TradeSim Documentation

Karissa (2020CS50428), Sonam (2021CS10123)

February 1, 2024

1 Insights and Functionalities of the website

1.1 Login

a. Access the website through the provided URL.
b. On the login page, enter your username and password .
c. Click the "Login" button to log in.
d. If you have an account else go to the register url.

1.2 Register

a. Access the registration page through the provided URL.
b. Fill in the required information, including your desired username, a secure password.
c. Click the "Register" button to create an account.
d. Once registered, you can proceed to the login page and use your credentials to log in.

1.3 Home Page

The home page of TradeSim provides easy access to essential features for stock analysis. Familiarize yourself with the following options:

1.4 Navigation Bar

The navigation bar at the top of the page allows seamless navigation to different sections of the website. Made with HTML and CSS, it has hovering effects and functional drop down menu. Here's a breakdown of the available options:

- **TradeSim:** Return to the main dashboard.
- **Home:** Navigate back to the home page.
- **About Us:** Learn more about TradeSim.
- **Options:** Access a drop down menu with the following options:
  - **Plot Graph:** Navigate to a page for plotting a single stock graph on a daily, weekly, monthly or yearly scale.
  - **Plot Multiple Graphs:** Directs to a page for generating multiple stock analysis graphs for some x number of years.
  - **List of Stocks:** Opens a page displaying a list of Nifty 50 companies with their stock symbols after applying 3 different filters : **average price, top 10 companies stock price, range of closing price**.
- **Documentation:** Open the documentation in a new tab for reference.
- **Logout:** Log out of the system.

1.5 Alerts

Upon successful login, a welcome alert made with HTML, CSS only is displayed briefly. You can close the alert using the "X" button.

1.6 Card Section

The card section made with HTML and CSS is added with shadow and hovering effects and transitions on the home page provides quick access to key functionalities:

1.6.1 Plot Single Graph

- **Description:** Plot the graph of a given stock on a defined time scale (daily, weekly, etc.).
- **Learn More:** Click on the "Learn More" button to navigate to the relevant page.

1.6.2 Plot Multiple Graphs

- **Description:** Plot graphs for multiple stocks on the same graph to visually compare their prices.
- **Learn More:** Click on the "Learn More" button to navigate to the relevant page.

1.6.3 Apply Filters

- **Description:** Apply technical filters to get a list of stocks passing the filters (average price, etc.).
- **Learn More:** Click on the "Learn More" button to navigate to the relevant page.

The home page serves as the central hub for various functionalities related to stock analysis. Here is an overview of its components:

1.7 Single Graph Page

The Single Graph page has a card div central allows users to plot the stock graph of a specified stock symbol over a defined time interval. It has 4 forms. The From and To fields have a calendar too to select the date. A drop down menu is also added. Follow the guidelines below to effectively use this feature.

1.7.1 Navigation Bar

1.7.2 Plotting a Single Graph

1.7.3 Stock Symbol:

Enter the stock symbol of the company for which you want to plot the graph.

- **Example:** `SBIN, CIPLA, WIPRO etc.` Note the symbols of the Nifty 50 companies list that we use for this assignment is given in the Apply Filters page. In the backend, a csv file has been added that reads the symbols from there.

1.7.4 Date Range:

Select the starting and ending dates for the time interval of the graph.

- The `"From"` and `"To"` fields allow you to specify the range for the stock data.

1.7.5 Time Interval:

Choose the desired time interval for the data points on the graph.The date entered by user in the "FROM" and "TO" field will serve as the time frame on which the graph will be displayed:

- **Daily:** Plot the graph with daily data points.
- **Weekly:** Aggregate the data on a weekly basis.
- **Monthly:** Aggregate the data on a monthly basis.
- **Yearly:** Aggregate the data on a yearly basis.

1.7.6 Plot Button:

Click the `"Plot Graph"` button to initiate the graph plotting process.

