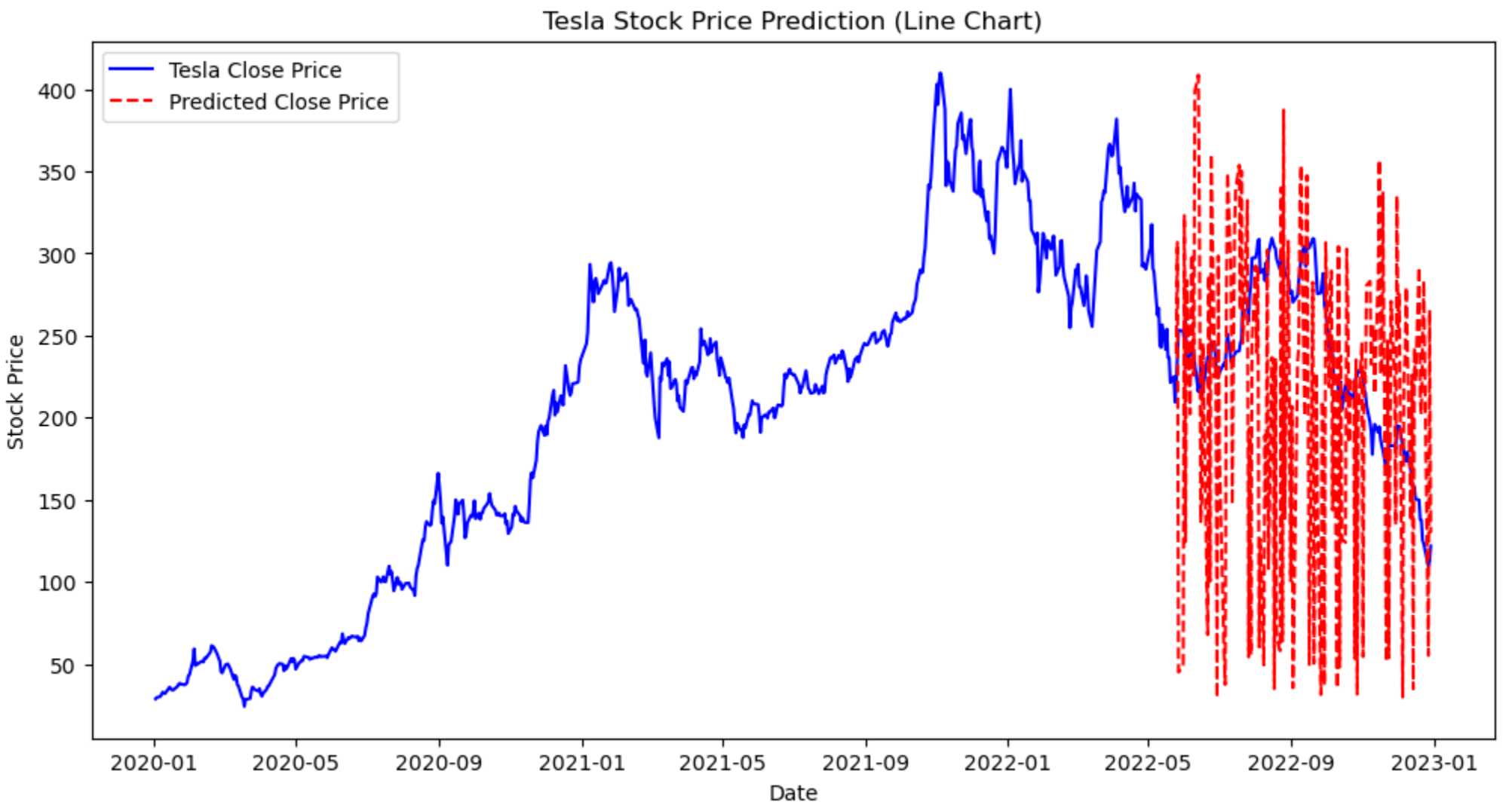
In [14]:
```
# Import necessary libraries
import pandas as pd
import yfinance as yf
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import plotly.graph_objects as go

# Download historical Tesla stock data
ticker = "TSLA"
start_date = "2020-01-01"
end_date = "2023-01-01"
tesla = yf.download(ticker, start=start_date, end=end_date)

# Prepare the data
tesla['Date'] = tesla.index
tesla.reset_index(drop=True, inplace=True)
tesla['Date'] = tesla['Date'].dt.date

# Create a feature for the 'next day' stock price
tesla['Next Close'] = tesla['Close'].shift(-1)

# Drop the last row with NaN
tesla = tesla[:-1]

# Create features (X) and target (y)
X = tesla[['Open', 'High', 'Low', 'Close', 'Volume']]
y = tesla['Next Close']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)

# Visualize the results with a line graph
plt.figure(figsize=(12, 6))
plt.plot(tesla['Date'], tesla['Close'], label='Tesla Close Price', color='blue')
plt.plot(tesla['Date'][-len(predictions):], predictions, label='Predicted Close Price', color='red', linestyle='dashed')
plt.xlabel('Date')
plt.ylabel('Stock Price')
plt.title('Tesla Stock Price Prediction (Line Chart)')
plt.legend()
plt.show()

# Create a DataFrame for predictions
predictions_df = tesla.loc[tesla.index.isin(X_test.index)].copy()
predictions_df['Predicted Close'] = predictions

# Create a candlestick chart with predicted prices
fig = go.Figure(data=[go.Candlestick(x=predictions_df['Date'],
                open=predictions_df['Open'],
                high=predictions_df['High'],
                low=predictions_df['Low'],
                close=predictions_df['Predicted Close'])])

fig.update_layout(title='Tesla Stock Price Prediction (Candlestick Chart)')
fig.show()
```

```
[**********100%%**********]  1 of 1 completed
```



In [ ]: