**A Project Report On**

**Stock Candle**

Submitted to SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**BY**

**Utkarsha M Abhang - B150318501**

**Piyush S Adhalikar - B150318503**

**Atharva M Mulay – B150318600**

**Anushka M Mulay – B150318599**

**UNDER THE GUIDANCE OF**

**PROF . YOGITA FATANGARE**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**PES’S MODERN COLLEGE OF ENGINEERING,**

**SHIVAJINAGAR, PUNE 05**

**2021-22**

**CERTIFICATE**

This is to clarify that the project report entitled

STOCK CANDLE

Submitted by

**Utkarsha M Abhang - B150318501**

**Piyush S Adhalikar - B150318503**

**Atharva M Mulay – B150318600**

**Anushka M Mulay – B150318599**

is a bonafide work has been carried out by them under the supervision of **Prof. Yogita Fatangare** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Information Technology).

This project report has not been earlier submitted to any other institute or university for the award of any degree or diploma.

Prof . Yogita Fatangare Prof. Sarita Deshpande

Internal Guide Head of Department

Department of Information Technology Department of Information Technology

Prof. Kalyani Joshi

External Examinar Principal

Date : PES MCOE

Place : Pune

Date :

ACKNOWLEDGEMENT

We take this to express our deep sense of gratitude towards our esteemed guide Prof. Yogita Fatangare for giving us this splendid opportunity to select and present this project and also providing facilities for successful completion.

We thank Prof. S D Deshpande, Head of Department of Information Technology Engineering, for opening the doors of the department towards the realization of the project, all the staff members, for their indispensable support, priceless suggestion and for most valuable time lent as and when required. With respect and gratitude, we would like to thank all the people, who have helped us directly or indirectly.

**Utkarsha M Abhang**

**Piyush S Adhalikar**

**Atharva M Mulay**

**Anushka M Mulay**

ABSTRACT

In the era of big data, deep learning for predicting stock market prices and trends has become even more popular than before. We will be collecting past 7 years of data from API and will propose a comprehensive customization of feature engineering and deep learning-based model for predicting price trend of stock markets. The proposed solution will include pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized deep learning based system for stock market price trend prediction.

Finance is highly nonlinear and sometimes stock price data can even seem completely random. Traditional time series methods such as ARIMA, SARIMA and GARCH models are effective only when the series is stationary. This is combated by using **Neural Networks** (FBProphet), which do not require any stationarity to be used.

We are using FBProphet for predicting the identified stock future prices. This model is trained by giving data from year 2015 of a particular company stock, then it is possible to do prediction of stock prices of next 4 years, upto 2026. To enhance performance of model, different optimization techniques can be used.

The proposed solution includes pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized deep learning based system for stock market price trend prediction.

**LIST OF FIGURES**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Chapter No. | Name of Chapter | Page Number |
| 1. | Introduction to Project   * 1. Introduction to Stock Candles.   2. Introduction to Project   3. Motivation behind project topic   4. Aim and Objective(s) of the work   5. Introduction to Fbphrophet | 07 |
| 2. | Background and Literature Survey  2.1 Background  2.2 Literature Survey | 09 |
| 3. | Specification  3.1 Introduction   1. Project Scope 2. User Characteristics 3. Assumptions and dependencies   3.2 Functional Requirement  3.3 External Interface Requirement  3.3.1 User Interfaces  3.3.2 Software Interfaces  3.3.3 Hardware Interfaces  3.4 Non-functional Requirements  3.4.1 Performance Requirement  3.4.2 Safety Requirement  3.4.3 Security  3.4.4 Software quality attributes  3.5 System Requirement  3.5.1 Database Requirement  3.5.2 Software Requirement  3.5.3 Hardware Requirement  3.6 Analysis Model | 13 |
| 4. | System Design  4.1 System Architecture  4.2 UML Diagrams | 19 |
| 5. | System Implementation  5.1 Implementation plan - I  5.2 Implementation plan - II | 27 |
| 6. | Result and Implementation  6.1 User Interface  6.2 Source Code  6.3 Database | 29 |
| 7. | Conclusion and Future Work  7.1 Conclusion  7.2 Future Work | 43 |
| 8. | References | 44 |
|  |  |  |

* 1. **INTRODUCTION**
  2. **Introduction to Stock Candle**
* Stock market allows businesses to go public, or raise additional capital for expansion. This project is to determine and predict the stock market value for the specific company.
* It is one of the most important sources for companies as well as for individuals to raise money.
* A stock market is a platform for trading of a company’s stocks and derivatives at an agreed price.
  1. **Introduction to Project**
* This project will provide a way to investors, stock markets to invest money in order to potentially earn a share of the company’s profits (knowing that the risk of losses exists too).
* Active investors and traders can easily buy and sell their securities due to the abundant liquidity in most major stock markets.
* Stock markets promote investment. The raising of capital allows companies to grow their businesses, expand operations and create jobs in the economy. This investment is a key driver for economic trade, growth and prosperity.
  1. **Motivation behind project topic**

•The fluctuation of stock market is violent and there are many complicated financial indicators. However, the advancement in technology, provides an opportunity to gain steady fortune from stock market

•It can help experts to find out the most informative indicators to make better prediction.

•The prediction of the market value should help investors in maximizing the profit of stock option purchase while keeping the risk low

* 1. **Aim and Objective(s) of the work**
* Aim of this stock market prediction project is to create software that analyzes previous stock data of certain companies.
* To predict and analyze the stock price of the company with the help of historical data and present data using fbprophet.

Objective of this project is

* To the study of past stock prices to predict the trend of prices in future.
* To help investors in the direction of movement of the share prices.
* To save investors from suffering a big loss in future.
* To determine the future value of a company stock or other financial instrument traded on an exchange.
  1. **Introduction to Fbprophet**

It represents the residual assets of the company. Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange.

The successful prediction of a stock's future price could yield significant profit. The result of the analysis is predicted with the help of fbprophet. Complex machine learning algorithms such as deep learning methods can analyze and detect complex data patterns.

Algorithm

* Step 1 : Import the necessary python packages
* Step 2 : Using ticker symbol, past 7 years of stock values are downloaded in a variable
* Step 3 : certain figures such as figures with subplots of different types, dual-axis plots, or faceted plots with multiple different types of traces are created with Plotly.graph\_objects.
* Step 4 : fbprophet Function is applied on the data loaded.
* Step 5 : Using predict function of fbprophet necessary values are forecasted and graph is ploted

**2. BACKGROUND AND LITERATURE SURVEY**

**2.1 Background**

An accurate prediction of stock market index is important for investors to reduce financial risk. Although quite a number of deep learning methods have been developed for the stock prediction, some fundamental problems, such as weak generalization ability and overfitting in training, need to be solved. In this paper, a new deep learning model named Random Long Short-Term Memory (RLSTM) is proposed to get a better predicting result. RLSTM includes prediction module, prevention module, and three full connection layers. Input of the prediction module is a stock or an index which needs to be predicted.

Zeroda and upstocks are the used softwares.

**2.2 Literature survey**

The study, “Stock Chart Pattern recognition with Deep Learning” [1] evaluated the performances of CNN and LSTM for recognizing common charts patterns in a stock historical data. It presents two common patterns, the method used to build the training set, the neural networks architectures and the accuracies obtained.

In paper [2], A “window” of various durations is run, breaking it into frames that scale in duration to a single size and to 1 in amplitude. The received frames are converted into 2D matrices and fed for analysis to a 2D convolutional NN, which determines the probability of frames belonging to the classes of patterns. The accuracy of the detector is about 98.6 % with a convolutional NN response speed of about 0.65 seconds per 1000 data samples, which corresponds to an analysis of the closing prices of trades on the exchange for more than 2.5 years.

A lot of studies provide strong evidence that traditional predictive regression models face significant challenges in out-of sample predictability tests due to model uncertainty and parameter instability. Recent studies introduce particular strategies that overcome these problems. Support Vector Machine (SVM) is a relatively new learning algorithm that has the desirable characteristics of the control of the decision function, the use of the kernel method, and the sparsely of the solution. In this paper, we present a theoretical and empirical framework to apply the Support Vector Machines strategy to predict the stock market. Firstly, four company-specific and six macroeconomic factors that may influence the stock trend are selected for further stock multivariate analysis. Secondly, Support Vector Machine is used in analyzing the relationship of these factors and predicting the stock performance. Our results suggest that SVM is a powerful predictive tool for stock predictions in the financial market. [3]

The prediction of the stock market can generate an actual financial loss or gain, so it is practically important to enhance the predictability of models. Consequently, many studies have been trying to model and predict financial time series, using statistical or soft computational skills that are capable of examining the complex and chaotic financial market. In recent years, deep learning techniques have been actively applied based on their excellent achievements in various classification problems. In this study, we constructed a stock price prediction model based on RNN using LSTM units, which is one of the typical methodologies of deep learning. We integrated GA and LSTM network to consider the temporal properties of the stock market, and utilized the customized architectural factors of a model. The LSTM network used in this study is composed with two hidden layers, which is a deep architecture for expressing nonlinear and complex features of the stock market more effectively. GA was employed to search the optimal or near-optimal value for the size of the time window and number of LSTM units in an LSTM network. [4]

Sentiment analysis has proven to be effective while analyzing people's attitudes by examining large social data. In this, a novel approach is designed to extract people's opinions on specific topics by relying on social media content. 70% of dataset is used for training whereas remaining dataset is used during testing. It has been determined that the proposed classifier, classifies the positive, negative and neutral sentiments with an accuracy of 98.32% From the experiment, it has been analyzed that the detection accuracy of sentiments has been increased by 8.99 % from the existing work. The main advantage of this work is that a stop word panel is added in to the GUI, so that a user can add or remove the stop words as per the need. [5]

A system is called offline when system takes images as inputs and tries to predict characters from given images by applying machine learning methods. We have worked on offline character recognition problem in this project. Many machine learning methods have been proposed over the years for solving this problem. In this paper we implemented 6 most popular machine learning methods to solve offline handwriting character recognition problem and compare the performance results to decide which method gives best accuracy results under pre-defined conditions. We have selected 92255 images from NIST Special 19 Database and used them as input images during the training phase of the selected machine learning methods. These methods are SVM, Decision Tree, Bag of Trees, Artificial Neural Networks (ANN), Deep learning network with autoencoders and Convolutional Neural Networks (CNN). We implemented all of these methods and compare the performance of the results according to accuracy metric. The results obtained from the comparison is going to help in deciding which ML method should be used to solve Offline Handwriting Character Recognition problem. [6]

