In [1]:
```
!python -m pip install yfinance
!python -m pip install pandas
!python -m pip install requests
!python -m pip install bs4
!python -m pip install plotly
!mamba install html5lib==1.1 -y
```

```
Requirement already satisfied: yfinance in c:\users\sham3oona\anaconda3\lib\site-package
s (0.2.40)
Requirement already satisfied: pandas>=1.3.0 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from yfinance) (2.0.3)
Requirement already satisfied: numpy>=1.16.5 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from yfinance) (1.24.3)
Requirement already satisfied: requests>=2.31 in c:\users\sham3oona\anaconda3\lib\site-p
ackages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\sham3oona\anaconda3\lib\s
ite-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in c:\users\sham3oona\anaconda3\lib\site-pack
ages (from yfinance) (4.9.3)
Requirement already satisfied: platformdirs>=2.0.0 in c:\users\sham3oona\anaconda3\lib\s
ite-packages (from yfinance) (3.10.0)
Requirement already satisfied: pytz>=2022.5 in c:\users\sham3oona\anaconda3\lib\site-pac
kages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\sham3oona\anaconda3\lib\sit
e-packages (from yfinance) (2.4.4)
Requirement already satisfied: peewee>=3.16.2 in c:\users\sham3oona\anaconda3\lib\site-p
ackages (from yfinance) (3.17.5)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\sham3oona\anaconda3\li
b\site-packages (from yfinance) (4.12.2)
Requirement already satisfied: html5lib>=1.1 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from beautifulsoup4>=4.11.1->yfinance) (2.4)
Requirement already satisfied: six>=1.9 in c:\users\sham3oona\anaconda3\lib\site-package
s (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\users\sham3oona\anaconda3\lib\site-pac
kages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sham3oona\anaconda3\li
b\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in c:\users\sham3oona\anaconda3\lib\site-p
ackages (from pandas>=1.3.0->yfinance) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sham3oona\anaconda3
\lib\site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\sham3oona\anaconda3\lib\site-pac
kages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sham3oona\anaconda3\lib\si
te-packages (from requests>=2.31->yfinance) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sham3oona\anaconda3\lib\si
te-packages (from requests>=2.31->yfinance) (2023.7.22)
Requirement already satisfied: pandas in c:\users\sham3oona\anaconda3\lib\site-packages
(2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\sham3oona\anaconda3\li
b\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\sham3oona\anaconda3\lib\site-pac
kages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\sham3oona\anaconda3\lib\site-p
ackages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\sham3oona\anaconda3\lib\site-package
s (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: requests in c:\users\sham3oona\anaconda3\lib\site-package
s (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sham3oona\anaconda3
```

```
\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\sham3oona\anaconda3\lib\site-pac
kages (from requests) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sham3oona\anaconda3\lib\si
te-packages (from requests) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\sham3oona\anaconda3\lib\si
te-packages (from requests) (2023.7.22)
Requirement already satisfied: bs4 in c:\users\sham3oona\anaconda3\lib\site-packages (0.
0.2)
Requirement already satisfied: beautifulsoup4 in c:\users\sham3oona\anaconda3\lib\site-p
ackages (from bs4) (4.12.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\sham3oona\anaconda3\lib\site-pa
ckages (from beautifulsoup4->bs4) (2.4)
Requirement already satisfied: plotly in c:\users\sham3oona\anaconda3\lib\site-packages
(5.22.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\sham3oona\anaconda3\lib\site-
packages (from plotly) (8.2.2)
Requirement already satisfied: packaging in c:\users\sham3oona\anaconda3\lib\site-packag
es (from plotly) (23.1)
'mamba' is not recognized as an internal or external command,
operable program or batch file.
```

In [2]:
```
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 [3]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical S
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=Tru
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=T
    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 [4]:
```
tesla = yf.Ticker('TSLA')
```

In [5]:
```
tesla_data = tesla.history(period="max")
tesla_data
```

Out[5]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0
...	...	...	...	...	...	...	...
2024-06-13 00:00:00-04:00	188.389999	191.080002	181.229996	182.470001	118984100	0.0	0.0

2024-06-14 00:00:00-04:00	185.800003	186.000000	176.919998	178.009995	82038200	0.0	0.0
2024-06-17 00:00:00-04:00	177.919998	188.809998	177.000000	187.440002	109786100	0.0	0.0
2024-06-18 00:00:00-04:00	186.559998	187.199997	182.369995	184.860001	68982300	0.0	0.0
2024-06-20 00:00:00-04:00	184.679993	185.210007	179.669998	181.570007	54953255	0.0	0.0

3518 rows × 7 columns

Question 1: Use yfinance to Extract Stock Data

In [6]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head(5)
```

