Documentation for BreakoutAI Assessment

API: Upstox

- Authorize (https://api.upstox.com/v2/login/authorization/dialog)
- Get Token (https://api.upstox.com/v2/login/authorization/token)
- Option Contracts (https://api.upstox.com/v2/option/contract)
- Put/Call Option Chain (https://api.upstox.com/v2/option/chain)

Approach

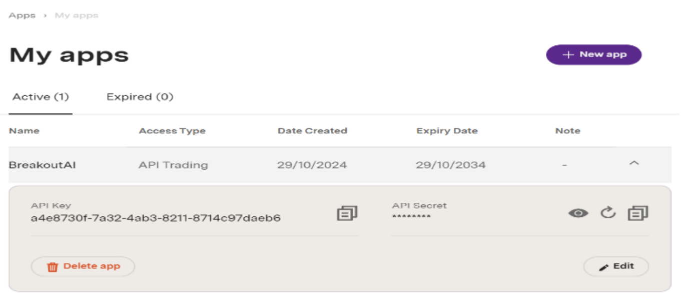
1. Authorization:

1.1 Create an Upstox Developer Account

- Register on the Upstox Developer Portal and create your developer account

1.2 Create an Application

- Log into your developer account and create a new app.
- Provide a name for the app and set a redirect URL (this can be any URL; it's needed for authentication flow).
- Once the app is created, you'll receive a **Client ID** and **Secret Key**.



Name	Access Type	Date Created	Expiry Date	Note
BreakoutAI	API Trading	29/10/2024	29/10/2034	-

API Key
a4e8730f-7a32-4ab3-8211-8714c97daeb6

API Secret
****

1.3 Obtain the Authorization Code

- Using your **Client ID** and **Redirect URL**, initiate the authentication process.
- Use the auth.py script to start the authorization flow, which will redirect you to the Upstox login page.
- After successful login, you will receive an **Authorization Code**.

  Example:

Redirected Login Url after authentication:
https://breakoutai.tech/?code=rLY_6E
Authorization Code: rLY_6E

1.4 Generate the Access Token

- Use the accesstoken.py script with the **Authorization Code** to request an **Access Token** from Upstox.
- The Access Token will be used for all further API calls.

2. Retrieve Option Chain Data

2.1 Set Up API Parameters

- Construct the “instrument_key” based on the instrument name, such as "NSE_INDEX|Nifty 50".
- Prepare the request headers with the **access token** for authorization.
- Set up parameters for the two endpoints:
  endpoint1: Fetches the general option contract details.
  endpoint2: Fetches details for each call or put option.

2.2 Make API Calls

- Send a GET request to endpoint1 to retrieve general data on the option contracts, including expiry date, strike price, and lot size.
- Send a GET request to endpoint2 to get specific market data for each option contract, like bid and ask prices.

2.3 Extract Data from Responses

- If both API responses are successful, parse the returned JSON data.
- Convert the data from each API call into separate DataFrames, df1 (contract details) and df2 (option prices).

2.4 Match Contracts with Prices

- For each row in df1 (representing each strike price), find the matching entry in df2 based on expiry date and strike price.
- Depending on the side parameter:
  a) For "PE" (Put Options), extract the **highest bid price** from the put option data.
  b) For "CE" (Call Options), extract the **highest ask price** from the call option data.
- Append the extracted data along with the relevant information (instrument key, lot size, expiry, strike price) to a list of results.

2.5 Organize Data into DataFrame:

- Convert the list of results into a DataFrame with columns: instrument_name, strike_price, side, bid/ask.
- Output this DataFrame to display the highest bid/ask prices per strike.

2.6 Invoke Margin Calculation:

- Pass the resulting DataFrame to the calculate_margin_for_contracts function for further processing (margin and premium calculation).

3. Calculate Margin and Premium Earned

3.1 Set Up API Endpoint

- Define the endpoint for margin calculation: https://api.upstox.com/v2/charges/margin.
- Include the required headers for authorization and content type.

3.2 Loop Through Each Option Contract

- For each row in the input DataFrame, extract instrument_key, lot_size, and option side.
- Set transaction_type to "SELL" and product to "D" to specify the desired parameters for margin calculation.

