# Baqai Medical University

Baqai Institute of Information Technology

**Stocks Trend Prediction System**

**Web App in Streamlit**

By

**Hassan Ahmed Khan**

**201907008**

Supervised by

###### Engr. Kashif Jamil

Assistant Professor

Project Submitted towards the partial fulfillment of the requirement of the Baqai

Medical University for the award of Bachelor of Science degree

in Software Engineering

Baqai Institute of Information Technology

III-B-3/17, Nazimabad, Karachi





**STOCKS TREND PREDICTION**

Web Application in Streamlit

**Project by**

**2023 Hassan Ahmed Khan**

**CERTIFICATE**

This is to certify that the project report entitled *Stocks Trend Prediction* being submitted by *Hassan Ahmed Khan* in partial fulfillment of circular requirements for the award of the Degree of Bachelor of Software Engineering is a record of benefited work carried out by them under our guidance and supervision.

Name: Engr. Kashif Jamil

Supervisor Designation: Assistant Professor

Name: Engr. Kashif Jamil

FYP Committee Member Designation: Assistant Professor (BS-SE Department)

Name: Ms. Khalida Naveed

FYP Committee Member Designation: Assistant Professor (BS-BI Department)

Name: Dr. Mohammad Shahid

FYP Committee Member Designation: Assistant Professor (BS-SE Department)

Name: Mr. FARRUKH ZAMAN

Director BIIT Designation: Associate Professor

**ACKNOWLEDGEMENT**

It is a pleasure to acknowledge my debt to the many people involved, directly or indirectly, in the completion of my final year project.

First of all, I want to thanks Mr. Farukkh Zaman Hai (HOD) of BIIT, for his kind cooperation and guidance. Help and guidance Mr. Kashif Jamil is also very much valuable in this regard.

Finally, I want to acknowledge all of them who guide and help me in completion of my project from the institution.

**ABSTRACT**

Investing in stock’s markets is risky and it needs a lot of research and time to make the right decision for earning money in stock markets. Novice investors are limited to specific investment knowledge and lack investment tools to gain wealth in the stock’s markets. A web application for stock price prediction using CNN – LSTM, Time Series Algorithm (Prophet) is developed to tackle the problem mentioned.

The project's goal is to provide a third-party investment web application for individual investors to use to navigate the stock market. This is accomplished through the application of Machine Learning and Deep Learning. Web technology that is Streamlit. Several methodologies and models for stock price prediction have been developed, such as Convolutional Neural Network, Long Short Term Memory (LSTM), and Prophet. The model architectures and hyper-parameters are automatically optimized. Evolutionary algorithms are used to seek for information. For trend prediction, promising results have been discovered. It provides the path for new features to be added and tested. In the future, I'll be constructing Auto ML models in the financial context.

**Keywords:** Convolutional Neural Network (CNN), Long short-term memory (LSTM), Machine Learning, Deep Learning, Recurrent Neural Networks (RNNs), Stock Exchange, Stock Market, Analysis, Prediction, Time-Series Prediction, Mean Squared Error (MSE), Mean Absolute Error (MAE).

**Table of Contents**

**7**

7

7

7

7

8

9

10

11

11

12

**13**

13

14

15

15

15

16

17

19

22

22

22

28

28

**29**

29

32

33

34

34

35

**41**

41

42

42

43

**44**

44

53

**57**

**58**

**60**

61

71

76

86

94

100

1. **Introduction……………………………………………………………………………………….**
   1. Overview/Background…………………………………………………………………………
   2. Objectives……………………………………………………………………………………...
   3. Literature Survey………………………………………………………………………………
      1. Paper I…………………………………………………………………………………
      2. Paper II………………………………………………………………………………..
      3. Paper III……………………………………………………………………………….
      4. Technical Review……………………………………………………………………
   4. Product Scope………………………………………………………………………………...
   5. Definitions, Acronyms & Abbreviations……………………………………………………..
   6. Project Deliverable…………………………………………………………………………...
2. **Methodology/ Observations & Calculations…………………………………………………...**
   1. Design………………………………………………………………………………………...
   2. Implementation……………………………………………………………………………….
      1. Data Collection………………………………………………………………………
      2. Data Pre-processing………………………………………………………………….
      3. Data Analysis………………………………………………………………………..
      4. Feature Engineering…………………………………………………………………
      5. Machine/Deep Learning……………………………………………………………..
      6. Algorithm……………………………………………………………………………
   3. Testing………………………………………………………………………………………..
      1. Test Plan……………………………………………………………………………..
      2. Test Cases……………………………………………………………………………
   4. Verification & Validation…………………………………………………………………….
   5. Product Functionality…………………………………………………………………….......
3. **Specific Requirements…………………………………………………………………………..**
   1. External Interface Requirement (User Interface, Software Interface)……………………….
   2. Software Process Model……………………………………………………………………..
   3. Functional & Non-Functional Requirements………………………………………………...
   4. Tools & Techniques………………………………………………………………………….
   5. Required Hardware & Software with Specification………………………………………….
   6. Behavior Requirement (Use Case View, UML, HLD, DLD)………………………………..
4. **Project Planning…………………………………………………………………………………**
   1. Distribution of Work (Project Management)………………………………………………...
   2. Gantt Chart…………………………………………………………………………………...
   3. Back End Development………………………………………………………………………
   4. Front End Development……………………………………………………………………...
5. **Program Coding…………………………………………………………………………………**
   1. Coding………………………………………………………………………………………..
   2. Output Screenshots…………………………………………………………………………...
6. **Conclusion / Future Recommendation………..………………………………………………..**
7. **References…………..……………………………………………………………………………**
8. **Other Reports / Documents……………………………………………………………………..**
   1. SRS…………………………………………………………………………………………...
   2. Software Analysis Report…………………………………………………………………….
   3. Project Management………………………………………………………………………….
   4. Software Design Report……………………………………………………………………...
   5. Software Test Reports………………………………………………………………………..
   6. FYP Poster Presentation…………………………………………………………………......
9. **Introduction**
   1. **Overview / Background**

The process of investing in stock markets is easy and convenient nowadays. Investors interested in investing in stock markets can open an account and trade wherever they want as long as they have their mobile smart devices with them. Technologies have made the investment in stock markets more convenient than before for investors. There are few ways that investors can earn through investing in stock markets which are capital gain and dividend. However, investing in stock markets is risky as investors may lose their money if they did not do proper research. Stocks investing requires investors to do many planning and analyses to find the right timing and right stocks to invest in, and it is a time-consuming task.

The prediction of a stock price is one strategy for investors to identify which stocks to invest. With predicted stock prices, investors will know whether the stock price will rise or drop in the future. Therefore, stock price prediction is an important aspect for investors to sort out the possible high return stocks and invest at the right time.

* 1. **Objectives**

1. To develop a web application for stock price prediction by the end of this project.
2. To provide stock price prediction using time series algorithms in the web application.
3. To provide information about stocks i.e. institutional shareholders, balance sheet, opening & closing price of current date etc.
4. To provide news about searched stocks.
   1. **Literature Reviews / Survey**

**1.3.1. Paper Title 1: Stock Market Prediction using Machine Learning Prof. S .P.**

**Pimpalkar, Jenish Karia , Muskaan Khan, Satyam Anand, Tusha Mukherjee**

The core motive of this project is to analyze the market and predict its performance using various machine learning techniques. The predictive architecture will use various attributes as the input and will predict whether the market value will be Positive or Negative. The various attributes that are used in the model include Oil rates, Foreign Exchange Rate, Interest rate, Gold and Silver rates, NEWS, twitter news feed and Pattern Matching. The different ML techniques including Support Vector Machine with regression (SVR) and Recurrent Neural Network are used. The algorithm SVR produced the most efficient results among all.

**Advantages of Paper**

1. The meta-learner will perform the task of optimizing the outputs of the algorithms.
2. It itself is another more powerful algorithm which will improve the overall efficiency of the system.

**Disadvantages of Paper**

1. Accuracy would decrease when setting more levels of stock market movement.
2. These results indicate that the stock price is unpredictable when traditional classifier is used.

**How to overcome the problems mentioned in Paper**

1. The input datasets will first be processed through the basic algorithms and the outputs from theses algorithms will be feeded to the LSTM layer.
2. The LSTM layer will perform the task of optimizing the outputs of the algorithms.
3. It itself is another more powerful algorithm which will improve the overall efficiency of the system.
4. The LSTM upon analyzing all the outputs will produce the final prediction for the consumer.

**1.3.2. Paper Title 2: Automated Stock Price Prediction Using Machine Learning**

**Mariam Moukalled Wassim El-Hajj Mohamad Jaber**

We propose an automated trading system that integrates mathematical functions, machine learning, and other external factors such as news sentiments for the purpose of achieving better stock prediction accuracy and issuing profitable trades. Particularly, we aim to determine the price or the trend of a certain stock for the coming end-of-day considering the first several trading hours of the day.

**Advantages of Paper**

1. Identified the best time interval for stock price prediction.
2. Identified the best news scenario and that each stock is affected differently by news.
3. ITEM 3

**Disadvantages of Paper**

1. The existence system reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores.
2. It can predict the stock value of one company at a time.

**How to overcome the problems mentioned in Paper**

1. The main component of this system is the financial news articles collected from yahoo finance and represented as noun phrases; all the collected noun phrases are represented as vector of binary values indicating the presence or absence of a phrase in the article.
2. The second main component of this system is the stock price data collected in one minute time frame.
3. Then, the final major task after collecting the data and formalizing the inputs was building and training the AI model.

**1.3.3. Paper Title 3: Stock Market Prediction Using Machine Learning Algorithms**

**K.Hiba Sadia, Aditya Sharma, Adarsh Paul, Sarmistha Padhi, Saura Sanyal**

The main objective of this paper is to find the best model to predict the value of the stock market. During the process of considering various techniques and variables that must be taken into account, we found out that techniques like random forest, support vector machine were not exploited fully. In, this paper we are going to present and review a more feasible method to predict the stock movement with higher accuracy. The first thing we have taken into account is the dataset of the stock market prices from previous year. The dataset was pre-processed and tuned up for real analysis. Hence, our paper will also focus on data preprocessing of the raw dataset. Secondly, after preprocessing the data, we will review the use of random forest, support vector machine on the dataset and the outcomes it generates.

**Advantages of Paper**

1. We are able to train the machine from the various data points from the past to make a future prediction.
2. The basic approach of the supervised learning model is to learn the patterns and relationships in the data from the training set and then reproduce them for the test data.
3. The tuned up dataframe allowed us to prepare the data for feature extraction.

**Disadvantages of Paper**

1. The existing system fails when there are rare outcomes or predictors, as the algorithm is based on bootstrap sampling.
2. The existence system reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores.
3. It doesn’t focus on external events in the environment, like news events or social media.

**How to overcome the problems mentioned in Paper**

1. The input datasets will first be processed through the basic algorithms and the outputs from theses algorithms will be feeded to the LSTM layer.
2. The LSTM layer will perform the task of optimizing the outputs of the algorithms.
3. It itself is another more powerful algorithm which will improve the overall efficiency of the system.
4. The meta-learner upon analyzing all the outputs will produce the final prediction for the consumer.

**1.3.4. Technical Review**

Learning tools represent key enablers for empowering material scientists and engineers to accelerate the development of novel materials, processes and techniques. One of the aims of using such approaches in the field of materials science is to achieve high-throughput identification and quantification of essential features along the process-structure-property-performance chain. In this contribution, machine learning and statistical learning approaches are reviewed in terms of their successful application to specific problems in the field of continuum materials mechanics. They are categorized with respect to their type of task designated to be either descriptive, predictive or prescriptive; thus to ultimately achieve identification, prediction or even optimization of essential characteristics. The respective choice of the most appropriate machine learning approach highly depends on the specific use-case, type of material, kind of data involved, spatial and temporal scales, formats, and desired knowledge gain as well as affordable computational costs.

**Advantages of Technology**

1. Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans.
2. With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own.

**Reasons to use this Technology**

1. To identify important insights in data, and prevent fraud.
2. Feature Engineering.
3. Efficiency
4. Best Training
   1. **Product Scope**

The goal of this project is to provide a web application to predict a stock price of a company according to its previous historical data. Stock Market Prediction is composed of main components: a company’s historical data of stock which will help to analyze the current and previous changes of stock price. The above proposed model is easy to implement considering the available technology infrastructure. The model is simple, secure and scalable. The proposed model is based on serial communication. These model will help the investors to invest their money according to the predicted value, investors may have less chances of loss and a very huge chance of making more profit.

* 1. **Definitions, Acronyms and Abbreviations**

**CNN-LSTM Layer:**

A CNN-LSTM model is a combination of CNN layers that extract the feature from input data and LSTMs layers to provide sequence prediction. The CNN-LSTM is generally used for activity recognition, image labeling, and video labeling. Their common features are that they are developed for the application of visual time series prediction problems and generating textual annotations from image sequences.

**List of Abbreviations:**

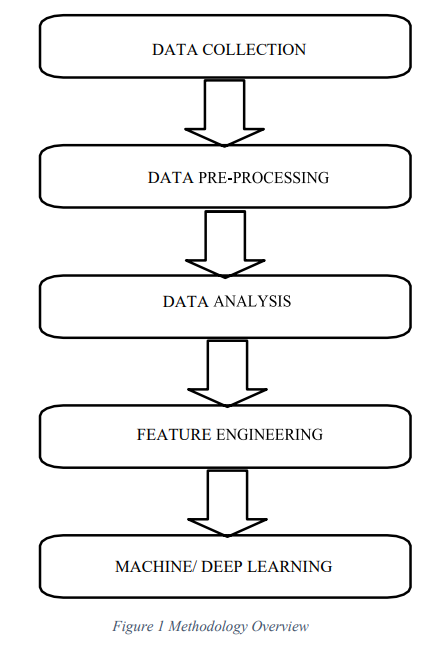
|  |  |
| --- | --- |
| LSTM | Long Short Term Memory Network |
| CNN | Convolutional Neural Network |
| ANN | Artificial Neural Network |
| MAE | Mean Absolute Error |
| VAL\_MAE | Value Mean Absolute Error |
| MSE | Mean Squared Error |
| VAL\_MSE | Value Mean Squared Error |
| ROE | Return on Equity |
| ML | Machine Learning |
| DL | Deep Learning |

* 1. **Project Deliverables**

|  |  |
| --- | --- |
| **Phases** | **Deliverables** |
| D1 | Final Year Project Proposal |
| P1 | Defense Presentations |
| D2 | Literature Review |
| D3 | Software Requirement Specification (SRS) |
| D4 | Project Analysis Report |
| D5 | Project Management Report |
| D6 | Software Design Report |
| D7 | Software Testing Report |
| D8 | Weekly Report Submission |
| P2 | Poster Presentation (Mid-Presentations) |
| D9 | FYP Report Submission |
| P3 | Final Project Presentation / Final Evaluation |

1. **Methodology**
   1. **Design**

It is essential to devise a cohesive methodology in order to ensure iterative development throughout the research. Thus, this chapter elucidates the steps that are required to complete the project. It explains the approach to collect, prepare and analyze the collected data. It then describes the machine learning approach. Lastly, the way in which various machine learning algorithms will be evaluated is discussed with the help of mathematical equations. The following figure gives a brief overview of the methodology that will be followed. A comprehensive explanation of this figure is given in the subsections of this chapter.

