(stock)
company_list = [AAPL, UBER,TSLA, MSFT] company_name = ["APPLE",
"UBER","TESLA", "MICROSOFT"]for company, com_name in
zip(company_list, company_name): company["company_name"] = com_name
df = pd.concat(company_list, axis=0)
df.tail(10)

# In[7]:
df['company_name'].value_counts()

# In[8]:
AAPL

# In[9]:
TSLA

# In[10]:
UBER
```

```
# In[11]:
AAPL.describe()

# In[12]:
AAPL.info()

# In[13]:
# Let's see a historical view of the closing price
plt.figure(figsize=(15, 10)) plt.subplots_adjust(top=1.25,
bottom=1.2)
for i, company in enumerate(company_list, 1):plt.subplot(2, 2, i)
company['Close'].plot() plt.ylabel('Close') plt.xlabel(None)
plt.title(f"Closing Price of {tech_list[i - 1]}")
plt.tight_layout()

# In[14]:
# Now let's plot the total volume of stock being traded each day
plt.figure(figsize=(15, 10))
plt.subplots_adjust(top=1.25, bottom=1.2)
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Volume'].plot()
plt.ylabel('Volume') plt.xlabel(None)
plt.title(f"Sales Volume for {tech_list[i - 1]}")
plt.tight_layout()
```

```
# In[15]:
company_list

# In[16]:
UBER[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]

# In[17]:
ma_day = [10, 20, 50]
for ma in ma_day:
for company in company_list: column_name = f"MA for {ma}
days"
company[column_name] = company['Close'].rolling(ma).mean()
fig, axes = plt.subplots(nrows=2, ncols=2) fig.set_figheight(10) fig.set_figwidth(15)
AAPL[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50
days']].plot(ax=axes[0,0])

axes[0,0].set_title('APPLE')UBER[['Close', 'MA for 10 days', 'MA for 20 days',
'MA for 50 days']].plot(ax=axes[0,1])

axes[0,1].set_title('UBER')TSLA[['Close', 'MA for 10 days', 'MA for 20 days', 'MA
for 50 days']].plot(ax=axes[1,0])

axes[1,0].set_title('TESLA')
MSFT[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50
days']].plot(ax=axes[1,1])

axes[1,1].set_title('MICROSOFT')
```

```
fig.tight_layout()
# In[18]:
# We'll use pct_change to find the percent change for each day for
company in company_list:company['Daily Return'] =
company['Close'].pct_change()
# Then we'll plot the daily return percentage fig, axes =
plt.subplots(nrows=2, ncols=2) fig.set_figheight(10)
fig.set_figwidth(15)
AAPL['Daily Return'].plot(ax=axes[0,0], legend=True, linestyle='--',
marker='o')

axes[0,0].set_title('APPLE')
UBER['Daily Return'].plot(ax=axes[0,1], legend=True, linestyle='--', marker='o')

axes[0,1].set_title('UBER')
MSFT['Daily Return'].plot(ax=axes[1,0], legend=True, linestyle='--', marker='o')

axes[1,0].set_title('MICROSOFT')
TSLA['Daily Return'].plot(ax=axes[1,1], legend=True, linestyle='--',
marker='o')

axes[1,1].set_title('TESLA')
fig.tight_layout()


# In[19]:
plt.figure(figsize=(12, 9))
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Daily Return'].hist(bins=50)
```

```
plt.xlabel('Daily Return') plt.ylabel('Counts')
plt.title(f'{company_name[i - 1]}')
plt.tight_layout()

# In[20]:
data=load_data("AMZN") data

# In[21]:
# data.columns = [col[0] for col in data.columns.values]

# In[22]:
data.index=data.Date

# In[23]:
df=data.copy()

# In[24]:
print("\nBasic Information:") df.info()

# In[25]:
print("\nSummary Statistics:") df.describe()

# In[26]:
print("\nMissing Values:") df.isnull().sum()

# In[27]:
```

```
print("\nChecking for Duplicate Values:") df.duplicated().sum()


# In[28]:

import plotly.express as px

import plotly.graph_objects as go

import plotly.io as pio

def plot_closing_price(df):

"""Plots closing price over time using Plotly."""

fig = px.line(df, x=df.index, y='Close', title='Stock Closing Price Over Time',

labels={'Close': 'Price'})

fig.show()

def plot_moving_averages(df):

"""Plots closing price with moving averages."""  df['50_MA'] =

df['Close'].rolling(window=50).mean()  df['200_MA'] =

df['Close'].rolling(window=200).mean()

fig = go.Figure()

fig.add_trace(go.Scatter(x=df.index, y=df['Close'], mode='lines',

name='Closing Price'))

fig.add_trace(go.Scatter(x=df.index, y=df['50_MA'], mode='lines',

name='50-Day MA'))

fig.add_trace(go.Scatter(x=df.index, y=df['200_MA'], mode='lines',

name='200-Day MA'))

fig.update_layout(title='Stock Price with Moving Averages', xaxis_title='Date',
```

```
yaxis_title='Price')
fig.show()
def plot_closing_price_distribution(df): """Plots distribution of
closing prices."""
fig = px.histogram(df, x='Close', nbins=50, title='Distribution of Closing Prices',
labels={'Close': 'Price'})
fig.show()
def plot_daily_returns_distribution(df): """Plots distribution of daily
returns."""" df['Daily Return'] = df['Close'].pct_change()
fig = px.histogram(df, x='Daily Return', nbins=50, title='Distribution of Daily
Returns', labels={'Daily Return': 'Return'})
fig.show()

# In[29]:
plot_closing_price(data)

# In[30]:
plot_moving_averages(data)

# In[31]:
plot_closing_price_distribution(data)

# In[32]:
plot_daily_returns_distribution(data)
```

```
# In[33]:
def plot_candlestick(df):
"""Plots a candlestick chart for stock price movements.""" fig =
go.Figure(data=[go.Candlestick(x=df.index,
open=df['Open'],
 high=df['High'],
  low=df['Low'], close=df['Close'],
  name='Candlestick')])
  fig.update_layout(title='Stock Price Candlestick Chart',
  xaxis_title='Date',
  yaxis_title='Price')
  fig.show()

# In[34]:
plot_candlestick(data)
data.to_csv('dataaset.csv',index=False)
#!/usr/bin/env python
# coding: utf-8
# #### Import and install necessary packages

# In[1]:
# !pip install pmdarima
# !pip install statsmodels

# In[2]:
```

```
#!pip install pmdarima

# In[3]:
Import warnings warnings.filterwarnings('ignore') import seaborn
as sns
import numpy as np import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight') from pylab import rcParams
rcParams['figure.figsize'] = 10, 6
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose #from prophet
import Prophet
from statsmodels.tsa.statespace.sarimax import SARIMAX from
statsmodels.tsa.arima_model import ARIMA
from pmdarima.arima import auto_arima from sklearn.preprocessing
import MinMaxScaler
# import pandas_profiling
from sklearn.metrics import mean_squared_error, mean_absolute_error
import math
from math import ceil, floor
# # Data Import and Preprocessing ## --- # #

# In[4]:
# #Import the dataset
 # data = pd.read_csv('/kaggle/input/nse-tataglobal/NSE-TATAGLOBAL.csv') #
```

```
data['Date'] = pd.to_datetime(data.Date,format='%Y/%m/%d')

# In[5]:
# !pip show yfinance # # Name: yfinance # #
Version: 0.2.52

# In[6]:
# !pip install yfinance==0.2.41 # !pip install
yfinance==0.2.52

# In[7]:
import yfinance as yf import datetime as dt
from datetime import date from datetime import datetime

# In[8]:
def load_data(ticker):
from datetime import date START = "2010-01-01"
TODAY = date.today().strftime("%Y-%m-%d") data =
yf.download(ticker, START, TODAY)
data.reset_index(inplace=True)
data.columns = [col[0] for col in data.columns.values]
data.index=data.Date
return data

# In[9]:
```

```
# Set up End and Start times for data grab tech_list = ['AAPL',
'GOOG', 'MSFT', 'AMZN']
# end = datetime.now()
# start = datetime(end.year - 5, end.month, end.day)
for stock in tech_list: globals()[stock] = load_data(stock)
 company_list = [AAPL, GOOG, MSFT, AMZN] company_name ["APPLE",
 "GOOGLE", "MICROSOFT", "AMAZON"]
for company, com_name in zip(company_list, company_name):
company["company_name"] = com_name
df = pd.concat(company_list, axis=0)
df.tail(10)