Out[6]:

	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

Question 2: Use Webscraping to Extract Tesla Revenue Data

In [7]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

In [8]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<html>
 <head>
 </head>
 <body>
  <div style="margin: 50px auto; width: 50%; border: 1px solid #dfdfdf; padding: 20px 50
px 30px 50px; font-family:helvetica;">
   <h1>
    We do not allow automated access to our servers.
   </h1>
   <h2>
    <p>
     Automated access to our data is prohibited by our data provider.
    </p>
    <p>
     If you are a user attempting to access the site via a browser, please follow this p
rocess to regain access:
    </p>
    <ul>
     <li>
      Go to
      <a href="https://whatismyipaddress.com/" rel="noopener noreferrer" target="_blan
k">
       whatismyipaddress
      </a>
      and obtain your IPv4 address
     </li>
     <li>
      Email us your IPv4 address at
      <a class="__cf_email__" data-cfemail="4821262e270825292b3a273c3a2d262c3b66262d3c"
```

```
href="/cdn-cgi/l/email-protection">
       [email protected]
      </a>
     </li>
     <li>
      We will add you to our whitelist within 24 hours
     </li>
    </ul>
   </h2>
  </div>
  <script data-cfasync="false">
   !function(){"use strict";function e(e){try{if("undefined"==typeof console)return;"err
or"in console?console.error(e):console.log(e)}catch(e){}}function t(e,t){var r=e.substr
(t,2);return parseInt(r,16)}function r(r,n){for(var c="",o=t(r,n),a=n+2;a<r.length;a+=2)
{var l=t(r,a)^o;c+=String.fromCharCode(l)}try{c=decodeURIComponent(escape(c))}catch(t){e
(t)}return function(e){return i.innerHTML='<a href="'+e.replace(/"/g,"&quot;")+'"></a>',
i.childNodes[0].getAttribute("href")||""}(c)}function n(t){try{(function(t){for(var n=t.
querySelectorAll("a"),o=0;o<n.length;o++)try{var a=n[o],i=a.href.indexOf(c);i>-1&&(a.hre
f="mailto:"+r(a.href,i+c.length))}catch(t){e(t)}})(t),function(t){for(var n=t.querySelec
torAll(o),c=0;c<n.length;c++)try{var i=n[c],l=i.parentNode,u=i.getAttribute(a);if(u){var
f=r(u,0),d=document.createTextNode(f);l.replaceChild(d,i)}}catch(t){e(t)}}(t),function
(t){for(var r=t.querySelectorAll("template"),c=0;c<r.length;c++)try{n(r[c].content)}catc
h(t){e(t)}}(t)}catch(t){e(t)}}var c="/cdn-cgi/l/email-protection#",o="__cf_email__",a
="data-cfemail",i=document.createElement("div");n(document),function(){var e=document.cu
rrentScript||document.scripts[document.scripts.length-1];e.parentNode.removeChild(e)}()}
();
  </script>
  <script>
   (function(){function c(){var b=a.contentDocument||a.contentWindow.document;if(b){var
d=b.createElement('script');d.innerHTML="window.__CF$cv$params={r:'896e764e5e733c71',
t:'MTcxODkxNDg1NC4wMDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/
cdn-cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head')
[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(document.body)
{var a=document.createElement('iframe');a.height=1;a.width=1;a.style.position='absolut
e';a.style.top=0;a.style.left=0;a.style.border='none';a.style.visibility='hidden';docume
nt.body.appendChild(a);if('loading'!==document.readyState)c();else if(window.addEventLis
tener)document.addEventListener('DOMContentLoaded',c);else{var e=document.onreadystatech
ange||function(){};document.onreadystatechange=function(b){e(b);'loading'!==document.rea
dyState&&(document.onreadystatechange=e,c())}}})();
  </script>
 </body>
</html>
```

In [9]:
```
tesla_revenue = pd.DataFrame(columns = ['Date', 'Revenue'])

for row in soup.find_all("body")[0].find_all("tr"):
    col = row.find_all("script")
    date = col[0].text
    revenue = col[1].text.replace("$", "").replace(",", "")

