```python
import matplotlib.pyplot as plt
import yfinance as yf
from keras.models import load_model
import streamlit as st

st.title('PSTP: Pepperoni Stock Trend Predictor')
st.caption("To restart - press :violet[R]")
st.divider()
user_input_symbol = st.text_input('Enter stock symbol:')
user_input_start_date = st.text_input('Enter starting date in a format "yyyy-mm-dd": ')
user_input_end_date = st.text_input('Enter end date in a format "yyyy-mm-dd": ')
st.divider()

df = yf.download(user_input_symbol, start=user_input_start_date, end=user_input_end_date)

#Visualizations

#Plotting a simple closing price chart
st.subheader('Closing Price in Dollars vs Time chart')
fig = plt.figure(figsize = (12, 6), dpi = 100)
plt.plot(df['Close'], 'b')
st.pyplot(fig)

#100 Moving Average
st.subheader('Closing Price vs Time chart with 100MA')
ma100 = df['Close'].rolling(100).mean()
fig = plt.figure(figsize = (12,6))
plt.plot(ma100, 'r')
plt.plot(df['Close'], 'b')
st.pyplot(fig)

#200 Moving Average
st.subheader('Closing Price vs Time chart with 100MA & 200MA')
ma100 = df['Close'].rolling(100).mean()
ma200 = df['Close'].rolling(200).mean()
fig = plt.figure(figsize = (12,6))
plt.plot(ma100, 'r')
plt.plot(ma200, 'g')
plt.plot(df['Close'], 'b')
st.pyplot(fig)

#Splitting data into training and testing
data_training = df['Close'][:int(len(df) * 0.7)]
data_testing = df['Close'][int(len(df) * 0.7):]

#scaling function
def custom_min_max_scaler(data):
  min_val = min(data)
  max_val = max(data)
  scaled_data = [(x - min_val) / (max_val - min_val) for x in data]
  return scaled_data

data_training_array = custom_min_max_scaler(data_training.tolist())

#loading the model
model = load_model('Stock_Predictor_Model.h5')
```

```python

#testing

#using python lists for data manipulation
past_100_days = data_training.tail(100)
final_df = past_100_days.tolist() + data_testing.tolist()
input_data = custom_min_max_scaler(final_df)

x_test = []
y_test = []

for i in range(100, len(input_data)):
    x_test.append(input_data[i - 100:i])
    y_test.append(input_data[i])

# Predictions
y_predicted = model.predict(x_test)

# Scaling back to original values
min_val = min(data_training)
max_val = max(data_training)
scale_factor = max_val - min_val

y_predicted = [y * scale_factor + min_val for y in y_predicted]
y_test = [y * scale_factor + min_val for y in y_test]

# Final graph
st.subheader('Predictions vs original')
fig2 = plt.figure(figsize=(12,6))
plt.plot(y_test, 'b', label = 'Original Price')
plt.plot(y_predicted, 'r', label = 'Predicted Price')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
st.pyplot(fig2)


#--- NEWS AGGREGATOR ---#

from newsapi import NewsApiClient
import streamlit as st

api = NewsApiClient(api_key='9efcc41031a041c49a6b9f72df29194d')

st.header("News related to the company")
user_input_stock = st.text_input('Enter the name of the company: ')
number_of_articles = st.selectbox("How many articles would you like to see", (3, 5, 10, 20, 50), label_visibility="visible")
response_data = api.get_everything(qintitle=user_input_stock, sources="the-wall-street-journal, fortune, australian-financial-review, bloomberg, business-insider, business-insider-uk, financial-post", sort_by="popularity", language="en", page_size=number_of_articles)

for article in response_data['articles']:
    st.subheader(article['title'])
    st.caption(article['source']['name'] + "  •  " + article['publishedAt'][:10])
    st.markdown(f'''
<a href={article['url']}><button style="background-color:#FF33E9;">Open the
article</button></a>
''',
unsafe_allow_html=True)


# CAPTION
st.caption("Pepperoni Stock Trend Predictor would like to remind you that the data contained on this website and via API might not necessarily be real-time nor accurate. All prices may differ from the actual market price, meaning prices are indicative and not appropriate for trading purposes. Therefore, Pepperoni Stock Trend Predictor does not bear any responsibility for any trading losses user might incur as a result of using this data. Pepperoni Stock Trend Predictor or anyone involved with it will not accept any liability for loss or damage as a result of reliance on the information contained within this website. Please be fully informed regarding the risks and costs associated with trading the financial markets. Pepperoni Stock Trend Predictor does not give any warranties (including, without limitation, as to merchantability or fitness for a particular purpose or use). Please note that the historical returns summaries provided on this website are based on past prices and are not a guarantee or indication of future returns. It is important to understand that past performance is not necessarily indicative of future results, and that investing carries inherent risks. It is important to carefully consider your own financial situation and risk tolerance before making any investment decisions.")

```