****

* 1. **Implementation**
     1. **Data Collection**

The first step is to collect legitimate and adequate amounts of financial data. This is an integral step because data acts as a fuel to machine learning models. Additionally, insufficient data can possibly lead to unsatisfactory and unreliable predictions. In this project, two types of data are collected, namely Numeric Data. Numeric data is essentially historic stock data and includes the high, low, open, close and volume of stock traded over a period of time.

|  |  |  |  |
| --- | --- | --- | --- |
| **S no.** | **Stock/Index** | **Description** | **Ticker** |
|  | Google Inc. | Large Cap Stock | GOOG |
|  | Apple Inc. | Large Cap Stock | AAPL |
|  | Tesla Inc. | Large Cap Stock | TSLA |

**Numeric Data:**

Thus, in order to make predictions more robust, this project aims to encompass aspects of numeric data. Only a few selected stocks and indexes are considered. In order to obtain a list of these stocks, their symbols and metadata are used. Once the list is available, python scripts are written to retrieve data of the stocks from reliable Application Program Interfaces (API's) such as Alpha Vantage, Yahoo Finance, etc.



The structure of the collected data is explained below:

1. Date: Date is the index of the Data-Frame.
2. High: High denotes the highest value of the stock on a particular trading day.
3. Low: Low denotes the lowest value of the stock on a particular trading day.
4. Open: Open denotes the opening price of the stock on a particular trading day.
5. Close: Close denotes the closing price of the stock on a particular trading day. It simply gives the cash value of a share of the stock at the end of a particular trading day.
6. Volume: Volume denotes the volume of stock that a company traded on a particular trading day.
7. Adj. Close: Adj. Close is the adjusted closing price of the stock on a particular trading day.
   * 1. **Data Pre-Processing**

The collected data can be disorganized, inconsistent and noisy. This is not desirable as the quality of data determines how accurate the predictive model will be. As a result, it becomes imperative to pre-process, clean and manipulate the collected data in order to prepare it for exploratory data analysis. Firstly, inconsistent or duplicate values (if any) as well as missing values are handled. The missing values are either removed or replaced with the mean/median. The latter approach is better because simply removing values might lead to a risk of losing important information whereas replacing them will give more comprehensive results. Lastly, standardization, normalization and vectorization will be performed if needed.

* + 1. **Data Analysis**

Once the data is pre-processed, it is prepared for analysis. Data analysis is performed in order to understand the data better and draw insights from it. These insights are in the form of patterns, trends and correlations. Common techniques of exploratory data analysis are generating bar charts, pair plots, scatter plots, line charts etc.

Since the scope of this project involves different stocks in diverse markets, thorough analysis can be done to devise interesting results. For instance, the deviation of different stocks can be calculated and pair-plots can be created in order to visualize these deviations. This will help in determination of risks and the returns on various stocks. Further, analysis of the best and worst single day returns can be carried out and the reason behind peculiar results can be explored.

* + 1. **Feature Engineering**

The next step is to perform feature engineering because it will prepare the dataset to be compatible with the requirements of machine learning algorithms. It will also enhance the performance of machine learning algorithms resulting in higher accuracy.

Feature engineering involves feature creation, feature selection and feature extraction.

The importance and implementation of each of these is explained in detailed.

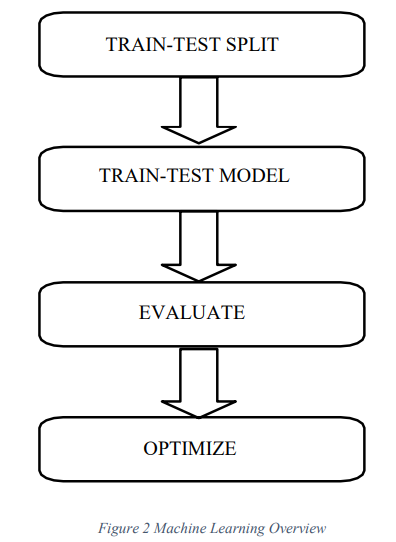
**Technical Indicators:**

Technical indicators are mathematical calculations that use past price and volume and help in predicting the price in future. The following table explains various technical indicators calculated along with their mathematical formula:

|  |  |  |  |
| --- | --- | --- | --- |
| **S. no.** | **Technical Indicator** | **Brief Explanation** | **Mathematical formulae** |
|  | Convolutional Neural Network (CNN) | The most established algorithm among various deep learning models is convolutional neural network (CNN), a class of artificial neural networks that has been a dominant method in computer vision tasks |  |
| 2. | Long short term memory (LSTM) | Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. | The mathematical formulation of LSTM in Keras? - Stack Overflow |
| 3. | CNN - LSTM | A CNN-LSTM model is a combination of CNN layers that extract the feature from input data and LSTMs layers to provide sequence prediction[65](https://www.nature.com/articles/s41598-021-93656-0#ref-CR65). |  |

* + 1. **Machine/Deep Learning**

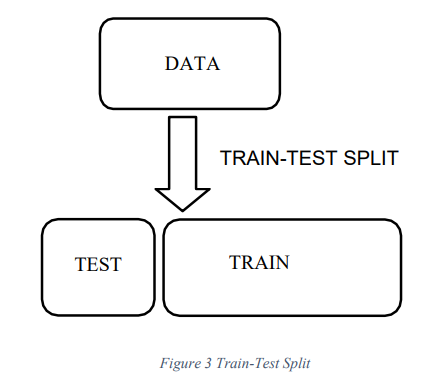
Once the data has been prepared and feature engineering has been performed, machine learning & Deep Learning algorithms can be implemented. The following figure gives an overview of the steps involved in machine learning:

****

As shown in the above figure, train-test split is performed in order to divide the data into two sets: training and testing datasets. After this split, the machine learning model is fit on the training data and it is tested on the testing data. The trained model is used to predict the prices and is evaluated to check the accuracy. It is then continuously optimized to achieve high accuracy.

* + - 1. **Train-Test Split**

Data is split into training data and testing data in the ratio 70:30 or 80:20. The following figure depicts this split of data:

****

The training data will be used to build machine learning & Deep Learning models while the testing data will be used to validate the results of the models.

* + - 1. **Train-Test Model**

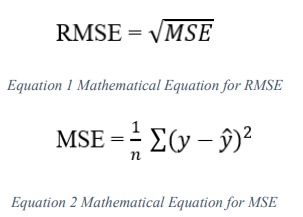
Various input features as well as output features of the machine learning model are decided. The input variables are used to predict an output variable. The input features can either be the raw features that were gathered during the process of data collection or other features such as technical indicators, lagging features, sentiment scores etc. can be calculated as well. On the other hand, the output variable can be the closing price, daily returns or stock movement. The problem of financial data forecasting can be considered as a regression problem or a classification problem. Depending on the approach, different machine learning/ deep learning algorithms can be implemented.

* + - 1. **Evaluation Metrics**

In order to ensure that the machine learning models are reliable, it is essential to evaluate their accuracy and find error scores. In case of classification problems for market trends, accuracy, precision, recall, F1 score as well as RMSE scores are popular methods of evaluation. Since these methods are widely DATA TEST TRAIN accepted by computer scientists and researchers globally, these metrics are considered. On the other hand, regression problems are often evaluated on the basis of Root Mean Squared Error (RMSE).

1. **Root Mean Squared Error (RMSE):**

RMSE is the square of root of the mean of squares of all the given errors. Although it is considered as a good measure of accuracy, it is scale- dependent i.e., it is not scaled to original error [18]. Below is the mathematical equation for RMSE:

****

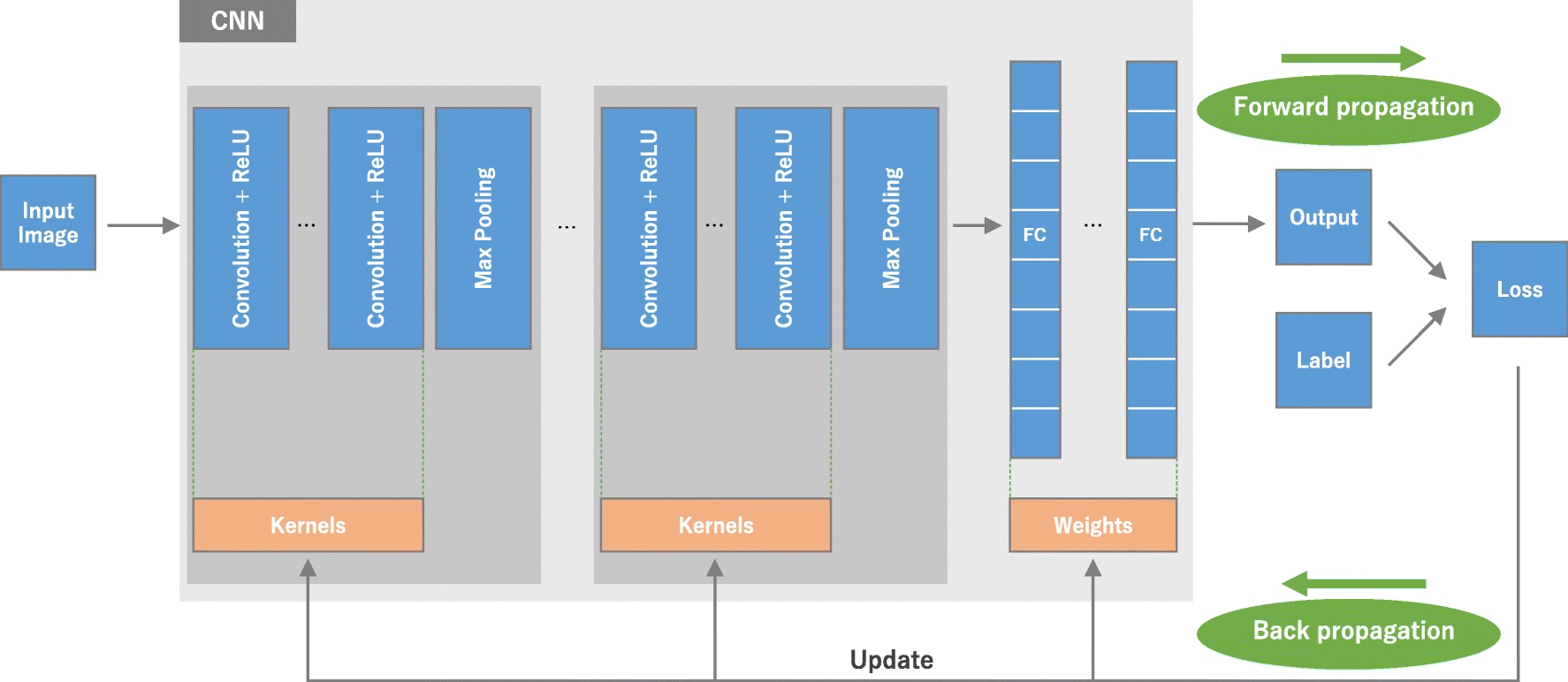
Where 'y' is actual value, 'ŷ' is predicted value and 'n' is number of samples.

* + - 1. **Optimization**

In order to achieve high accuracy, it is important to continuously optimize the machine learning models. This is done by finding hyper parameters that give the best possible performance. For instance, grid search can be implemented to iterate through different combinations of parameters. After this iteration, optimal parameters that give the highest accuracy are selected. Alternatively, features with more importance can also be visualized in order to decide which feature should be considered as an input variable and which feature can be dropped. Additionally, in case of neural networks, the number of nodes per layer and the activation function can be varied to produce better results.

* + 1. **Algorithm**

CNN is a type of deep learning model for processing data that has a grid pattern, such as images, which is inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns. CNN is a mathematical construct that is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification.

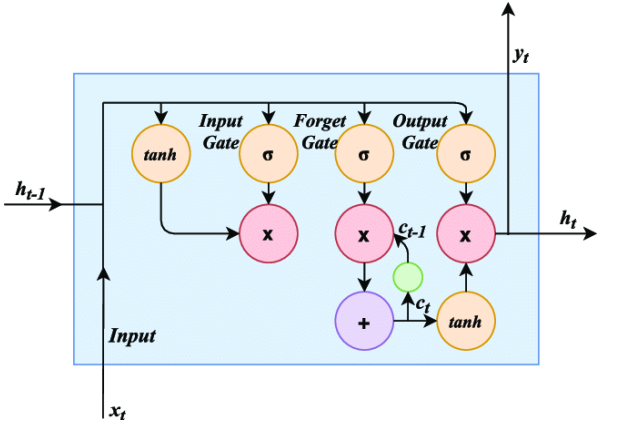


Recurrent Neural Networks or RNN are a type of neural network which are able to persist memory unlike traditional neural networks. RNNs are often implemented when dependence on historic data or data with a sequence is involved. This is because RNN's are able to understand the context and thus, can connect previous information to the present task. However, it was discovered by Hochreiter (1991) and Bengio et all (1994) that in practice, RNNs are not able to handle long term dependencies. This is because as the gap between the relevant information and the point where this information is needed increases, it is more probable that the information is lost along the chain.

As financial data forecasting involves time series data of a long period of time (or long-term dependencies), thus, RNN's are not very successful in stock price prediction.

To solve the shortcoming of RNN, there is a special type of RNN called Long Short-Term Dependencies or LSTMs which is able to handle long term dependencies. LSTM is a state-of-the-art deep learning method very popular for time-series prediction. It helps in capturing complex hidden patterns as well as time-series related patterns by storing past information and processing long sequences of data. In LSTM, when errors are back propagated from the output, they tend to remain in the memory of the block. Till the training is complete, error is continuously fed in each gate making back-propagation effective allowing LSTM to remember information for a longer duration.

In LSTM, small alterations to the information are made by operations such as multiplication and addition and this information flows through cell states. As a result, LSTM can selectively remember or forget information based on their importance. The sigmoid neural network layer and multiplication point wise function help in deciding which information is let through. As shown in the figure below, typical LSTM cell contains three gates, namely: input, forget and output.

****

**CNN-LSTM Model:**

We can define a CNN LSTM model in Keras by first defining the CNN layer or layers, wrapping them in a TimeDistributed layer and then defining the LSTM and output layers.