1.8 Multiple Graph Page

The Multiple Graph page on TradeSim enables users to generate plots for multiple stock symbols over a specified time interval. Follow the guidelines below for effective use:

1.8.1 Navigation Bar

1.8.2 b. Generating Multiple Stock Plots

Follow the steps below to generate plots for multiple stock symbols:

1. **Stock Symbol 1:** Enter the stock symbol of the first company for analysis (e.g., SBIN).
2. **Stock Symbol 2:** Enter the stock symbol of the second company for analysis (e.g., WIPRO).
3. **Stock Symbol 3:** Enter the stock symbol of the third company for analysis (e.g., CIPLA).
4. **Number of Years:** Specify the number of years for the analysis (e.g., 5).
5. Click the **"Generate Plots"** button to initiate the graph generation process.

Note: The generated plots will provide a visual comparison of the stock performance of the specified companies over the chosen time interval.

1.9 List of Stocks Page

The List of Stocks page on TradeSim provides users with a comprehensive list of Nifty 50 companies, along with options to apply various filters for focused analysis. Follow the guidelines below for effective use:

1.9.1 Navigation Bar

1.9.2 Applying Filters

Filter stocks above average price:

- Check the "Filter stocks above average price" checkbox.
- Click the **"Apply"** button to filter the list based on stocks above the average price. The data is based on that from 1st February 2024.

Filter Top 10 stocks:

- Check the "Filter Top 10 stocks" checkbox.
- Click the **"Apply"** button to filter the list based on the top 10 stocks ordered by closing price.

Filter company by closing price range:

- Enter the minimum and maximum closing price range in the "From" and "To" input boxes, respectively.
- Click the **"Apply"** button to filter the list based on the specified closing price range.

Note: The filtered list will be displayed below the filter options, providing a refined view based on the chosen criteria.

1.9.3 List of Companies

The page displays the Nifty 50 companies along with relevant information. The list will be updated based on the applied filters.

Important Information:

- The stock data is as of 01-02-2024.
- Use the filters to customize the list and focus on specific stock criteria.

1.10 About Us

The About Us page on Trading Simulator and Analyzer provides information about the platform's mission, the team behind it, and contact details. Below is a guide on understanding and using this page effectively.

1.11 Documentation

The Documentation page is attached to the report.pdf that serves as a comprehensive guide to understanding and using the Trading Simulator and Analyzer platform.

2 Design decisions

2.1 Technologies Used

- **Flask:** Chosen as the web framework for its simplicity and flexibility in building web applications.
- **SQLite:** Utilized as the database management system to store user information and stock data.
- **Jinja2 Templates:** Integrated into Flask for dynamic HTML rendering.
- **Pandas and Matplotlib:** Employed for data manipulation and visualization to plot graph, respectively.
- **HTML, CSS:** Applied for responsive and visually appealing front-end design from scratch.
- **Jugaad Data NSE:** Used for fetching National Stock Exchange (NSE) data and storing it in csv files in backend using dataframes in Python module Pandas.

2.2 Architecture Overview

The website follows the Model-View-Controller (MVC) architecture:

- **Model:** Defined using SQLAlchemy, a Python SQL toolkit, to interact with the SQLite database. The `User` model manages user information, while the main application deals with stock data.

- **View:** Utilizes HTML templates with embedded Jinja2 code to render dynamic content. Bootstrap enhances the user interface for better user experience.
- **Controller:** Handled by Flask, which manages routes and interacts with models to fetch or store data. Python scripts are executed using subprocess on the main.py file that we used in Subtask 1 to generate stock data for a given number of years that user enters in the website.

2.3 User Authentication

- Users can register accounts with a unique username and a securely hashed password.
- Sessions are implemented for user authentication, allowing users to stay logged in during their session.

2.4 Stock Data Generation

- The website fetches historical stock data using the `jugaad_data` library, which retrieves the NSE Bhavcopy (Equity) data.
- The `main.py` script is executed with user-provided stock symbols and years, generating CSV files with historical stock prices.

