A

MINI PROJECT REPORT

ON

Stock Price Prediction

Submitted by

Smit Ghelani (18IT401)

Parth Panchani (18IT410)

Jigar Shekhat (18IT427)

(Prof. Kanu Petel)

Faculty Guide

Information Technology Department

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

AY: 2020-21, Semester VI

Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology

DepartmentAY: 2019-20,

Semester VI

CERTIFICATE

This is to certify that project entitled with **Stock Price Prediction** has been successfully carried out by **Smit Ghelani(18IT401), Parth Panchani(18IT410), Jigar Shekhat(18IT427)** for the subject of **3IT31- Mini Project** under my guidance during the academic year 2020-21, Semester VI. The Mini Project work carried out by the students of 6th semester is satisfactory.

Date: 27/4/2021

Prof. Kanu Patel	**Prof. Nilesh Prajapati**	**Prof. Keyur Brahmbhatt**
Faculty GuideIT Department BVM	Course Coordinator IT Department BVM	Head of the Department IT Department BVM

Index

Chapter 1: Introduction

1.1 Aim of the Project:

Stock prices may depends upon past data of any stock. Past data defines growth rate of any stock so that our aim is to using the past data of any stock predicting the future price of stock by applying machine learning algorithm.

1.2 Scope of the project:

Prediction of stock's price using machine learning and deep learning will be useful for new investors as well as old investors to invest in stock market based on the various factors considered by the software. Demand and Supply of shares of a company is a major reason price change in stocks. When Demand Increase and Supply is less, price rises.

- **Overall Description:**

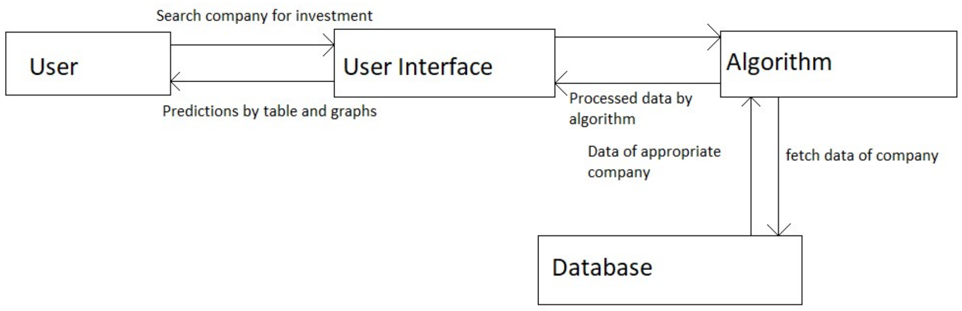

Search company for investment
User
User Interface
Algorithm
Predictions by table and graphs
Processed data by algorithm
Data of appropriate company
fetch data of company
Database

Fig. 1. Basic working of system

1.3 Objectives:

In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price.

This project will be implemented in Python. The system must be able to access a list of historical prices by nsepy python module. It must calculate the estimated price of stock based on the historical data. It must also provide an instantaneous visualization of the market index.

1.4 Modules:

1.4.1 Login:

The first page that users will encounter will be the login page. All users will be required to either login to an existing account or sign up for a new one. This page will show a simple box with two input areas where the user can enter their user name and password. Underneath this input are two buttons next to each other. The left button will be the Login button and the button next to it is the Sign-up button.

Clicking the Login button will either bring you to the home page if you entered the correct username and password or it will just refresh the login page with an error stating that you have entered your credentials wrong.

1.4.2 Sign Up / Registration:

On the login page, if a user is not already registered, they can sign up to create a new account by clicking on the Sign-up button. When clicked, this button will bring them to the registration page where they will be able to create a username and password. They will also be required to enter an email address and entering their name.

1.4.3 UI design:

User interface should be easy to use and user friendly for any system. UI is main attraction of any system. In the project user interact with the system through UI. User interface through user can get desired result very quickly and easily by just one click. The purpose of this module is to provide the user interface and view functions for the system. This is the system with which the user directly interacts and get prediction data. This module is created to provide the user interface to the system.

1.4.4 Stock search:

Through this user can search stocks which are registered under NSE. To get information about stocks, current trend of stock and predictions about stock, user has to search about the require stock by its name through the list menu. It is easy and convenient way to search about anything for user without fail.

1.4.5 Table and graph formation:

This section is where the graph of the stock's historical and prediction data will be displayed. The graph will have an indicator of the current data and which points on the graph represent the stock prediction. Underneath this section, will be a smaller section that will show the open, high, low, and current value of the stock for that given hour or however long we decide to refresh the data.

1.4.6 Price prediction:

Back side of system UI there should be a mechanism which find the prediction price for user. This mechanism is algorithm which will work for user to give satisfactory output about stock, so the output of algorithm guide user to invest in particular stocks. So, user can minimize the risk associated with stock market trading.

1.5 Project Basic Requirements:

1.5.1 Software Interface Requirements:

- **Python 3.8 with machine learning libraries** (e.g. scikit-learn, Keras, Tensorflow)

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python Libraries:

- **NumPy**: NumPy is basically a module or you can say a library that is available in python for scientific computing now it contains a lot of things it contains a powerful dimensional array object then

- **Keras:** Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make implementing deep learning models as fast and easy as possible for research and development.

- **Tensorflow**: TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

- **Scikit-learn:** Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

- **Pandas:** It is likewise a Python library for information control and examination. Specifically, it offers information structures and activities for controlling numerical table and time arrangement information.

- **Visual Studio Code / Sublime Text for programming**

  Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

- **Google Chrome for debugging web applications**

  Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, and was later ported to Linux, macOS, iOS, and Android where it is the default browser built into the OS. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.

1.5.2 Hardware Interface requirements:

Automated Stock Price Prediction may sound too complex but requires very few hardware costs, you just need a good computer with a good

editor and you're ready to go, not much requirement of extra hardware specifications.

Chapter 2: Analysis, Design Methodology and Implementation Strategy

2.1Comparison with Existing Application:

2.1.1 QuantShare: Budget Neural Network Stock Backtesting Software:

Pros:

- Automate Trading
- Sharing Servers for System Sharing
- Active Community

Cons:

- **Price: $245 One Time**
- Programming Knowledge Required
- Interface
- No User-Friendly UI

2.1.2 VectorVest: Solid Scanning & Market Timing Stock Trading Software:

Pros :

- Market Timing Signals
- Specific Buy & Sell Signals
- Simple to Follow System

Cons:

- **Price:** $69-$129
- Expensive, Especially with Add-ons
- Provides Signals with No Proven Track Record
- Limited Number of Chart Indicators
- No Easy to use

2.1.3 TrendSpider: Winner Best Automated Stock Chart Analysis Software

Pros:

- Automated Trendline & Fibonacci Detection
- Real-time Data Included
- Automatic Multi-Timeframe Analysis

Cons:

- **Price:** $27 -$69
- No Auto Trading (Yet)
- No Social
- Complex UI

2.2 Feasibility Study:

Stock market cannot be accurately predicted. The future, like any complex problem, has far too many variables to be predicted. The stock market is a place where buyers and sellers converge. When there are more buyers than sellers, the price increases. When there are more sellers than buyers, the price decreases. So, there is a factor which causes people to buy and sell. It has more to do with emotion than logic. Because emotion is unpredictable, stock market movements will be unpredictable. It's futile to try to predict where markets are going. They are designed to be unpredictable.

The proposed system will not always produce accurate results since it does not account for the human behaviors. Factors like change in company's leadership, internal matters, strikes, protests, natural disasters, and change in the authority cannot be taken into account for relating it to the change in Stock market by the machine.

The objective of the system is to give an approximate idea of where the stock market might be headed. It does not give a long-term forecasting of a stock value. There are way too many reasons to acknowledge for the long-term output of a current stock. Many things and parameters may affect it on the way due to which long term prediction is just not feasible.

2.2.1 Operational Feasibility:

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change. The operational feasibility is the one that will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. It measures the viability of a system in terms of the **PIECES** framework. The **PIECES** framework can help in identifying operational problems to be solved, and their urgency:

1. **Performance:** Does current mode of operation provide adequate throughput and response time?

   As compared to traditional methods of manually retrieving the stock data from the web and forecasting the stock prices with large number of manual calculations, this system plays a very important role in designing an application that automates the process of data retrieval and stock movement/price prediction with the help of a user-friendly dashboard, thus making the process easier and faster.

2. **Information:** Does current mode provide end users and managers with timely, pertinent, accurate and usefully formatted information?

   System provides end users with timely, pertinent, accurate and usefully formatted information. Since all the stock related information is being pulled from Yahoo Finance against a unique NSE Stock Symbol, it will provide for meaningful and accurate data to the investor. The investing decisions are made by the traditional investors manually. This results in loss of validity of data due to human error. The information handling and the investing decision in the proposed system will be driven by computerized and automatically updated prediction and validation of stock data. The human errors will be minimal. The data will be automatically updated from time to time and will be validated before the data is processed into the system.

3. **Economy:** Does current mode of operation provide cost-effective information services to the business? Could there be a reduction in costs and/or an increase in benefits?

   Determines whether the system offers adequate service level and capacity to reduce the cost of the business or increase the profit of the business. The deployment of the proposed system, manual work will be reduced and will be replaced by an IT savvy approach. Moreover, it has also been shown in the economic feasibility report that the recommended solution is definitely going to benefit economically in the long run. The system is built on Excel, R and JavaScript. Excel and Javascript do not need any additional installation; they are in-built in every system. R needs installation but it is free software. So, overall, the application is very economically feasible.

4. **Control:** Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?

   As all the data is pulled from Yahoo Finance, which is a public stock data provider, it does not contain any confidential information which can be misused, so on that contrast there should be no use of any security corner for this system.

5. **Efficiency:** Does current mode of operation makes maximum use of available resources, including people, time, and flow of forms?

   Efficiency work is to ensure a proper workflow structure to store patient data; we can ensure the proper utilization of all the resources. It determines whether the system makes maximum use of available resources including time, people, flow of forms, minimum processing delay. In the current system a lot of time is wasted as the investing decisions are made by the traditional investors manually. The proposed system will be a lot efficient as it will be driven by computerized and automatically updated prediction and validation of stock data. The data will be automatically updated from time to time and will be validated before the data is processed into the system.

6. **Services:** Does current mode of operation provide reliable service? Is it flexible and expandable?

   The system is desirable and reliable services to those who need it and also whether the system is flexible and expandable or not. The proposed system is very much flexible for better efficiency and performance of the organization. The scalability of the proposed system will be inexhaustible as the storage capacity of the system can be increased as per requirement. This will provide a strong base for expansion. The new system will provide a high level of flexibility.

2.2.2 Technical Feasibility:

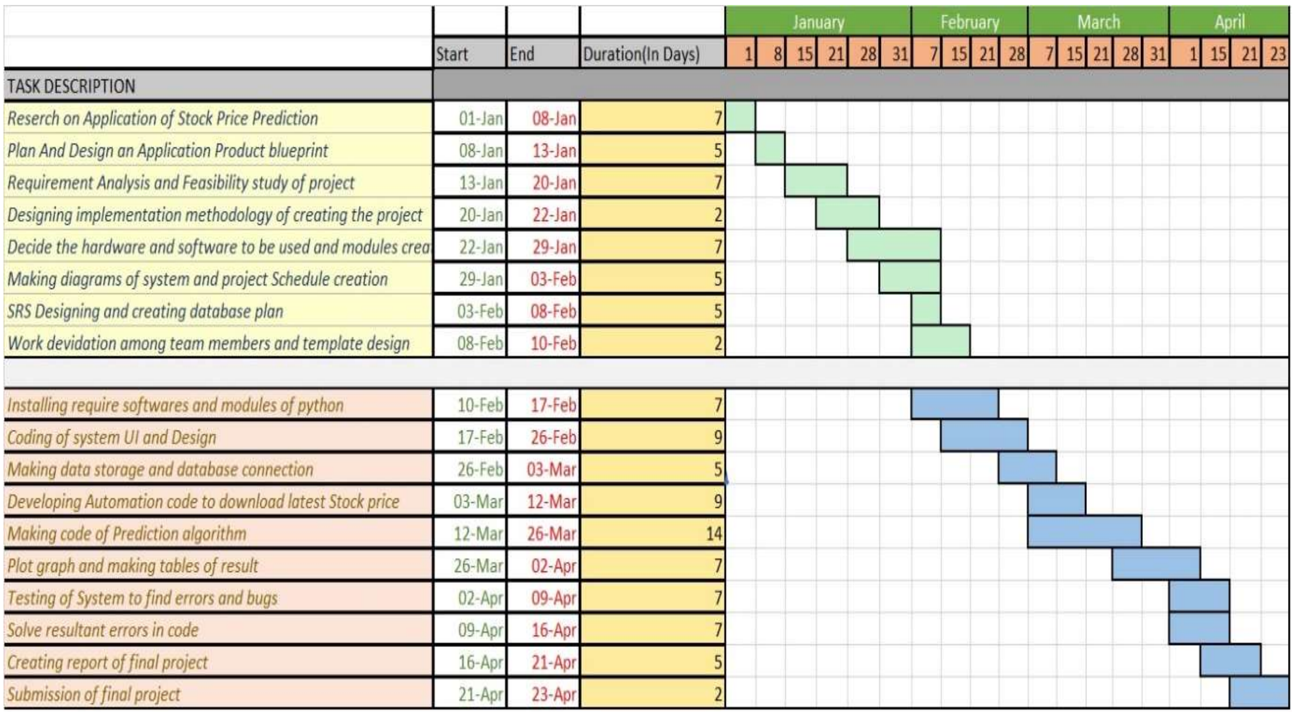
A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The developer must find out whether current technical resources can be upgraded or added to in a manner that fulfils the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility. The essential questions that help in testing the technical feasibility of a system include the following:

1. Is the project feasible within the limits of current technology?
2. Does the technology exist at all?
3. Is it available within given resource constraints?
4. Is it a practical proposition?
5. Manpower- programmers, testers & debuggers
6. Software and hardware
7. Are the current technical resources sufficient for the new system?
8. Can they be upgraded to provide to provide the level of technology necessary for the new system?
9. Do we possess the necessary technical expertise, and is the schedule reasonable?
10. Can the technology be easily applied to current problems?
11. Does the technology have the capacity to handle the solution?
12. Do we currently possess the necessary technology?

