density

HACK-AN-INTERN

28th October 22

Problem Statement:

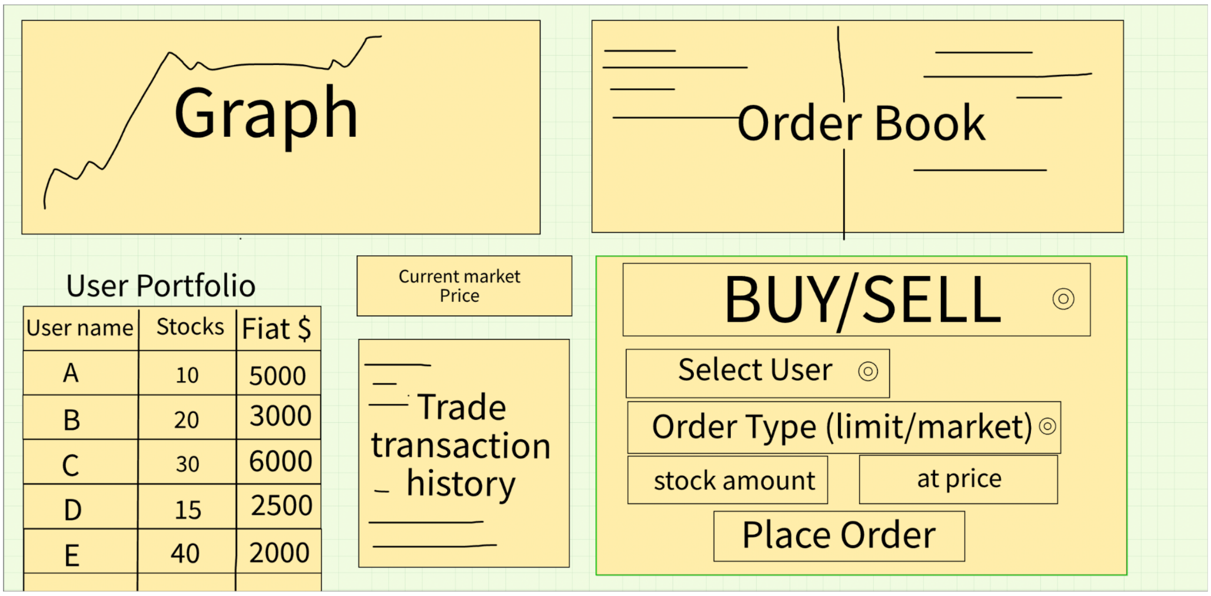
Designing stock market auction mechanism

Reference you can use:
https://www.youtube.com/watch?v=mUvB89DMtww&ab_channel=CriticalTrading
https://www.youtube.com/watch?v=9A7a7Fk2F9g&ab_channel=CriticalTrading

Features required:

- OrderBook
- Graph
- Users portfolio
- Executed trades history
- Current Market Price
- Notifications

Basic flow of screen:



User Portfolio

User name	Stocks	Fiat $
A	10	5000
B	20	3000
C	30	6000
D	15	2500
E	40	2000

Each components:

User Portfolio:

This is the starting point of the application where we will assign each user some assets (stocks and fiat). For the scope of the hackathon we will be having only 5-10 fixed users (say A,B,C,D,E,F…) whose assets portfolio will be visible/editable and will change in real time if that user makes a trade.

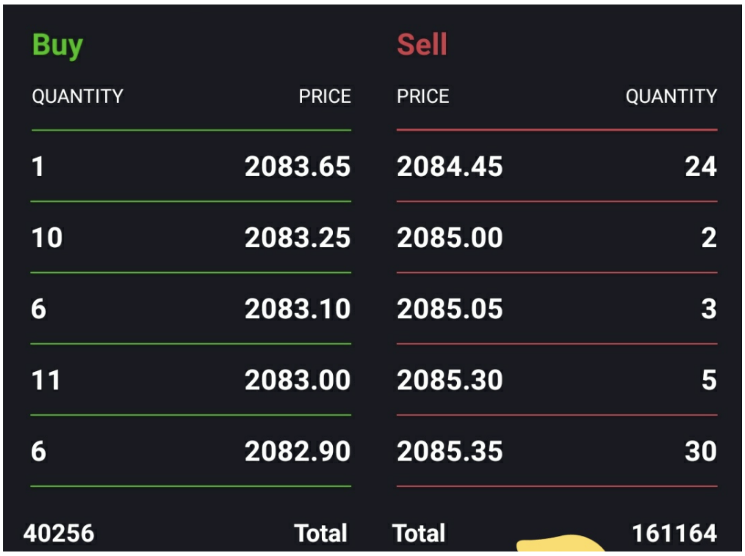
Buy/Sell control:

This single pane control should allow us to place a trade order (buy/sell) for any user we select.