# In[10]:
df['company_name'].value_counts()

# In[11]:
# Let's see a historical view of the closing price plt.figure(figsize=(15,
10)) plt.subplots_adjust(top=1.25, bottom=1.2)
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Close'].plot(linewidth=1.5)
plt.ylabel('Close') plt.xlabel(None)
plt.title(f"Closing Price of {tech_list[i - 1]}")
plt.tight_layout()

# In[12]:
```

```
# Now let's plot the total volume of stock being traded each day
plt.figure(figsize=(15, 10))
plt.subplots_adjust(top=1.25, bottom=1.2)
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Volume'].plot(linewidth=1.5)
plt.ylabel('Volume')
plt.xlabel(None)
plt.title(f"Sales Volume for {tech_list[i - 1]}")
plt.tight_layout()


# In[13]:
ma_day = [10, 20, 50]
for ma in ma_day:
for company in company_list: column_name = f"MA for {ma}
days"
company[column_name] = company['Close'].rolling(ma).mean()
fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)
fig.set_figwidth(15)AAPL[['Close', 'MA for 10 days',
'MA for 20 days', 'MA for 50
days']].plot(ax=axes[0,0],linewidth=1.5)
axes[0,0].set_title('APPLE')
GOOG[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50
days']].plot(ax=axes[0,1],linewidth=2.5)
```

```
axes[0,1].set_title('GOOGLE')
MSFT[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50
days']].plot(ax=axes[1,0],linewidth=2)

axes[1,0].set_title('MICROSOFT')
AMZN[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50
days']].plot(ax=axes[1,1],linewidth=3)

axes[1,1].set_title('AMAZON')
fig.tight_layout()


# In[14]:
        # We'll use pct_change to find the percent change for each day for
        company in company_list:
company['Daily Return'] = company['Close'].pct_change()
                    # Then we'll plot the daily return percentage
                    fig, axes = plt.subplots(nrows=2, ncols=2)
                    fig.set_figheight(10)
                fig.set_figwidth(15)
      AAPL['Daily Return'].plot(ax=axes[0,0], legend=True, linestyle='--',
      marker='o',linewidth=1.5)

axes[0,0].set_title('APPLE')
     GOOG['Daily Return'].plot(ax=axes[0,1], legend=True, linestyle='--',
     marker='o',linewidth=1.5)

axes[0,1].set_title('GOOGLE')
```

```
MSFT['Daily Return'].plot(ax=axes[1,0], legend=True, linestyle='--',
marker='o',linewidth=1.5)

                    axes[1,0].set_title('MICROSOFT')
AMZN['Daily Return'].plot(ax=axes[1,1], legend=True, linestyle='--',
marker='o',linewidth=1.5)

axes[1,1].set_title('AMAZON')
fig.tight_layout()


# In[15]:
plt.figure(figsize=(12, 9))
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i)
company['Daily Return'].hist(bins=50) plt.xlabel('Daily
Return') plt.ylabel('Counts') plt.title(f'{company_name[i -
1]}')plt.tight_layout()


# In[16]:
START = "2010-01-01"
TODAY = date.today().strftime("%Y-%m-%d")print(TODAY)
# Define a function to load the dataset
def load_data(ticker):
data = yf.download(ticker, START, TODAY)
data.reset_index(inplace=True)
return data
```

data = load_data('NFLX') data

In[17]:

data.columns = [col[0] for col in data.columns.values]

In[18]:

#Reverse the order of the dataset so that the latest year data will arrive in the tail of the dataframe

data = data.iloc[::-1]

In[19]:

#View the dataset data

In[20]:

data.index=data.Date

In[21]:

#Use date column as the index

#data.reset_index(inplace=True)

In[22]:

data

In[23]:

#Get the overview of the dataset

pandas_profiling.ProfileReport(data)

From the overview above, we can see that the data is clean and does not need any serious cleaning. There are no missing values or any duplicate. Let **us** move forward

In[24]:

#Plot all the variables against the Date and check for the relationships, patterns and trends for column in data.columns[2:]: plt.figure(figsize=(10, 6))
plt.plot(data['Date'], data[column],linewidth=1.5) plt.xlabel('Date')
plt.ylabel(column)
plt.title(f'{column} vs. Date') plt.tight_layout()
plt.show()

From above plot, it can be seen that the prices, turnover and total trade quantity is increasing from 2017

Checking Stationarity

In[25]:

Define a function which will give us rolling mean and standard deviation and perform ADF Test

def test_stationarity(timeseries): #Determing rolling statistics
rolmean = timeseries.rolling(12).mean() rolstd =
timeseries.rolling(12).std() #Plot rolling statistics:
plt.plot(timeseries, color='blue',label='Original') plt.plot(rolmean, color='red', label='Rolling Mean',linewidth=2.5)
plt.plot(rolstd, color='black', label = 'Rolling Std',linewidth=1.5)

```
plt.legend(loc='best')
plt.title('Rolling Mean and Standard Deviation')
plt.show(block=False)
print("Results of dickey fuller test") adft =
adfuller(timeseries,autolag='AIC')
# output for dft will give us without defining what the values are.
#hence we manually write what values does it explains using a for loop
output = pd.Series(adft[0:4],index=['Test Statistics','p-value','No. of lags
used','Number of observations used'])

for key,values in adft[4].items(): output['critical value (%s)'%key] =
values
print(output)
#


# In[26]:
test_stationarity(data['Open'])
# Through the above graph, we can see the increasing mean and standard
deviation and hence our series is not stationary.
#
# We see that the p-value is greater than 0.05 so we cannot reject the Null
hypothesis. Also, the test statistics is greater than the critical values. so the data
is non-stationary.

#
# In order to perform a time series analysis, we may need to separate seasonality
and trend from our series. The resultant series will become stationary through this
```

process.

#

So let us separate Trend and Seasonality from the time series.

In[27]:

Decompose the series to separate the trend and the seasonality from a time series,

result = seasonal_decompose(data['Open'], model='multiplicative', period = 30)

fig = plt.figure() fig = result.plot()
fig.set_size_inches(16, 9)

In[28]:

def test_stationarity(timeseries):
# Transformation to make the data stationary timeseries_log =
np.log(timeseries) timeseries_log_diff =
timeseries_log.diff().dropna()# Determing rolling statistics
rolmean = timeseries_log_diff.rolling(window=12).mean() rolstd =
timeseries_log_diff.rolling(window=12).std()
# Plot rolling statistics plt.figure(figsize=(10, 6))
plt.plot(timeseries_log_diff, color='blue', label='Original')
plt.plot(rolmean, color='red', label='Rolling Mean') plt.plot(rolstd,
color='black', label='Rolling Std') plt.legend(loc='best')
plt.title('Rolling Mean and Standard Deviation') plt.xlabel('Date')
plt.ylabel('Log-Difference') plt.show()

```
# Perform ADF test
adft = adfuller(timeseries_log_diff, autolag='AIC')
# Output the ADF test results
output = pd.Series(adft[0:4], index=['Test Statistic', 'p-value', 'Number of Lags Used',
'Number of Observations Used'])for key, value in adft[4].items(): output[f'Critical
Value ({key})'] = value
print(output)

# In[29]:
test_stationarity(data['Open'])
# Based on the ADF test results above, the test statistic is significantly lower than
the critical values at all levels of significance (1%, 5%, and 10%).
Additionally, the p-value is very close to zero, indicating strong evidence against
the null hypothesis of non-stationarity.
#
# Therefore, based on these results, we can conclude that the time series is
stationary after the transformation.
# # Model Fitting
# ## Moving Average

# In[30]:
import numpy as np
def moving_avg_prediction(data): shape = data.shape[0]
df_new = data[['Open']]
df_new['Open'] = np.log(df_new['Open']) # Apply logarithmictransformation
```

```
train_set = df_new.iloc[:ceil(shape * 0.9)] valid_set =
df_new.iloc[ceil(shape * 0.9):]
print('Shape of Training Set:', train_set.shape)print('Shape of Validation Set:',
valid_set.shape)
preds = []
for i in range(0, valid_set.shape[0]):
a = train_set['Open'][len(train_set) - valid_set.shape[0] + i:].sum() +
sum(preds)b = a / (valid_set.shape[0]) preds.append(b)
rms = np.sqrt(np.mean(np.power((np.array(valid_set['Open']) - preds), 2)))
print('RMSE value on validation set:', rms)
valid_set['Predictions'] = preds
plt.plot(train_set['Open'],linewidth=1.5)
plt.plot(valid_set[['Open','Predictions']],linewidth=1.5) plt.xlabel('Date',
size=20)
plt.ylabel('Log-transformed Open Price', size=20)
plt.title('Log-transformed Open Price Prediction by Moving Averages',
size=20)
plt.legend(['Model Training Data', 'Actual Data', 'Predicted Data'])


