Project Report - Milestone 3

Krishnanshu Jain
2019CS10368

Prakhar Jagwani
2019CS10382

Souvagya Ranjan Sethi
2019CS10405

30 April, 2023

Contents

1 Overall architecture

1. **Frontend**: We are writing the code in `TypeScript`, using the `React` library and styling using the `TailwindCSS` library.
2. **Backend**: The backend is in `python`. We are using the `Flask` library with the `FlaskRESTful` extension. To connect with the database, we are using `psycopg2`
3. **User Authentication**: We are using JWT tokens for user authentication. The backend checks the supplied token in some of the API requests, like those for viewing or editing user data.
4. **Data Visualization**: We are using `Chart.js` to create dynamic and interactive charts.

2 Application Frontend Design

Major web pages:

1. **LogIn and SignUp**: Users can gain access to an account from these pages. When a user logs in or signs up, the backend returns a JWT token which gets stored in the local storage. This token is sent with every query to the backend.
2. **Home**: Shows an overview of the market, with the top gainers for the week, the month and the year.
3. **Dashboard**: Shows user's details, current value of the portfolio and the stocks and mutual funds in it.
4. **Screener**: Allows the users to find stocks or mutual funds based on some filters. For example, they can put a filter to find the stocks with a `CAGR` of over `20%` in the last year.
5. **Compare**: Users can select two stocks/indices/mutual funds and track their returns over a period.
6. **Optimization**: Users can provide a basket of stocks, and our website will generate the distribution of stocks which have the best information ratio. We also show a comparison with the `NIFTY 50` index.
7. **Stocks & MF Pages**: A page for each stock, index, mutual fund and ETF. We show an overview (returns, CAGR, volatility, sharpe ratio, price history, etc.). Other parts of this page include:
   (a) **Backtesting**: Users can specify their swing trading strategy using a set of rules and see its performance over the specified period.
   (b) **Peers**: Users can see stocks in the same sector and ETFs tracking the same asset.
   (c) **Constituents**: Constituents of an index.
   (d) **SIP**: To see how an SIP would grow for a given stock/mf.

   Users can also add/remove/edit their portfolio by going to these pages.
8. **Admin Panel**: We can add a new day's data through this page.

3 Supported Transactions

We have written the following APIs:

1. To insert new data from CSV into the daily price relations and to refresh the views.
2. Given two stocks and a start date, list both of their prices on every following date.
3. Find all ETFs with the same underlying asset. Also returns their CAGR, Volatility and Sharpe Ratio fro comparisons.
4. Get all information about a specific stock, mutual fund, or ETF.
5. Get all constituents of an index, with information for each.
6. Check if a mutual fund, stock or ETF is in the database.
7. Check if a mutual fund, stock or ETF is in the curent user's portfolio. (We get the current user's id by decoding the JWT token in the header of the API request).
8. Insert/remove/edit a mutual fund, stock or ETF from a user's portfolio.
9. API that takes a list of filters to apply on the database of stocks and returns the filtered results
10. Get the price history of a mutual fund, stock or ETF, starting from a period of 1 month, 1 year, 5 years, or max possible time before.
11. Search for mutual funds or securities using their name or ticker.
12. Get the last price and price change for a mutual fund or security.
13. APIs for backtesting, portfolio optimization, and SIP.

4 Demonstration Scenario

Figure 1: User first lands on the home page, where they can see the current stock market