Automated Stock Prediction system deals with the modern technology system that needs the well efficient technical system to run this project. All the resource constrains must be in the favor of the better influence of the system. Keeping all this fact in mind we had selected the favorable hardware and software utilities to make it more feasible.

2.3 Project Timeline chart:



TASK DESCRIPTION	Start	End	Duration(In Days)
Reserch on Application of Stock Price Prediction	01-Jan	08-Jan	7
Plan And Design an Application Product blueprint	08-Jan	13-Jan	5
Requirement Analysis and Feasibility study of project	13-Jan	20-Jan	7
Designing implementation methodology of creating the project	20-Jan	22-Jan	2
Decide the hardware and software to be used and modules crea	22-Jan	29-Jan	7
Making diagrams of system and project Schedule creation	29-Jan	03-Feb	5
SRS Designing and creating database plan	03-Feb	08-Feb	5
Work devidation among team members and template design	08-Feb	10-Feb	2
Installing require softwares and modules of python	10-Feb	17-Feb	7
Coding of system UI and Design	17-Feb	26-Feb	9
Making data storage and database connection	26-Feb	03-Mar	5
Developing Automation code to download latest Stock price	03-Mar	12-Mar	9
Making code of Prediction algorithm	12-Mar	26-Mar	14
Plot graph and making tables of result	26-Mar	02-Apr	7
Testing of System to find errors and bugs	02-Apr	09-Apr	7
Solve resultant errors in code	09-Apr	16-Apr	7
Creating report of final project	16-Apr	21-Apr	5
Submission of final project	21-Apr	23-Apr	2

Fig.2. Time line chart of system

2.4 Modules:

2.4.1 Login:

Initially user can see and access the content of home page, If user want to access any other tabs from this application. User must need to login first. This page will show a simple box with two input areas where the user can enter their user name and password. Underneath this input are two buttons next to each other. The left button will be the Login button and the button next to it is the Sign-up button.

Clicking the Login button will either bring you to the home page if you entered the correct username and password or it will just refresh the login page with an error message that you have entered invalid credentials.

2.4.2 Sign Up / Registration:

On the login page, if a user is not already registered, they can sign up to create a new account by clicking on the Sign-up button. When clicked, this button will bring them to the registration page where they will be able to create a username and password. They will also be required to enter an email address and enter their name.

In this page we have set some basic requirements to create new account such that username must me alphabetic. Numeric and special characters are not acceptable for username. For email id we have to enter email-id in valid format i.e. username123@xyz.com

And we have added most common exception that user should be unique no existing credential are valid for new account creation.

2.4.3 UI design:

User interface should be easy to use and user friendly for any system. UI is main attraction of any system. In the project user interact with the system through UI. User interface through user can get desired result very quickly and easily by just one click.

So we made user friendly environment by adding all the functionality at the top of the page and bottom of the page and also our contact details to reach to us if any query, help needed. So user can easily move from one tab to another by just one click. Also we have added user and admin profile section so that user can manage his or her profile and see the his or her history and at the admin side we have set some authority to add, remove, track users so we can maintain the privacy by removing suspicious user.

2.4.4 Stock search:

Through this user can search stocks which are registered under NSE. To get information about stocks, current trend of stock and predictions about stock, user has to search about the require stock by its name through the options of stocks. And user also can set some duration to find specific amount of data about stock.

Duration field we have set one condition that user only enter numeric integer value otherwise it shows error message. And result of stock data will display just below the stock selection menu and duration field in tabular form.

2.4.5 Table and graph formation:

This is basically sub module of stock price prediction but we have separate it out for easiness. In this we get some data about stock price prediction and display in form of table and graph so user can easily got the idea about future behavior of stocks.

In the graphs we display the past 2 year closed prices with date and continue the data with prediction values. We have distinguished the past and predicted prices by different colors. Just below to the graph we will display two tables such as past values with date and predicted value with date.

2.4.6 Price prediction:

Back side of system UI there should be a mechanism which find the prediction price for user. This mechanism is algorithm which will work for user to give satisfactory output about stock.

- In this module we simply collect past 2 year data using nsepy library of python. And using this data we separate out closed prices and volume of data.

- After this we create prediction column and apply value of closed prices of data which is registered after prediction duration.
- Why we are doing this ? because we have to first train our model for predicting the value by adding prediction value so model train it self to find prediction as close as past values.
- Then we normalize our data to less then 1. So that it consume less time to process.
- After this we divide this data into 80/20 train and test model. And this 80/20 division is random sample.
- Then apply train values to train model which internally set slope and intercept using gradient decent method for future prediction.
- Then we apply latest 120 data to predict next 120 days values.
- Then this return 120 prediction values and we apply it timestamp and send to table and graph module for data visual.

2.5 Project SRS:

2.5.1 Use Case diagram:

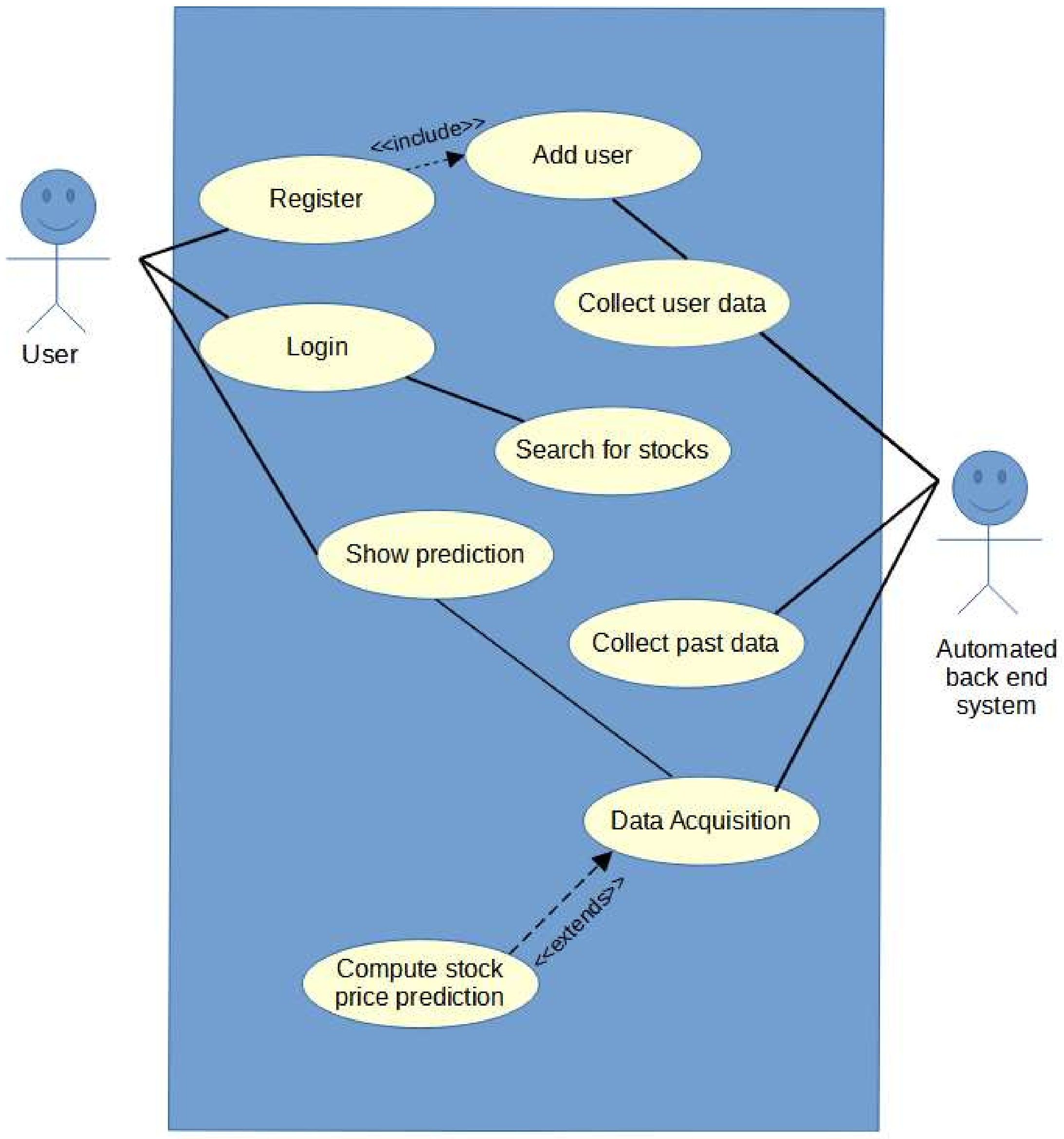


Fig.3. Use Case Diagram

2.5.2 Data Flow Diagram:

Level-0:

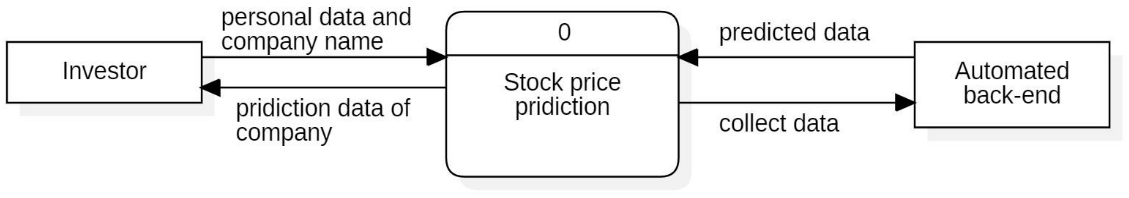


Fig.4. Data Flow Diagram (Level-0)

Level-1:

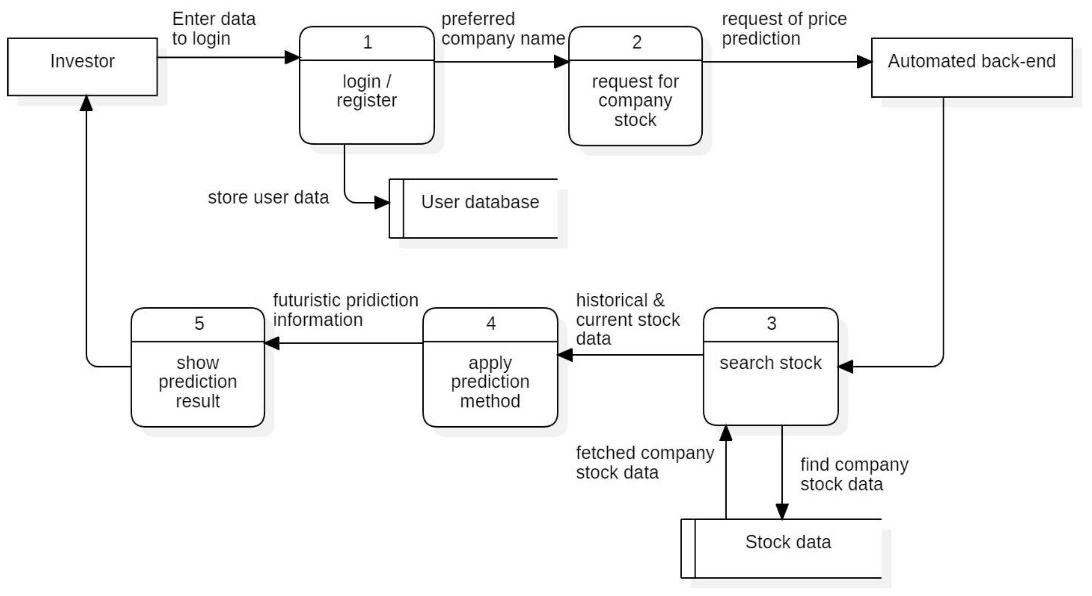


Fig.5. Data Flow Diagram (Level-1)