# In[31]:
from math import ceil, floor
moving_avg_prediction(data)
  # ##### Interpretation: The RMSE value is close to 0.362 but the results are not
  that good (as we see from the plot).
```

ARIMA

In[32]:

```
def moving_avg_prediction(data): shape = data.shape[0] df_new =
data[['Open']]
df_new['Open'] = np.log(df_new['Open']) # Apply logarithmic
transformation
train_set = df_new.iloc[:ceil(shape * 0.9)] valid_set =
df_new.iloc[ceil(shape * 0.9):]
print('Shape of Training Set:', train_set.shape) print('Shape of
Validation Set:', valid_set.shape) training = train_set['Open']
validation = valid_set['Open']
model = auto_arima(training, start_p=0, start_q=0, max_p=3, max_q=3,start_P=0,
seasonal=True, d=1, D=1, trace=True, error_action='ignore',
suppress_warnings=True) model.fit(training)
forecast = model.predict(n_periods=len(validation))
forecast.index = validation.index
forecast = pd.DataFrame(forecast,columns =['Prediction'])
#forecast = pd.DataFrame(forecast, index=valid_set.index, columns=['Prediction'])
rms = np.sqrt(np.mean(np.power((np.array(validation) -
np.array(forecast['Prediction'])), 2)))

print('RMSE value on validation set:', rms)
plt.plot(train_set['Open'],linewidth=1.5)
plt.plot(valid_set['Open'],linewidth=1.5)
plt.plot(forecast['Prediction'],linewidth=1.5) plt.xlabel('Date',
```

```
size=20)
plt.ylabel('Stock Price', size=20) plt.title('Stock Price Prediction by Auto
ARIMA', size=20)
plt.legend(['Model Training Data', 'Actual Data', 'Predicted Data'])

# In[33]:
data.set_index('Date',inplace=True)

# In[34]:
moving_avg_prediction(data)
  # #### Interpretation: An auto ARIMA model uses past data to understand the
  pattern in the time series. Using these values, the model captured an increasing
  trend in the series. Although the predictions using this technique are far better
  than that of the previously implemented model, these predictions are close than
  previous moving average model.
# ## LSTM

# In[35]:
from sklearn.model_selection import GridSearchCV from
sklearn.preprocessing import MinMaxScaler from keras.models
import Sequential
from keras.layers import Dense, Dropout, LSTM

# In[36]:
```

data

```
# In[37]:
shape = data.shape[0]
df_new=data[['Open']] dataset = df_new.values
train = df_new.iloc[:ceil(shape * 0.9)] valid =
df_new.iloc[ceil(shape * 0.9):]
print('Shape of Training Set',train.shape) print('Shape of
Validation Set',valid.shape) scaler =
MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(df_new)
x_train, y_train = [], [] for i in range(40,len(train)):
x_train.append(scaled_data[i-40:i,0])  y_train.append(scaled_data[i,0])
x_train, y_train = np.array(x_train), np.array(y_train) x_train =
np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))
model = Sequential()
model.add(LSTM(units=50, return_sequences=True,
input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=50))  model.add(Dense(1))

# In[38]:
model.compile(loss='mean_squared_error',  optimizer='adam')
model.fit(x_train, y_train, epochs=15, batch_size=1, verbose=1) # batch_size=1,
verbose=2
```

```
# In[39]:
inputs = df_new[len(df_new) - len(valid) - 40:] inputs
inputs.values.reshape(-1,1) inputs = scaler.transform(inputs)
X_test = []
for i in range(40,inputs.shape[0]): X_test.append(inputs[i-
40:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
opening_price = model.predict(X_test)
opening_price= scaler.inverse_transform(opening_price)
rms=np.sqrt(np.mean(np.power((valid-opening_price),2)))
print('RMSE value on validation set:',rms)

# In[40]:
#Visualize the prediction plt.figure(figsize=(10,5), dpi=100)
valid['Predictions'] = opening_price
plt.plot(df_new['Open']['2018-01-01':],linewidth=1.5)
plt.plot(valid[['Open','Predictions']],linewidth=1.5) plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.title('Stock Price Prediction by Long Short Term Memory (LSTM)')
plt.legend(['Model Training Data','Actual Data','Predicted Data'])

# In[41]:
valid['Predictions'] = opening_price
```

```
# In[42]:
valid

# In[43]:
plt.figure(figsize = (12,6)) plt.plot(valid[['Open']], 'b', label = "Original
Price",)
plt.plot(valid[['Predictions']], 'r', label = "Predicted Price") plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.grid(True) plt.show()

# In[44]:
import pandas as pd valid=pd.read_csv('valid.csv')
valid.index=val['Date']
valid

# In[45]:
import plotly.graph_objects as go import plotly.io as pio
# Create the figure fig = go.Figure()
# Add original price trace (blue)
fig.add_trace(go.Scatter(y=valid['Open'], mode='lines', name='Original Price',
line=dict(color='green')))
# Add predicted price trace (red)
fig.add_trace(go.Scatter(y=valid['Predictions'], mode='lines',
```

```
name='Predicted Price', line=dict(color='red')))
# Customize layout fig.update_layout(
title="Stock Price Prediction", xaxis_title="Time",
yaxis_title="Price", legend=dict(x=0, y=1),
template="plotly_dark"
fig.write_html("prediction.html") # Show interactive plot
fig.show()

# In[46]:
import plotly.graph_objects as go
# Create a bar chart figure fig_bar = go.Figure()
# Add Original Price (Blue)
fig_bar.add_trace(go.Bar(y=valid['Open'], name='Original Price',
marker=dict(color='blue')))
# Add Predicted Price (Red)
fig_bar.add_trace(go.Bar(y=valid['Predictions'], name='Predicted Price',
marker=dict(color='red')))
# Customize layout fig_bar.update_layout(
title="Stock Price Prediction - Bar Chart", xaxis_title="Time",
yaxis_title="Price", legend=dict(x=0, y=1),
template="plotly_dark",
barmode='group' # Grouped bars
# Save as an HTML file
fig_bar.write_html("stock_price_bar_chart.html")
# Show interactive plot fig_bar.show()
```

```
# In[47]:
import plotly.graph_objects as go
# Create a bar chart fig_bar = go.Figure()
# Add Actual Prices (Blue) fig_bar.add_trace(go.Bar(
x=valid.index, y=valid['Open'], name="Actual Price",
marker_color='blue'))
# Add Predicted Prices (Red)
fig_bar.add_trace(go.Bar( x=valid.index,
y=valid['Predictions'], name="Predicted Price",
marker_color='red'))
# Customize layout fig_bar.update_layout(
title="Stock Price Prediction - Bar Chart", xaxis_title="Time",
yaxis_title="Price",
barmode='group', # Bars appear side by side template="plotly_white"
)
# Save as an HTML file
fig_bar.write_html("stock_price_barchart.html")
# Show interactive plot fig_bar.show()


# In[48]:
# Take the last 40 days from the dataset last_40_days = df_new[-
40:].values # Shape (40, 1)
last_40_scaled = scaler.transform(last_40_days) # Scale the data
# Reshape to fit LSTM input
```

```
X_future = np.array([last_40_scaled]) # Shape (1, 40, 1)# Predict the next day's
price predicted_price = model.predict(X_future)
# Inverse transform to get actual price predicted_price =
scaler.inverse_transform(predicted_price)
print("Predicted Next Day Opening Price:", predicted_price[0][0])


# In[49]:
valid[::-1]

# In[50]:
valid['Open']

# In[51]:
valid['Predictions']

# In[52]:
from sklearn.metrics import r2_score
# Actual values actual = valid['Open']
# Predicted values predicted = valid['Prediction# Calculate
the R2 score
r2 = r2_score(actual, predicted)
print("R2 score:", r2)