2.5 Plotting Functionality

- Users can generate plots for daily, weekly, monthly, and yearly stock prices based on their selected date range.
- Plots are created using Matplotlib, and the images are converted to `BytesIO` objects before being displayed on the website. The Python module subprocess on main.py is used to generate the data on which graphs are plotted.

2.6 Filtering and Listing Stocks

- Users can filter stocks based on average price or the top 10 stocks with the highest closing prices.
- The website lists filtered stocks, allowing users to explore individual stock details.
- The csv files, generated from functions in app.py and after calling subprocess on main.py, are then used in functions in app2.py to filter the data based on
  (i) filtering only the symbols that are in the Nifty 50 list given in Subtask 1,
  (ii) top 10 closing price,
  (iii) mean of all listed filtered symbol-companies
  (iv)displaying only those symbols which have closing price within the range that the user enters in the forms.
- In the range filter we have done exception handling by adding some client-side JavaScript validation too in case the user doesn't enter the fields "From" or "To".

2.7 PDF Viewing

Users can view a sample PDF document by clicking the 'Documentation' button in the navigation bar. It opens the pdf on a new tab.

3 Tradeoffs

3.1 User-Friendly Interface:

- **Pros**: The website has a clean and intuitive interface, making it easy for users to navigate and interact with different features.
- **Cons:** Website is built in HTML and CSS only. Images sometimes take longer to load.

3.2 Future Enhancements

- Implementing user-specific data storage for more personalized analysis.
- Enhancing data visualization features.
- Improving filtering options and adding more financial metrics.

3.3 Database Integration:

- **Pros:**
  SQLAlchemy is used for seamless integration with a SQLite database, providing an organized structure for user data.
- **Cons:**
  The database is currently a local SQLite instance, which might limit scalability for large user bases. Consider migrating to a more scalable solution in the future.

3.4 Data Visualization:

- Matplotlib is utilized for generating stock price graphs, providing users with a visual representation of the data. Since it is an image file user cannot interact with any chart and it takes more time to load if the number of years are more.
- The website uses subprocess calls to execute external Python scripts. This takes lot of time and might be enhanced using a more robust method for data visualization within the Flask application.
- If we use a newly used company SYMBOL which on which we have never run a test before for the plot graphs, it may take more time than the ones on which we have run test before because it just takes the data from the previously generated CSV files.

3.5 File Handling:

- **Pros:** The application allows users to download stock graphs directly as image files.
- **Cons:** The handling of file paths and temporary files could be improved for better reliability instead of having to re-generate the csv files again and again. They could have been stored in database.

3.6 Registration and Authentication:

- **Pros:** Users can register, log in, and access personalized dashboards. Passwords are securely hashed.
- **Cons:** Currently, the website lacks features such as password recovery and account verification, which are crucial for user account management.

3.7 External Dependencies:

- **Pros:** The application utilizes external libraries such as Pandas, Matplotlib, and Jugaad Data for efficient data processing and visualization.
- **Cons:** Dependency management is essential, and potential updates or deprecations in these libraries could affect the application's functionality.

3.8 Dynamic Data Updates:

The website fetches the stock data latest of the date 01-01-2024.

There might be a delay or potential issues if the external data sources are unavailable. Implementing error handling for such scenarios could be beneficial.

3.9 Performance Considerations:

- **Pros:** The application calculates date differences and filters data efficiently.
- **Cons:** Depending on the size of the dataset and the number of users, performance could be impacted. Implementing caching mechanisms or optimizing queries might be considered for scalability.

In conclusion, while the website exhibits several positive aspects, addressing the outlined cons could lead to an improved user experience, security, and maintainability. Regular code reviews, security audits, and feature updates will contribute to the ongoing success of the application.

