Here are the answers:

## Question 1

To extract Tesla stock data using yfinance, you can use the following code:

Python

import yfinance as yf

tesla\_data = yf.Ticker("TSLA")

tesla\_stock\_data = tesla\_data.history(period="max")

print(tesla\_stock\_data)

## Question 2

To extract Tesla revenue data using web scraping, you can use the following code:

Python

import requests

from bs4 import BeautifulSoup

url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

revenue\_data = soup.find\_all('table')[1]

for row in revenue\_data.find\_all('tr'):

print([cell.text for cell in row.find\_all('td')])

## Question 3

To extract GameStop stock data using yfinance, you can use the following code:

Python

import yfinance as yf

gamestop\_data = yf.Ticker("GME")

gamestop\_stock\_data = gamestop\_data.history(period="max")

print(gamestop\_stock\_data)

## Question 4

To extract GameStop revenue data using web scraping, you can use the following code:

Python

import requests

from bs4 import BeautifulSoup

url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

revenue\_data = soup.find\_all('table')[1]

for row in revenue\_data.find\_all('tr'):

print([cell.text for cell in row.find\_all('td')])

## Question 5

To create a Tesla stock and revenue dashboard, you can use the following code:

Python

import dash

import dash\_core\_components as dcc

import dash\_html\_components as html

from dash.dependencies import Input, Output

import plotly.express as px

import pandas as pd

tesla\_data = yf.Ticker("TSLA")

tesla\_stock\_data = tesla\_data.history(period="max")

url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

revenue\_data = soup.find\_all('table')[1]

tesla\_revenue\_data = []

for row in revenue\_data.find\_all('tr'):

tesla\_revenue\_data.append([cell.text for cell in row.find\_all('td')])

tesla\_revenue\_df = pd.DataFrame(tesla\_revenue\_data[1:], columns=tesla\_revenue\_data[0])

app = dash.Dash(\_\_name\_\_)

app.layout = html.Div([

html.H1('Tesla Stock and Revenue Dashboard'),

dcc.Dropdown(

id='dropdown',

options=[

{'label': 'Stock Price', 'value': 'stock'},

{'label': 'Revenue', 'value': 'revenue'}

],

value='stock'

),

dcc.Graph(id='graph')

])

@app.callback(

Output('graph', 'figure'),

Input('dropdown', 'value')

)

def update\_graph(selected\_value):

if selected\_value == 'stock':

fig = px.line(tesla\_stock\_data, title='Tesla Stock Price')

elif selected\_value == 'revenue':

fig = px.bar(tesla\_revenue\_df, x='Date', y='Revenue', title='Tesla Revenue')

return fig

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server()

## Question 6

To create a GameStop stock and revenue dashboard, you can use the following code:

Python

import dash

import dash\_core\_components as dcc

import dash\_html\_components as html

from dash.dependencies import Input, Output

import plotly.express as px

import pandas as pd

gamestop\_data = yf.Ticker("GME")

gamestop\_stock\_data = gamestop\_data.history(period="max")

url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

revenue\_data = soup.find\_all('table')[1]

gamestop\_revenue\_data = []

for row in revenue\_data.find\_all('tr'):

gamestop\_revenue\_data.append([cell.text for cell in row.find\_all('td')])

gamestop\_revenue\_df = pd.DataFrame(gamestop\_revenue\_data[1:], columns=gamestop\_revenue\_data[0])

app = dash.Dash(\_\_name\_\_)

app.layout = html.Div([

html.H1('GameStop Stock and Revenue Dashboard'),

dcc.Dropdown(

id='dropdown',