# In[53]:
# Plotting the R2 score fig, ax = plt.subplots()
```

```
 ax.barh(0, r2, color='skyblue') ax.set_xlim([-1, 1])
 ax.set_yticks([]) ax.set_xlabel('R2 Score') ax.set_title('R2
 Score')
 # Adding the R2 score value on the bar ax.text(r2, 0, f'{r2:.2f}',
 va='center', color='black')
 plt.show()
 from flask import Flask, render_template, request,
 jsonify,session,redirect,url_for

 import yfinance as yf import datetime as dt import
 pandas as pd import numpy as np
 import plotly.graph_objects as g
 from plotly.offline import plot
from sklearn.linear_model import LinearRegression from sklearn
import preprocessing, model_selectionimport json
import sqlite3
app = Flask(_name_)
app.secret_key = 'supersecretkey'
def get_db_connection():
try:
conn = sqlite3.connect('mydb.db') cursor = conn.cursor()
# Create the 'users' table cursor.execute('"
 CREATE TABLE IF NOT EXISTS users (
 id INTEGER PRIMARY KEY AUTOINCREMENT,
 username TEXT NOT NULL UNIQUE, email TEXT NOT NULL
 UNIQUE,
```

```
password TEXT NOT NULL, datetime TEXT NOT NULL)"')
conn.row_factory = sqlite3.Row
return conn except sqlite3.Error as e:
print(f"Database connection failed: {e}") return None
# def load_data(ticker):
#data = yf.download(ticker, START, TODAY)
#data.reset_index(inplace=True)
#data.columns = [col[0] for col in data.columns.values]
#data.index=data.Date
#return data
@app.route('/') def index():
if 'user_id' in session:username=session['user_name']
data = yf.download(tickers=['AAPL', 'AMZN', 'QCOM', 'META', 'NVDA', 'JPM'],
group_by='ticker', threads=True, period='1mo',
interval='1d')
data.reset_index(level=0, inplace=True)
fig_left = go.Figure()
for ticker in ['AAPL', 'AMZN', 'QCOM', 'META', 'NVDA', 'JPM']:
fig_left.add_trace(go.Scatter(x=data['Date'],   y=data[ticker]['Close'],
name=ticker))fig_left.update_layout(paper_bgcolor="#14151b",
plot_bgcolor="#14151b", font_color="white")plot_div_left = plot(fig_left,
auto_open=False, output_type='div')# ========================== To
show recent stocks=================
# tickers = ['AAPL', 'AMZN', 'GOOGL', 'UBER', 'TSLA', 'TWTR']
# dfs = []
```

```
# for ticker in tickers:
#df = yf.download(tickers=ticker, period='1d', interval='1d') #data.columns =
[col[0] for col in data.columns.values]
#df.insert(0, "Ticker", ticker)
#dfs.append(df)
# df = pd.concat(dfs, axis=0)
# df.reset_index(level=0, inplace=True)
# print(df.columns)
# df.columns = ['Date','Ticker' 'Open', 'High', 'Low', 'Close', 'Volume']
# df.drop('Date', axis=1, inplace=True)
# json_records = df.reset_index().to_json(orient='records')# recent_stocks
= json.loads(json_records)
tickers = ['AAPL', 'AMZN', 'GOOGL', 'UBER', 'TSLA',]
dfs = []
for ticker in tickers:
df = yf.download(tickers=ticker, period='1d', interval='1d') df.columns =
[col[0] for col in df.columns.values]
# print(df.columns) df.insert(0, "Ticker", ticker)
dfs.append(df)
df = pd.concat(dfs, axis=0) df.reset_index(level=0,
inplace=True) df.drop('Date', axis=1, inplace=True)
json_records = df.reset_index().to_json(orient='records') recent_stocks =
json.loads(json_records)
recent_stocks print("username username",username)
with open("./static/prediction.html", "r", encoding="utf-8") as f:
```

```
plot_html = f.read()
with open("./static/stock_price_bar_chart.html", "r", encoding="utf-8")
as f:bar_chart = f.read
return render_template('home.html',recent_stocks=recent_stocks,
plot_div_left=plot_div_left,username=username,plot_html=plot_html,bar_chart=bar_
chart)
return render_template('index.html')
# ==================== Left Card Plot=========================
@app.route('/add_user', methods=['POST']) def add_user():
try:
from datetime import datetime username =
request.form['name'] print(username)
email = request.form['email'] password =
request.form['password']
datetime_now = datetime.now().isoformat()
conn = get_db_connection() if conn:
cursor = conn.cursor()
cursor.execute('INSERT INTO users (username, email, password, datetime) VALUES
(?, ?, ?, ?)',(username, email, password, datetime_now)) conn.commit()
conn.close()return redirect(url_for('index'))else:
return jsonify({'status': 'failed', 'error': 'Database connection failed'})
except Exception as e:print(f"Error in add_user: {e}")
return jsonify({'status': 'failed', 'error': str(e)})
def validate(username, password):
try:
```

```
conn = get_db_connection()
if conn:
cursor = conn.cursor()
cursor.execute('SELECT * FROM users WHERE username = ?',
(username,))
user = cursor.fetchone()conn.close()
if user and user['password'] == password: return True, username
return False, username except sqlite3.Error as e:
print(f"Database error during validation: {e}")return False, username
@app.route('/verify_user', methods=['POST', 'GET'])def login():
try:
if request.method == 'POST':
username = request.form['username'] password =
request.form['password']
completion, username = validate(username, password) if completion:
conn = get_db_connection()if conn:
cursor = conn.cursor()
cursor.execute('SELECT id FROM users WHERE username = ?',
(username,))
user = cursor.fetchone() conn.close()
if user: session['user_id'] = user['id']
session['user_name']=username session['logged_in'] = Truereturn
redirect(url_for('index'))
return render_template('index.html', error = 'User not found')
```

```
return render_template('index.html', error = 'Invalid credentials')
else:render_template('index.html')
except Exception as E:
return render_template('index.html',error=E)
@app.route('/logout') def logout():
session.pop('logged_in', None) session.pop('user_id',
None)session.pop('user_name', None)
return redirect(url_for('index')
@app.route('/search') def search():
if 'user_id' in session:
return render_template('search.html') return redirect(url_for('index'))
@app.route('/getinfo') def getinfo():
if 'user_id' in session:
return render_template('getinfo.html') return
redirect(url_for('index'))
@app.route('/ticker')def ticker():
if 'user_id' in session:
# Simulating ticker data
ticker_df = pd.read_csv('Data/new_tickers.csv')json_ticker =
ticker_df.reset_index().to_json(orient='records')
ticker_list = json.loads(json_ticker)
return render_template('ticker.html', ticker_list=ticker_list) return
redirect(url_for('index'))
@app.route('/stock/<string:ticker_value>/') def stock(
ticker_value):
```

```
if 'user_id' in session: print(ticker_value)
try:
# ticker_value = request.POST.get('ticker') ticker_value =
ticker_value.upper()
stock = yf.Ticker(ticker_value)stock_info = stock.info
return render_template('stock.html',stock_info=stock_info)
except:
return render_template( 'API_Down.html')
@app.route('/predict/<string:ticker_value>/<int:number_of_days>') def
predict( ticker_value, number_of_days):
if 'user_id' in session: print(ticker_value,number_of_days)
try:
# ticker_value = request.POST.get('ticker') ticker_value =
ticker_value.upper()
df = yf.download(tickers = ticker_value, period='3mo', interval='1d') df.columns
= [col[0] for col in df.columns.values] print("Downloaded ticker = {}
successfully".format(ticker_value)) print("number of days =",number_of_days)
except:return render_template( 'API_Down.html')
try:
# number_of_days = request.POST.get('days')number_of_days =
int(number_of_days)
except:
return render_template( 'API_Down.html')
if number_of_days < 0:
return render_template( 'API_Down.html')
```

```
if number_of_days > 365:
return render_template( 'API_Down.html
fig = go.Figure() fig.add_trace(go.Candlestick(x=df.index,
open=df['Open'],
high=df['High'],
low=df['Low'], close=df['Close'], name = 'market data'))
fig.update_layout(
title='{} live share price evolution'.format(ticker_value), yaxis_title='Stock
Price (USD per Shares)')
fig.update_xaxes( rangeslider_visible=True,
rangeselector=dict(
buttons=list([
dict(count=15, label="15m", step="minute", stepmode="backward"),
dict(count=45, label="45m", step="minute", stepmode="backward"),
dict(count=1, label="HTD", step="hour", stepmode="todate"), dict(count=3,
label="3h", step="hour", stepmode="backward"), dict(step="all")])
)
)
# fig.update_layout(paper_bgcolor="#14151b", plot_bgcolor="#14151b",
                    font_color="darkslategray")
fig.update_layout( title=dict(
text="Prediction Chart", font=dict(color="white") # White title for dark background),
xaxis=dict( title=dict( text="Date",
font=dict(color="lightgray") # Light gray for x-axis title),
```