4 Screenshots

Figure 1: Login.html

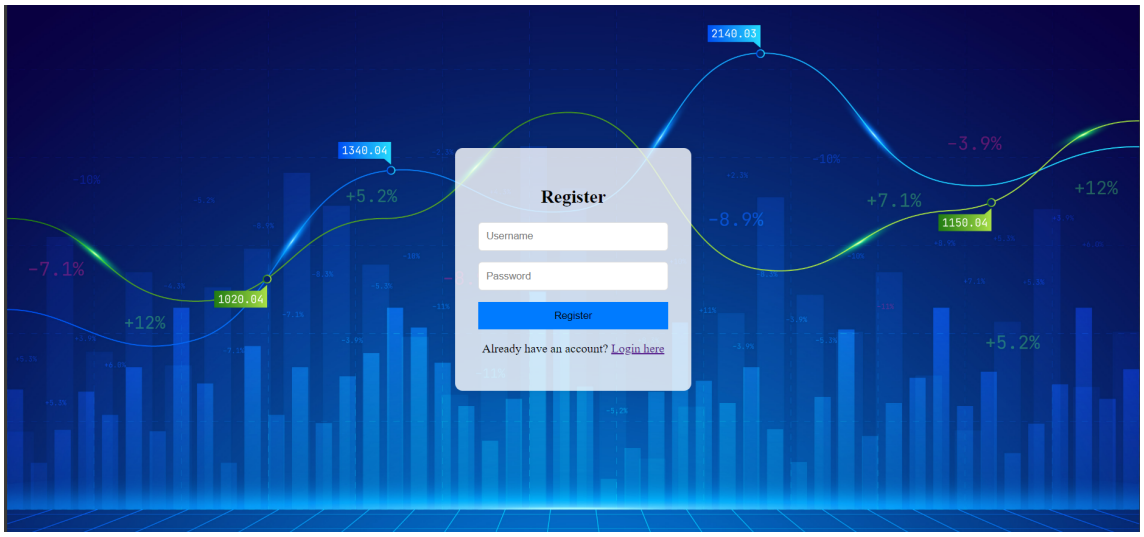

Figure 2: register.html

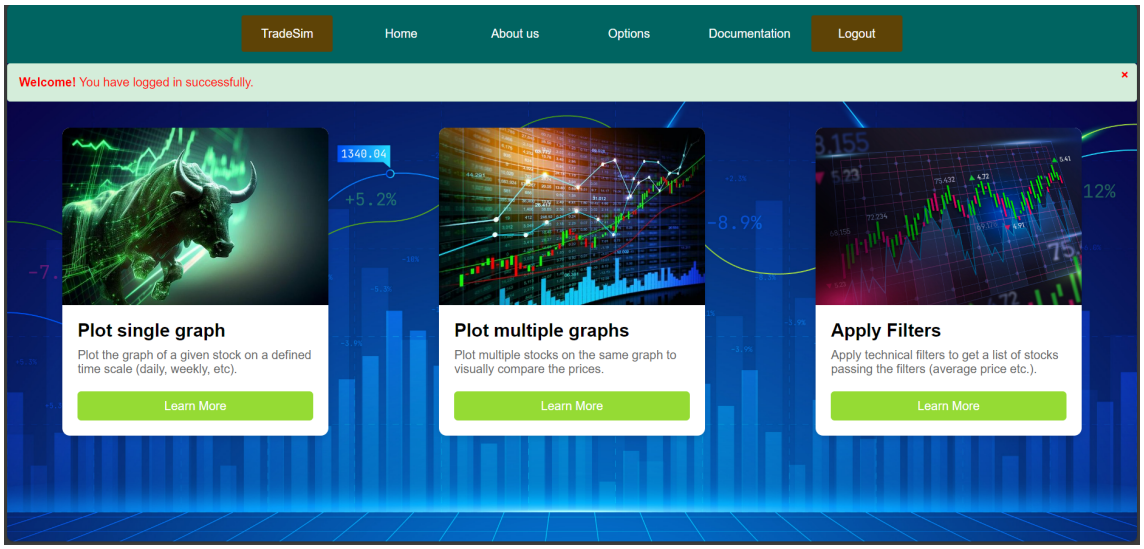