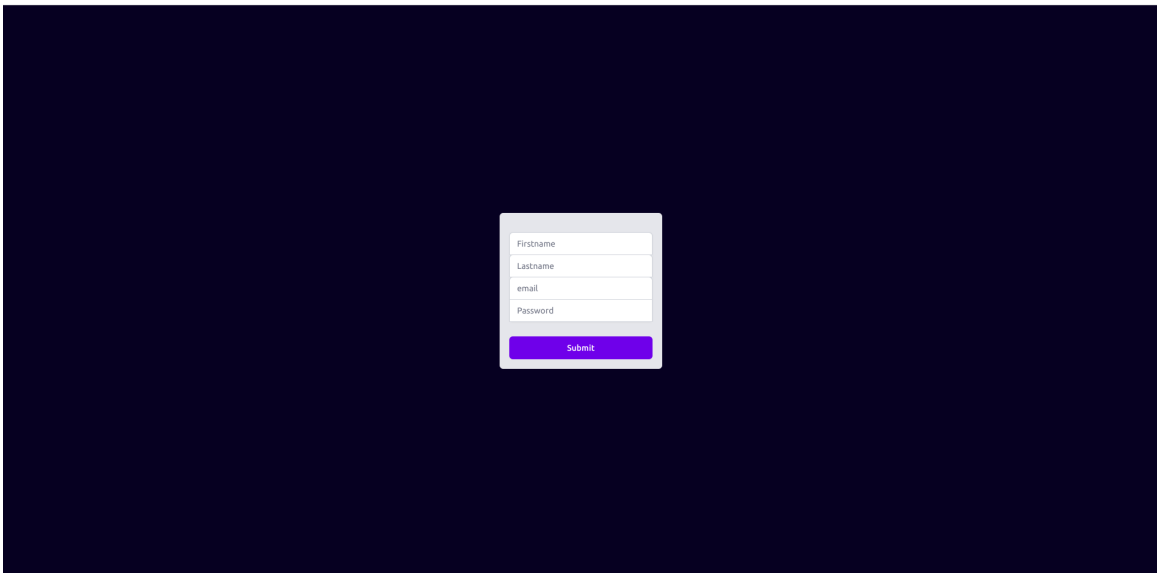

Figure 2: Users will then create an account and sign in

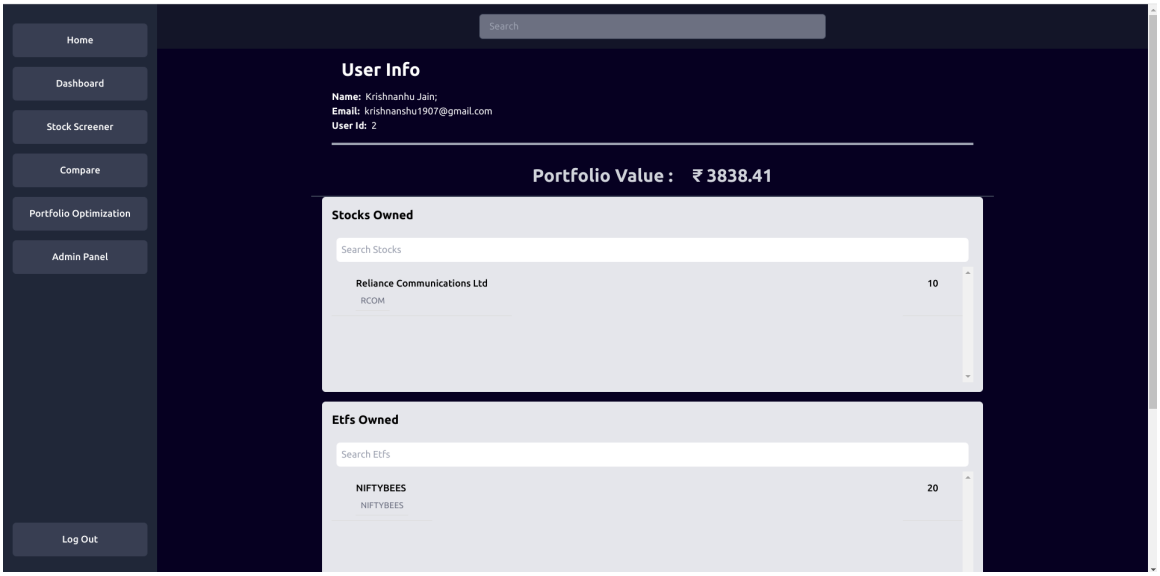

Figure 3: After Login, user is able to see its details,portfolio value and portfolio details