```
tickfont=dict(color="lightyellow") # Light yellow for x-axis ticklabels),
yaxis=dict( title=dict(text="Prediction Value",
font=dict(color="lightgray") # Light gray for y-axis
title), tickfont=dict(color="lightgreen") # Light green
for y-axis tick labels),
legend=dict(font=dict(color="white") # White for legend font color),
font=dict(color="darkslategray"), # Light gray for all text elements
plot_bgcolor='black', # Set background color of the plot to black
paper_bgcolor='black' # Set the background color of the entire figure
to black)

plot_div = plot(fig, auto_open=False, output_type='div'
# ====================== Machine Learning ======================
try:
df_ml = yf.download(tickers=ticker_value, period='3mo',
interval='1h')
except: ticker_value = 'AAPL'
df_ml = yf.download(tickers=ticker_value, period='3mo',
interval='1m')
df_ml.columns = [col[0] for col in df_ml.columns.values] df_ml =
df_ml[['Close']]
df_ml['Prediction'] = df_ml[['Close']].shift(-number_of_days)

X = np.array(df_ml.drop(['Prediction'], axis=1))
```

```
 X = preprocessing.scale(X)
X_forecast = X[-number_of_days:]
X = X[:-number_of_days]
y = np.array(df_ml['Prediction'])
 y = y[:-number_of_days]
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,

test_size=0.2)
clf = LinearRegression()clf.fit(X_train, y_train)
confidence = clf.score(X_test, y_test)
forecast_prediction = clf.predict(X_forecast)
forecast =forecast_prediction.tolist()
# Plotting predicted data pred_dict = {"Date": [], "Prediction": []}
for i in range(0, len(forecast)):pred_dict["Date"].append(dt.datetime.today() +
dt.timedelta(days=i)) pred_dict["Prediction"].append(forecast[i])
pred_df = pd.DataFrame(pred_dict)
pred_fig = go.Figure([go.Scatter(x=pred_df['Date'], y=pred_df['Prediction'])])

pred_fig.update_xaxes(rangeslider_visible=True)
pred_fig.update_layout(paper_bgcolor="#14151b", plot_bgcolor="#14151b",
font_color="white")

plot_div_pred = plot(pred_fig, auto_open=False, output_type='div')
        # ========================================== Display Ticker Info
            ==========================================
            stock = yf.Ticker(ticker_value)
             stock_info = stock.info
```

```
        # ========================================== Page Render
          section ==========================================

  return render_template( "result.html", plot_div= plot_div,
  confidence=confidence,
   forecast= forecast,
   ticker_value=ticker_value,
number_of_days=number_of_days,
 plot_div_pred=plot_div_pred,
 stock_info=stock_info)
return redirect(url_for('index'))
                if _____name___== '__main__':
                    app.run(debug=True)

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Stock Market Prediction</title>
<meta content="width=device-width, initial-scale=1.0" name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">
<!-- Favicon -->
<link rel="icon" href="{{url_for('static',filename='image/stkicon.png')}}">
<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&family=
Roboto:wght@500;700&display=swap" rel="stylesheet">
<!-- Icon Font Stylesheet -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">
<!-- Libraries Stylesheet -->
<link href="

{{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}"
rel="stylesheet">

<link href="

{{url_for('static',filename='lib/tempusdominus/css/tempusdominus-bootstrap-
4.min.css')}}" rel="stylesheet" />
<!-- Customized Bootstrap Stylesheet -->
<link href="{{url_for('static',filename='css/bootstrap.min.css')}}"

rel="stylesheet">

<!-- Template Stylesheet -->
<link  href="{{url_for('static',filename='css/style.css')}}"  rel="stylesheet">
</head>
<body>
```

```
<div class="container-fluid position-relative d-flex p-0">
<!-- Spinner Start -->
<div id="spinner" class="show bg-dark position-fixed translate-middle w-100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height:3rem;" role="status">

<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-secondary navbar-dark">
<a href="#" class="navbar-brand mx-4 mb-3">
<h3 ><img src="{{url_for('static',filename='image/stkicon.png')}} " style="width: 25px;" alt="logo">Trade Strategy</h3>

</a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">
<img class="rounded-circle" src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<div class="bg-success rounded-circle border border-2 border- white position-absolute end-0 bottom-0 p-1"></div>
```

```
</div>
<div class="ms-3">
<h6 class="mb-0">Welcome</h6>
<span>{{username}}</span>
</div>
</div>
<div class="navbar-nav w-100" style="background: rgb(219, 153,
153)smoke;">
<a href="#" class="nav-item nav-link active"><i class="fa fa- home me-
2"></i>Home</a>
<a href="/search" class="nav-item nav-link"><i class="fa fa- search me-
2"></i>Predict</a>
<a href="/getinfo" class="nav-item nav-link"><i class="fa fa-list-alt me-2"></i>Get
Info</a>
<a href="/ticker" class="nav-item nav-link"><i class="fa fa-list-alt me-
2"></i>Ticker Info</a>
</div>
</nav>
</div>
<!-- Sidebar End -->
<!-- Content Start -->
<div class="content">
<!-- Navbar Start -->
```

```
<nav class="navbar navbar-expand bg-secondary navbar-dark sticky-
top px-4 py-0">
<a href="index.html" class="navbar-brand d-flex d-lg-none me-4">
<!-- <h2 class="text-primary mb-0"><i class="fa fa-user-edit"></i></h2> -->
</a>
<a href="#" class="sidebar-toggler flex-shrink-0 bars" >
<i class="fa fa-bars"></i>
</a>
<!-- <form class="d-none d-md-flex ms-4">
<input class="form-control bg-dark border-0" type="search" placeholder="Search">
</form> -->
<div class="navbar-nav align-items-center ms-auto">
<a ><span class="d-none d-lg-inline-flex"><a href="/logout"
>LOG OUT</a></span>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">
<img class="rounded-circle me-lg-2"
src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<span class="d-none d-lg-inline-flex">{{username}}</span>
</a>
</a>
<div class="dropdown-menu dropdown-menu-end bg-secondary border-0 rounded-0
rounded-bottom m-0">
```

```
<a href="/search" class="dropdown-item">Predict</a>
<a href="/getinfo" class="dropdown-item">Get Info</a>
<a href="/ticker" class="dropdown-item">Ticker Info</a>
</div>
</div></div>
</nav>
<!-- Navbar End -->
<!-- Sales Chart Start -->
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-12">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Active Stocks</h6>
<a href="">Show All</a>
</div>
{% block Left_Plot %}
{% autoescape off %}
{{ plot_div_left }}
{% endautoescape %}
{% endblock Left_Plot %}
</div>
</div>
</div>
</div>
```

```
<!-- Sales Chart End -->
<!-- Recent Sales Start -->
<div class="container-fluid pt-4 px-4">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Recent Stocks</h6>
<a  href="https://finance.yahoo.com/trending-tickers">Show  All</a>
</div>
<div class="table-responsive">
<table class="table text-start align-middle table-bordered table- hover mb-0">
<thead>
<tr class="text-white">
<th scope="col">Ticker</th>
<th scope="col">Open</th>
<th scope="col">High</th>
<th scope="col">Low</th>
<th scope="col">Close</th>
<th scope="col">Volume</th>
</tr>
</thead>
<tbody>
{% if recent_stocks %}
{% for i in recent_stocks %}
<tr>
<td>{{i.Ticker}}</td>
```

```
<td>{{i.Open}}</td>
<td>{{i.High}}</td>
<td>{{i.Low}}</td>
<td>{{i.Close}}</td>
<td>{{i.Volume}}</td>
</tr>
{% endfor %}
{% endif %}
</tbody>
</table>
</div>
</div>
</div>
<!-- Recent Sales End -->
<!-- Sales Chart Start -->
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-12">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Model Prediction with comparison</h6>
<!-- <a href="">Show All</a> -->
</div>
{{ plot_html|safe }}
</div>
```