It is seen that, regression architectures are capable of capturing dynamics and are able to make predictions. We trained the model using the data of stock and was able to predict stock price of stock. This shows that, the proposed system is capable of identifying some inter relation with in the data. Also, it is evident from the results that, SVR, RFR & DTR model is capable of identifying the changes in trends. For the proposed methodology DTR is identified as the best model. It uses the information given at a particular instant for prediction. Even though the other two models are used in many other time dependent data analysis, it is not out performing the DTR model in this case. This is due to the sudden changes that occurs in stock markets. The changes occuring in the stock market may not always be in a regular pattern or may not always follow the same cycle. Based on the companies and the sectors, the existence of the trends and the period of their existence will differ. The analysis of these type of trends and cycles will give more profit for the investors. To analyze such information we must use networks like DTR as they rely on the current information. [7]

Based on the results obtained, we conclude that the two companies under consideration have less correlation. The stock value change does not depend on the stock exchange index. It is dependent on the sentiments of social media. The prediction using machine learning algorithms do not give accurate results since the correlation between them is less. Results are not accurate as the dependency is less than 50% for all variables. But the graph trends between TCS and Infosys show similar variation except at some points where it was opposite. Combination of natural language processing techniques for analysis and summarization of text can help in handling such cases. Apart from the parameters which are considered in this paper there can be other parameters which can affect the stock shares such as Inflation, Deflation, International currency and gold rates and International economic policies, etc. Other techniques that can be used are Momentum, Mean Reversion and Martingales. [8]

This paper presents a survey of different techniques such as machine learning techniques, hidden Markov model, ARIMA model and also deep learning techniques. It is observed that selection of the right parameters for the dataset used for prediction plays important role good prediction accuracy. Various machine learning models as well as hybrid and ensemble model give higher rate of accuracy. To get even better accuracy fundamental analysis can be used which uses sentiment analysis and feature selection along with machine learning and deep learning techniques. [9]

Stock price prediction is an important issue in the financial world, as it contributes to the development of effective strategies for stock exchange transactions. In this paper, we propose a generic framework employing Long Short-Term Memory (LSTM) and convolutional neural network (CNN) for adversarial training to forecast high-frequency stock market. This model takes the publicly available index provided by trading software as input to avoid complex financial theory research and difficult technical analysis, which provides the convenience for the ordinary trader of nonfinancial specialty. Our study simulates the trading mode of the actual trader and uses the method of rolling partition training set and testing set to analyze the effect of the model update cycle on the prediction performance. Extensive experiments show that our proposed approach can effectively improve stock price direction prediction accuracy and reduce forecast error. [10]

**3. SPECIFICATION**

1. **Introduction**
   1. **Project scope**

Although system will be accurate but may not predict the stock value correctly when two similar dataset (stock pattern) are compared.

* 1. **User classes and Characteristics**

**User**

* Stock whose data you want system to predict
* Training data
  1. **Assumptions and dependencies**

It is assume perfect working conditions with min 2 GB RAM and above 1GHz speed

1. **Functional requirement**

* **User Module**
  + When user will enter the web application, he/she will see the stock related news on the home page.
  + User can also see the predictions of that stock.
* **Admin**
  + Admin takes the stock data from Yahoo finanace API.
  + Admin will load the data using ticker symbols.
  + Store data of different values of particular stock
* **Prediction**
  + Using various best algorithms to improve the accuracy of stock prediction we proposes a stock price method based on FBProphet for the stock closing price of the next days.
  + We can adopt fbprophet to efficiently extract features from the data which are the items of the past 7 years.

1. **External Interface Requirement**

**3.3.1 User Interfaces**

* The system shall use standard user interface controls such as buttons, text boxes, radio buttons, check boxes, labels, list boxes, spin boxes, combo boxes, sliders, scroll bars, tabs, tool tips, progress bars, and file selection dialogs .
* The user interfaces shall be presented as web pages and shall be displayed by web browsers.
* The system shall provide a visual display such as a progress bar icon while any function is being performed.
* Front-end tools : Visual Studio Code, Streamlit
* Back-end tools: Python, Firebase

**3.3.2 Hardware Interfaces**

* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor – Dell G3

**3.3.3 Software Interfaces**

* Programming language: Python
* Database: Firebase
* Fbprophet model

1. **Non-functional Requirement**

**3.4.1 Performance Requirements**

* RAM should be empty enough to run program smoothly.

## 3.4.2 Safety Requirements

* The data safety must be ensured by arranging for a secure and reliable transmission media.
* The source and destination information must be entered correctly to avoid any misuse or malfunctioning.
* Safety requirements against the natural disaster and accidents.
* Failures due to technical issues.

**3.4.3 Security**

* The access to the system is given only to valid users. We need a specific ID and password to get access to the system.
* Communication needs to be restricted when the application is validating the user or license.
* By incorporating a robust and proven SQL into the system, reliable performance and integrity of data is ensured. There must be a power backup for server system.

**3.4.4 Software Quality Attributes**

* **Reliability**

In order to ensure reliability, this system is being designed using software that is established to be stable and easy to use.

* **Availability**

This system is designed to run 24/7 and be readily available to the user.

* **Software Quality Assurance**

The quality of software and the security patches will be provided by the team of concerning developers, if the user finds any loopholes he/she may contact the concerned team and the team will provide them with suitable patches.

* **Maintainability**

The parts of the system coded in Python.

* **Portability**

The user interface shall run on Microsoft Internet Explorer 8.0, Mozilla Firefox 3.5, Google Chrome 3.0, and Apple Safari 4.0.

1. **System Requirement**
   * 1. **Database requirement**

Firebase

* + 1. **Software requirement**

1. Operating System - Windows 10, Ubuntu 20
2. Front End - Streamlit
3. IDE - Python Idle
4. Algorithm- Fbprophet
   * 1. **Hardware Requirement**
5. Processor - Intel i3/i5/i7
6. Speed - 1.1 GHz
7. RAM - 2 GB(min)
8. Hard Disk - 40 GB
9. Floppy Drive - 1.44 MB
10. Key Board - Standard Windows Keyboard
11. Mouse - Two or Three Button Mouse
12. Monitor – DELL G3
13. **Analysis Model: SDLC model Implementation**

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/Implementation and maintenance. Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In the waterfall approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

**1. Requirement Gathering and analysis**: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

**2. System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

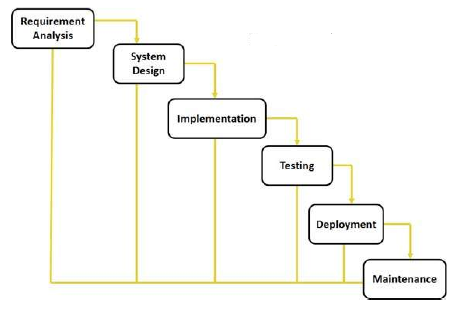


Fig 3.6

**3. Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

**4. Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**5. Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

**6. Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name “Waterfall Model”. In this model phases do not overlap.

**4. System Design**

**4.1 System Architecture**

Stock price of particular commodity and stock value in previous years is acts as input to training. Statistical features are extracted and analyzed from given data and fed to classifier for comparison.it forms stock chart as training data. Current stock values of commodity are given as input to system. Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.

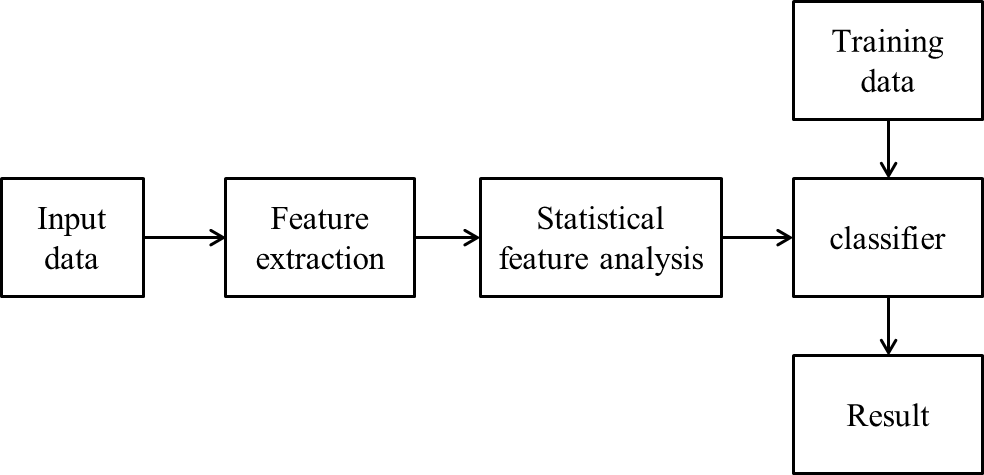


Fig 4.1 Architechture

Statistical analysis theory is the frequently-used method of data feature extraction. It can analyze the statistical laws when several objects and several indices are interrelated; it is a comprehensive analysis method. Statistical methods are based on forceful theory, have lots of algorithms, and can effectively analyze and process the data. Analyzing the data features or classifying the data subsets should subject to statistics irrelevant assumption.

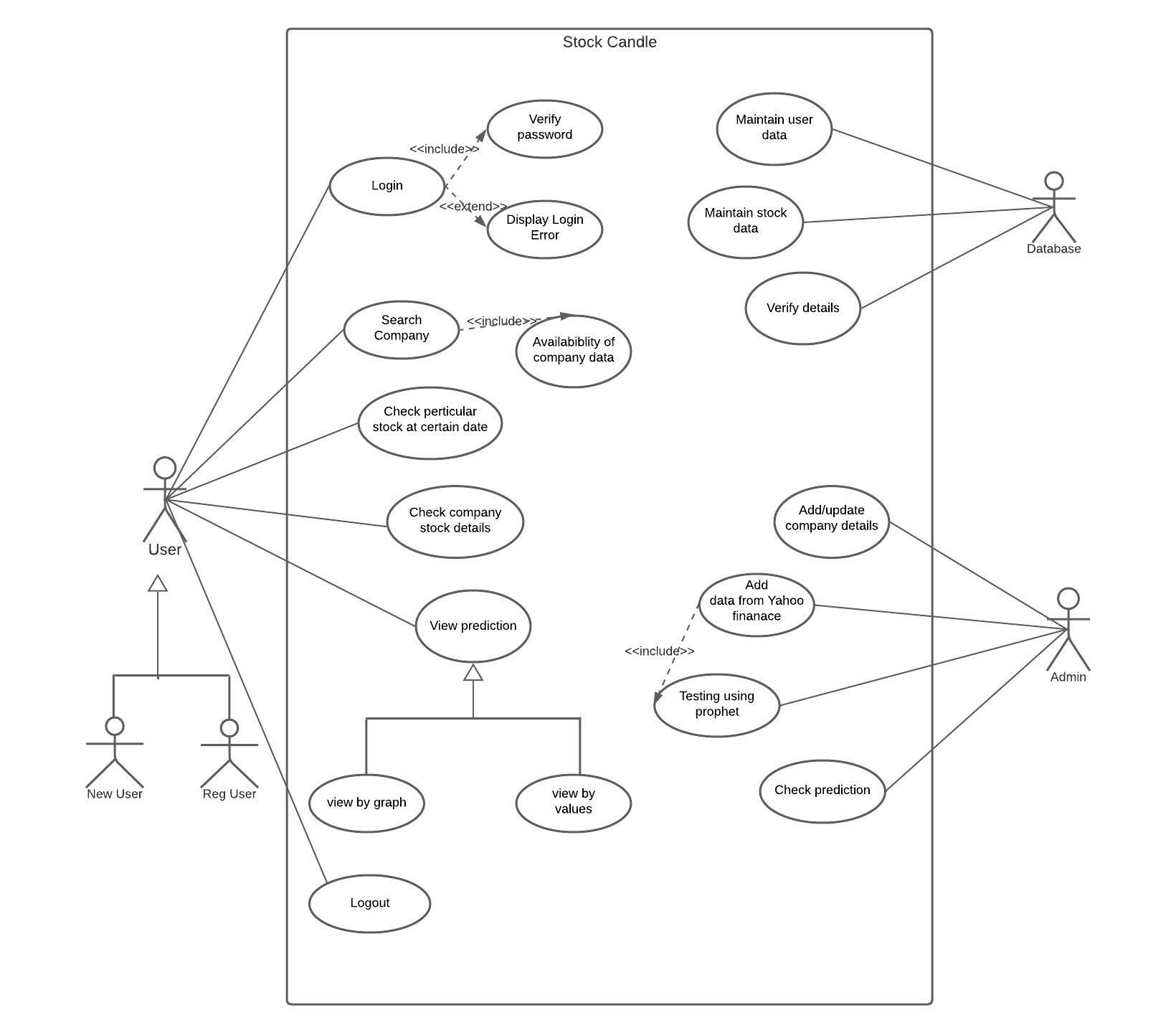
Classifier classifies features from input data and matched with statistical features from data and according to comparison classifies system stock into following categories:

