STOCK MARKET TREND PREDICTION (WITH STOCK SCREENER)

ABSTRACT

This project aims to collect real-time data of any stock index and using a Machine Learning model, produce a prediction of the stock price's movement.

The produced prediction and real-time data of the stock is to be displayed on a webpage with the help of HTML, CSS, JavaScript, Bootstrap Technologies.

The API-provided data would be real-time but the predictions would be drawn up using just daily data.

Prediction output would be just 2 discrete values i.e.,

- 1: Price going Up
- 0: Price going Down or Neutral

1. Introduction

1.1. Purpose

This report serves as a functional overview to the different components of the project. It gives information regarding the project's data source, data preparation, techniques and finally end results.

1.2. Product Scope

This project was being worked upon with the aim of providing a solid prediction of any stock's trend along with a screener to give the user a complete experience without having to visit any other place.

2. Overall Description

2.1. Project Perspective

This project is built with the perspective of learning newer technological revelations like Machine Learning and combining it with our new found knowledge of web designing, databases, backend along with python programming.

2.2. Project Modules

Basic Interface having the following functionalities:

a. Simple Stock Data (Current MP, Open, Close, High, Low, etc).

b. Latest News pertaining to the selected stock.

3. Machine Learning

3.1. Data Collection

First and Foremost requirement of any machine learning model is Data.
Data Sources:

- API: AlphaVantage
- For initial predictions: Yahoo Finance

```
data = pd.read_csv('SENSEX.csv')
data = data.iloc[:,1:]
data.head()
```

	Open	High	Low	Close	Adj Close	Volume
0	14493.839844	14727.490234	14355.519531	14645.469727	14645.469727	28800.0
1	14694.309570	14764.349609	14469.690430	14658.490234	14658.490234	31200.0
2	14552.719727	14945.849609	14499.740234	14913.049805	14913.049805	29000.0
3	14962.120117	15097.870117	13959.440430	14043.400391	14043.400391	43400.0
4	14103.650391	14251.879883	14000.679688	14170.450195	14170.450195	35800.0

Fig 1: Data read using .csv file

Once Data is obtained, the next stage is Processing the data.

3.2. Data Preprocessing

To predict any stock, certain technical indicators are used.
The indicators selected vary from investor to investor.

Indicators used:

1. Moving Averages (Simple, Exponential)
2. Stochastic Indicator (R%, K%)
3. Moving Average Convergence/Divergence
4. Relative Strength Index
5. William's R%
6. Accumulation/ Distribution Index
7. Commodity Channel Index

```
X = data.iloc[:,1:]
X = X.drop('Adj Close',1)
X['SMA'] = X['Close'].rolling(window=10).mean()
X['EMA'] = X['Close'].ewm(span=20, adjust=False).mean()
X['STCK'] = momentum.stoch(X['High'], X['Low'], X['Close'], n=14, fillna=False)
X['STCD'] = momentum.stoch_signal(X['High'], X['Low'], X['Close'], n=14, d_n=3, fillna=False)
X['MACD'] = trend.macd(X['Close'], n_fast=12, n_slow=26, fillna=False)
X['RSI'] = momentum.rsi(X['Close'], n=10,fillna=False)
X['WR'] = momentum.wr(X['High'], X['Low'], X['Close'], lbp=14, fillna=False)
X['AD'] = volume.acc_dist_index(X['High'], X['Low'], X['Close'], X['Volume'], fillna=False)
X['CCI'] = trend.cci(X['High'], X['Low'], X['Close'], n=20, c=0.015, fillna=False)
X = X.iloc[25:,:]
X = X.dropna()
X.head()
normalized_X = preprocessing.normalize(X)
#normalized_X
X.head()
```