3.3 Prepare Payload for Margin Request

- Construct the payload for the margin API with the following fields
  a) instrument_key
  b) quantity (lot size)
  c) transaction_type as "SELL"
  d) product as "D"

3.4 Make Margin API Request

- Send a POST request to the margin API with the payload.
- If successful, extract the total_margin or the relevant margin field from the response.
- Update the margin_required column for the respective row with this value.
- If the request fails, log the error and set margin_required to None.

3.5 Calculate Premium Earned:

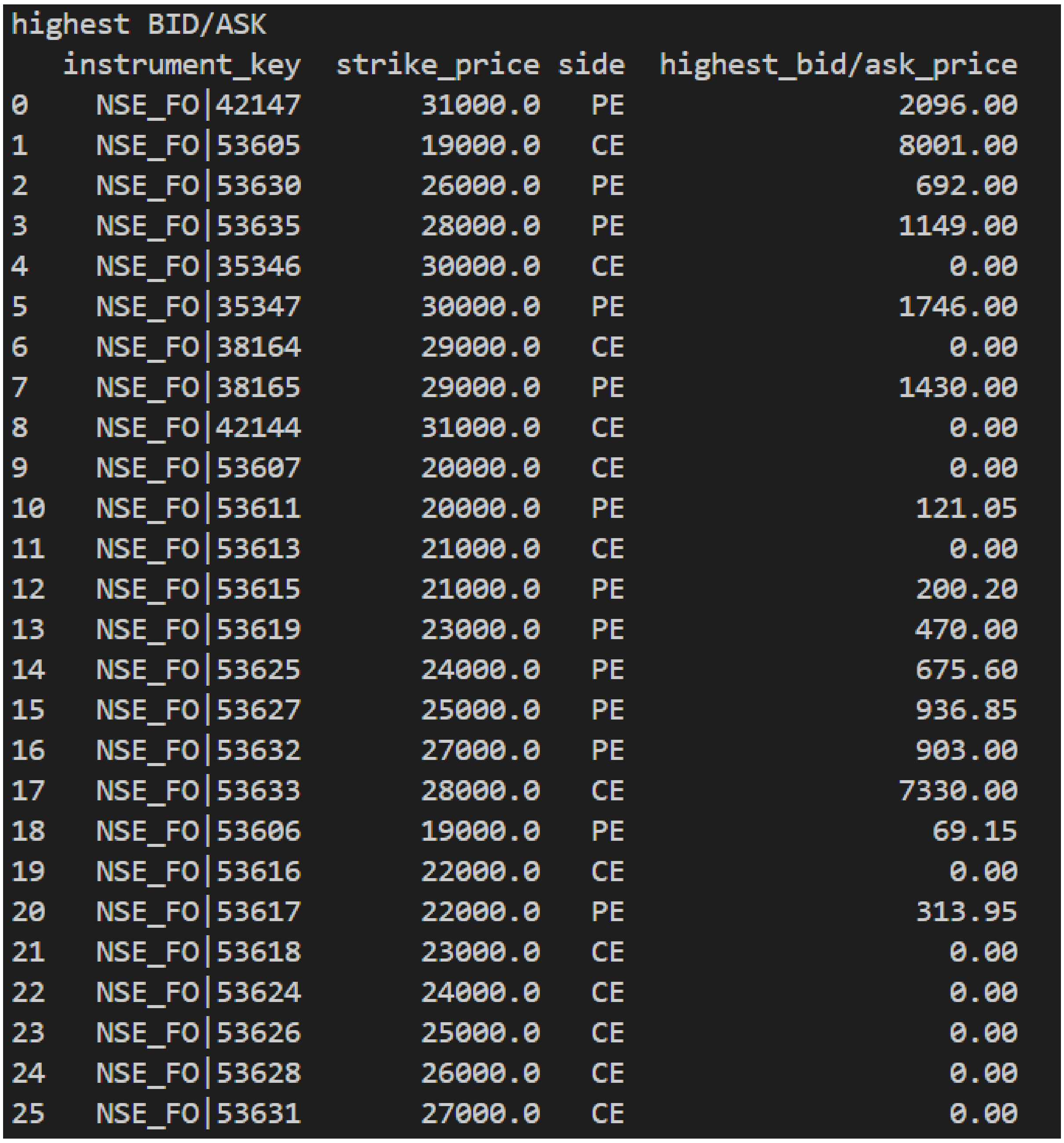
- Calculate premium_earned as highest_bid/ask_price * quantity.
- Store this calculated premium in the premium_earned column for the respective row.