1. Abrupt decline: if stock values are abruptly decreasing over particular period of time in past years then system predicts that stocks prices will abruptly decline in future.
2. Smooth decline: system predicts that in near future prices of particular stock will decline smoothly (comparing pattern from last some years as specified in stock chart).
3. Stable: stock prices will be stable over the time period
4. Smoothly increase: value of stock will smoothly increase
5. Abruptly increase: if stock values are abruptly increasing over particular period of time in past years then system predicts that stocks prices will abruptly decline in future.

**4.2 UML Diagrams**

**4.2.1 Use Case**

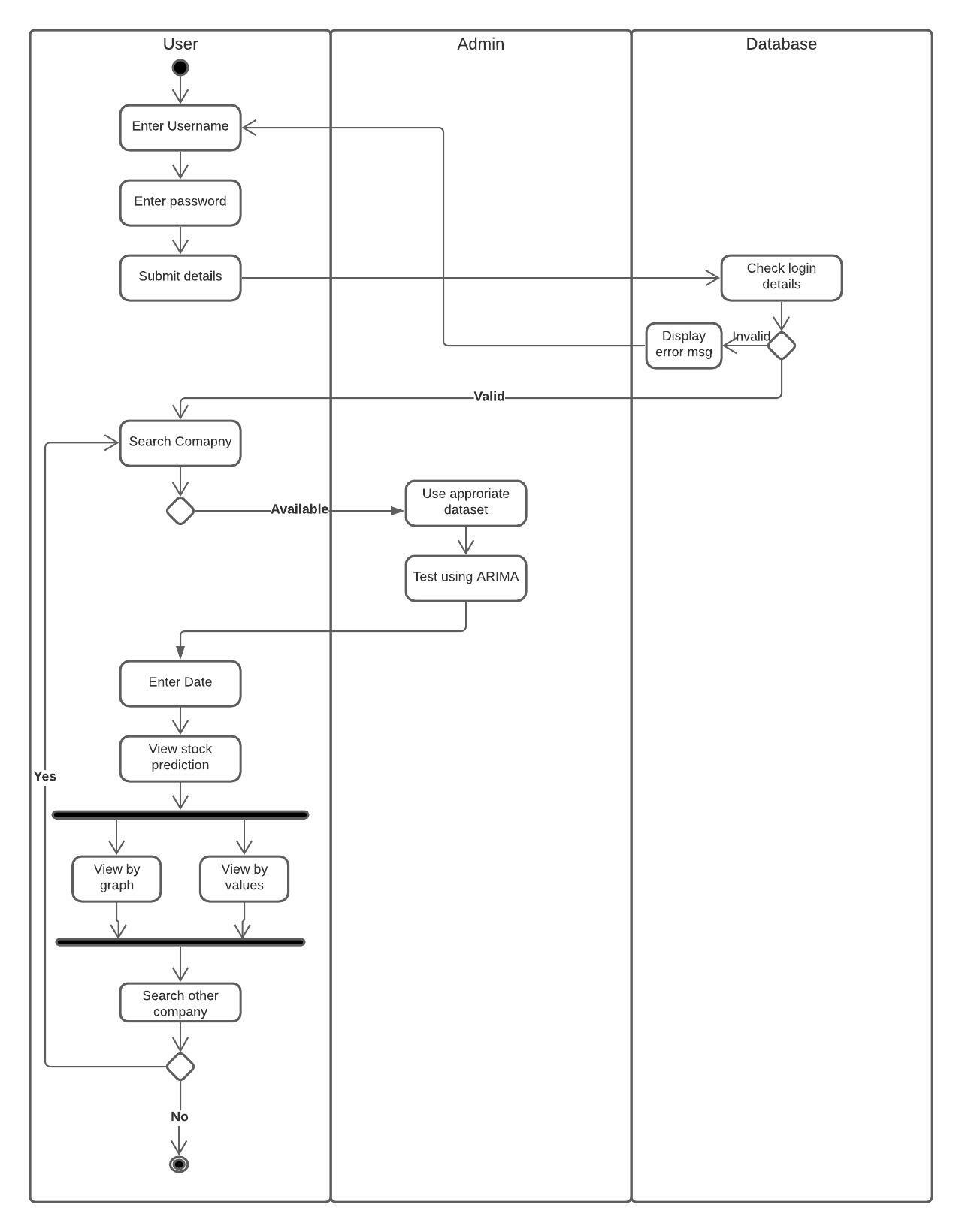
The purpose of use case diagram is to capture the dynamic aspects of a system. Use case diagrams are used to gather the requirements of a system including external and internal influences. These Requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionality, use case are prepared and actors are identified.

****

**Fig 4.2.1**

**4.2.2 Activity Diagram**

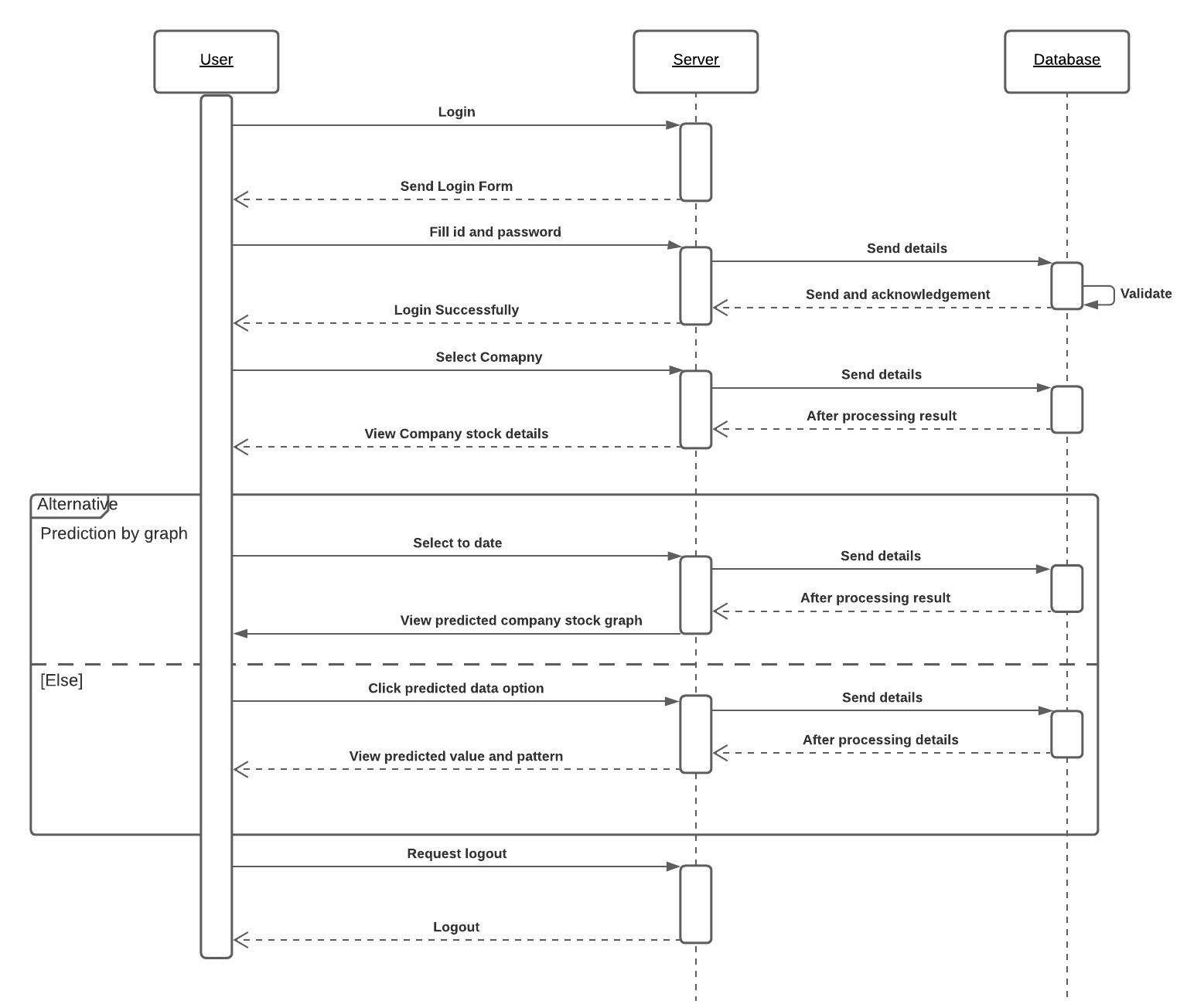
Activity is a particular operation of the system. Activity is not only used for visualizing the dynamic nature of the system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