	High	Low	Close	Volume	SMA	EMA	STCK	STCD	MACD	RSI	WR	AD	CCI
25	4548.790039	4477.500000	4548.020020	0.0	4297.772998	4298.947369	99.826381	97.571757	33.774560	89.939986	-0.173619	0.0	202.067636
26	4605.410156	4505.060059	4523.009766	0.0	4324.960986	4320.286645	83.524218	93.304274	44.938459	82.090726	-16.475782	0.0	186.805109
27	4546.910156	4439.479980	4447.060059	0.0	4347.029981	4332.360303	68.338236	83.896278	47.688752	62.005934	-31.661764	-0.0	131.412153
28	4467.180176	4373.200195	4397.540039	0.0	4367.802979	4338.567897	54.787280	68.883245	45.710305	51.888912	-45.212720	-0.0	81.868255
29	4482.509766	4423.689941	4481.299805	0.0	4397.784961	4352.161412	72.198135	65.107884	49.718750	64.023879	-27.801865	0.0	97.693285

Fig 2: Code Snippet along with calculated indicators

Data Preprocessing:

- Normalization using in-built functions.
- Manually processing the data.

Manual method used in this project

Convert all the values according to below algorithm:

If previous_value<current_value

Then current_value=1

Else

current_value=0

	Close	SMA	EMA	STCK	STCD	MACD	RSI	WR	AD	CCI
0	15514.030273	-1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
1	15160.240234	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
2	15009.769531	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	1.0	-1.0
3	15074.589844	-1.0	-1.0	1.0	-1.0	-1.0	1.0	1.0	1.0	-1.0
4	15020.160156	-1.0	-1.0	1.0	-1.0	-1.0	-1.0	1.0	1.0	-1.0

Fig 3: Resultant Data

3.3. Model Selection & Results

Model implemented in this project:

- Logistic Regression
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Random Forest Classifier

```
logit = linear_model.LogisticRegression(solver='lbfgs')
logit.fit(X_train, y_train)
logit.score(X_test, y_test)
```

```
0.5258928571428572
```

Fig 4: Logistic Regression with 52.58% accuracy

```
model = discriminant_analysis.LinearDiscriminantAnalysis()
model.fit(X_train,y_train)
```

```
C:\Users\shail\Anaconda3\lib\site-packages\sklearn\discriminant_analysis.py:3
  warnings.warn("Variables are collinear.")
```

```
LinearDiscriminantAnalysis(n_components=None, priors=None, shrinkage=None,
              solver='svd', store_covariance=False, tol=0.0001)
```

```
model.score(X_test, y_test)
```

```
0.5294642857142857
```

Fig 5: Linear Discriminant Analysis with 52.94% accuracy

```
qda = discriminant_analysis.QuadraticDiscriminantAnalysis()
model = qda.fit(X_train, y_train)
```

```
C:\Users\shail\Anaconda3\lib\site-packages\sklearn\discriminant_a
  warnings.warn("Variables are collinear")
```

```
model.score(X_test, y_test)
```

```
0.49375
```

Fig 6: Quadratic Discriminant Analysis with 49.37% accuracy

3.4. Example of a Prediction

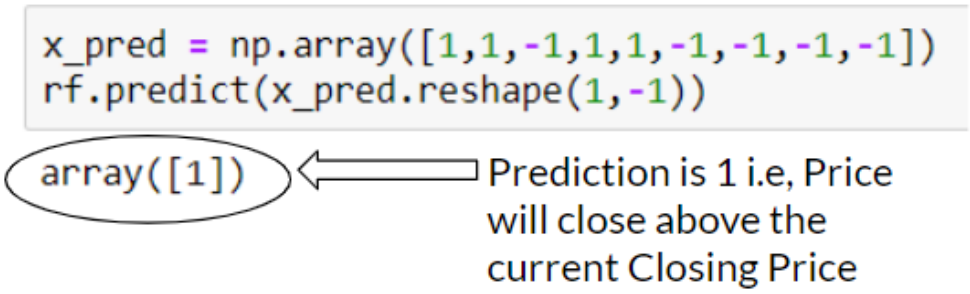


Fig 7: Code input and output as on 31st October 2019

Date	Open	High	Low	Close*	Adj. close**	Volume
01-Nov-2019	40,196.07	40,283.30	40,014.23	40,165.03	40,165.03	57,000
31-Oct-2019	40,211.99	40,392.22	40,054.89	40,129.05	40,129.05	87,90,000