- First dropdown will allow us to select if a user wants to place a buy or sell order.
- Second dropdown will allow us to select user (from the fixed pool of users who we have assigned assets)
- Third dropdown will be an option to select order type (market order or limit order)
- Next we will set amount of stocks to buy or sell and a price (in case of limit order)
- Now with a single click, after confirmation, an order should be placed validating the asset portfolio of the user. The placed order should be immediately visible in the order book.

Order Book:

- This contains a list of all placed pending orders in two columns buy and sell.

- 

Buy		Sell	
QUANTITY	PRICE	PRICE	QUANTITY
1	2083.65	2084.45	24
10	2083.25	2085.00	2
6	2083.10	2085.05	3
11	2083.00	2085.30	5
6	2082.90	2085.35	30
40256	Total	Total	161164

- You have liberty to choose the best or basic order books to implement.

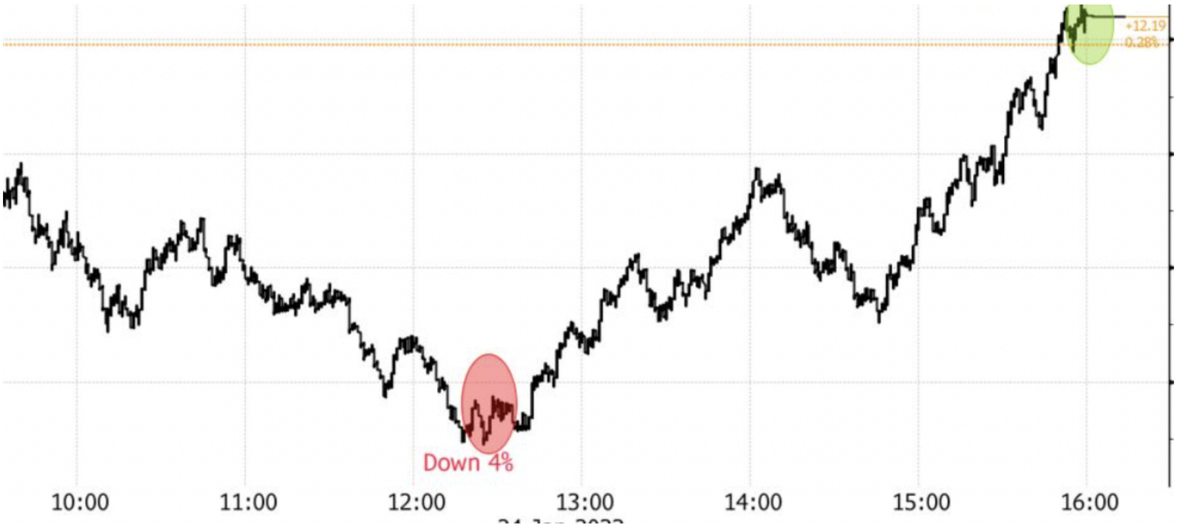
Current Market Price:

- This is just an informational box displaying the current market price.
- Initially when no order is executed, it should display NA.
- The price should change dynamically each time a trade happens and price changes. The same should also be reflected in the graph.

Trade Transaction History:

- This information box should display trade transactions sequentially in the order they were executed. (eg. 3 stocks from A transferred to B at price 500$).

Graph:

- Each time a trade is executed, this should be reflected here.
- You are free to choose the graph style, type, hover information, time interval to make it more intuitive.
- 

Notifications:

- There should be a notification modal/area where each event triggered is visible.

Initial Conditions:

- Initially users' asset portfolio should default to zero in fiat and stocks. We will edit their asset value before starting to trade (hence should be easily editable, which is equivalent to transferring money in users trading account).
- Initially the market price is NA.
- Initial order book and graph will be empty.

Scope of innovation and judging criteria:

- You are allowed to innovate in each component. Show your creativity in additional functionalities to enhance user experience.
- You are allowed to make multiple screens and design your website in a better way.
- The above listed components are basic which is required at the end of the day.
- Each component and it's working will fetch you points.
- Major 4/6 working components will allow you to qualify for the prizes.
- We will be evaluating continuously based on your strategy, execution, timing, collaboration etc.
- Make sure two of your team members are always present in the venue while the hackathon is ongoing, this will facilitate us to take reports on timely progress.
- **You can use the database of your choice even sqlite will do. We will be focusing on correctness and user experience.**
- **In Backend Golang is preferred but not necessary, while in frontend we require you to use React/redux.**
- All the best!

Bonus Points for the following implementation:

- Stream the real time cryptocurrency market data from the API using websockets and create a graph for the same.
- You can use the API of any vendor of your choice.

- Reference: https://nordicapis.com/9-apis-for-real-time-cryptocurrency-data/