```
</div>
</div>
</div>
<!-- Sales Chart End -->
<!-- Sales Chart Start -->
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-12">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Comparion bar chart</h6>
<!-- <a href="">Show All</a> -->
</div>
{{ bar_chart |safe }}
</div>
</div>
</div>
</div>
<!-- Sales Chart End -->
<!-- Footer Start -->
<div class="container-fluid pt-4 px-4">
<div class="bg-secondary rounded-top p-4">
<div class="row">
<div class="col-12 col-sm-6 text-center text-sm-start">&copy; <a >My site</a>, All Right Reserved.
```

```
</div>
<div class="col-12 col-sm-6 text-center text-sm-end"> Designed By <a >Me</a>
</div>
</div>
</div>
</div>
<!-- Footer End -->
</div>
<!-- Content End -->
<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-lg-square back-to-top" style="background:
#16c4eb;"><i class="bi bi-arrow-up"></i></a>
</div>
<!-- JavaScript Libraries -->
<script  src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.mi
                                        n.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.8.0/chart.min.js"
 integrity="sha512-
 sW/w8s4RWTdFFSduOTGtk4isV1+190E/GghVffMA9XczdJ2MDzSzLEubK
 As5h0wzgSJOQTRYyaz73L3d6RtJSg==" crossorigin="anonymous"
 referrerpolicy="no-referrer"></script>

<script src="

{{url_for('static',filename='lib/easing/easing.min.js')}}"></script>
<script
```

```
src="{{url_for('static',filename='lib/waypoints/waypoints.min.jsg')}}"></script>
<script src="
{{url_for('static',filename='lib/owlcarousel/owl.carousel.min.js')}}"></script><scri
pt src="
{{url_for('static',filename='lib/tempusdominus/js/moment.min.js')}}"></script
>
<script src="{{url_for('static',filename='lib/tempusdominus/js/moment-
timezone.min.js')}}"></script>
<script src="
 {{url_for('static',filename='lib/tempusdominus/js/tempusdominus-bootstrap-
                                    4.min.js')}}"></script>
<!-- Template Javascript -->
<script src=" {{url_for('static',filename='js/main.js')}}"></script>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8"><title>Stock Market Prediction</title>
<meta content="width=device-width, initial-scale=1.0" name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">
<!-- Favicon -->
<link rel="icon" href="{{url_for('static',filename='image/stkicon.png')}}">
```

```
<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&family
=Roboto:wght@500;700&display=swap"  rel="stylesheet">
<!-- Icon Font Stylesheet -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.10.0/css/all.min.css" rel="stylesheet">
<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">
<!-- Libraries Stylesheet -->
<link href="

{{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}"
rel="stylesheet">

<link href="

{{url_for('static',filename='lib/tempusdominus/css/tempusdominus-bootstrap-
4.min.css')}}" rel="stylesheet" />
<!-- Customized Bootstrap Stylesheet -->
<link href="{{url_for('static',filename='css/bootstrap.min.css')}}"

rel="stylesheet">
<!-- Template Stylesheet -->
<link  href="{{url_for('static',filename='css/style.css')}}"  rel="stylesheet">
</head>
```

```
<body><div class="container-fluid position-relative d-flex p-0">
<!-- Spinner Start -->
<div id="spinner" class="show bg-dark position-fixed translate-middle w-100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height: 3rem;" role="status">
<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-secondary navbar-dark">
<a href="#" class="navbar-brand mx-4 mb-3">
<h3 ><img src="
{{url_for('static',filename='image/stkicon.png')}} " style="width: 25px;" alt="logo">Trade Strategy</h3>
</a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">
<img class="rounded-circle" src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<div class="bg-success rounded-circle border border-2 border- white position-absolute end-0 bottom-0 p-1"></div>
```

```
</div>
<div class="ms-3">
<h6 class="mb-0">Welcome</h6>
<span>{{username}}</span>
</div>
</div>
<div class="navbar-nav w-100" style="background: rgb(219, 153,
                                    153)smoke;">
<a href="/" class="nav-item nav-link "><i class="fa fa-home me- 2"></i>Home</a>
<a href="/search" class="nav-item nav-link active"><i class="fa fa-search me-
2"></i>Predict</a>
<a href="/getinfo" class="nav-item nav-link"><i class="fa fa-list- alt me-
2"></i>Get Info</a>
<a href="/ticker" class="nav-item nav-link"><i class="fa fa-list alt me-
2"></i>Ticker Info</a>
</div>
</nav>
</div>
<!-- Sidebar End -->
<!-- Content Start -->
<div class="content">
<!-- Navbar Start -->
<nav class="navbar navbar-expand bg-secondary navbar-dark sticky-
top px-4 py-0">
```

```
<!-- <a href="index.html" class="navbar-brand d-flex d-lg-none me-4">
<h2 class="text-primary mb-0"><i class="fa fa-user- edit"></i></h2>
</a> -->
<a href="#" class="sidebar-toggler flex-shrink-0 bars" >
<i class="fa fa-bars"></i>
</a>
<!-- <form class="d-none d-md-flex ms-4">
<input class="form-control bg-dark border-0" type="search" placeholder="Search">
</form> -->
<div class="navbar-nav align-items-center ms-auto">
<a ><span class="d-none d-lg-inline-flex"><a href="/logout"
>LOG OUT</a></span>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">
  <img class="rounded-circle me-lg-2"
  src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<span class="d-none d-lg-inline-flex">{{username}}</span>
</a>
</a>
<div class="dropdown-menu dropdown-menu-end bg- secondary border-0 rounded-0
rounded-bottom m-0">
<a href="/search" class="dropdown-item">Predict</a>
<a href="/getinfo" class="dropdown-item">Get Info</a>
```

```
<a href="/ticker" class="dropdown-item">Ticker Info</a>
</div>
</div>
</div>
</nav>
<!-- Navbar End -->
<div class="container-fluid">
<div class="row h-100 align-items-center justify-content-center" style="min-height: 75vh;">
<div class="col-10 col-sm-10 col-md-10 col-lg-9 col-xl-10">
<div class="bg-secondary rounded p-2 p-sm-5 my-2 mx-2">
<div class="d-flex align-items-center justify-content-between">
<a class="">
<h3><i class="fa fa-user-edit me-2"></i>Get Stock Information</h3>

</a>
</div>
<div class="form-floating mb-3">
  <input type="text" class="form-control" id="tickerinp" placeholder="Ticker Name"
  name="ticker" autocomplete="on" required>

<label for="tickerinp">Ticker Name</label>
</div>
<!-- <div class="form-floating mb-4">
 <input type="text" class="form-control" id="days" placeholder="Number of days"
 name="days" autocomplete="on" required>
 <label for="days">Number of Days</label>
```

```
</div> -->
<div class="d-flex align-items-center justify-content-between mb-4">
<a href="../ticker">Search Ticker Value</a>
</div>
<button type="submit" class="btn py-3 w-100 mb-4" style="background: rgb(26, 68,
97);color: white;"
onclick="LoadURL()">Predict</button>
<script type="text/javascript"> function LoadURL() {
var ticker = document.getElementById("tickerinp").value;
var server = window.location.origin;
 var path = "/stock/" + ticker + "/";
 var url = server + path; window.location.href = url;
}
</script>
</div>
</div>
</div>
</div>
<!-- Widgets End -->
<!-- Footer Start -->
<div class="container-fluid pt-4 px-4">
<div class="bg-secondary rounded-top p-4">
<div class="row">
<div class="col-12 col-sm-6 text-center text-sm-start"> &copy; <a >My site</a>, All
```

```
Right Reserved.
</div>
<div class="col-12 col-sm-6 text-center text-sm-end"> Designed By <a >Me</a>
</div>
</div>
</div>
</div>
<!-- Footer End -->
</div>
<!-- Content End -->
<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-lg-square back-to-top" style="background:
#16c4eb;"><i class="bi bi-arrow-up"></i></a>

</div>
<!-- JavaScript Libraries -->
<script  src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.mi
                                        n.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.8.0/chart.min.js"
 integrity="sha512-
 sW/w8s4RWTdFFSduOTGtk4isV1+190E/GghVffMA9XczdJ2MDzSzLEubK
 As5h0wzgSJOQTRYyaz73L3d6RtJSg==" crossorigin="anonymous"
 referrerpolicy="no-referrer"></script>