Data from Yahoo Finance

Fig 8: Prediction verification from online source

4. Stock Screener

4.1. Screen Shots

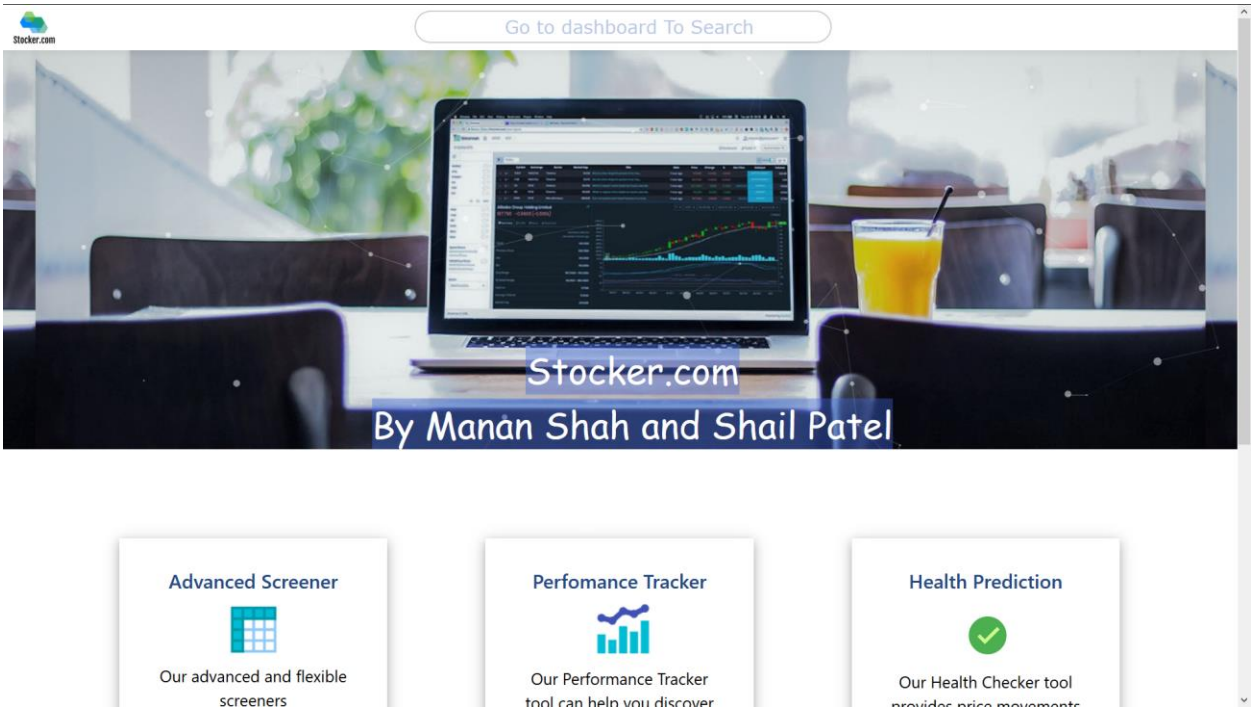