****

**Fig 4.2.2**

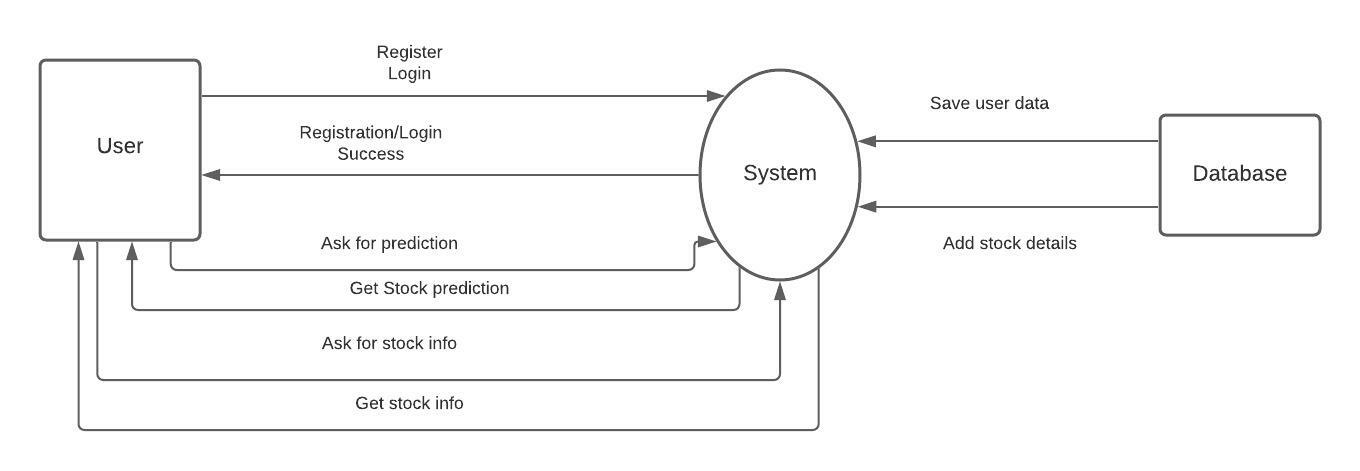
**4.2.3 Sequence Diagram**

The sequence diagram captures the time sequence of the message from one object to another and the collaboration diagram describes the organization of objects in the system taking part in the message flow

****

**Fig 4.2.3**

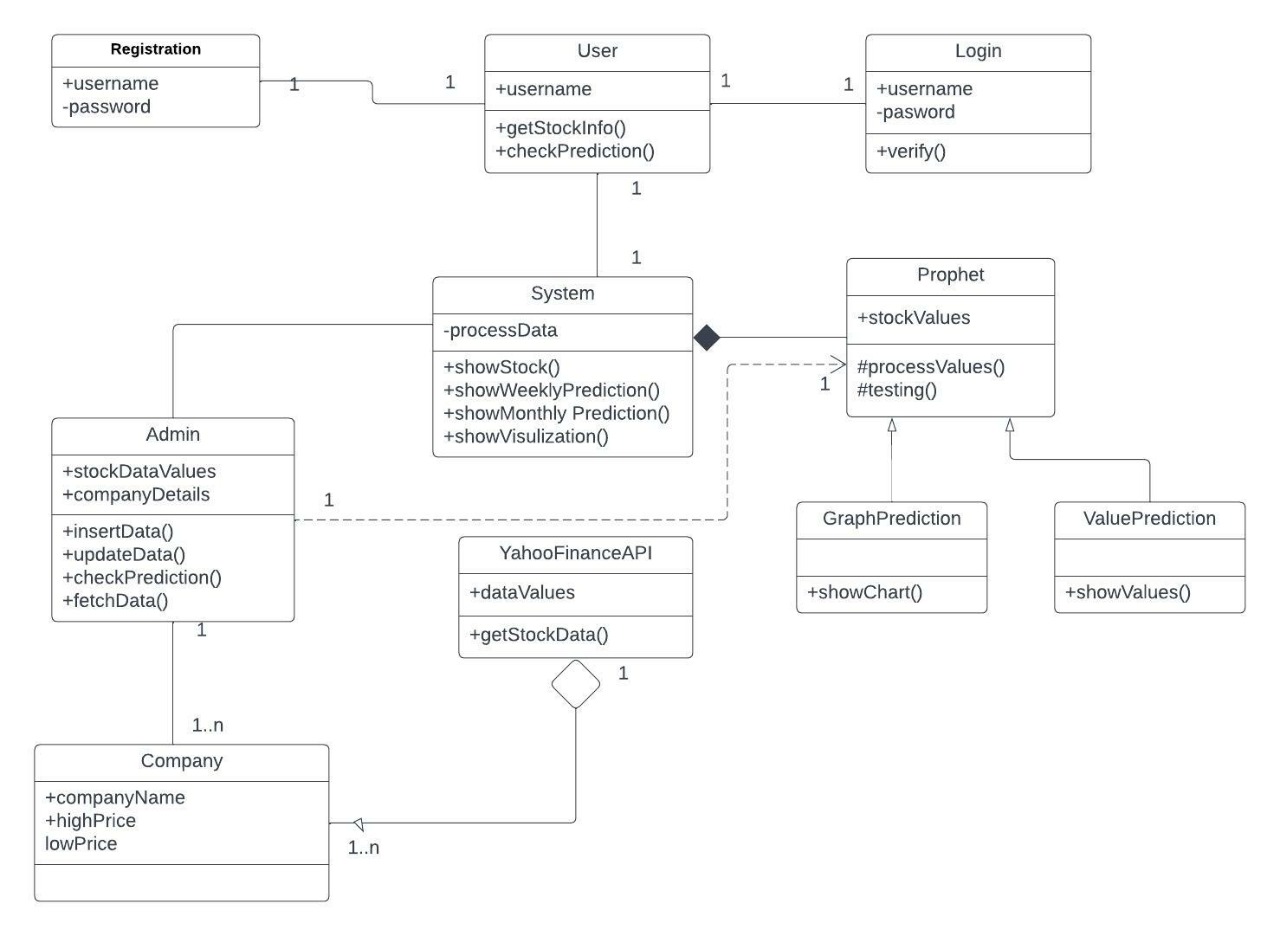
**4.2.4 DFD**



**Fig 4.2.4**

**4.2.5 Class Diagram**

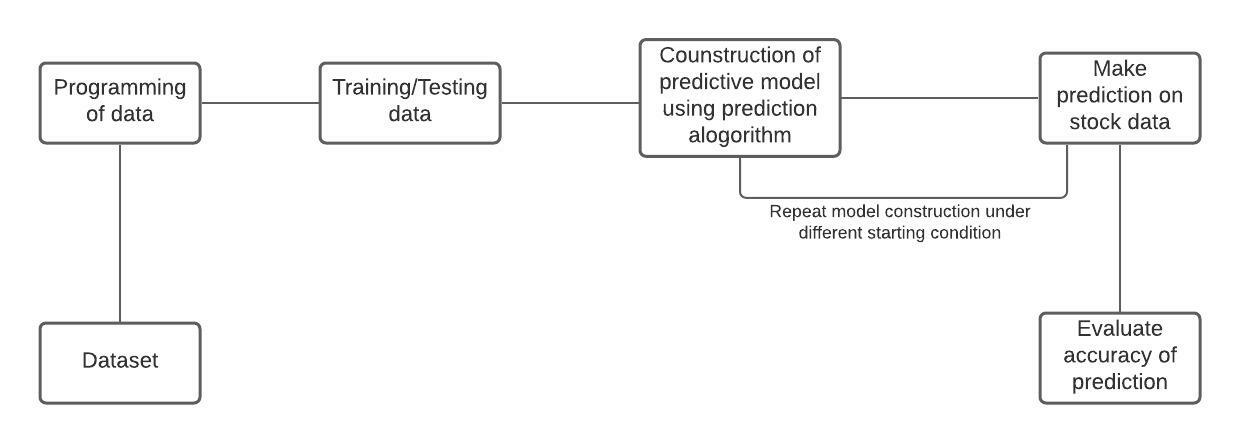
The Purpose of class diagram is to model the state view of an application. Class diagram are the only diagram which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

****

**Fig 4.2.5**

**4.2.6 Architecture diagram**

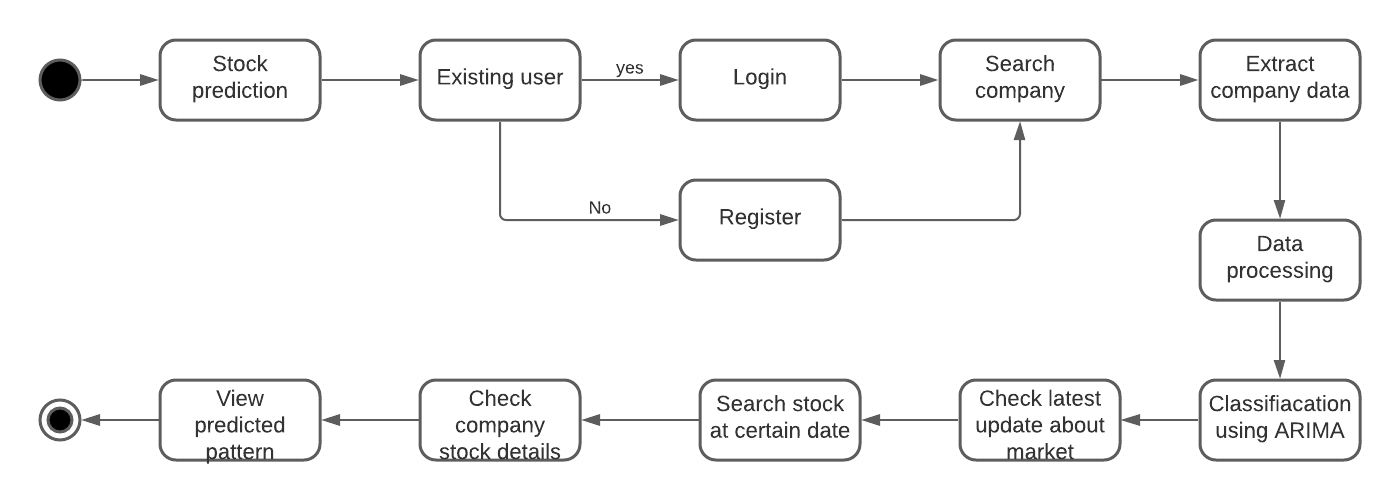
A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.



**Fig 4.2.6**

**4.2.7 State Diagram**

State diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of state diagram is to model lifetime of an object from creation to termination.



