Assignment-1: Trading simulator and analyzer

Name: Sarthak Panda(2022CS51217) Name: P V Revanth (2022CS51650)

Subtask 2

Introduction:

For this subtask, we have developed a web-based tool using Plotly, a data visualization library. This tool allows users to effortlessly explore and analyze stock market data, offering dynamic features to enhance the user experience.

Buttons and Features:

1. Timeline Buttons:

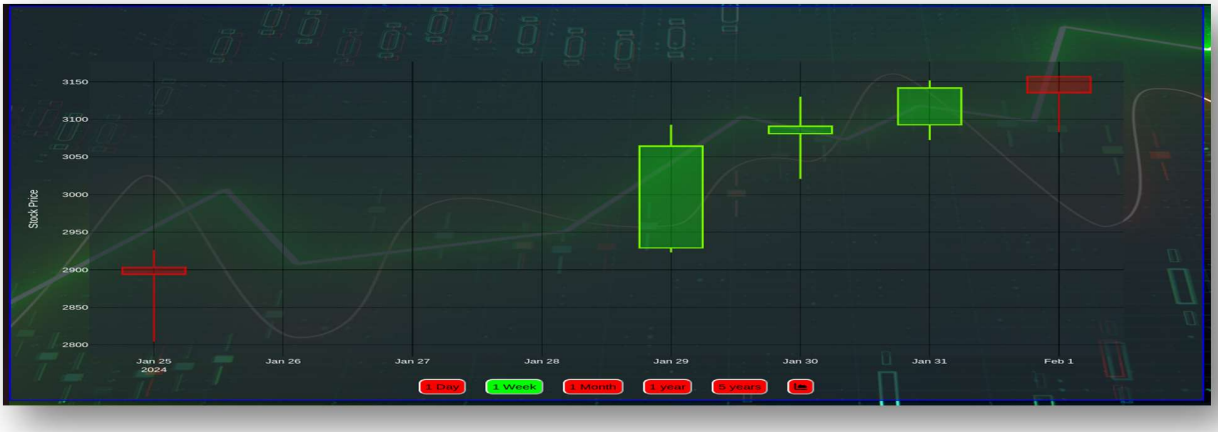
These buttons serve to adjust the time frame for stock data, providing options for daily, monthly, yearly, or overall views. Users can observe how a stock has performed over various periods, aiding in comprehensive analysis.