Figure 4.1: screenshot of website home page

Figure 4.2: screenshot of website home page-2

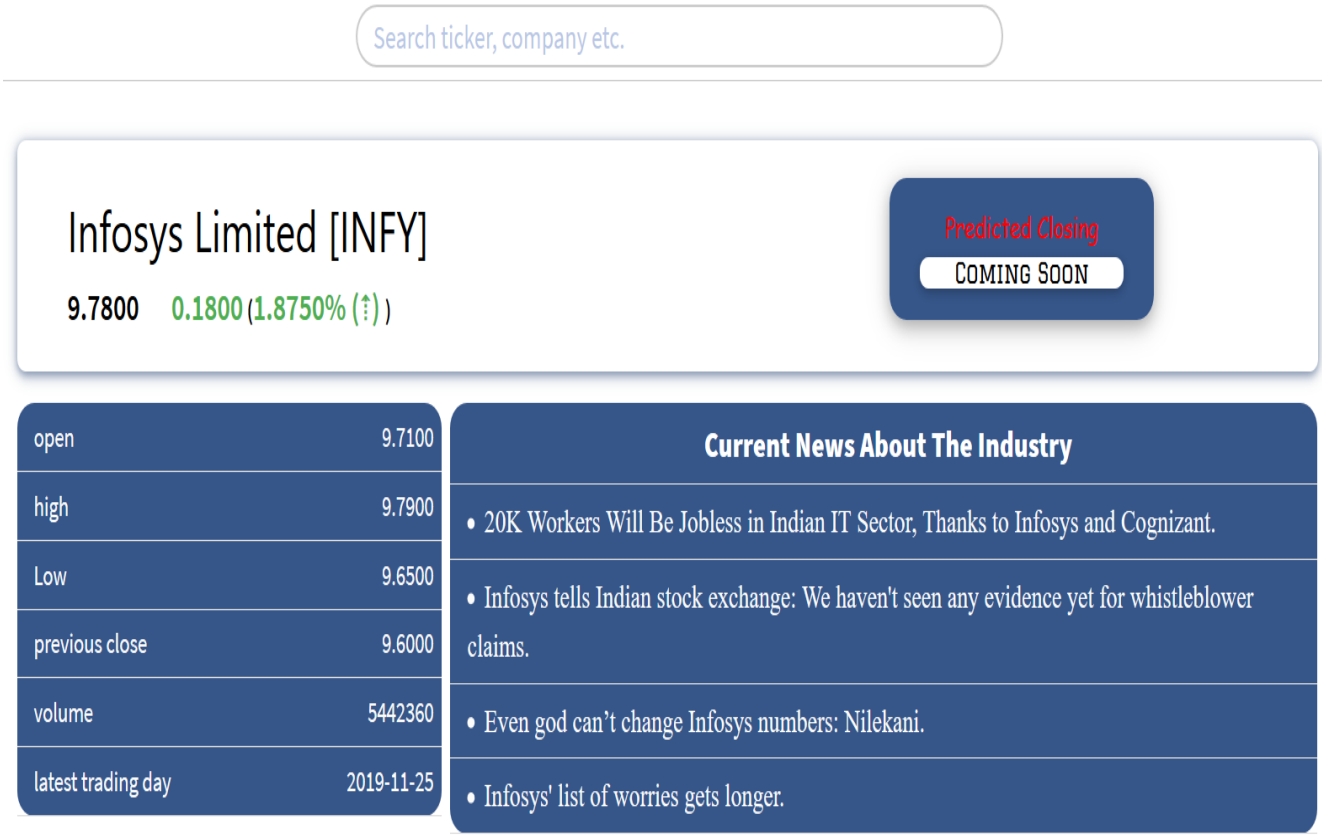
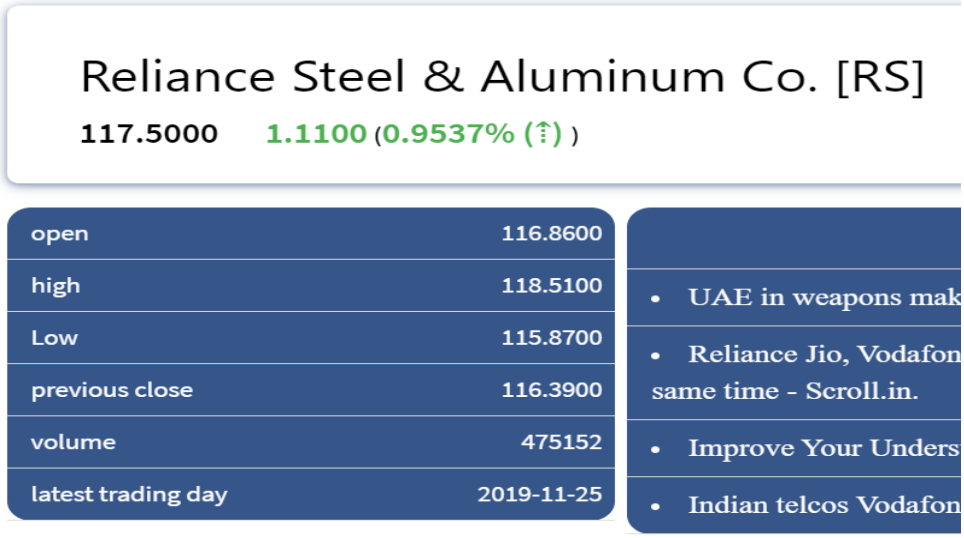