We have two ways to define the model that are equivalent and only differ as a matter of taste.

You can define the CNN model first, then add it to the LSTM model by wrapping the entire sequence of CNN layers in a TimeDistributed layer, as follows:

# CNN layers

    model.add(TimeDistributed(Conv1D(64, kernel\_size=3,

                                     activation='relu', input\_shape=(None, 100, 1))))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Conv1D(128, kernel\_size=3, activation='relu')))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Conv1D(64, kernel\_size=3, activation='relu')))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Flatten()))

    # LSTM layers

    model.add(Bidirectional(LSTM(100, return\_sequences=True)))

    model.add(Dropout(0.5))

    model.add(Bidirectional(LSTM(100, return\_sequences=False)))

    model.add(Dropout(0.5))

    # Final layers

    model.add(Dense(1, activation='linear'))

    model.compile(optimizer='adam', loss='mse', metrics=['mse', 'mae'])

* 1. **Testing**
     1. **Test Plan**

I used the unit testing and system testing technique, which is a very common testing method, to test a system to determine if it is functioning properly or not. A single unit or component of the software is tested in a unit test case. Here, every unit or component is tested, and I develop a unique test case for every element. System testing is to test the system that has all units fully integrated. System testing can verify the functionality and flow of the whole system.

* + 1. **Test Cases**
       1. **Unit Testing**
* **Create Account Module**

***Table 1. Using Test Cases for ‘Create Account’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Sign Up page navigation | 1. Users do not have any account to login. 2. Users want to register an account. 3. Click on dropdown button “login/signup” quote to navigate to Sign Up page. | The view navigates from Log In page to Sign Up page. | Pass |
| Field required validation | 1. Let username, email, and password. 2. Click on the “Create Account” button to register an account | Users will be alerted by a warning line in the sidebar that the field is required. | Pass |
| Email format validation | 1. Type an invalid email. (example: 1234word) 2. Fill in other fields with valid values. 3. Click on the “Create Account” button to register an account. | “Error: Unable not create user; Please try again” will display as a warning message on the sidebar. | Pass |
| Same username or email validation | 1. Fill in username or email that already exists in the system. 2. Fill in all other fields with valid values. 3. Click on the “Create Account” button to register an account. | “Error: Unable not create user; Please try again” will display as a warning message on the sidebar. | Pass |
| Account registration | 1. Fill in all fields with valid values.  2. Click on the “Create Account” button to register an account. | The "Welcome Default; Login with login drop down selection" success message will display. | Pass |

* **Log In Module**

***Table 2. Using Test Cases for ‘Log In’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Log In page initiation | 1. Launch the web application. | The view is Log In page once launching application. | Pass |
| Field required validation | 1. Leave “Email or Username” or “Password” field empty. 2. Click “Login” button to login to the system. | Users will be alerted by a warning line in the sidebar that “Error: please enter both email and password.” in the sidebar. | Pass |
| Users Log In | 1. Fill in email or username and password that are registered in the system. 2. Click “Login” button to login to the system. | Users successful logging in and navigate to prediction page. | Pass |

* **Prediction Module**

***Table 3. Using Test Cases for ‘Prediction’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Search tab navigation | 1. Click on the tab menu which contains a prediction shape icon and stated “Predictions”. | The view navigates to Predictions tab. | Pass |
| Loading the page with default setting | 1. Default set date & time. 2. Default stock ticker (AAPL). | Returns all columns with plots including the predictions. | Pass |
| Search stock with stock symbol (stock name) in capital letter. | 1. Enter stock ticker. 2. Update Start & End Date. | Stocks Predictions shown in the tab with different analyzing the dataset. | Pass |
| Moving Average analysis for 10, 50 & 100 days + Daily return percentage. | 1. Enter stock ticker (the previous one, only one time) 2. Update Start & End Date (the previous one, only one time). | Returns the data in plot figure for “MA for 10 days” etc. | Pass |
| Training the Model | 1. Getting the input from previous setting the date and ticker | Returns the TRAIN & VAL Loss, TRAIN - MSE & VAL - MSE, TRAIN - MAE & VAL - MAE in figure plot. | Pass |
| Testing the Model | 1. Getting the input from previous setting the date and ticker | EDA for visualizing the collecting data | Pass |
| EDA for Heat Maps | 1. Getting the input from previous setting the date and ticker | Displaying the relationship between the features of the data (During COVID-19) & (Before COVID-19). | Pass |

* **Stocks Info. Module**

***Table 4. Using Test Cases for ‘Stocks Info.’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Stocks Info. tab navigation | 1. Click on the tab menu which contains a magnifying glass shape icon and stated “Stocks Info.” | The view navigates to Stocks Info tab. | Pass |
| Search stock with stock symbol (stock name) in capital letter. | 1. Type in specific stock name in capital letter. 2. Click “GO” button to search the stocks. | Searched stocks related to the stock name shown in the following order with Daily closing price, last closing price, daily volume for searched stocks. | Pass |
| Search stock with stock symbol (stock name) in small letter. | 1. Type in specific stock name in capital letter. 2. Click “GO” button to search the stocks. | Searched stocks related to the stock name shown in the following order with Daily closing price, last closing price, daily volume for searched stocks | Pass |
| Stock Actions | 1. Place the check mark in the stock action checkbox. | Displaying the results for stocks actions about searched stocks. | Pass |
| Institutional Shareholders | 1. Place the check mark in the Institutional Shareholders checkbox. | Displaying the results for Institutional shareholders about searched stocks. | Pass |
| Balance Sheet | 1. Place the check mark in the Balance Sheet checkbox. | Displaying the results for Balance Sheet about searched stocks. | Pass |
| Cash Flow | 1. Place the check mark in the Cash Flow checkbox. | Displaying the results for Cash Flow about searched stocks. | Pass |

* **Time Series Module**

***Table 5. Using Test Cases for ‘Time Series info.’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Time series tab navigation | 1. Click on the tab menu which contains a information shape icon and stated “Time Series” | The view navigates to Time Series tab. | Pass |
| Search stock with stock symbol (stock name) in capital letter. | 1. Type in specific stock name in capital letter. | Searched stocks related to the stock name shown in the search section | Pass |
| Search stock with stock symbol (stock name) in small letter. | 1. Type in specific stock name in capital letter. | Searched stocks related to the stock name shown in the search section | Pass |
| Start Date | 1. Tap on text input widget named as start date. | Displaying the calendar for user to select start date | Pass |
| End Date | 1. Tap on text input widget named as end date. | Displaying the calendar for user to select end date | Pass |
| Number of Days | 1. Enter the number of days you want to predict. | Displaying the days that user enters | Pass |
| Submit Button | 1. Click “Submit” button in order to get the result. | Displaying the predicted results for searched ticker. | Pass |

* **Stocks News Module**

***Table 6. Using Test Cases for ‘Stocks News’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Stocks News tab navigation | 1. Click on the tab menu which contains news shape icon and stated “Stocks News” | The view navigates to Stocks News tab. | Pass |
| Search stock with stock symbol (stock name) in capital letter. | 1. Type in specific stock name in capital letter. 2. Click “Get News!” button to get the news about Stocks. | Searched stocks related to the stock name shown in the following order with news | Pass |
| Search Stocks without keyword | 1. Leave empty for the field 2. Click “Get News!” button to get the news. | “No news for this stock” should be displayed. | Pass |
| Search stock with stock symbol (stock name) in small letter. | 1. Type in specific stock name in small letter. 2. Click “Get News!” button to get the news about Stocks. | Searched stocks related to the stock name shown in the following order with news | Pass |

* **Help**

***Table 7. Using Test Cases for ‘Help’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Help tab navigation | 1. Click on the tab menu which contains question mark shape icon and stated “Help”. 2. Some useful instructions for use the application are displayed. 3. A helpful video showing how to use the application is displayed. | The view switches to the About Us tab, where the contents of the tabs are seen. | Pass |

* **About Us Module**

***Table 8. Using Test Cases for ‘About Us’ Module***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| About Us tab navigation | 1. Click on the tab menu which contains information shape icon and stated “About Us”. 2. There's some information displayed regarding the application's author. 3. Displayed contact form for feedback or any other application-related queries. | The view switches to the About Us tab, where the contents of the tabs are seen. | Pass |

* + - 1. **System Testing**

***Table 9. Test Case for System Testing***

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Test Execution Steps** | **Expected Results** | **Results (Pass/Fail)** |
| Log In to predict stocks prices | 1. Fill in the credentials in Log In page. 2. Click on the “Log In” button on Log In page. 3. Click on the tab menu which contains a prediction shape icon and stated “Predictions”. 4. Click on the tab menu which contains a magnifying glass shape icon and stated “Stocks Info.” 5. Select the "Time Series" tab from the tab menu that has an information-shape icon. 6. Click on the tab menu which contains a news shape icon and stated “Stocks News”. 7. Type in specific stock name in capital or small letter. 8. Click on the tab menu which contains a question mark shape icon and stated “Help”. 9. Click on the tab menu which contains a information shape icon and stated “About Us”. | The view navigates to the user's preferred tab to obtain predictions, stock information, time series analysis, and stock news, among other features. | Pass |

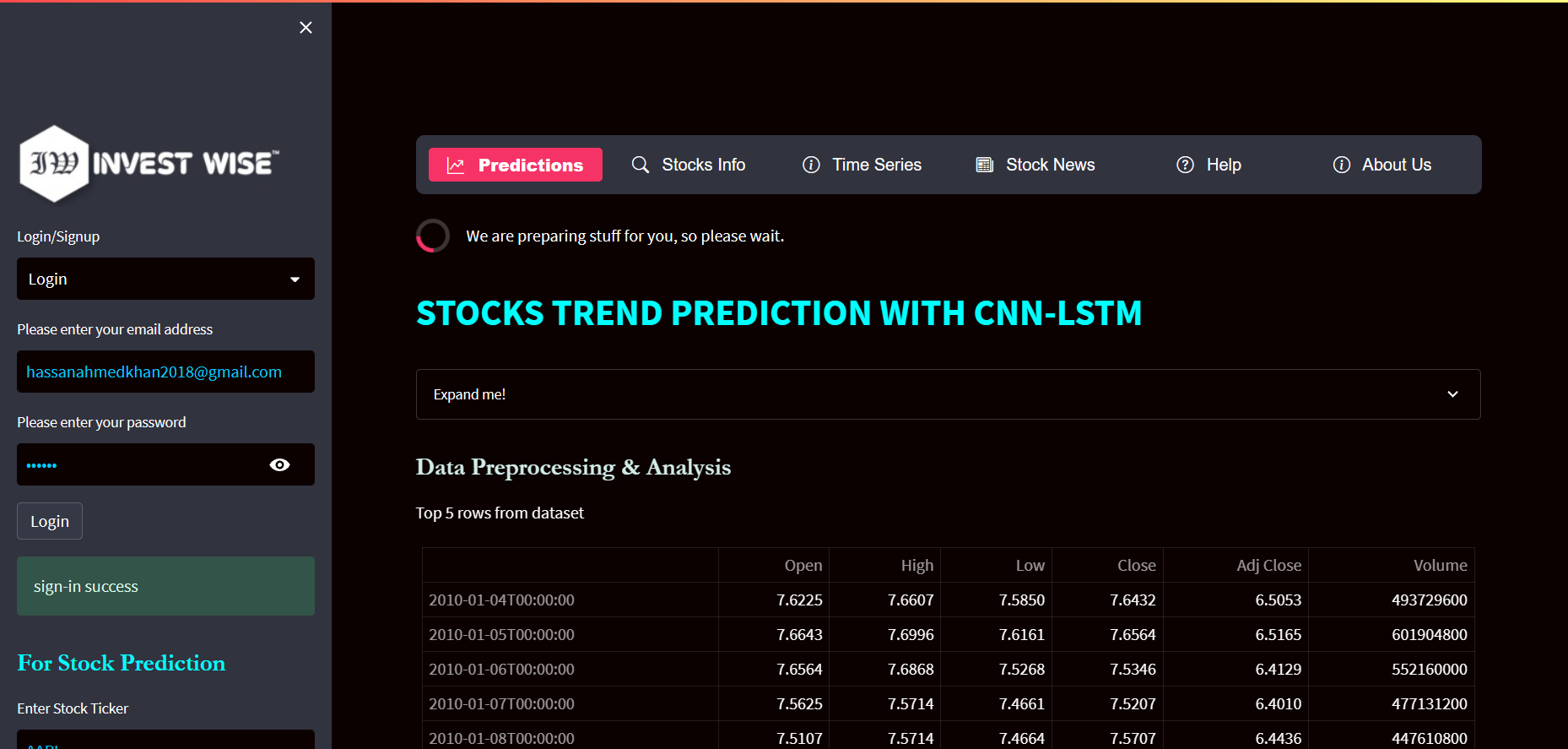
* 1. **Verification and Validation**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Inputs** | **Outputs** |
| **Initiation** | User Requirements  User constraints  Acceptance Criteria such as investors, trader, financial experts. | Software Requirement Specification |
| **Planning** | Software Requirement Specification | Project Plan  Test plan  Constraints list  Phase Review |
| **Execution** | Project Plan  Test plan  Constraints list  Phase Review | Deliverable Product  Test Results  Phase Review |
| **Evaluation** | Deliverable Product  Test Results  Phase Review | Release of deliverable product  User documentation  Final review report |