Figure 3: home.html

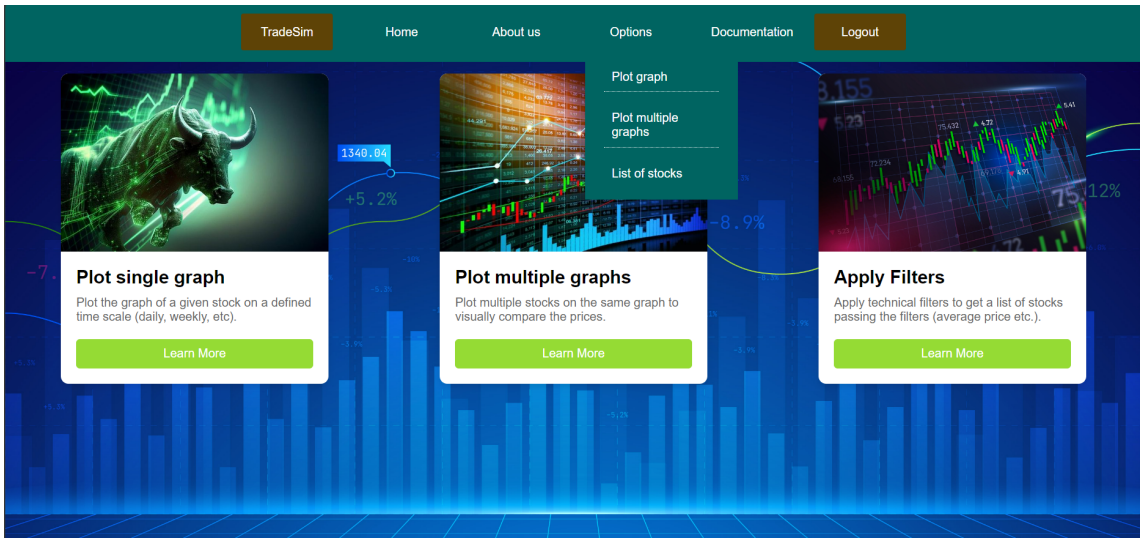

Figure 4: Dropdown menu

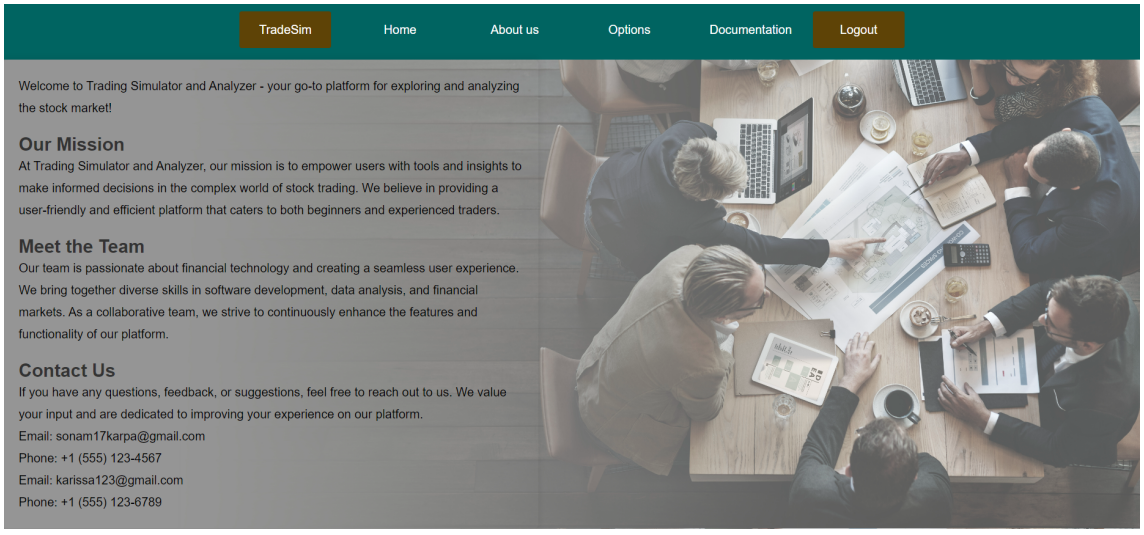