    tesla_revenue = tesla_revenue.append({"Date": date, "Revenue": revenue}, ignore_inde
```

In [10]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

In [11]:
```
tesla_revenue.dropna(axis=0, how='all', subset=['Revenue']) #drop NaN values
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""] #drop empty string values
```

In [12]:
```
tesla_revenue.tail(5)
```

Out[12]:

	Date	Revenue

Question 3: Use yfinance to Extract Stock Data

In [13]:
```
Gm = yf.Ticker('GME')
```

In [14]:
```
gme_data = Gm.history(period = "max")
```

In [15]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[15]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620129	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712707	1.716073	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683251	1.687459	1.658002	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	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

In [16]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

In [17]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<html>
 <head>
 </head>
 <body>
  <div style="margin: 50px auto; width: 50%; border: 1px solid #dfdfdf; padding: 20px 50
px 30px 50px; font-family:helvetica;">
   <h1>
    We do not allow automated access to our servers.
   </h1>
   <h2>
    <p>
     Automated access to our data is prohibited by our data provider.
    </p>
    <p>
     If you are a user attempting to access the site via a browser, please follow this p
rocess to regain access:
    </p>
    <ul>
     <li>
      Go to
      <a href="https://whatismyipaddress.com/" rel="noopener noreferrer" target="_blan
k">
       whatismyipaddress
      </a>
      and obtain your IPv4 address
     </li>
     <li>
      Email us your IPv4 address at
      <a class="__cf_email__" data-cfemail="4821262e270825292b3a273c3a2d262c3b66262d3c"
href="/cdn-cgi/l/email-protection">
       [email protected]
      </a>
     </li>
     <li>
      We will add you to our whitelist within 24 hours
     </li>
    </ul>
```

```
   </h2>
  </div>
  <script data-cfasync="false">
   !function(){"use strict";function e(e){try{if("undefined"==typeof console)return;"err
or"in console?console.error(e):console.log(e)}catch(e){}}function t(e,t){var r=e.substr
(t,2);return parseInt(r,16)}function r(r,n){for(var c="",o=t(r,n),a=n+2;a<r.length;a+=2)
{var l=t(r,a)^o;c+=String.fromCharCode(l)}try{c=decodeURIComponent(escape(c))}catch(t){e
(t)}return function(e){return i.innerHTML='<a href="'+e.replace(/"/g,"&quot;")+'"></a>',
i.childNodes[0].getAttribute("href")||""}(c)}function n(t){try{(function(t){for(var n=t.
querySelectorAll("a"),o=0;o<n.length;o++)try{var a=n[o],i=a.href.indexOf(c);i>-1&&(a.hre
f="mailto:"+r(a.href,i+c.length))}catch(t){e(t)}})(t),function(t){for(var n=t.querySelec
torAll(o),c=0;c<n.length;c++)try{var i=n[c],l=i.parentNode,u=i.getAttribute(a);if(u){var
f=r(u,0),d=document.createTextNode(f);l.replaceChild(d,i)}}catch(t){e(t)}}(t),function
(t){for(var r=t.querySelectorAll("template"),c=0;c<r.length;c++)try{n(r[c].content)}catc
h(t){e(t)}}(t)}catch(t){e(t)}}var c="/cdn-cgi/l/email-protection#",o="__cf_email__",a
="data-cfemail",i=document.createElement("div");n(document),function(){var e=document.cu
rrentScript||document.scripts[document.scripts.length-1];e.parentNode.removeChild(e)}()}
();
  </script>
  <script>
   (function(){function c(){var b=a.contentDocument||a.contentWindow.document;if(b){var
d=b.createElement('script');d.innerHTML="window.__CF$cv$params={r:'896e7655eea6d506',
t:'MTcxODkxNDg1NS4wMDAwMDA='};var a=document.createElement('script');a.nonce='';a.src='/
cdn-cgi/challenge-platform/scripts/jsd/main.js';document.getElementsByTagName('head')
[0].appendChild(a);";b.getElementsByTagName('head')[0].appendChild(d)}}if(document.body)
{var a=document.createElement('iframe');a.height=1;a.width=1;a.style.position='absolut
e';a.style.top=0;a.style.left=0;a.style.border='none';a.style.visibility='hidden';docume
nt.body.appendChild(a);if('loading'!==document.readyState)c();else if(window.addEventLis
tener)document.addEventListener('DOMContentLoaded',c);else{var e=document.onreadystatech
ange||function(){};document.onreadystatechange=function(b){e(b);'loading'!==document.rea
dyState&&(document.onreadystatechange=e,c())}}})();
  </script>
 </body>
</html>
```