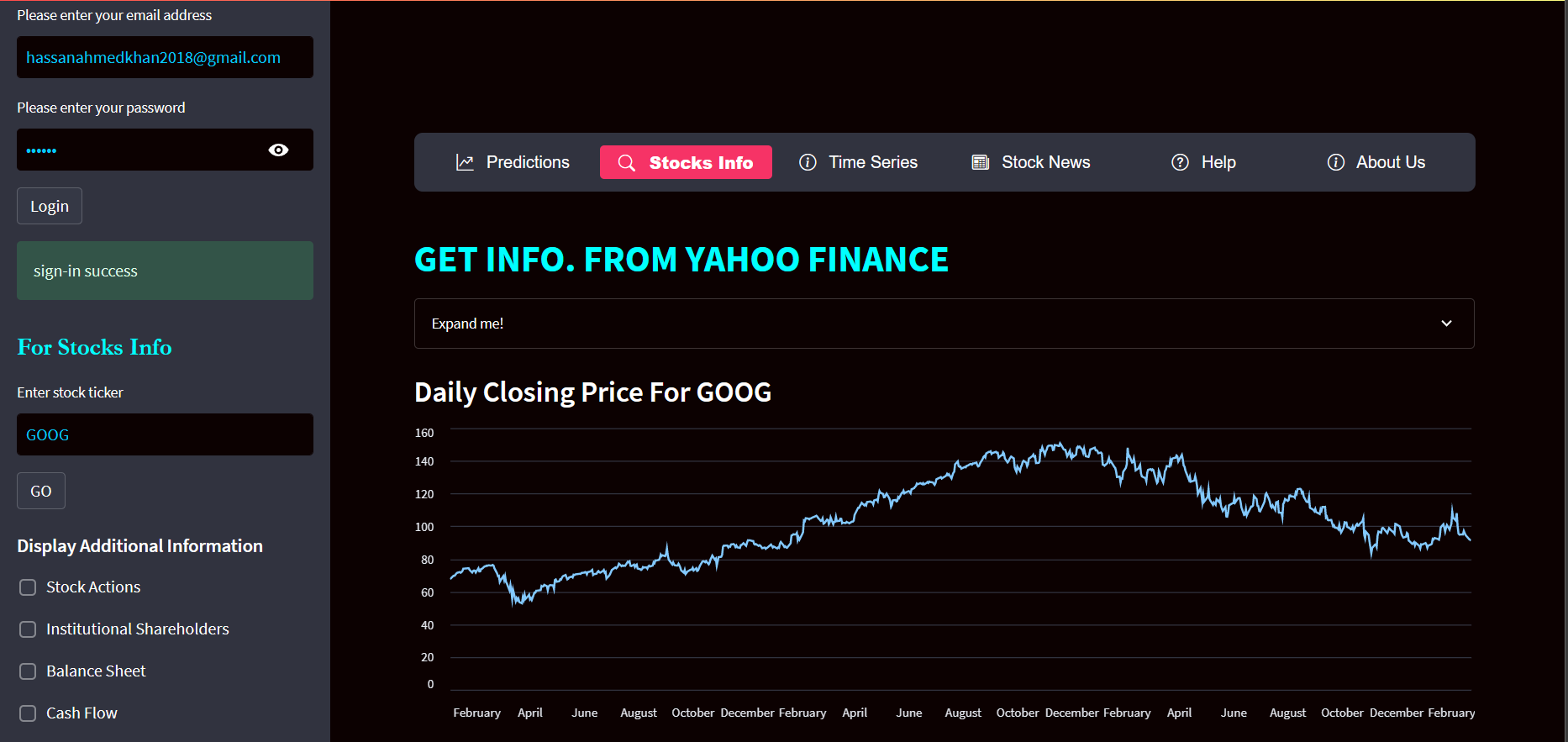
* 1. **Product Functionality**
* Maximum price received during the selected years or months or days.
* Minimum price received during the selected years or months or days.
* Average price received during the selected years or months or days.
* Standard deviation of prices received during the selected years or months or days.
* Slope of linear model applied on tick data of the respective years or months or days, which gives an idea about the trend during the last hour.
* EDA (Exploratory Data Analysis) for the searched stocks.
* Predicted Result
* Testing the Models with different Datasets, again Prediction of data
* Some EDA’S for before & after the COVID-19 pandemic.
* Searching for stocks with volume of data, institutional shareholders, annual balance sheet, Cash flow, Income statements etc.
* Time Series Prediction
* Stocks News for investing in stock markets.
* User must select the ticker (stocks symbol), starting and closing date in order to get prediction for days, months and years etc.
* New users can find instructions and an introduction video under the help tab.
* The user can submit the contact form under the "about us" tab to provide comments and learn more about the author of the application.

1. **Specific Requirements**
   1. **External Interface Requirement**
      1. **User Interface**