3.6 Organize and Display Results

- Append the updated rows to a new DataFrame and include the calculated margin_required and premium_earned.
- Output this final DataFrame with columns: instrument_name, strike_price, side, bid/ask, margin_required, premium_earned.

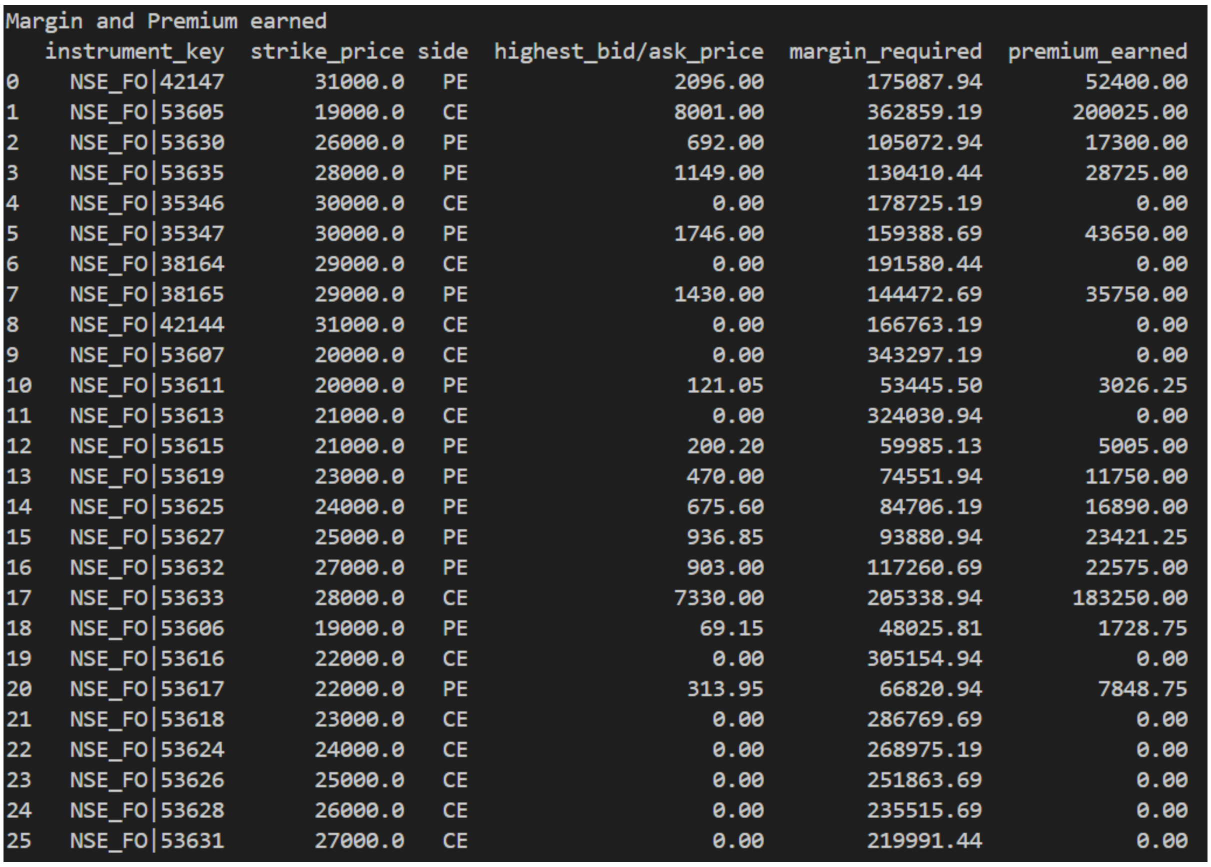
4. Output

Task -1 :



```
highest BID/ASK
   instrument_key  strike_price side  highest_bid/ask_price
0    NSE_FO|42147       31000.0   PE                2096.00
1    NSE_FO|53605       19000.0   CE                8001.00
2    NSE_FO|53630       26000.0   PE                 692.00
3    NSE_FO|53635       28000.0   PE                1149.00
4    NSE_FO|35346       30000.0   CE                   0.00
5    NSE_FO|35347       30000.0   PE                1746.00
6    NSE_FO|38164       29000.0   CE                   0.00
7    NSE_FO|38165       29000.0   PE                1430.00
8    NSE_FO|42144       31000.0   CE                   0.00
9    NSE_FO|53607       20000.0   CE                   0.00
10   NSE_FO|53611       20000.0   PE                 121.05
11   NSE_FO|53613       21000.0   CE                   0.00
12   NSE_FO|53615       21000.0   PE                 200.20
13   NSE_FO|53619       23000.0   PE                 470.00
14   NSE_FO|53625       24000.0   PE                 675.60
15   NSE_FO|53627       25000.0   PE                 936.85
16   NSE_FO|53632       27000.0   PE                 903.00
17   NSE_FO|53633       28000.0   CE                7330.00
18   NSE_FO|53606       19000.0   PE                  69.15
19   NSE_FO|53616       22000.0   CE                   0.00
20   NSE_FO|53617       22000.0   PE                 313.95
21   NSE_FO|53618       23000.0   CE                   0.00
22   NSE_FO|53624       24000.0   CE                   0.00
23   NSE_FO|53626       25000.0   CE                   0.00
24   NSE_FO|53628       26000.0   CE                   0.00
25   NSE_FO|53631       27000.0   CE                   0.00
```

Task – 2:



```
Margin and Premium earned
   instrument_key  strike_price side  highest_bid/ask_price  margin_required  premium_earned
0     NSE_FO|42147       31000.0   PE                2096.00        175087.94        52400.00
1     NSE_FO|53605       19000.0   CE                8001.00        362859.19       200025.00