- **5 years**

- ***1 year***

- ***1 Month***

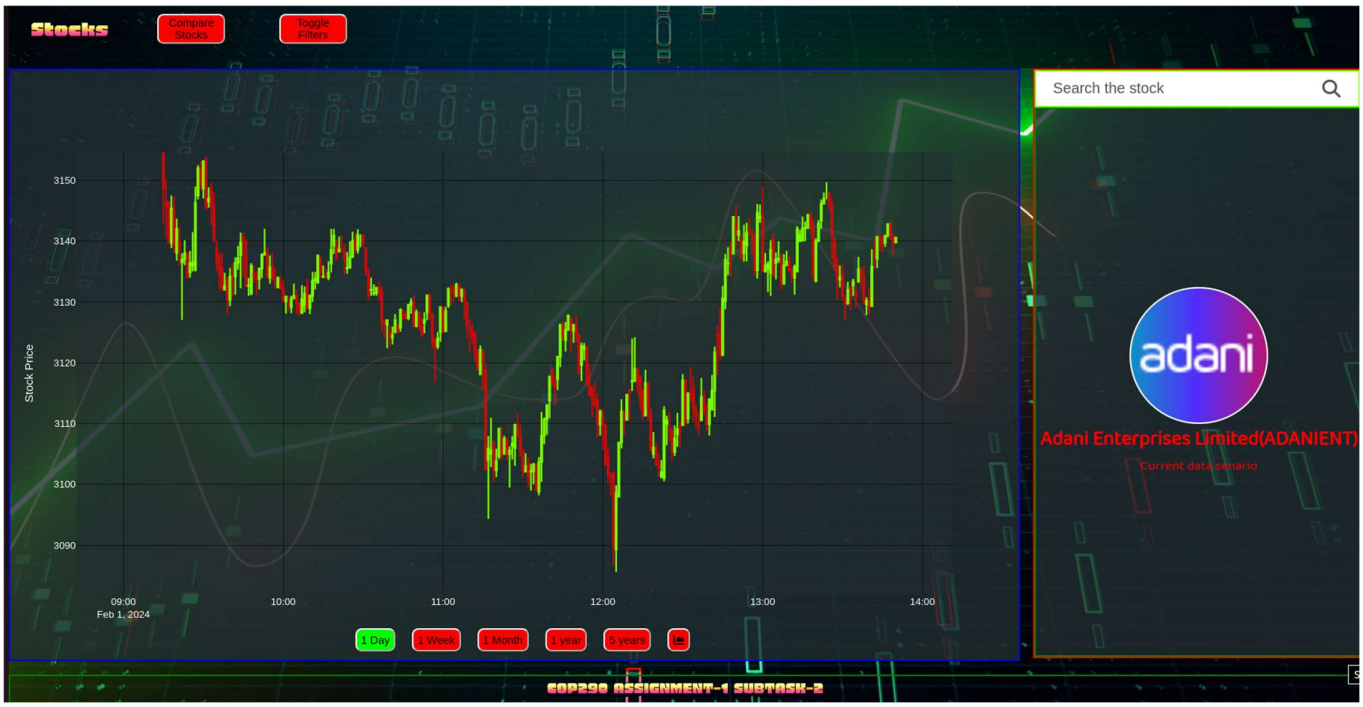
- ***1 Week***



- ***1 Day(Live data 9AM to 3:30PM)***

2. <u>Navigation Buttons:</u>

Single Stock View: *Users can focus on a specific stock by default, allowing them to search and analyze individual stock data.(Here we display logo of few companies on right side)*



Multiple Stock View: *For comparative analysis, users can switch to this view, accommodating up to four stocks. Side-by-side or in a single graph, this feature facilitates easy comparison.(Here right side is a watchlist)*

Filters: *This functionality allows users to apply filters based on sectors, industries, market caps, and more, refining the displayed stock data. These additional indicators, combined with SMA, EMA, and ADX, offer a comprehensive toolkit for users to delve deeper into stock market trends and make more informed investment decisions.on this is a scroll page appears. You may select or deselect filter as needed.*

- ***Simple Moving Average (SMA):*** *This indicator reveals the average stock price over a specified period, helping smooth out price fluctuations and identify the trend's direction.*
- ***Exponential Moving Average (EMA):*** *Similar to SMA, EMA assigns more weight to recent prices. This allows it to be more responsive to price changes and closely track the prevailing trend.*
- ***Average Directional Index (ADX):*** *This indicator gauges the strength of a trend, providing a numerical value ranging from 0 to 100. A higher value indicates a more robust trend. It's important to note that ADX doesn't specify the trend's direction but rather its magnitude.*
- ***Relative Strength Index (RSI):*** *RSI measures the magnitude of recent price changes and helps identify overbought or oversold conditions in the market. RSI values range from 0 to 100, with readings above 70 indicating overbought conditions and readings below 30 indicating oversold conditions. Traders often use RSI to identify potential reversal points or confirm the strength of a trend.*

- ***Volumes:*** *The volume indicator provides insights into the trading activity of a stock. It represents the total number of shares traded during a specific period. Analyzing volume can help users assess the strength and sustainability of a price movement. For example, a significant price increase accompanied by high trading volume may suggest a strong uptrend, while low volume during a price movement could indicate a lack of strong market conviction.*

  3. <u>Plot Type Button:</u>

  Users have the flexibility to toggle between candle-stick graphs and area graphs. The line graph visually represents stock data as a line, while the area graph highlights the closing price as an area, catering to diverse user preferences.

Libraires USED :

Plotly ,Pandas,Cufflinks

Fonts:

We have used the following fonts and icons in our web-based tool:

- *We imported four Google fonts: Baloo Bhai 2, Baloo Bhaina 2, Bree Serif, and Ephesis. These fonts are used to style the text in our web page. For example, we applied the 'Baloo Bhai 2' font to certain elements using the CSS property font-family: 'Baloo Bhai 2', cursive;.*
- *We also used Material Icons in our tool. For instance, we displayed a chart area icon using the tag, which is often used for data visualization.*

Data Fetching:

We fetched the data from Jugaad and Yfinance. For historical data. While we fetched live daily data from Money control website.