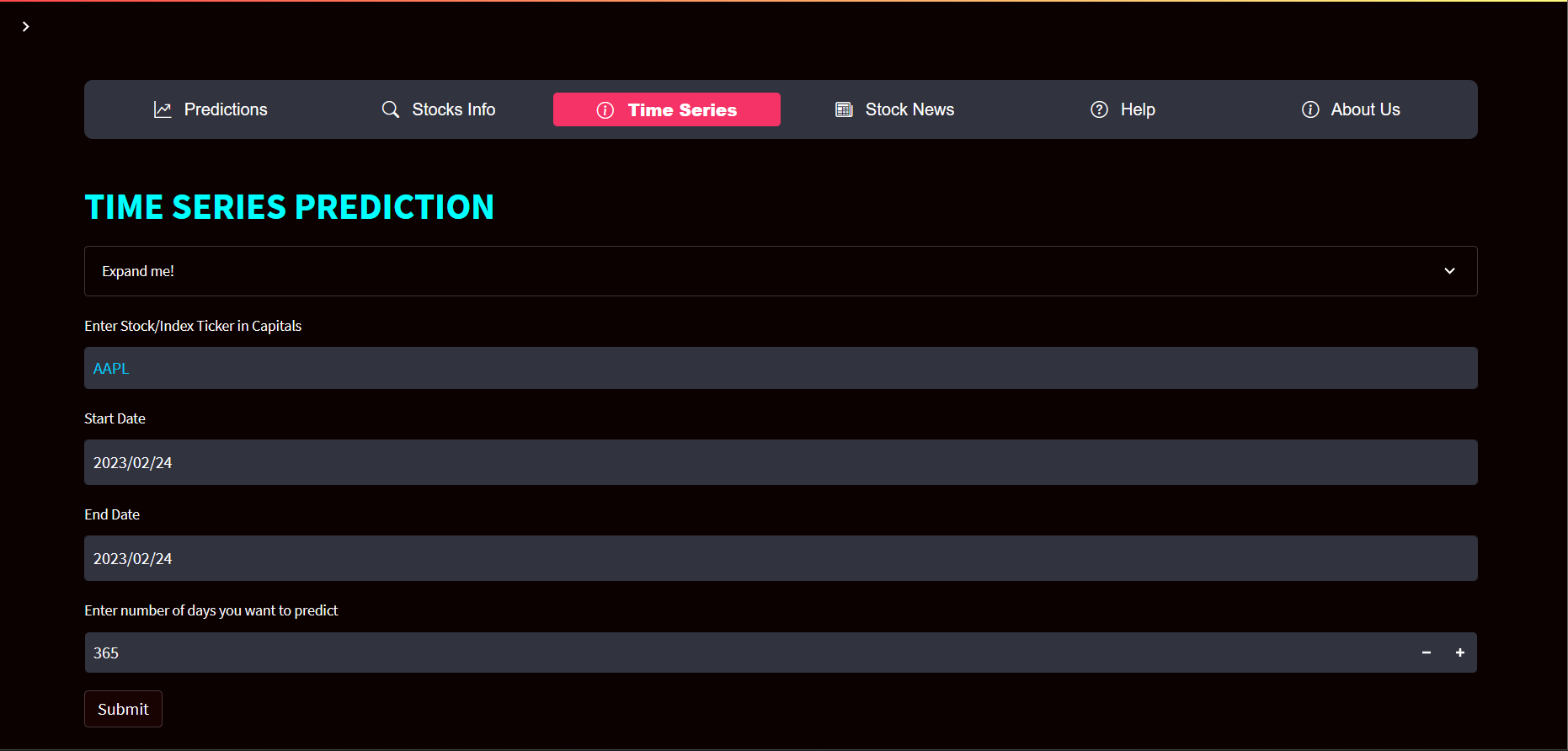
|  |
| --- |
| **Tab 1** |

****

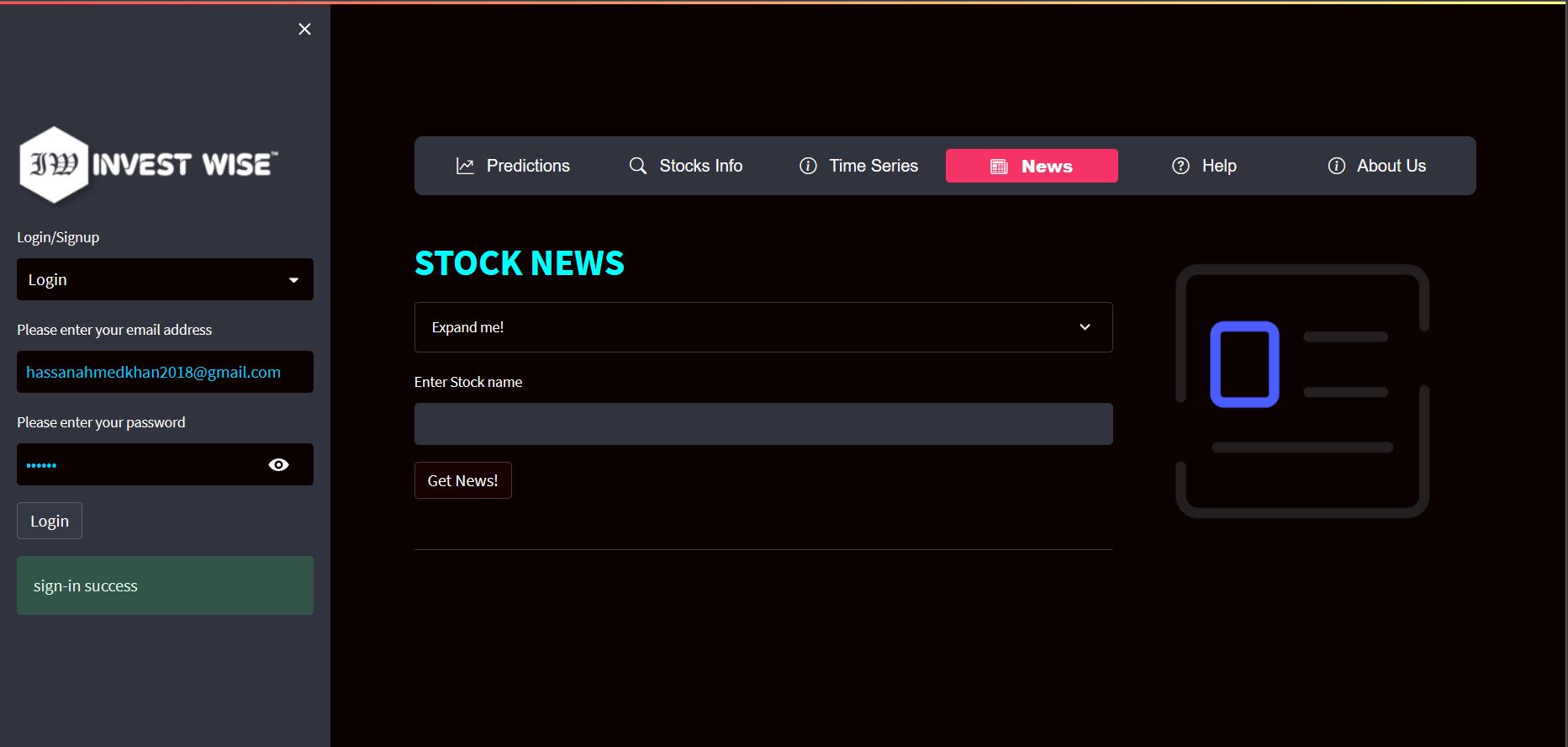
|  |
| --- |
| **Tab 2** |

****

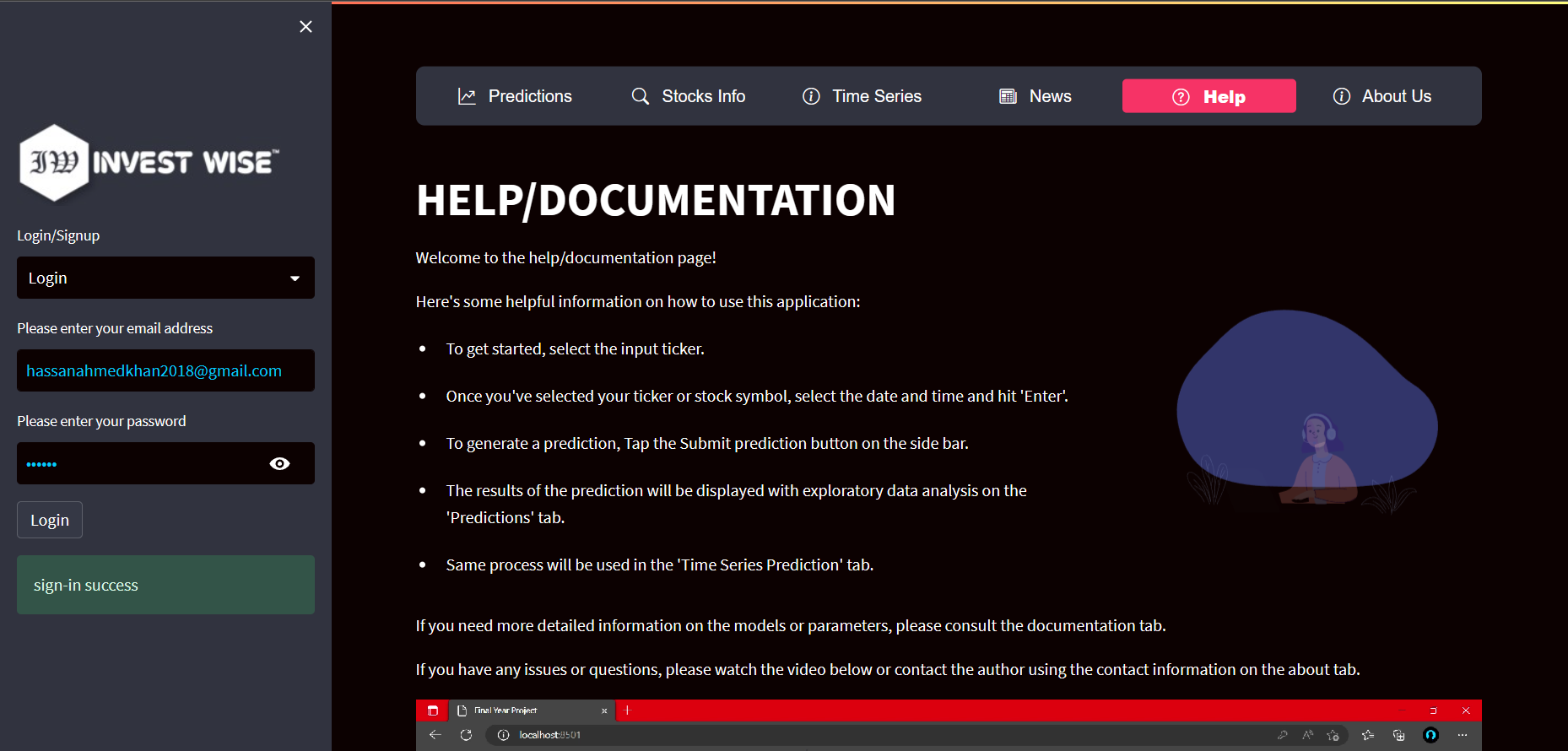
|  |
| --- |
| **Tab 3** |

****

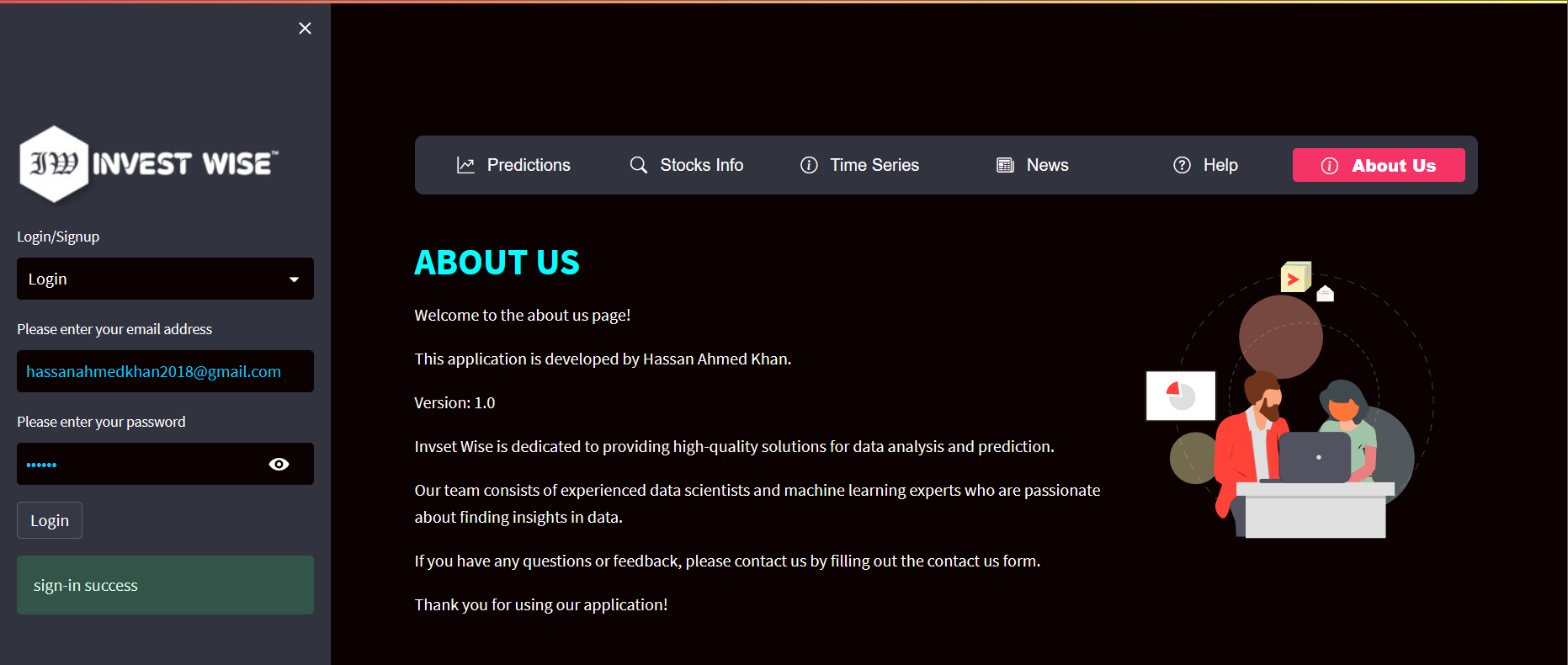
|  |
| --- |
| **Tab 4** |

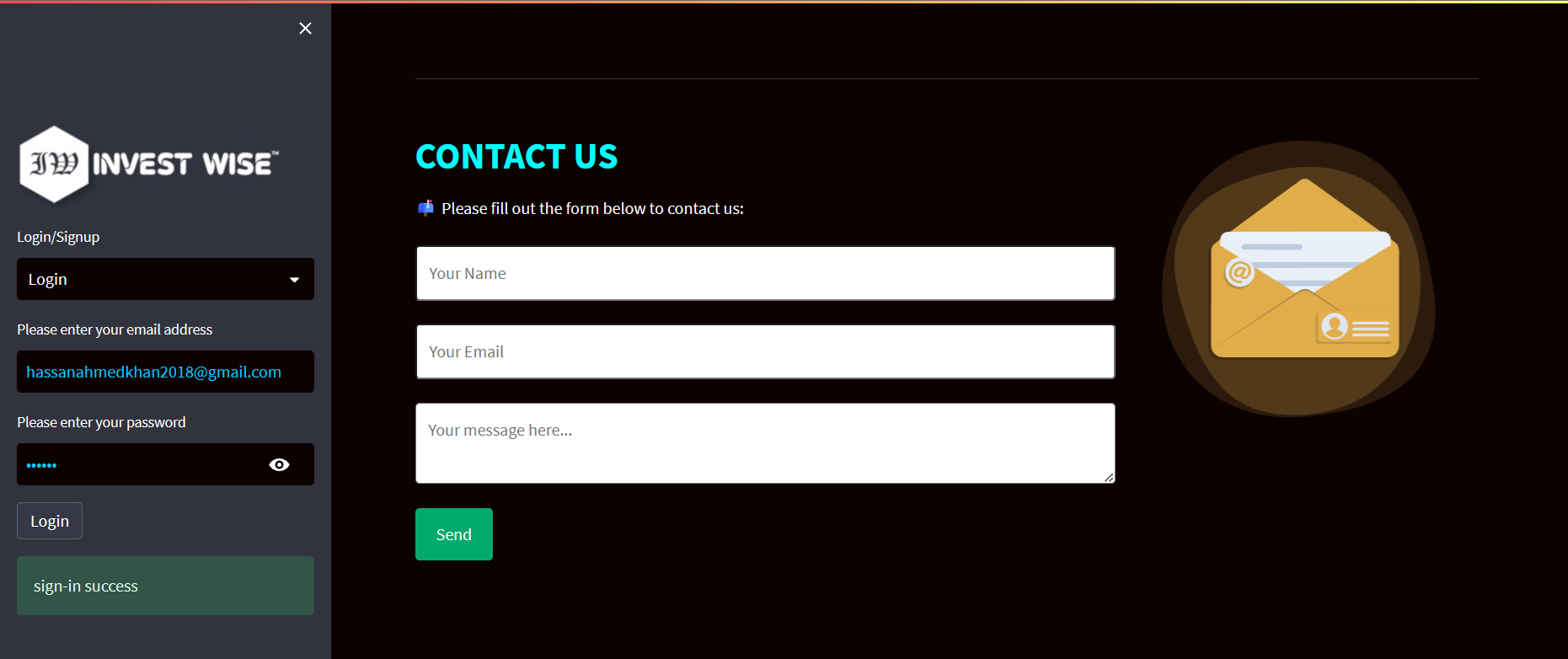
****

|  |
| --- |
| **Tab 5** |

****

|  |
| --- |
| **Tab 6** |

****

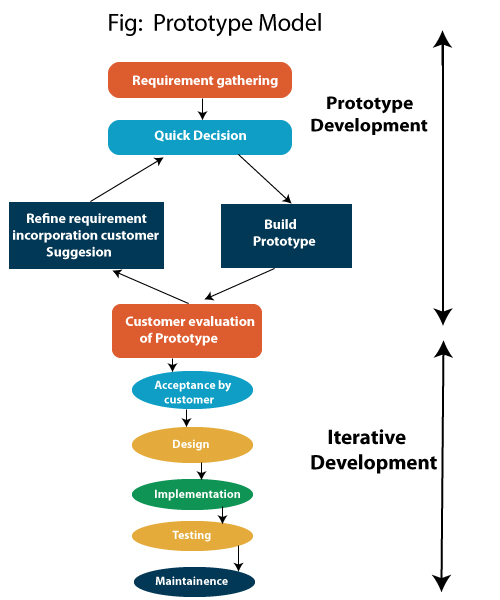
****

* + 1. **Software Interface**
* The web application is connected with the yahoo finance website by using its API.
* The web application is developed for windows 7, windows 8, windows 10, windows 11 etc.
* The connection of web application with other libraries are as following:
  + Numpy
  + Pandas
  + Sickit-learn
  + Pandas\_datareader
  + Cufflinks
  + Seaborn
  + Yfinance
  + Keras
  + Prophet
  + Gnews
  + Tensorflow
  + Alpha-vantage
  + Matplotlib
  + Streamlit Lottiefiles
    1. **Communication Interface**
* Google chrome or different web browser have different versions. Our website must work ok with all versions or just on a new one.
* Network server communication protocols HTTP, HTTPS, or FTP.
  1. **Software Process Model**

Developing a system requires involves not only code writing but also many other activities such as requirement gathering, document writing, testing and others. System development is not an easy task as all the activities have to be done in a well-planned manner. Hence, the development methodologies are guidelines or blueprint to develop the system effectively and efficiently.

**The Prototype Model will be used here:**

The prototype model requires the involvement of users or clients in the early phases, such as the requirement gathering phase and design phase, as this can help to develop the best design before implementation. The prototype model is suitable for projects with unclear requirements as clients can add the requirements during the design phase. Since the prototype model will keep building the prototype until the prototype accepted by users and customers, it is best to be used in the projects that have less understanding of specifications so that the development team can alter the requirements from time to time.



* 1. **Functional & Non-Functional Requirement**

**Functional Requirements:**

* The system shall allow users to search the about the stocks in the Malaysian stocks by entering stock’s name and stock’s code.
* The system shall be able to extract historical stock prices for prediction purposes from another stock portal.
* The system shall be able to predict future stocks prices by using time series algorithms.
* The system shall be able to view the predicted stock prices to the users.
* The system shall be able to display the predicted stock prices charts to the users.
* The system shall allow users to get news on searched ticker.

**Non - Functional Requirements:**

* The system shall be secure for the users to use.
* The system does not contain any critical data.
* The databases that are accessed are locally executed.
  1. **Tools & Techniques**

Development Tools:

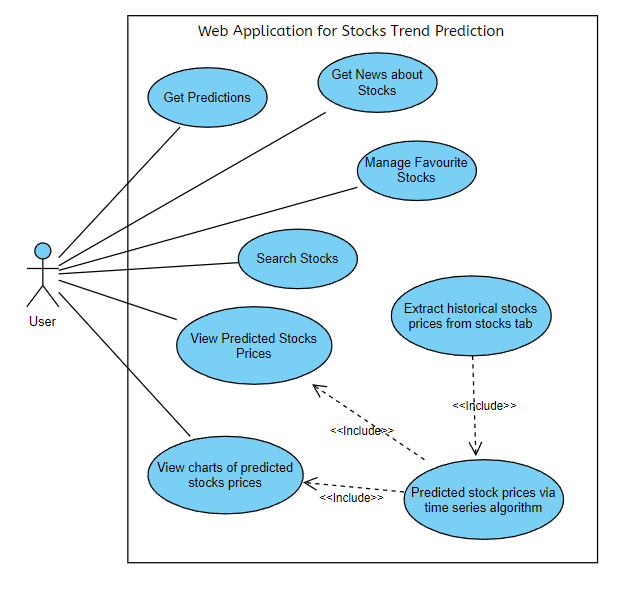
* Python
* Anaconda Navigator
* Visual Studio
* Streamlit
* Jupyter Notebook
  1. **Required Hardware & Software with Specification**
     1. **Software**

|  |  |
| --- | --- |
| Operating system: | Windows, Linux & Mac OS |
| Data | .csv file |
| Set Server : | Web Server with HTTP, HTTPS etc. |

* + 1. **Hardware**

|  |  |
| --- | --- |
| Processor : | Intel i5 or above |
| Ram: | Minimum 225MB or more. |
| Hard Disk : | Minimum 2 GB of space |
| Input Device : | Keyboard |
| Output Device : | Screens of Monitor or a Laptop |

* 1. **Behavior Requirement**
     1. **Use Case View**

****

* + 1. **Use Case Description:**

***Table 6: Use Case description for “Stocks Prediction”***

|  |  |  |
| --- | --- | --- |
| Use Case Name: Predict Stock Prices with CNN - LSTM | Use Case ID: 1 | Importance Level: High |
| Primary Actor: - | Use Case Type: Detailed, Essential | |
| Stakeholders and interests:  User – able to know the stock prices using CNN - LSTM algorithm | | |
| Brief Description: This use case description describes how the system predicts the future stock prices using CNN – LSTM algorithm. | | |
| Trigger: User wants to get the predicted stock prices. | | |
| Relationship:  Association: User  Include: View Predicted Stock Prices  View Charts of Predicted Stock Prices  Extend: -  Generalization: - | | |
| Normal flow of events:  1. A few time series algorithms will be used to test the best model for stock price prediction.  2. Those time series algorithms will go through evaluation method to find the best and most accurate model.  3. The model is then used to predict the future stock prices i.e. CNN-LSTM.  4. The input data will be the historical stock data retrieve from database.  5. The predicted stock prices will be sent to the web application for stock price prediction. | | |
| Sub flows: None | | |
| Alternative/exceptional flows: None | | |

***Table 7: Use Case description for “Stocks Info”***

|  |  |  |
| --- | --- | --- |
| Use Case Name: Stocks Info | Use Case ID: 2 | Importance Level: High |
| Primary Actor: - User | Use Case Type: Detailed, Essential | |
| Stakeholders and interests:  User – wants to find stocks which are new to them. | | |
| Brief Description: This use case description describes how user search the stocks | | |
| Trigger: User wants to get the predicted stock prices. | | |
| Relationship:  Association: User  Include: -  Extend: -  Generalization: - | | |
| Normal flow of events:  1. User found some new stocks from stocks info tab.  2. User click on the magnifying glass icon on the menu bar to navigate to search page.  3. User enters the details of the stocks on the search bar.  4. User clicks on the “GO” button to search for the stocks.  5. The stocks which related to the stocks’ details entered will be shown on the search page | | |
| Sub flows:  3.1 User enters the name of the stocks that wanted to be searched.  3.2 User check mark for additional information of the stocks that wanted to be searched. | | |
| Alternative/exceptional flows:  3.1 The name or code of the stocks that wanted to be searched could not be found.  3.2 The use case ended. | | |

***Table 8: Use Case description for “Time Series Prediction”***

|  |  |  |
| --- | --- | --- |
| Use Case Name: Predict Stock Prices with time series algorithm. | Use Case ID: 3 | Importance Level: High |
| Primary Actor: - | Use Case Type: Detailed, Essential | |
| Stakeholders and interests:  User – able to know the stock prices using Prophet algorithm | | |
| Brief Description: This use case description describes how the system predicts the future stock prices using Prophet algorithm. | | |
| Trigger: User wants to get the predicted stock prices. | | |
| Relationship:  Association: User  Include: View Predicted Stock Prices  View Charts of Predicted Stock Prices  Extend: -  Generalization: - | | |
| Normal flow of events:  1. A few time series algorithms will be used to test the best model for stock price prediction.  2. Those time series algorithms will go through evaluation method to find the best and most accurate model.  3. The model is then used to predict the future stock prices i.e. Prophet.  4. The input data will be the historical stock data retrieve from database.  5. The predicted stock prices will be sent to the web application for stock price prediction. | | |
| Sub flows: None | | |
| Alternative/exceptional flows: None | | |