2     NSE_FO|53630       26000.0   PE                 692.00        105072.94        17300.00
3     NSE_FO|53635       28000.0   PE                1149.00        130410.44        28725.00
4     NSE_FO|35346       30000.0   CE                   0.00        178725.19            0.00
5     NSE_FO|35347       30000.0   PE                1746.00        159388.69        43650.00
6     NSE_FO|38164       29000.0   CE                   0.00        191580.44            0.00
7     NSE_FO|38165       29000.0   PE                1430.00        144472.69        35750.00
8     NSE_FO|42144       31000.0   CE                   0.00        166763.19            0.00
9     NSE_FO|53607       20000.0   CE                   0.00        343297.19            0.00
10    NSE_FO|53611       20000.0   PE                 121.05         53445.50         3026.25
11    NSE_FO|53613       21000.0   CE                   0.00        324030.94            0.00
12    NSE_FO|53615       21000.0   PE                 200.20         59985.13         5005.00
13    NSE_FO|53619       23000.0   PE                 470.00         74551.94        11750.00
14    NSE_FO|53625       24000.0   PE                 675.60         84706.19        16890.00
15    NSE_FO|53627       25000.0   PE                 936.85         93880.94        23421.25
16    NSE_FO|53632       27000.0   PE                 903.00        117260.69        22575.00
17    NSE_FO|53633       28000.0   CE                7330.00        205338.94       183250.00
18    NSE_FO|53606       19000.0   PE                  69.15         48025.81         1728.75
19    NSE_FO|53616       22000.0   CE                   0.00        305154.94            0.00
20    NSE_FO|53617       22000.0   PE                 313.95         66820.94         7848.75
21    NSE_FO|53618       23000.0   CE                   0.00        286769.69            0.00
22    NSE_FO|53624       24000.0   CE                   0.00        268975.19            0.00
23    NSE_FO|53626       25000.0   CE                   0.00        251863.69            0.00
24    NSE_FO|53628       26000.0   CE                   0.00        235515.69            0.00
25    NSE_FO|53631       27000.0   CE                   0.00        219991.44            0.00
```