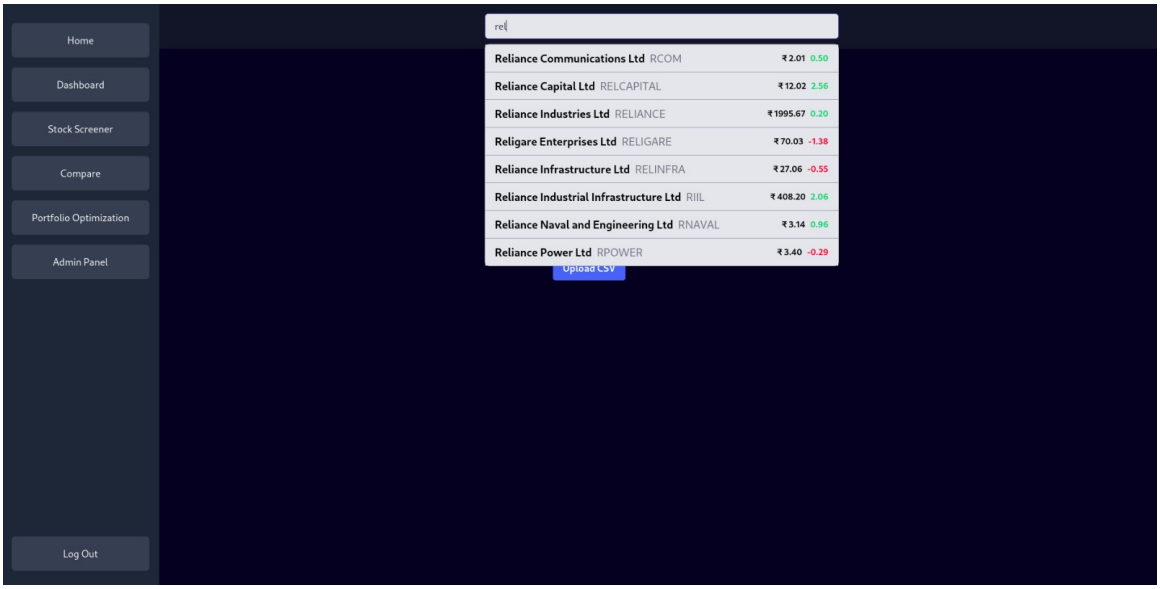

Figure 4: User can search for a stock/MF/ETF from the search bar

Figure 5: User can see how the stock has performed over the past month, Here it also shows user to add the selected