2.5.3 Entity Relationship Diagram:

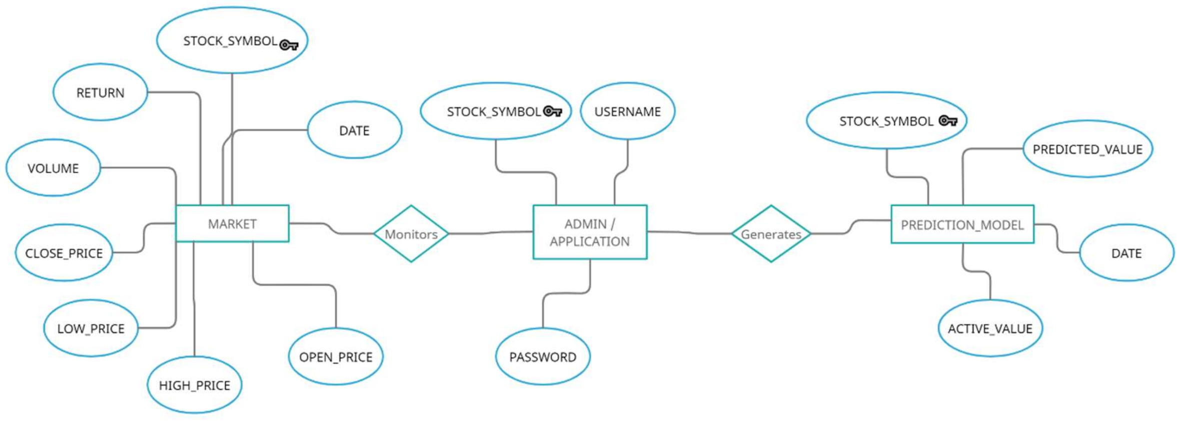


Fig.5. ER Diagram

2.5.4 Event Trace Diagram:

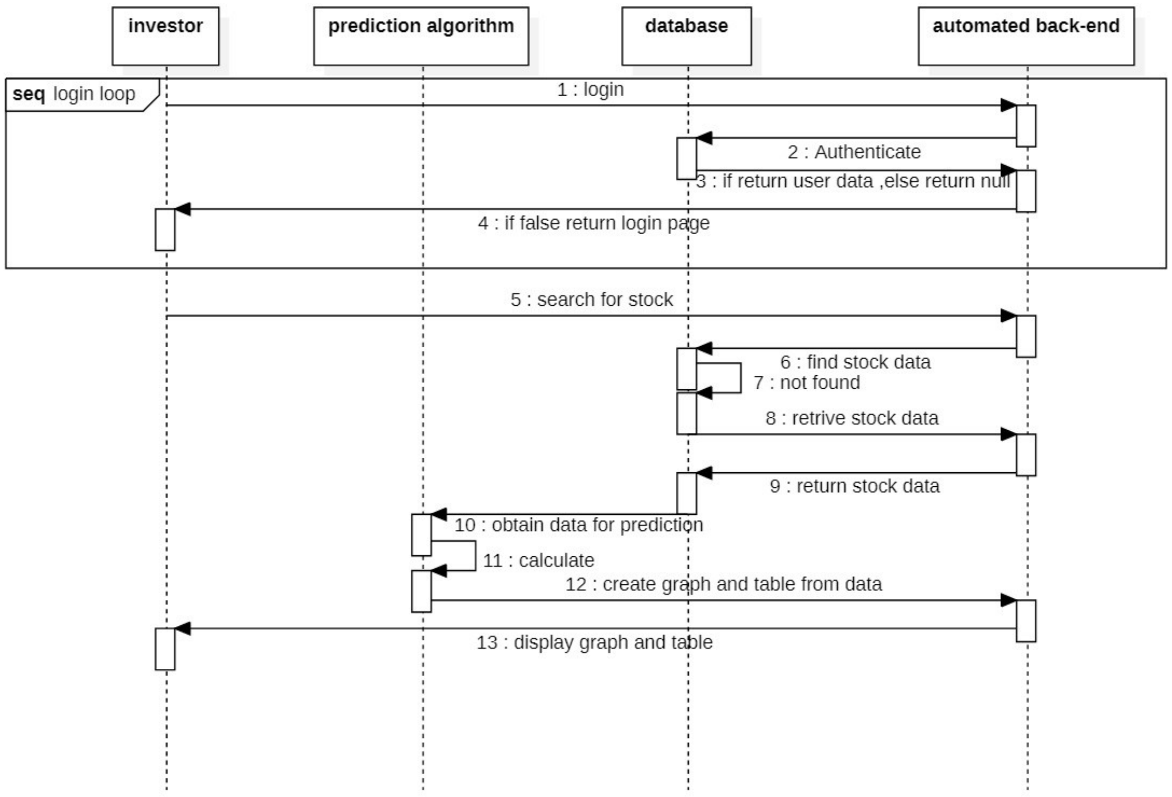


Fig.7. Event Trace Diagram

2.5.5 Class Diagram:

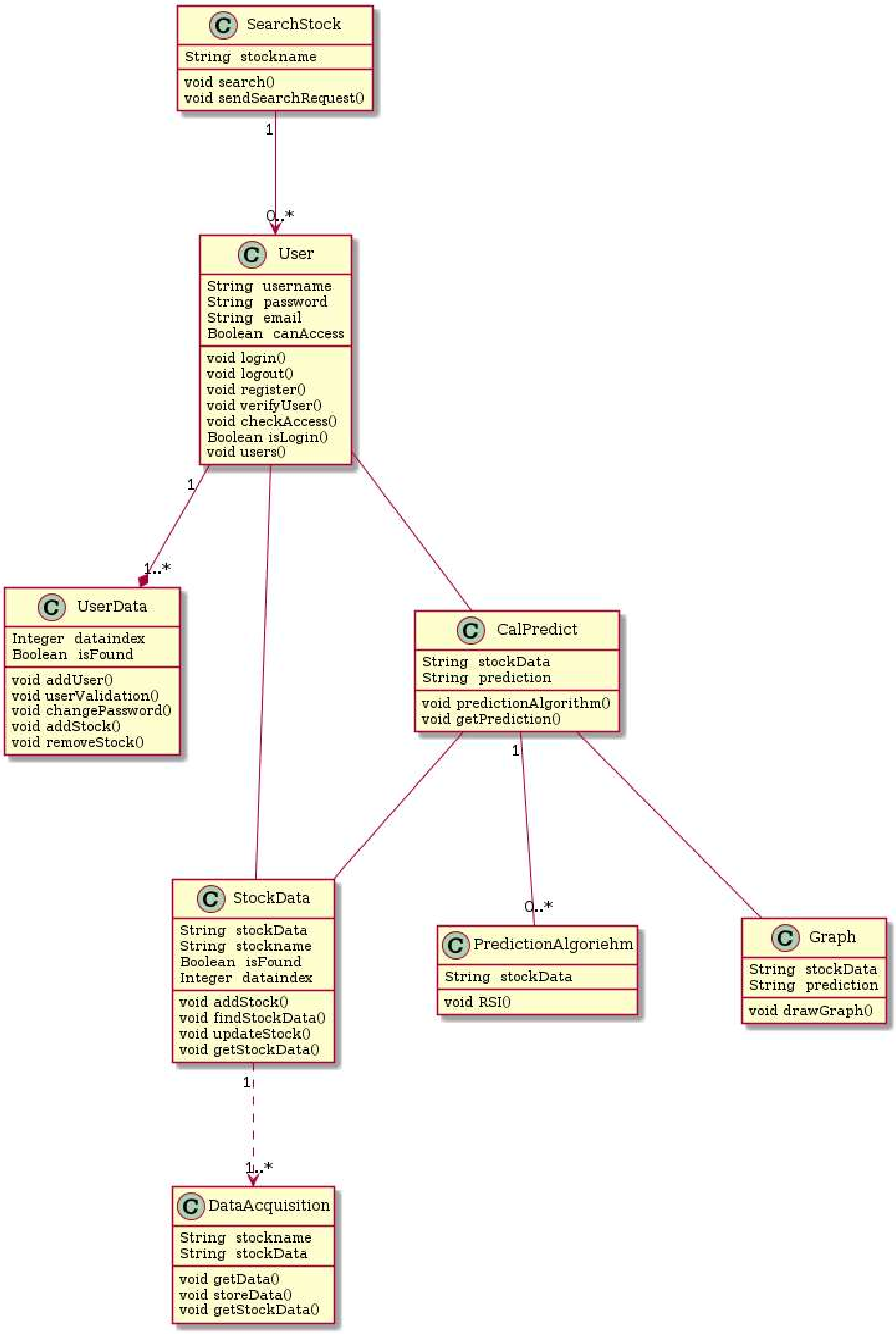


Fig.8. Class Diagram

2.6 Database Design and Normalization:

2.6.1 Table Name: Stock_Market
Primary Key: Stock_Symbol
Description: To store data of stocks

Sr. No.	Name	Datatype	Constraint	Description
1	Stock_Symbol	Varchar	Primary key	To store stock symbol
2	Date	Date	Not null	To store date
3	Series	Varchar	Not null	To store series
4	Prev_Close	Float	Not null	To store previous closed price of stock
5	Open	Float	Not null	To store open price of stock
6	High	Float	Not null	To store Highest price of stock
7	Low	Float	Not null	To store Lowest price
8	Last	Float	Not null	To store Last price
9	Close	Float	Not null	To store close price
10	Volume	Float	Not null	To store Volume of trades
11	Trades	Int	Not null	To store trades
12	Deliverable_Volume	Int	Not null	To store Deliverable Volume
13	%Deliverable	Float	Not null	To store percentage of Deliverable

Table 1. Details of stock

2.6.2 Table Name: Admin
Primary Key: Username
Foreign Key: Stock_symbol
Description: To store the admin Details

Sr. No.	Name	Datatype	Constraint	Description
1	Username	Varchar	Primary key	To store the user name
2	Stock_symbol	Varchar	Foreign Key	To store stock symbol
3	Password	Varchar	Not null	To store password

Table 2. Admin Details

2.6.3 Table Name: Prediction_model
Primary Key: Stock_Symbol
Description: To store predictions of stocks

Sr. No.	Name	Datatype	Constraint	Description
1	Stock_Symbol	Varchar	Primary key	To store symbol of stock
2	Date	Date	Not null	To store date
3	Active_value	Float	Not null	To store active value
4	Predicted_value	Float	Not null	To store predicted value