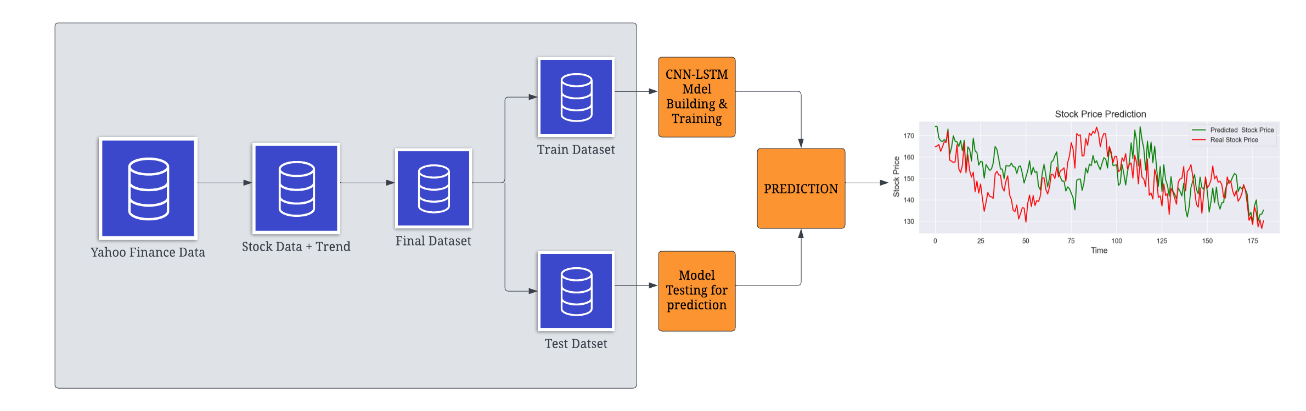
***Table 9: Use Case description for “View charts of stocks prediction”***

|  |  |  |
| --- | --- | --- |
| Use Case Name: View charts of predicted stocks prices | Use Case ID: 4 | Importance Level: High |
| Primary Actor: - | Use Case Type: Detailed, Essential | |
| Stakeholders and interests:  User – able to view the predicted stock prices in an easier way which is in charts form. | | |
| Brief Description: This use case description describes how user view the predicted stock prices in a chart view. | | |
| Trigger: User wants to view the charts of predicted stock prices. | | |
| Relationship:  Association: User  Include: -  Extend: -  Generalization: - | | |
| Normal flow of events:  1. User wants to know the predicted stock prices of certain stocks in chart form.  2. User search the stocks from the system.  3. User click on the “Submit” button to predict the stock prices.  4. The predicted stock prices in a chart form appear on the page for the user. | | |
| Sub flows: None | | |
| Alternative/exceptional flows: None | | |

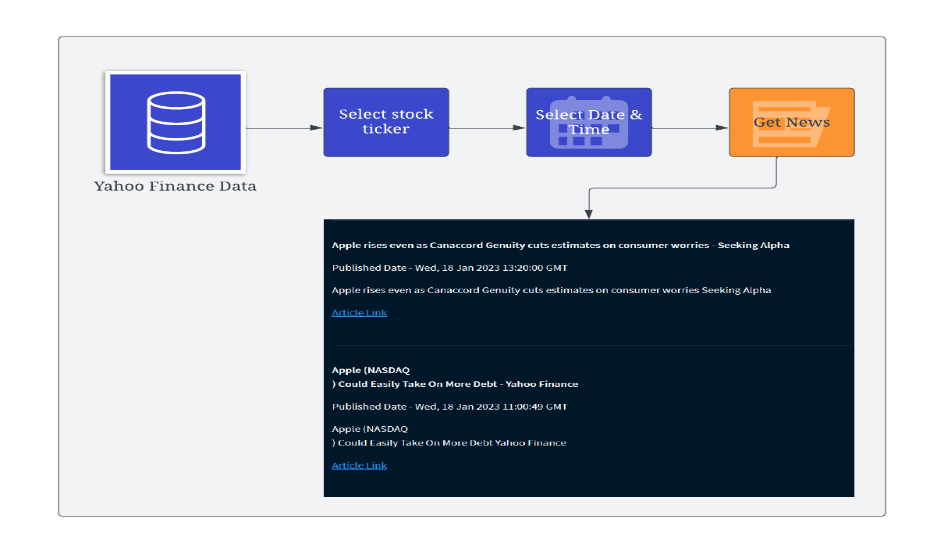
***Table 10: Use Case description for “Stocks News”***

|  |  |  |
| --- | --- | --- |
| Use Case Name: Get news for searched ticker | Use Case ID: 5 | Importance Level: High |
| Primary Actor: - User | Use Case Type: Detailed, Essential | |
| Stakeholders and interests:  User – able to view the stock news in an easier way which is in textual form. | | |
| Brief Description: This use case description describes how user get the news on searched ticker. | | |
| Trigger: User wants to view the stocks news | | |
| Relationship:  Association: User  Include: -  Extend: -  Generalization: - | | |
| Normal flow of events:  1. User wants to know the predicted stock prices of certain stocks in chart form.  2. User search the stocks from the system.  3. User click on the “Get News” button to predict the stock prices.  4. The predicted stock prices in a chart form appear on the page for the user. | | |
| Sub flows: None | | |
| Alternative/exceptional flows: None | | |

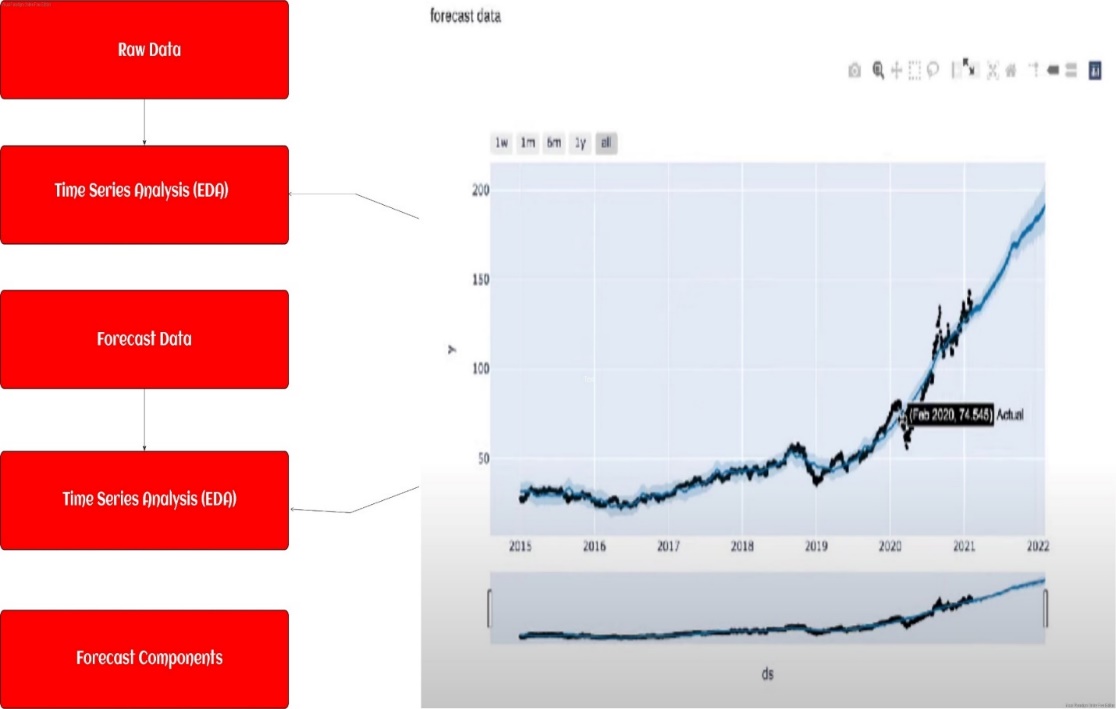
* + 1. **High Level Design**
       1. **System Architecture**

****

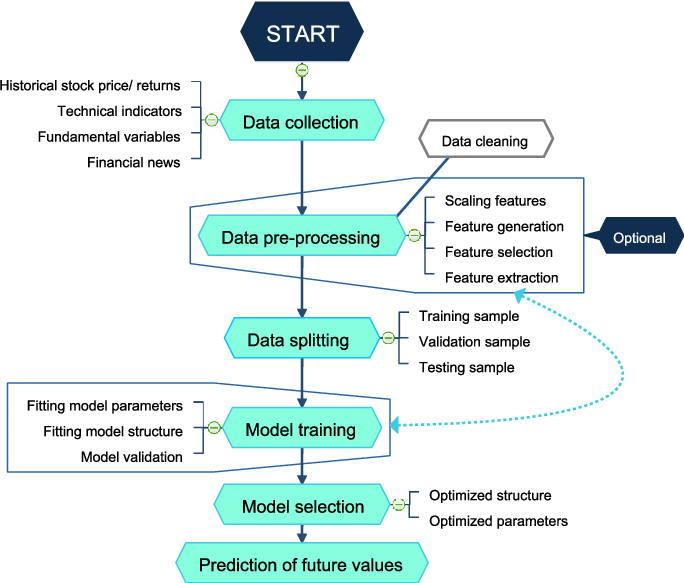
|  |  |
| --- | --- |
| https://documents.lucid.app/documents/83fd8058-3b6f-4b85-b73e-552704e34ede/pages/0_0?a=1399&x=112&y=629&w=1496&h=682&store=1&accept=image%2F*&auth=LCA%2068a64592ca16b8b9aa08e345aaecfd88d3f3d88c-ts%3D1674048154 | https://documents.lucid.app/documents/83fd8058-3b6f-4b85-b73e-552704e34ede/pages/0_0?a=1399&x=158&y=1263&w=1364&h=814&store=1&accept=image%2F*&auth=LCA%2018da220d20d34cd2c6eeadb64b5c4228bcb1a51b-ts%3D1674048154 |



* + - 1. **Web Application Framework**

****

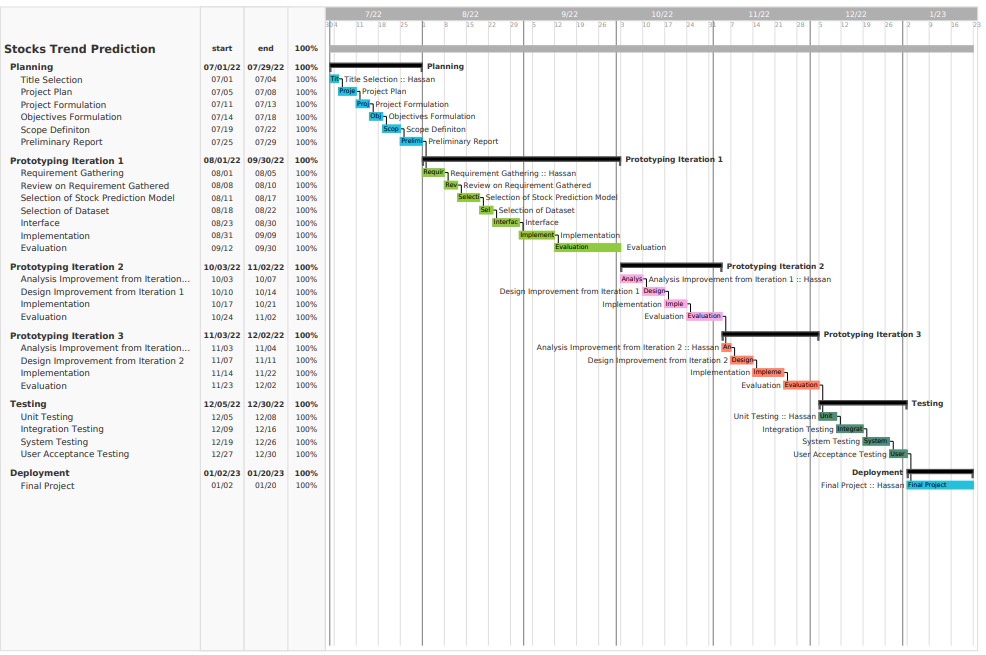
* + - 1. **Work Flow of Stocks Trend Prediction**



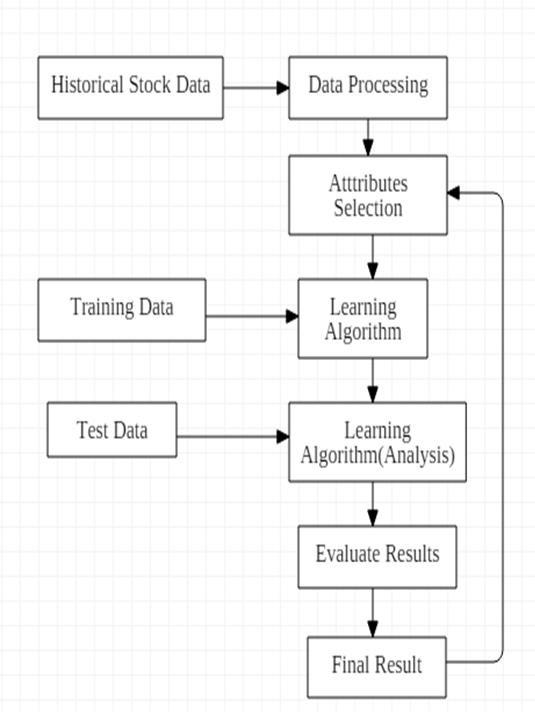
1. **Project Planning**
   1. **Distribution Of Work (Project Management)**

|  |  |
| --- | --- |
| **Task** | **Hassan Ahmed Khan** |
| Do the Literature Survey |  |
| Analyze Social Networks |  |
| Design Data Techniques |  |
| Design the Database |  |
| Design Data Algorithms |  |
| Design the User Interface |  |
| Develop the Data |  |
| Build the Database |  |
| Develop the Data Algorithms |  |
| Build the User Interface |  |
| Test the Web Application |  |
| Test the Database |  |
| Test the Data Algorithms |  |
| Test the User Interface |  |
| Perform Unit & System Testing |  |
| Write the Proposal |  |
| Write the Monthly Reports |  |
| Write the Progress Report |  |
| Write the Final Report |  |
| Prepare for the Presentation |  |
| Design the Project Poster |  |

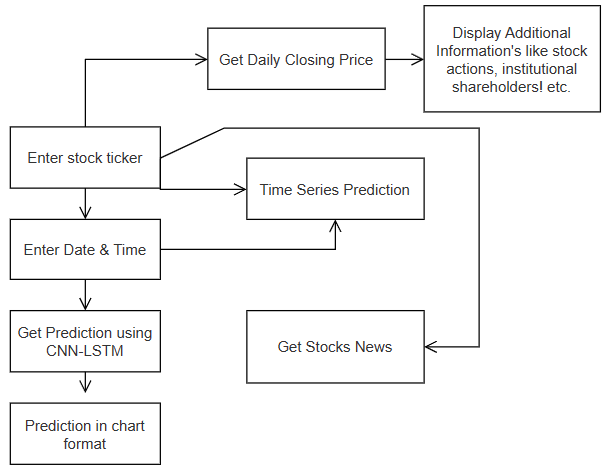
* 1. **Gantt Chart**

****

* 1. **Back-End Development**
     1. **Block Diagram for back-end development**



* 1. **Front-End Development**
     1. **Block Diagram for front-end development**

****

1. **Program Coding**
   1. **Results Coding**

Below, just the main modules have been covered.