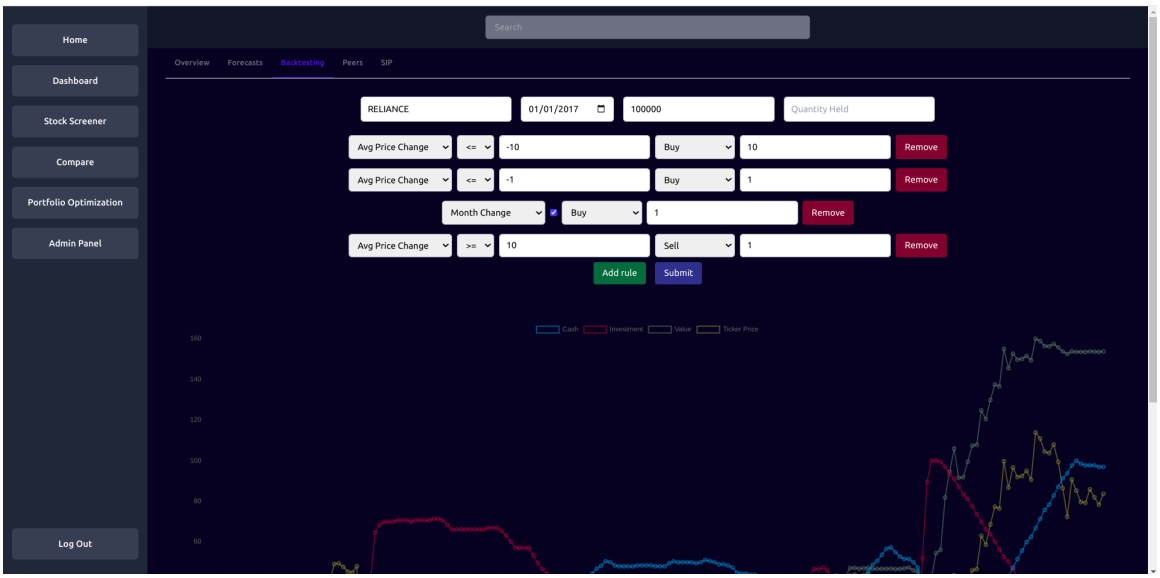

Figure 6: User may want to try a swing trading strategy. They will supply rules as follows: buy on dips, buy every month and sell when at 10% profit

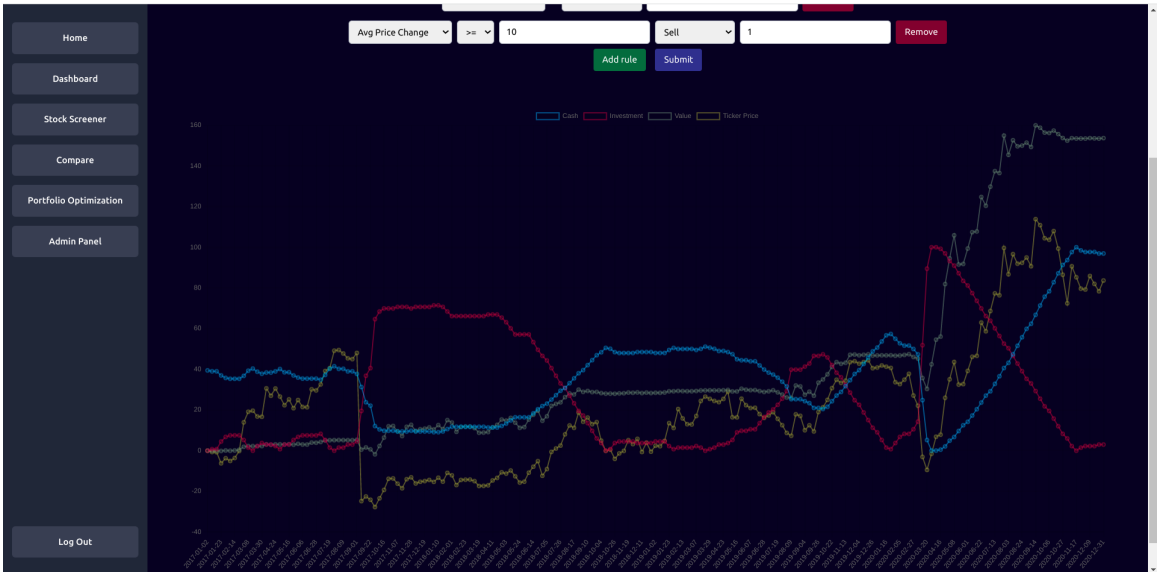

Figure 7: See how their strategy performs. At the end, the returns from this strategy (in green) are much higher than the returns from Reliance (in yellow). The blue line indicates the cash in their account and the red line is the number of stocks invested in.

