Assignment II

Design and Development of Electronic Stock Exchange

Stock trading broadly refers to any buying and selling of stock, but is colloquially used to refer to more short-term investments made by very active investors. Most stocks are traded on physical or virtual exchanges. The New York Stock Exchange (NYSE), for example, is a physical exchange where some trades are placed manually on a trading floor—yet, other trading activity is conducted electronically. NASDAQ, on the other hand, is a fully electronic exchange where all trading activity occurs over an extensive computer network, matching investors from around the world with each other in the blink of an eye. Investors and traders submit orders to buy and sell shares, either through a broker or by using an online platform such as an E*Trade.

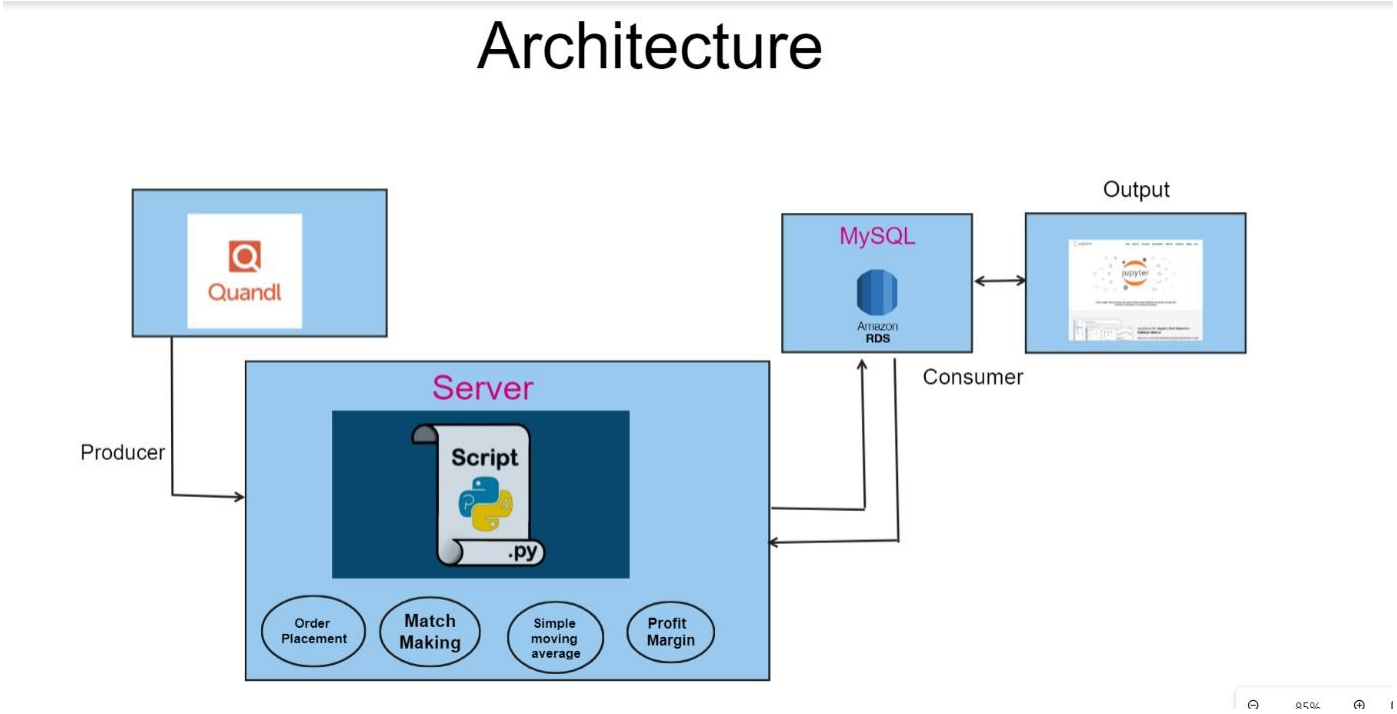
A buyer bids to purchase shares at a specified price (or at the best available price) and a seller asks to sell the stock at a specified price (or at the best available price). When a bid and an ask match, a transaction occurs and both orders will be filled. In a very liquid market, the orders will be filled almost instantaneously. In a thinly traded market, however, the order may not be filled quickly or at all. On an electronic exchange, such as NASDAQ, buyers and sellers are matched electronically. Market makers (similar in function to the specialists at the physical exchanges) provide bid and ask prices, facilitate trading in certain security, match buy and sell orders, and use their own inventory of shares, if necessary.

Active trading is when an investor who places 10 or more trades per month. They often use strategies that rely heavily on timing the market. They try to take advantage of short-term events (at the company or in the market) to turn a short-term profit. Day trading means playing hot potato with stocks — buying and selling the same stock in a single trading day. Day traders care little about the inner workings of the businesses. They try to make a few bucks in the next few minutes, hours or days based on daily price swings.

You have to design and implement an electronic exchange where day trading is facilitated. Make necessary assumptions while designing and implementing such a system and ensure that you have carefully drafted out the critical design decisions.