Table 3. Prediction Data

2.7 Database Relation Diagram:

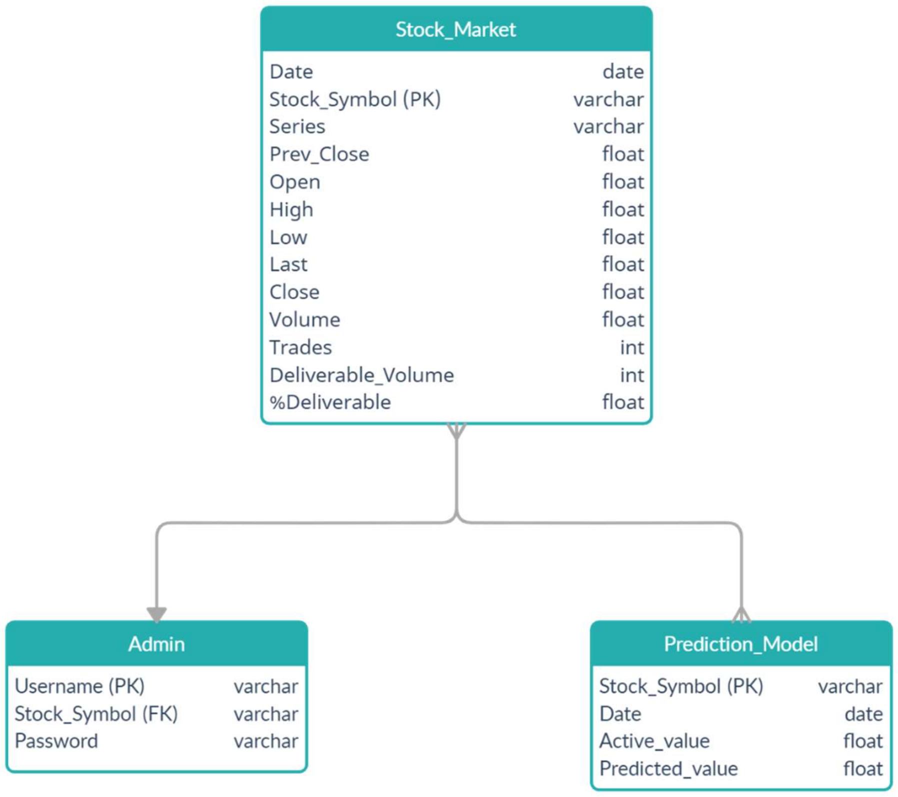
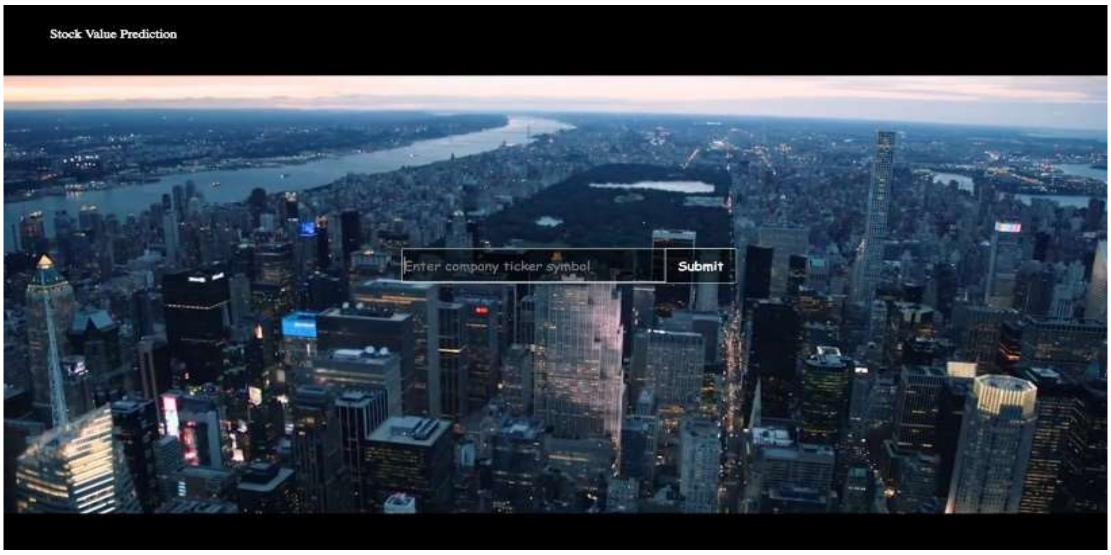
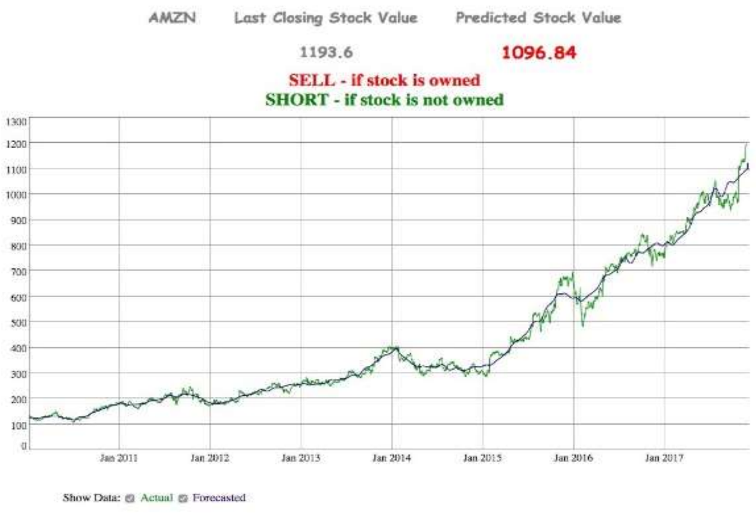


Fig.9. Database Relationship Diagram

2.8 Template Design:



Img 1. Home page



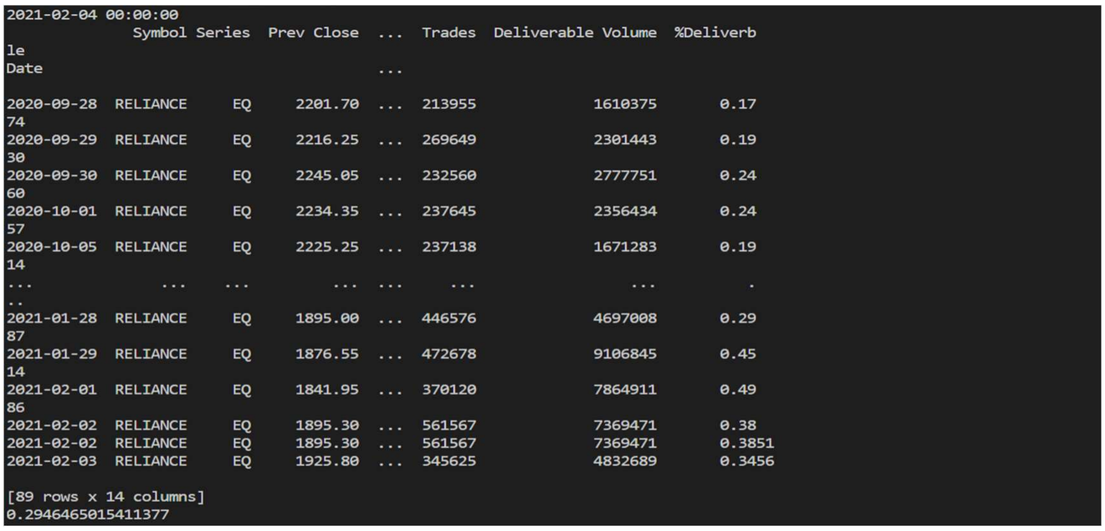
AMZN Last Closing Stock Value Predicted Stock Value

1193.6 1096.84

SELL - if stock is owned

SHORT - if stock is not owned

Img 2. Prediction result in graph form



```
2021-02-04 00:00:00
              Symbol Series  Prev Close  ...  Trades  Deliverable Volume  %Deliverb
le
Date                                     ...

2020-09-28  RELIANCE     EQ     2201.70  ...  213955             1610375       0.17
74
2020-09-29  RELIANCE     EQ     2216.25  ...  269649             2301443       0.19
30
2020-09-30  RELIANCE     EQ     2245.05  ...  232560             2777751       0.24
60
2020-10-01  RELIANCE     EQ     2234.35  ...  237645             2356434       0.24
57
2020-10-05  RELIANCE     EQ     2225.25  ...  237138             1671283       0.19
14
...              ...    ...         ...  ...     ...                 ...          .
..
2021-01-28  RELIANCE     EQ     1895.00  ...  446576             4697008       0.29
87
2021-01-29  RELIANCE     EQ     1876.55  ...  472678             9106845       0.45
14
2021-02-01  RELIANCE     EQ     1841.95  ...  370120             7864911       0.49
86
2021-02-02  RELIANCE     EQ     1895.30  ...  561567             7369471       0.38
2021-02-02  RELIANCE     EQ     1895.30  ...  561567             7369471       0.3851
2021-02-03  RELIANCE     EQ     1925.80  ...  345625             4832689       0.3456

[89 rows x 14 columns]
0.2946465015411377
```

img 3. Prediction in tabular form

Chapter 3: Implementation and Testing

3.1 Software and Tools

3.1.1 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

3.1.2 Python IDE:

An integrated development environment (IDE) refers to a software application that offers computer programmers with extensive software development abilities. IDEs most often consist of a source code editor, build automation tools, and a debugger. Most modern IDEs have intelligent code completion. In this article, you will discover the best Python IDEs currently available and present in the market.

We have used following two ide for this project:

1) **<u>PyCharm:</u>**

- PyCharm is a widely used Python IDE created by JetBrains
- This IDE is suitable for professional developers and facilitates the development of large Python projects
- Price: Freemium
- The most notable features of PyCharm include:
  - Support for JavaScript, CSS, and TypeScript
  - Smart code navigation
  - Quick and safe code refactoring
  - Support features like accessing databases directly from the IDE

2) **<u>Visual Studio Code:</u>**

- Visual Studio Code is an open-source (and free) IDE created by Microsoft. It finds great use for Python development
- VS Code is lightweight and comes with powerful features that only some of the paid IDEs offer
- Price: Free
- The most notable features of Visual Studio Code include:
  - One of the best smart code completion is based on various factors

- Git integration
- Code debugging within the editor

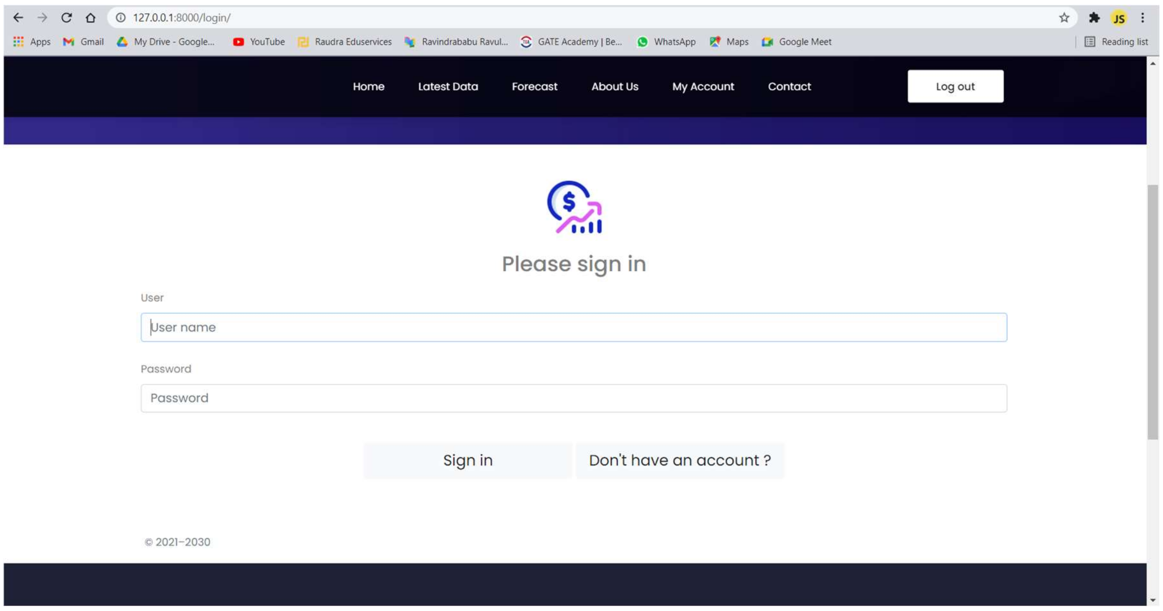
3.1.3 Google Chrome:

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, and was later ported to Linux, macOS, iOS, and Android where it is the default browser built into the OS. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.We used chrome browser to test our web application.

3.2 User Interface and Snapshot

3.2.1 Login :

- login forms are one of the most important elements that a web page contains and hence designing these online forms is one of the most significant features when it comes to designing the website.
- When user didn't login then login page restrict their activity and don't give permission to use other tabs.