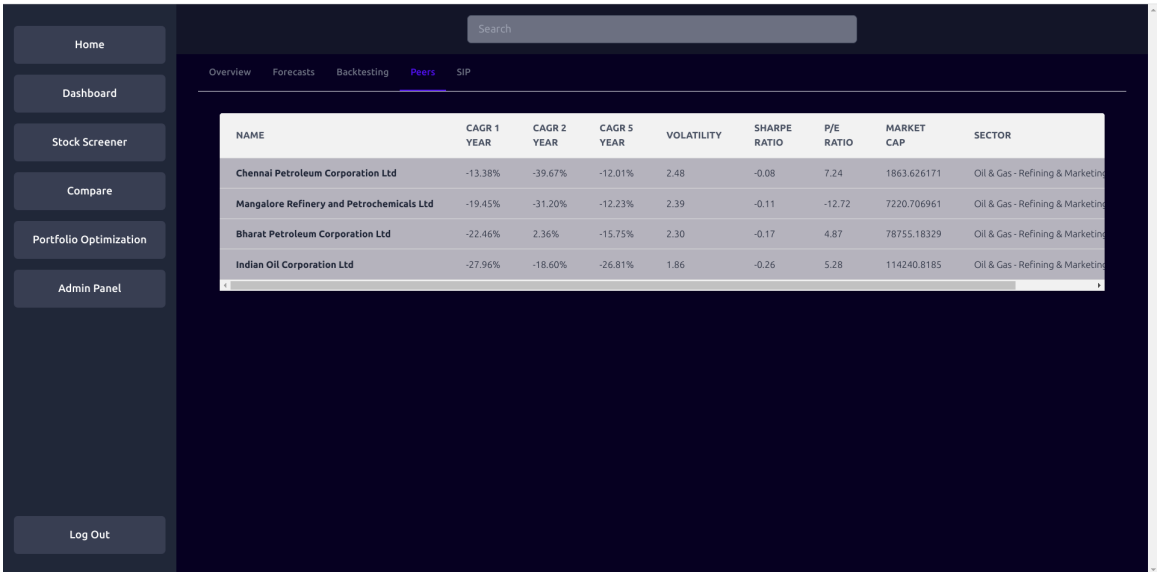

Figure 8: User can compare performance with other securities from the same sector

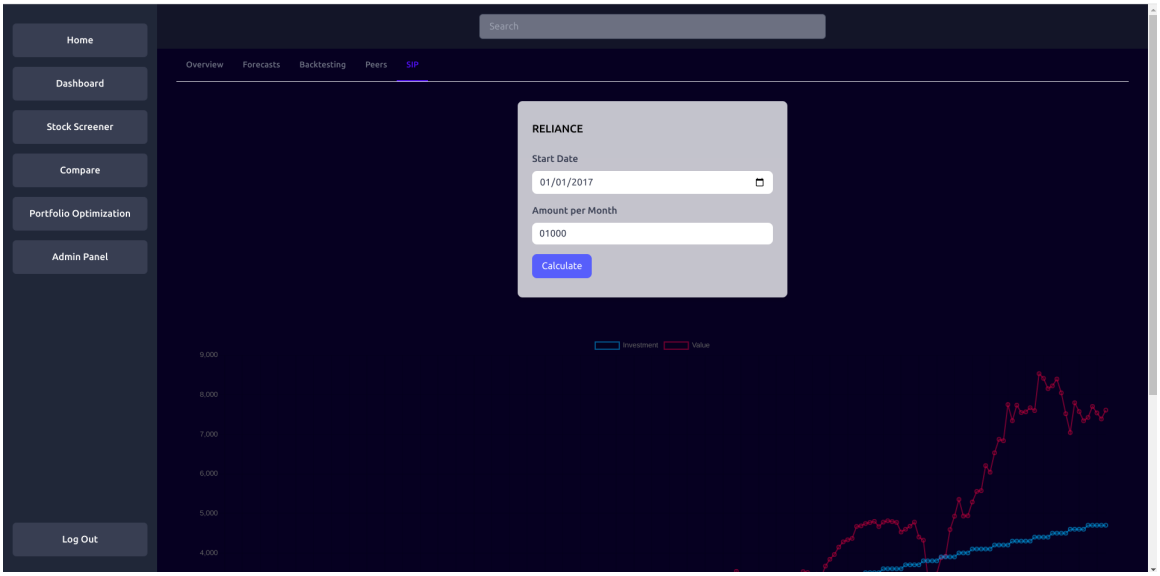

Figure 9: This tab allows the user to specify a SIP strategy for the selected security and see its performance over time.