Figure 4.3 : screenshot of dashboard

4.2. Features

4.2.1. Live Data

We provide real time data of any stock at accuracy of 5s.
We provide Its Live Price, open, percentage decrease/increase, close, previous close, volume, last trading date, stock ticker,news.
We take data from api alpha vantage.



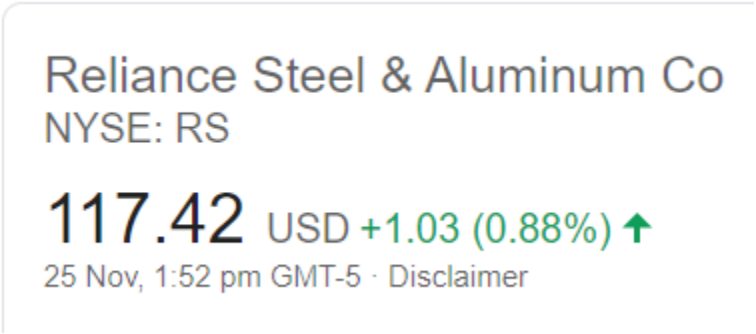


Figure 4.4 : live data ss Figure 4.5: live data google ss

4.2.2. Latest News of the company

With help of news api we provide currently trending news of the stock bearing company, as it would somewhat give inssights of the companies news.

Figure 4.6 : news ss1

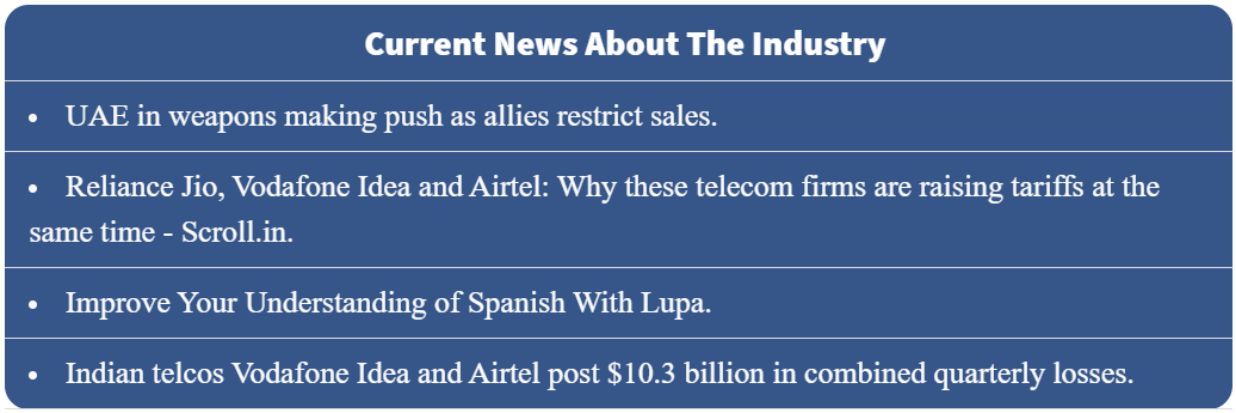


Figure 4.7: news ss2

As we can see we have news about reliance.
Clicking on the text would lead to the original article from where the data is being taken.