Exercise 1: Architecture [3 marks]

Provide a suitable architecture diagram with appropriate description matching the system specification described in the following exercises.



The architecture diagram for the electronic exchange system will have the following components:

1.) Order Producer:Thiscomponent isresponsible for receiving the orders from traders and validating them based on the instruments available onthe exchange.

2.) Match Maker:This component will receivethe orders from the Order Producer and apply a match-making algorithm to complete the trades forthe instruments

3.) Simple MovingAverage Calculator:This component will receive the trades from the Match Maker and calculatethe simple moving average closing price for the instruments in a 5-minute sliding

window for the last 10 minutes.

4.) Profit Calculator:This component will receive the closing pricesfrom the Simple Moving Average Calculator and calculate the profit for each instrument in a 5-minute sliding window for the last 10 minutes.

5.) Streaming Data Pipeline:Thiscomponent will integratethe various components mentioned above to process the real-time data of trader's orders, performnecessary pre-processing,and produce the trades and analyticsthat can be sent tothe traders' mobile devices.

6.) Data Store: This component willstorethe intermediate data generated duringthe order placement and any transactions made.

The streaming data pipeline architecture

Server(API-Quandl) --> Stock Data(Python) <--> Buy/Sell Orders (AWS-RDS-MYSQL Database) <--> Jupyter Notebook UI

Components used and purposes of the same

Quandl:

Quandl is a platform that provides its users with economic, financial and alternative datasets. Users can download free data, buy paid data or sell data to Quandl. In this guide, we will cover how to extract data using Excel and Python (separately).

Python:

Python's standard library is very extensive, offering a wide range of facility to plot the graphs and visualize the data…

Amazon RDS:

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud.

Description:

Lets assume that there is data source Quandl UI, We are extracting the Live data of stock market from Quandl.

This data which we are extracting we are putting in Server , where we are running python scripts and these python scripts place the buy and sell order which is getting stored in Amazon RDS – My

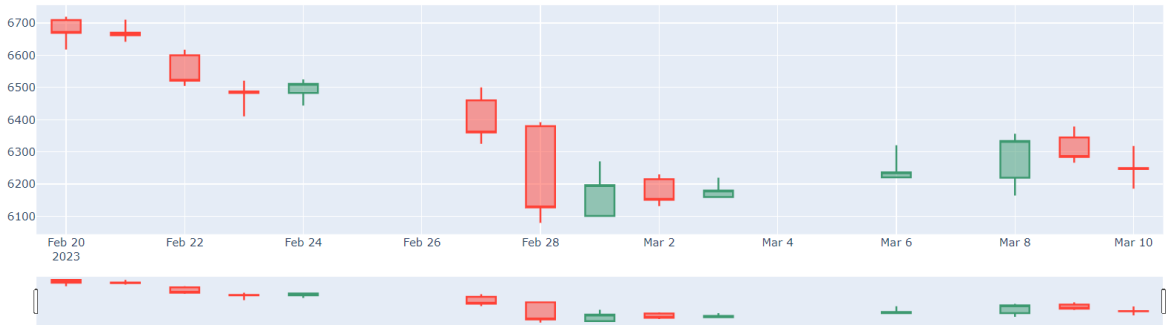
SQL Database and this process is bidirectional as we are extracting as well as dumping the data together.

These python scripts which are there in the server they have different functionalities which we have added in our exercise

2 (Placing the order) - When user is placing an order of either Buy or Sell, We are checking followings, Transaction should be on same Date.

```
p1.get_plot_from_quandl(input("Start Date:"),input("End Date:"))
```

```
Start Date:2023-02-20
End Date:2023-03-10
```



3(Match Making) - If placing order of Buy, Sell order should be present and vice versa.

```
result_dataFrame.head(10)
```

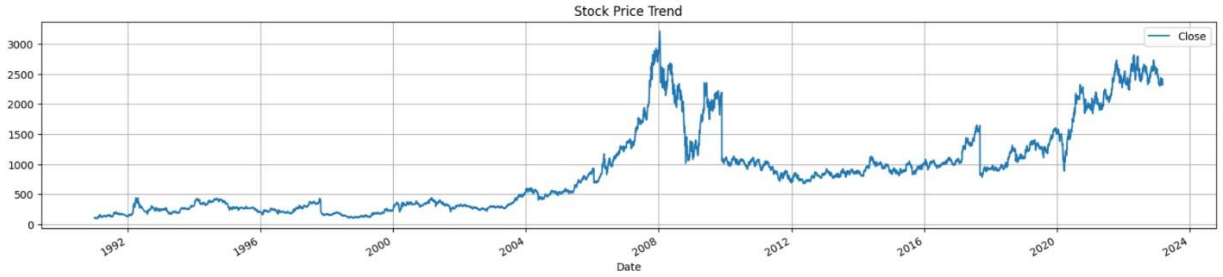
```
C:\Users\Abhishek Jaiswal\AppData\Local\Temp\ipykernel_9856\2398559311.py:8: UserWarning:

pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
```