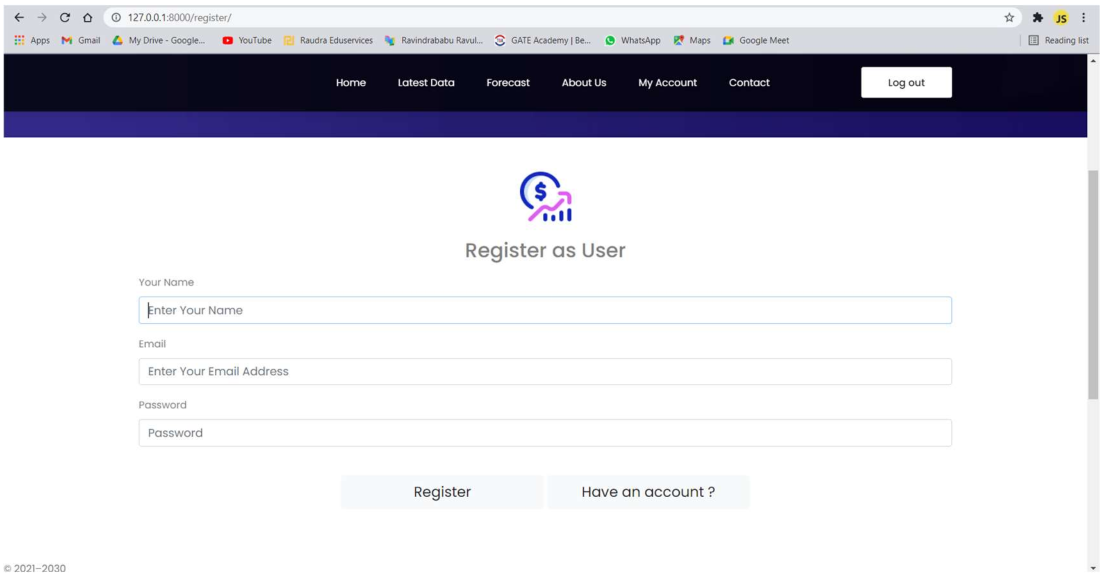
Img 4. Login Page

- **Main logics blocks :**

```
def login2(request):
    print(request)
    if request.method == "POST":
        login_user = request.POST.get('login_username')
        login_password = request.POST.get('login_password')
        user = authenticate(username=login_user, password=login_password)
        if user is not None:
            print("login")
            login(request, user)
            return redirect('/index')
        else:
            messages.success(request, 'Invalid Username or Password')
            return render(request, 'login.html')
    return render(request, 'login.html')
```

3.2.2 Sign Up/Register :

- If user don't have an account then they must have to create an account with valid name, email id and password.
- When user successfully created an account, they redirect to login page.



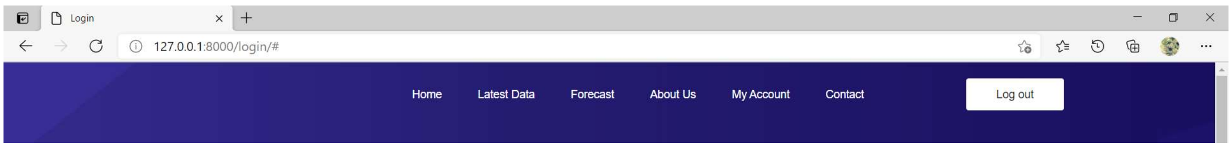
Img 5. Sign up page

- **main logics blocks:**

```
def register(request):
    if request.user.is_authenticated:
        return redirect('/')
    else:
        pass
    if request.method == "POST":
        reg_user = request.POST.get('reg_username')
        reg_email = request.POST.get('reg_email')
        reg_password = request.POST.get('reg_password')
        user = User.objects.create_user(reg_user, reg_email, reg_password)
        user.save()
        return redirect('/login')
    return render(request, 'register.html')
```

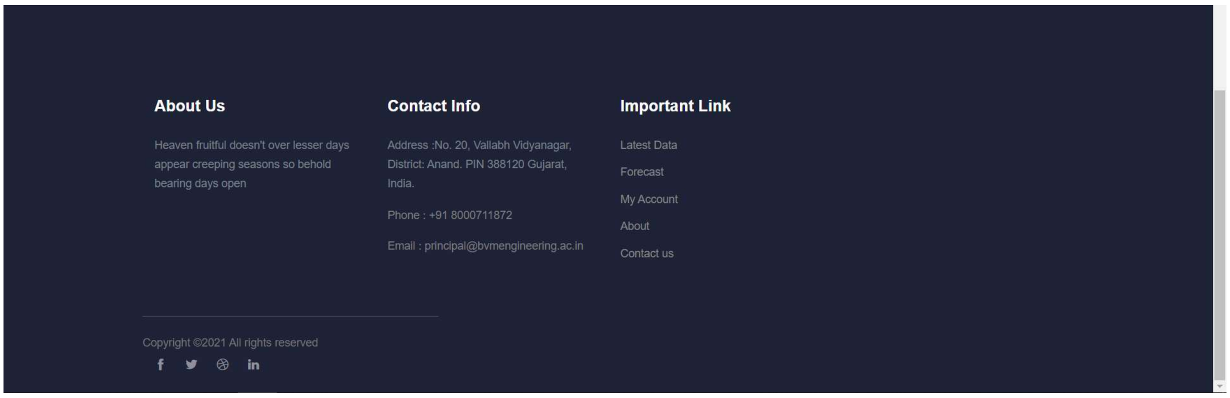
3.2.3 User interface:

- **Top of the pages:**
  At the top of the page use can get all the tabs so user can easily move from one page to another and also login or logout button.



Img 6. Top navigation bar

- **Bottom of the pages:**
  At the bottom of the page user can get contact details and also all the page link which are in our system.



About Us

Heaven fruitful doesn't over lesser days appear creeping seasons so behold bearing days open

Contact Info

Address :No. 20, Vallabh Vidyanagar, District: Anand. PIN 388120 Gujarat, India.

Phone : +91 8000711872

Email : principal@bvmengineering.ac.in

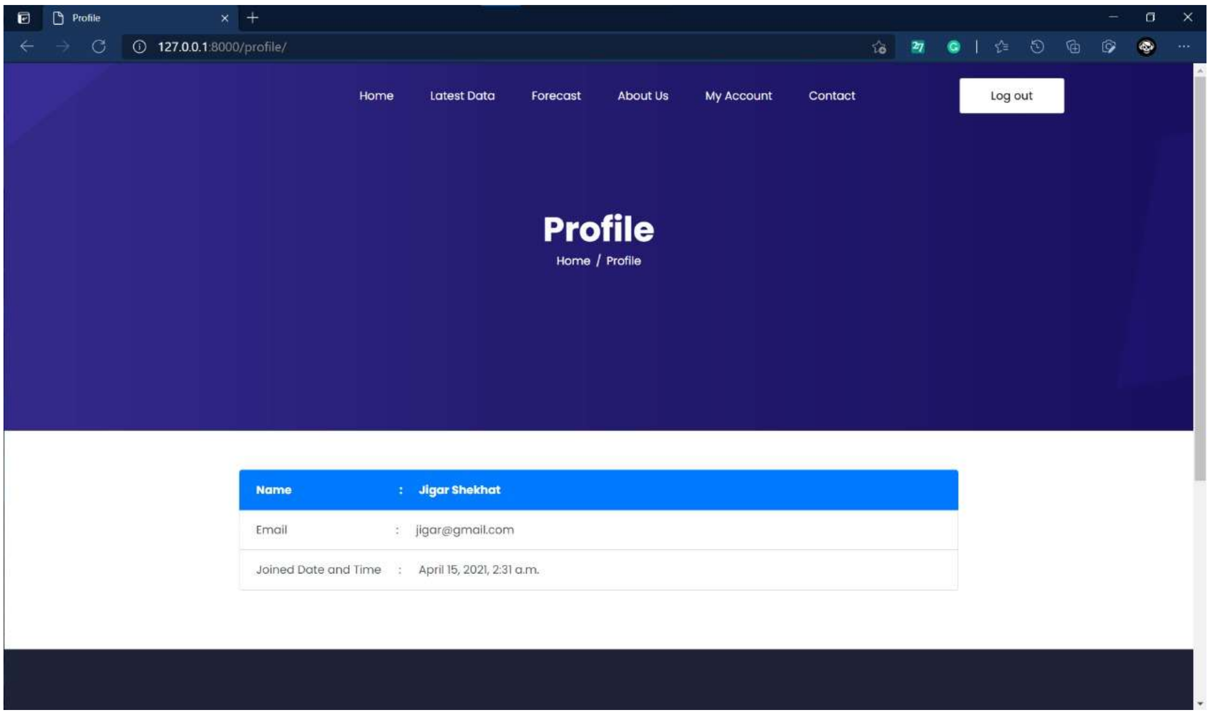
Important Link

Latest Data
Forecast
My Account
About
Contact us

Copyright ©2021 All rights reserved

Img 7. Bottom footer

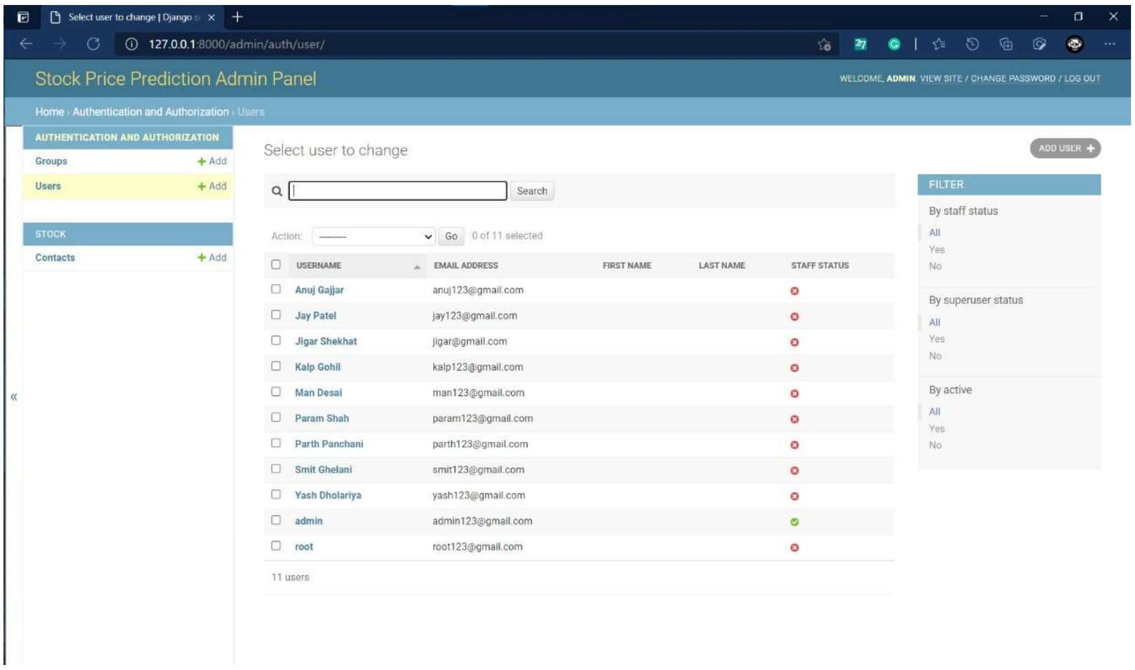
- **User profile:**
  User can maintain their profile through user profile section.



Name	:	Jigar Shekhat
Email	:	jigar@gmail.com
Joined Date and Time	:	April 15, 2021, 2:31 a.m.

Img 8. User profile

- **Admin profile:**
  Through admin section we can main user activity by adding removing and checking user activities. This admin panel is default panel provided by Django.