<script src="
```

{{url_for('static',filename='lib/easing/easing.min.js')}}"></script>

<script

src="{{url_for('static',filename='lib/waypoints/waypoints.min.jsg')}}"></script>

<script src="

{{url_for('static',filename='lib/owlcarousel/owl.carousel.min.js')}}"></script>

<script src="

{{url_for('static',filename='lib/tempusdominus/js/moment.min.js')}}"></script>

<script src="{{url_for('static',filename='lib/tempusdominus/js/moment-timezone.min.js')}}"></script>

<script src="

{{url_for('static',filename='lib/tempusdominus/js/tempusdominus-bootstrap-4.min.js')}}"></script>

<!-- Template Javascript -->

<script src=" {{url_for('static',filename='js/main.js')}}"></script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Stock Market Prediction</title>

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<meta content="" name="keywords">

<meta content="" name="description">

```
<!-- Favicon -->
<link rel="icon" href="{{url_for('static',filename='image/stkicon.png')}}
">
<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&f
        amily=Roboto:wght@500;700&display=swap" rel="stylesheet">

<!-- Icon Font Stylesheet -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">
<!-- Libraries Stylesheet -->
<link href="
  {{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}"
  rel="stylesheet">

<link href="

{{url_for('static',filename='lib/tempusdominus/css/tempusdominus-bootstrap-
4.min.css')}}" rel="stylesheet" />
<!-- Customized Bootstrap Stylesheet -->
<link href="{{url_for('static',filename='css/bootstrap.min.css')}}"

rel="stylesheet">
```

```
<!-- Template Stylesheet -->
<link href="{{url_for('static',filename='css/style.css')}}" rel="stylesheet">
</head>
<body>
<div class="container-fluid position-relative d-flex p-0">
<!-- Spinner Start -->
<div id="spinner" class="show bg-dark position-fixed translate-middle w-100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height: 3rem;" role="status">
<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-secondary navbar-dark">
<a href="#" class="navbar-brand mx-4 mb-3">
<h3 ><img src="{{url_for('static',filename='image/stkicon.png')}} " style="width: 25px;" alt="logo">Trade Strategy</h3></a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">
<img class="rounded-circle" src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
```

```
<div class="bg-success rounded-circle border border-2 border- white position-
absolute end-0 bottom-0 p-1">
</div>
</div>
<div class="ms-3">
<h6 class="mb-0">Welcome</h6>
<span>{{username}}</span>
</div>
</div>
<div class="navbar-nav w-100" style="background: rgb(219, 153,
153)smoke;">
<a href="/" class="nav-item nav-link "><i class="fa fa-home me- 2"></i>Home</a>
<a href="/search" class="nav-item nav-link active"><i class="fa fa-search me-
2"></i>Predict</a>
<a href="/getinfo" class="nav-item nav-link"><i class="fa fa-list- alt me-
2"></i>Get Info</a>
<a href="/ticker" class="nav-item nav-link"><i class="fa fa-list- alt me-
2"></i>Ticker Info</a>
</div>
</nav>
</div>
<!-- Sidebar End -->
<!-- Content Start -->
<div class="content">
```

```
<!-- Navbar Start -->
<nav class="navbar navbar-expand bg-secondary navbar-dark sticky-top px-4 py-
0">
<!-- <a href="index.html" class="navbar-brand d-flex d-lg-none me-4">
<h2 class="text-primary mb-0"><i class="fa fa-user- edit"></i></h2>
</a> -->
<a href="#" class="sidebar-toggler flex-shrink-0 bars" >
<i class="fa fa-bars"></i>
</a>
<!-- <form class="d-none d-md-flex ms-4">
<input class="form-control bg-dark border-0" type="search" placeholder="Search">
</form> -->
<div class="navbar-nav align-items-center ms-auto">
<a ><span class="d-none d-lg-inline-flex"><a href="/logout"
>LOG OUT</a></span>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">
  <img class="rounded-circle me-lg-2"
  src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<span class="d-none d-lg-inline-flex">{{username}}</span>
</a>
</a>
<div class="dropdown-menu dropdown-menu-end bg- secondary border-0 rounded-0
```

```
rounded-bottom m-0">

<a href="/search" class="dropdown-item">Predict</a>
<a href="/getinfo" class="dropdown-item">Get Info</a>
<a href="/ticker" class="dropdown-item">Ticker Info</a>
</div>
</div>
</div>
</nav>
<!-- Navbar End -->
<div class="container-fluid">
<div class="row h-100 align-items-center justify-content-center" style="min-
height: 75vh;">
<div class="col-10 col-sm-10 col-md-10 col-lg-9 col-xl-10">
<div class="bg-secondary rounded p-2 p-sm-5 my-2 mx-2">
<div class="d-flex align-items-center justify-content-between">
<a class="">
<h3><i class="fa fa-user-edit me-2"></i>Stock Market Predictor</h3>

</a>
</div>
<div class="form-floating mb-3">
  <input type="text" class="form-control" id="tickerinp" placeholder="Ticker Name"
  name="ticker" autocomplete="on" required>

<label for="tickerinp">Ticker Name</label>
</div>
```

```
<div class="form-floating mb-4">
 <input type="text" class="form-control" id="days" placeholder="Number of days"
 name="days" autocomplete="on" required>

<label for="days">Number of Days</label>
</div>
<div class="d-flex align-items-center justify-content-between mb-4">
<a href="../ticker">Search Ticker Value</a>
</div>
<button type="submit" class="btn py-3 w-100 mb-4" style="background: rgb(26, 68,
97);color: white;"
onclick="LoadURL()">Predict</button>
<script type="text/javascript"> function LoadURL() {
var ticker = document.getElementById("tickerinp").value;

var days = document.getElementById("days").value; var server =
window.location.origin;
var path = "/predict/" + ticker + "/" + days;
var url = server + path;
window.location.href = url;
}
</script>
</div>
</div>
</div>
</div>
```

```
<!-- Widgets End -->
<!-- Footer Start -->
<div class="container-fluid pt-4 px-4">
<div class="bg-secondary rounded-top p-4">
<div class="row">
<div class="col-12 col-sm-6 text-center text-sm-start"> &copy; <a >My site</a>, All
Right Reserved.
</div>
<div class="col-12 col-sm-6 text-center text-sm-end"> Designed By <a
>Me</a></div>
</div>
</div>
</div>
<!-- Footer End -->
</div>
<!-- Content End -->
<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-lg-square back-to-top" style="background:
#16c4eb;"><i class="bi bi-arrow-up"></i></a>

</div>
<!-- JavaScript Libraries -->
<script  src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.mi
                                        n.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.8.0/chart.min.js"
```

```
integrity="sha512-
sW/w8s4RWTdFFSduOTGtk4isV1+190E/GghVffMA9XczdJ2MDzSzLEubK
As5h0wzgSJOQTRYyaz73L3d6RtJSg==" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

<script src="

{{url_for('static',filename='lib/easing/easing.min.js')}}"></script>
<script
src="{{url_for('static',filename='lib/waypoints/waypoints.min.jsg')}}"></script>
<script src="
{{url_for('static',filename='lib/owlcarousel/owl.carousel.min.js')}}"></script>
<script src="

{{url_for('static',filename='lib/tempusdominus/js/moment.min.js')}}"></script

>
<script src="{{url_for('static',filename='lib/tempusdominus/js/moment-
timezone.min.js')}}"></script>

<script src="

 {{url_for('static',filename='lib/tempusdominus/js/tempusdominus-bootstrap-
                                    4.min.js')}}"></script>
<!-- Template Javascript -->
<script src=" {{url_for('static',filename='js/main.js')}}"></script>
</body>
</html>
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="utf-8">
<title>Stock Market Prediction</title>
<meta content="width=device-width, initial-scale=1.0" name="viewport">
<meta content="" name="keywords">
<meta content="" name="description">
<!-- Favicon -->
<link rel="icon" href="{{url_for('static',filename='image/stkicon.png')}}">
<!-- Google Web Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600&f
        amily=Roboto:wght@500;700&display=swap" rel="stylesheet">
<!-- Icon Font Stylesheet -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.10.0/css/all.min.css" rel="stylesheet">

<link href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">
<!-- Libraries Stylesheet -->
<link href="

  {{url_for('static',filename='lib/owlcarousel/assets/owl.carousel.min.css')}}"
  rel="stylesheet">

