In [1]:
```
!pip install yfinance
#!pip install pandas
#!pip install requests
!pip install bs4
#!pip install plotly
```

```
Requirement already satisfied: yfinance in c:\users\user\anaconda3\lib\site-packages (0.2.65)
Requirement already satisfied: pandas>=1.3.0 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (1.26.4)
Requirement already satisfied: requests>=2.31 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (3.10.0)
Requirement already satisfied: pytz>=2022.5 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (2024.1)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (2.4.2)
Requirement already satisfied: peewee>=3.16.2 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (3.18.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (4.12.3)
Requirement already satisfied: curl_cffi>=0.7 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (0.12.0)
Requirement already satisfied: protobuf>=5.29.0 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (6.33.3)
Requirement already satisfied: websockets>=13.0 in c:\users\user\anaconda3\lib\site-packages (from yfinance) (15.0.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\user\anaconda3\lib\site-packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: cffi>=1.12.0 in c:\users\user\anaconda3\lib\site-packages (from curl_cffi>=0.7->yfinance) (1.17.1)
Requirement already satisfied: certifi>=2024.2.2 in c:\users\user\anaconda3\lib\site-packages (from curl_cffi>=0.7->yfinance) (2025.6.15)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\user\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in c:\users\user\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.2.3)
Requirement already satisfied: pycparser in c:\users\user\anaconda3\lib\site-packages (from cffi>=1.12.0->curl_cffi>=0.7->yfinance) (2.21)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.16.0)
Requirement already satisfied: bs4 in c:\users\user\anaconda3\lib\site-packages (0.0.2)
Requirement already satisfied: beautifulsoup4 in c:\users\user\anaconda3\lib\site-packages (from bs4) (4.12.3)
Requirement already satisfied: soupsieve>1.2 in c:\users\user\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.5)
```

In [2]:
```
!pip install html5lib
```

```
Requirement already satisfied: html5lib in c:\users\user\anaconda3\lib\site-packages (1.1)
Requirement already satisfied: six>=1.9 in c:\users\user\anaconda3\lib\site-packages (from html5lib) (1.16.0)
Requirement already satisfied: webencodings in c:\users\user\anaconda3\lib\site-packages (from html5lib) (0.5.1)
```

In [3]:
```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [4]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historical Revenue"), vertical_spacing = .3)
    fig.add_trace(go.Scatter(pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share Price"), row=1, col=1)
    fig.add_trace(go.Scatter(pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.Revenue.astype("float"), name="Revenue"), row=2, col=1)
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

In [5]:
```
tesla = yf.Ticker("TSLA")
```

In [6]:
```
tesla_data = tesla.history(period="max")
```

In [7]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

Out[7]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

In [8]:
```
url= "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data=requests.get(url).text
```

In [9]:
```
soup = BeautifulSoup(html_data,"html5lib")
```

In [12]:
```
html_data = requests.get(url, headers={"User-Agent": "Mozilla/5.0"}).text
tesla_revenue = pd.read_html(html_data, match="Tesla Quarterly Revenue", flavor="bs4")[0]

# Check columns
print(tesla_revenue.columns)

# Rename correctly
tesla_revenue = tesla_revenue.rename(columns={
    tesla_revenue.columns[0]: 'Date',
    tesla_revenue.columns[1]: 'Revenue'
})

# Now clean Revenue
tesla_revenue["Revenue"] = tesla_revenue["Revenue"].str.replace(",", "").str.replace("$", "")
tesla_revenue.head()
```

```
Index(['Tesla Quarterly Revenue (Millions of US $)', 'Tesla Quarterly Revenue (Millions of US $).1'], dtype='object')
C:\Users\user\AppData\Local\Temp\ipykernel_17912\1899627132.py:2: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.
  tesla_revenue = pd.read_html(html_data, match="Tesla Quarterly Revenue", flavor="bs4")[0]
```

Out[12]:

	Date	Revenue
0	2025-03-31	19335
1	2024-12-31	25707
2	2024-09-30	25182
3	2024-06-30	25500
4	2024-03-31	21301

In [13]:
```
tesla_revenue
```

Out[13]:

	Date	Revenue
0	2025-03-31	19335
1	2024-12-31	25707
2	2024-09-30	25182
3	2024-06-30	25500
4	2024-03-31	21301
...	...	...
59	2010-06-30	28
60	2010-03-31	21
61	2009-12-31	NaN
62	2009-09-30	46
63	2009-06-30	27

64 rows × 2 columns

In [14]:
```
tesla_revenue.dropna(inplace=True)
tesla_revenue.tail()
```

Out[14]:

	Date	Revenue
58	2010-09-30	31
59	2010-06-30	28
60	2010-03-31	21
62	2009-09-30	46
63	2009-06-30	27

In [15]:
```
gamestop = yf.Ticker("GME")
```

In [16]:
```
gme_data=gamestop.history(period="max")
```

In [17]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[17]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712708	1.716074	1.670626	1.683251	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0

In [21]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url, headers={"User-Agent": "Mozilla/5.0"}).text

gme_revenue = pd.read_html(html_data, match="GameStop Quarterly Revenue", flavor="bs4")[0]

# Print columns to confirm
print(gme_revenue.columns)

# Rename columns using index instead of string matching
gme_revenue = gme_revenue.rename(columns={
    gme_revenue.columns[0]: 'Date',
    gme_revenue.columns[1]: 'Revenue'
})

# Clean the revenue column
gme_revenue["Revenue"] = gme_revenue["Revenue"].str.replace(",", "").str.replace("$", "")
gme_revenue.head()
```

```
Index(['GameStop Quarterly Revenue (Millions of US $)', 'GameStop Quarterly Revenue (Millions of US $).1'], dtype='object')
C:\Users\user\AppData\Local\Temp\ipykernel_17912\1934032330.py:4: FutureWarning: Passing literal html to 'read_html' is deprecated and will be removed in a future version. To read from a literal string, wrap it in a 'StringIO' object.
  gme_revenue = pd.read_html(html_data, match="GameStop Quarterly Revenue", flavor="bs4")[0]
```

Out[21]:

	Date	Revenue
0	2025-04-30	732
1	2025-01-31	1283
2	2024-10-31	860
3	2024-07-31	796
4	2024-04-30	882

In [22]:
```
gme_revenue.dropna(inplace=True)
gme_revenue.tail()
```

Out[22]:

	Date	Revenue
61	2010-01-31	3524
62	2009-10-31	1835
63	2009-07-31	1739
64	2009-04-30	1981
65	2009-01-31	3492

In [23]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla Stock Data Graph')
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_17912\1276540637.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future version. A strict version of it is now the default, see https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You can safely remove this argument.

C:\Users\user\AppData\Local\Temp\ipykernel_17912\1276540637.py:4: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future version. A strict version of it is now the default, see https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You can safely remove this argument.
```


In [24]:
```
make_graph(gme_data, gme_revenue, 'GameStop Stock Data Graph')
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_17912\1276540637.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future version. A strict version of it is now the default, see https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You can safely remove this argument.

C:\Users\user\AppData\Local\Temp\ipykernel_17912\1276540637.py:4: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future version. A strict version of it is now the default, see https://pandas.pydata.org/pdeps/0004-consistent-to-datetime-parsing.html. You can safely remove this argument.
```