	Date	user_id	BSE_Code	Price	Quantity	Transaction_Type
0	2023-03-03	Ela	BSE/BOM500325	2386	100	Sell
1	2023-03-10	Ela	BSE/BOM500325	2386	220	Sell
2	2023-03-10	Lipika	BSE/BOM500325	2386	50	Buy
3	2023-03-10	Abhishek	BSE/BOM500408	2386	500	Sell
4	2023-03-10	Shiva	BSE/BOM500408	6247	55	Buy

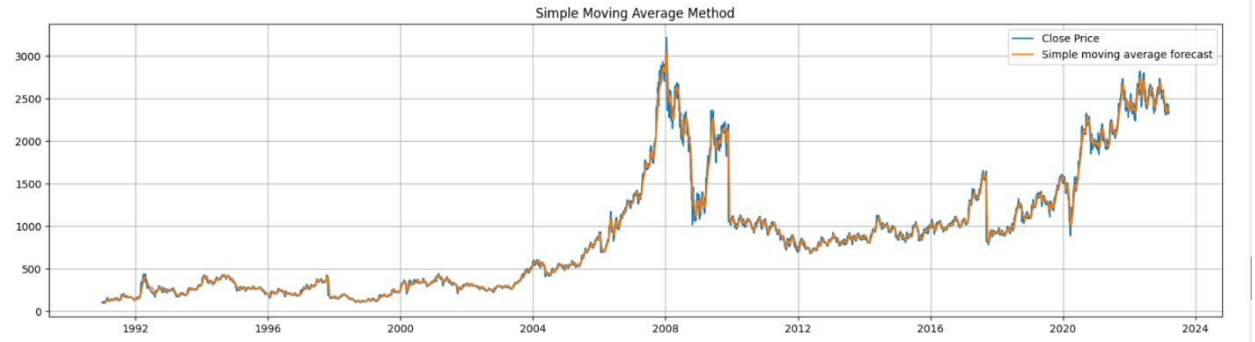
4(Simple Moving Average) - SMA is the easiest moving average to construct. It is simply the average price over the specified period. The average is called "moving" because it is plotted on the chart bar by bar, forming a line that moves along the chart as the average value changes.
Formula : A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average.

```
#Plot time series data
import matplotlib.pyplot as plt
%matplotlib inline
data = stdata[['Date','Close']]
data = data.set_index('Date')
data.plot(figsize=(20, 4))
plt.grid()
plt.legend(loc='best')
plt.title('Stock Price Trend')
plt.show(block=False)
```




```
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
plt.figure(figsize=(20,5))
plt.grid()
plt.plot(data['Close'], label='Close Price')
plt.plot(y_hat_sma['sma_forecast'], label='Simple moving average forecast')
plt.legend(loc='best')
plt.title('Simple Moving Average Method')
plt.show()
```



5(Profit Margin Calculations) – We have taken four instruments giving maximum profit (average closing price - average opening price) in a 5-days sliding window for the last 30 days.

```
    #Profit calculation
    stdata = stdata.fillna(0)
    stdata['Profit_Lose_Booking'] = stdata['Close_Sma'] - stdata['Open_Sma']
    print("Stock code: ",stdata['BSE_Code'].unique(), "Made booking of: ",stdata['Profit_Lose_Booking'][-1:].values[0])
```

```
Stock code:  ['BSE/BOM500325'] Made booking of:  -2.0
Stock code:  ['BSE/BOM500408'] Made booking of:  17.0
Stock code:  ['BSE/BOM500470'] Made booking of:  0.3999999999999915
Stock code:  ['BSE/BOM500483'] Made booking of:  -1.0
```

```
#Explanation:
```

Basically, the server is running all these python scripts.
All these activities whoch are being performed is saved in Amazon RDS – MySQL Database and post that when you perform the activities of placing the order or simple moving averages etc, this all we are displaying in the form of charts or plots in the jupyterlab..

Explanation:

Architecture and Tech Stack: The electronic exchange will have a distributed architecture to handle large volumes of data and provide high availability. The tech stack will include the following components:

- Real-time streaming data extraction: Quandl API
- Data's intermittent storage: Amazon RDS
- Data preprocessing: Python

    - Business logic for placing the offers: When user is placing an order of either Buy or Sell, we are checking followings, Transaction should be on same Date and If placing order of Buy, sell order should be present and vice versa.
    - Final representation of the outcome: We have created Real Time Stock order placing Process via Python Script, AWS RDS and Quandl.

Business logic used:
We have used following components like Quandl, Python and AWS RDS due to following reasons:
#1. Able to extract stocks data at real time from Quandl
#2. Perform Data Preprocessing, Cleaning and logic creation via Python for its high efficiency
#3. Used Amazon AWS RDS for data storage for transaction data for stability and high performance

#With above tech stacks we are able to perform following items:
#1. Stock Analysis via Candle Plot, Historical Data, etc
#2. Perform stock forecasting via Simple Moving Average to check if stock will go up or down
#3. Stocks profits analysis via SMA for right stock order placement from list of stocks.

Front end: HTML/CSS/Js
https://puzzledpalegreenemulators.lipikasharma.repl.co/