**Fig 4.2.7**

* + - 1. **Implementation**

**5.1 Implementation plan - I**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **July 15** | **July 31** | **Aug 20** | | **Sept 25** | **Oct 12** | **Oct 26** | | **Nov 22** | **Dec 10** | **Dec 28** |
| Initiated the project |  |  |  | |  |  |  | |  |  |  |
| Communication |  |  |  | |  |  |  | |  |  |  |
| Literature survey |  |  |  | |  |  |  | |  |  |  |
| Defined scope |  |  |  |  |  |  |  | |  |  |  |
| Developed SRS |  |  |  |  |  |  |  | |  |  |  |
| Planned the project |  |  |  | |  |  |  | |  |  |  |
| Designed mathematical model |  |  |  | |  |  |  | |  |  |  |
| Feasibility Analysis |  |  |  | |  |  |  | |  |  |  |
| Developed work breakdown structure |  |  |  | |  |  |  |  |  |  |  |
| Planned project schedule |  |  |  | |  |  |  |  |  |  |  |
| Designed UML and other diagrams |  |  |  | |  |  |  | |  |  |  |
| Designed test plan |  |  |  | |  |  |  | |  |  |  |
| Designed risk management plan |  |  |  | |  |  |  | |  |  |  |

**Table 5.1**

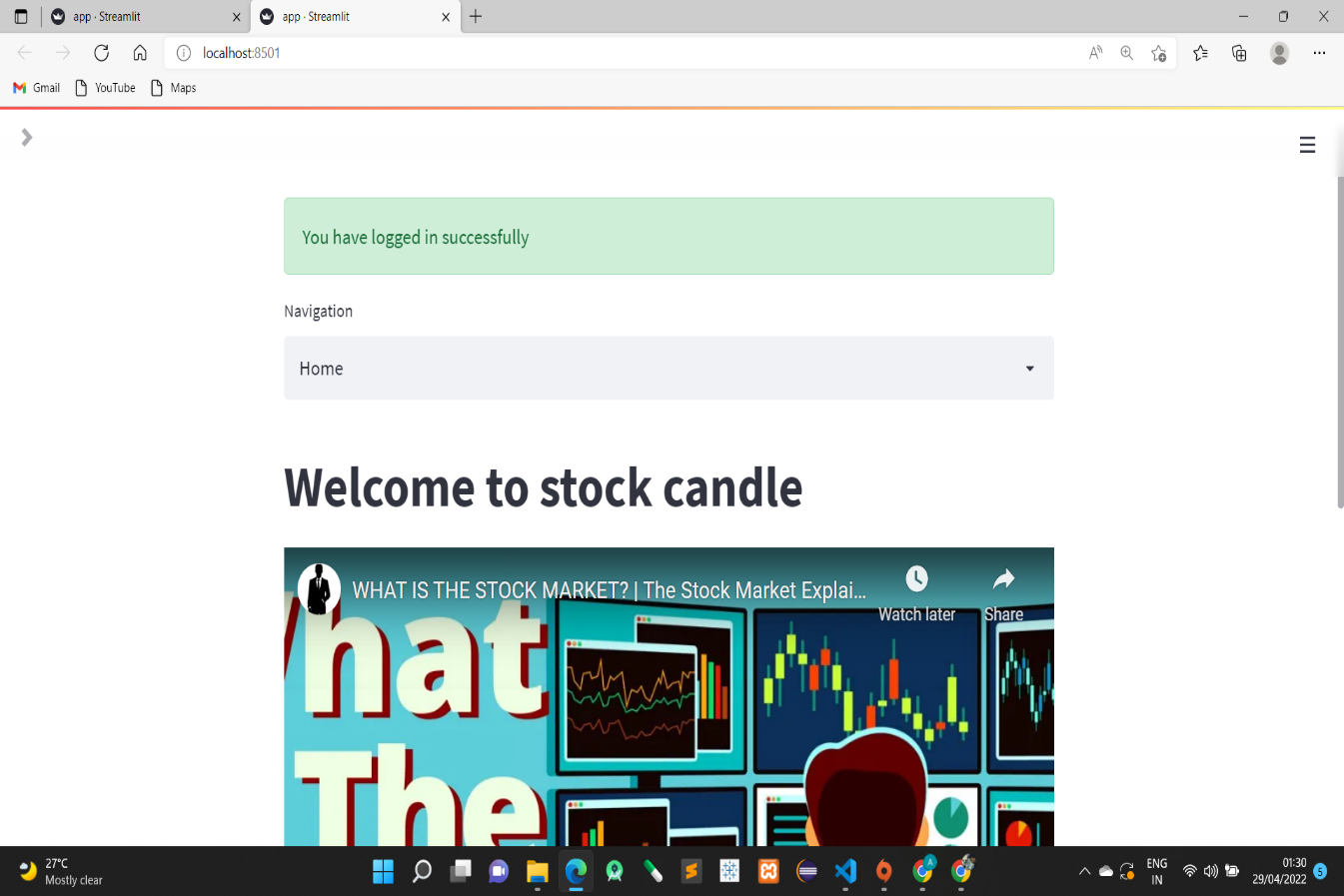
**5.2 Implementation plan - II**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Dec 12 | Jan 01 | Jan 24 | | Jan 27 | Feb 09 | Feb 25 | | Mar 14 | Apr 03 | Apr 15 |
| Project planning and requirement gathering |  |  |  | |  |  |  | |  |  |  |
| Learned Streamlit and flask Framework |  |  |  | |  |  |  | |  |  |  |
| Study of fb’s prophet algorithm |  |  |  | |  |  |  | |  |  |  |
| Updated UML diagrams |  |  |  |  |  |  |  | |  |  |  |
| Prepared project report |  |  |  |  |  |  |  | |  |  |  |
| Designind front end –login/registration |  |  |  | |  |  |  | |  |  |  |
| Creating firebase database |  |  |  | |  |  |  | |  |  |  |
| Extracted and Added ticker symbols |  |  |  | |  |  |  |  |  |  |  |
| Creating prediction model |  |  |  | |  |  |  | |  |  |  |
| Connecting model to front end |  |  |  | |  |  |  | |  |  |  |