Figure 5: aboutus.html

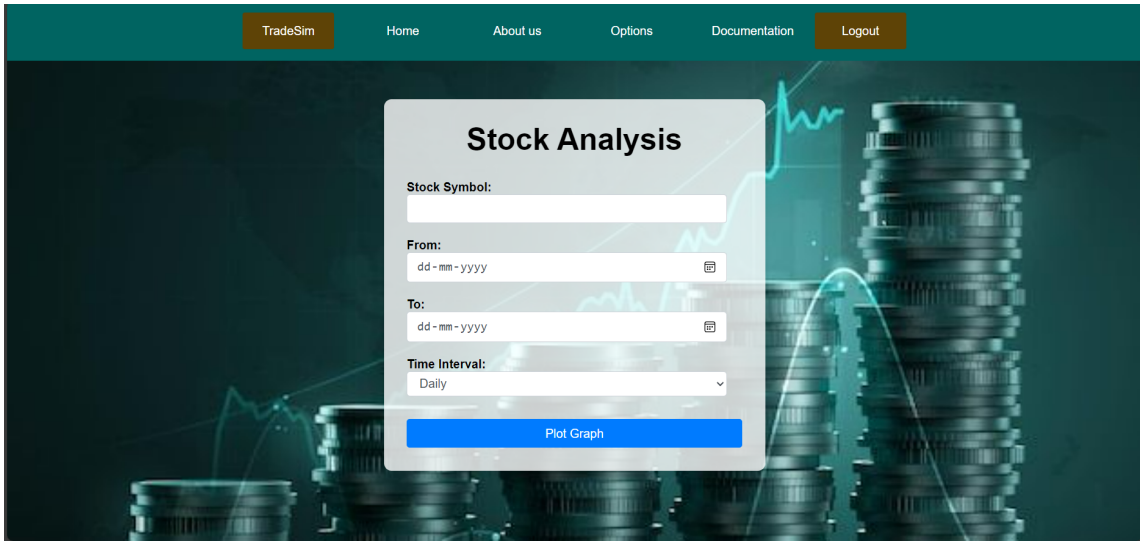

Figure 6: singlegraph.html

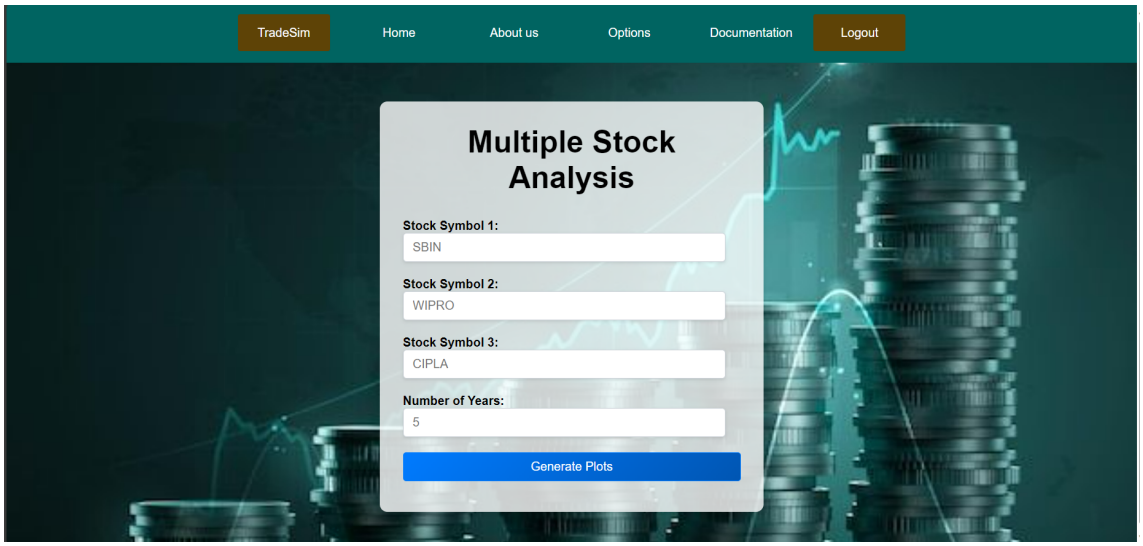