Figure 10: The Graph represents the result of the above-specified SIP strategy. The blue line shows the amount invested and the Red line shows the current value.

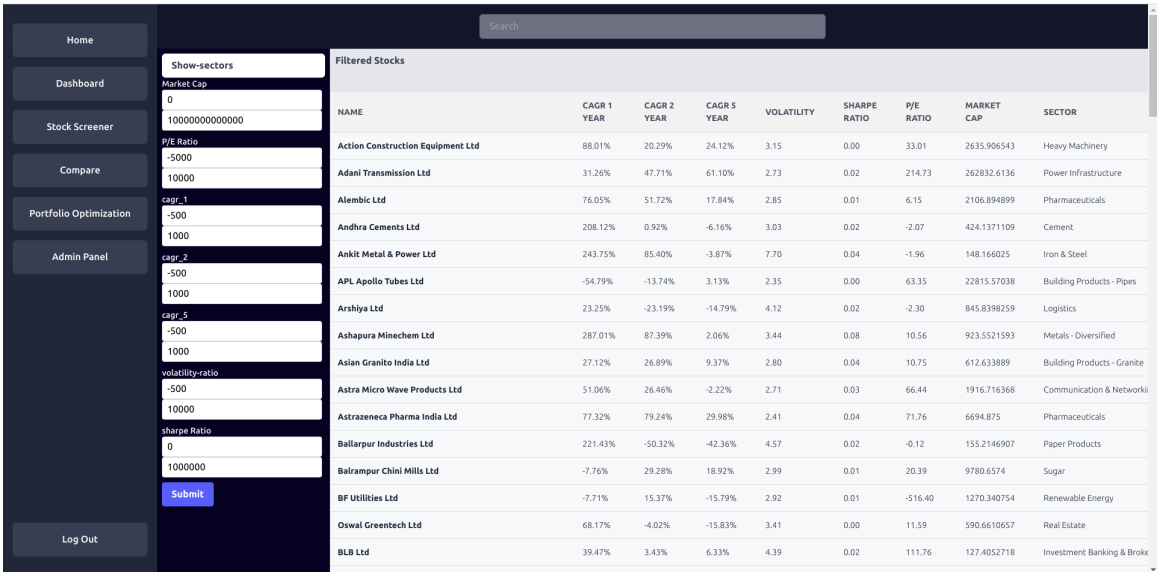

Figure 11: This page provides Various filters by which users can filter securities based on their preferences.