**Setting up the dataset:**

# importing libraries

import tensorflow as tf

from pandas.plotting import autocorrelation\_plot

import datetime as dt

import math

import os

import pandas\_datareader as dr

import cufflinks as cf

import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import yfinance as yahooFinance

import streamlit as st

from sklearn.linear\_model import LinearRegression, Ridge, Lasso

from sklearn.model\_selection import train\_test\_split, GridSearchCV, RandomizedSearchCV

from sklearn.metrics import r2\_score, mean\_squared\_error

from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor

from datetime import datetime

from streamlit\_option\_menu import option\_menu

from warnings import simplefilter

simplefilter(action='ignore', category=FutureWarning)

from alpha\_vantage.fundamentaldata import FundamentalData

from sklearn.metrics import max\_error

from sklearn.metrics import r2\_score

from sklearn.metrics import explained\_variance\_score, mean\_poisson\_deviance, mean\_gamma\_deviance

from sklearn.tree import DecisionTreeRegressor

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from tensorflow.keras.metrics import RootMeanSquaredError

from tensorflow.keras.metrics import Accuracy

from tensorflow.keras.regularizers import L1, L2

from tensorflow.keras.layers import MaxPooling1D, Flatten

from tensorflow.keras.layers import Conv1D, LSTM, Dense, Dropout, Bidirectional, TimeDistributed

from prophet import Prophet

from prophet.plot import plot\_plotly

from gnews import GNews

    # Describing the data

    st.write('Top 5 rows from dataset')

    st.table(data.head())

    st.write('Last 5 rows from dataset')

    st.table(data.tail())

    st.write('Statistical description of dataframe')

    st.table(data.describe())

    st.write("Analyzing The Dataset")

    st.line\_chart(data)

    st.line\_chart(data['Close'])

    st.line\_chart(data['Open'])

    st.line\_chart(data['High'])

    st.line\_chart(data['Low'])

    data.isnull().sum()

    data.reset\_index(drop=True, inplace=True)

    data.fillna(data.mean(), inplace=True)

    # Monthly Average Data

    ma\_day = [10, 50, 100]

    for ma in ma\_day:

        column\_name = "MA for %s days" % (str(ma))

        data[column\_name] = pd.DataFrame.rolling(data['Close'], ma).mean()

    st.write("Daily Return Percentage")

    data['Daily Return'] = data['Close'].pct\_change()

    # plot the daily return percentage

    fig2 = plt.figure(figsize=(14, 5))

    st.line\_chart(data['Daily Return'])

    # st.pyplot(fig2)

    sns.displot(data['Daily Return'].dropna(), bins=100,

                color='green', height=4, aspect=3)

    st.pyplot()

    st.write('After filling the null values with mean...')

    data.reset\_index(drop=True, inplace=True)

    data.fillna(data.mean(), inplace=True)

    st.table(data.head())

    data.nunique()

    data.sort\_index(axis=1, ascending=True)

    cols\_plot = ['Open', 'High', 'Low', 'Close', 'Volume',

                 'MA for 10 days', 'MA for 50 days', 'MA for 100 days', 'Daily Return']

    axes = data[cols\_plot].plot(

        marker='.', alpha=0.7, linestyle='None', figsize=(12, 15), subplots=True)

    for ax in axes:

        ax.set\_ylabel('Daily trade')

    plt.plot(data['Close'], label="Close price")

    plt.xlabel("Timestamp")

    plt.ylabel("Closing price")

    st.pyplot()

    df = data

    df.describe().transpose()

**Training the dataset:**

from sklearn.model\_selection import train\_test\_split

    X = []

    Y = []

    window\_size = 100

    for i in range(1, len(df) - window\_size - 1, 1):

        first = df.iloc[i, 2]

        temp = []

        temp2 = []

        for j in range(window\_size):

            temp.append((df.iloc[i + j, 2] - first) / first)

        temp2.append((df.iloc[i + window\_size, 2] - first) / first)

        X.append(np.array(temp).reshape(100, 1))

        Y.append(np.array(temp2).reshape(1, 1))

    x\_train, x\_test, y\_train, y\_test = train\_test\_split(

        X, Y, test\_size=0.2, shuffle=True)

    train\_X = np.array(x\_train)

    test\_X = np.array(x\_test)

    train\_Y = np.array(y\_train)

    test\_Y = np.array(y\_test)

    train\_X = train\_X.reshape(train\_X.shape[0], 1, 100, 1)

    test\_X = test\_X.reshape(test\_X.shape[0], 1, 100, 1)

    #CNN and LSTM

    # For CNN, the layers are created with sizes 64,128,64 with kernel size = 3. In every layer, TimeDistributed function is added to track the features for every temporal slice of data with respect to time. In between, MaxPooling layers are added.

    # After that, it's passed to Bi-LSTM layers

    # For creating model and training

    model = tf.keras.Sequential()

    # Neural Network model

    # CNN layers

    model.add(TimeDistributed(Conv1D(64, kernel\_size=3,

                                     activation='relu', input\_shape=(None, 100, 1))))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Conv1D(128, kernel\_size=3, activation='relu')))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Conv1D(64, kernel\_size=3, activation='relu')))

    model.add(TimeDistributed(MaxPooling1D(2)))

    model.add(TimeDistributed(Flatten()))

    # LSTM layers

    model.add(Bidirectional(LSTM(100, return\_sequences=True)))

    model.add(Dropout(0.5))

    model.add(Bidirectional(LSTM(100, return\_sequences=False)))

    model.add(Dropout(0.5))

    # Final layers

    model.add(Dense(1, activation='linear'))

    model.compile(optimizer='adam', loss='mse', metrics=['mse', 'mae'])

    history = model.fit(train\_X, train\_Y, validation\_data=(

        test\_X, test\_Y), epochs=40, batch\_size=40, verbose=1, shuffle=True)

**Testing the dataset:**

# First we need to save a model

    # In this part, the model is saved and loaded back again. Then, it's made to train again but with different data to check it's loss and prediction

    model.save("model.h5")

    new\_model = tf.keras.models.load\_model("./model.h5")

    subheader3 = '''<div><h2 style = "color:#D0ECE7; font-size:23px; font-family: Garamond;">Testing The Model</h2></div>'''

    st.markdown(subheader3, unsafe\_allow\_html=True)

    st.write(

        'Testing the model but with with different dataset to check its loss and prediction')

    data2 = dr.data.get\_data\_yahoo(input, start='2019-01-01', end='2022-12-31')

    data2.reset\_index(drop=True, inplace=True)

    data2.fillna(data.mean(), inplace=True)

    st.table(data2.head())

    df2 = data2

    X = []

    Y = []

    window\_size = 100

    for i in range(1, len(df2) - window\_size - 1, 1):

        first = df2.iloc[i, 4]

        temp = []

        temp2 = []

        for j in range(window\_size):

            temp.append((df2.iloc[i + j, 4] - first) / first)

        # for j in range(week):

        temp2.append((df2.iloc[i + window\_size, 4] - first) / first)

        # X.append(np.array(stock.iloc[i:i+window\_size,4]).reshape(50,1))

        # Y.append(np.array(stock.iloc[i+window\_size,4]).reshape(1,1))

        # print(stock2.iloc[i:i+window\_size,4])

        X.append(np.array(temp).reshape(100, 1))

        Y.append(np.array(temp2).reshape(1, 1))

    x\_train, x\_test, y\_train, y\_test = train\_test\_split(

        X, Y, test\_size=0.2, shuffle=False)

    train\_X = np.array(x\_train)

    test\_X = np.array(x\_test)

    train\_Y = np.array(y\_train)

    test\_Y = np.array(y\_test)

    train\_X = train\_X.reshape(train\_X.shape[0], 1, 100, 1)

    test\_X = test\_X.reshape(test\_X.shape[0], 1, 100, 1)

    model.evaluate(test\_X, test\_Y)

    predicted = model.predict(test\_X)

    test\_label = test\_Y.reshape(-1, 1)

    predicted = np.array(predicted[:, 0]).reshape(-1, 1)

    len\_t = len(train\_X)

    for j in range(len\_t, len\_t + len(test\_X)):

        temp = data2.iloc[j, 3]

        test\_label[j - len\_t] = test\_label[j - len\_t] \* temp + temp

        predicted[j - len\_t] = predicted[j - len\_t] \* temp + temp

**Tab 2 Modules of Stocks Info.:**

def main():

        st.subheader("""Daily \*\*Closing Price\*\* For """ + selected\_stock)

        # get data on searched ticker

        stock\_data = yahooFinance.Ticker(selected\_stock)

        # get historical data for searched ticker

        stock\_df = stock\_data.history(

            period='1d', start='2020-01-01', end=None)

        # print line chart with daily closing prices for searched ticker

        st.line\_chart(stock\_df.Close)

        st.subheader("""Last \*\*closing Price\*\* For """ + selected\_stock)

        # define variable today

        today = datetime.today().strftime('%Y-%m-%d')

        # get current date data for searched ticker

        stock\_lastprice = stock\_data.history(

            period='1d', start=today, end=today)

        # get current date closing price for searched ticker

        last\_price = (stock\_lastprice.Close)

        # if market is closed on current date print that there is no data available

        if last\_price.empty == True:

            st.write("No data available at the moment")

        else:

            st.table(last\_price)

        # get daily volume for searched ticker

        st.subheader("""Daily \*\*Volume\*\* For """ + selected\_stock)

        st.line\_chart(stock\_df.Volume)

        # additional information feature in sidebar

        st.sidebar.subheader("""Display Additional Information""")

        # checkbox to display stock actions for the searched ticker

        actions = st.sidebar.checkbox("Stock Actions")

        if actions:

            st.subheader("""Stock \*\*Actions\*\* For """ + selected\_stock)

            display\_action = (stock\_data.actions)

            if display\_action.empty == True:

                st.write("No data available at the moment")

            else:

                st.table(display\_action)

        # checkbox to display quarterly financials for the searched ticker

        #financials = st.sidebar.checkbox("Quarterly Financials")

        # if financials:

        #    st.subheader("""\*\*Quarterly financials\*\* for """ + selected\_stock)

        #    display\_financials = (stock\_data.quarterly\_financials)

        #    if display\_financials.empty == True:

        #        st.write("No data available at the moment")

        #    else:

        #       st.write(display\_financials)

        # checkbox to display list of institutional shareholders for searched ticker

        major\_shareholders = st.sidebar.checkbox("Institutional Shareholders")

        if major\_shareholders:

            st.subheader(

                """\*\*Institutional Investors\*\* For """ + selected\_stock)

            display\_shareholders = (stock\_data.institutional\_holders)

            if display\_shareholders.empty == True:

                st.write("No data available at the moment")

            else:

                st.table(display\_shareholders)

        # checkbox to display quarterly balance sheet for searched ticker

        balance\_sheet = st.sidebar.checkbox("Balance Sheet")

        key = '6MK9ZOT10UNQ8MY0'

        fd = FundamentalData(key, output\_format='pandas')

        if balance\_sheet:

            st.subheader('Balance Sheet For ' + selected\_stock)

            balance\_sheet = fd.get\_balance\_sheet\_annual(balance\_sheet)[0]

            bs = balance\_sheet.T[2:]

            bs.columns = list(balance\_sheet.T.iloc[0])

            st.write(bs)

        # checkbox to display quarterly cashflow for searched ticker

        cashflow = st.sidebar.checkbox("Cash Flow")

        if cashflow:

            st.subheader('Cash Flow For ' + selected\_stock)

            cashflow = fd.get\_cash\_flow\_annual(cashflow)[0]

            cf = cashflow.T[2:]

            cf.columns = list(cashflow.T.iloc[0])

            st.write(cf)

**Tab 3 Modules of Time Series Prediction:**

if submit:

        # get data from yahoo

        df = dr.data.get\_data\_yahoo(company, start, end)

        # data preprocessing

        df = df.reset\_index()

        new\_df = df[['Date', 'Close']]

        new\_df = new\_df.rename(columns={'Date': 'ds', 'Close': 'y'})

        # initialize prophet model

        fp = Prophet(daily\_seasonality=True)

        fp.fit(new\_df)

        # make future predictions

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

        forecast = fp.predict(future)

        st.subheader('Predicted Result')

        # Plot the predictions

        fig = plot\_plotly(fp, forecast)

        fig.update\_xaxes(title\_text='Time')

        y\_text = '{company\_name} Stock price'.format(company\_name=company)

        fig.update\_yaxes(title\_text=y\_text)

        fig.update\_layout(

            autosize=False,

            width=720,

            height=400,)

        st.plotly\_chart(fig)

**Tab 4 Modules of News:**

user\_input = st.text\_input("Enter Stock name")

    state = st.button("Get News!")

    st.markdown("""---""")

    if state:

        news = GNews().get\_news(user\_input)

        if news:

            for i in news:

                st.markdown(f"\*\*{i['title']}\*\*")

                st.write(f"Published Date - {i['published date']}")

                st.write(i["description"])

                st.markdown(f"[Article Link]({i['url']})")

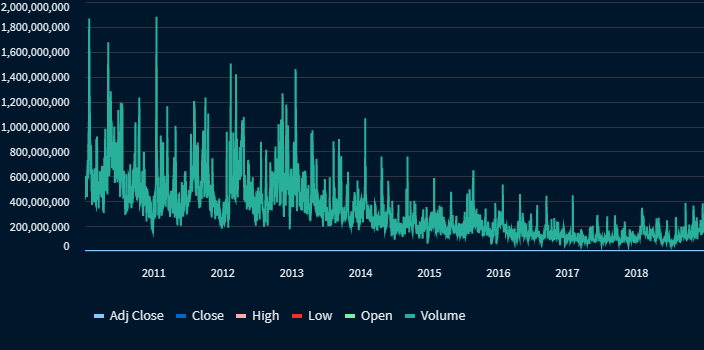
                st.markdown("""---""")