Select user to change | Django
127.0.0.1:8000/admin/auth/user/
Stock Price Prediction Admin Panel
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Authentication and Authorization › Users
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
STOCK
Contacts + Add
Select user to change
ADD USER +
Search
Action: --------- Go 0 of 11 selected
FILTER
By staff status
All
Yes
No
By superuser status
All
Yes
No
By active
All
Yes
No

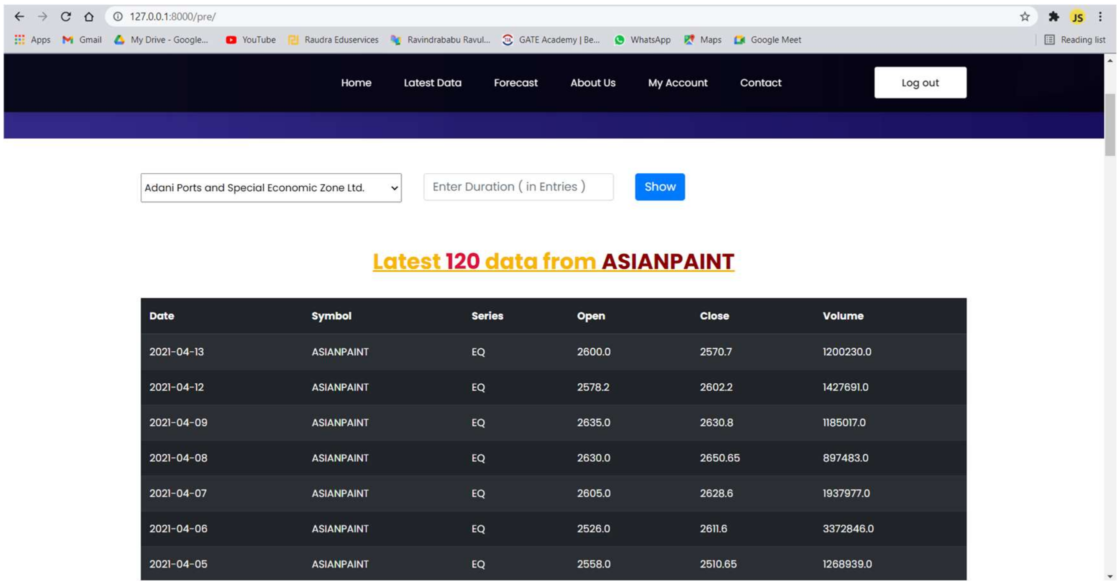
USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
Anuj Gajjar	anuj123@gmail.com			✗
Jay Patel	jay123@gmail.com			✗
Jigar Shekhat	jigar@gmail.com			✗
Kalp Gohil	kalp123@gmail.com			✗
Man Desai	man123@gmail.com			✗
Param Shah	param123@gmail.com			✗
Parth Panchani	parth123@gmail.com			✗
Smit Ghelani	smit123@gmail.com			✗
Yash Dholariya	yash123@gmail.com			✗
admin	admin123@gmail.com			✓
root	root123@gmail.com			✗

11 users

Img 9. Admin Panel

3.2.4 Stock search :

- Whenever User choose a particular stock-name then this module gives the latest data about Open, Close and Volume of stocks.
- User have to enter number of entries according to their requirements.



Latest 120 data from ASIANPAINT

Date	Symbol	Series	Open	Close	Volume
2021-04-13	ASIANPAINT	EQ	2600.0	2570.7	1200230.0
2021-04-12	ASIANPAINT	EQ	2578.2	2602.2	1427691.0
2021-04-09	ASIANPAINT	EQ	2635.0	2630.8	1185017.0
2021-04-08	ASIANPAINT	EQ	2630.0	2650.65	897483.0
2021-04-07	ASIANPAINT	EQ	2605.0	2628.6	1937977.0
2021-04-06	ASIANPAINT	EQ	2526.0	2611.6	3372846.0
2021-04-05	ASIANPAINT	EQ	2558.0	2510.65	1268939.0

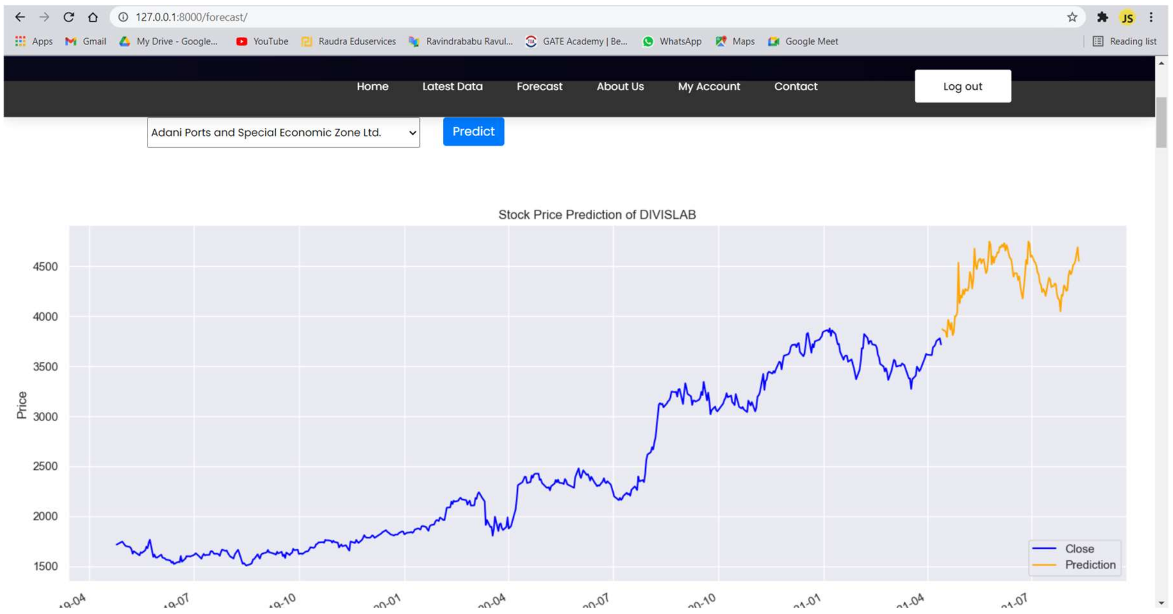
Img 10. Stock Search

- **main logics blocks :**

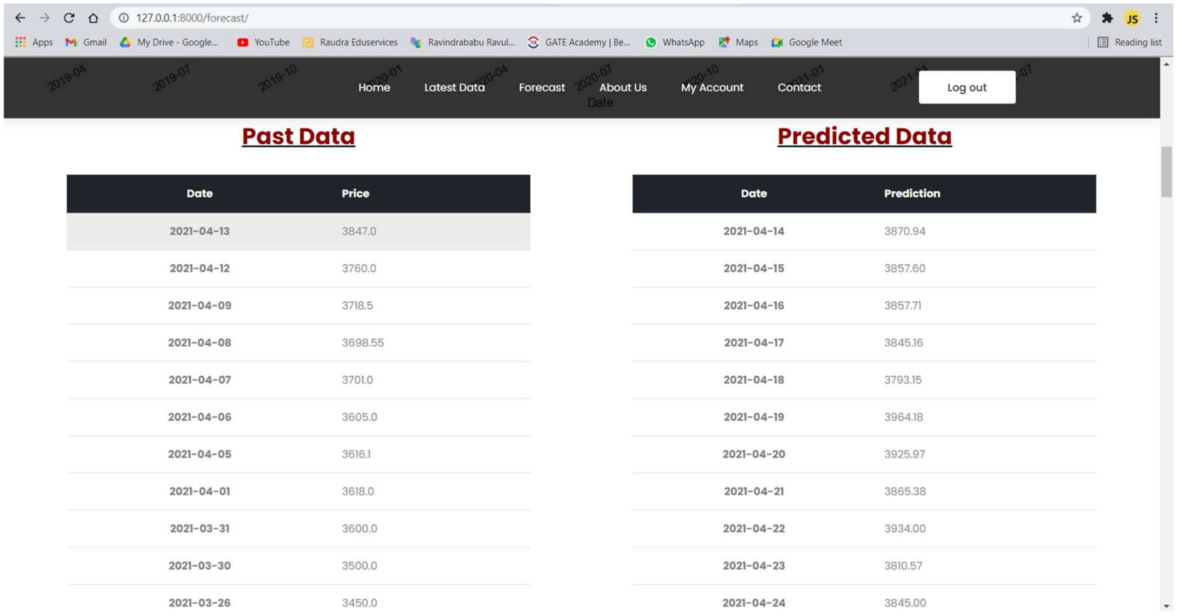
```
def pre(request):
    if request.user.is_authenticated:
        pass
    else:
        return redirect('/login')
    if request.method == "POST":
        symbol_name = request.POST.get('symbol_name')
        duration = int(request.POST.get('dura'))
        print("symbol_name")
        print(duration)

        today = date.today()
        start = today + timedelta(-duration)
        stockData = nsepy.get_history(symbol=symbol_name, start=start,
end=today)
        stockData = stockData.reset_index()

        stockData['Date'] = stockData['Date'].astype(str)
        stockData['Open'] = stockData['Open'].astype(float)
        stockData['Volume'] = stockData['Volume'].astype(float)
        #chart = get_plot(stockData['Open'].tolist(), stockData['Volume'].tolist())
        stockData.plot()
        chart = get_graph()

        stockData = stockData.reindex(index=stockData.index[::-1])
        json_records = stockData.reset_index().to_json(orient='records')

        data = []
        data = json.loads(json_records)
        context = {'d': data,
                   'duration' : duration,
                   'sym_name':symbol_name,
                   'visible' : "visible",
                   'chart': chart}
        return render(request, 'pre.html', context)

    context = {'visible': "invisible"}
    return render(request, 'pre.html',context)
```

3.2.5 Price Prediction & Visualization:

- User have to select a particular stock-name then this module gives prediction values of 120 days and gives the graphical representation of the data with table of past and predicted data.



Img 11. Prediction Algorithm Output



Past Data

Date	Price
2021-04-13	3847.0
2021-04-12	3760.0
2021-04-09	3718.5
2021-04-08	3698.55
2021-04-07	3701.0
2021-04-06	3605.0
2021-04-05	3616.1
2021-04-01	3618.0
2021-03-31	3600.0
2021-03-30	3500.0
2021-03-26	3450.0

Predicted Data

Date	Prediction
2021-04-14	3870.94
2021-04-15	3857.60
2021-04-16	3857.71
2021-04-17	3845.16
2021-04-18	3793.15
2021-04-19	3964.18
2021-04-20	3925.97
2021-04-21	3865.38
2021-04-22	3934.00
2021-04-23	3810.57
2021-04-24	3845.00

Img 12. Tabular data

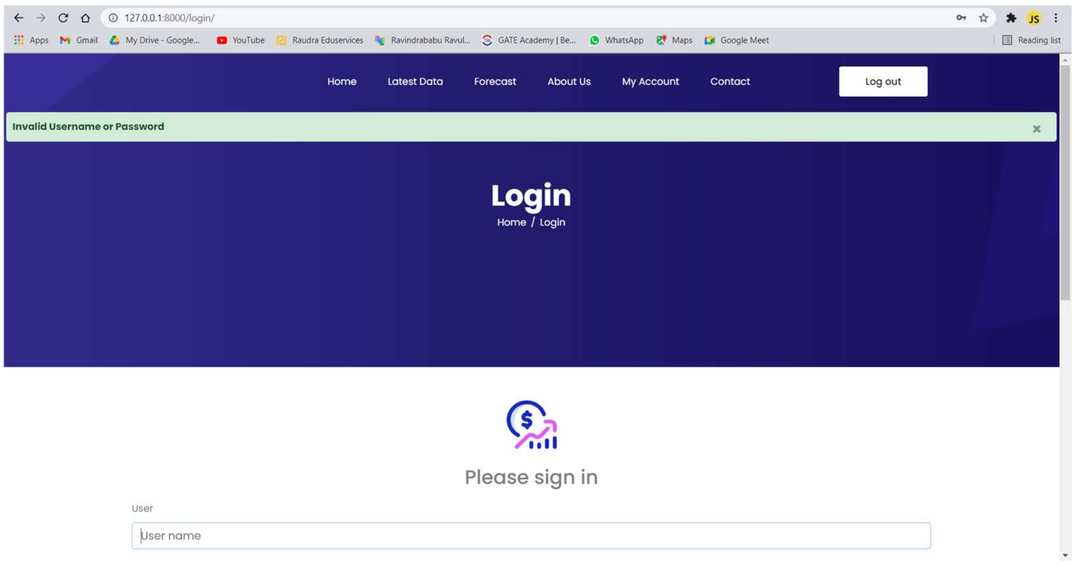
- **main logics blocks :**

```
import numpy as np

class LinearRegression:

    def __init__(self, learning_rate=0.01, n_iters=1000):
        self.lr = learning_rate
        self.n_iters = n_iters
        self.slope = None
        self.c = None

    def train(self, x, y):
        n_samples, n_features = x.shape
        self.slope = np.zeros(n_features)
        self.c = 0

        for _ in range(self.n_iters):
            y_predicted = np.dot(x, self.slope) + self.c
            dw = (1 / n_samples) * np.dot(x.T, (y_predicted - y))
            db = (1 / n_samples) * np.sum(y_predicted - y)
            self.slope -= self.lr * dw
            self.c -= self.lr * db

    def predict(self, x):
        y_approx = np.dot(x, self.slope) + self.c
        return y_approx
```

