

Market and Stock Analyzer

COP290-Assignment1-Subtask2

Group Members:
Priyanshu Agrawal (2022CS11641)
Vedant Sameer Talegaonkar (2022CS11603)

Introduction:

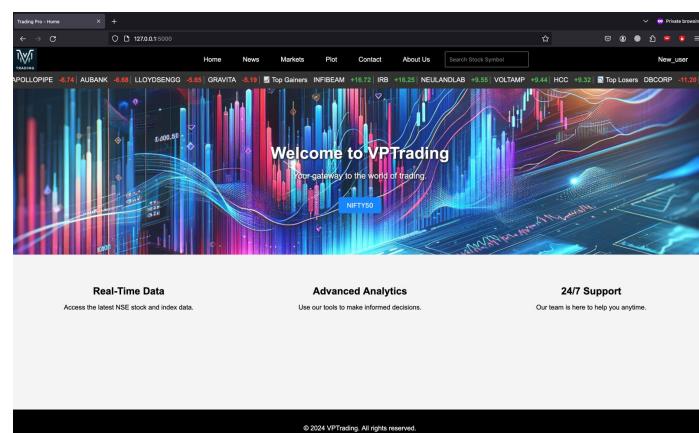
In this assignment, our aim was to create a user-friendly and reliable analyzer and information tool for NSE market stocks. We used Python's Flask library for the back-end, while HTML/CSS (along with a little JavaScript) for the front-end of our website.

Functionalities:

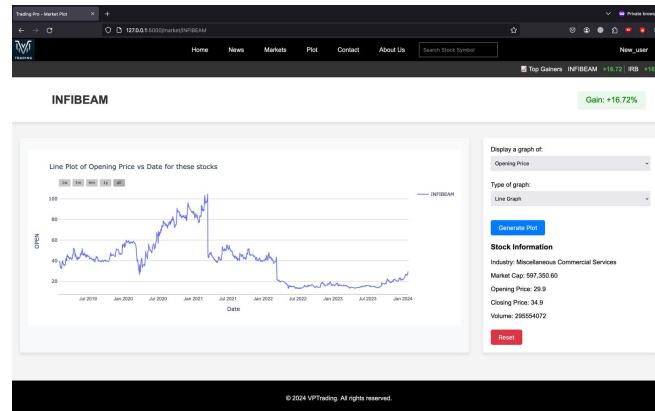
1. A login/register method is added (based on the startercode) to accept users into the database, only after which the user can access all the features. Without login, users only have limited features like stock news.
2. A watchlist feature on the user dashboard, which allows the user to selectively track and analyse their stocks.
3. A webpage dedicated to displaying market data of all the stocks, and provisions to add filters based on relevant parameters, along with an option to search for the stock based on its symbol. Moreover, every stock has been hyperlinked to its own page with its data along with plot.
4. A graphing functionality for displaying various parameters of a stock, in multiple timeframes as well as in multiple formats (line graph, candle stick).
5. A news webpage, for displaying the latest updates of the stock market in a user-friendly format.
6. Displaying the top 5 gainers/losers for the day on the home page, and a webpage dedicated to more information about these stocks.

Snippets of the website:

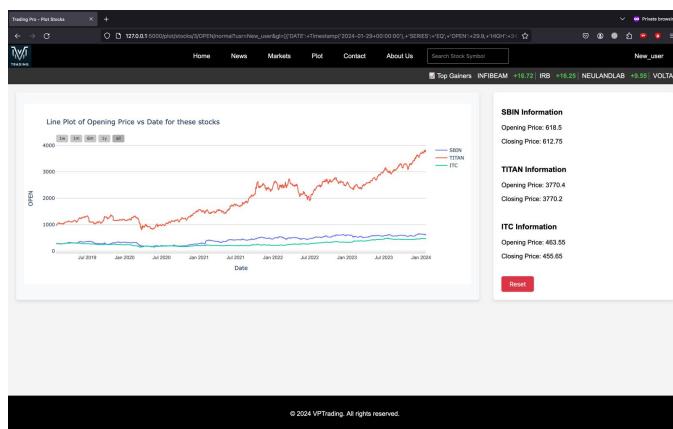
1. Home page after login:



2. Stock-info page:



3. Multiple stocks comparison plot:



4. Market Page

Market Data

View the latest market trends and data.