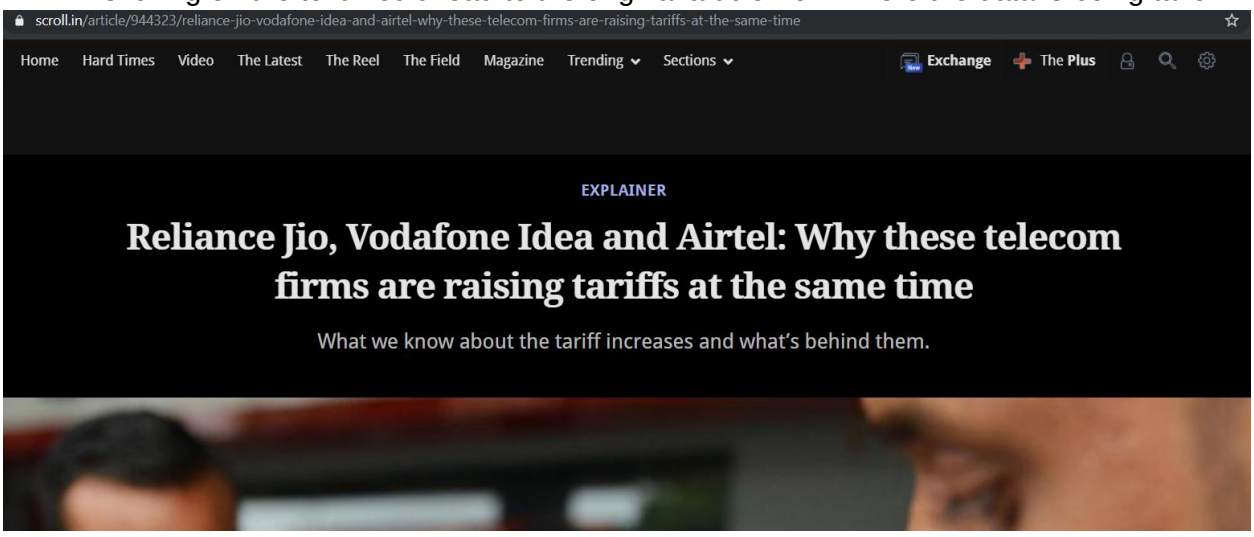


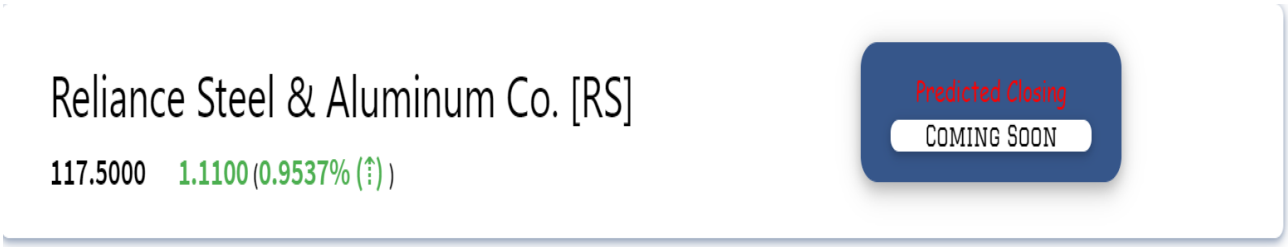
Figure 4.9: news ss3

4.2.3. Search Capability

The website provides capability of searching if also you do not the search ticker of the company.