<link href="
```

```
{{url_for('static',filename='lib/tempusdominus/css/tempusdominus-bootstrap-4.min.css')}}" rel="stylesheet" />
<!-- Customized Bootstrap Stylesheet -->
<link href="{{url_for('static',filename='css/bootstrap.min.css')}}"
rel="stylesheet">
<!-- Template Stylesheet -->
<link href="{{url_for('static',filename='css/style.css')}}" rel="stylesheet">
</head>
<body>
<div class="container-fluid position-relative d-flex p-0">
<!-- Spinner Start -->
<div id="spinner" class="show bg-dark position-fixed translate-middle w-100 vh-100 top-50 start-50 d-flex align-items-center justify-content-center">
<div class="spinner-border text-primary" style="width: 3rem; height: 3rem;" role="status">
<span class="sr-only">Loading...</span>
</div>
</div>
<!-- Spinner End -->
<!-- Sidebar Start -->
<div class="sidebar pe-4 pb-3">
<nav class="navbar bg-secondary navbar-dark">
<a href="#" class="navbar-brand mx-4 mb-3">
<h3 ><img src="
{{url_for('static',filename='image/stkicon.png')}} " style="width: 25px;"
```

```
alt="logo">Trade Strategy</h3>

</a>
<div class="d-flex align-items-center ms-4 mb-4">
<div class="position-relative">
<img class="rounded-circle" src="{{url_for('static',filename='image/user.png')}} "
alt="" style="width:
                                    40px; height: 40px;">
<div class="bg-success rounded-circle border border-2 border- white position-
absolute end-0 bottom-0 p-1"></div>
</div>
<div class="ms-3">
<h6 class="mb-0">Welcome</h6>
<span>{{username}}</span>
</div>
</div>
<div class="navbar-nav w-100" style="background: rgb(219, 153,153)smoke;">
<a href="/" class="nav-item nav-link "><i class="fa fa-home me- 2"></i>Home</a>
<a href="/search" class="nav-item nav-link"><i class="fa fa- search me-
2"></i>Predict</a>

<a href="/getinfo" class="nav-item nav-link"><i class="fa fa-list- alt me-
2"></i>Get Info</a>
<a href="/ticker" class="nav-item nav-link"><i class="fa fa- building me-
2"></i>Ticker Info</a>
<a href="#" class="nav-item nav-link active"><i class="fa fa-check-circle me-
```

```
2"></i>Prediction</a>
</div>
</nav>
</div>
<!-- Sidebar End -->
<!-- Content Start -->
<div class="content">
<!-- Navbar Start -->
<nav class="navbar navbar-expand bg-secondary navbar-dark sticky-
top px-4 py-0">
<a href="index.html" class="navbar-brand d-flex d-lg-none me-4">
<!-- <h2 class="text-primary mb-0"><i class="fa fa-user- edit"></i></h2> -->
</a>
<a href="#" class="sidebar-toggler flex-shrink-0 bars" >
<i class="fa fa-bars"></i>
</a>
<!-- <form class="d-none d-md-flex ms-4">
<input class="form-control bg-dark border-0" type="search" placeholder="Search">
</form> -->
<div class="navbar-nav align-items-center ms-auto"><a ><span class="d-none d-lg-
inline-flex"><a href="/logout"
>LOG OUT</a></span>
<div class="nav-item dropdown">
<a href="#" class="nav-link dropdown-toggle" data-bs- toggle="dropdown">
```

```
<img class="rounded-circle me-lg-2"
src="{{url_for('static',filename='image/user.png')}} " alt="" style="width:
40px; height: 40px;">
<span class="d-none d-lg-inline-flex">{{username}}</span>
</a>
</a>
<div class="dropdown-menu dropdown-menu-end bg- secondary border-0 rounded-0
rounded-bottom m-0">

<a href="/search" class="dropdown-item">Predict</a>
<a href="/getinfo" class="dropdown-item">Get Info</a>
<a href="/ticker" class="dropdown-item">Ticker Info</a>
</div>
</div>
</div>
</nav>
<!-- Navbar End -->
<!-- Start of Main Content -->
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-10">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Information</h6>
<a href="https://finance.yahoo.com/quote/{{ticker_value}}?p={{ticker_value}}">
Show More</a>
```

```
</div>
<div class="table-responsive">
<table class="table text-start align-middle table-bordered table-hover mb-0">
<tbody>
<tr>
<td>Symbol</td>
<td>{{stock_info['symbol']}}</td>
</tr>
<tr>
<td>Name</td>
<td>{{stock_info['shortName']}}</td>
</tr>
<tr>
<td>Last Sale</td>
<td>{{stock_info['previousClose']}}</td>
</tr>
<tr>
<td>Net Change</td>
<td>{{stock_info['fiftyDayAverageChangePercent']}}</td>
</tr>
<tr>
<td>Percent Change</td>
<td>{{stock_info['fiftyDayAverageChangePercent']}}</td>
</tr>
<tr>
```

<td>Market Capital</td>
<td>{{stock_info['totalDebt']}}</td>
</tr>
<tr>
<td>city</td>
<td>{{stock_info['city']}}</td>
</tr>
<!-- <tr>
<td>IPO Year</td>
<td>{{stock_info['previousClose']}}</td>
</tr> -->
<tr>
<td>Volume</td>
<td>{{stock_info['volume']}}</td>
</tr>
<tr>
<td>Sector</td>
<td>{{stock_info['sectorKey']}}</td>
</tr>
<tr>
<td>Industry</td>
<td>{{stock_info['industry']}}</td>
</tr>
</body >
</table>

```
</div>
</div>
</div>
<!-- <div class="col-sm-12 col-xl-2">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-between mb-4">
<h6 class="mb-0">Scan the QR Code</h6>
</div>
<img src="https://api.qrserver.com/v1/create-
qrcode/?size=150x150&data=https://stock-prediction-
    system.herokuapp.com/predict/{{ticker_value}}/{{number_of_days}}/"alt="QR
    CODE">
</div>
</div> -->
</div>
</div>
<div class="container-fluid pt-4 px-4">
<div class="row g-4">
<div class="col-sm-12 col-xl-6">
<div class="bg-secondary text-center rounded p-4">
<div class="d-flex align-items-center justify-content-betweenmb-4">
<h6 class="mb-0">Recent Stock price of{{ticker_value}}</h6>
 <a
 href="https://finance.yahoo.com/quote/{{ticker_value}}/chart?p={{ticker_value}}#"
 >
```

Show More

{% block Left_Plot %}

{% autoescape off %}

{{ plot_div }}

{% endautoescape %}

{% endblock Left_Plot %}

<h6 class="mb-0">Predicted Stock price of {{ticker_value}} for next

{{number_of_days}} days</h6>

{% block Right_Plot %}

{% autoescape off %}

{{ plot_div_pred }}

{% endautoescape %}

{% endblock Right_Plot %}

```
<!-- End of Main Content -->
<!-- Footer Start -->
<div class="container-fluid pt-4 px-4">
<div class="bg-secondary rounded-top p-4">
<div class="row">
<div class="col-12 col-sm-6 text-center text-sm-start"> &copy; <a >My site</a>, All Right Reserved.
</div>
<div class="col-12 col-sm-6 text-center text-sm-end"> Designed By <a >Me</a>
</div>
</div>
</div>
</div>
<!-- Footer End -->
</div>
<!-- Content End -->
<!-- Back to Top -->
<a href="#" class="btn btn-lg btn-lg-square back-to-top" style="background: #16c4eb;"><i class="bi bi-arrow-up"></i></a>
</div>
<!-- JavaScript Libraries -->
<script  src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.8.0/chart.min.js"
```

```
integrity="sha512-
sW/w8s4RWTdFFSduOTGtk4isV1+190E/GghVffMA9XczdJ2M
DzSzLEubK As5h0wzgSJOQTRYyaz73L3d6RtJSg=="
crossorigin="anonymous" referrerpolicy="no-
referrer"></script>

<script src="

{{url_for('static',filename='lib/easing/easing.min.js')}}"></script>
<script
src="{{url_for('static',filename='lib/waypoints/waypoints.min.jsg
')}}"></scrip t>
<script src="

{{url_for('static',filename='lib/owlcarousel/owl.carousel.min.js')}}"></script>
<script src="

{{url_for('static',filename='lib/tempusdominus/js/moment.min.js')}}"></script

>
<script src="{{url_for('static', filename='lib/tempus dominus/js /moment-
timezone.min.js')}}"></script>
<script src="

{{url_for('static',filename='lib/tempusdominus/js/tempusdominus-bootstrap
                                4.min.js')}}"></script>
<!-- Template Javascript -->
<script src=" {{url_for('static', filename='js/main.js')}}"></script>
</body>
</html>
```