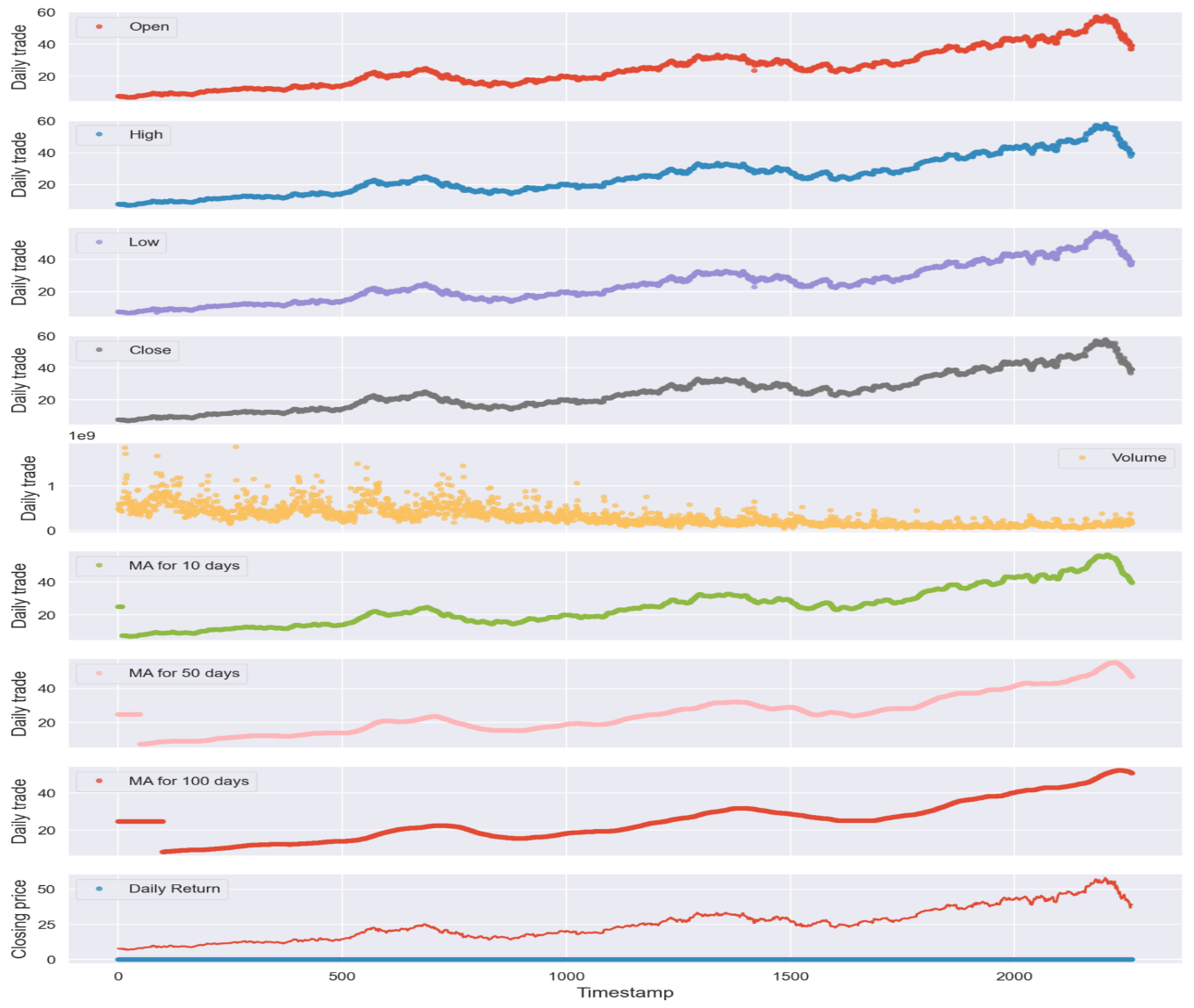
        else:

            st.write("No news for this stock")

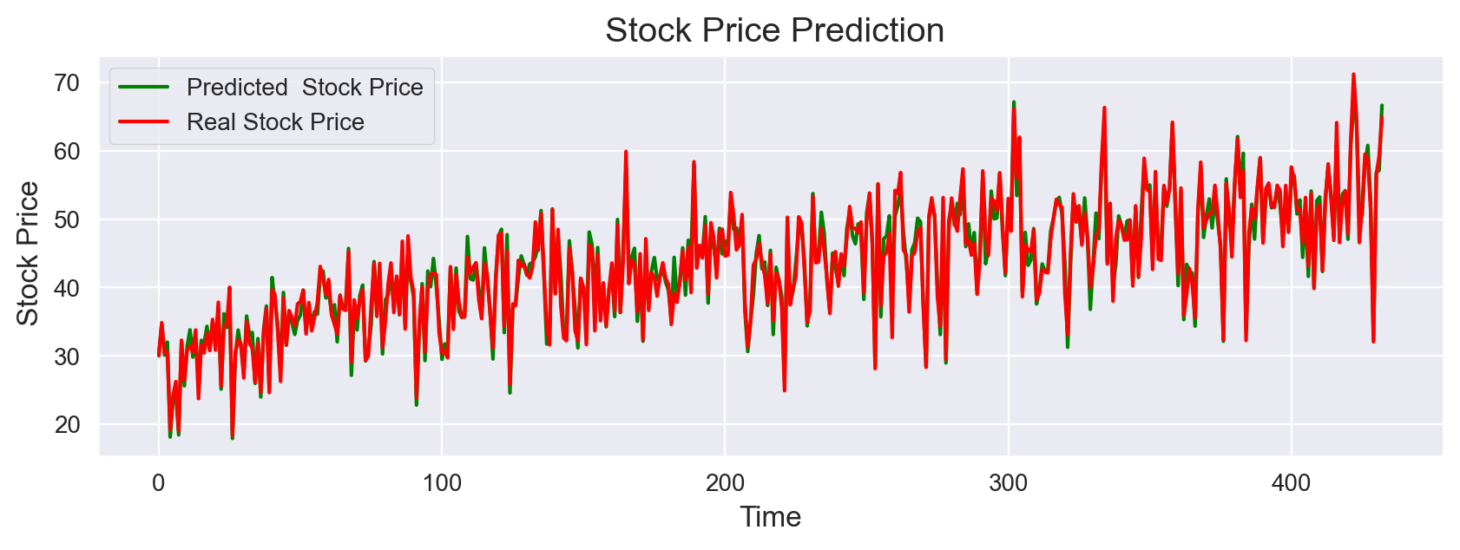
* 1. **Output Screenshots**

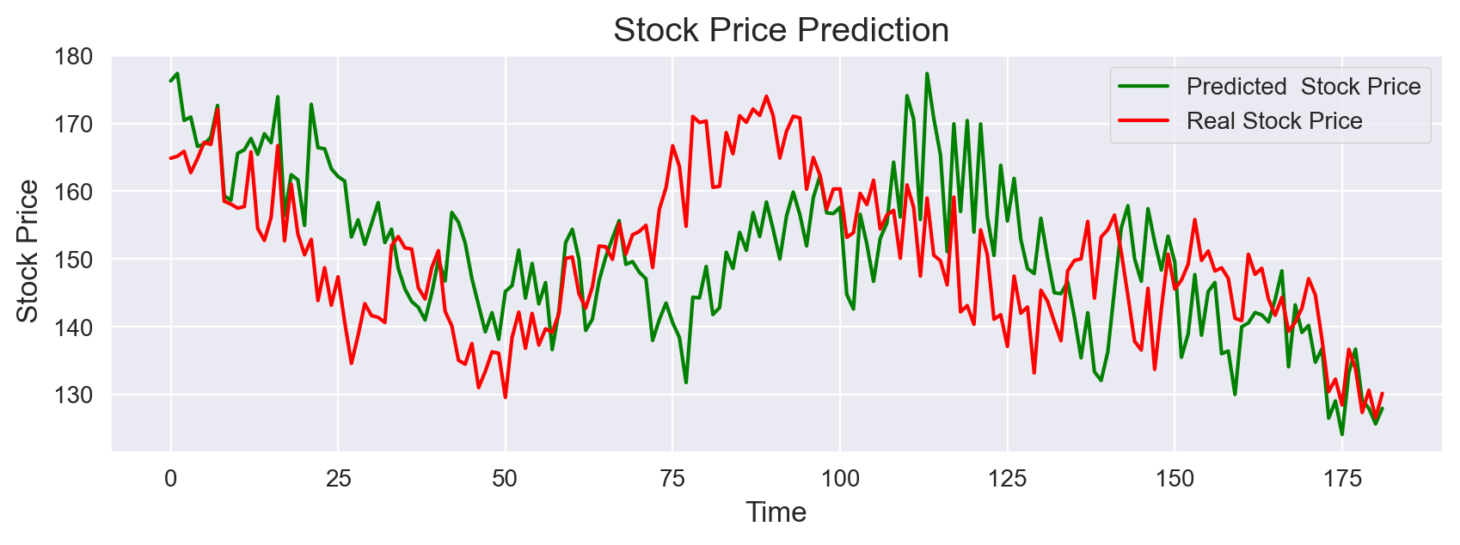
**Analyzing the dataset:**

****



**Predicted Result:**





**Exploratory Data Analysis:**

|  |  |
| --- | --- |
| 00 | 0 |

**Stocks Information Tab:**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

**Time Series Prediction:**

****

**Stocks News:**

****

1. **Conclusion**
   1. **Conclusion**

I conclude that a web application for stock price prediction is developed to let users have a better view on the stock market trend. Five time-series algorithms had been evaluated to find the best prediction models for the web application. CNN-LSTM has the best accuracy among all five time series algorithms, and it has been chosen as the prediction model for the web application for stock price prediction. It has led to the conclusion that it is possible to predict the stock market with more accuracy and efficiency using machine learning techniques. In the future, the stock market prediction system can be further improved by utilizing a much bigger dataset than the one being used currently. This would help increase the accuracy of our prediction models. Furthermore, other models of machine learning could also be studied to check the accuracy rate resulted by them.

* 1. **Future Recommendation**

Future scope of this project will involve adding more parameters and factors like the financial ratios, multiple instances, etc. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the corporate employee. The use of traditional algorithms and data mining techniques can also help predict the corporation performance structure as a whole. Adding up whole company stocks. Designing a system to buy and sell the particular stocks. Providing agent services for new customers to help them in investing their stock. Some improvement can be taken if there are new solutions found to increase the accuracy of the prediction model and decrease the time used to predict future stock prices.

1. **References**
2. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.
3. K. Raza, "Prediction of Stock Market performance by using machine learning techniques," 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT),Karachi, 2017, pp. 1- 1.
4. H. Gunduz, Z. Cataltepe and Y. Yaslan, "Stock market direction prediction using deep neural networks," 2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, 2017, pp. 1-4.
5. C K Gomathy and V Geetha. Article: A Real Time Analysis of Service based using Mobile Phone Controlled Vehicle using DTMF for Accident Prevention.InternationalJournal of Computer Applications 138(2):11-13, March 2016. Published by Foundation of Computer Science (FCS), NY, USA,ISSN No: 0975-8887
6. C K Gomathy and V Geetha. Article: Evaluation on Ethernet based Passive Optical Network Service Enhancementthrough Splitting ofArchitecture.International Journal of ComputerApplications 138(2):14-17, March 2016. Published by Foundation of Computer Science (FCS), NY, USA, ISSN No: 0975-8887
7. Dr.C.K.Gomathy , K. Bindhu sravya , P. Swetha , S.Chandrika Article: A Location Based Value Prediction for Quality of Web Service, Published by International Journal of Advanced Engineering Research and Science (IJAERS),Vol-3, Issue-4 , April- 2016] ISSN: 2349-6495
8. DR.C.K.Gomathy , V.Geetha , S.Madhumitha , S.Sangeetha , R.Vishnupriya Article: A Secure With Efficient Data Transaction In Cloud Service, Published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 4, March 2016, ISSN: 2278 – 1323.
9. C.K.Gomathy.(2010),"Cloud Computing: Business Management for Effective Service Oriented Architecture" International Journal of Power Control Signal and Computation (IJPCSC), Volume 1, Issue IV, Oct - Dec 2010, P.No:22-27, ISSN: 0976-268X
10. Stock Market Prediction using Machine Learning; Prof. S .P. Pimpalkar, Jenish Karia, Muskaan Khan, Satyam Anand, Tushar Mukherjee,International Journal of Advance Engineering and Research Development,Dec 2017. bibitema3Stock Market Analysis: A Review and Taxonomy of Prediction Techniques,Dev Shah, Haruna Isah \* and Farhana Zulkernine,International Journal Of Financial Studies,May 2019.
11. Althelaya, K.A., El-Alfy, E.S.M. and Mohammed, S., 2018. Stock Market Forecast Using Multivariate Analysis with Bidirectional and Stacked (LSTM, GRU). In: 21st Saudi Computer Society National Computer Conference, NCC 2018. Institute of Electrical and Electronics Engineers Inc.
12. Chandra, V., 2015. Comparison between Various Software Development Methodologies. International Journal of Computer Applications, 131(9), pp.7–10.
13. Chau, T.M., Suen, H.P., To, C.L. and Wong, C.K., 2019. Stock Price Prediction App using Machine Learning Models Optimized by Evolution. pp.2018–2019.
14. Cheng, J., Dong, L. and Lapata, M., 2016. Long short-term memory-networks for machine reading. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, pp.551–561.
15. Chowdhury, A.E., Bhowmik, A., Hasan, H. and Rahim, M.S., 2018. Analysis of the Veracities of Industry Used Software Development Life Cycle Methodologies. [online] pp.1–7. Available at.
16. Fatimah, D.D.S., Supriatna, A.D. and Kurniawati, R., 2018. Design of personnel information systems using rapid application development method. MATEC Web of Conferences, 197.
17. Hegazy, O., Soliman, O.S. and Salam, M.A., 2014. A Machine Learning Model for Stock Market Prediction. International Journal for Research in Applied Science and Engineering Technology, 8(6), pp.209–216.
18. Iyawa, G.E., Herselman, M.E. and Coleman, A., 2016. Customer interaction in software development: A comparison of software methodologies deployed in Namibian software firms. Electronic Journal of Information Systems in Developing Countries, 77(1), pp.1–13.
19. **Other Projects Reports/Documents**

**8.1. SRS………………………………………………………………………………………61-70**

**8.2. Software Analysis Report...............................................................................................71-75**

**8.3. Project Management Report…………………………………………………………..76-84**

**8.4. Software Design Report………………………………………………………………..86-93**

**8.5. Software Testing Report……………………………………………………………….94-99**

**8.6. Project Poster……………………………………………………………………………..100**