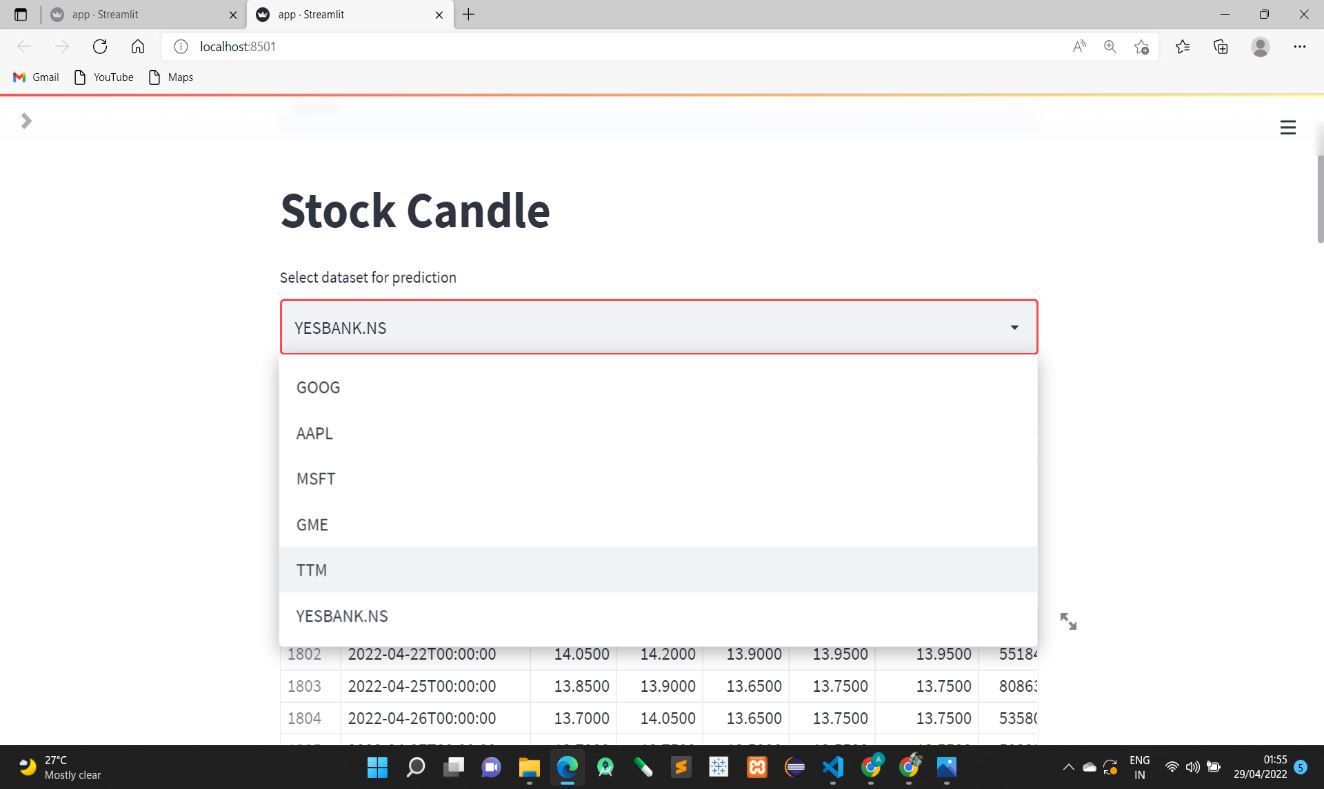
**Table 5.1.2**

1. **Result and Evaluation**

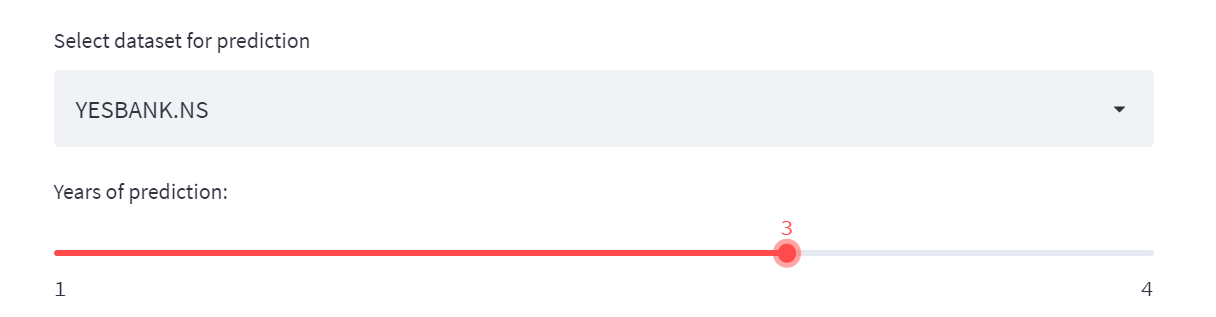
**6.1 User Interface**



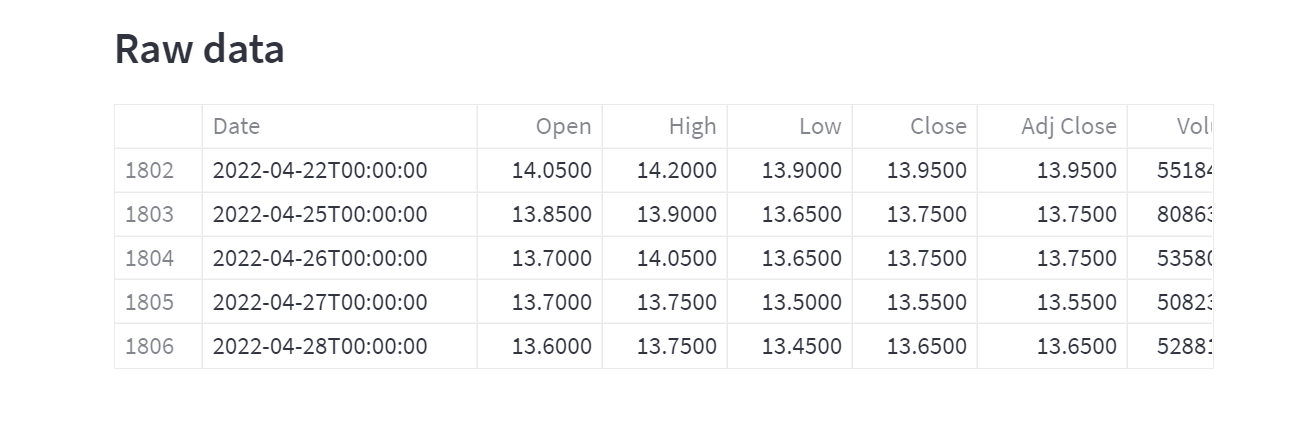
**Fig 6.1.1 Home Page**

****

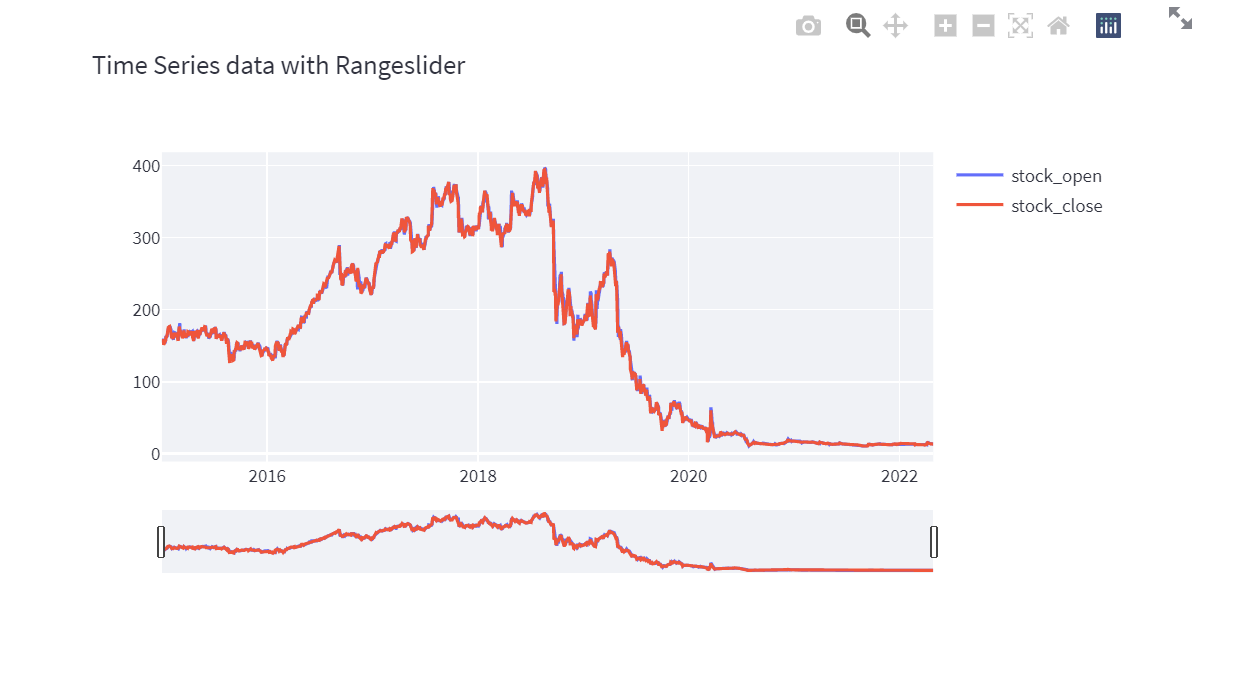
**Fig 6.1.2 Selecting ticker symbol of company**

****

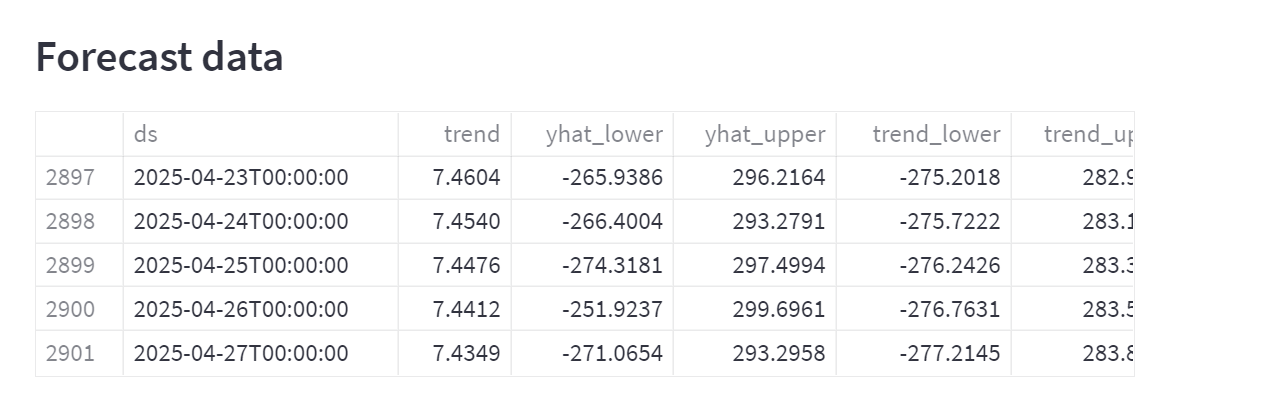
**Fig 6.1.3 Choosing the year with slider**

****

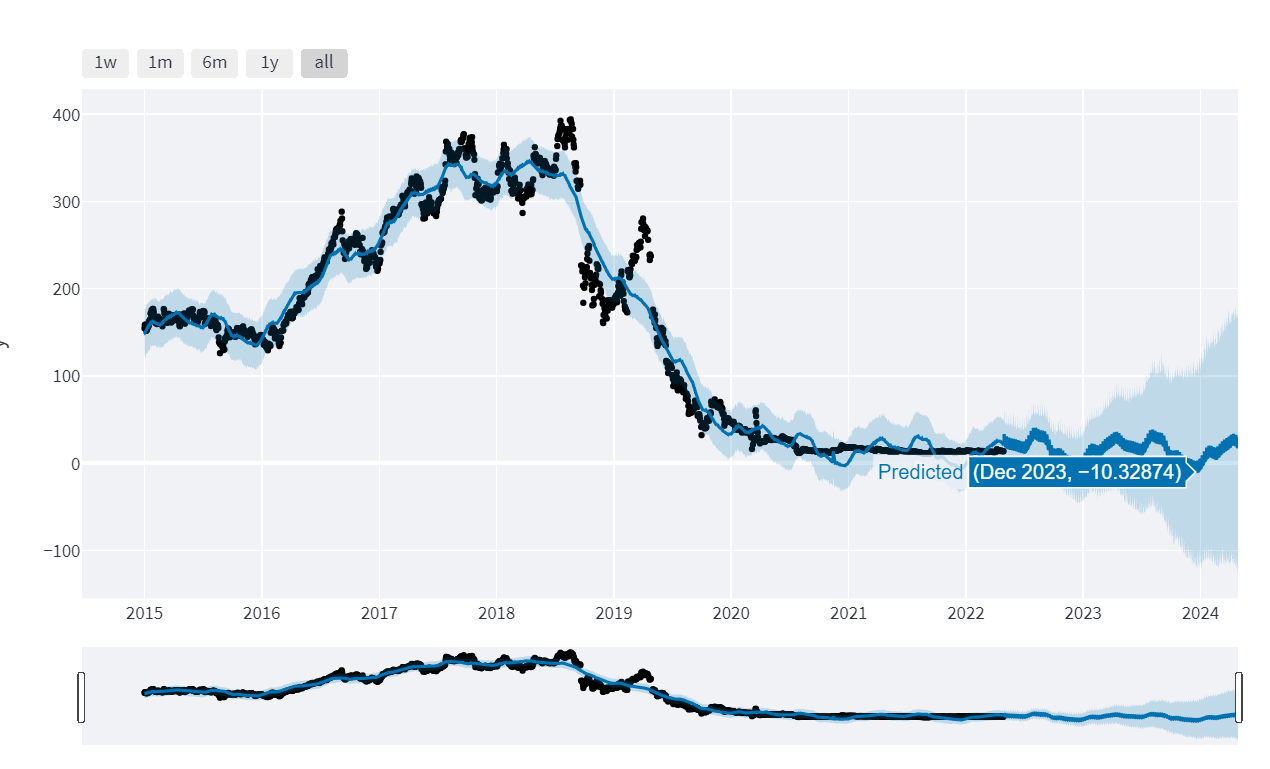
**Fig 6.1.4 Raw data with tail values**

****

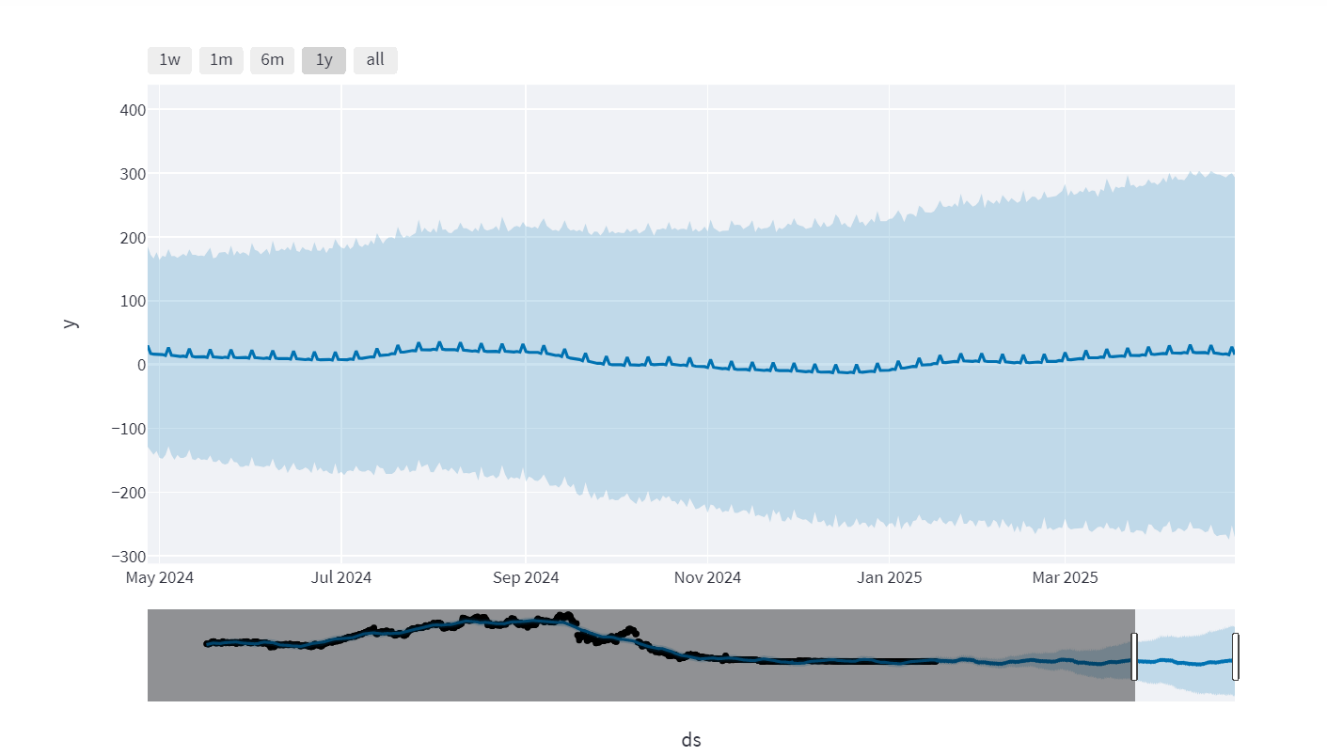
**Fig 6.1.5 Raw data in graphical format**