4.2.4. Trend Prediction

We provide predicted trend of our data predicted from the machine learning model here



.Figure 4.10: predicted closing ss

4.3. Technologies Used

We have used html, css and javascript.

- HTML
  We have used it to generate basic web-page with content.
  Code Snipets:

```
<div class="w3-white" style="padding-left: 41px;padding-right: 41px;">
    <div class="w3-container my-card-paleblue w3-round-large w3-padding-16"
      style="margin-left: 109px;margin-right: 109px; margin-top: 2.5rem;">
      <!-- Div above table -->
      <div class="w3-container w3-cell-row w3-block w3-padding">
        <div class="w3-container w3-twothird w3-large s-bold w3-cell">
          <h1><span id="name">AAAA</span> [<span id="symbol">AA</span>]
          </h1>
          <span class="w3-xlarge" id="price">000</span>
            <span class="w3-xlarge w3-right-align w3-text-red"
id="change_red"></span><span
            class="w3-xlarge w3-right-align w3-text-green" id="change_green"></span>
          (<span class="w3-xlarge w3-right-align w3-text-red"
id="change_percent_red"></span><span
            class="w3-xlarge w3-right-align w3-text-green" id="change_percent_green"></span>)
        </div>
        <div class="w3-container w3-third w3-large s-bold w3-row">
          <div class="w3-container w3-card-4 w3-round-xlarge w3-padding-16 my-pale-blue w3-
center"
            style="width: 70%; height: 100%;">
            <span class="w3-cell-middle" id="predict"
              style="color: red; font-family: cursive;font-size: 100%;font-variant: normal; font-weight:
10;">Predicted
              Closing</span><br>
            <span class="w3-cell-middle w3-card-2 w3-white w3-round-large" id="predict"
              style="padding-right: 15%; padding-left: 15%; font-family: 'Graduate', cursive;font-size:
100%;font-weight: 600; font-weight: 550;">Coming
              Soon</span>

          </div>
        </div>
      </div></div>
```

- CSS
  We have used css to provide styling to the page.

We have used w3.css, which helped in maintaining uniformity in the code.
To bring up a theme based approach w3.css is very useful.

```
body {
   opacity: 1;
   transition: 5s opacity;
}

input{
   outline: none;
}

.my-blue,
.my-hover-blue:hover {
   color: #fff !important;
   background-color: #224886 !important
}

.my-pale-blue,
.my-hover-pale-blue {
   background-color: rgba(27, 63, 122, 0.884) !important;
   color: #fff !important;
}


.my-white-blue {
   background-color: #E8EDF0 !important;
   color: black !important;

}
```

- Javascript
  We mainly used javascript to bring data from the sources.
  Wwe used ajax for url queries.

```
function search20() {

  //
  //

  var search_ticker = document.getElementById('search').value;
  $.get("https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=" + search_ticker
+
    "&apikey=7GAS3OYJZBO54GSD",
    function (data1) {
      console.log(data1);
      console.log("HIHIHI");
      var symbol = data1.bestMatches[0]["1. symbol"];


      $.get("https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=" + symbol +
        "&apikey=7GAS3OYJZBO54GSD",
        function (data) {
          // checker = data["Global Quote"]["02. open"];
          if (data["Global Quote"]["02. open"] == undefined) {
            return false;
          }
          // console.log(data);
          $("#symbol").html(data1.bestMatches[0]["1. symbol"]);
          $("#name").html(data1.bestMatches[0]["2. name"]);
          //
          $("#open").html(data["Global Quote"]["02. open"]);
          $("#high").html(data["Global Quote"]["03. high"]);
          $("#low").html(data["Global Quote"]["04. low"]);
          $("#price").html(data["Global Quote"]["05. price"]);
          $("#volume").html(data["Global Quote"]["06. volume"]);
          $("#previous_close").html(data["Global Quote"]["08. previous close"]);

          $("#latest_trading_day").html(data["Global Quote"][
            "07. latest trading day"
          ]);

          $('#arrow').html("");
```

```
$("#change_red").html("");
$("#change_percent_red").html("");
$('#arrow').html("");
$("#change_green").html("");
$("#change_percent_green").html("");

//
//

if (data["Global Quote"]["09. change"] < 0) {
   // $('#arrow').html("&#8675;");
   $("#change_red").html(data["Global Quote"][
      "09. change"
   ]);
   $("#change_percent_red").html(data["Global Quote"][
      "10. change percent"
   ] + " (&#8675;)  ");
} else {
   $('#arrow').html("&#8673;");
   $("#change_green").html(data["Global Quote"][
      "09. change"
   ]);
   $("#change_percent_green").html(data["Global Quote"][
      "10. change percent"
   ] + " (&#8673;)  ");
}
```