Figure 7: multiplegraphs.html

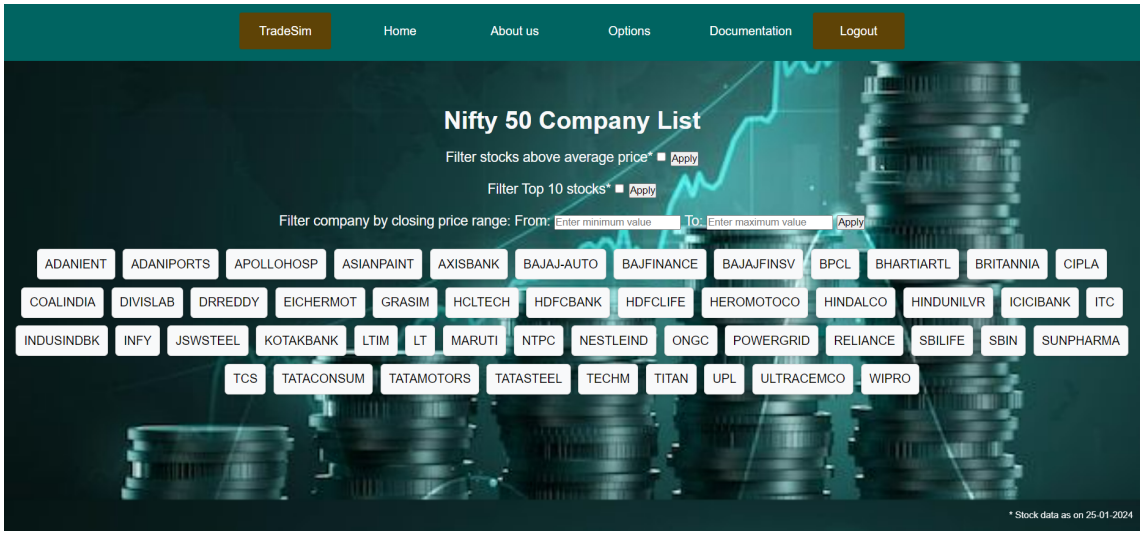

Figure 8: list.html

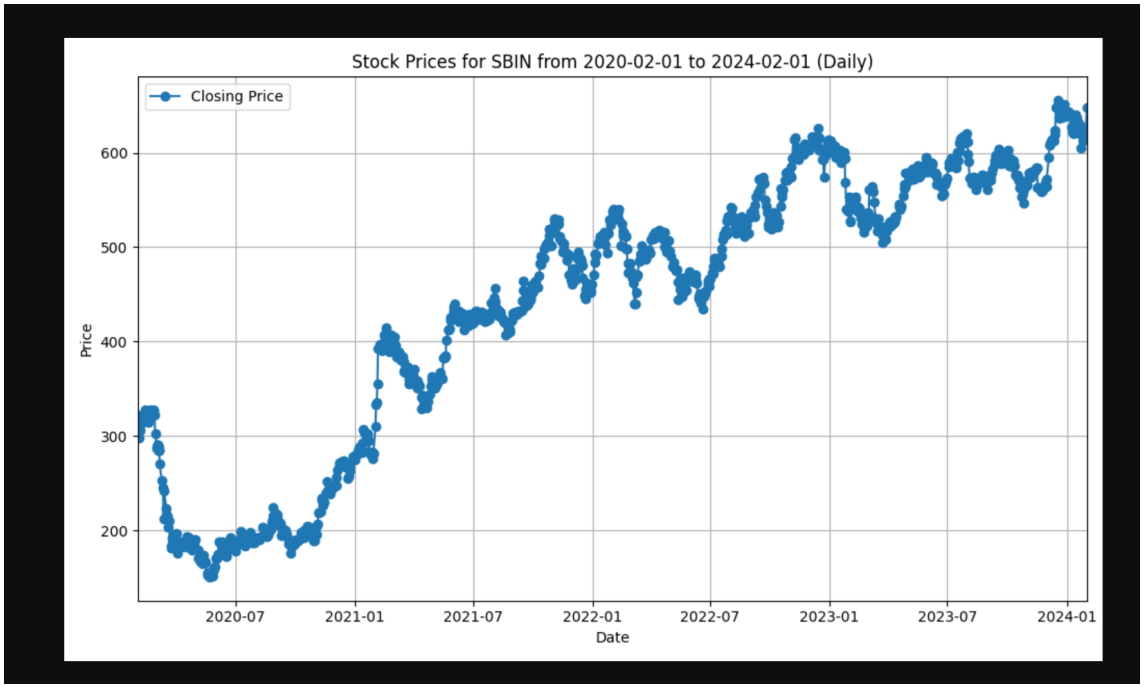