In [18]:
```
Gm_revenue = pd.DataFrame(columns = ["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("GameStop Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

            Gm_revenue = Gm_revenue.append({"Date":Date, "Revenue":Revenue}, ignore_inde
```

In [19]:
```
Gm_revenue.tail()
```

Out[19]:

	Date	Revenue

Question 5: Plot Tesla Stock Graph

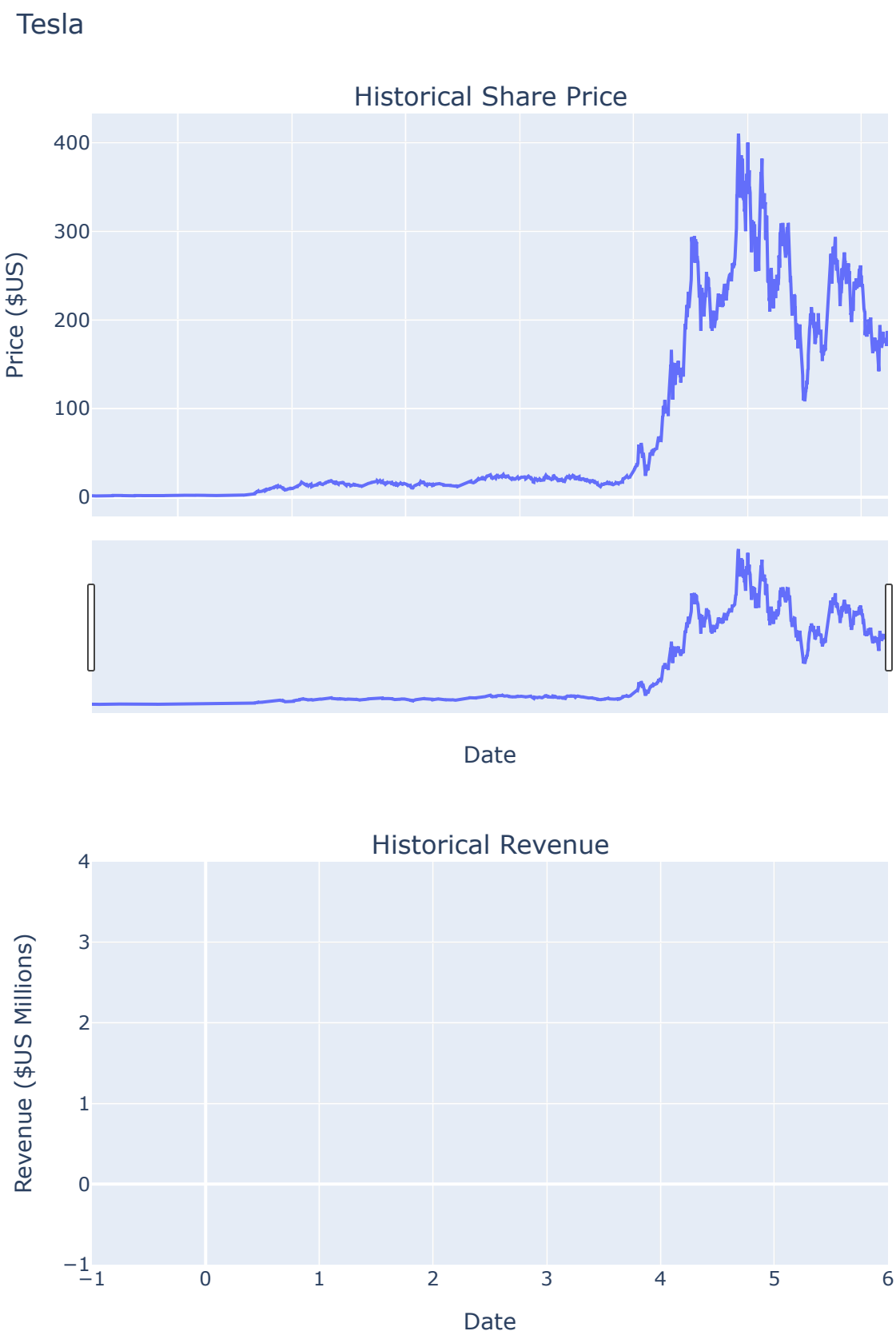
In [20]:
```
make_graph(tesla_data, tesla_revenue, "Tesla")
```

```
C:\Users\sham3oona\AppData\Local\Temp\ipykernel_14740\1276540637.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future versi
on. 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\sham3oona\AppData\Local\Temp\ipykernel_14740\1276540637.py:4: UserWarning:
```

```
The argument 'infer_datetime_format' is deprecated and will be removed in a future versi
on. 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.
```