4.4. Survey of trends

4.4.1. For text

The rule is that serifs are for print as well as sans-serifs are for screen. Sans has good readability, but trend has set it for websites more brands are turning towards bold serifs is some of designs such as headers. As serifs are decorative they emphasize the content.

4.4.2. Micro-interaction

Micro-interactions are to surprise the user and create inviting and humanly effects. A small action of user is responded by a small effect in return which is a micro interaction. Reloaded twitter page make beep sound which is a micro interaction, Reload of facebook feed gives you red view count which is micro-interaction.
These were common and basic interactions but there are heavy usage of it like scrolling animations, chimes, and much more. It makes the user more attentive and gives information about their actions and makes web pages look smarter and interactive or involving.

4.4.3. Design High Quality Symbol

By using svg there is trend to generate high quality symbols or logos for each use in website so that rescaling would not decrease the quality.

4.4.4. Color Schemes

In branding and website design, colors are too important.
It is seen in researches most instantan buys, basically 90% of them are by perception of colors.
To make a memorable, attractive,trustworthy and profitable website choosing right color scheme is a must. The first impression is what makes a website recognisable. The reason behind choosing any color should also be clear.
15 second span is basic span for average user to get his attention and tin that time making impression is what as a designers we need.

4.5. Implementation of trend

4.5.1. Fonts

We have used different fonts at different elements according to their effectiveness.

Eg. Helvetica
robotta

4.5.2. Interaction

With the use of particles github project, we have implemented interactions in the homepage. As we move our cursor over the image in homepage, users see particles gathering around the pointer.
We have transitions where divisions zoo-out generating effect.

4.5.3. Color Scheme

We have used blue color as our basic color in the web-site.
Our color scheme is blue and it's bright and dark shades with white color.
Having dark blue color in many backgrounds generate effect of dark mode.

Figure 4.11: blue theme ss

4.6. Features used in the web-site

4.6.1. Frame Work

We have used w3.css framework.
W3.CSS is a Cascading Style Sheet (CSS) developed by w3schools.com. It is helpful to design websites which are faster, beautiful, and respnsive.It is inspired from Google Material Design.

4.6.2. Data Source

- Alpha vantage: https://alphavantage.co/
  Composed of a tight-knit community of researchers, engineers, and business professionals, Alpha Vantage Inc. is a leading provider of free APIs for realtime and historical data on stocks, forex (FX), and digital/crypto currencies. Our success is driven by rigorous research, cutting edge technology, and a disciplined focus on democratizing access to data.
- News api: https://newsapi.org
  This api provides news search capability according to keyword, relevance, sentiment etc.

5. Future work

- To have graph capability with different indicators
- Provide personalised service by adding user accounts
- Adding facilities like watchlist

6. References & Websites:

- For stock data alphavantage api:https://www.alphavantage.co
- For news api : https://newsapi.org
- For content : https://websitesetup.org/website-color-schemes/
- For content : https://www.canva.com/colors/color-wheel/
- For Logo: https://www.freelogodesign.org/