Figure 9: Sample output on single graph plot on daily scale

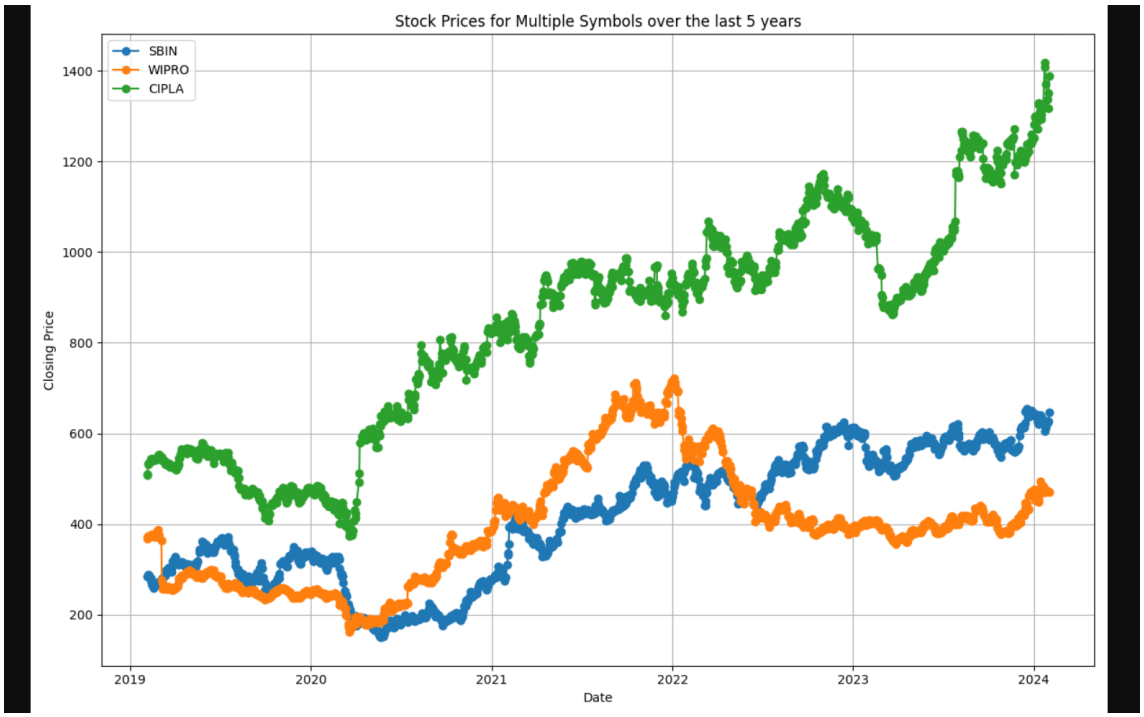

Figure 10: Sample output on multiple graph plots