****

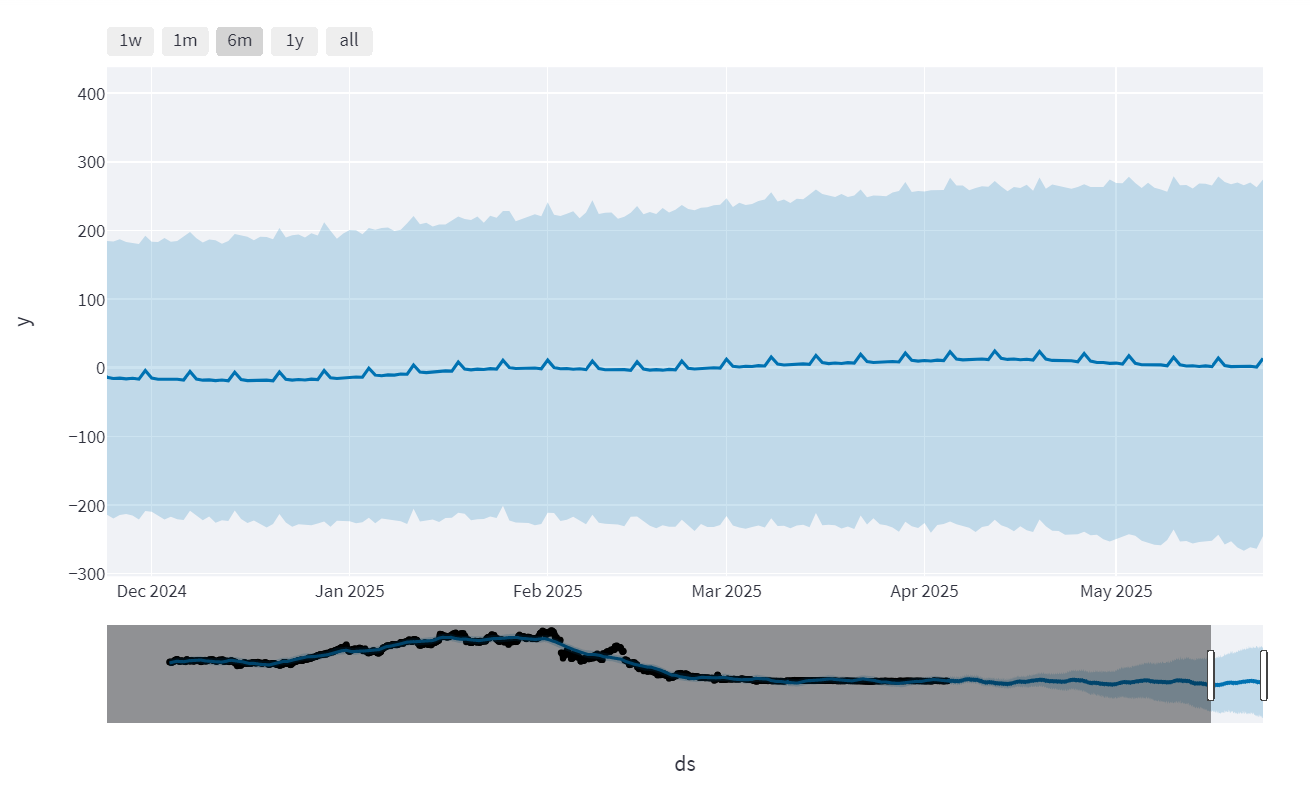
**Fig 6.1.6 Forecast data with head values**

****

**Fig 6.1.7 Raw and forecast data**

****

**Fig 6.1.8 Year wise stock prices**

****

**Fig 6.1.9 Monthly stock prediction**

* 1. **Source Code**

**app.py**

import streamlit as st

from multiapp import MultiApp

from apps import home, data, model # import your app modules here

#import libraries

import pyrebase

from datetime import datetime

#configuration

firebaseConfig = {

'apiKey': "AIzaSyD77BfmCYnREsOm5ccGPcICR4NGXuzHqP4",

'authDomain': "stock-eb968.firebaseapp.com",

'projectId': "stock-eb968",

'databaseURL': "https://stock-eb968-default-rtdb.firebaseio.com/",

'storageBucket': "stock-eb968.appspot.com",

'messagingSenderId': "983107711191",

'appId': "1:983107711191:web:f655d875f4170c617bc092",

' measurementId': "G-HT1ETMS7J1"

}

#firebase authentication

firebase = pyrebase.initialize\_app(firebaseConfig)

auth = firebase.auth()

#database

db= firebase.database()

storage = firebase.storage()

st.sidebar.title("Login / Sign Up")

#authentication

choice = st.sidebar.selectbox('login/signup', ['Login', 'Sign up'])

email = st.sidebar.text\_input('Enter email')

password = st.sidebar.text\_input('Enter password', type = 'password')

if choice == 'Sign up':

handle = st.sidebar.text\_input('Please input your app handle name', value = 'Default')

submit = st.sidebar.button('Create my account')

if submit :

user = auth.create\_user\_with\_email\_and\_password(email, password)

st.success('Your account is created successfully')

st.balloons()

#sign in

user = auth.sign\_in\_with\_email\_and\_password(email, password)

db.child(user['localId']).child("Handle").set(handle)

db.child(user['localId']).child("ID").set(user['localId'])

st.title('Welcome ' + handle)

st.info('Login via login drop down')

if choice == 'Login' :

login = st.sidebar.button('Login')

if login :

user = auth.sign\_in\_with\_email\_and\_password(email, password)

#st.write('<style>div.row-widget.stRadio > div{flex-direction:row;}</style>')

st.success('You have logged in successfully')

app = MultiApp()

#st.markdown("""

# Multi-Page App

#This multi-page app is using the [streamlit-multiapps](https://github.com/upraneelnihar/streamlit-multiapps) framework developed by [Praneel Nihar](https://medium.com/@u.praneel.nihar). Also check out his [Medium article](https://medium.com/@u.praneel.nihar/building-multi-page-web-app-using-streamlit-7a40d55fa5b4).

#""")

# Add all your application here

app.add\_app("Home", home.app)

#app.add\_app("Data", data.app)

app.add\_app("Model", model.app)

# The main app

app.run()

**home.py**

from cgitb import html

import streamlit as st

from PIL import Image

import streamlit.components.v1 as components

def app():

st.title('Welcome to stock candle')

#video2 = open("Video1.mp4", "rb")

#st.video(video2)

#st.video(video1, start\_time = 25)

#image = Image.open('img2.jpeg')

# col1, col2 = st.columns([1, 1])

# video\_html = """

#<video controls width="250" autoplay="true" muted="true" loop="true">

# <source

# src="https://www.youtube.com/watch?v=DrKLYvLPidw"

# type="video/mp4" />

#</video>

#"""

#col2.markdown(video\_html, unsafe\_allow\_html=True)

st.info('Refer below video if you are beginner')

st.video("https://www.youtube.com/watch?v=NOX53AJPypw")

st.info('Below video describes about future and option trading')

st.video("https://www.youtube.com/watch?v=MiybniIIvx0" , start\_time = 30)

# st.video("https://media.istockphoto.com/videos/moving-financial-chart-with-downtrend-line-candlestick-graph-and-in-video-id1157256430")

#st.image("https://www.google.com/search?q=online+stock+market+images&tbm=isch&ved=2ahUKEwjytq2XmKr3AhW\_k9gFHXYZA-4Q2-cCegQIABAA&oq=online+stock+market+images&gs\_lcp=CgNpbWcQAzoHCCMQ7wMQJzoGCAAQCBAeUNsFWL8XYMwYaABwAHgAgAHVAYgB1AmSAQUwLjcuMZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=cvFjYvK1Br-n4t4P9rKM8A4&bih=746&biw=1536&rlz=1C1CHBF\_enIN867IN871#imgrc=DTMQECynulqK3M", caption='Sunrise by the mountains')

#st.write('This is the `home page` of this multi-page app.')

#st.write('In this app, we will be building a simple classification model using the Iris dataset.')

# bootstrap 4 collapse example

# """https://medium.com/analytics-vidhya/ep5-adding-media-files-in-our-streamlit-web-app-74564af03642#:~:text=To%20add%20a%20video%20we,saved%20your%20opened%20video%20file."""