3.3 Testing using Use Cases

3.3.1 Login :

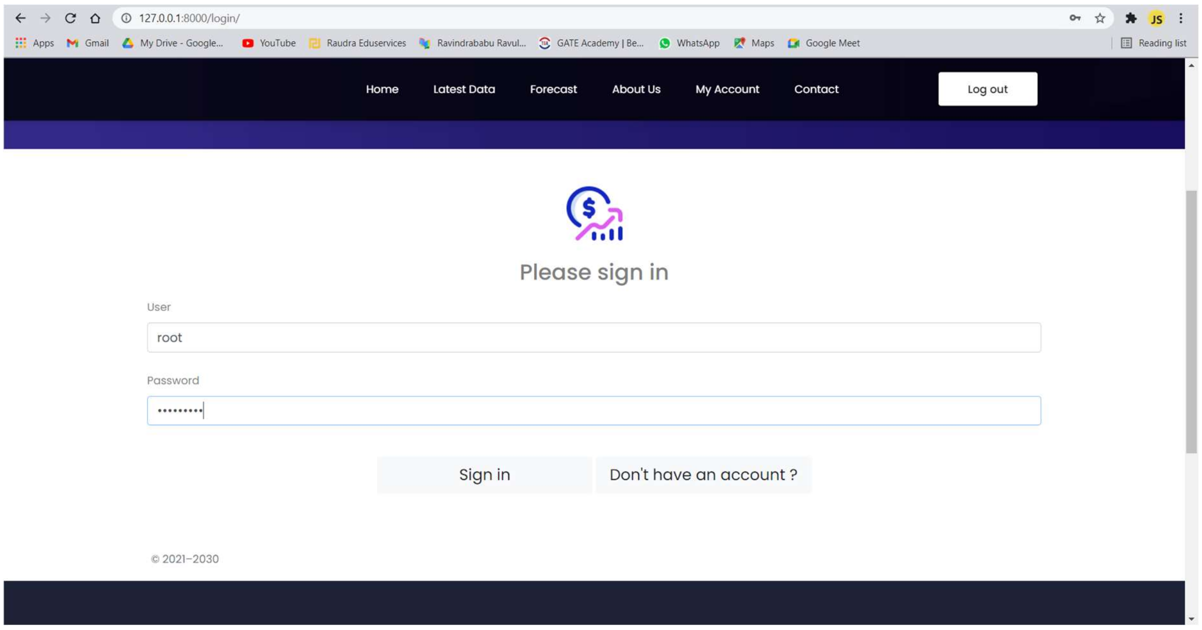
Case – 1 : When we enter invalid username or password, it denies the access.



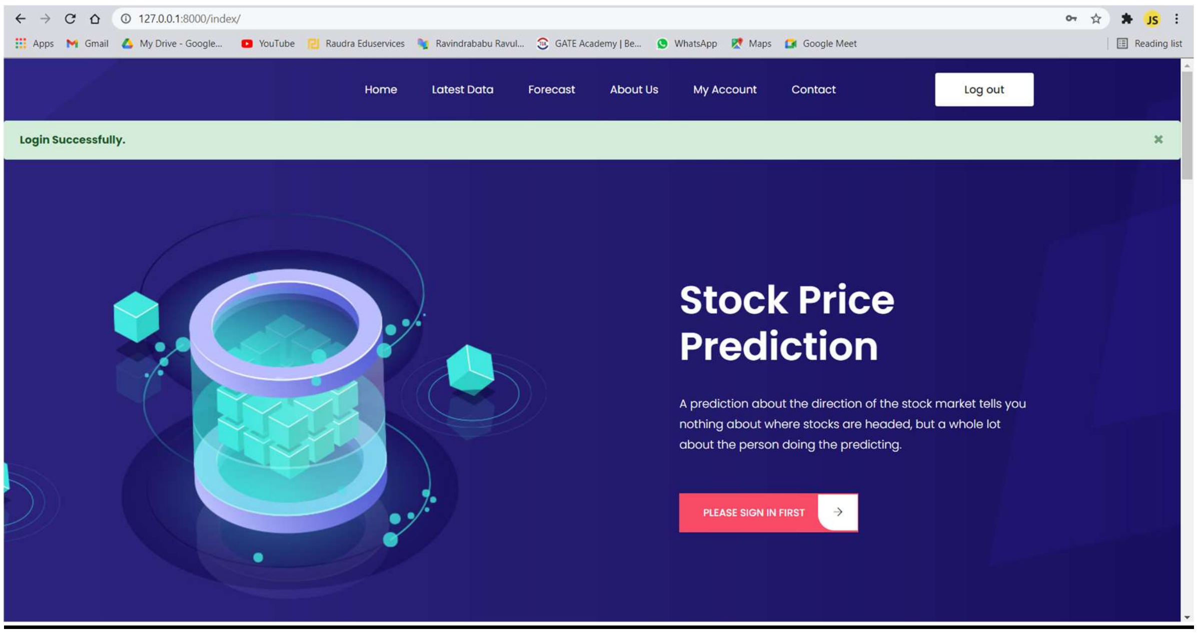
Invalid Username or Password

Img 13. Invalid Username and Password

Case – 2 : When we enter valid username and password, it allow access of homepage.



Img 14. Valid Input

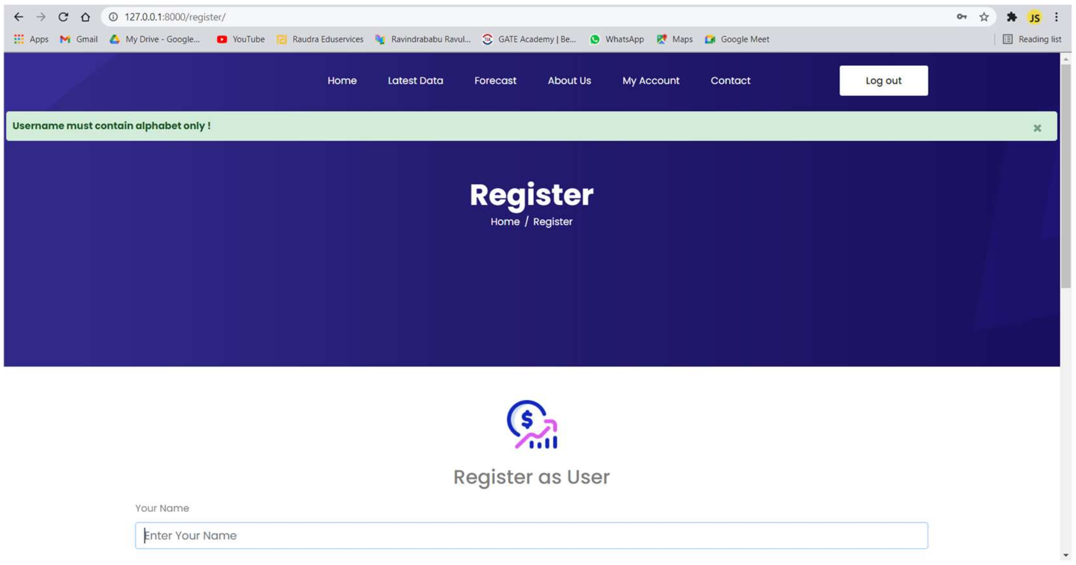


A prediction about the direction of the stock market tells you nothing about where stocks are headed, but a whole lot about the person doing the predicting.

Img 15. Successful login

3.3.2 Sign Up / Register :

Case – 1 : When we enter non-alphabetic in username then it will denied and throws alert message



Username must contain alphabet only !

Register

Home / Register

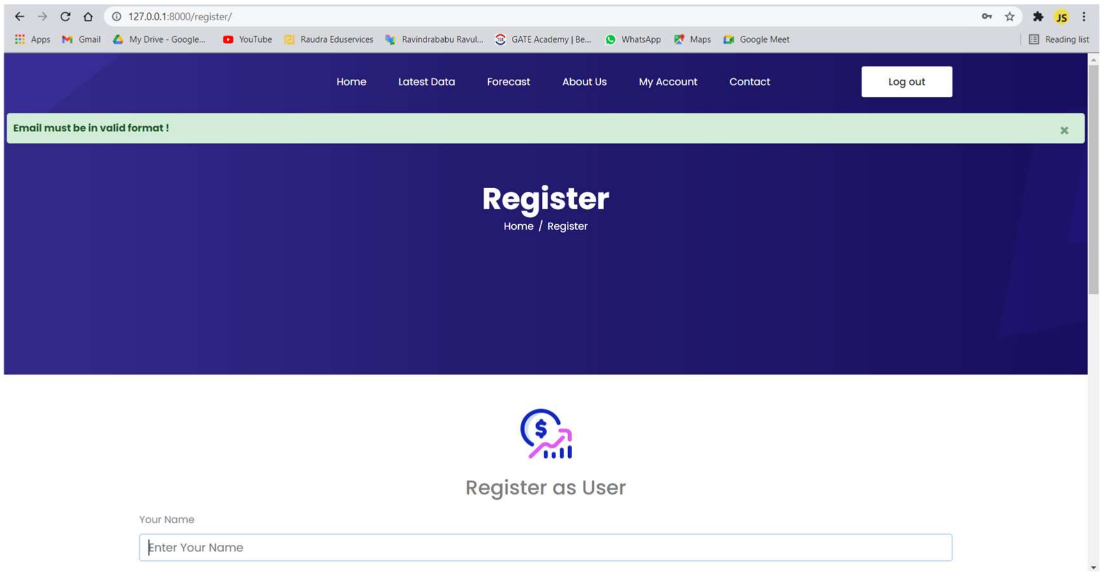
Register as User

Your Name

Enter Your Name

Img 16. Invalid Username input

Case – 2 : When we enter invalid email format then it will denied and throws alert message.



Email must be in valid format !

Register

Home / Register

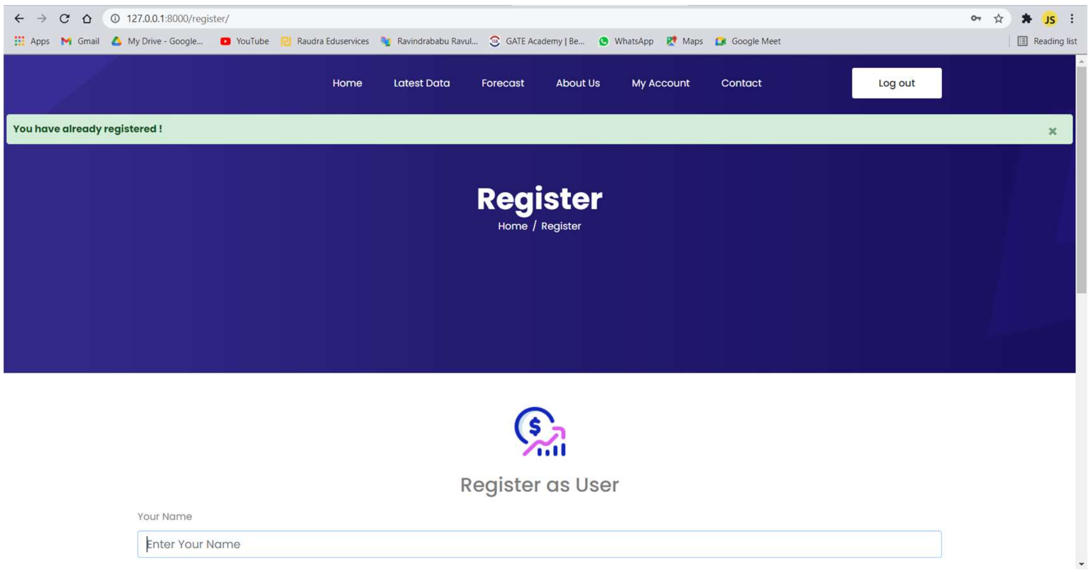
Register as User

Your Name

Enter Your Name

Img 17. Invalid Email Format

Case – 3 : When we have already registered, it throws Exception.

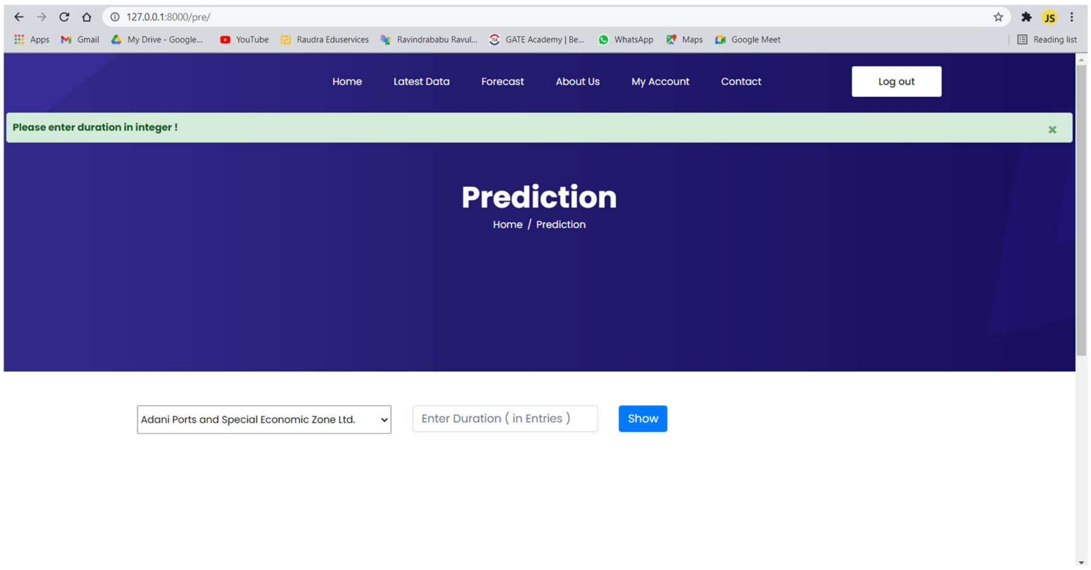


You have already registered !