- *Jugaad is used for 5years,1year,1Month old data*
- *Yfinance is used for 1week olld data*

Reason for Using Plotly Instead of Matplotlib:

We chose Plotly over Matplotlib because Plotly creates more interactive and web-friendly plots with less code and more customization options. Plotly also supports 3D and geographic plots, which are not available in Matplotlib.

Search Box Implementation:

We implemented a search box in our tool. It includes an auto-complete list and allows users to add elements to their watch list for multiple stock displays.

Overview of the code:

AJAX Queries:

Though Ajax queries were used as means to communicate between python and javascript. Here we specifically explained for 1 Day (due to its complicacy due to live data we have explained it in summarized way below).

1. ***Toggle Filter Function:*** *Triggered by the "Toggle Filters" button. When clicked, it changes the appearance of the button by toggling the "active" class, which highlights the button in green,and sends an AJAX POST request to the /processfilter endpoint. If the current timeline range is not "1 Day," it fetches new data and updates the Plotly chart.*

   /processfilter Route: *Retrieves the current date and determines the start date based on the current range. Calls the appropriate plot function with the current stock symbol, series, start date, end date, filter status, and range. Converts the resulting chart data to JSON format and sends the JSON data as a response.*
2. ***Toggle Button Function:*** *Triggered by the "Toggle Plot Type" button. It toggles the "active" class on the plot type button and sends an AJAX POST request to the /processplottype endpoint. If the current timeline range is not "1 Day," it fetches new data and updates the Plotly chart.*

   /processplottype Route: *Updates the current plot type based on the received data. Retrieves the current date and determines the start date based on the current range. Calls the appropriate plot function with the current stock symbol, series, start date, end date, filter status, and range. Converts the resulting chart data to JSON format and sends the JSON data as a response.*
3. ***Live Data Plot Function:*** *Initiates a live data plot by sending an AJAX POST request to the /processrange endpoint with the current range set to "1 Day." It renders the initial Plotly chart and sets up an interval to periodically update the chart.*
4. ***Update Plot Function:*** *Sends an AJAX POST request to the /processrange endpoint with the current range set to "1 Day" to fetch updated data. It uses Plotly.react to update the existing chart with the new data.*
5. ***Stop Updates Function:*** *Clears the interval set by livedataplot, effectively stopping the periodic updates.*

   /processrange Route: *Updates the current time range based on the received data. Retrieves the current date and determines the start date based on the updated time range. Calls the appropriate plot function with the current stock symbol, series, start date, end date, filter status, and range. Converts the resulting chart data to JSON format and sends the JSON data as a response.*

How does the website interact when displaying a graph on selecting a particular stock?

The main function, sendData(), is used to send an AJAX request to the server with the selected stock symbol and series. This function is triggered when the user clicks the search button in the search box. Here's a simplified breakdown of its operation:

1. ***Check Input:*** *It first checks if the input element is present. If not, it alerts that the stock is not found.*
2. ***If Stock Found:*** *If the stock is found, it performs the following actions:*
    - ***Plot Loading Chart:*** *Plots a loading chart in the chart container using Plotly.*

- ***Disable Inputs:*** *Disables all the inputs and buttons except the input box and the search button.*
- ***Reset Active Class:*** *Removes the active class from the timeline and resets the plot type.*
- ***Display Stock Details:*** *Hides the previous stock details and displays them for the selected stock.*
- ***Send AJAX Request:*** *Sends an AJAX request to the server with the stock symbol and series as parameters.*
- ***Receive Response:*** *Receives the response from the server, which contains the chart data for the selected stock.*
- ***Plot Chart Data:*** *Plots the chart data in the chart container using Plotly.*

Enable Inputs: *Enables the input box and buttons and adds the active class to the timeline and the plot type.*

This process ensures a smooth and interactive experience for the user when selecting a particular stock to display on the graph.

Conclusion:

Flask helped to make web development faster and allowed to use some good python libraries to display in web, over basic HTML/CSS/JS.