**model.py**

import streamlit as st

from datetime import date

import yfinance as yf

from fbprophet import Prophet

from fbprophet.plot import plot\_plotly

from plotly import graph\_objs as go

def app():

START = "2015-01-01"

TODAY = date.today().strftime("%Y-%m-%d")

st.title('Stock Candle')

stocks = ('GOOG', 'AAPL', 'MSFT', 'GME' , 'TATAMOTORS.NS' , 'YESBANK.NS' , 'NESTLEIND.NS' , 'ITC.NS')

selected\_stock = st.selectbox('Select dataset for prediction', stocks)

n\_years = st.slider('Years of prediction:', 1, 4)

period = n\_years \* 365

@st.cache

def load\_data(ticker):

data = yf.download(ticker, START, TODAY)

data.reset\_index(inplace=True)

return data

data\_load\_state = st.text('Loading data...')

data = load\_data(selected\_stock)

data\_load\_state.text('Loading data... done!')

st.subheader('Raw data')

st.write(data.tail())

# Plot raw data

def plot\_raw\_data():

fig = go.Figure()

fig.add\_trace(go.Scatter(x=data['Date'], y=data['Open'], name="stock\_open"))

fig.add\_trace(go.Scatter(x=data['Date'], y=data['Close'], name="stock\_close"))

fig.layout.update(title\_text='Time Series data with Rangeslider', xaxis\_rangeslider\_visible=True)

st.plotly\_chart(fig)

plot\_raw\_data()

# Predict forecast with Prophet.

df\_train = data[['Date','Close']]

df\_train = df\_train.rename(columns={"Date": "ds", "Close": "y"})

m = Prophet()

m.fit(df\_train)

future = m.make\_future\_dataframe(periods=period)

forecast = m.predict(future)

# Show and plot forecast

st.subheader('Forecast data')

st.write(forecast.tail())

st.write(f'Forecast plot for {n\_years} years')

fig1 = plot\_plotly(m, forecast)

st.plotly\_chart(fig1)

st.write("Forecast components")

fig2 = m.plot\_components(forecast)

st.write(fig2)

**multiapp.py**

"""Frameworks for running multiple Streamlit applications as a single app.

"""

import streamlit as st

class MultiApp:

"""Framework for combining multiple streamlit applications.

Usage:

def foo():

st.title("Hello Foo")

def bar():

st.title("Hello Bar")

app = MultiApp()

app.add\_app("Foo", foo)

app.add\_app("Bar", bar)

app.run()

It is also possible keep each application in a separate file.

import foo

import bar

app = MultiApp()

app.add\_app("Foo", foo.app)

app.add\_app("Bar", bar.app)

app.run()

"""

def \_\_init\_\_(self):

self.apps = []

def add\_app(self, title, func):

"""Adds a new application.

Parameters

----------

func:

the python function to render this app.

title:

title of the app. Appears in the dropdown in the sidebar.

"""

self.apps.append({

"title": title,

"function": func

})

def run(self):

# app = st.sidebar.radio(

app = st.selectbox(

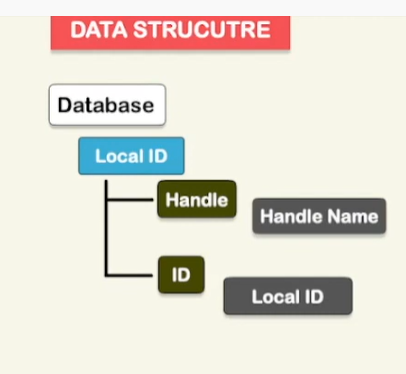
'Navigation',

self.apps,

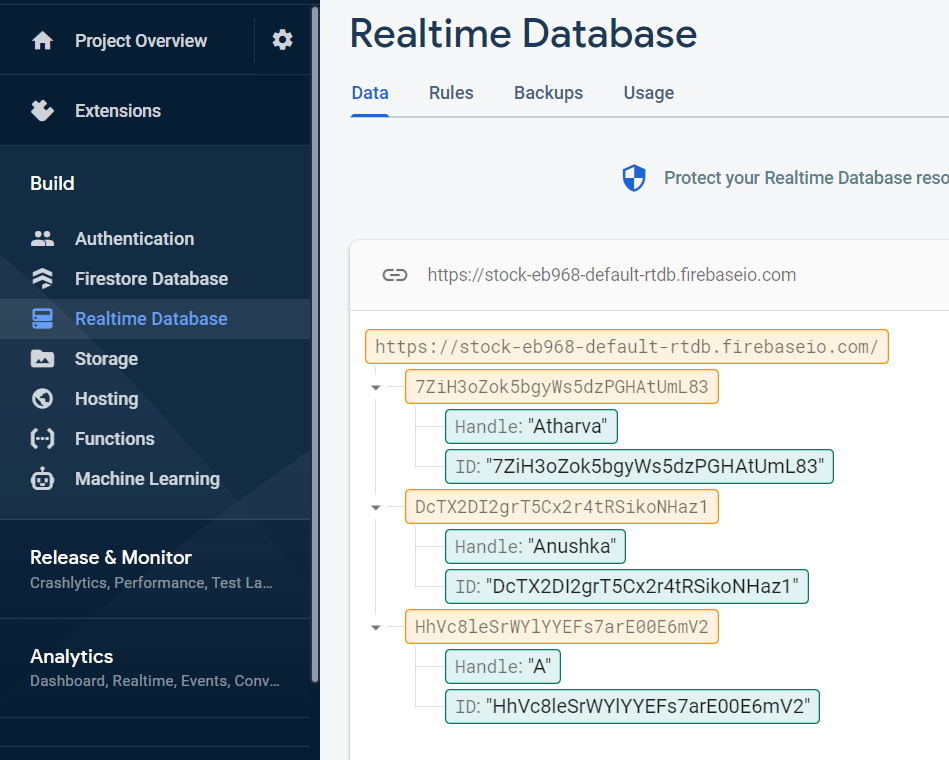
format\_func=lambda app: app['title'])

app['function']()

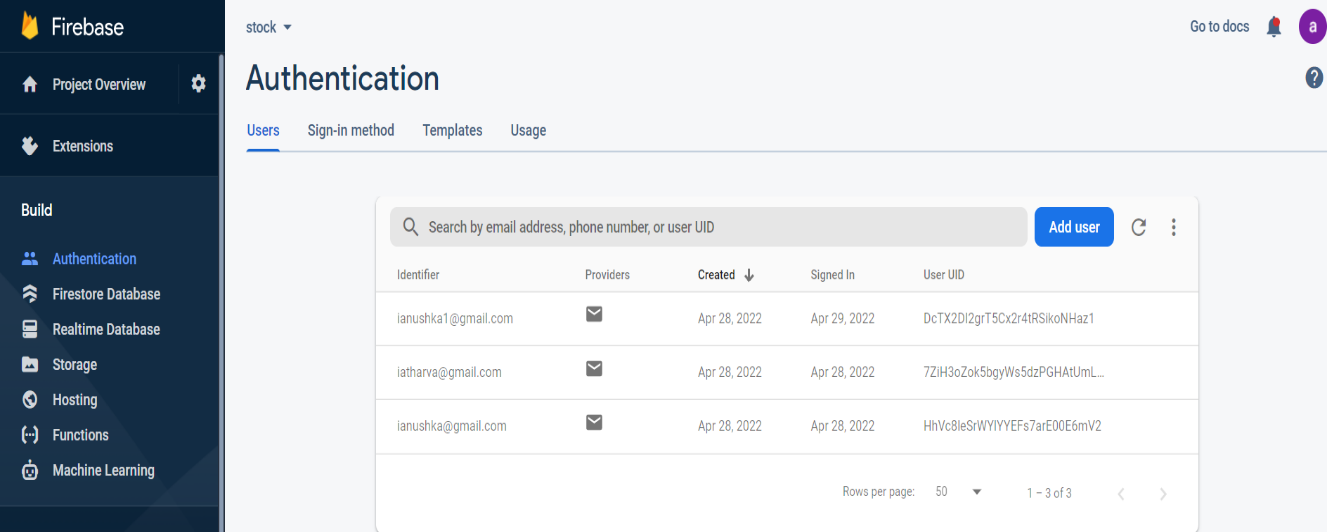
* 1. **Firebase Database :**

****

**Fig 6.3.1 Database Structure (A)**

****

**Fig 6.3.1 Database Structure (B)**

****

**Fig 6.3.2 Registered Users**

* + - 1. **CONCLUSION AND FUTURE WORK**

**6.1 Conclusion**

Anticipating the securities exchange cost is exceptionally well known among financial specialists as speculators need to know the arrival that they will get for their ventures. Generally the specialized experts and intermediaries used to foresee the stock costs dependent on costs, volumes, value designs and the essential patterns. Today the stock value expectation has turned out to be mind boggling than before as stock costs are influenced because of organization's money, political environment and cataclysmic events and so on. The arrival from the offer market is constantly unsure in nature thus conventional procedures won't give precise expectation. A ton explore has been made around there and progressed insightful procedures going from unadulterated numerical models and master frameworks to neural systems have likewise been proposed by numerous budgetary exchanging frameworks for stock value expectation.

New solution to our website would be chatbox and nifty fifty future option. Limitation to this can be, If the system takes data along with some noise it recognizes the data as completely different information whereas the human visual system will identify it as the same information with the noise.

**6.2 Future Work**

* Use of fbprophet for time series analysis.
* We can use better pre-processing technique to eliminate noise from information so that in further processes like classification and prediction noise doesn’t make any impact
  + - 1. **REFERANCES**

1. Marc Velay and Fabrice Daniel, “Stock Chart Pattern recognition with Deep Learning”, Researchgate, June 2018
2. Victor Skuratov, Konstantin Kuzmin, Igor Nelin, Mikhail Sedankin, “Application of a convolutional neural network to create a detector of technical analysis figures on exchange quotes charts”, (2019), «EUREKA: Physics and Engineering» Number 6 DOI: 10.21303/2461-4262.2019.001055
3. International Journal of Current Trends in Engineering & Technology Volume: 02, Issue: 01 (JAN-FAB, 2016) 18 Stock Market Prediction Using Support Vector Machine Mr. Sachin Sampat Patil, Prof. Kailash Patidar, Assistant Prof. Megha Jain
4. Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction Hyejung Chung and Kyung-shik Shin Sustainability 2018, 10, 3765; doi:10.3390/su10103765
5. An optimized CNN based robust sentiment analysis system on big social data using text polarity feature Komalpreet Kaur, Chitender Kaur, Tarandeep Kaur Bhatia International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6, April 2019
6. Performance Comparison of Machine Learning Methods for Solving Handwriting Character Recognition Problem Ş.G.KIVANÇ1, A.E. BAKTIR2 and B.ŞEN3 International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES’18), May 11-13, 2018 Safranbolu, Turkey
7. JSRD - International Journal for Scientific Research & Development| Vol. 6, Issue 07, 2018 | ISSN (online): 2321-0613 All rights reserved by www.ijsrd.com 408 Stock Market Prediction using RFR, DTR & SVR Ravikant1 Suman Kumar Swarnkar2L. P. Bhaiya
8. International Journal of Computer Sciences and Engineering Open Access Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693 Stock Market Analysis and Prediction using Hadoop and Machine Learning Piyush Jain, Kaustubh Bhat, HarshalKesharwani3 , Pritesh Bhate, Khushboo P Khurana
9. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072 © 2019, IRJET | Impact Factor value: 7.211 | ISO 9001:2008 Certified Journal | Page 4842 Stock Market Forecasting Techniques: A Survey Rashmi Sutkatti1, Dr. D. A. Torse
10. Xingyu Zhou, Zhisong Pan, Guyu Hu, Siqi Tang, Cheng Zhao “Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets”, Mathematical Problems in Engineering Volume 2018, Article ID 4907423, 11 pages