Question 6: Plot GameStop Stock Graph

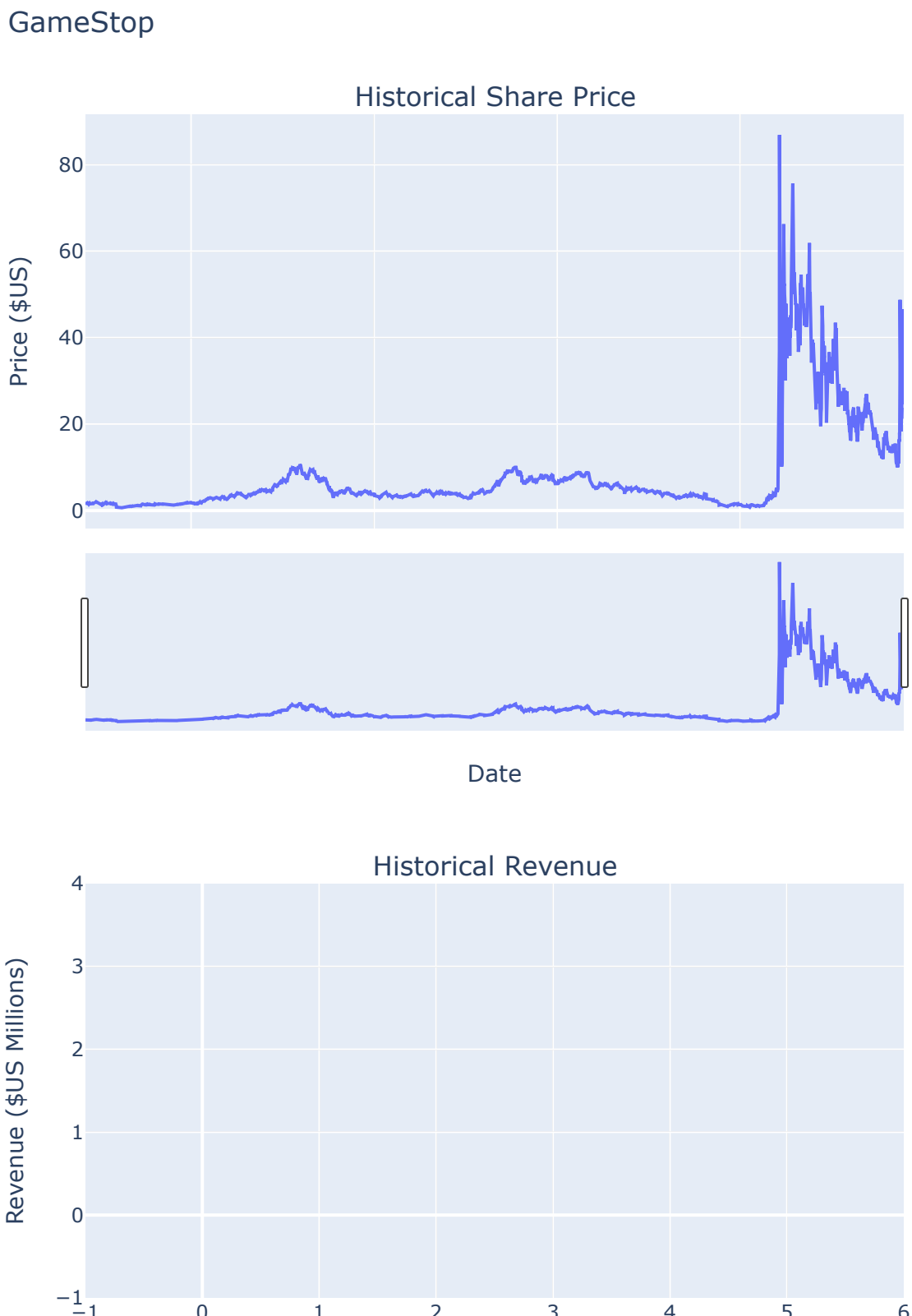
In [21]:
```
make_graph(gme_data, Gm_revenue, 'GameStop')
```

```
C:\Users\sham3oona\AppData\Local\Temp\ipykernel_14740\1276540637.py:3: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future versi
on. 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\sham3oona\AppData\Local\Temp\ipykernel_14740\1276540637.py:4: UserWarning:

The argument 'infer_datetime_format' is deprecated and will be removed in a future versi
on. 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.
```



Date

In [ ]: