Name: Kafil Abbas Momin

Project Name: Yam Option

Project Goals = The main goal of the project is to provide live data of futures and options of NSE and MCX exchanges with analysis.

**Problems/Challenges: =>**

1)Find a good and realiable source of data

2)To process the data in a perticular format and store it

3)Setting up a backend server which will provide this data though websockets connection to the clients

4)make the data visually apealing

**Methodology: =>**

1)To find the perfect source i have to go through a bunch of the sources some of them were. grabing the official site of nse using python, fetching the data using yahoo finance, using websocket of fyres api, using websocket of kite api, using websocket of shoonya api, using the upstocks api. i tried all option but every option either provide data with high limitation like only 2 min delay data is served on nse official website and on fyres api you can only subscribe 50 symbols with websocket and kite websocket charges 2000rs per month for tick-by-tick data.

So the best i found was shoonya api it was free and have a limatation, i can make only one websocket atmost. But the best part is you can subscribe as much symbols which is provided by nse and mcx so it means whole market only with one websocket so it was the best option for fetching data with no cost and also the data is tick by tick it means it is updating at a rate of 0.05 sec

**2)Connect Shoonya Api: =>**

.1) To make symbol name list:

The basic structure of symbol is 'option'+'expiry'+'P'(for puts) or +'C'(for call)+strike price

Example: BANKNIFTY28SEP23P52500 this is a symbol for Bank Nifty with expiry 28SEP23 and P for puts options and having strike price 52500

So, we have to make a list of all the symbols name we need to subscribe to WebSocket, So in my program I need to connect to a total of 6 Option each with 2 expiries and each having 42 symbol so a total of 504 symbol name will be there

.2) Generate token list:

As we get symbol name list, we will need to search for their token now there are 2 methods to find the token first is to find it using Api itself and second it to download token file from shoonya website and find the symbol name there and extract the token there.

As finding the token from the Api takes too much time, I choose second option to find the token from the downloaded file

.3) Making instrument list:

As we get token list, we will need to make an instrument list, A basic instrument have a structure like

‘exchange’+’|’ + ‘token’ so to make a instrument we need to have information about the exchange of the token So after this we can get a instrument list

.4) Connection to WebSocket:

Now the final step is to connect to WebSocket so after making the connection with Api and after login we need to subscribe to the instrument list which we can do by passing the instrument list as a argument in api.subscribe()

**3)Processing the data: =>**

.1) Distribute the received data:

The received data is in Json format and it include the token number, So we can distribute the data as per the token number

.2) To find the index:

As we have token number, next find the index. To do this we have created a dictionary earlier while making token list, which have tokens as keys and it gives us index so after finding the index, we can get the particular numpy array where we have to update the values

.3) Convert the format of data:

The data stored earlier is in numpy array and has a lot of fields so to simpliy it and store. The data is converted from numpy array to dataframe the fields are reduced from 17 to 6 as those are most important.

.4) To store the data:

The data is stored as csv and json files inside local folder and it is updated every 0.25 seconds

**4)Send the data to frontend:**

.1) The backend server is made with Django and the data is streamed between frontend and backend by with Django channels making a websocket connection between them

.2) The data from the backend is converted to json fromat with few keys and sent it to the frontend and at front end and the data is divided according to keys.

.3) for plotting the chart I have used chart js the chart is updating in realtime which is a must feature for a Realtime stock market website