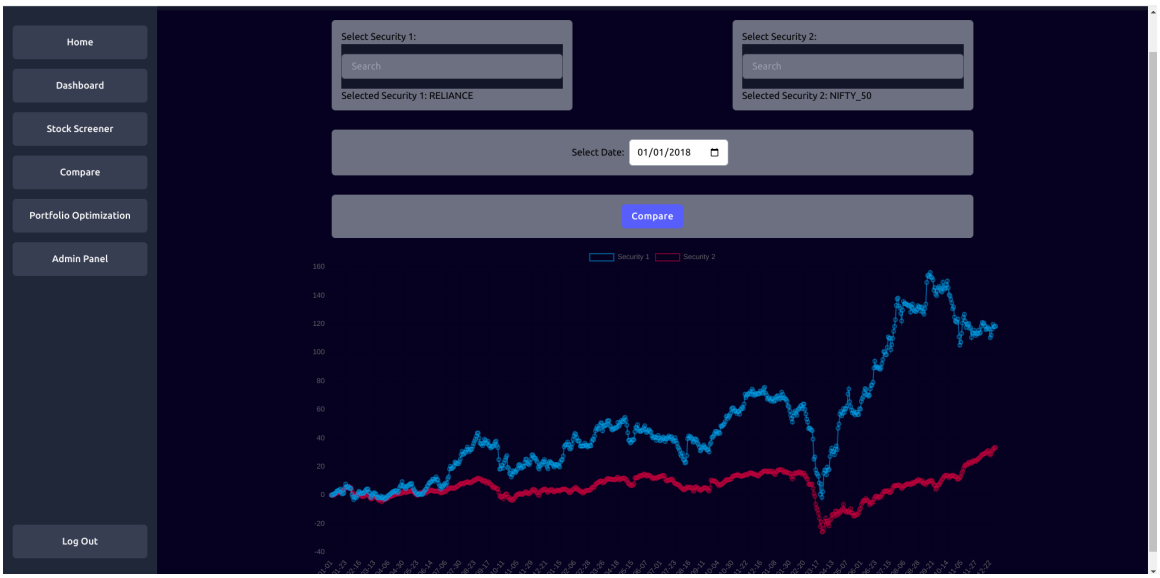

Figure 12: User can select two stocks and compare them

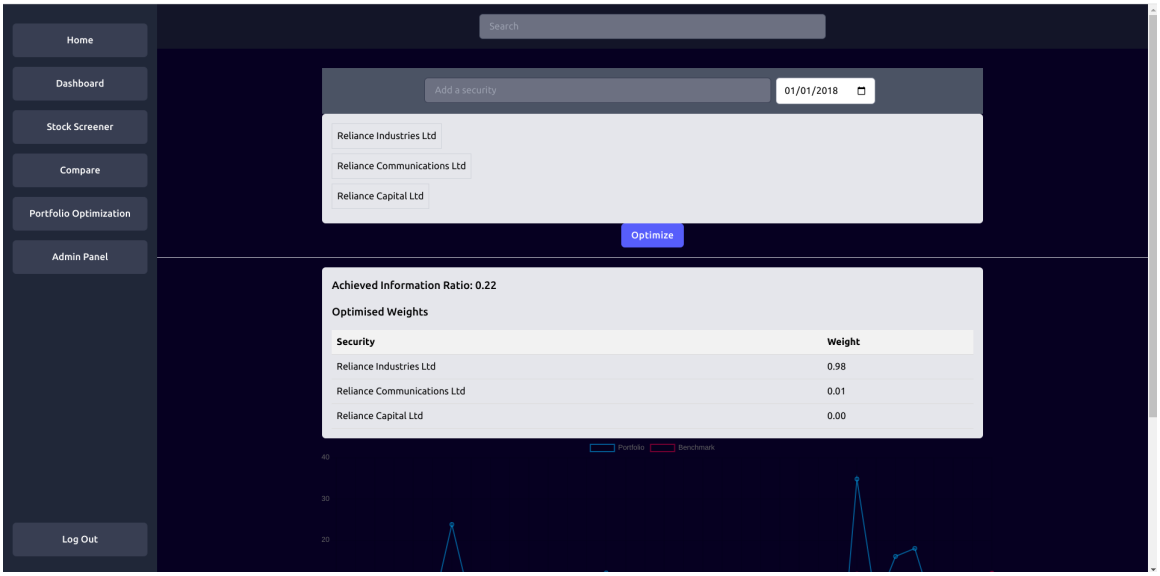

Figure 13: User wants to reorganize their portfolio. They will specify the stocks they are interested in