Img 18. Registered User

3.3.3 Stock-Search :

Case – 1 : When we enter alphabet or decimal value in duration field then it shows alert message.



Please enter duration in integer !

Img 19. Invalid Duration

3.3.4 Price prediction :

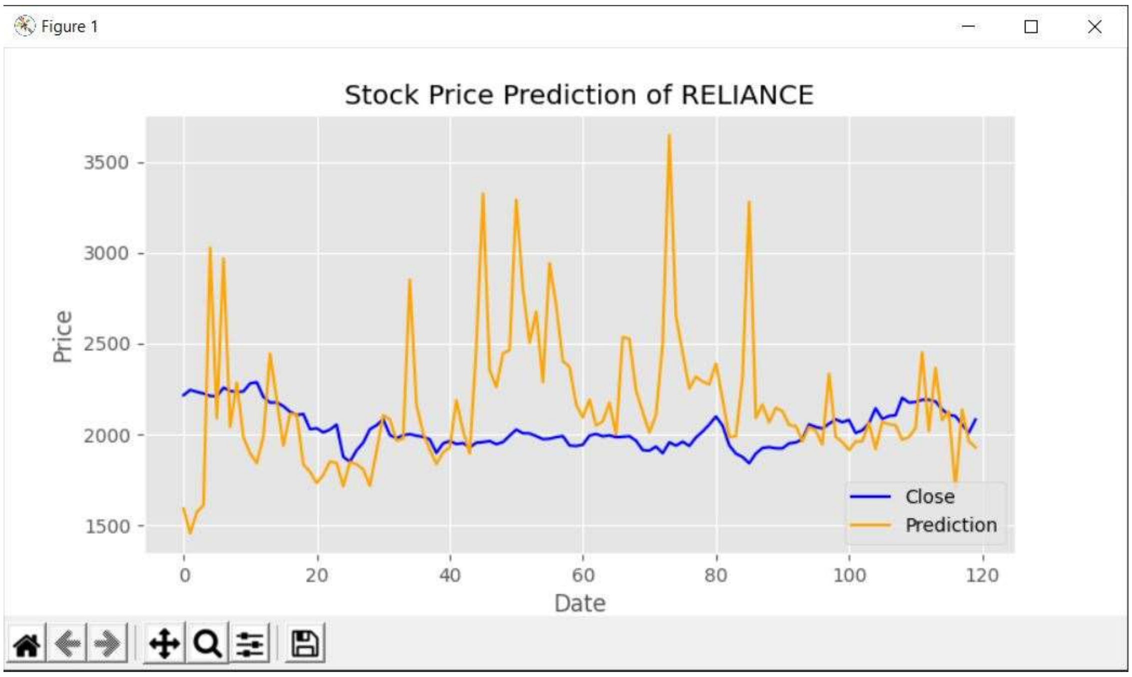
Case - 1: In this case we compared the past 120 days of RELIANCE stock close prices with the prediction algorithm outputs. It is not pretty accurate but we can say that it gives idea about stock price increment or decrement**.**

Comparison table:

Date	Symbol	Close	Prediction
28-09-2020	RELIANCE	2216.25	1591.792
29-09-2020	RELIANCE	2245.05	1457.004
30-09-2020	RELIANCE	2234.35	1573.177
01-10-2020	RELIANCE	2225.25	1611.334
05-10-2020	RELIANCE	2212.2	3026.628
06-10-2020	RELIANCE	2210.35	2088.228
07-10-2020	RELIANCE	2257.5	2968.066
08-10-2020	RELIANCE	2239.25	2042.042
09-10-2020	RELIANCE	2233.45	2284.92
12-10-2020	RELIANCE	2237.05	1984.282
…	…	…	…
05-03-2021	RELIANCE	2178.7	2034.587
08-03-2021	RELIANCE	2191.1	2451.756
09-03-2021	RELIANCE	2191.05	2019.868
10-03-2021	RELIANCE	2181.95	2365.84
12-03-2021	RELIANCE	2137.6	2079.788
15-03-2021	RELIANCE	2108.9	2126.025
16-03-2021	RELIANCE	2100.6	1706.33
17-03-2021	RELIANCE	2055.35	2136.639
18-03-2021	RELIANCE	2009.1	1963.894
19-03-2021	RELIANCE	2082	1929.669

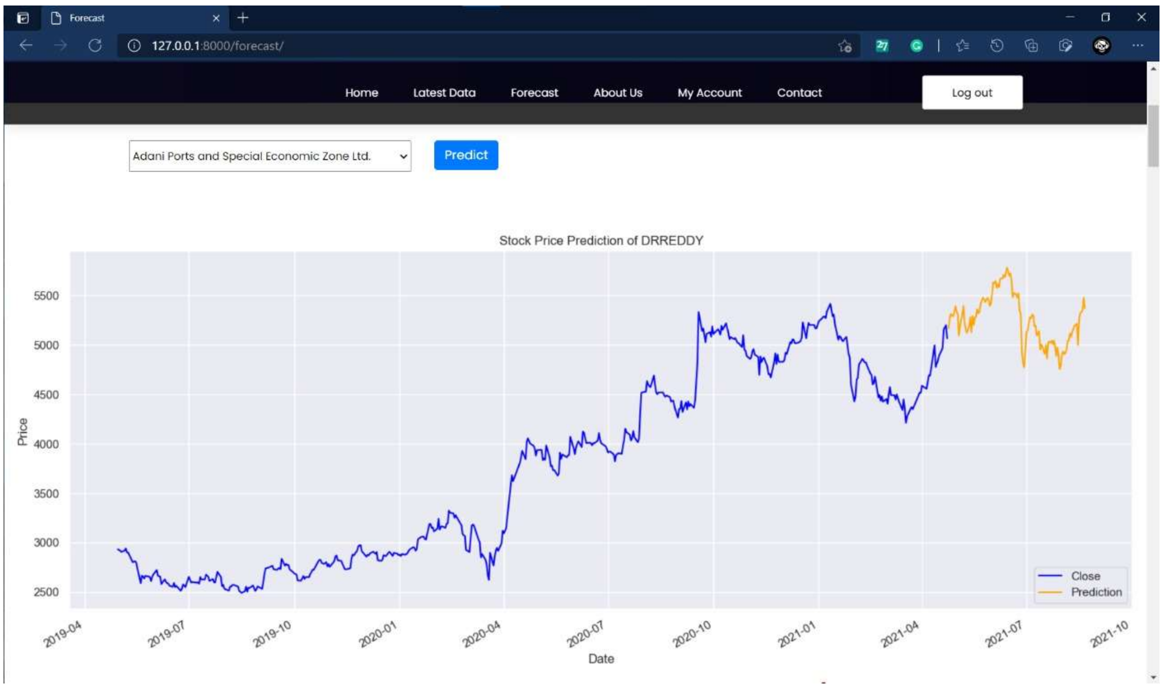
Table 4 - Closed price and Predicted price comparison

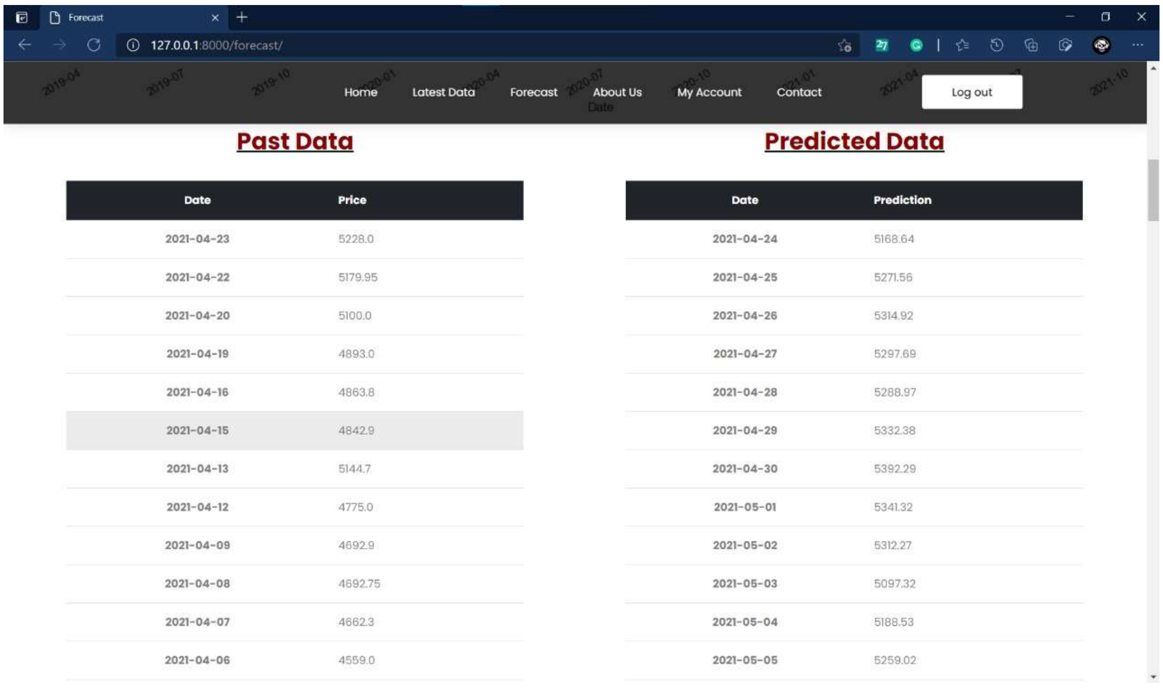
Comparison Graph:



Img 20. Comparison Graph

Case – 2: Actual working of module in web application by selection stock name from drop down menu. For example: we have selected here Dr. Reddy's Laboratories Ltd.





Past Data

Date	Price
2021-04-23	5228.0
2021-04-22	5179.95
2021-04-20	5100.0
2021-04-19	4893.0
2021-04-16	4863.8
2021-04-15	4842.9
2021-04-13	5144.7
2021-04-12	4775.0
2021-04-09	4692.9
2021-04-08	4692.75
2021-04-07	4662.3
2021-04-06	4559.0

Predicted Data

Date	Prediction
2021-04-24	5168.64
2021-04-25	5271.56
2021-04-26	5314.92
2021-04-27	5297.69
2021-04-28	5288.97
2021-04-29	5332.38
2021-04-30	5392.29
2021-05-01	5341.32
2021-05-02	5312.27
2021-05-03	5097.32
2021-05-04	5188.53
2021-05-05	5259.02

Chapter 4: Conclusion and Future work

4.1 Conclusion:

It is fully working model and tested by us. This project qualified to predict stock prices and this model is sufficient to guide new investors and also the existing once. It is sufficiently predict market behavior for short to large period of time.

Before the project starting we have considered that stock prices are some where depends upon past data of stock. And at the end of this project we conclude that it not a accurate model through which we can find stock price accurately and some what possible to predict stock prices more accurately.

So we can conclude that we need to make some changes in this model to make this model more and more accurate.

4.2 Future work:

In this model lots of work is require to establish fully perfect stock price prediction but it is not possible to get 100% accurate stock prices but we can make more accurate by changing machine learning algorithm which more accurately predict data by preprocessing the acquired data and testing algorithm behavior.

also by adding some factors which affect the stock price changes i.e. market behavior, company's market relation and market value, future deals, current revenue growth, latest new about stock, rumors and many more.

References:

- Linear Regression For Beginners with Implementation in Python
  https://www.analyticsvidhya.com/blog/2020/10/linear-regression-for-absolute-beginners-with-implementation-in-python/

- (Batch) Gradient descent algorithm
  https://www.bogotobogo.com/python/python_numpy_batch_gradient_descent_algorithm.php

- Existing applications with review
  https://www.liberatedstocktrader.com/top-10-best-stock-market-analysis-software-review/