Name-Symbol	Industry	Opening Price	Closing Price	Last Traded Price	52 Wk. High	52 Wk. Low	Volume	Market Capitalization	%Change
MRF	Automotive Aftermarket	136857.15	143873.6	143450.0	150254.15	81380.05	26375	5,495,641.29	+4.82%
ULTRACEMCO	Construction Materials	10000.1	10274.05	10237.6	10326.0	6642.55	334463	30,200,005.71	+2.37%
PAGEND	Apparel/Footwear	36711.1	37635.2	37540.0	43570.0	34952.65	41350	4,295,663.61	+2.26%
ABBOTINDIA	Pharmaceuticals Major	25060.75	25568.05	25560.0	26381.65	19900.0	32932	nan	+2.07%
BOSCHLTD	Auto Parts: OEM	22860.95	23248.8	23150.0	23688.95	16545.0	64222	6,546,708.84	+1.53%
SHREECEM	Construction Materials	27998.75	28411.55	28316.0	29250.0	21860.0	35863	10,338,324.97	+1.13%
PGHH	Household/Personal Care	17169.0	17266.95	17279.0	19250.0	13140.1	2747	5,637,147.64	+0.64%
MARUTI	Motor Vehicles	9931.15	9991.4	9990.0	10532.85	8130.0	446516	32,390,853.58	+0.59%
ZFCVINDIA	Auto Parts: OEM	16892.0	16790.6	16800.0	17700.0	8970.85	13874	3,036,122.20	-0.54%
NORNAT	Electronics	38900.0	38873.26	38850.6	41160.0	34345.4	3706	9,287,636.09	-0.20%

Filters

Min Opening Price: 0.00

Max Opening Price: 0

Min Last Price: 0

Max Last Price: 0

Min Volume: 0

Max Volume: 0

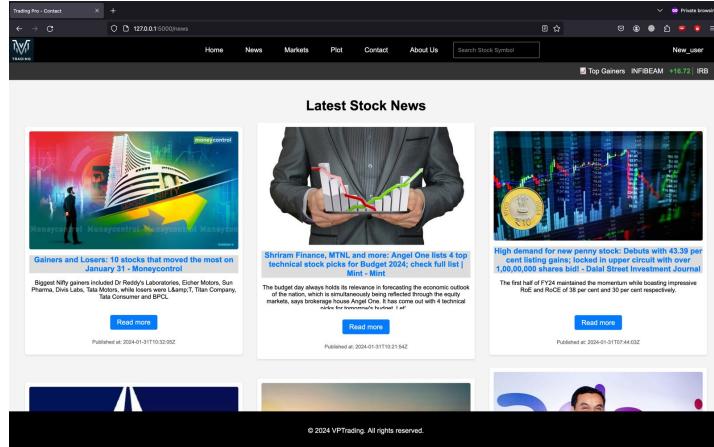
Min %Change: 0

Max %Change: 0

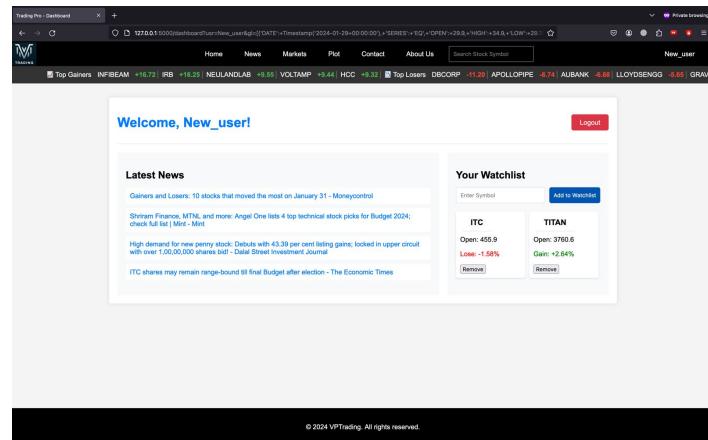
Stock Name: Select a Stock

Apply Filters | Reset

5. News Page



7. User dashboard



Design Decisions:

1. We created a navbar for navigating across the different webpages, which leads to access throughout the website.
2. To encourage the users to create an account on our website, we have hidden most of our functionalities behind a wall of login. Features like checking market, plotting graphs, and creating watchlist are available only after the user logs into the website.
3. We decided to use all of the stocks present in the NIFTY Total Market (~750 stocks), not just the stocks in the NIFTY50 list.
4. In the plot section, we decided to make two pages – one which takes in the number of stocks the user wanted to compare, while the other takes that many stock symbols and plots them on a graph. This was done to maintain clarity in the website.
5. Industry data for the stocks was not available in the jugaad-data library, so we scraped this data from a few reliable websites and used it statically for the market section.
6. For encapsulation in code, we created separate .py files for the data and news module. Therefore, gathering and processing of the data is not done in the main file, which leads to less cluttered code.

7. For clean html code, we kept a base file (base.html) which contained the features which do not change from page to page, while every other html file simply adds to the base file, which leads to tremendous reduction in number of lines of code.

8. All of the data that we need for the following features – market list, watchlist and top gainers/losers – is fetched in the website start-up and is then cached using Flask's Cache module. This leads to almost immediate loading times in most cases and hence, improves user experience, with just a one-time delay during server start-up.

Tradeoffs/Notable points:

1. We initially wanted to display the live data for all of the stocks in the market section. However, due to the slow nature of fetching data from jugaad-data library, we shifted to using 1 day old data for the stocks. This decision significantly sped up the loading process.
2. We have added the functionality of plotting Candlestick model of the stock, which is a comprehensive way to judge the performance of the stock.

--- END OF REPORT ---