Figure 14: These are the weights for the optimal basket

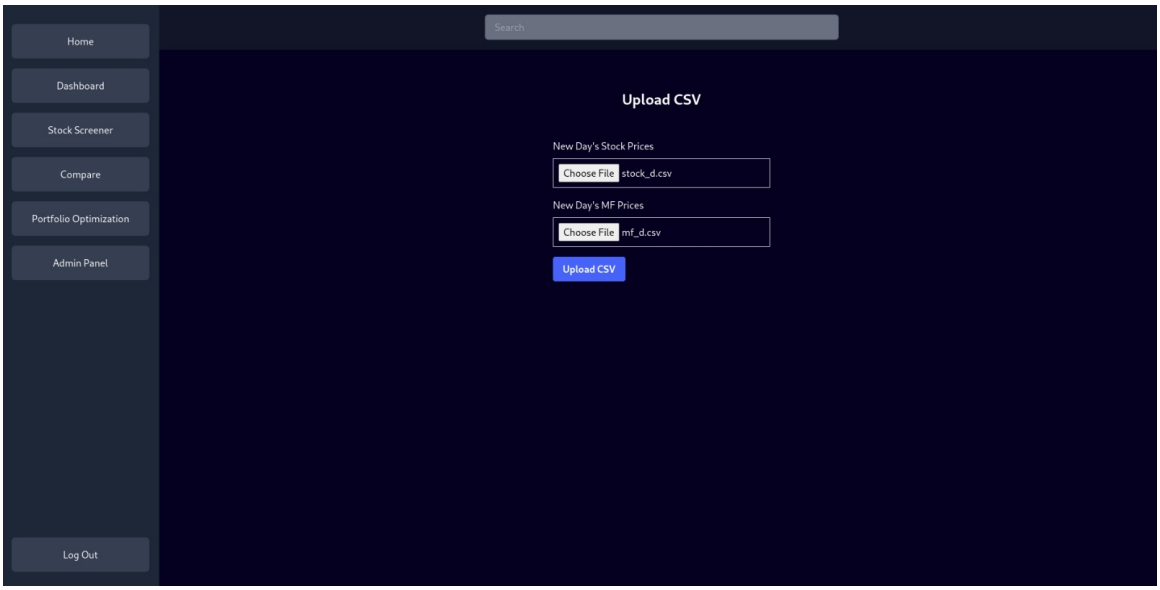

Figure 15: The admin can add a day's data through this page

Figure 16: Before adding date through the admin portal

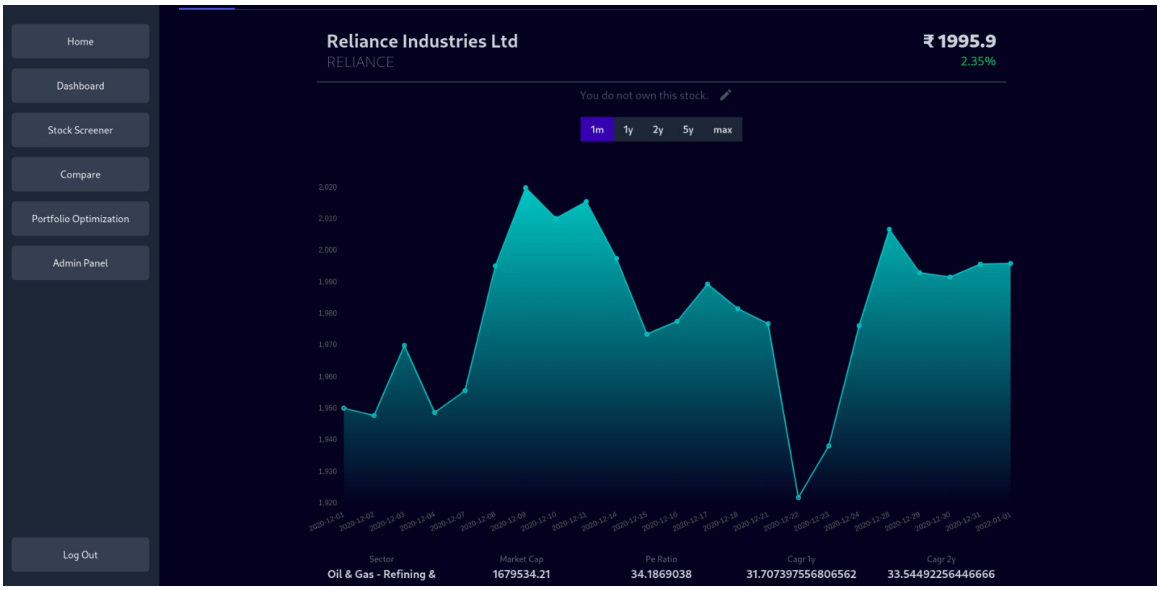

Figure 17: After adding date through the admin portal. A new date has been added.